

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**ANÁLISE DE DESEMPENHO DE CONSULTAS
OLAP ESPAÇO-TEMPORAIS EM FUNÇÃO DA
ORDEM DE PROCESSAMENTO DOS PREDICADOS
CONVENCIONAL, ESPACIAL E TEMPORAL**

CESAR JOAQUIM NETO

ORIENTADOR: PROF. DR. RICARDO RODRIGUES CIFERRI

COLABORADORA: PROFA. DRA. CRISTINA DUTRA DE AGUIAR CIFERRI

São Carlos - SP
Março/2016

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**ANÁLISE DE DESEMPENHO DE CONSULTAS OLAP
ESPAÇO-TEMPORAIS EM FUNÇÃO DA ORDEM DE
PROCESSAMENTO DOS PREDICADOS
CONVENCIONAL, ESPACIAL E TEMPORAL**

CESAR JOAQUIM NETO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Engenharia de Software, Banco de Dados e Interação Humano Computador.
Orientador: Prof. Dr. Ricardo Rodrigues Ciferri
Colaboradora: Profa. Dra. Cristina Dutra de Aguiar Ciferri (USP/ICMC)

São Carlos - SP
Março/2016

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

J623a Joaquim Neto, Cesar
 Análise de desempenho de consultas OLAP espaço-
temporais em função da ordem de processamento dos
predicados convencional, espacial e temporal / Cesar
Joaquim Neto. -- São Carlos : UFSCar, 2016.
 173 p.

 Dissertação (Mestrado) -- Universidade Federal de
São Carlos, 2016.

 1. Data Warehouse. 2. Bancos de dados espaço-
temporais. 3. Data warehouses espaço-temporais. 4.
Índices bitmap. I. Título.



UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a defesa de dissertação de mestrado do candidato Cesar Joaquim Neto, realizada em 08/03/2016.

Prof. Dr. Ricardo Rodrigues Ciferri
(UFSCar)

Prof. Dr. Renato Bueno
(UFSCar)

Profa. Dra. Valéria Cesario Times
(UFPE)

Certifico que a sessão de defesa foi realizada com a participação à distância do membro Profa. Dra. Valéria Cesario Times e, depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da comissão examinadora redigido no relatório de defesa do aluno Cesar Joaquim Neto.

Prof. Dr. Ricardo Rodrigues Ciferri
Presidente da Comissão Examinadora
(UFSCar)

À minha família...

Perto ou longe...

De sangue ou não...

AGRADECIMENTO

A todas as pessoas que estiveram ao meu lado das mais variadas maneiras... Algumas no dia-a-dia, como a minha esposa e meus filhos... vendo o meu esforço e entendendo que as horas passadas trabalhando são precisas para crescer mais e mais...

Algumas mais longe, como minha mãe, minha irmã e minha sogra... sabendo que não tenho dado tanta atenção a elas por tentar uma conquista pessoal...

Algumas somente em pensamento pois já se foram, mas ainda moram no meu coração e na minha mente...

Algumas liderando o caminho, como os professores que me acolheram no PPGCC e especialmente o meu orientador e sua esposa... pois já o percorreram e sabem o que é preciso fazer e o que é preciso evitar...

Algumas ao meu lado na caminhada pelas trilhas do conhecimento e que entendem o que tenho passado nestes anos por estarmos juntos com objetivos parecidos...

Quando me casei, eu e minha esposa pedimos a visita do padre que nos casou para abençoar também a nossa casa. Ele nos falou uma coisa que me marcou muito: "Mais do que abençoar as coisas, devemos abençoar as pessoas, pois elas é o que realmente importa na nossa caminhada"

Me sinto abençoado por vocês todos os dias da minha vida e agradeço toda a companhia, sabedoria e entendimento que me passaram

Muito obrigado!

SEGUNDO AGRADECIMENTO

À IBM Brasil Indústria Máquinas e Serviços.

A IBM sempre foi conhecida por ser uma empresa séria e comprometida com os seus clientes e o sucesso destes.

Isso não aconteceria se ela não fosse séria e comprometida com os seus funcionários e o sucesso destes dentro e fora dela.

Como sempre é falado internamente, isso está no DNA da empresa.

Eu senti isso no meu dia a dia e atesto – com esta pesquisa – que é verdade.

A flexibilidade em aceitar pessoas trabalhando de suas casas já existia.

Foram flexíveis para aceitar mudanças no meu horário de trabalho para que eu pudesse participar das aulas, reuniões na universidade e até um congresso.

Só me pediram que eu tivesse a mesma seriedade ao entregar meu trabalho aos clientes que atendo.

Tal visão de negócio só se concretiza quando há pessoas que a entendem e defendem... como eu... como meus colegas de projeto... como meus gerentes em qualquer nível.

Muito obrigado!

Posso até demorar... mas chego onde quero...

Autoria própria.

RESUMO

Por proverem uma capacidade de processamento de dados cada vez maior, várias tecnologias de bancos de dados têm se tornado importantes ferramentas de apoio a empresas e instituições. A necessidade de se incluir e controlar novos tipos de dados aos bancos de dados já existentes fizeram também surgir novos desafios e novas linhas de pesquisa, como é o caso dos bancos de dados espaciais, temporais e espaço-temporais. Além disso, novas capacidades analíticas foram se fazendo necessárias culminando com o surgimento dos data warehouses e, mais uma vez, com a necessidade de se incluir dados espaciais e temporais (ou ambos) surgindo os data warehouses espaciais, temporais e espaço-temporais. As consultas relacionadas a cada tipo de banco de dados também evoluíram culminando com as consultas STOLAP (*Spatio-Temporal OLAP*) que são compostas basicamente por predicados envolvendo dados convencionais, espaciais e temporais e cujo processamento pode ser auxiliado por estruturas de indexação especializadas. Este trabalho pretende investigar como a execução de cada um dos tipos de predicados afeta o desempenho de consultas STOLAP variando-se os índices utilizados, a ordem de execução dos predicados e a seletividade das consultas. Índices Bitmap de Junção auxiliarão na execução dos predicados convencionais e de algumas partes dos predicados temporais que também contarão com o auxílio de consultas SQL, enquanto os índices SB-index e HSB-index serão utilizados para auxiliar na execução dos predicados espaciais das consultas. O STB-index também será utilizado nas comparações e envolve ambos os predicados espacial e temporal. Espera-se obter uma análise das melhores opções de combinação de execução dos predicados em consultas STOLAP tendo em vista também a seletividade das consultas. Outra contribuição deste trabalho é a evolução do HSB-index para uma versão hierarquizada chamada HSTB-index e que servirá para complementar as opções de processamento de consultas STOLAP.

Palavras-chave: Data Warehouse, Bancos de Dados Espaço-Temporais, Data Warehouses Espaço-Temporais, Índices Bitmap.

ABSTRACT

By providing ever-growing processing capabilities, many database technologies have been becoming important support tools to enterprises and institutions. The need to include (and control) new data types to the existing database technologies has brought also new challenges and research areas, arising the spatial, temporal, and spatio-temporal databases. Besides that, new analytical capabilities were required facilitating the birth of the data warehouse technology and, once more, the need to include spatial or temporal data (or both) to it, thus originating the spatial, temporal, and spatio-temporal data warehouses. The queries used in each database type had also evolved, culminating in the STOLAP (*Spatio Temporal OLAP*) queries, which are composed of predicates dealing with conventional, spatial, and temporal data with the possibility of having their execution aided by specialized index structures. This work's intention is to investigate how the execution of each predicate affects the performance of STOLAP queries by varying the used indexes, their execution order and the query's selectivity. Bitmap Join Indexes will help in conventional predicate's execution and in some portions of the temporal processing, which will also count with the use of SQL queries for some of the alternatives used in this research. The SB-index and HSB-index will aid the spatial processing while the STB-index will be used to process temporal and spatial predicates together. The expected result is an analysis of the best predicate order while running the queries also considering their selectivity. Another contribution of this work is the evolution of the HSB-index to a hierarchized version called HSTB-index, which should complement the execution options.

Keywords: Data Warehouse, Spatio-Temporal Databases, Spatio-Temporal Data Warehouses, Bitmap Indexes.

LISTA DE FIGURAS

Figura 1: Representação gráfica de alguns dos relacionamentos topográficos	22
Figura 2: Exemplo de IRQ, ERQ e CRQ (adaptado de Gaede e Günther, 1998).....	24
Figura 3: Esquemas Estrela e Floco de Neve	30
Figura 4: Treliça de dependência entre as visões materializadas das dimensões Part, Supplier e Customer – adaptado de Harinarayan (1996)	31
Figura 5: Exemplo de consulta MDX	32
Figura 6: Exemplo de construção de IBJs – adaptado de Siqueira (2009) e O’Neil e Graefe (1995)	39
Figura 7: Exemplo da técnica de <i>binning</i> – adaptada de Wu, Stockinger e Shoshani (2008).....	40
Figura 8: Exemplo de compressão – adaptado de Wu, Otoo e Arie (2006)	41
Figura 9: Exemplo de codificação de índices binários – adaptado de Wu e Buchmann (1998).....	42
Figura 10: Exemplo de aR-Tree adaptado de Papadias et al. (2001)	44
Figura 11: Estrutura do SB-index – adaptado de Siqueira (2009)	45
Figura 12: Etapas de processamento do SB-index - adaptado de Siqueira (2009)...	46
Figura 13: Estrutura do HSB-index (adaptado de Siqueira et al.; 2012).....	47
Figura 14: Cubo de dados e uma possível janela de consulta sobre estes para os exemplos da a3DR-tree e aRB-tree – adaptada de Papadias et al. (2002)	49
Figura 15: a3DR-tree resultante do exemplo proposto.....	49
Figura 16: aRB-tree resultante da agregação do cubo de dados de exemplo – adaptada de Papadias et al. (2002)	50
Figura 17: Estrutura do STB-index – adaptado de Tsuruda (2013).....	52
Figura 18: Estrutura do HSTB-index	53
Figura 19: Esquema de dados do Spatial Star Schema Benchmark - retirado de Nascimento et. al. (2011)	55
Figura 20: Consulta Q3.1 do Spatial Star Schema Benchmark – retirada de Nascimento et. al. (2011)	56
Figura 21: Exemplo de consulta STOLAP a ser utilizada - adaptado de Tsuruda (2013).....	62

Figura 22: Esquema estrela a ser utilizado no desenvolvimento da pesquisa – adaptado de Tsuruda (2013)	66
Figura 23: Consulta Q1.2 do Benchmark Spatial SSB modificada para se tornar uma consulta STOLAP	67
Figura 24: Consulta Q1.3 do Benchmark Spatial SSB modificada para se tornar uma consulta STOLAP	68
Figura 25: Consulta Q3.1 do Benchmark Spatial SSB modificada para se tornar uma consulta STOLAP	68
Figura 26: Possíveis casos temporais e a respectiva decisão de inclusão ou exclusão do registro.	72
Figura 27: Descrição dos índices usados na visão materializada altamente indexada	73
Figura 28: Técnicas de processamento em consultas STOLAP e a sua representação gráfica.....	76
Figura 29: Opções em <i>Pipeline</i> com execução do Predicado Espacial a priori.....	77
Figura 30: Opções em Pipeline com execução do Predicado Temporal a priori	78
Figura 31: Opções em Pipeline com execução dos Predicados Temporal e Espacial num mesmo passo	79
Figura 32: Opções de processamento Independente com execução dos Predicados Temporal, Espacial e Convencional separadamente.	80
Figura 33: Opções de processamento Independente com execução dos Predicados Temporal e Espacial em conjunto e do Predicado Convencional em separado.	81
Figura 34: Alteração da execução dos índices Espaço-Temporais para contemplar a poda temporal entre as fases de filtragem e refinamento espaciais - adaptado de Siqueira (2012).....	83
Figura 35: Representação da opção intercalada baseada no STB-index (pipeTEs ..	84
Figura 36: Representação da opção intercalada baseada no HSTB-index (pipeTEhstb).....	84
Figura 37: Resumo das opções de processamento implementadas, suas classes e as respectivas cores utilizadas nas análises	85
Figura 38: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S0-caddress	89
Figura 39: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S0-saddress	93

Figura 40: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S0-street.....	95
Figura 41: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S0-city	98
Figura 42: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S1-caddress	103
Figura 43: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S1-saddress	106
Figura 44: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S1-street.....	109
Figura 45: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S1-city	112
Figura 46: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S2-caddress	116
Figura 47: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S2-saddress	120
Figura 48: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S2-street.....	123
Figura 49: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S2-city	126
Figura 50: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S3-caddress	129
Figura 51: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S3-saddress	131
Figura 52: Desempenho das opções de processamento com relação ao <i>baseline</i> de visão materializada e com relação à opção de processamento vencedora – S3-street.....	134

Figura 53: Desempenho das opções de processamento com relação ao *baseline* de opção materializada e com relação à opção de processamento vencedora – S3-city.....137

LISTA DE TABELAS

Tabela 1: Exemplo de registros com tempos de validade e transação de um TDB ..	26
Tabela 2 - Comparativo entre as características do trabalho correlato a esta dissertação de mestrado	59
Tabela 3: Configurações de processamento a serem investigadas	70
Tabela 4 - Quantidade total de registros nas tabelas envolvidas nas consultas	70
Tabela 5 - Consultas com suas respectivas seletividades e porcentagens de seletividade temporal, espacial, espaço-temporal e convencional	71
Tabela 6: Comparação entre as duas opções de visão materializada utilizadas	73
Tabela 7: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora – S0-address.....	88
Tabela 8: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S0-saddress	92
Tabela 9: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S0-street.....	95
Tabela 10: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora – S0-city .	98
Tabela 11: Tempos médios em cada predicado e tempo total das opções vencedoras (S0-city)	100
Tabela 12: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S1-caddress	102
Tabela 13: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora – S1-saddress.....	105
Tabela 14: Tempos obtidos por cada opção de processamento vencedora na opção de processamento S1-saddress.....	107
Tabela 15: – Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S1-street.....	109

Tabela 16: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S1-city.....	112
Tabela 17: Tempos médios de processamento das opções indpTpEsbCf e indpTpEhsbCf.....	114
Tabela 18: Tempos médios de processamento das opções indpTfEsbCf e indpTfEhsbCf.....	114
Tabela 19: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S2-caddress	116
Tabela 20: Tempos médios de processamento das opções da classe <i>Pipeline</i> T-E-C para a configuração S2-caddress.....	117
Tabela 21: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S2-saddress	120
Tabela 22: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S2-street.....	122
Tabela 23: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S2-city.....	125
Tabela 24: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S3-caddress	129
Tabela 25: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S3-saddress	131
Tabela 26: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora (melhor desempenho) – S3-street.....	134
Tabela 27: Comparação das opções de processamento com relação ao <i>baseline</i> de visão materializada e à opção de processamento vencedora – S3-city	136
Tabela 28: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para caddress com todas as seletividades	141
Tabela 29: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para saddress com todas as seletividades	144

Tabela 30: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para street com todas as seletividades.....	147
Tabela 31: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para city com todas as seletividades	149
Tabela 32: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para a seletividade muito baixa (S0) com todos os tipos de dados.....	153
Tabela 33: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para a seletividade baixa (S1) com todos os tipos de dados.....	155
Tabela 34: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para a seletividade média (S2) com todos os tipos de dados.....	157
Tabela 35: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para a seletividade alta (S3) com todos os tipos de dados.....	159

LISTA DE ABREVIATURAS E SIGLAS

BDNC - Bancos de Dados Não Convencionais

BDE - Banco de Dados Espaciais

CRQ - *Containment Range Query*

DW - *Data Warehouse*

DWE - Data Warehouse Espacial

DWET - Data Warehouse Espaço-Temporal

DWT - Data Warehouse Temporal

ERQ - *Enclosure Range Query*

ETL - *Extract, Transform and Load*

GPS - *Global Positioning System*

HSB-index - *Hierarchical Spatial Bitmap index*

IBJ - Índice Bitmap de Junção

IRQ - *Intersection Range Query*

OLTP – *On-Line Transactional Processing*

OLAP – *On-Line Analytical Processing*

MDX - *Multi-Dimensional eXtensions*

PCP - Planejamento e Controle da Produção

REM - Retângulo Envolvente Mínimo

RFID - *Radio Frequency Identification*

ROLAP - *Relational OLAP*

ROT - Relação de Ordem Total

SB-index - *Spatial Bitmap index*

SIG - Sistemas de Informações Geográficas

SOLAP - *Spatial OLAP*

Spatial SSB - *Spatial Star Schema Benchmark*

STB-index - *Spatio-Temporal Bitmap index*

STOLAP - *Spatio-Temporal OLAP*

TDB - *Temporal Databases*

TOLAP - *Temporal OLAP*

XML - *eXtensible Markup Language*

WAH - *Word Aligned Hybrid code*

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO.....	13
1.1 Contexto	13
1.2 Motivação.....	14
1.3 Objetivos	15
1.4 Organização	15
CAPÍTULO 2 - FUNDAMENTAÇÃO TEÓRICA.....	17
2.1 Banco de Dados Não Convencionais	17
2.1.1 Exemplo Proposto	19
2.2 Banco de Dados Espaciais.....	20
2.2.1 Análise do exemplo proposto utilizando-se BDEs	25
2.3 Banco de Dados Temporais	25
2.3.1 Análise do exemplo proposto utilizando-se BDTs	28
2.4 Data Warehouse	28
2.4.1 Análise do exemplo proposto utilizando-se DWs	32
2.5 Data Warehouse Espacial	32
2.5.1 Análise do exemplo proposto utilizando-se DWEs	33
2.6 Data Warehouse Temporal	33
2.6.1 Análise do exemplo proposto utilizando-se DWTs	34
2.7 Data Warehouse Espaço-Temporal	34
2.7.1 Análise do exemplo proposto utilizando-se DWETs.....	35
2.8 Considerações Finais	36
CAPÍTULO 3 - PROCESSAMENTO DE CONSULTAS ANÁLITICAS	37
3.1 Junção Estrela e Visões Materializadas	37
3.2 Índices Bitmap de Junção	38
3.2.1 Binning	39
3.2.2 Compressão	41
3.2.3 Codificação.....	42
3.3 Índices Espaciais para Data Warehouse.....	43
3.3.1 aR-Tree	43

3.3.2 SB-index.....	45
3.3.3 HSB-index	47
3.4 Índices Espaço-Temporais para Data Warehouse	48
3.4.1 a3DR-tree.....	48
3.4.2 aRB-tree	50
3.4.3 STB-index.....	52
3.4.4 HSTB-index	53
3.5 Benchmark Spatial SSB	54
3.5.1 Considerações Finais	56
CAPÍTULO 4 - TRABALHOS CORRELATOS	57
4.1.1 Especificação e Avaliação de Algoritmos para Processamento Multidimensional-Geográfico	57
4.1.2 Diferenças e Contribuições	59
4.1.3 Considerações finais	60
CAPÍTULO 5 - ATIVIDADES DE PESQUISA.....	61
5.1 Objetivo	61
5.2 Hipóteses	63
5.3 Metodologia.....	65
5.3.1 Carga de trabalho.....	65
5.3.2 Ambiente Computacional e Medidas de Desempenho.....	74
5.4 Técnicas de Processamento	74
5.4.1 Processamento em Pipeline dos Predicados Espacial, Temporal e Convencional.....	76
5.4.2 Processamento Independente dos Predicados Espacial, Temporal e Convencional.....	79
5.4.3 Processamento Intercalado dos Predicados Espacial e Temporal.....	82
5.4.4 Resumo das Configurações para o Processamento dos Predicados Espacial, Temporal e Convencional.....	84
5.4.5 Considerações Finais	85
CAPÍTULO 6 - ANÁLISE DOS DADOS.....	86
6.1 Análise das Opções de Processamento em Função da Seletividade Espaço- Temporal e do Tipo de Dado Espacial	86

6.1.1	Objetivo da Análise.....	86
6.1.2	Análise da Configuração S0-caddress	87
6.1.3	Análise da Configuração S0-saddress	90
6.1.4	Análise da Configuração S0-street”	94
6.1.5	Análise da Configuração S0-city.....	97
6.1.6	Análise da Configuração S1-caddress	101
6.1.7	Análise da Configuração S1-saddress	104
6.1.8	Análise da Configuração S1-street.....	107
6.1.9	Análise da Configuração S1-city.....	110
6.1.10	Análise da Configuração S2-caddress	114
6.1.11	Análise da Configuração S2-saddress	118
6.1.12	Análise da Configuração S2-street.....	121
6.1.13	Análise da Configuração S2-city.....	124
6.1.14	Análise da Configuração S3-caddress	127
6.1.15	Análise da Configuração S3-saddress	130
6.1.16	Análise da Configuração S3-street.....	132
6.1.17	Análise da Configuração S3-city.....	135
6.1.18	Conclusões.....	138
6.2	Análise da Influência da Seletividade no Desempenho Relativo das Técnicas de Processamento	139
6.2.1	Objetivo da Análise.....	139
6.2.2	Análise das Configurações Envolvendo Pontos de Clientes (caddress)	139
6.2.3	Análise das Configurações Envolvendo Pontos de Fornecedores (saddress)	142
6.2.4	Análise das Configurações Envolvendo Linhas de Ruas (street)	145
6.2.5	Análise das Configurações Envolvendo Polígonos de Cidades (city).....	148
6.2.6	Conclusões.....	150
6.3	Análise da Influência do Tipo de Dado Espacial no Desempenho das Opções de Processamento	151
6.3.1	Objetivo da Análise.....	151
6.3.2	Análise das Configurações Envolvendo Seletividade Muito Baixa (S0)	152
6.3.3	Análise das Configurações Envolvendo Seletividade Baixa (Seletividade S1)	154
6.3.4	Análise das Configurações Envolvendo Seletividade Média (S2)	156

6.3.5 Análise das Configurações Envolvendo Seletividade Alta (S3)	158
6.3.6 Conclusões.....	160
CAPÍTULO 7 - CONCLUSÕES.....	162
7.1 Considerações Finais	162
7.2 Contribuições e Limitações	166
7.3 Trabalhos futuros.....	167
CAPÍTULO 8 - REFERÊNCIAS	169

Capítulo 1

INTRODUÇÃO

*Diante do enorme volume de dados disponível atualmente, empresas e instituições têm utilizado várias tecnologias para analisarem os seus dados com maior eficiência e precisão. A utilização de dados **não convencionais** também é um desejo e, muitas vezes, encontrar a maneira correta para utilizá-los representa um desafio a ser transposto. Este capítulo apresenta o **contexto**, a **motivação** e os **objetivos** desta pesquisa de mestrado e, por fim, a **organização** desta dissertação.*

1.1 Contexto

Para atender à necessidade das empresas em conhecer melhor os seus negócios e como as suas decisões influenciam nos resultados obtidos, o conceito de **Data Warehouse** surgiu a partir da convergência de tecnologias já existentes e se popularizou por permitir uma rápida análise dos dados para a tomada de decisão estratégica. O processamento de consultas foi alterado devido à passagem da natureza transacional dos dados (OLTP – *On-Line Transactional Processing*), a qual é caracterizada pela pouca quantidade de tuplas nas seleções e grande quantidade de operações de inserção, remoção e atualização, para uma natureza analítica (OLAP – *On-Line Analytical Processing*) – caracterizada por uma grande quantidade de dados e enfoque na seleção (KIMBALL e ROSS, 2002) (INMON, 2005).

O avanço da tecnologia também trouxe novos tipos de dados, tais como documentos XML, imagens de todos os tipos, dados de natureza espacial (como a representação de mapas na sua forma geométrica) e temporal (para permitir a manutenção do histórico de alterações nos dados e no esquema) (ELMASRI e

NAVATHE, 2005). A incorporação destes novos tipos de dados aos bancos de dados já existentes fez com que estruturas já consolidadas (por exemplo, índices e formas de armazenamento) precisassem ser repensadas cunhando o termo **Bancos de Dados Não Convencionais** (BDNCs).

O movimento natural foi a incorporação de dados não convencionais também às estruturas utilizadas em Data Warehouses, onde as tabelas de dimensão se alteraram para permitir a inclusão de dados espaciais (gerando os *Data Warehouses Espaciais* – ou DWEs), a inclusão de dados temporais (gerando os *Data Warehouses Temporais* – ou DWTs), a inclusão de ambos os dados espaciais e temporais (gerando os *Data Warehouses Espaço-Temporais* – ou DWETs) e a inclusão de dados de imagens (gerando os *Data Warehouses de Imagens* – ou DWIs). As consultas também foram se modificando ao incorporar dados não convencionais, originando as consultas SOLAP (*Spatial OLAP*), TOLAP (*Temporal OLAP*), STOLAP (*Spatio-Temporal OLAP*) e IOLAP (*Image OLAP*) (RIVEST, BÉDARD e MARCHAND, 2001) (MENDELZON e VAISMAN, 2000) (VAISMAN e ZIMÁNYI, 2009). Consultas STOLAP possuem por característica a presença de predicados convencional, espacial e temporal, sendo o foco desta pesquisa.

1.2 Motivação

A motivação desta pesquisa de mestrado está na consideração de três fatores. O primeiro deles foi a recente implementação do índice espaço-temporal STB-index feito por Tsuruda (2013) que indicou a melhoria no desempenho para determinadas consultas STOLAP. O segundo foi o trabalho de Siqueira et al. (2012) que introduziu o HSB-index e sinalizou a existência de um nível de seletividade espacial onde a execução das consultas usando um índice baseado em hierarquia (cuja característica principal é a poda de registros) já não seria mais vantajosa frente ao uso de um índice baseado em vetor (o SB-index, no caso). O terceiro fator foi o trabalho de Medeiros et al. (2008) que investigou a ordem de processamento de consultas SOLAP em DWEs e mostrou poder haver diferenças no desempenho das consultas quando variamos a ordem de processamento dos seus predicados. Com esses fatos em mente, considerou-se conduzir neste trabalho uma investigação

sobre consultas STOLAP considerando-se a variação na ordem de execução dos predicados e das seletividades envolvidas, a fim de: i) analisar e apontar situações onde há queda de desempenho; ii) identificar se os pontos de ruptura entre certas técnicas de processamento apontados anteriormente ainda são válidos para consultas STOLAP; iii) investigar como a ordem de processamento dos predicados pode ajudar ou atrapalhar o resultado final do processamento. Tais averiguações podem ser utilizadas por outros pesquisadores na área, bem como por empresas que usem (ou venham a usar) DWETs nos seus negócios.

1.3 Objetivos

Esta pesquisa de mestrado tem como objetivo investigar a influência da ordem de execução dos predicados convencional, espacial e temporal no desempenho de consultas STOLAP. Além disto, será investigado como a seletividade dos predicados (i.e. baixa ou alta seletividade) pode afetar a escolha da ordem de execução dos predicados.

Pretende-se, portanto, investigar algumas das possibilidades de execução dos três tipos de predicados encontrados em consultas STOLAP (variando-se a ordem de execução bem como a seletividade dos predicados) e saber em quais situações uma certa ordem de execução é preferível. O segundo objetivo é propor e implementar a evolução do índice HSB-index proposto por Siqueira et al. (2012) incluindo a data de validade dos MBRs na estrutura do índice e incluindo a checagem temporal no processamento do mesmo.

1.4 Organização

Esta dissertação está organizada da seguinte maneira:

- Capítulo 2: são descritos os conceitos fundamentais para o entendimento desta dissertação.

- Capítulo 3: discorre sobre como é feito o processamento de consultas analíticas e introduz os índices a serem utilizados na pesquisa.
- Capítulo 4: os trabalhos correlatos são analisados.
- Capítulo 5: as atividades de pesquisa são detalhadas como a metodologia utilizada no trabalho, os objetivos que se esperam alcançar, hipóteses, contribuições e limitações desta pesquisa.
- Capítulo 6: os resultados de desempenho são analisados.
- Capítulo 7: as conclusões e considerações finais são feitas.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

As bases teóricas permeando esta pesquisa de mestrado serão apresentadas neste capítulo. O ponto de partida são os conceitos mais básicos sobre Banco de Dados Não Convencionais e de onde surgiu a necessidade destes bancos de dados, passando pelos conceitos de banco de dados espaciais e temporais. Será apresentado como as estruturas evoluíram para os Data Warehouses até a necessidade de se incluir dados espaciais e temporais aos mesmos. Tal inclusão se deu inicialmente em separado até que os dois conceitos foram mesclados dando-se origem aos Data Warehouses Espaço-Temporais.

2.1 Banco de Dados Não Convencionais

A teoria de bancos de dados convencionais vem se desenvolvendo ao longo do tempo e já está bem consolidada comercialmente com muitas implementações e muitas empresas e instituições utilizando os seus conceitos. Seus índices e consultas se valem do conceito de Relação de Ordem Total (ROT) inerente aos tipos de dados que podem abrigar.

Podemos definir Relação de Ordem Total como um conjunto de estados que podem ser ordenados de acordo com uma certa propriedade inerente a um domínio de dados. Mais formalmente, Rakoto-Ravalontsalama (2001) faz a seguinte definição:

Definição 1 (Relação de Ordem Total $<_1$). Uma relação de ordem total $<_1$ (“menor que”) é definido como um conjunto de estados $Q = \{q_1, q_2, q_3, \dots, q_n\}$ tal que para $i = 1 \dots n$,

$$q_1 <_1 q_2 <_1 q_3 <_1 \dots <_1 q_n$$

Se tomarmos como exemplo uma coluna utilizando os números inteiros como tipo de dados, verifica-se, que dado um conjunto exemplo $Q = \{7, 3, 58, 22\}$, este pode ser ordenado de acordo com a definição acima de maneira que $3 <_1 7 <_1 22 <_1 58$ e consultas podem ser executadas de maneira mais rápida. Este mesmo exercício de ordenação pode ser feito com números reais, data, hora e texto.

Com a evolução das necessidades dos aplicativos, novos tipos de dados (espaciais, imagens, sons, vídeos, biológicos, documentos XML, para citar alguns) precisaram ser introduzidos em bancos de dados convencionais, mas eles se tornaram um desafio por não possuírem ROTs. Como determinar se uma imagem antecede outra? Como ordenar regiões, linhas e pontos de maneira que faça sentido em um espaço Euclidiano n-dimensional? Como fazer comparações entre documentos XML? Vale ressaltar que, apesar de utilizar colunas do tipo data em sua construção (que possuem ROT), bancos de dados temporais são considerados Bancos de Dados Não Convencionais (BDNCs), pois o intuito das datas de validade transacional e de negócio é conferir histórico aos registros e assim tornar possível capturar as alterações realizadas em um banco de dados ao longo tempo. Tudo muda novamente quando são também capturadas as alterações de esquema de uma base pois estas precisam ser registradas em estruturas separadas. Tais colunas e estruturas podem ser consideradas metadados e a ordenação que elas seguem apenas influi na consideração ou não de uma tupla para uma determinada consulta.

Bancos com tais tipos de dados começaram a ser chamados de Bancos de Dados Não Convencionais e novas linhas de pesquisa buscando alcançar maneiras de tratar eficientemente estes dados foram criadas. Na área de Bancos de dados Espaciais (ou SIG – “Sistemas de Informações Geográficas”), Gaede e Günther (1998) fazem um resumo de mais de 50 técnicas de indexação de dados geográficos através do uso de Métodos de Acesso Multidimensionais, enquanto Carélo et al. (2011) apresentam um refinamento de uma das técnicas utilizando Métodos de

Acesso Métricos chamada *Onion-tree* e Chávez et al (2001) propõem-se a concentrar e classificar os vários trabalhos na área de Métodos de Acesso Multidimensionais uma vez que, na opinião dos autores, muitos deles estavam implementando os mesmos conceitos mas em áreas distintas de aplicação. Para Bancos de dados Temporais (ou TDBs – do inglês “*Temporal Data Bases*”), Kline (1993) faz o quinto de uma série de artigos tentando agrupar as pesquisas encontradas na área e lista 313 referências encontradas após o quarto artigo e que mostram o crescimento da área. Ainda sobre TDBs, Jensen et al. (1994) percebem a grande quantidade de novos termos sendo cunhados pelos pesquisadores da área e preparam, com a ajuda destes, um glossário de termos relacionados a este tipo de banco de dados. Dados espaciais e dados temporais aplicados em data warehouses são o foco deste trabalho.

2.1.1 Exemplo Proposto

Buscando melhor expor as necessidades de um BDNC, o exemplo abaixo será usado ao longo desta dissertação.

“Pensando em como melhorar o gerenciamento da evolução de seus produtos, a companhia XYZ deseja automatizar o ciclo de vida dos seus desenhos de engenharia. Tal ciclo de vida começa com a criação do projeto do produto que contém o desenho de engenharia em si, a lista de materiais utilizados em sua fabricação e as características específicas do produto (como tensão de alimentação, temperatura de trabalho, peso, cuidados de conservação, entre outros). Numa segunda fase, o produto é liberado para produção e sua estrutura passa a ser visível em outras áreas da empresa como PCP (Planejamento e Controle da Produção), vendas, expedição e contabilidade. Na parte de vendas, pretende-se ter um controle acurado dos centros de distribuição de acordo com sua distribuição geográfica para facilitar a logística de venda e entrega do produto. Após cumprir seu tempo em produção, o produto é então substituído por uma versão melhorada ou simplesmente deixa de ser produzido, mas a empresa ainda precisa que este permaneça no sistema para fins históricos”.

Percebe-se que o exemplo acima possui características convencionais (como as características do produto), espaciais (o controle de distribuição e venda por

regiões geográficas) e temporais (a clara distinção entre as fases de projeto, produção e histórico alterando visibilidade do produto).

Em um banco de dados convencional, o controle e distribuição das mercadorias precisaria de classificações e cálculos baseados em valores fixos, sendo que os cálculos de área de cobertura ficariam aquém das necessidades pretendidas pela empresa.

Já na parte temporal, fazer o produto visível a outras áreas somente quando o desenho está pronto e aprovado dependeria de muitos controles (*flags*) ou até mesmo de estruturas de dados separadas (uma tabela ou mesmo um banco de dados para produtos em desenvolvimento, outro para produtos em produção e um terceiro para produtos históricos). Além disso, novos centros de distribuição precisariam estar visíveis a priori para configuração e formação de estoque, mas não serem considerados como passíveis de vendas, assim como os antigos ainda precisariam de um tempo a mais de visibilidade antes de serem desativados.

SIGs e TDBs (vistos a seguir) podem dar respostas melhores para estes desafios sem a necessidade de soluções intrincadas como as citadas anteriormente.

2.2 Banco de Dados Espaciais

Por não possuírem ROT, dados relacionados a eventos e posições geográficas tinham historicamente sua representação exageradamente simplificada para poderem ser utilizados em bancos de dados convencionais. Perguntas como: “Qual a loja mais próxima da minha localização num raio de 10km?”, “Quais as cidades cobertas pelo meu sistema de entrega considerando uma distância máxima de 15km da loja mais próxima?” ou “Qual é a menor rota para transferir os produtos entre três lojas considerando-se as estradas existentes na região?” ficam muito difíceis de responder utilizando bancos de dados convencionais. Buscando melhorar tal representação, os pesquisadores começaram a incorporar tipos de dados baseados em elementos de cartografia como pontos, linhas e geometrias apoiados por um sistema de coordenadas e escala apropriados (CÂMARA, CASANOVA, *et al.*, 1996). A generalização desse tipo de representação pode também ser utilizada para descrever outros tipos de fenômenos não necessariamente relacionados a

dados geo-referenciados gerando-se o termo dados espaciais e o seu banco de dados relacionado.

Câmara et al. (1996) também descreve algumas das maneiras como estes elementos vetoriais de um SIG (pontos, linhas e polígonos) podem se relacionar topologicamente gerando relacionamentos específicos como “à esquerda de”, “à direita de”, “acima”, “próximo a”, “longe de”, “disjunto”, “encontram”, “igual”, “contém”, “cruzando”, entre tantos outros. Alguns deles são exemplificados na Figura 1 e a implementação computacional de tais relacionamentos é a base das linguagens de consulta em SIGs.

Os algoritmos para processar tais relacionamentos não são triviais como mostrado por Casanova et al. (2005) no capítulo 2, e começaram a impactar o desempenho dos SIGs, demandando novas estruturas de índices e armazenamento dos dados.

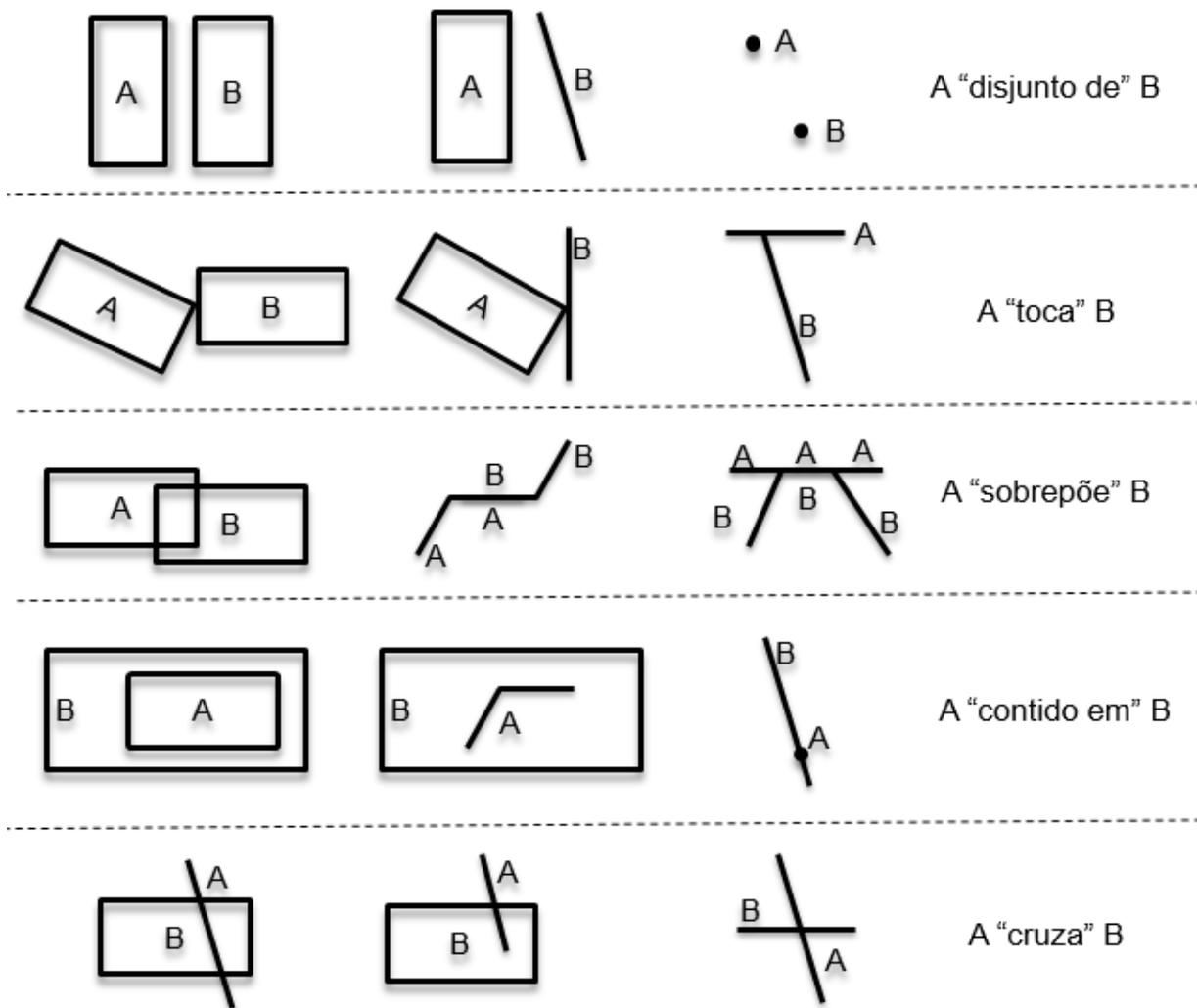


Figura 1: Representação gráfica de alguns dos relacionamentos topográficos

A solução destes problemas é discutida pelos mesmos autores, onde são apresentadas técnicas de classificação das geometrias para posterior armazenamento no disco. São apresentadas a *k-d tree* (BENTLEY, 1975) que implementa uma estrutura de árvore binária para a organização dos registros de acordo com sua ordem de chegada e sua posição geográfica em relação aos elementos já inseridos, o *Grid File* (NIEVERGELT, HINTERBERGER e SEVCIK, 1984) que utiliza técnicas de *hashing* para particionar o espaço métrico em quadrados sobre a superfície da Terra e utilizá-los como *buckets*, a *Quadtree* (SAMET, 1984) que monta uma árvore com divisões sucessivas sempre em quatro partes até que todos os pontos dentro do espaço possam ser representados unicamente em um nó folha e a *R-tree* (GUTTMAN, 1984) que utiliza Retângulos Envolventes Mínimos (REM) englobando todos os elementos de cada nível. As

estruturas acima, à exceção da *R-Tree* que já utiliza REMs, podem ser generalizadas para também armazenar REMs e, assim, indexar todos os tipos de geometrias.

A estratégia para o processamento de consultas também é discutida por Casanova et al. (2005). Com as novas técnicas de indexação e armazenamento citadas acima, um processamento em três fases se faz necessário. A primeira fase é chamada de filtragem e consiste no acesso do índice e eliminação das geometrias que, com certeza, não fazem parte da resposta pela simples poda de nós da árvore ou exclusão por *buckets*. Apesar de acelerar o processo de execução, a simplificação do processo feito através dos REMs no índice pode incluir falsos candidatos à resposta. A segunda fase faz o refinamento destes falsos candidatos e consiste no acesso da geometria em si e cálculo da relação topológica especificada na consulta para que restem somente as geometrias realmente pertencentes ao conjunto resposta. O pós-processamento consiste no acesso em disco aos dados para retorno à camada superior do banco de dados e é a terceira e última fase.

Quanto aos tipos de consultas que podem ser processadas, Gaede e Günther (1998) apresentam em seu trabalho vários tipos de consultas, entre elas *Intersection Range Query*, *Enclosure Range Query* e *Containment Range Query*. Tomando-se em consideração que todos os objetos espaciais estão inclusos em um espaço Euclidiano d -dimensional (E^d) ou em um subespaço deste e chamando de $o.G$ o atributo geográfico de um objeto o contendo também outros atributos não-geográficos, define-se:

Intersection Range Query (IRQ): Dado um intervalo d -dimensional $I^d = [l_1, u_1] \times [l_2, u_2] \times \dots \times [l_d, u_d]$, encontre todos os objetos o tendo pelo menos um ponto em comum com I^d .

$$IRQ(i^d) = \{o | I^d \cap o.G \neq \emptyset\}$$

Enclosure Range Query (ERQ): dado um objeto o' com extensão espacial $o'.G \subseteq E^d$, encontre todos os objetos o contendo o' .

$$ERQ(o') = \{o | (o'.G \cap o.G) = o'.G\}$$

Containment Range Query (CRQ): dado um objeto o' com extensão espacial $o'.G \subseteq E^d$, encontre todos os objetos o contidos em o' .

$$CRQ(o') = \{o | (o'.G \cap o.G) = o.G\}$$

A Figura 2 ilustra os três tipos de consulta.

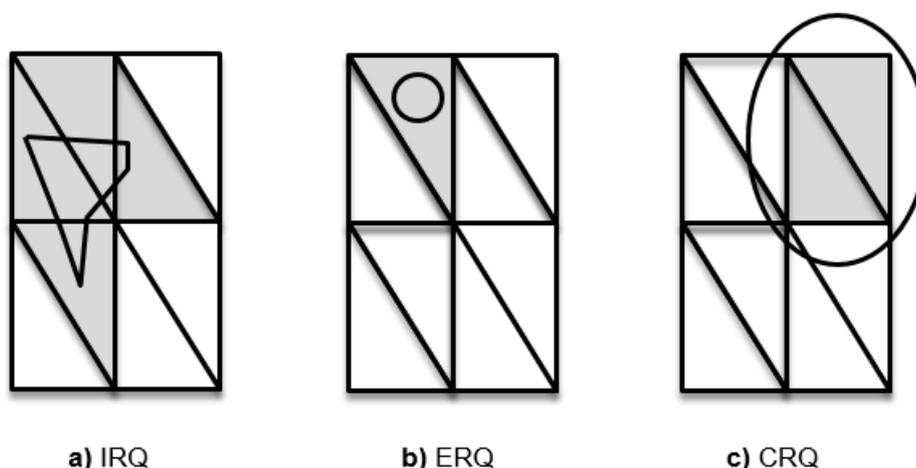


Figura 2: Exemplo de IRQ, ERQ e CRQ (adaptado de Gaede e Günther, 1998)

Além da teoria mostrada até agora, alguns pesquisadores perceberam que algumas aplicações estavam focadas em capturar o movimento de um certo objeto no espaço e não a sua forma propriamente dita. Como apresentado por Spaccapietra et al (2007), é o caso de carros, pessoas ou aviões equipados com dispositivos GPS, animais com dispositivos cujo sinal venha a ser capturado por satélites ou mesmo pacotes que possam ser identificados com *chips* de rádio frequência (RFID). Gütting et al (2000) distingue tais objetos como objetos discretamente móveis e objetos continuamente móveis, sendo que o primeiro pode ser facilmente implementado gravando-se a posição do objeto no espaço em intervalos de tempo definidos, enquanto o segundo precisa da definição de uma curva ou uma linha poligonal num espaço bi-dimensional (i.e., a sua trajetória) e algumas outras informações sobre o comportamento deste ao longo da trajetória para inferir-se a posição aproximada do objeto. Regiões móveis (nuvens de poluição ou a área de uma enchente, por exemplo) poderiam também ser mapeadas usando a abordagem de Gütting et al (2000) através da descrição da geometria do objeto junto com o seu comportamento ao longo da trajetória. Tais abordagens são chamadas de **bancos de dados de trajetórias** e estão fora do escopo deste

trabalho, uma vez que o modelo proposto se preocupa em capturar alterações na geometria dos dados e não a mobilidade destes.

2.2.1 Análise do exemplo proposto utilizando-se BDEs

A partir do exemplo proposto no item 2.1.1, vê-se que o conceito de GIS/BDE pode ser utilizado no controle dos centros de distribuição, que poderiam ser implementados como pontos em um mapa e suas respectivas áreas de cobertura como polígonos. Os clientes e pedidos podem ser também incluídos para que pudesse ser elegido o centro de distribuição mais próximo de acordo com a região de cobertura. Se o mapa contiver informações de estados e cidades do Brasil, consultas como “quais estados estão cobertos, mesmo que parcialmente, pelos centros de distribuição” (do tipo IRQ), “em quais estados há centros de distribuição” (do tipo CRQ) e “quais estados estão completamente cobertos pelos centros de distribuição” (do tipo ERQ) podem ser feitas sem grandes problemas. Além disso, se houver a informação das rodovias que cortam as regiões, as menores rotas de distribuição podem ser calculadas com mais facilidade.

2.3 Banco de Dados Temporais

Bancos de dados convencionais, por seu próprio conceito, guardam somente o último valor associado aos registros de suas tabelas significando que, uma vez atualizadas ou excluídas, o valor anterior das tuplas não pode ser mais recuperado. Tais bancos de dados são chamados transitórios. Mas a necessidade de guardar o histórico das alterações nos dados, bem como do esquema das tabelas, foi se mostrando crescente com o passar dos anos, surgindo as linhas de pesquisa em bancos de dados temporais (ou TDB – do inglês *Temporal Data Bases*).

Com relação às alterações de estado dos dados, Golfarelli e Rizzi (2009) apresentam as duas principais dimensões temporais que as tuplas de um TDB podem ter: **tempo de validade** e **tempo de transação**. O primeiro indica quando uma certa tupla no banco de dados é válida para o negócio (ou o tempo no mundo real), enquanto o segundo indica quando a tupla é registrada e válida no banco de

dados (ou o tempo no sistema). De acordo com a capacidade de um banco de dados em lidar com essas dimensões temporais, pode-se classificá-los em bancos de dados de tempos de validade, bancos de dados de tempos de transação ou bancos de dados bitemporais. Ainda de acordo com Golfarelli e Rizzi (2009), o principal benefício de um banco de dados bitemporal (que tem a capacidade de guardar ambos os tempos de validade e de transação dos dados) é a capacidade de se obter o mesmo resultado de uma consulta independente do momento em que esta é feita.

A Tabela 1 exemplifica uma sequência de registros contendo tempos de validade e tempos de transação. Nela podemos ver um cadastro de centros de distribuição com seus endereços e capacidades. A validade do registro para o negócio é dado pelas colunas V_Inicial e V_Final enquanto a validade para o sistema é dado por T_Inicial e T_Final. O centro de distribuição 1 (C1) está apenas cadastrado no sistema mas ainda não é considerado como válido para o negócio. Já o centro de distribuição 2 (C2) foi cadastrado inicialmente no endereço 2 mas percebeu-se que o cadastro estava errado procedendo-se o encerramento do registro para o sistema (preenchimento de T_Final) e criando-se um novo registro. O mesmo centro de distribuição começou a ser considerado válido para o negócio em fevereiro (preenchimento da V_Inicial) e ainda estava válido quando o registro foi encerrado no sistema. Note-se que a data de validade do negócio pode ser anterior à validade de sistema mostrando a independência entre os conceitos – o centro de distribuição 2 já era válido para o negócio antes de ser cadastrado no sistema e continuava válido até ser finalizado neste. O caso do centro de distribuição 3 (C3) é provavelmente o mais comum onde o registro é cadastrado no sistema (T_Inicial é preenchida), depois fica válido para o negócio (V_Inicial é informada) e, depois de um tempo de atividade, para de ser válido para o negócio e é invalidado no sistema alguns dias após (V_Final e T_Final, respectivamente, são preenchidas).

Tabela 1: Exemplo de registros com tempos de validade e transação de um TDB

CDist	Capacidade	Endereço	V_Inicial	V_Final	T_Inicial	T_Final
C1	300	End1	-	-	2013/01/15	-
C2	200	End2	-	-	2013/02/10	2013/02/14
C2	200	End3	2013/02/10	-	2013/02/14	2013/05/13
C3	400	End4	2013/03/01	2013/08/12	2013/02/15	2013/08/15

Apesar de poder ser implementado com tecnologia de bancos de dados convencionais, tal esquema torna extremamente difícil decidir quais registros são válidos para o sistema ou para o negócio, uma vez que todas as consultas precisam ser escritas tomando-se em consideração tais datas de validade. Além disso, se implementados sem a devida segurança, os campos de validade podem ser alterados à revelia do sistema e causar inconsistências nos registros. Para diminuir tais problemas, os pesquisadores têm procurado desenvolver extensões da linguagem SQL para facilitar o desenvolvimento dos sistemas que utilizam TDBs. Tais diretivas temporais são incorporadas normalmente através de reescrita de consultas. É o caso da SQL-92 e da sua evolução chamada TSQL2. Houve movimentação para incluir uma evolução das construções propostas na TSQL2 na especificação ISO da linguagem SQL3 (chamada SQL/Temporal) mas uma discordância entre os autores atrapalhou os esforços fazendo com que o comitê fosse desfeito em 2001. Entretanto, conceitos e construções da SQL/Temporal foram incluídas no padrão SQL:2011 e implementados em bases de dados como Oracle (chamado "*flashback queries*"), Teradata e IBM DB2 (SNODGRAS, s.d.).

Quanto à captura das alterações de esquema dos bancos de dados, Roddick (1995) introduz três tipos de alteração possíveis: i) Modificação de Esquema; ii) Evolução de Esquema e iii) Versionamento de Esquema. O primeiro acontece quando o banco de dados permite alterações no esquema, mas não oferece garantia quanto à integridade dos dados, enquanto o segundo permite a alteração de esquema e também garante a integridade dos dados. Além de permitir a alteração e garantir a integridade dos dados, o terceiro tipo permite o acesso aos dados tanto anteriores quanto posteriores à última versão do esquema. Este último tipo pode ser ainda dividido em versionamento de esquema parcial (quando somente os dados da última versão podem ser alterados – apesar de permitir acesso a todos os dados) ou versionamento de esquema total (quando a alteração é permitida em todas as versões). Implementar tais bancos de dados normalmente requer a introdução de metadados descrevendo o esquema através do tempo e funções de conversão entre versões. A reescrita de consultas é também uma técnica comum para adequar o que é consultado à versão apropriada do esquema. Um exemplo de tais implementações é o metamodelo COMET (EDER, KONCILIA e MORZY, 2006) que faz o uso de tabelas e relacionamentos para descrever a evolução do esquema de dados e como eles se relacionam para que o processamento das consultas possa ser adequado à

temporalidade envolvida. O modelo prevê condições para uma evolução segura, funções de transformação dos dados entre versões e facilidades para decidir quando a reescrita das consultas precisa extrapolar versões ou não, a fim de melhorar o desempenho da reescrita e execução das mesmas.

Seguindo o esquema proposto nesta pesquisa, apenas o conceito de validade de transação será exercitado. O problema de alteração de esquema também não será considerado nesta pesquisa, sendo o foco apenas a alteração de instâncias.

2.3.1 Análise do exemplo proposto utilizando-se BDTs

A partir do exemplo proposto no item 2.1.1, pode-se inferir que o projeto do produto pode utilizar conceitos de bases de dados bitemporais com as datas de transação e validade alterando a visibilidade do produto entre as várias áreas (engenharia, PCP, vendas) sem a necessidade de estruturas de controle alternativas. O mesmo se aplica aos centros de distribuição que podem ser alocados e estocados sem interferirem com o plano de vendas da empresa antes do tempo esperado.

2.4 Data Warehouse

Data Warehouses (DWs) surgiram a partir da convergência de tecnologias já existentes e se popularizaram por permitir uma rápida análise dos dados para a tomada de decisão estratégica, vindo ao encontro das necessidades analíticas das empresas.

O esquema lógico relacional que normalmente é implementado considera uma tabela de fatos e várias tabelas de dimensão. A tabela de fatos contém as métricas numéricas (previamente agregadas de acordo com a granularidade dos dados estipulada) e chaves estrangeiras para as tabelas de dimensão. As tabelas de dimensão contêm as informações complementares que descrevem as tuplas da tabela de fatos. O esquema mais comum está mostrado na Figura 3 (item a) e é chamado “esquema estrela” por conter apenas uma tabela de fatos e duas ou mais tabelas de dimensão em seu entorno. Caso possua duas ou mais tabelas de fatos, o

esquema passa a ser chamado de “constelação de fatos”. Uma outra característica é o desejo de se evitar junções de tabelas ao máximo levando à “desnormalização” das tabelas de dimensão. Caso haja necessidade, pode-se normalizar em certo grau as tabelas de dimensão gerando-se o esquema “flocos de neve” (Figura 3 item b). Vale ressaltar que tais esquemas são utilizados para a implementação de DWs sobre bancos de dados relacionais (sistemas ROLAP – do inglês *Relational OLAP*).

Cada dimensão possibilita operações como *slice and dice* (quando apenas parte das informações são retornadas), *pivot* (alteração da ordem das dimensões) e *drill-across* (relatórios envolvendo mais de uma tabela de fatos possuindo uma tabela de dimensão em comum). Elas ainda podem ser organizadas em hierarquias e são estas que garantem as operações de *drill-down* e *roll-up* (maior e menor detalhe dos dados, respectivamente). O trabalho de Ciferri (2002) mostra as possíveis operações nos dados, resultados esperados assim como outras considerações de implementação como o nível de granularidade dos dados na tabela de fatos, políticas de atualização dos dados e o processo de ETL (do inglês *Extraction, Transform and Load* – Extração, Transformação e Carga).

Devido à grande quantidade de dados envolvida e à necessidade de retornar dados de consultas complexas no menor tempo possível, DWs precisam de mecanismos especiais de armazenamento e indexação. A pré-agregação dos dados de acordo com as tabelas de dimensão (chamadas de “Visões Materializadas”) é uma técnica bastante utilizada mas precisa ser planejada com cuidado pois tem que levar em consideração o espaço de armazenamento, o desempenho da consulta e o tempo de construção das visões durante o processo de atualização dos dados. Vários trabalhos exploraram essas alternativas e, como exemplo, Harinarayan et al. (1996) se baseiam em uma “treliça” de visões (do inglês *lattice*) para indicar a dependência entre as várias visões materializadas possíveis em um DW, bem como o espaço de armazenamento gasto por cada uma das visões. A Figura 4 foi adaptada deste trabalho e mostra a relação de dependência entre as dimensões *Part*, *Supplier* e *Customer* (abreviadas por *p*, *s* e *c*, respectivamente) advindas do *benchmark* TPC-D usado como exemplo do artigo.

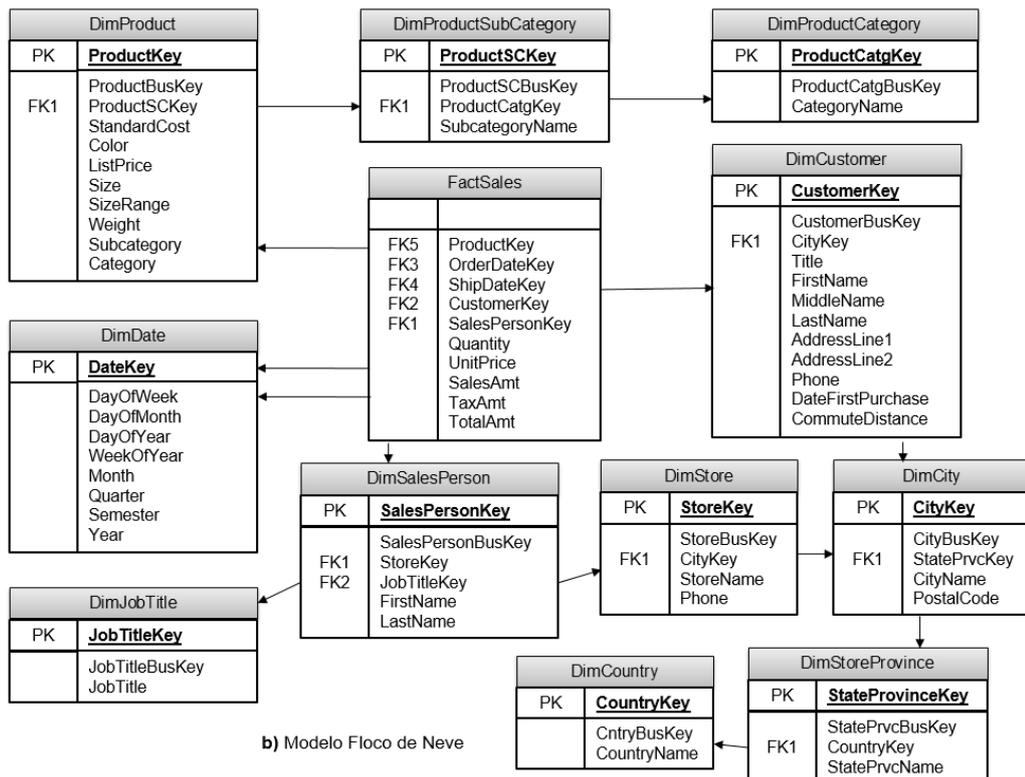
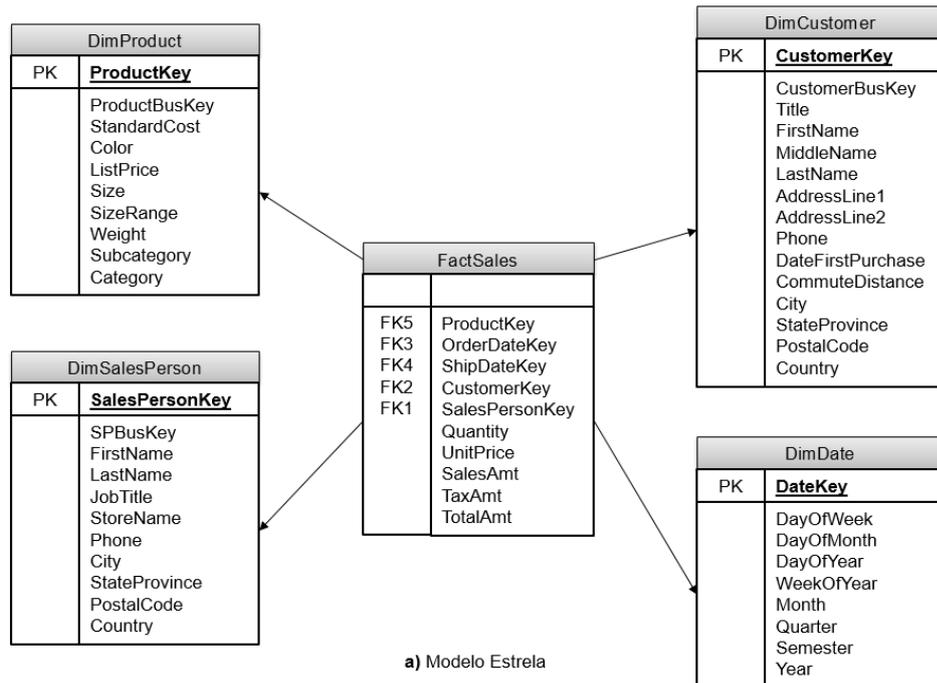


Figura 3: Esquemas Estrela e Floco de Neve

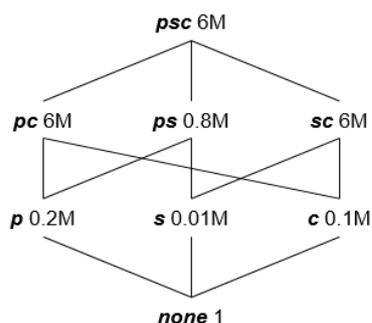


Figura 4: Treliça de dependência entre as visões materializadas das dimensões Part, Supplier e Customer – adaptado de Harinarayan (1996)

As consultas a este tipo de banco de dados (chamadas de consultas OLAP – *On-Line Analytical Processing*) também possuem características próprias como a presença constante de funções de sumarização dos dados, junção entre a tabela de fatos com as tabelas de dimensão relacionadas e agrupamento nas colunas de interesse. Por isso a presença visões materializadas (ou outros mecanismos de pré-processamento e organização dos dados) se tornou importante por significar um “atalho” às constantes junções e sumarizações dos mesmos.

Buscando facilitar a sintaxe em DWs, as consultas MDX (do inglês *Multi-Dimensional eXtensions*) surgiram como uma alternativa de consultas a cubos OLAP em bancos de dados SQL Server da Microsoft e depois foram adotadas em vários produtos comerciais como Oracle e Cognos. Sua sintaxe possibilita organizar os dados em múltiplas dimensões e não somente em linhas e colunas (como é feito no SQL tradicional) e já tem implícita todas as junções de uma tabela de fatos com as suas tabelas de dimensão e a organização dos membros de acordo com as hierarquias das suas dimensões. A Figura 5 mostra a sintaxe de uma consulta MDX simples e a identificação das partes da mesma (MICROSOFT, s.d.).

```

SELECT
  { [Measures].[Unit Sales], [Measures].[Store Sales] } ON AXIS(0),
  { [Time].[1997], [Time].[1998] } ON AXIS(1)
FROM Sales
WHERE ( [Store].[USA].[CA] )

```

Métricas a serem mostradas
 Membros selecionados
 Fato/Cubo de Dados
 Limitação dos Dados (*slice and dice*)

Figura 5: Exemplo de consulta MDX

Esta pesquisa utilizará o conceito de visões materializadas como base para se comparar com as configurações que processam consultas STOLAP usando índices.

2.4.1 Análise do exemplo proposto utilizando-se DWs

Desenvolvendo as ideias do exemplo proposto, um DW com as informações de vendas poderia ser formado a fim de identificar quão fortes são produtos e centros de distribuição e promover ações corretivas como promoções de vendas, readequação de preços ou a própria exclusão do produto em certos mercados.

2.5 Data Warehouse Espacial

A quantidade crescente de informações geográficas sendo geradas por sistemas de telemetria por satélite ou de monitoramento remoto (para citar alguns), fez com que surgisse a necessidade de se ter métodos eficientes de análise destes dados e uma arquitetura de DW poderia prover tais métodos a fim de melhor organizar, registrar e obter informações valiosas a respeito dos objetos por ele mapeados.

Isto posto, um sistema de Data Warehouse Espacial (ou DWE) consiste na adição de colunas de caráter geográfico (ou espacial) a uma ou mais tabelas de um DW (STEFANOVIC, HAN e KOPERSKI, 2000).

O primeiro problema que surge é com relação à integração dos dados a partir de fontes e sistemas heterogêneos. As diferenças podem ser encontradas tanto no formato (representação *raster* contra a baseada em geometrias, por exemplo) quanto na implementação feita por cada um dos fornecedores de *software*. O

segundo problema (e mais relevante para este trabalho) envolve o aumento na complexidade das consultas. Como já discutido no capítulo 2.2, os cálculos envolvidos nas consultas espaciais demandam maior esforço naturalmente acrescentando-se aí a explosão no número de registros a serem processados e agregados devido à natureza histórica de um DW.

Em um DWE as consultas OLAP passam a se chamar SOLAP (*Spatial OLAP*) devido ao novo tipo de dados introduzido e elas seguem as mesmas características das consultas OLAP (discutidas no capítulo 2.4) com a inclusão de uma cláusula espacial impondo limitações do tipo IRQ, ERQ ou CRQ (vistas no capítulo 2.2).

2.5.1 Análise do exemplo proposto utilizando-se DWEs

Tomando por base o exemplo proposto no item 2.1.1, a inclusão de colunas espaciais informando a localização de cada um dos centros de distribuição poderia formar um DWE com capacidade de responder não só a perguntas sobre as vendas, mas também perguntas mais complexas envolvendo a distância entre os centros de distribuição ou a localização destas em regiões específicas de cobertura.

2.6 Data Warehouse Temporal

DWs convencionais possuem normalmente um caráter histórico através da tabela de dimensão de tempo, mas este é limitado aos registros da tabela de fatos. Historicamente os registros de um DW eram considerados imutáveis por definição, mas o avanço na utilização e pesquisas mostraram que um certo nível de alteração é necessário também num DW. Kimball e Ross (2002) propõem algumas alternativas para tratar as alterações nas tabelas de dimensão, mas estas ainda não significam uma adaptação do DW incondicional às alterações dos dados.

A solução encontrada foi, nos mesmos moldes implementados nas bases de dados temporais (apresentados no item 2.3), incluir colunas de validade temporal às tabelas de dimensão criando-se, assim os Data Warehouses Temporais (DWTs). Assim sendo, os tempos de validade e de transação dos registros nas tabelas de dimensão possibilitaram a captura de alterações também nos dados das tabelas de

dimensão. Quanto às alterações de esquemas (também vistos no item 2.3), as mesmas técnicas para captura de alterações de esquema podem ser utilizadas.

As consultas OLAP dos DWs passaram a incluir as cláusulas temporais de validade dos dados, vindo a serem adaptadas de outras linguagens temporais (como TSQL2 ou SQL/TP) e chamadas de TOLAP (*Temporal OLAP*) por Mendelzon e Vaisman (2000). Além de apresentar formalmente a linguagem, os autores também apresentam em seu trabalho alguns exemplos de consultas que podem ser executadas pelas consultas TOLAP assim como discutem possíveis implementações que envolvem a tradução das cláusulas em SQL comum ou para a linguagem TSQL2.

Mendelzon e Vaisman (2000) consideram a otimização do desempenho na execução das consultas através de mecanismos de otimização de junções e podas de consultas. Já em Vaisman, Izquierdo e Ktenas (2006), um terceiro mecanismo utilizando visões materializadas é utilizado sendo que versões passadas são guardadas sem alterações para poderem ser utilizadas pelo mecanismo de reescrita de consultas.

2.6.1 Análise do exemplo proposto utilizando-se DWTs

Dando continuidade às possibilidades de implementação do exemplo proposto no item 2.1.1, um DWT possibilitaria consultas analíticas com caráter histórico. Perguntas como “qual o montante de vendas por tipo de produto usando a categorização de tipo na data da venda?”, ou “como os clientes eram categorizados há três anos?” (MENDELZON e VAISMAN, 2000) poderiam ser respondidas sem problemas.

2.7 Data Warehouse Espaço-Temporal

O avanço nas tecnologias de DWTs e DWEs se deu em paralelo por alguns anos até que os pesquisadores perceberam que certos tipos de sistemas geográficos precisavam também capturar as alterações feitas nas colunas espaciais. Além disso, boa parte dos aplicativos precisava dos dados consolidados seja para

facilitar a análise ou atender restrições comerciais (por exemplo a imposição de não poder revelar dados de localização de dispositivos móveis) (PAPADIAS, TAO, *et al.*, 2002). Como os DWTs já possuíam o que era necessário para controlar alterações tanto de instanciação de dados quanto de esquemas, o passo natural foi juntar os dois tipos de bancos de dados multidimensionais, surgindo os Data Warehouses Espaço-Temporais (DWETs).

Quanto à modelagem, os DWETs mais simples contêm tanto colunas espaciais quanto colunas contendo datas de validade inicial e final dos dados significando que qualquer consulta feita deverá escolher uma janela de execução temporal a fim de considerar apenas os registros válidos no intervalo de datas alvo. As variações das colunas espaciais podem ser tanto de posição quanto de formato.

As estruturas de indexação até agora exploradas consistem no uso de métodos de indexação e agregação usados nos DWEs e DWTs, que vão desde as visões materializadas (já apresentadas no item 2.4) até o uso de Índices Bitmap de Junção (IBJ) em junção com REMs (TSURUDA, 2013) para auxiliar no mapeamento das informações.

As consultas a um DWET são chamadas STOLAP (*Spatio-Temporal OLAP*) e possuem como característica três tipos de predicados: i) convencional; ii) espacial e iii) temporal. Tais tipos de predicados serão explorados no Capítulo 3 - desta dissertação que irá descrever como se dá o processamento de consultas analíticas em DWETs. Mais uma vez, a grande quantidade de registros (característico de um DW e suas especializações) em conjunção com as complexidades de processamento dos dados espaciais e temporais imporão grandes restrições de desempenho.

2.7.1 Análise do exemplo proposto utilizando-se DWETs

Continuando as possibilidades de implementação considerando o exemplo proposto no item 2.1.1, um DWET possibilitaria avaliar incrementos ou decrementos na área de cobertura de um centro de distribuição para investigação de motivos e melhoria no processo de negócios.

2.8 Considerações Finais

Neste capítulo foi feito o embasamento teórico dos conceitos necessários para o entendimento do trabalho proposto. Tal teoria envolve o surgimento dos bancos de dados não convencionais, a convergência de tecnologias provendo o surgimento dos data warehouses e suas especializações na área geográfica e temporal gerando os DWETs que são o foco desta pesquisa.

Também foi visto como as novas tecnologias e pesquisas podem ajudar na melhoria do ambiente computacional de uma empresa através do exemplo proposto.

No próximo capítulo serão detalhados alguns dos mecanismos que possibilitam a execução de consultas analíticas em DWs, DWEs e DWETs.

CAPÍTULO 3

PROCESSAMENTO DE CONSULTAS ANÁLITICAS

Este capítulo introduz como o processamento de consultas analíticas em Data Warehouses é possível de ser realizado por meio da utilização de técnicas mais simples como Junção Estrela e Visões Materializadas ou de técnicas mais elaboradas como Índices Bitmap de Junção e Índices específicos para os tipos de dados não convencionais envolvidos neste trabalho.

3.1 Junção Estrela e Visões Materializadas

Como visto na Seção 2.4, a maioria dos DWs atuais são modelados de acordo com o esquema estrela, no qual tem-se uma tabela de fatos e várias tabelas de dimensão. A junção estrela consiste na junção dos registros da tabela de fatos com uma ou mais de suas respectivas descrições nas tabelas de dimensão.

Apesar de fácil de descrever, tal junção pode causar imensos problemas de desempenho devido à grande quantidade de registros de um DW. Uma cuidadosa indexação precisa ser feita e, ainda assim, esta pode não se mostrar suficiente. A fim de ajudar no processamento de consultas de um DW, as visões materializadas (também descritas na Seção 2.4) tornaram-se uma boa alternativa por sumarizar os dados em diferentes níveis e, assim evitar a junção estrela para várias das consultas OLAP feitas pelos usuários.

Mesmo com claros avanços no processamento de consultas OLAP, vários pesquisadores pensaram em como ajudar a melhorar o desempenho da junção estrela e das visões materializadas com novos esquemas de indexação e um dos mais promissores é o Índice Bitmap de Junção (IBJ) a ser visto a seguir.

3.2 Índices Bitmap de Junção

Índices bitmap têm sido usados há um bom tempo em bancos de dados convencionais como uma alternativa mais rápida aos tradicionais índices baseados em árvore por prover processamento em nível bit-a-bit, que é mais próximo das arquiteturas de *hardware*. Sua desvantagem em ambientes transacionais é que sua manutenção é cara frente ao grande número de inserções, atualizações e exclusões de registros. Entretanto, ambientes OLAP já têm como característica o baixo nível de operações de atualização (normalmente restritas ao processo de ETL) tornando viável a escolha deste tipo de índice para apoiar o processamento das consultas.

Em um DW, a junção dos registros da tabela de fatos com suas descrições nas tabelas de dimensão é uma das operações mais comuns encontradas e, pensando nesse fato, alguns pesquisadores criaram uma especialização dos índices bitmap já levando em conta tal junção. Tal tipo de índice foi chamado de Índice Bitmap de Junção (IBJ).

A Figura 6 representa um esquema de construção dos IBJs para a coluna *s_address* da tabela dimensão *Supplier* contra a tabela fato *Lineorder*. Nela podemos notar que são formados n vetores com m posições cada, onde n é o número de diferentes valores do domínio da coluna sendo mapeada na tabela de dimensão e m é o número de linhas da tabela de fatos. A construção segue num laço onde cada registro da tabela de fatos (setas horizontais saindo da tabela de fatos para os IBJs) é avaliado contra o registro da tabela de dimensão sendo considerado (setas saindo da tabela dimensão para os IBJs). Caso a linha da tabela de fatos corresponda a uma das chaves mapeadas pela coluna sendo avaliada na tabela de dimensão, o valor “1” é escolhido para a sua posição correspondente no IBJ. Caso contrário, o valor “0” é atribuído. Para o vetor “A”, pode-se notar que somente as duas primeiras posições do vetor possuem o valor “1”, pois somente estas duas linhas da tabela de fatos correspondem às chaves da tabela de dimensão mapeadas pelo valor “A” na coluna *s_address*. As outras posições do vetor não dizem respeito a esta linha da tabela de dimensão e, portanto, recebem o valor “0”.

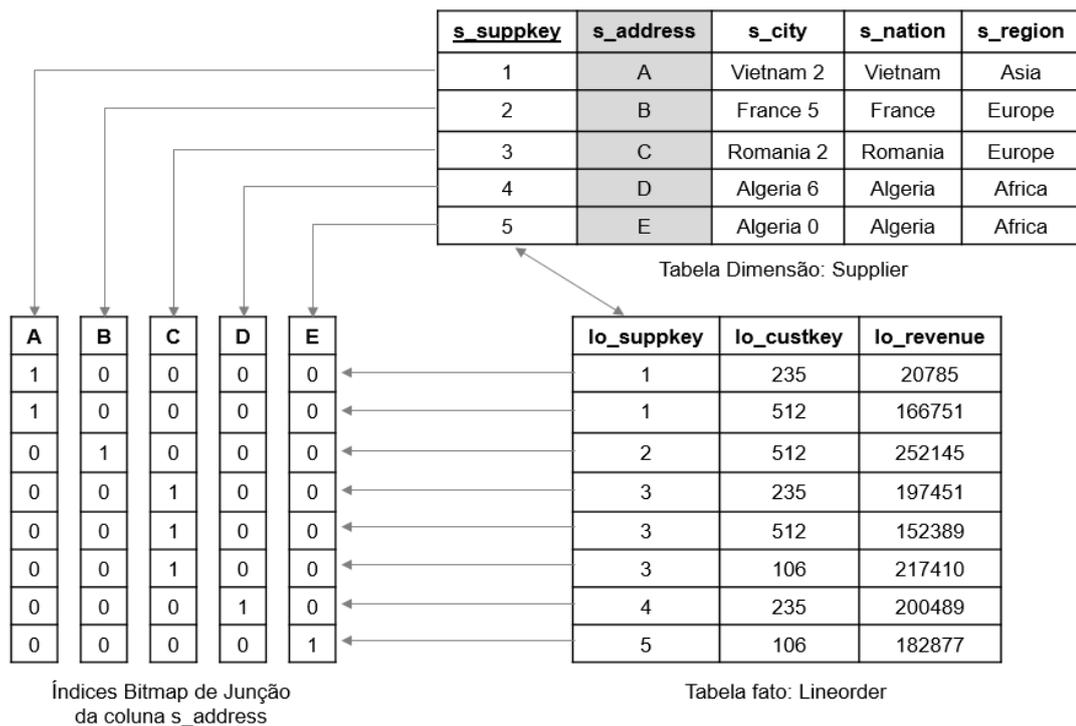


Figura 6: Exemplo de construção de IBJs – adaptado de Siqueira (2009) e O’Neil e Graefe (1995)

O processamento de uma consulta envolvendo tais índices envolve a execução de operações lógicas bit-a-bit nos índices pertinentes para posterior cálculo da resposta utilizando os valores da tabela de fatos correspondendo aos registros cujo resultado final foi ‘1’ nos cálculos bit-a-bit. No exemplo da Figura 6, a consulta “qual o total de vendas para os endereços C e E” seria respondida fazendo-se uma operação *OR* bit-a-bit entre os vetores C e E. O resultado desta operação indicaria que a soma dos valores de *lo_revenue* nas linhas 4,5,6 e 8 da tabela de fatos corresponde à resposta desejada.

Apesar da facilidade de representação com zeros e uns, até índices bitmap apresentaram dificuldades de escalabilidade ao enfrentar o grande número de registros encontrados em um DW. As três técnicas apresentadas a seguir foram idealizadas pelos pesquisadores para enfrentar tais problemas.

3.2.1 Binning

A técnica de *Binning* apresentada por Wu, Stockinger e Shoshani (2008) consiste no agrupamento de várias chaves das tabelas de dimensão em estruturas

de bitmap semelhantes chamadas “bins” (compartimentos, em tradução livre) e, assim, requerer menos sequências de bitmap para codificar uma certa coluna.

Como exemplo, observa-se na Figura 7 como uma coluna numérica com valores entre 0 e 100 pode ser dividida em intervalos de 20 e mapeada em cinco compartimentos distintos. Apesar da economia de espaço, a técnica envolve a adição de uma fase de refinamento da solução, uma vez que falsos candidatos podem ser incluídos. Para consultas de valores fixos (“quais os registros onde $X = 26,9$ ”, por exemplo), o compartimento B_1 é retornado, mas há a necessidade de refinamento da resposta, uma vez que o registro 1 também faria parte da solução erroneamente. O grande ganho vem com as consultas por intervalos onde os compartimentos totalmente contidos nos intervalos da consulta podem ser retornados sem processamento adicional. O exemplo da figura mostra que, para uma consulta onde $35 \leq X \leq 65$, o compartimento B_2 pode ser incluído sem refinamento enquanto os compartimentos B_1 e B_3 irão precisar deste para retornar a solução correta.

RID	X	B_0 [0;20)	B_1 [20;40)	B_2 [40;60)	B_3 [60;80)	B_4 [80;100)
1	32,5	0	1	0	0	0
2	95,0	0	0	0	0	1
3	26,9	0	1	0	0	0
4	18,5	1	0	0	0	0
5	62,8	0	0	0	1	0
6	68,2	0	0	0	1	0
7	59,3	0	0	1	0	0

Valores da coluna

Intervalo da consulta: $35 \leq X < 65$

Figura 7: Exemplo da técnica de *binning* – adaptada de Wu, Stockinger e Shoshani (2008)

Apesar da possível introdução de uma fase de refinamento, os ganhos obtidos com a exclusão de compartimentos são maiores fazendo compensar a utilização da técnica.

3.2.2 Compressão

A compressão em IBJs foi introduzida por Wu, Otoo e Arie (2006) através de um algoritmo especializado de compressão de dados bitmap chamado WAH (Word Aligned Hybrid code).

Devido à própria natureza de um DW de conter um grande número de registros a serem mapeados entre a tabela de fatos e suas tabelas de dimensão, os vetores de bits são muitas vezes “esparços” significando que contêm grandes seqüências de ‘0’s e ‘1’s consecutivos. O algoritmo WAH se vale dessa propriedade separando o vetor de bits em partes iguais (de tamanho 31 ou 63 – de acordo com a arquitetura do processador) e criando dois tipos de interpretação dos dados chamados ‘cauda’ e ‘preenchimento’. Caudas possuem ‘0’ como bit inicial indicando que as posições de bits restantes precisam ser respeitadas da maneira como estão. Preenchimentos começam com o bit ‘1’, têm o segundo bit indicando o qual valor deve ser repetido (‘0’ ou ‘1’) naquele bloco e o restante dos bits indicando a quantidade de blocos a serem repetidos com o mesmo valor.

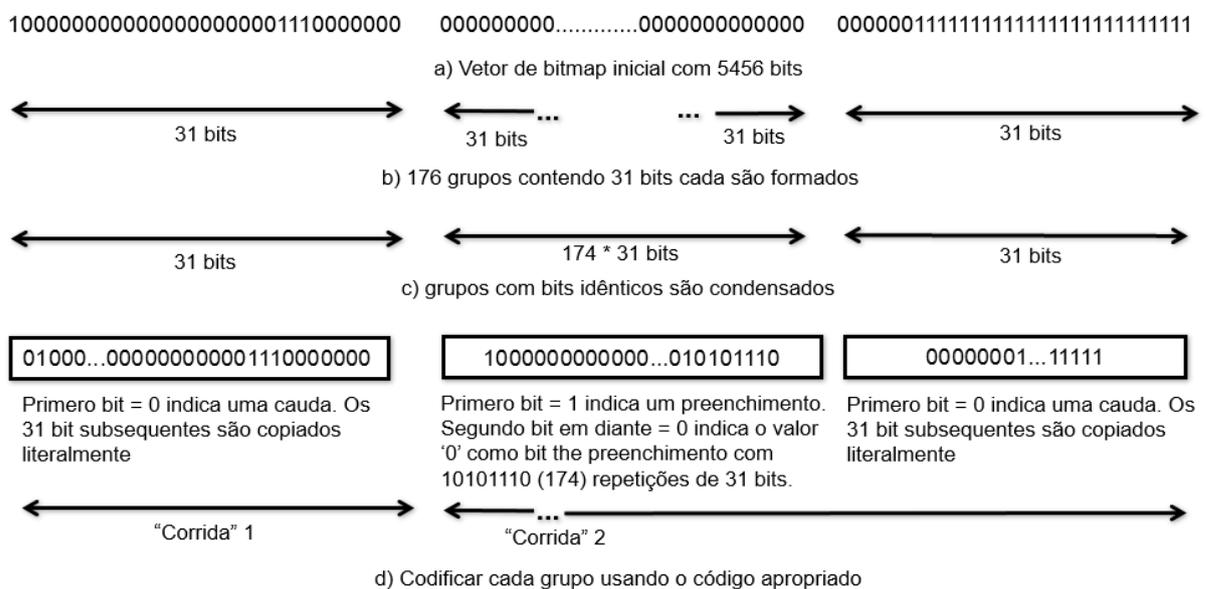


Figura 8: Exemplo de compressão – adaptado de Wu, Otoo e Arie (2006)

No exemplo dado pelo artigo (Figura 8), um vetor contendo 5456 bits (e separado por blocos de 31 posições) pôde ser representado utilizando-se 96 bits (uma cauda inicial seguida por um bloco de preenchimento indicando 174*31 bits com valor ‘0’ e finalizado por um último bloco de cauda).

A exemplo da técnica de *binning*, a compressão também adiciona processamento dos registros para leitura e escrita dos seus valores, mas os ganhos com economia de espaço fazem compensar a utilização da técnica.

3.2.3 Codificação

A técnica de codificação foi apresentada por Wu e Buchmann (1998) e tem seu ponto central na utilização de códigos binários representando os valores do domínio de dados encontrado na coluna a ser codificada com os vetores de bits sendo criados no número de bits usado em tal codificação (ao invés de um vetor de bits para cada valor do domínio).

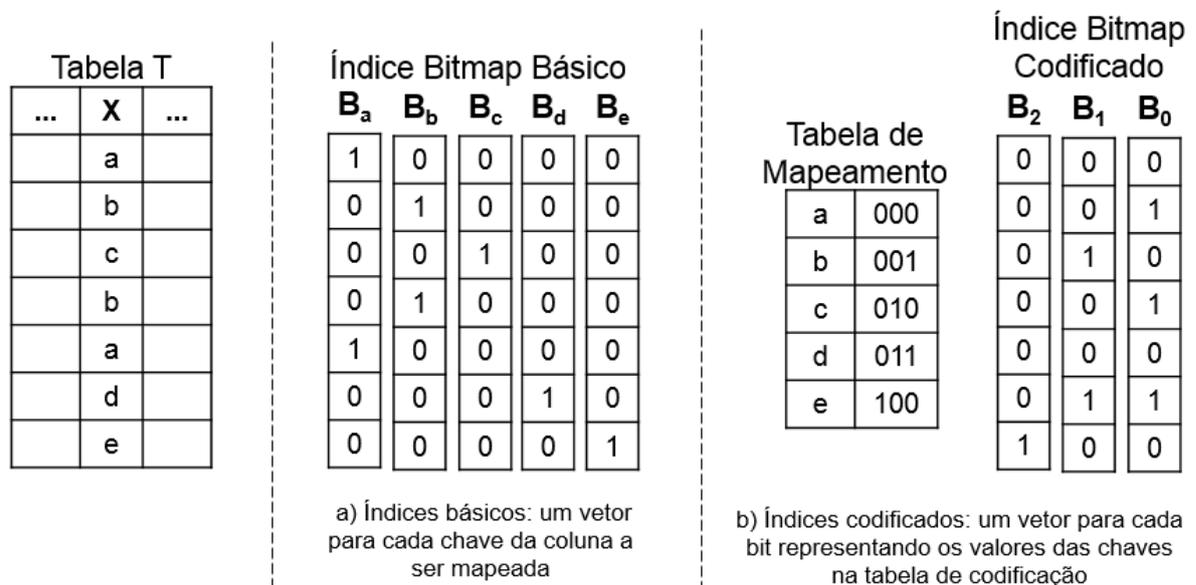


Figura 9: Exemplo de codificação de índices binários – adaptado de Wu e Buchmann (1998)

A Figura 9 mostra o exemplo do artigo onde uma coluna com 5 valores caractere ('a' a 'e') – que requereriam 5 vetores de bits para o mapeamento (item 'a' da figura) – pôde ser codificada usando-se 3 bits (000 para 'a', 001 para 'b', e assim por diante) e o mesmo mapeamento pôde ser feito utilizando-se os 3 vetores codificados (ao invés dos 5 iniciais), como mostrado no item 'b' da figura. Extrapolando a ideia, poder-se-ia ter um domínio de dados com 1 milhão de valores podendo ser codificados em apenas 20 vetores (ao invés dos 1 milhão de vetores iniciais). A contrapartida é que os valores do resultado precisam ser decodificados antes de serem retornados, mas, mais uma vez, tal custo é aceitável quando se leva

em consideração a economia no número de vetores e nos cálculos que podem ser feitos ao utilizá-los.

3.3 Índices Espaciais para Data Warehouse

As técnicas e índices apresentados até agora podem ser utilizadas em qualquer DW sem adaptação. Para DWEs, entretanto, novas técnicas e índices foram necessários devido à presença de um novo tipo de dados lidando com formas espaciais. Apresentaremos a seguir algumas técnicas e índices específicos para auxiliar na execução de consultas SOLAP presentes em DWEs.

3.3.1 aR-Tree

Proposto por Papadias et al. (2001), a aR-Tree é uma estrutura de árvore contendo REMs e a sumarização da tabela de fatos para os objetos espaciais contidos em cada um deles. A ideia principal é utilizar o valor pré-agregado na árvore para os REMs inteiramente contidos na janela de consulta e, assim, evitar cálculos de sumarização envolvendo todos os objetos espaciais a cada consulta.

A construção da aR-Tree começa pelos nós folha, que contém os REMs dos dados espaciais na sua granularidade mais baixa e a sumarização dos dados na tabela de fatos para cada um deles. A estrutura prevê ponteiros de ligação entre os nós de cada nível e a possibilidade de extensão dos dados. No exemplo proposto pelos autores, a sumarização da tabela de fatos poderia prover a quantidade de carros em várias estradas de uma certa região e a extensão poderia conter a capacidade máxima de uma estrada para posterior cálculo de média de ocupação da mesma. O passo seguinte consiste no agrupamento dos REMs dos nós folhas (e de seus dados) em REMs maiores até chegar ao topo da árvore contendo a sumarização de todos os objetos do DWE. A Figura 10a toma o exemplo dado pelos autores e mostra 5 estradas com seus respectivos REMs (a_1 , a_2 , a_3 , a_4 e a_5) e cujos pontos representam a quantidade de carros transitando em cada uma delas (0, 3, 1, 4 e 2, respectivamente). Os REMs são então incluídos nos nós folhas da árvore de acordo com a sua proximidade (nível 0) e, sucessivamente, reagrupados nos níveis

superiores gerando os REMs A1 e A2 que contém o total de carros transitando nas estradas por eles englobadas

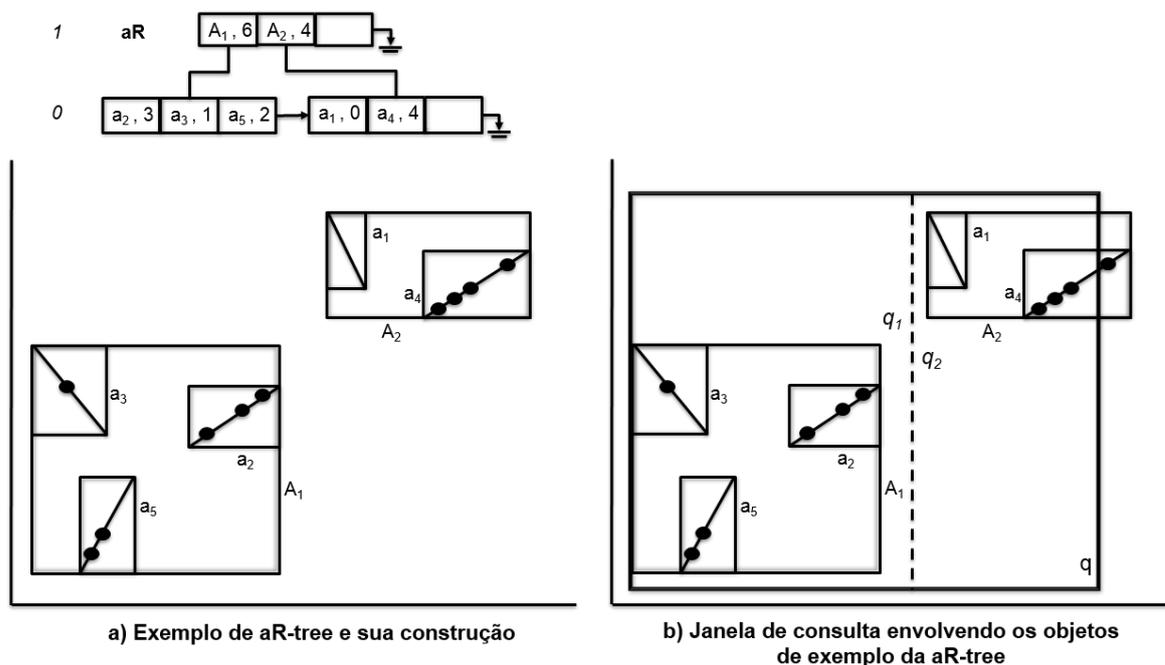


Figura 10: Exemplo de aR-Tree adaptado de Papadias et al. (2001)

A Figura 10b mostra uma possível janela de consulta q que engloba totalmente o REM A1 mas não totalmente A2. Duas estratégias de execução são necessárias para o retorno dos dados. Para A1, basta percorrer a árvore até encontrar o dado sumarizado correspondente a este. Para os outros objetos não totalmente englobados o acesso e cálculo normal da sumarização é necessária. O pior caso são janelas de consulta pequenas e que não englobam totalmente nenhum dos REMs formados ao redor dos objetos espaciais mapeados fazendo com que o método tenha o desempenho de uma consulta espacial normal. O melhor caso são janelas de consulta que englobam totalmente todos os objetos e na qual apenas o acesso ao topo da árvore é necessário.

O artigo também prova que a organização do índice com REMs que se englobam pode prover uma hierarquia entre os objetos e, assim, se aproveitar da teoria de treliça de visões proposta por Harinarayan (1996) e, assim, poder decidir por uma melhor estratégia de materialização dos dados.

3.3.2 SB-index

Com todas as funcionalidades de IBJs em mãos (incorporadas por seus autores no software intitulado Fastbit), Siqueira (2009) introduz um novo tipo de estrutura de índice para DWEs chamado *Spatial Bitmap Index* (SB-index). Como visto na Figura 11, o SB-index pode ser representado por um vetor cujo número de registros corresponde às chaves de uma tabela de dimensão contendo uma coluna de dados espaciais e cujo conteúdo é o REM de cada região mais um ponteiro para o início do IBJ correspondente.

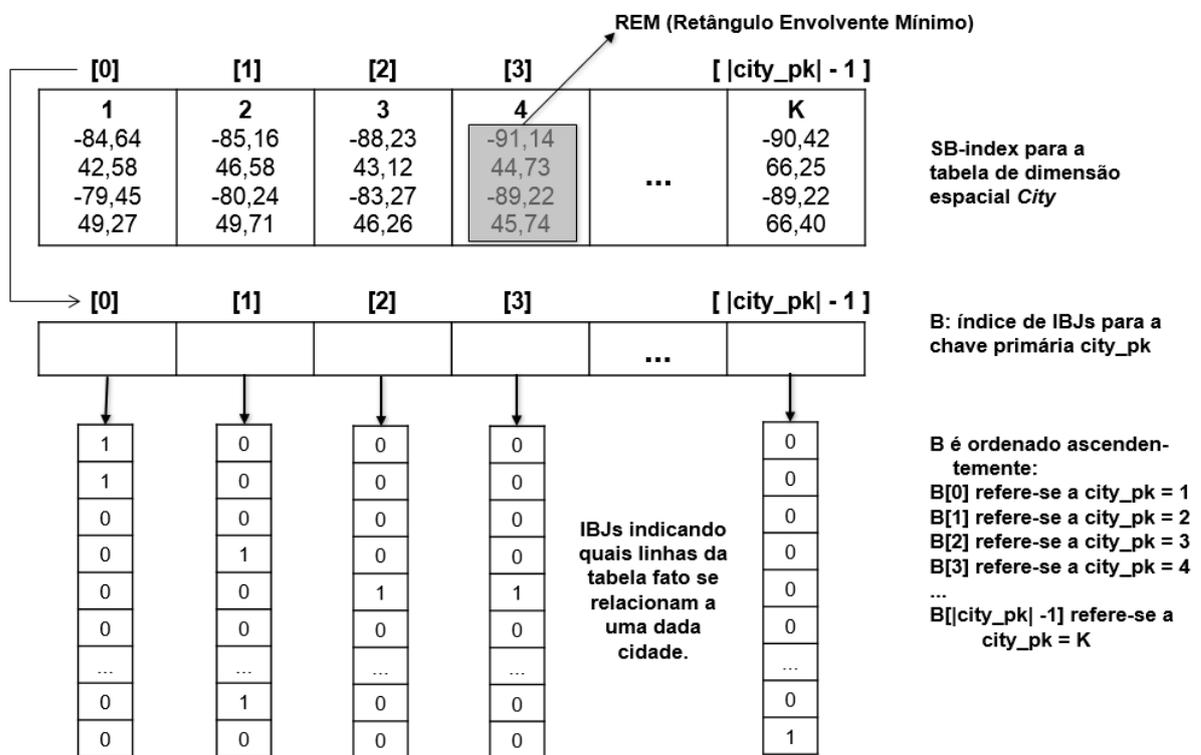


Figura 11: Estrutura do SB-index – adaptado de Siqueira (2009)

A ideia é ter um primeiro passo de busca sequencial dos REMs contidos no vetor do índice para retirar os valores que, com certeza, não fazem parte da solução da consulta (por não estarem contidas no REM da região). O segundo passo do processamento é o refinamento da solução acessando as geometrias de cada um dos candidatos e eliminando a incerteza introduzida pela utilização de REMs. De posse dos registros que devem constar na solução final, os IBJs correspondentes são acessados para se calcular o resultado final da consulta. A Figura 12 mostra esses dois estágios do processamento e mostra particularidades como a

organização das entradas respeitando os tamanhos dos blocos dos dados em disco para leitura e inclusão na memória principal (passo 1), a verificação dos REMs e formação de candidatos cujos REMs intersectam a janela de consulta (passos 2 e 3), o refinamento da solução através do acesso à geometria completa diretamente no banco de dados para exclusão de falsos candidatos (passo 4) e a finalização com a inclusão das chaves primárias que satisfazem à consulta para posterior processamento no software Fastbit (passo 5). No final, o predicado espacial original escrito em termos de coordenadas é reescrito em um predicado convencional escrito em termos de chaves primárias e operadores lógicos OR.

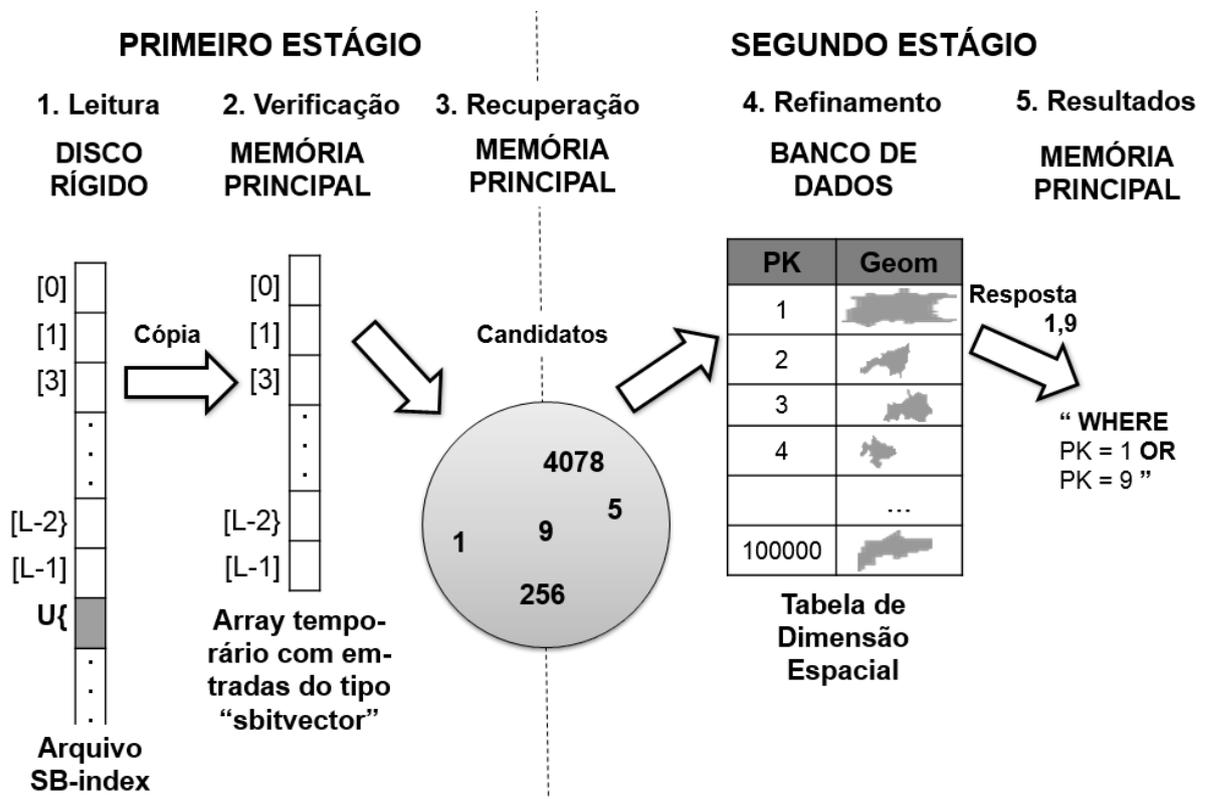


Figura 12: Etapas de processamento do SB-index - adaptado de Siqueira (2009)

Com este esquema, troca-se muitos cálculos de intersecção de polígonos (mais "custosos" para o DWE) por vários cálculos com quadrados (mais "baratos") e alguns cálculos com polígonos (durante a fase de refinamento da solução).

O resultado de tal esquema de processamento em comparação com o método tradicional de visões materializadas foi um ganho de 76% a 96% no tempo de resposta das consultas SOLAP. O trabalho de Siqueira (2009) também analisou

este índice sobre dados DWGs redundantes e, com uma adaptação para tal esquema de dados, foi possível um ganho mínimo de 80% (contra 25% iniciais).

3.3.3 HSB-index

Mesmo com os sólidos ganhos verificados pelo SB-index, avanços neste foram idealizados pelo mesmo autor e os resultados foram apresentados em Siqueira et al. (2012).

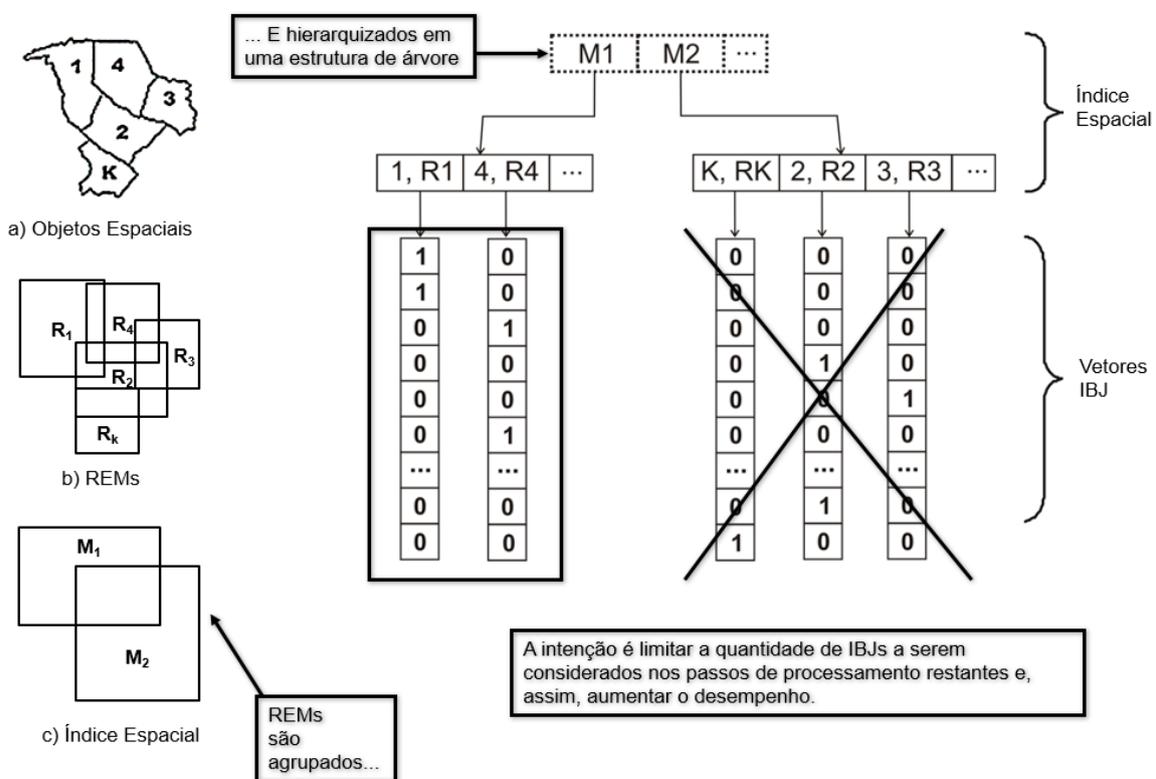


Figura 13: Estrutura do HSB-index (adaptado de Siqueira et al.; 2012)

O artigo mostra o HSB-index (*Hierarchical Spatial Bitmap Index*) que consiste na troca do vetor de REMs proposto no SB-index por uma estrutura hierárquica (como uma árvore R*) formada pelo agrupamento sucessivo de REMs (Figura 13). O processamento do índice se utiliza da estrutura de árvore para podar nós inteiros da estrutura e abreviar a fase de exclusão de registros através de seus REMs (ao invés de processar todos os REMs contidos no vetor do SB-index). Outra melhoria foi a inclusão de um *buffer pool* para guardar a estrutura dos registros por mais tempo em memória e, assim, agilizar consultas sucessivas.

Tanto o SB-index quanto o HSB-index se mostraram muito superiores à técnica de visões materializadas, mas quando comparados um ao outro, o HSB-index se mostrou mais vantajoso que o SB-index somente quando a seletividade do predicado espacial não passasse de 200 registros (de um total de 1 milhão de registros espaciais nos testes realizados). Caso contrário, a necessidade de navegação em muitos dos nós da árvore não trouxe grandes vantagens sobre o algoritmo de varredura e comparação de todos os registros provido pelo SB-index.

3.4 Índices Espaço-Temporais para Data Warehouse

Técnicas e índices para DWs e DWEs foram vistos nos itens anteriores deste capítulo. DWETs representam novos desafios (por combinar colunas com tipos de dados espaciais e temporais) requerendo também a utilização de técnicas e índices específicos às suas características. Alguns desses (a serem vistos a seguir) auxiliarão na execução de consultas STOLAP pertinentes aos DWETs.

3.4.1 a3DR-tree

Papadias et. al. (2002) vêm nos mostrar duas extensões do índice aR-tree que conferem a este a possibilidade de lidar com dados históricos. A primeira estratégia utilizada foi chamada a3DR-tree e consiste em modificar os nós da aR-tree para incluir o intervalo temporal e, a cada alteração dos dados agregados, incluir um novo registro ao nó correspondente da árvore. A Figura 14a mostra um exemplo de cubo de dados a ser mapeado para as estruturas aqui apresentadas e a Figura 14b a distribuição espacial das regiões do cubo de dados assim como uma possível janela de consulta aos dados.

A Figura 15 mostra a a3DR-tree resultante do mapeamento do exemplo da Figura 14a onde pode-se notar como os nós da árvore vão acumulando registros à medida que os dados agregados vão se alterando nos intervalos de tempo.

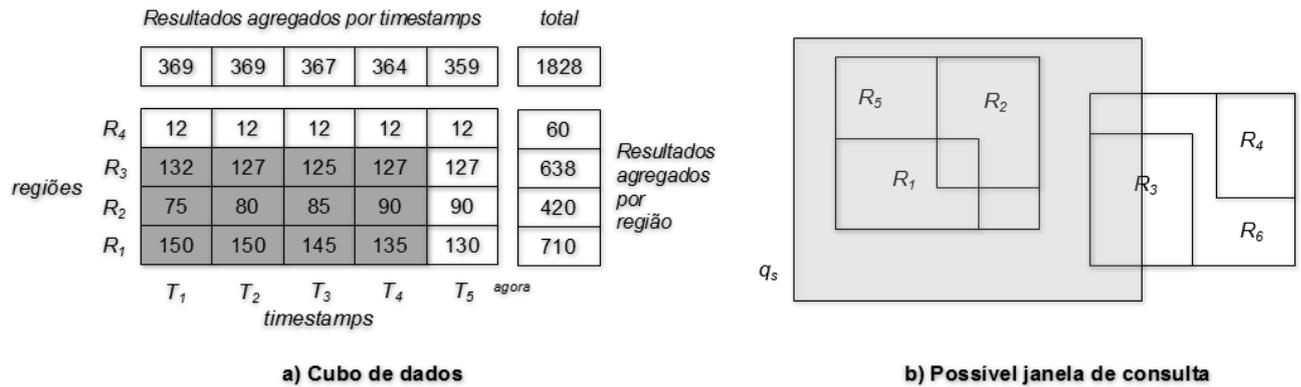


Figura 14: Cubo de dados e uma possível janela de consulta sobre estes para os exemplos da a3DR-tree e aRB-tree – adaptada de Papadias et al. (2002)

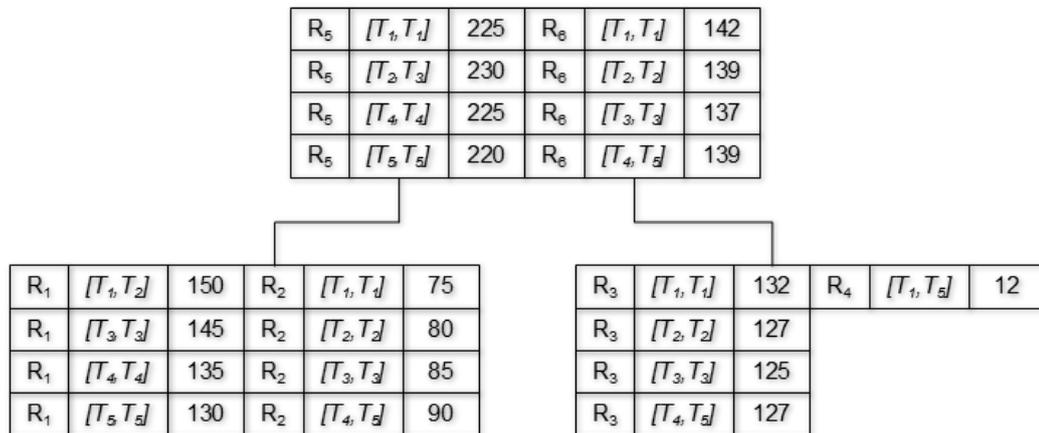


Figura 15: a3DR-tree resultante do exemplo proposto

Os autores ponderam que a a3DR-tree possui vantagens ao processar uma consulta por já conter os dados agregados nas regiões podendo evitar o gasto ao se processar registros próximos em termos espaciais, mas possivelmente separados fisicamente no espaço em disco (chamado *column scanning*). Como desvantagens, são citados o desperdício de espaço ao se armazenar o mesmo MBR para uma dada região a cada registro inserido e o tamanho exagerado da estrutura dos nós contendo poucas ligações entre eles (chamado *fanout*), o que compromete a execução das consultas.

3.4.2 aRB-tree

Avançando nas suas ideias, Papadias et al. (2002) mostram uma segunda extensão do índice aR-tree que tenta solucionar os problemas encontrados pela a3DR-tree. A estratégia utilizada foi incluir estruturas de árvore binária para informar as alterações dos valores agregados através do tempo. Tal estrutura possui nós do tipo $\langle b.time, b.pointer, b.aggr[] \rangle$ onde $b.aggr$ é o valor agregado no tempo $b.time$ e $b.pointer$ faz a ligação para o próximo nível da árvore. Uma regra de economia é não replicar o valor de $b.aggr$ caso este não mude entre tempos consecutivos. A Figura 14 e a Figura 16 reproduzem o exemplo utilizado no artigo, onde um cubo de dados com contagens por região e tempo é mapeado na estrutura da aRB-tree.

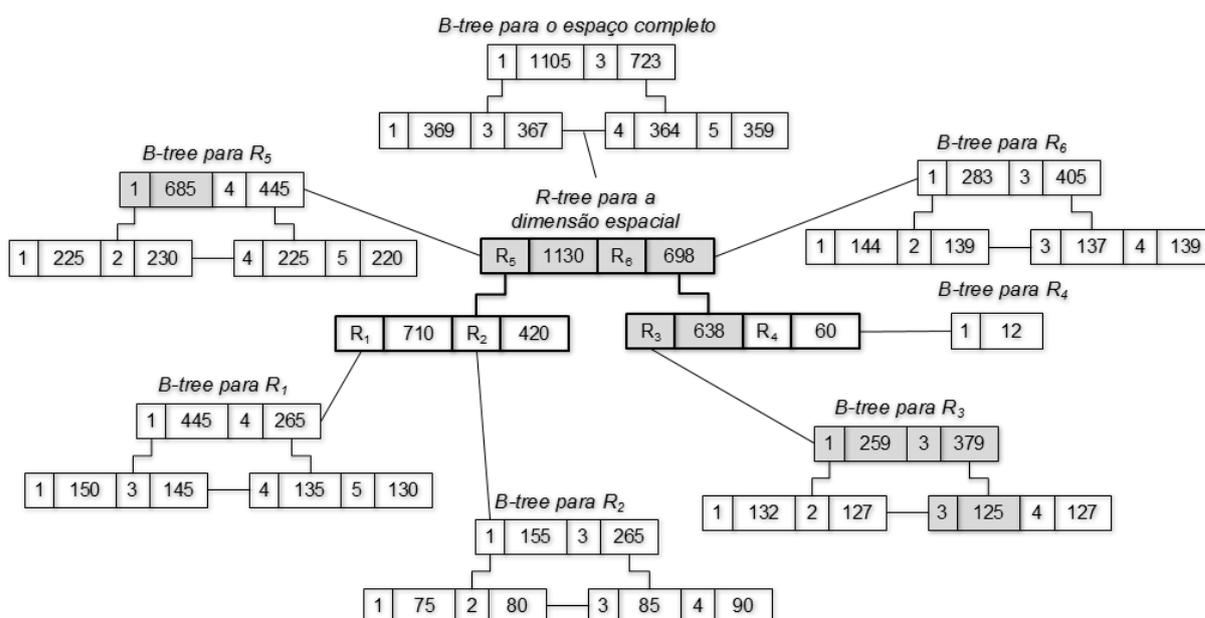


Figura 16: aRB-tree resultante da agregação do cubo de dados de exemplo – adaptada de Papadias et al. (2002)

O processo de construção da aRB-tree envolve a construção da R-tree inicial (em negrito na Figura 16) contendo a agregação dos valores para cada região da dimensão espacial e, assim como na aR-tree, os REMs de cada região são utilizados e depois englobados em REMs maiores para formação dos níveis superiores da árvore (R_1 e R_2 são englobados em R_5 enquanto R_3 e R_4 são englobados por R_6).

A seguir, cada região é detalhada temporalmente através da estrutura de B-tree complementar descrita anteriormente contendo o *timestamp* inicial e final com suas respectivas contagens. Observando-se a B-tree para mapear a região R_1 , pode-se notar que uma simplificação foi feita ao omitir-se o tempo 2, uma vez que este possui o mesmo valor do tempo 1 naquela região. Tal simplificação é máxima na região R_4 , uma vez que há somente um valor para todos os tempos e, portanto, somente um nó da árvore se faz necessário. Além dos detalhes temporais para cada região, uma última árvore é adicionada com o detalhamento por tempo da região completa.

A R-tree inicial (em negrito na Figura 16) pode responder diretamente a consultas contendo somente restrições espaciais, enquanto a “B-tree para o espaço completo” pode responder às consultas contendo somente restrições temporais. Quando as duas restrições são utilizadas na consulta, uma navegação seletiva pelos nós dos dois tipos árvores se faz necessária (nós em destaque na Figura 16). Uma consulta limitada à região q_s da Figura 14b no intervalo de tempo $[1,3]$, por exemplo, iniciaria com a navegação ao nó $\{R_5, 1130\}$ da R-tree e, pela região R_5 estar completamente contida em q_s , somente a B-tree temporal descrevendo esta região precisa ser acessada (as B-trees para as regiões R_1 e R_2 já estão sumarizadas em R_5 e, portanto, podem ser descartadas da avaliação). Ao acessar a B-tree temporal de R_5 , vê-se que o nó raiz possui os totais para os intervalos de tempo $[1,3]$ e $[4,5]$ e, portanto, somente o acesso ao nó $\{1, 685\}$ já fornece o total que se precisa para esta região. Continuando a avaliação, tem-se que a janela de consulta q_s também intersecta a região R_6 e esta precisa ser avaliada. Porém, ao contrário da R_5 , R_6 é englobada somente parcialmente por q_s e, portanto, o acesso aos nós folha de R_6 (correspondendo às regiões R_3 e R_4) é necessário. Já um nível abaixo na R-tree, vê-se que R_4 não é intersectada por q_s e pode ser descartada. Não é o caso da região R_3 , cuja B-tree é acessada e onde vê-se os intervalos $[1,2],[3,5]$ descritos no primeiro nível. O nó $\{1, 259\}$ pode ser utilizado por estar totalmente contido na restrição temporal mas não o nó $\{3, 379\}$ por conter totais nos tempos $[4,5]$ que devem ser desconsiderados do conjunto resposta. Descendo mais um nível na B-tree temporal descrevendo a região R_6 , vê-se os intervalos $[3],[4,5]$ sendo mapeados e pode-se utilizar o nó $\{3, 125\}$ como dado final para formar a resposta da consulta. Ao final, têm-se que os nós $\{1, 685\}$ (primeiro nível da B-tree para R_5), $\{1, 259\}$

(primeiro nível da B-tree para R_3) e $\{3,125\}$ (segundo nível da B-tree para R_3) respondem, somados, à consulta.

3.4.3 STB-index

O trabalho de Tsuruda (2013) foi responsável pela incorporação de dados temporais no processamento do SB-index criando-se o STB-index (*Spatio-Temporal Bitmap Index*).

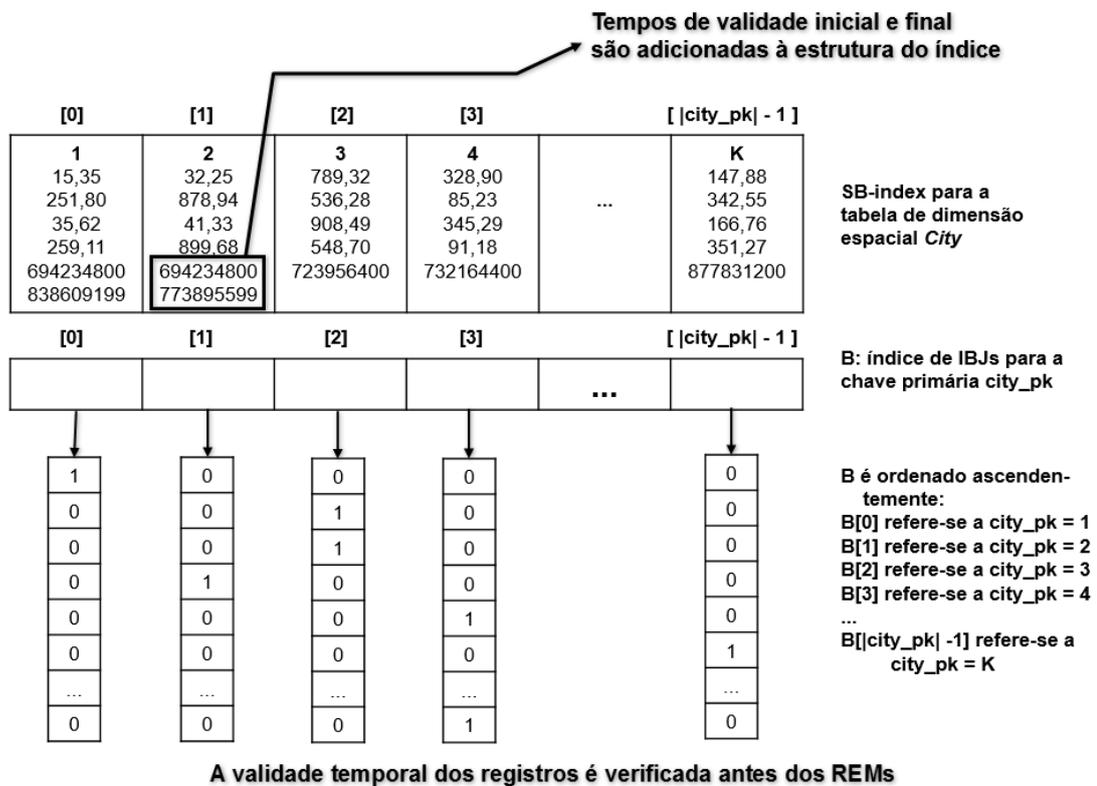


Figura 17: Estrutura do STB-index – adaptado de Tsuruda (2013)

Destinado a DWETs, o STB-index inclui os tempos de validade inicial e final de cada registro contido no vetor de chaves e REMs do SB-index com a intenção de excluir da avaliação espacial os registros que não estão contidos no intervalo temporal da consulta. O esquema de dados é exemplificado na Figura 17 e o processamento do índice utiliza os mesmos passos do SB-index (mostrados na Figura 12) com a diferença que a validade temporal de cada REM do vetor é analisada antes da avaliação espacial, excluindo-se do processamento os REMs temporalmente inválidos.

A exemplo do SB-index, o STB-index também apresentou melhores resultados no processamento de consultas STOLAP quando comparado ao método tradicional de visões materializadas.

3.4.4 HSTB-index

Proposta deste trabalho, o HSTB-index é a evolução do índice HSB-index para transformá-lo em um índice espaço-temporal.

A Figura 18 demonstra os nós da hierarquia de MBRs utilizada no HSTB-index e cuja estrutura (que foi derivada do HSB-index) possui as datas de validade inicial e final dos MBRs que se encontram nos nós folha.

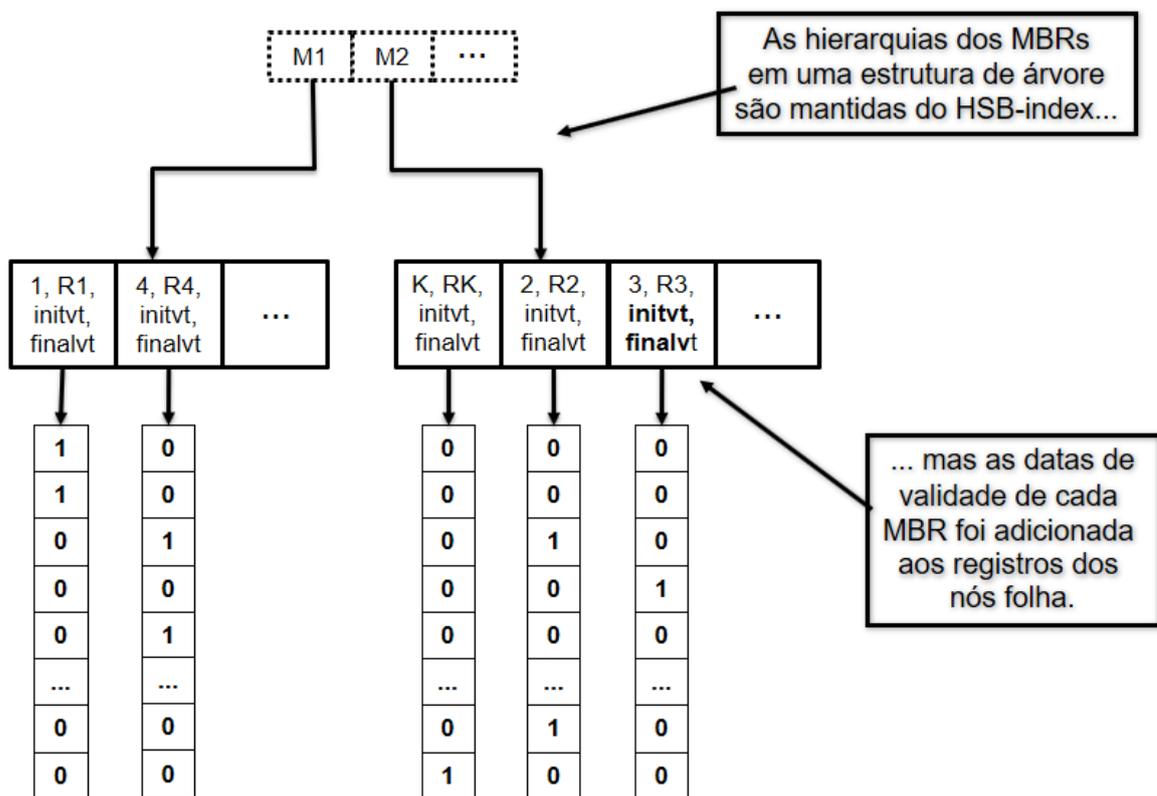


Figura 18: Estrutura do HSTB-index

Com a alteração na estrutura, foram necessárias modificações para carregar e executar o índice. Na parte de carga do índice foi necessário: i) alterar a consulta que realiza a formação dos MBRs para incluir a extração das datas de validade inicial e final dos mesmos no formato juliano; ii) alterar a carga dos nós para incluir

tal data na cópia dos registros em memória quando um nó folha fosse alcançado. O algoritmo de pesquisa do índice precisou incluir, além da verificação dos MBRs, a verificação da validade temporal destes quando um nó folha é identificado.

3.5 Benchmark Spatial SSB

O benchmark Spatial SSB (ou Spatial Star Schema Benchmark) foi proposto por Nascimento et. al. (2011) e oferece a possibilidade de construção de um DW para testes das várias situações de dados que podem ser encontradas nos bancos de dados espaciais. O benchmark possibilita a variação tanto da seletividade dos dados espaciais quanto dos dados convencionais, assim como o fator de escalabilidade e a complexidade das formas geométricas a serem geradas.

O Spatial SSB foi proposto com base no Star Schema Benchmark (O'NEIL, 2009) e o estende através da inclusão de colunas espaciais nas tabelas dimensão e na inclusão da limitação espacial nas consultas propostas pelo benchmark original. A Figura 19 mostra o esquema de dados do Spatial SSB onde se pode visualizar as colunas geográficas que foram adicionadas (colunas com final “_geo”), a informação da quantidade de registros a serem gerados de acordo com o fator de escala utilizado ($CSF * \langle \text{quantidade} \rangle$ no topo de cada tabela) e os relacionamentos entre cada uma das tabelas.

A adaptação teve que ser feita também nas consultas que poderiam ser feitas no DW gerado uma vez que estas precisaram incluir os vários níveis de hierarquia espacial para diferentes janelas de consulta. A Figura 20 exemplifica a adaptação feita à consulta Q3.1, onde espera-se obter a renda bruta obtida com a venda de produtos em um certo período de tempo.

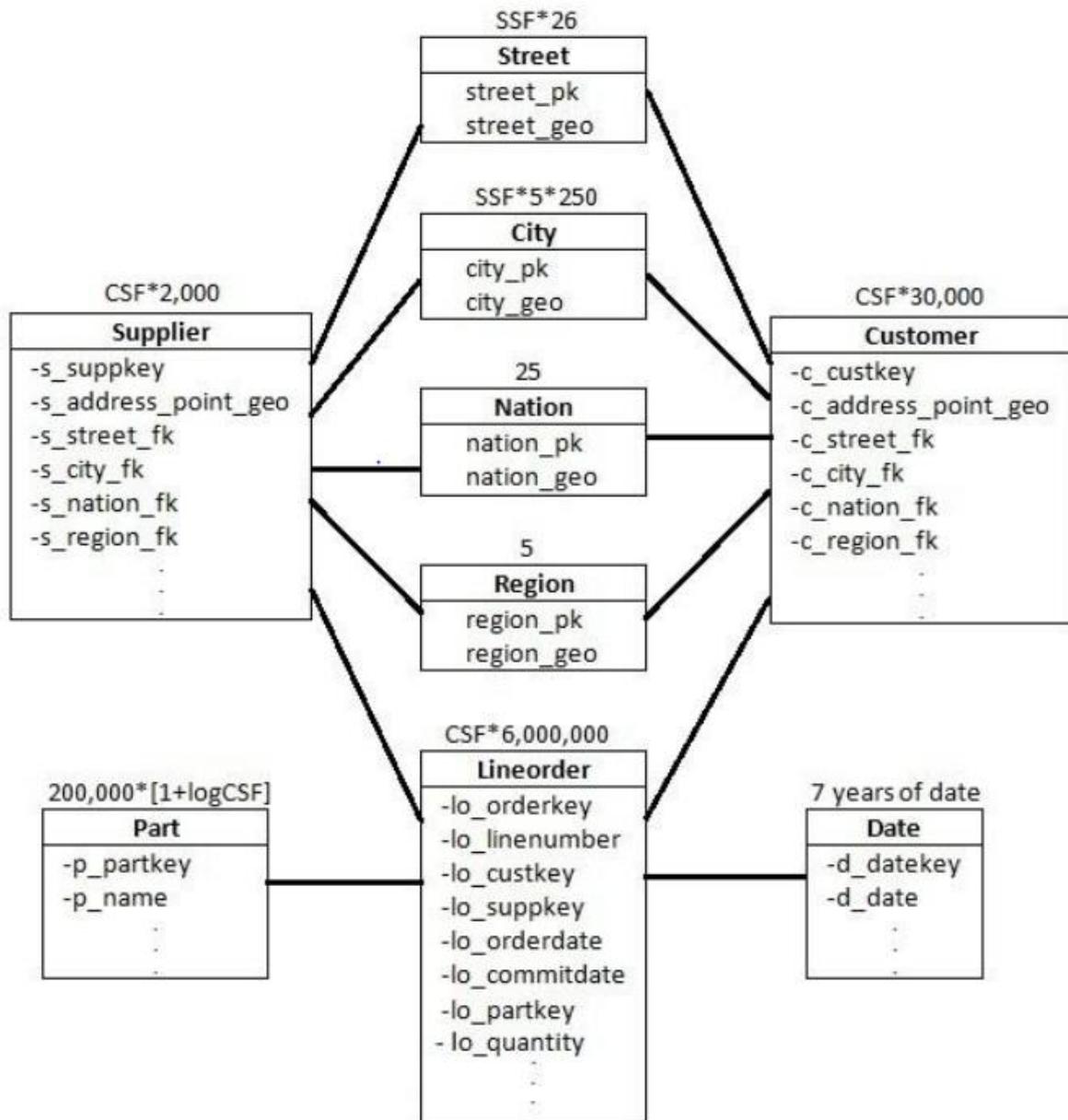


Figura 19: Esquema de dados do Spatial Star Schema Benchmark - retirado de Nascimento et. al. (2011)

```
SELECT C_NATION, S_NATION, D_YEAR, SUM (LO_REVENUE)
AS REVENUE
FROM CUSTOMER, LINEORDER, DATE, SUPPLIER
WHERE LO_ORDERDATE = D_DATEKEY
AND LO_CUSTKEY = C_CUSTKEY
AND LO_SUPPKEY = S_SUPPKEY
AND INTERSECTS (C_REGION, QW1) AND INTERSECTS (S_REGION, QW1)
AND INTERSECTS (C_NATION, QW2) AND INTERSECTS (S_NATION, QW2)
AND INTERSECTS (C_CITY, QW3) AND INTERSECTS (S_CITY, QW3)
AND INTERSECTS (C_STREET, QW4) AND INTERSECTS (S_STREET, QW4)
AND INTERSECTS (C_ADDRESS, QW5) AND INTERSECTS (S_ADDRESS, QW5)
AND INTERSECTS (EMPTY_AREA, QW6)
AND D_YEAR >= 1992 AND D_YEAR <= 1997
GROUP BY C_NATION, S_NATION, D_YEAR
ORDER BY D_YEAR ASC, REVENUE DESC
```

Figura 20: Consulta Q3.1 do Spatial Star Schema Benchmark – retirada de Nascimento et. al. (2011)

A consulta Q3.1 é a que servirá como base para este trabalho, mas a sua versão já estendida para o esquema de dados espaço-temporal como proposto por Tsuruda (2013).

3.5.1 Considerações Finais

Assim como a evolução do esquema de dados e suas possíveis consultas, foram apresentadas neste capítulo algumas das técnicas e índices que podem ser utilizadas para facilitar a execução de consultas em DWs, DWEs e DWETs, sendo que algumas delas serão empregadas neste trabalho.

Continuando a construção dos pilares que suportam este trabalho de mestrado, um trabalho correlato é apresentado no próximo capítulo.

Capítulo 4

TRABALHOS CORRELATOS

O trabalho a ser apresentado neste capítulo servirá de base e comparação quanto à exploração de algumas alternativas uma vez que esta pesquisa de mestrado o estende em alguns aspectos.

4.1.1 Especificação e Avaliação de Algoritmos para Processamento Multidimensional-Geográfico

O trabalho de Medeiros et al. (2008) possui certa similaridade com este trabalho de mestrado e as suas conclusões servirão de base para alguns aspectos a serem explorados.

A motivação destaca o fato de que um DWE precisa de ferramentas e métodos especiais para integrar a execução dos predicados convencional e espacial e cita os trabalhos de Kouba et al. (2002), Ferreira et al. (2001); Sampaio et al. (2006); Bimonte et al. (2007); Rivest et al. (2005) e Scotch e Parmanto (2006) para mostrar que isso é conseguido por meio da “cooperação entre um serviço multidimensional e uma ferramenta geográfica, sendo principalmente da interface gráfica a responsabilidade de abstrair do usuário a complexidade de manipular ambientes que operam separadamente e com linguagens de consulta diferentes”. Também é destacado que tais trabalhos não detalham a forma de execução das consultas.

A partir daí Medeiros et al. (2008) pondera que as consultas em um DWE podem ser de **mapeamento** (quando os elementos não possuem restrição espacial, i.e. o conjunto-resposta é formado apenas filtrando-se a parte multidimensional (com dados convencionais) e mostrados em um mapa) ou de **integração** (quando há restrições multidimensional e espaciais).

Por serem o foco do trabalho, as consultas de integração são pormenorizadas, sendo definido o conceito denominado **Consulta multidimensional-geográfica de mapeamento** como sendo um par ordenado de

constantes (*consultaMD*, *consultaGeo*), onde *consultaMD* denota o tipo de consulta Multi-Dimensional enquanto *consultaGeo* representa o tipo de consulta Geográfica.

Seguindo com o raciocínio, é ponderado que consultas de integração podem ser executadas limitando-se primeiro a parte multidimensional e depois a geográfica – (*consultaMD*, *consultaGeo*) – ou invertendo-se a ordem de limitação ao excluir-se primeiro os registros que não satisfaçam a parte geográfica – (*consultaGeo*, *consultaMD*).

Uma segunda definição é feita para uma **Consulta multidimensional-geográfica de integração** q onde esta é representada por um par ordenado (x, y) com x e $y \in \{consultaMD, consultaGeo\}$ e $x \neq y$. Usa-se o operador $FST(q)$ (em referência a *first* do inglês) para identificar o primeiro predicado a ser considerado durante o processamento da consulta q .

Dois algoritmos de execução são então apresentados sendo um para a execução dos predicados multidimensionais antes dos predicados geográficos e vice-versa (chamados *MDFirst* e *GEOFirst*, respectivamente).

Uma consideração importante é feita com relação ao tipo de agregação realizada na parte multidimensional das consultas. Shekhar et al. (2001) classifica as funções de agregação em um DW como **distributivas** (por exemplo, soma, mínimo, máximo e quantidade), **algébricas** (como média) e **holísticas** (como percentil, mediana e ranqueamento). Devido à sua característica de precisar acessar diretamente todos os registros do conjunto resposta para gerar um resultado correto, agregações holísticas são uma restrição à ordem de execução dos predicados e devem ser executadas somente após a aplicação da restrição geográfica. Já as funções distributivas e algébricas não restringem a ordem de execução dos predicados.

Após exemplos de tal constatação, o trabalho segue definindo o método de execução dos testes com a definição do esquema dos dados a ser utilizado, a carga dos dados e a definição dos operadores multidimensionais e geográficos a serem utilizados. Em uma parte subsequente, é apresentada uma forma de analisar estatisticamente a amostragem das consultas realizadas (uma vez que nem todas as combinações poderiam ser exercitadas) considerando um intervalo de confiança, a média amostral, o desvio padrão amostral e uma fórmula envolvendo o quartil de uma variável t-Student com $n-1$ graus de liberdade. Tal análise permite verificar quão distantes dois pares de valores são distintos entre si e onde, caso o intervalo de

confiança contenha o valor 0, pode-se concluir que os valores medidos não são significativamente diferentes.

Uma tabela com os resultados obtidos é mostrada ao final do trabalho concluindo-se que grande parte dos testes não apresentaram diferença significativa de desempenho com pequena diferença a favor do algoritmo *GEOFirst* quando tal diferença foi encontrada. Ao ponderarem sobre uma heurística para decidir qual o melhor algoritmo a ser usado para cada consulta a ser executada, os autores concluíram que não compensaria usá-la uma vez o ganho de tempo na execução das consultas não seria menor a ponto de sobrepujar o tempo dos cálculos adicionais. Uma consequência direta é a redução da complexidade do sistema.

4.1.2 Diferenças e Contribuições

Ao comparar o trabalho de Medeiros et al. (2008) a este trabalho de mestrado, pode-se notar que o primeiro se aplica a DWEs enquanto que esta pesquisa de mestrado enfoca DWETs e, portanto, uma nova definição de consulta se faz necessária para incluir o predicado temporal. Com relação ao escopo, este mestrado fica restrito a consultas de integração (onde todos os predicados estão presentes) e tal consulta pode ser redefinida como *Consulta multidimensional-geográfica-temporal de integração* onde um trio ordenado (ao invés de um par ordenado) pode ser definido com os valores *MDFirst*, *GEOFirst* e *TIMEFirst* para os predicados multidimensionais (chamados de convencionais neste trabalho), geográficos e temporais, respectivamente. A Tabela 2 mostra as principais diferenças encontradas entre os dois trabalhos.

Tabela 2 - Comparativo entre as características do trabalho correlato a esta dissertação de mestrado

Trabalho	Tipo de DW	Tipos de Consultas
Medeiros et al. (2008)	DWE	Mapeamento e Integração
Trabalho atual	DWET	Integração

Possíveis contribuições para esta dissertação vêm da consideração da função de agregação na parte multidimensional e da tratativa estatística dos dados. A primeira consiste na diferenciação das funções de agregação dos predicados

convencionais em distributivas, algébricas e holísticas para posterior limitação do escopo às funções distributivas e algébricas uma vez que as opções de implementação vistas na proposta de trabalho preveem a livre comutação na ordem de execução dos predicados. A segunda contribuição é a apresentação de uma possível maneira de análise estatística dos dados coletados e que pode se aplicar a este trabalho.

4.1.3 Considerações finais

Mais importante que as semelhanças, o trabalho correlato mostrou importantes diferenças com possibilidade de melhorias à proposta deste trabalho de mestrado.

Até aqui foram apresentados os conceitos teóricos e um trabalho relacionado formando as ferramentas necessárias para a apresentação da metodologia de pesquisa utilizada neste trabalho. Por fim, segundo o nosso conhecimento (orientador e orientando de mestrado), não existem na literatura outros trabalhos que investiguem a influência da ordem de execução de predicados convencionais e espaciais em consultas submetidas a DWEs, ou ainda dos predicados convencionais, espaciais e temporais em consultas submetidas a DWETs.

Capítulo 5

ATIVIDADES DE PESQUISA

Neste capítulo serão descritos os **objetivos** desta pesquisa de mestrado com relação à investigação do processamento de consultas STOLAP em data warehouses espaço-temporais, as **hipóteses** a serem confirmadas ou refutadas e a **metodologia** a ser utilizada para atingir os objetivos. Dentre outros itens, as principais **contribuições** esperadas e o **escopo** da pesquisa serão também delimitados.

5.1 Objetivo

Uma consulta STOLAP (*spatio-temporal OLAP*) possui três diferentes tipos de predicado. Mais especificamente, uma consulta STOLAP possui um predicado convencional (e.g. "idade > 10"), um predicado espacial (e.g. "cidade.geometry INTERSECTS query-window) e um predicado temporal (e.g. tempo_inicial >= "2013-01-01" AND tempo_final <= "2013-12-31"). Desta forma, no processamento de uma consulta STOLAP, deve-se escolher qual será a ordem de processamento destes três tipos de predicado, sendo que a ordem escolhida pode afetar o desempenho final do processamento de uma consulta STOLAP, devido à filtragem de tuplas em processamentos parciais.

A Figura 21 ilustra um exemplo de consulta STOLAP a ter sua execução investigada (TSURUDA, 2013). Para esta consulta, pode-se destacar os predicados convencional, espacial e temporal. Neste exemplo, o predicado convencional limita as tuplas ao selecionar um nome de cliente, o predicado espacial leva em consideração a intersecção de cidades com a janela de consulta (*intersection range query*) e o predicado temporal filtra as tuplas com base no tempo inicial e final de validade dos dados.

```

SELECT c_nation, s_nation, d_year,
       SUM(lo_revenue) as revenue
FROM customer, lineorder
WHERE lo_custkey = c_custkey
      AND lo_suppkey = s_suppkey
      AND lo_orderdate = d_datekey
      AND c name = 'Customer#000000001'
      AND INTERSECTS(JC, city)
      AND initialvt = 1992
      AND finalvt = 1997
GROUP BY c_nation, s_nation, d_year
ORDER BY d_year ASC, revenue DESC

```

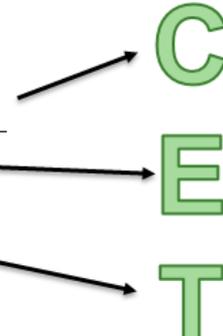


Figura 21: Exemplo de consulta STOLAP a ser utilizada - adaptado de Tsuruda (2013)

O objetivo desta pesquisa de mestrado é investigar a influência da ordem de processamento dos predicados convencional, espacial e temporal no desempenho de consultas STOLAP. Tal processamento dos predicados serão apoiados pelo uso de índices, sendo planejado o uso de índices bitmap de junção para apoiar o processamento do predicado convencional, o uso dos índices SB-index e HSB-index para apoiar o processamento do predicado espacial, o uso de consultas ao banco de dados e índices bitmap para apoiar o processamento do predicado temporal e o uso do índice STB-index para apoiar o processamento do predicado temporal e espacial conjuntamente, nesta ordem de processamento. Desde que o índice STB-index primeiramente processa o predicado temporal e depois processa o predicado espacial, ele será considerado no restante desta seção na descrição das propostas como um índice que apoiará o predicado temporal.

Além disso, pretende-se investigar se a seletividade dos predicados influencia na escolha da melhor ordem de processamento de uma consulta STOLAP. Neste trabalho entende-se por seletividade de uma consulta a quantidade de registros (ou tuplas) recuperados na sua resposta (ou na resposta parcial de um predicado). Com isto, uma alta seletividade indica uma maior quantidade de registros recuperados, enquanto que uma baixa seletividade indica uma menor quantidade de registros recuperados (CIFERRI, 2002).

Conforme descrito em Siqueira et al. (2012), a seletividade influenciou no tempo de processamento das consultas SOLAP, especificamente no caso do uso

dos índices SB-index e HSB-index. O trabalho de Tsuruda (2013) incluiu a parte temporal ao considerar o tempo de validade dos MBRs e essas duas situações trouxeram à tona as seguintes perguntas: Como a seletividade espacial afeta a eficiência do uso dos índices desenvolvidos? A seletividade temporal também teria influência? A ordem de execução dos predicados convencional, espacial e temporal de uma consulta STOLAP pode efetivamente afetar o desempenho? Executar os índices separadamente e utilizar os resultados como filtro para a execução usando os IBJs produziriam um melhor resultado do que usar o resultado de algum dos predicados da consulta para limitar os candidatos a serem processados pelos predicados subsequentes?

5.2 Hipóteses

Pensando em responder tais perguntas, as seguintes **hipóteses** serão analisadas:

H1: A seletividade de um predicado (de uma consulta STOLAP) afeta o custo de processamento de uma consulta STOLAP. Quanto maior a seletividade de um predicado, maior a quantidade de tuplas selecionadas em uma tabela de dimensão e consequentemente maior o custo da junção requerida para se processar o esquema estrela de um data warehouse, no caso da consulta STOLAP ser processada sem a ajuda de índices. Quando a consulta STOLAP é apoiada com o uso de índices, quanto maior a seletividade de um predicado, maior a quantidade de elementos a serem processados no índice e consequentemente maior o custo para se percorrer vários ramos e nós de uma árvore n-ária ou para se processar diferentes entradas em vetores baseados em um índice bitmap (por exemplo, para o índice SB-index).

H2: No processamento de uma consulta STOLAP, o uso de um índice é apropriado somente para pequenos valores de seletividade para os predicados que o índice apoia. Uma alta seletividade não favorece o uso de um índice. O valor a ser considerado como baixo ou alto para a seletividade é empírico e varia conforme o tipo de predicado. Por exemplo, para predicados espaciais, Siqueira (2012) considerou consultas retornando menos de 200 registros (dentro um total de 1.000.000 de registros) na fase de filtragem para indicar uma baixa seletividade.

H3: A seletividade dos predicados convencional, espacial e temporal na maioria dos casos é diferente e em alguns casos a seletividade dos predicados espacial e temporal é bem menor do que a seletividade do predicado convencional. Portanto, processar primeiro os predicados espacial e temporal e por fim o predicado convencional gerará um melhor desempenho no processamento de consultas STOLAP.

H4: O custo do processamento do predicado espacial tende a ser muito maior do que o custo do processamento do predicado temporal. Isto ocorre em função do alto custo para se processar dados espaciais compostos geralmente por muitos vértices (pontos) e com a execução de operações complexas para se detectar relacionamentos topológicos, mesmo para valores bem menores de seletividade para o predicado espacial. Portanto, processar primeiramente o predicado temporal para atuar como filtro para o processamento do predicado espacial é a melhor ordem de processamento destes dois predicados.

H5: O processamento independente dos predicados convencional, espacial e temporal com a geração de listas de resultados parciais contendo os IDs das tuplas que satisfazem o predicado e o subsequente processamento das listas de resultados parciais por meio de processamento cosequencial (i.e. interseção e união de listas de IDs para computar operadores AND e OR) possui custo maior do que usar um *pipeline* de processamento para os predicados convencional, espacial e temporal, onde cada predicado atua como um filtro para o próximo predicado do *pipeline*.

H6: Processar uma consulta STOLAP considerando o filtro da parte temporal em conjunto com a parte espacial (i.e., já limitando os seus resultados antes da fase de refinamento do predicado espacial) proverá melhor desempenho que o processamento separado dos predicados temporal e espacial.

Tentando confirmar ou negar as hipóteses H1 a H6, esta pesquisa visa investigar como a ordem de processamento dos predicados convencional, espacial e temporal afeta o desempenho no processamento de uma consulta STOLAP e adicionalmente, como diferentes valores de seletividade dos predicados convencional, espacial e temporal influenciam na escolha da melhor ordem de processamento dos predicados. A resposta a esses questionamentos ajudará outros pesquisadores a entenderem os fatores que podem influenciar no desempenho de consultas STOLAP e a traçar a melhor estratégia de processamento conforme a sua necessidade.

Para possibilitar uma investigação mais completa, temos como segunda contribuição desta pesquisa de mestrado a proposta e a implementação de uma evolução do índice HSB-index proposto por Siqueira et al. (2012) que será chamado HSTB-index. A implementação do HSTB-index introduzirá uma forma alternativa de hierarquizar a parte temporal tendo por objetivo a poda a priori de entradas que não estiverem dentro do tempo de validade especificado na consulta STOLAP.

5.3 Metodologia

As seções que seguem irão explicar a forma de realização dos experimentos, a carga de trabalho aplicada, a forma de construção de alguns dos dados e das seletividades, a ordem de processamento e o que se espera em cada uma delas.

5.3.1 Carga de trabalho

No tocante aos dados a serem utilizados, esta pesquisa utilizará como base o mesmo DWET utilizado por Tsuruda (2013) e cujo esquema estrela está representado na Figura 22. O modelo tem por base o esquema estrela apresentado pelo *Benchmark Spatial SSB* cujo esquema é híbrido, ou seja, as tabelas contendo os dados geográficos (*region*, *nation*, *city* e *street* respectivamente) são compartilhadas pelas dimensões *customer* e *supplier* que possuem, elas próprias, uma coluna geográfica com o endereço do cliente ou do fornecedor, respectivamente. A parte temporal é incorporada ao esquema pela definição das colunas *initialvt* e *finalvt* nas tabelas com dados geográficos *street*, *city*, *nation* e *region*, assim como nas tabelas *customer* e *supplier* que são tabelas de dimensões do esquema. A tabela de fatos é chamada *lineorder* e possui como chave primária as colunas *lo_orderkey* e *lo_linenum* e ligação com cada uma das tabelas de dimensão através das colunas *lo_custkey* (para a tabela *Customer*), *lo_partkey* (para a tabela *Part*), *lo_suppkey* (para a tabela *Supplier*) e *lo_orderdate* (para a tabela *Date*).

Quanto às consultas STOLAP a serem verificadas, o escopo desta pesquisa ficou limitado às consultas do *Benchmark Spatial SSB* incluindo-se a limitação

temporal como realizado por Tsuruda (2013) e utilizando predicados espaciais do tipo CRQ (*Containment Range Query*) e IRQ (*Intersection Range Query*). Outra restrição a ser respeitada é que os três tipos de predicados precisam estar presentes nas consultas para que a avaliação de todos os predicados envolvidos possa ser feita.

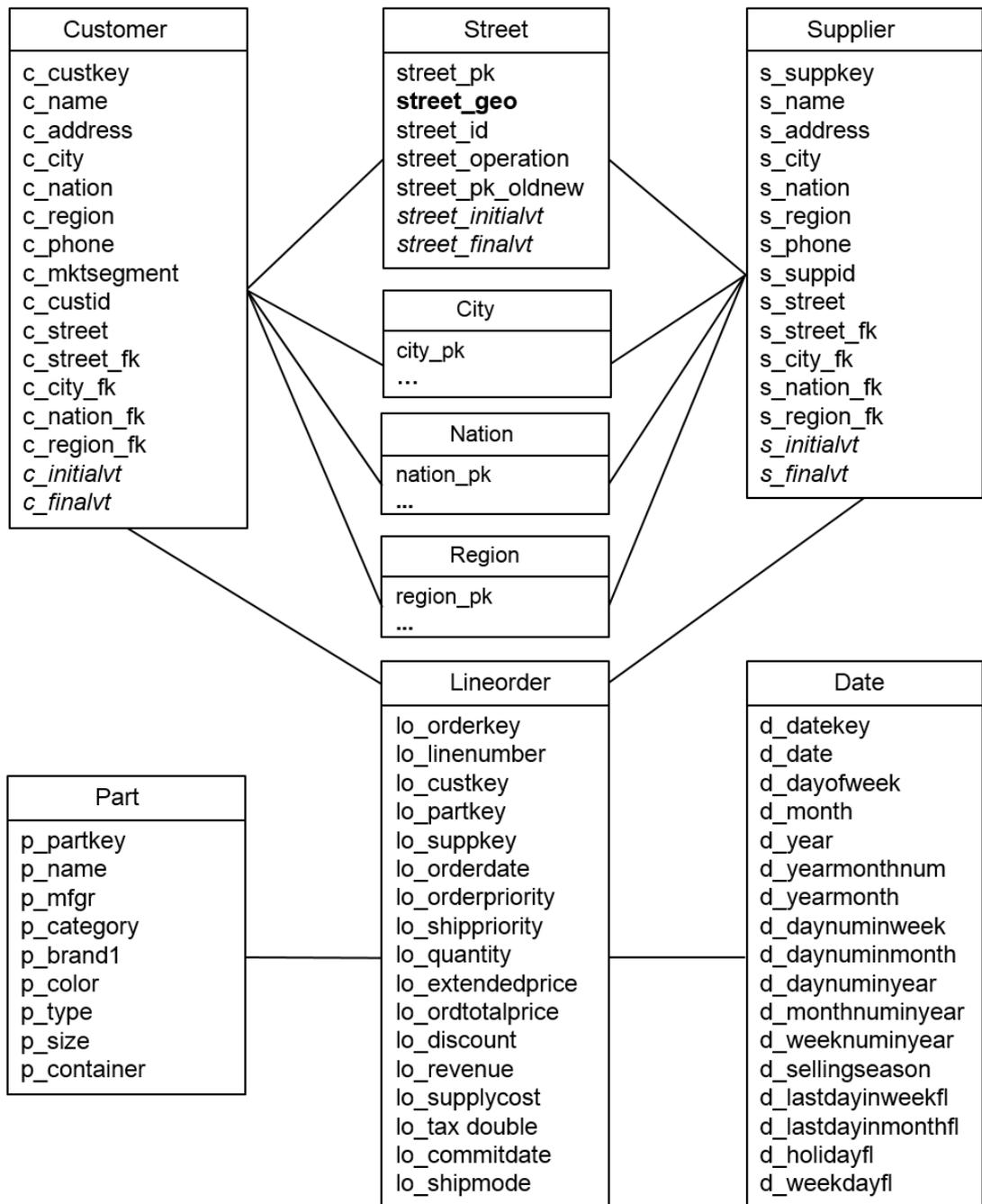


Figura 22: Esquema estrela a ser utilizado no desenvolvimento da pesquisa – adaptado de Tsuruda (2013)

Para este trabalho, a consulta que foi adaptada para se incluir a parte temporal foi a Q3.1 cuja estrutura por ser vista na Figura 25. Ela sumariza a quantidade de receitas brutas agrupadas por clientes, fornecedores e ano e foi escolhida por possibilitar o acesso a todos os tipos de dados espaciais a serem investigados.

As Figura 23 e Figura 24 exemplificam a extensão feita às consultas Q1.2 e Q1.3 respectivamente do Spatial SSB cujas seletividades espaciais mais se assemelharam ao intervalo de seletividade testado por Siqueira (2012). Os predicados estão evidenciados por diferentes estilos de letras onde a parte sublinhada corresponde ao predicado Convencional e pela letra “C”, em negrito tem-se os possíveis predicados Espaciais (letra “E”) e, por último, as linhas em *itálico* exemplificando o predicado Temporal (letra “T”).

Q1.2

```

SELECT SUM(lo_extendedprice*lo_discount) as REVENUE
FROM lineorder, date
WHERE lo_orderdate = d_datekey
  AND d_yearmonthnum = 199401
  AND lo discount BETWEEN 4 AND 6
  AND lo quantity BETWEEN 26 AND 35
  AND INTERSECTS (REGION, QW)
  AND INTERSECTS (NATION, QW)
  AND INTERSECTS (CITY, QW)
  AND INTERSECTS (STREET, QW)
  AND INTERSECTS (ADDRESS, QW)
  AND initialvt = 1992
  AND finalvt = 1997

```

The diagram illustrates the mapping of SQL predicates to the acronym CEST. The 'C' (Convencional) is linked to the underlined WHERE clause. The 'E' (Espaciais) is linked to the five INTERSECTS clauses. The 'T' (Temporal) is linked to the initialvt and finalvt clauses.

Figura 23: Consulta Q1.2 do Benchmark Spatial SSB modificada para se tornar uma consulta STOLAP

As consultas Q1.2 e Q1.3 procuram mostrar a quantidade de receita que se poderia ganhar se certos descontos fossem revogados. A diferença entre elas está na seletividade dos predicados convencionais, uma vez que a primeira limita os dados a um mês e a segunda a apenas uma semana. Além disso, as janelas de consulta espacial (QW no segundo parâmetro da cláusula *INTERSECTS*) também podem variar assim como a granularidade espacial que se deseja (*Region, Nation, City* ou *Street*).

Q1.3

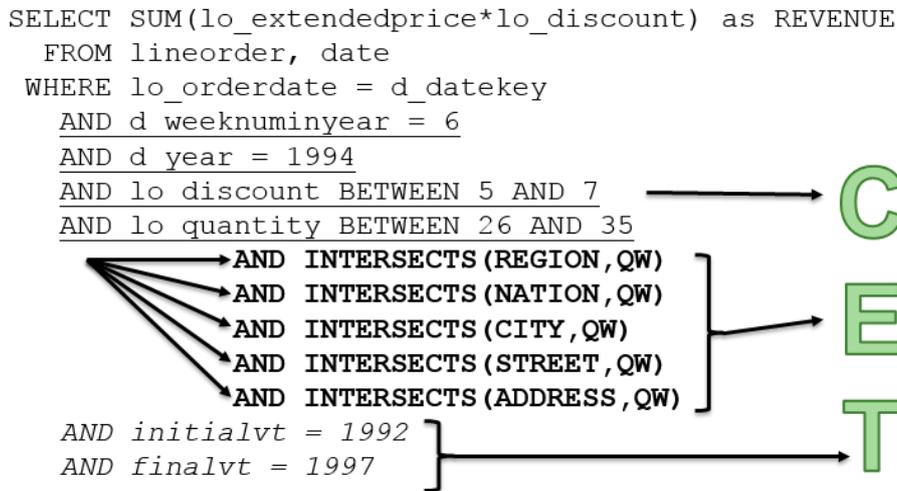


Figura 24: Consulta Q1.3 do Benchmark Spatial SSB modificada para se tornar uma consulta STOLAP

Q3.1

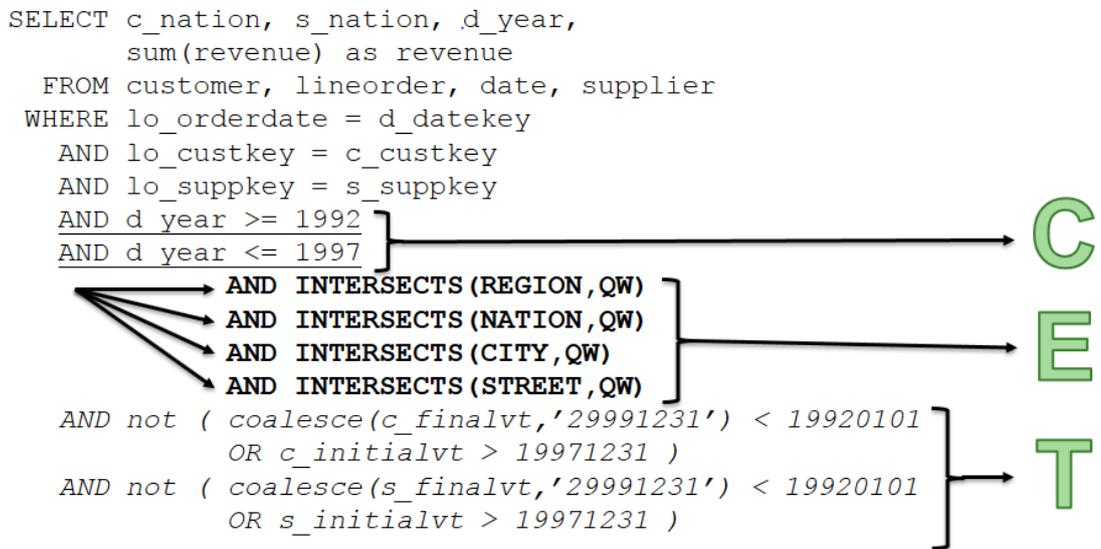


Figura 25: Consulta Q3.1 do Benchmark Spatial SSB modificada para se tornar uma consulta STOLAP

Os tipos de dados espaciais a serem investigados correspondem aos seguintes tipos e tabelas:

- a) **Pontos** – representam os endereços de clientes e fornecedores armazenados nas tabelas “customer” e “supplier”, respectivamente. As consultas utilizando este tipo de dado são identificadas pelos nomes **caddress** e **saddress**, também respectivamente;
- b) **Linhas** – representam as ruas onde os clientes e fornecedores estão localizados e cujos dados pertencem a tabela “street”. As consultas utilizando este tipo de dado são identificadas pelo nome **street**;
- c) **Polígonos** – representam as cidades onde as ruas estão localizadas. Seus dados vêm da tabela “city” e as consultas utilizando tal tipo de dado estão identificadas pelo nome **city**.

Os dados temporais foram gerados automaticamente através de scripts os quais, dada uma porcentagem de alteração, selecionam aleatoriamente os endereços de clientes e fornecedores a serem atualizados, bem como geometrias de ruas e cidades a serem alteradas. Tais alterações de geometria representam operações de junção ou divisão de linhas (para as ruas) e polígonos (para as cidades) (TSURUDA, 2013).

Como mencionado no item 5.1, os experimentos preveem uma variabilidade da seletividade (quantidade de registros selecionados) onde consideraremos os seguintes valores:

- a) **S0** – muito baixa – a quantidade de registros selecionados após a filtragem espaço-temporal é de 1 a 2 (ordem 10^0);
- b) **S1** – baixa – a quantidade de registros selecionados após a filtragem espaço-temporal é de 10 a 20 (ordem 10^1);
- c) **S2** – média – a quantidade de registros selecionados após a filtragem espaço-temporal é de 100 a 200 (ordem 10^2);
- d) **S3** – alta – a quantidade de registros selecionados após a filtragem espaço-temporal é de 1000 a 2000 (ordem 10^3);

Uma vez definidas as seletividades e os tipos de dados espaciais a serem investigados, a Tabela 3 sumariza as 16 configurações de processamento a serem investigadas.

Tabela 3: Configurações de processamento a serem investigadas

		Tipo de dado espacial			
		Pontos de clientes	Pontos de fornecedores	Linhas de ruas	Polígonos de cidades
Seletividade	muito baixa	S0-caddress	S0-saddress	S0-street	S0-city
	baixa	S1-caddress	S1-saddress	S1-street	S1-city
	média	S2-caddress	S2-saddress	S2-street	S2-city
	alta	S3-caddress	S3-saddress	S3-street	S3-city

Baseando-se na consulta adaptada da Figura 25, para se atingir a contagem aproximada de registros espaço-temporais acima, foi fixada uma janela temporal com seletividade maior que o total de registros alvo e alterada a janela de consulta espacial até que se pudesse atingir a seletividade exigida. Para as seletividades mais baixas, foi necessário inserir registros na tabela fato (*“lineorder”*) para que se pudesse encontrar registros na resposta final após a limitação convencional. Para as seletividades mais altas, registros na tabela de fornecedores (*“supplier”*) precisaram ser inseridos de maneira a se conseguir uma única janela espacial que atendesse a ambas as situações (endereços de clientes e fornecedores). A Tabela 4 contém as quantidades totais de registros em cada uma das tabelas envolvidas nas consultas e a Tabela 5 sumariza as seletividades alcançadas em cada etapa do processamento e a porcentagem desta seletividade em relação ao total de registros da respectiva tabela da consulta.

Tabela 4 - Quantidade total de registros nas tabelas envolvidas nas consultas

Tabela	Número de Registros
Customer	95967
Supplier	7375
Street	42594
City	1399
Lineorder	6001171

Tabela 5 - Consultas com suas respectivas seletividades e porcentagens de seletividade temporal, espacial, espaço-temporal e convencional

Consulta	Seletividade temporal	Porc seletividade temporal	Seletividade espacial	Porc seletividade espacial	Seletividade espaço-temporal	Porc seletividade espaço-temporal	Seletividade convencional	Porc seletividade convencional
S0-caddress	30000	31,261	6	0,006	2	0,002	67	0,001
S0-saddress	2482	33,654	5	0,068	2	0,027	914	0,015
S0-street	41931	98,443	3	0,007	3	0,007	2	0,000
S0-city	1263	90,279	4	0,286	2	0,143	73	0,001
S1-caddress	65968	68,740	27	0,028	19	0,020	75	0,001
S1-saddress	5375	72,881	17	0,231	12	0,163	2705	0,045
S1-street	41931	98,443	44	0,103	31	0,073	5	0,000
S1-city	1263	90,279	19	1,358	15	1,072	176	0,003
S2-caddress	65968	68,740	160	0,167	108	0,113	1604	0,027
S2-saddress	5375	72,881	179	2,427	123	1,668	25977	0,433
S2-street	41931	98,443	100	0,235	100	0,235	57	0,001
S2-city	1263	90,279	133	9,507	119	8,506	809	0,013
S3-caddress	65968	68,740	1860	1,938	1282	1,336	17531	0,292
S3-saddress	5375	72,881	1106	14,997	1071	14,522	16383	0,273
S3-street	41931	98,443	1020	2,395	1020	2,395	558	0,009
S3-city	1263	90,279	1364	97,498	1230	87,920	9506	0,158

Uma discussão à parte foi a consideração da janela de consulta temporal, que não tinha sido o foco principal do trabalho de Tsuruda (2013) mas que não podia ser posta em segundo lugar neste trabalho. A fim de atender a todas as granularidades temporais de maneira única, a tabela de visão materializada foi criada considerando datas com ano, mês e dia (ao invés de somente ano), o que gerou uma gama de possíveis combinações entre a data de validade dos registros (convencionando-se T1 e T2) e as datas da janela de consulta temporal (convencionando-se J1 e J2). O pensamento que surgiu foi que o registro precisa ser considerado como válido se sua janela de tempo de validade “tocar” a janela de consulta temporal em algum dos seus limites. A Figura 26 mostra todos os casos temporais possíveis para cada registro e a conclusão é que, dos 9 casos, apenas dois excluiriam o registro da consulta e, assim, ficaria mais fácil negar os dois casos na hora de considerar o predicado temporal das consultas a fim de simplificá-las.

Algumas das descrições a serem usadas no próximo capítulo possuem um número de consulta associado. Tal número é uma identificação interna usada na programação das opções de processamento e correspondem a um certo tipo de dado espacial e certa seletividade espaço temporal. São eles:

- a) **0, 4, 8 e 12** – as consultas desses números correspondem a pontos para a seletividade muito baixa, baixa, média e alta, respectivamente;
- b) **1, 5, 9 e 13** – as consultas desses números correspondem a linhas para a seletividade muito baixa, baixa, média e alta, respectivamente;

- c) **2, 6, 10 e 14** – as consultas desses números correspondem a polígonos para a seletividade muito baixa, baixa, média e alta, respectivamente.

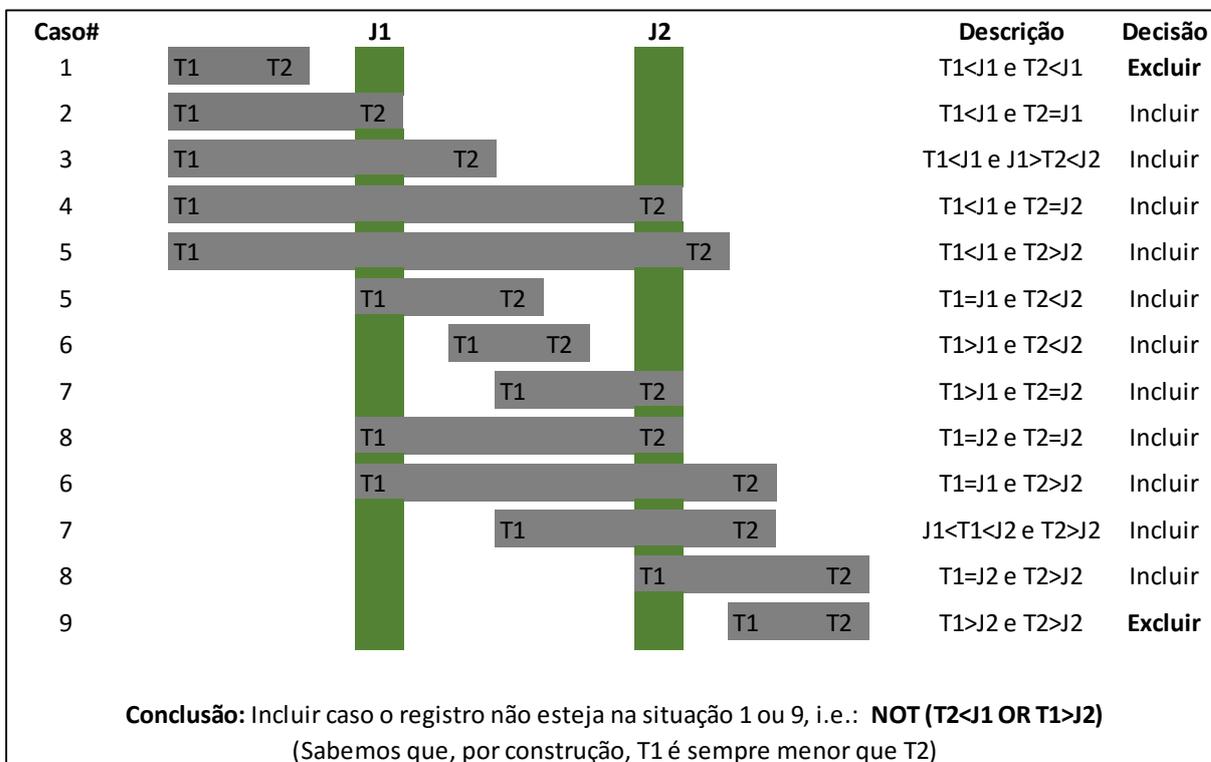


Figura 26: Possíveis casos temporais e a respectiva decisão de inclusão ou exclusão do registro.

Todas as variações acima auxiliarão nas combinações a serem comparadas contra o *baseline de visão materializada*, que consiste no pré-armazenamento do resultado de uma junção estrela e no processamento das consultas STOLAP diretamente na visão materializada para filtrar as tuplas dos predicados convencional, espacial e temporal. O *baseline de visão materializada* teve duas implementações. A primeira implementação de visão materializada foi baseada nos testes realizados por Siqueira (2009) e Tsuruda (2013) e compreende uma visão materializada sem nenhum índice para auxiliar na junção com as tabelas de dimensão depois da limitação espaço-temporal com índices espaciais (GiST) e convencionais para manipular data (B-tree) nas tabelas de dimensão. Já a segunda implementação de visão materializada foi proposta por este pesquisador com indexação inclusive das colunas da visão materializada responsáveis pela limitação dos candidatos após a limitação espaço-temporal nas tabelas dimensão. Para esta

segunda implementação, mais especificamente, foram criados os índices especificados na Figura 27.

Nome do índice	Tipo	Coluna	Tabela alvo	Tipo de dado	Descrição
mvq3_ymd_ix1	btree	c_custkey	Customer	caddress	Chave para a tabela de dimensão "Customer"
mvq3_ymd_ix2	btree	c_street_fk	Street	street	Chave para a tabela de dimensão "Street"
mvq3_ymd_ix4	btree	c_city_fk	City	city	Chave para a tabela de dimensão "City"
mvq3_ymd_ix5	btree	s_suppkey	Supplier	saddress	Chave para a tabela de dimensão "Supplier"
mvq3_ymd_ix6	btree	s_street_fk	Street	street	Chave para a tabela de dimensão "Street"
mvq3_ymd_ix7	btree	s_city_fk	City	city	Chave para a tabela de dimensão "City"
mvq3_ymd_ix8	btree	yearmonthday	-	-	Coluna convencional usada na limitação do predicado

Figura 27: Descrição dos índices usados na visão materializada altamente indexada

Respectivamente, as duas implementações de visão materializada foram chamadas de VMB (ou Visão Materializada Básica – sem indexação) e VMA (ou Visão Materializada Avançada – altamente indexada). A intenção inicial era eleger a que melhor desempenhasse dentre as configurações de processamento espaço-temporal. Porém, o comportamento das duas implementações de visão materializada variou bastante e nenhum padrão foi encontrado. Assim, decidiu-se pela utilização da opção de visão materializada que obteve o melhor resultado de desempenho em cada opção de processamento espaço-temporal. Desta forma, o resultado apresentado nos resultados de desempenho para a visão materializada corresponde ao melhor desempenho dentre as implementações VMA e VMB. A Tabela 6 mostra a diferença entre elas, inclusive com os tamanhos que elas ocuparam no banco de dados

Tabela 6: Comparação entre as duas opções de visão materializada utilizadas

Visão Materializada	Índices espaciais e temporais nas tabelas de dimensão	Índice nas colunas da visão materializada	Tamanho da tabela de VM (MB)	Tamanho dos Índices da VM (MB)
VMA	Sim	Sim	735,49	899,06
VMB	Sim	Não	735,49	0,00

Uma vez definidos o esquema de dados e a forma da construção destes, as próximas seções descreverão as variações de processamento que foram exploradas nesta pesquisa.

5.3.2 Ambiente Computacional e Medidas de Desempenho

O ambiente computacional utilizado nos testes de desempenho foi um computador com 2 processadores Intel Core i7-4770 CPU a 3.40GHz de clock (num total de 8 núcleos de processamento), 32 GB de RAM, dois discos rígidos de 4 TB com o sistema operacional CentOS 6.5.

Os programas utilizados foram:

- PostgreSQL 9.3.3 – SGBD objeto-relacional de código aberto;
- PostGIS 2.2.0 – extensão espacial para o PostgreSQL também utilizando código aberto;
- FastBit 1.3.8 – software para mapear e utilizar índices bitmap;
- gcc, gcc-c++ 4.4.7-11.el6 – Compiladores C e C++.

Os tempos são coletados através de scripts automatizados que executam 25 vezes cada opção de processamento com limpeza de caches e buffers entre cada uma delas. O tempo final corresponde à média aritmética dos 25 tempos coletados. Procurando melhor automatizar a tomada de tempo, há a inserção dos tempos para cada predicado da consulta STOLAP diretamente em tabelas no banco de dados PostgreSQL. Os tempos para processamento entre os predicados da consulta que são mandatórias à técnica sendo executada (como ordenação de chaves ou processamento de listas) são também coletados enquanto outros processamentos que não são pertinentes à execução das opções são desprezados.

5.4 Técnicas de Processamento

A partir dos conceitos e das várias maneiras de se processar os predicados de uma consulta STOLAP apresentados no Capítulo 3 - , as técnicas de processamento elencadas na Figura 28 serão utilizadas nos testes de desempenho.

A exploração das alternativas considerará os três tipos de predicados presentes em uma consulta STOLAP (i.e. convencional, espacial e temporal) e as técnicas utilizadas para a execução destes (i.e. o uso de diferentes tipos de índices).

As alternativas serão comparadas tomando-se como *baseline* o processamento utilizando visões materializadas.

O primeiro tipo de predicado a ser descrito é o convencional (representado pela letra “C”) que é compreendido por colunas contendo tipos de dados já bastante explorados nos bancos de dados comerciais como texto, datas simples e números, os quais possuem relação de ordem total. Pretende-se processar predicados referentes a tais colunas utilizando-se IBJs e a sua implementação no *software* Fastbit. A representação a ser utilizada para este tipo de predicado está mostrada como “Cf” da Figura 28.

Os predicados espaciais vêm em seguida (representados pela letra “E”) e estes correspondem a geometrias como polígonos para representar regiões e cidades, linhas para representar ruas e pontos para representar os endereços de clientes e fornecedores. Este trabalho irá se valer do índices SB-index e do HSB-index para a execução deste tipo de predicados em data warehouses cujas representações estão exemplificadas respectivamente nos itens “Esb” e “Ehsb” da Figura 28.

Em seguida, tem-se os predicados temporais (representados pela letra “T”) que correspondem à verificação das datas de validade dos registros das dimensões com quatro possibilidades de implementação: i) como uma consulta SQL ao SGBD PostgreSQL com limitação temporal a partir das datas de validade inicial e final dos registros (“Tp”); ii) como uma consulta SQL ao SGBD PostgreSQL com limitação temporal adicionada de limitação aos candidatos espaciais já encontrados em processamento espacial anterior (“Tep”); iii) como uma consulta ao *software* de indexação bitmap (Fastbit) utilizando-se apenas a limitação temporal (“Tf”); iv) como uma consulta ao *software* de indexação bitmap (Fastbit) utilizando-se a limitação temporal adicionada de limitação aos candidatos espaciais encontrados em processamento espacial anterior (“Tef”).

O uso do índice STB-index (“TEstb”) pode ser considerado um caso à parte no processamento temporal, pois ele pressupõe a execução do predicado temporal imediatamente seguido da parte espacial não sendo possível a opção contrária. O HSTB-index (“TEhstb”) compreende a implementação do HSB-index adicionada de limitação temporal e também possui essa característica à parte. Ambos são representados pelas letras “TE” na Figura 28.

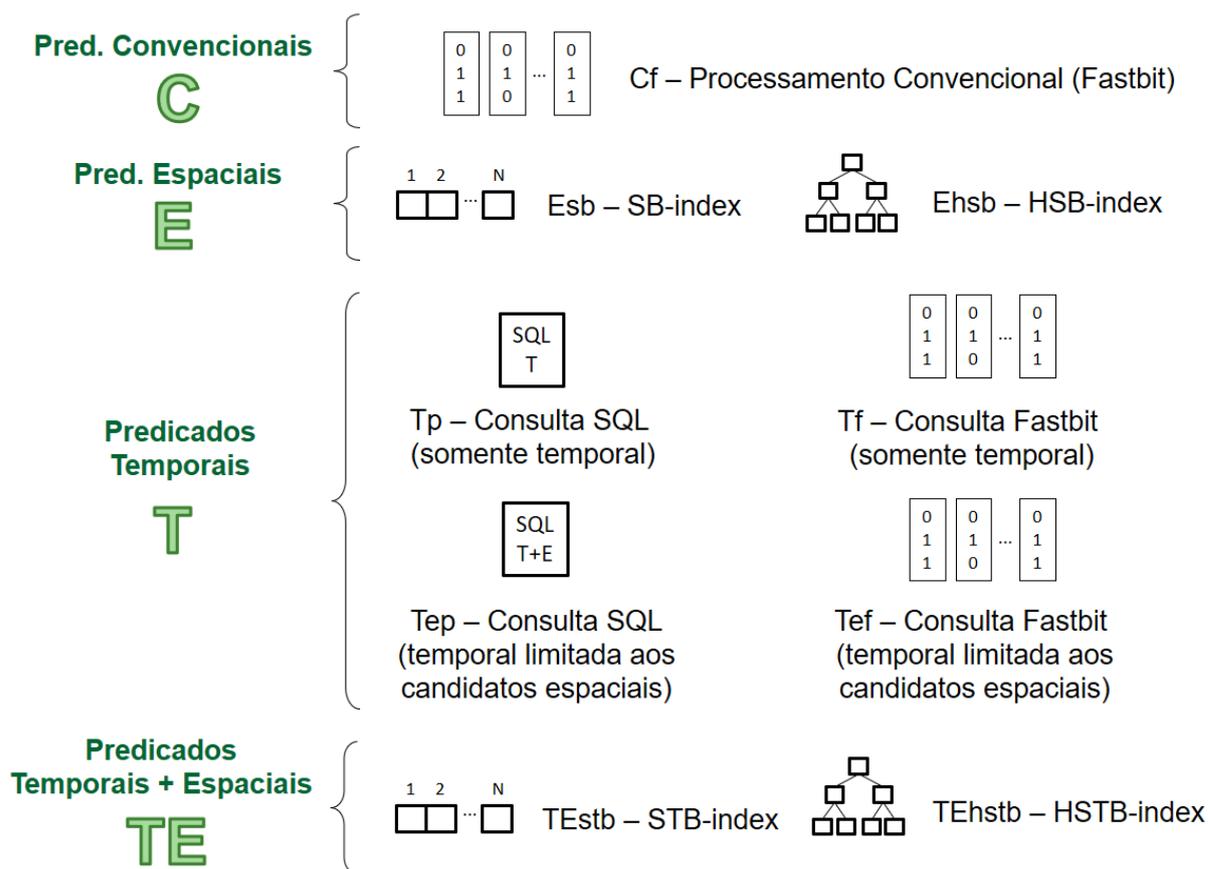


Figura 28: Técnicas de processamento em consultas STOLAP e a sua representação gráfica

As técnicas de processamento dos predicados elencadas acima podem ser combinadas de diversas maneiras e, a seguir, descreve-se as que foram consideradas nos testes de desempenho desta pesquisa.

5.4.1 Processamento em Pipeline dos Predicados Espacial, Temporal e Convencional

O processamento em *pipeline* indica a tentativa de se limitar o processamento de um predicado com os candidatos já encontrados no processamento de um predicado anterior. A intenção é limitar ao máximo cada etapa até que um número mínimo de registros seja considerado na etapa final de consulta ao *software* de indexação bitmap que irá fornecer a resposta final da consulta.

A Figura 29 mostra graficamente as opções de processamento em *pipeline* com o processamento do Predicado Espacial a priori usando os índices SB-index e HSB-index. Os candidatos apurados por estes índices são utilizados no

processamento do Predicado Temporal com as opções Tp (consulta SQL) ou Tf (consulta ao *software* de indexação bitmap). Os candidatos apurados até esta etapa são utilizados na consulta final ao *software* de indexação bitmap juntamente com o processamento dos Predicados Convencionais.

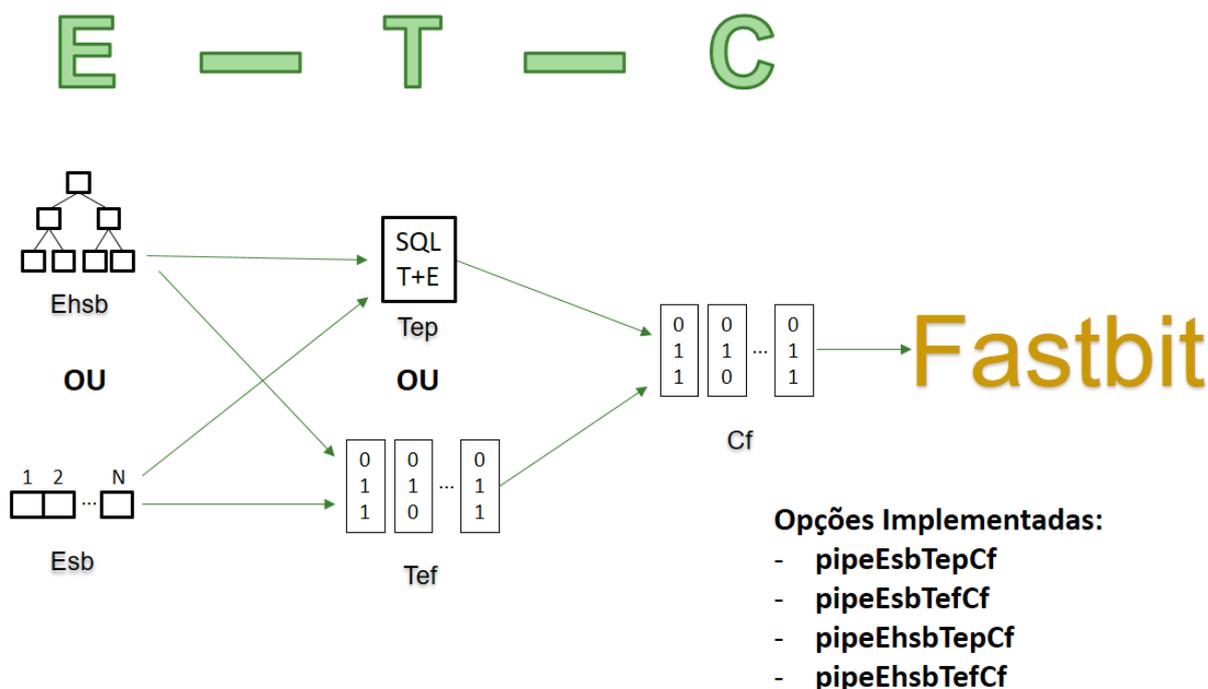


Figura 29: Opções em *Pipeline* com execução do Predicado Espacial a priori

Por conter códigos para cada uma das combinações da Figura 29 em separado, durante o desenvolvimento da pesquisa convencionou-se que as opções seriam identificadas pelo tipo de processamento seguido da identificação do índice conforme as técnicas listadas na Figura 28.

Para melhor visualizar a relação entre as opções nas análises posteriores, classes de processamento foram idealizadas e as opções deste processamento fazem parte da classe “**Pipeline E-T-C**”.

No ponto de vista deste pesquisador e do seu orientador, o ponto forte desta classe de processamento é a limitação de seletividade entre etapas, o que pode gerar bons resultados.

Ainda se valendo do processamento em *pipeline*, pode-se inverter o tipo de predicado a ser processado a priori e, assim ter-se a classe de processamento “**Pipeline T-E-C**” cujas opções são evidenciadas na Figura 30.

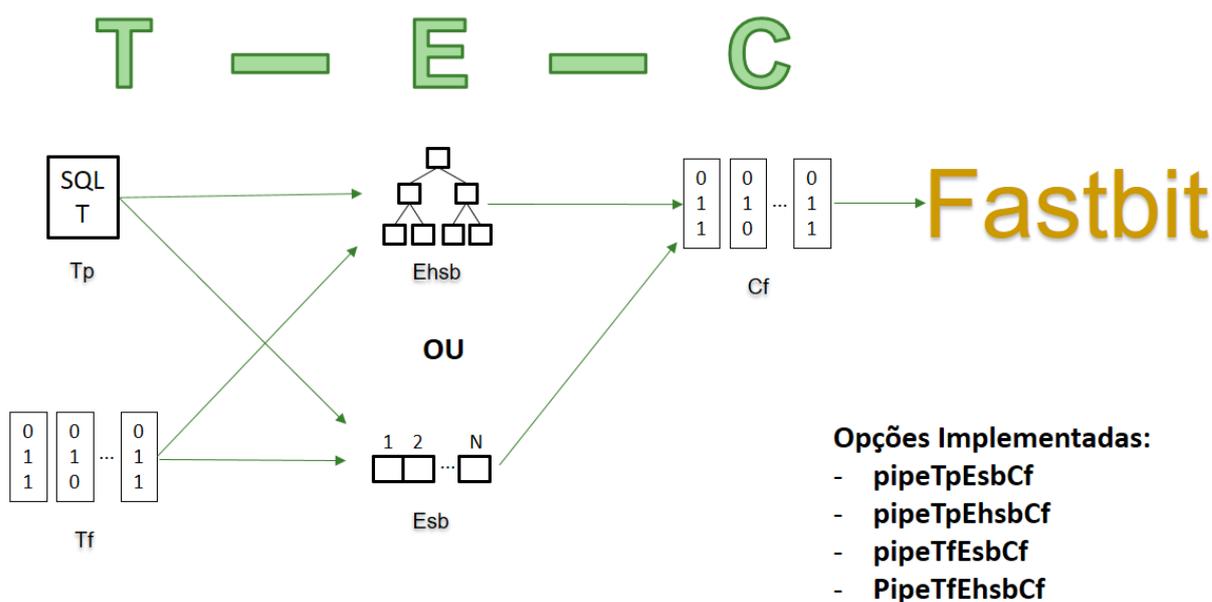


Figura 30: Opções em Pipeline com execução do Predicado Temporal a priori

Esta classe de processamento irá utilizar primeiro as técnicas de processamento do predicado temporal que não têm a limitação de predicados anteriores (“Tp” – consulta SQL – ou “Tf” – consulta ao software de indexação bitmap) seguida da limitação espacial através dos índices SB-index e HSB-index (“Esb” ou “Ehsb”) modificados para incluir na pesquisa somente as chaves já selecionadas como válidas temporalmente. O processamento continua com a limitação do predicado convencional (“Cf”) e da formação da resposta da consulta num último passo utilizando o *software* de indexação bitmap.

Assim como a classe de processamento “Pipeline E-T-C”, o ponto forte desta classe de processamento é proporcionar uma maior limitação de candidatos em cada etapa e a promessa de melhor desempenho a cada etapa.

A terceira e última classe de processamento em *pipeline* compreende as opções onde os índices STB-index e HSTB-index são utilizados em sua forma pura. Por já conter o processamento dos predicados Temporal e Espacial em uma única lógica, tal classe de processamento será chamada “**Pipeline TE-C**”. A representação de suas opções é encontrada na Figura 31.

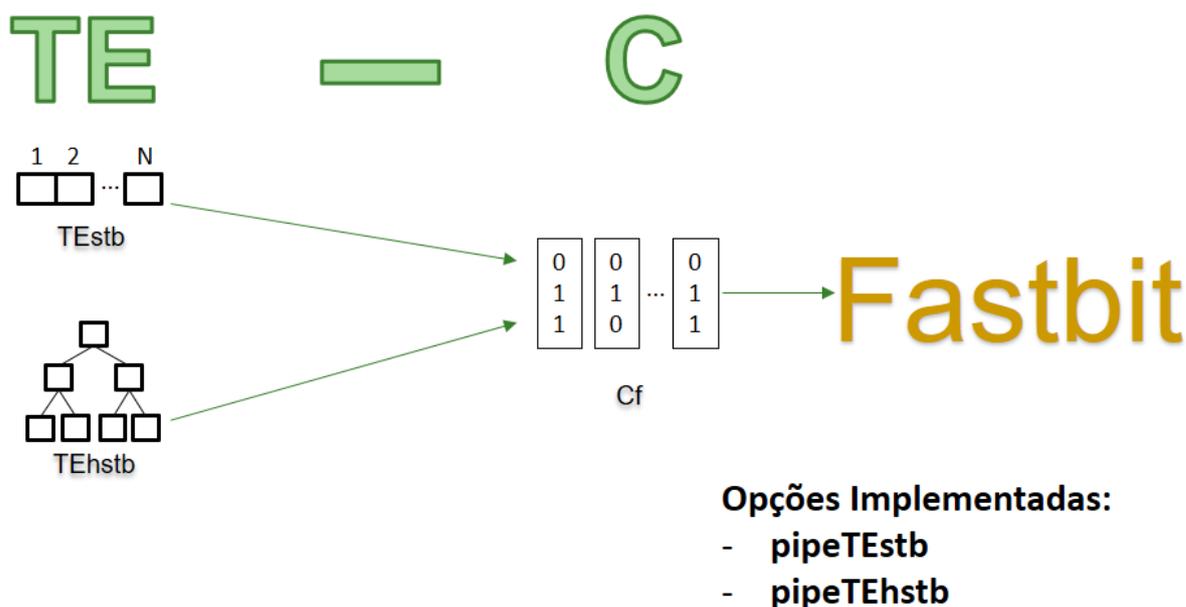


Figura 31: Opções em Pipeline com execução dos Predicados Temporal e Espacial num mesmo passo

O processamento realizado nos índices STB-index (“TEstb”) e HSTB-index (“TEhstb”) já contemplam ambos os predicados Temporais e Espaciais com a informação de validade temporal e o MBR da representação espacial na estrutura do índice e seu processamento pode ser feito localmente registro a registro, o que não acontecia nas outras classes. Após o processamento Espaço-Temporal, o processamento do predicado convencional (“Cf”) é feito utilizando o *software* de indexação bitmap e a lista de candidatos anteriores como limitador.

A opção de processamento pipeTEstb corresponde ao STB-index implementado por Tsuruda (2013) e a opção de processamento pipeTEhstb corresponde ao HSTB-index implementado por esta pesquisa.

5.4.2 Processamento Independente dos Predicados Espacial, Temporal e Convencional

A técnica de processamento Independente segue o caminho oposto do processamento em *pipeline* e pressupõe que cada predicado deva ser executado sem nenhum tipo de interferência do outro e os seus resultados guardados em listas de chaves candidatas. Tais listas são depois processadas duas a duas por um

algoritmo cosecucional de intersecção (operação AND) ou união de listas (operação OR) (FOLK e ZOELLICK, 1991).

As opções de processamento independente se dividem em duas classes, sendo a primeira composta pelas técnicas de processamento dos predicados que executam cada um deles separadamente – chamada “**Independente T/E/C**” – e a segunda por aquelas que os executam em conjunto – chamada “**Independente TE/C**”.

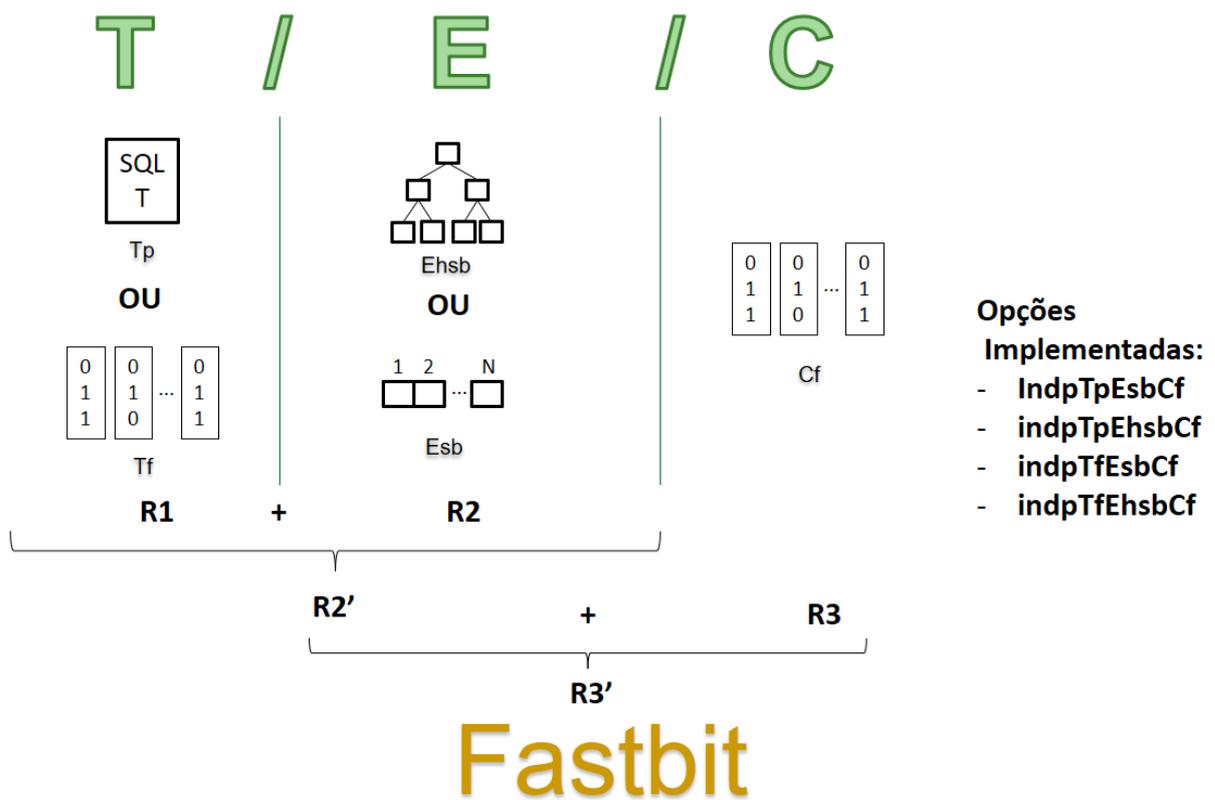


Figura 32: Opções de processamento Independente com execução dos Predicados Temporal, Espacial e Convencional separadamente.

A Figura 32 ilustra as possíveis combinações a serem utilizadas no processamento da classe “Independente T/E/C”. Para o predicado Temporal, somente as técnicas “Tp” (consulta SQL) e “Tf” (consulta ao software de indexação bitmap) podem ser utilizadas por não dependerem de resultados anteriores para a sua execução. Para o Predicado Espacial, as técnicas “Esb” (SB-index) e “Ehsb” (HSB-index) serão utilizadas. Para o Predicado Convencional, a técnica “Cf” prevê a consulta ao software de indexação bitmap para formação da lista de candidatos. O

processamento das listas duas a duas também são indicadas na Figura 32 com as listas R1 e R2 (registros temporalmente válidos e obedecendo à restrição espacial) gerando uma lista de candidatos Espaço-Temporais (R2') e esta última sendo processada cosequentialmente com a lista de candidatos Convencionais (R3). Os candidatos finais são obtidos na lista R3' que é usada para compor a consulta ao *software* de indexação bitmap que irá gerar a resposta da consulta.

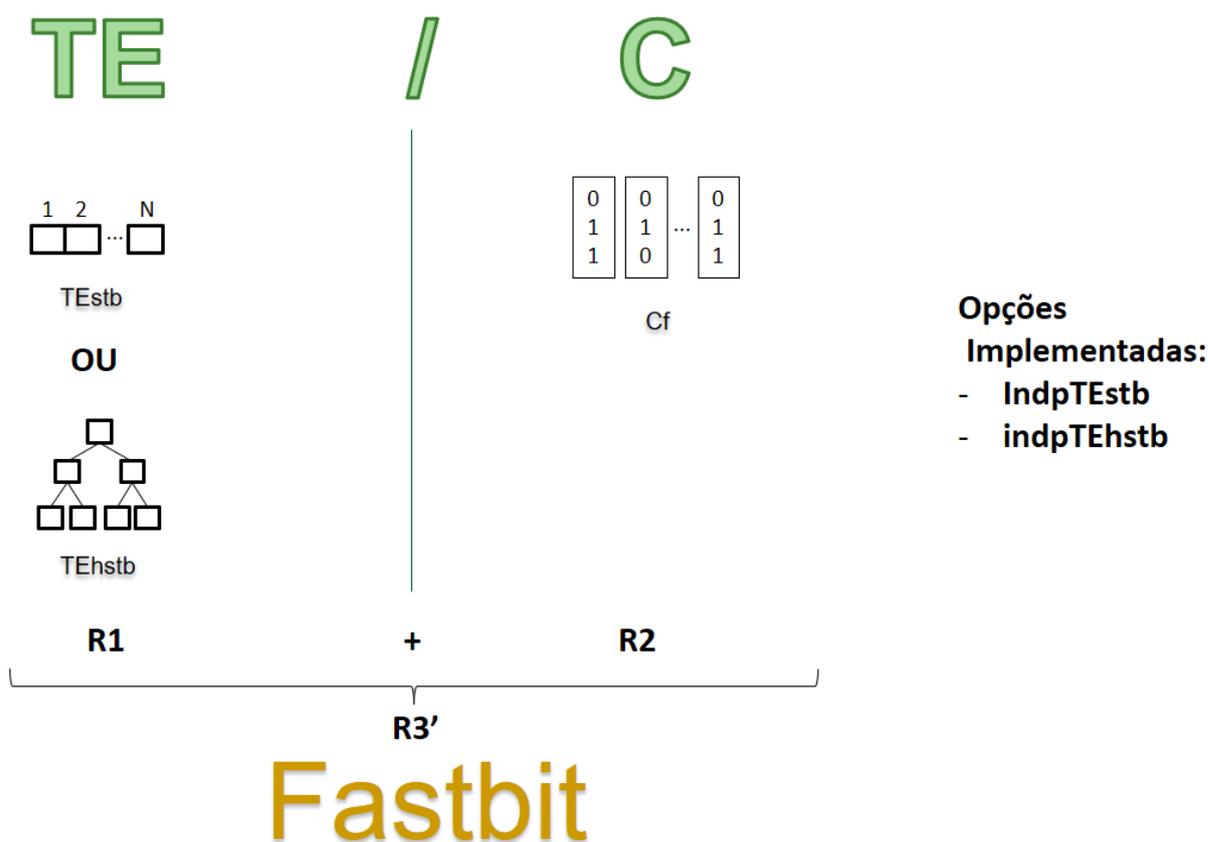


Figura 33: Opções de processamento Independente com execução dos Predicados Temporal e Espacial em conjunto e do Predicado Convencional em separado.

Exemplificada na Figura 33, a classe “Independente TE/C” irá se valer da mesma forma de processamento de listas cosequenciais com a diferença que há somente uma junção de listas, uma vez que o processamento dos Predicados Temporal e Espacial são feitos utilizando-se as técnicas “TEstb” (STB-index) e “TEhstb” (HSTB-index) já inerentemente Espaço-Temporais.

A expectativa para estas opções de processamento independente é de que o processamento dos predicados em separado seja mais simples e mais rápido a

ponto de compensar os processamentos adicionais demandados pela classe e, assim, rivalizar com as outras classes de processamento.

5.4.3 Processamento Intercalado dos Predicados Espacial e Temporal

Uma contribuição desta pesquisa de mestrado, as opções de processamento Intercaladas (*“Intermingled”*, em tradução livre) preveem a quebra da execução do predicado espacial considerando as fases de filtragem (F) e refinamento (R) e a inversão da ordem de processamento do predicado temporal. Nesse sentido, o predicado temporal será processado entre as fases F e R, formando o pipeline F-Temporal-R-Convencional (ou FTRC). O resultado esperado é que se reduza ainda mais a quantidade de objetos a serem processados na fase R do predicado espacial através da limitação temporal logo após a fase F. Esta configuração será testada para se verificar se ela é melhor do que as opções da classe “Pipeline T-E-C” já descrita no item 5.4.1.

O processamento desta configuração pode ser detalhado da seguinte forma: conforme foi apresentado nas seções 3.3.2 e 3.3.3, o processamento tanto do SB-index quanto do HSB-index passa por uma fase inicial de filtragem das tuplas com a avaliação dos MBRs dos objetos espaciais para depois fazer o refinamento das tuplas e eliminação dos falsos candidatos introduzidos pela aproximação feita com o MBR. O mesmo acontece com o STB-index (apresentado na seção 3.4.3) com a diferença que a filtragem através da parte temporal antecede a avaliação dos MBRs. Se a fase de filtragem espacial for representada pela letra “F” e a de refinamento espacial pela letra “R” (e ainda considerando as letras “C”, “E” e “T” para os predicados convencional, espacial e temporal respectivamente) a execução de “T-E-C” poderia ser reescrita como “T-F-R-C”, onde a limitação temporal seria a primeira a diminuir o número de chaves a serem consideradas no pipeline (Figura 34a). Outra configuração a ser pensada é a “E-T-C” (que resulta na configuração “F-R-T-C”), significando que a fase R (mais custosa) fica entre as fases F e T, que são de verificação mais simples e que poderia fazer a diferença no desempenho. A configuração a ser testada é a permuta entre as fases T e F resultando em “F-T-R-C”, i.e., a filtragem espacial é seguida imediatamente da filtragem temporal antes de entrar na fase de refinamento espacial. O intuito dessa configuração é limitar temporalmente os registros antes da avaliação espacial feita na fase de refinamento

do índice, a qual é muito custosa pois requer o acesso à geometria exata do objeto espacial (que pode variar bastante) e a computação de caras operações sobre geometrias complexas. Além disso, a inversão traria vantagem quando a seletividade espacial for menor que a seletividade temporal. A Figura 34b ilustra tal alteração.

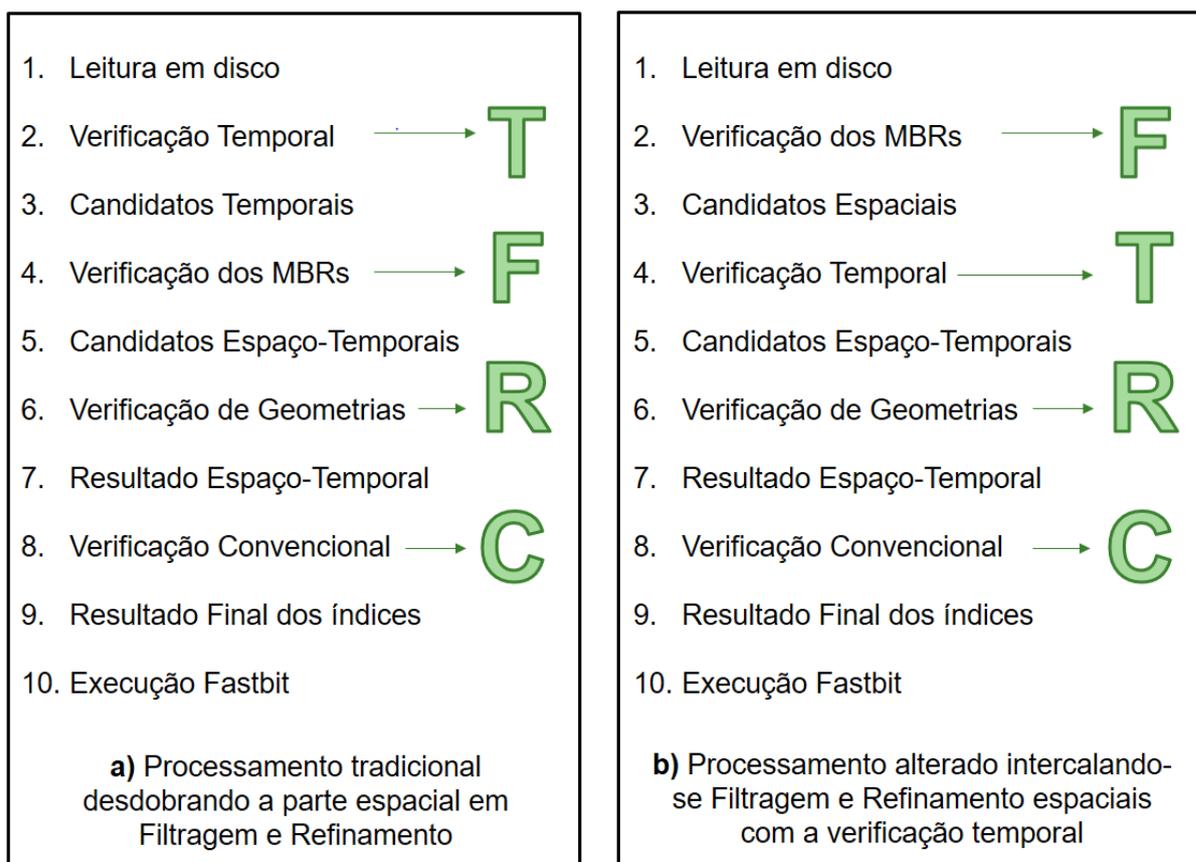


Figura 34: Alteração da execução dos índices Espaço-Temporais para contemplar a poda temporal entre as fases de filtragem e refinamento espaciais - adaptado de Siqueira (2012).

Isto posto, a classe de processamento “**ET Intercalado (FTRC)**” compreende a inversão das fases F e T das técnicas de processamento incluídas na classe “Pipeline TE-C” (que são baseadas nos índices STB-index – “TEstb” – e HSTB-index – “TEhstb”) que são representadas na Figura 35 e Figura 36.

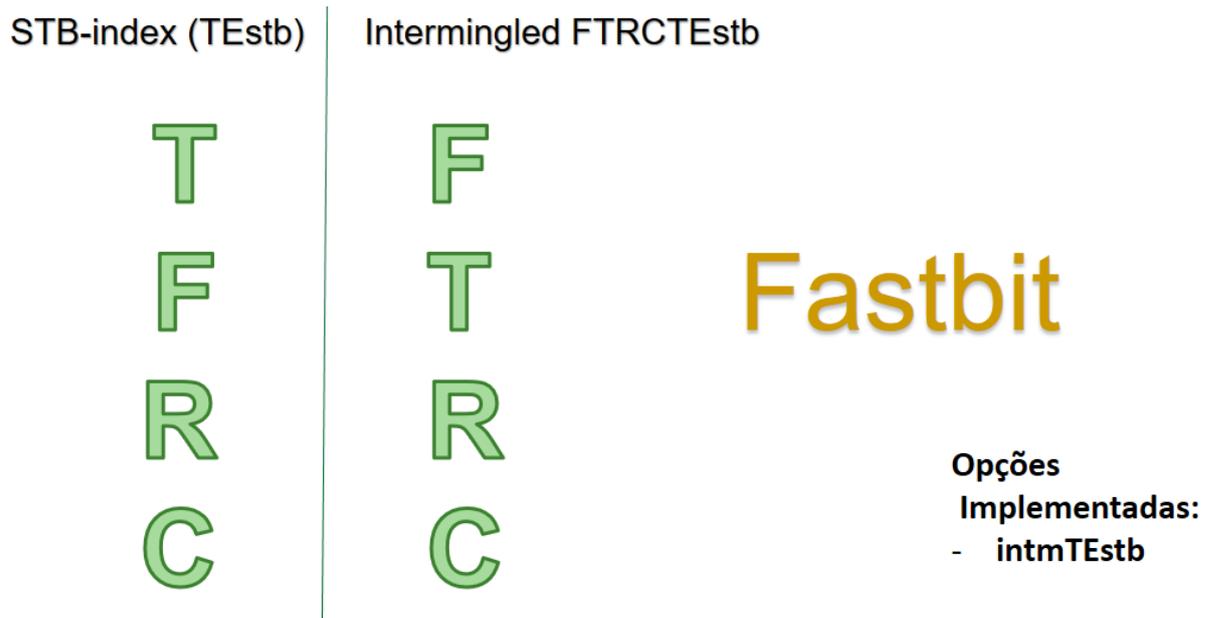


Figura 35: Representação da opção intercalada baseada no STB-index (pipeTEs

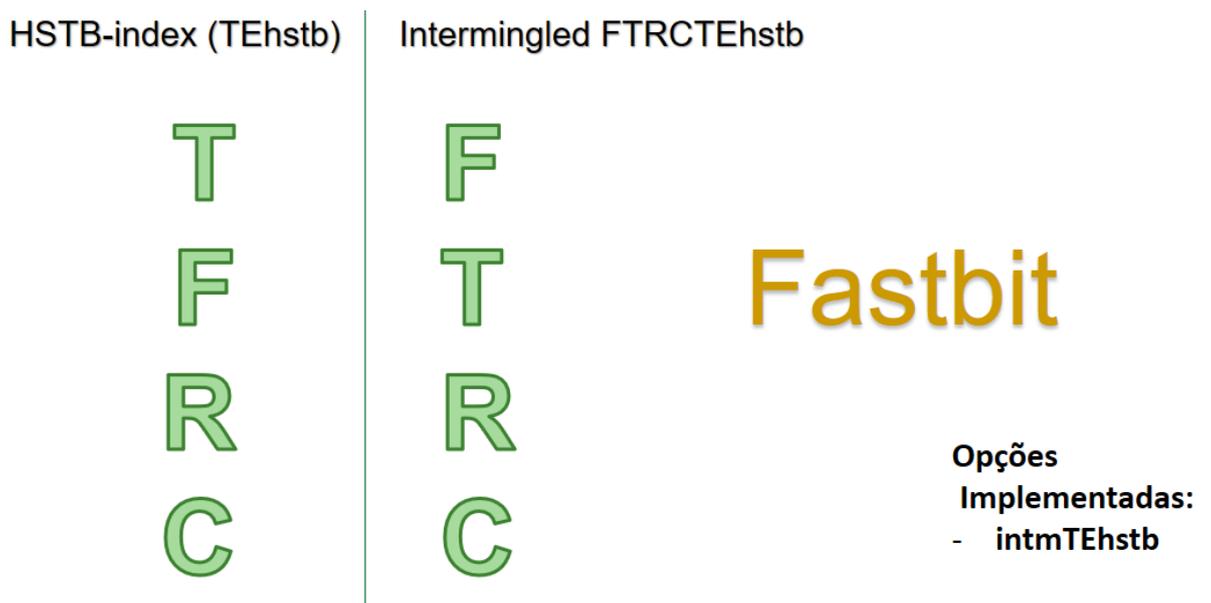


Figura 36: Representação da opção intercalada baseada no HSTB-index (pipeTEhstb)

5.4.4 Resumo das Configurações para o Processamento dos Predicados Espacial, Temporal e Convencional

No total, portanto, serão investigadas nesta pesquisa de mestrado 18 opções para o processamento dos predicados convencional, espacial e temporal de uma consulta STOLAP em um data warehouse representado segundo um esquema estrela, sendo: (i) 4 configurações usando o pipeline Temporal-Espacial-

Convencional (classe “Pipeline T-E-C”); (ii) 4 configurações usando o pipeline Espacial-Temporal-Convencional (classe “Pipeline E-T-C”); (iii) 2 configurações usando o pipeline Espaço-Temporal-Convencional (classe “Pipeline ET-C”); iv) 4 configurações usando o processamento independente de predicados com o uso de listas de IDs (classe “Independente T/E/C”); v) 2 configurações usando o processamento independente de predicados Espaço-Temporais e Convencionais com o uso de lista de IDs (classe “Independente TE/C”); e (vi) 2 configurações específicas para o pipeline F-Temporal-R-Convencional (classe “ET Intercalado FTRC”). A Figura 37 resume as opções, suas classes e cores a serem utilizadas nas análises do Capítulo 6 - .Todas as configurações serão testadas contra o *baseline* de visão materializada.

Classe	Opção de Processamento
Pipeline T-E-C	pipeTpEhsbCf
Pipeline T-E-C	pipeTfEhsbCf
Pipeline T-E-C	pipeTpEsbCf
Pipeline T-E-C	pipeTfEsbCf
Pipeline E-T-C	pipeEhsbTepCf
Pipeline E-T-C	pipeEsbTepCf
Pipeline E-T-C	pipeEhsbTefCf
Pipeline E-T-C	pipeEsbTefCf
Pipeline TE-C	pipeTEstb
Pipeline TE-C	pipeTEhstb
Independente T/E/C	indpTpEhsbCf
Independente T/E/C	indpTfEhsbCf
Independente T/E/C	indpTpEsbCf
Independente T/E/C	indpTfEsbCf
Independente TE/C	indpTEstb
Independente TE/C	indpTEhstb
ET Intercalado (FTRC)	intmTEstb
ET Intercalado (FTRC)	intmTEhstb

Cores usadas

- Pipeline T-E-C
- Pipeline E-T-C
- Pipeline TE-C
- Independente T/E/C
- Independente TE/C
- ET Intercalado (FTRC)

Figura 37: Resumo das opções de processamento implementadas, suas classes e as respectivas cores utilizadas nas análises

5.4.5 Considerações Finais

Uma vez apresentadas e definidas todas as partes necessárias para demonstrar como os testes de desempenho foram realizados, será apresentada no próximo capítulo a análise dos dados coletados e as suas conclusões.

Capítulo 6

ANÁLISE DOS DADOS

Uma vez definidos o escopo do trabalho, os dados e as técnicas a serem utilizadas para o processamento dos predicados espacial, temporal e convencional, os experimentos puderam ser executados e os tempos de processamento coletados. Como próximo passo desta pesquisa, este capítulo irá tratar da análise e interpretação dos resultados de desempenho coletados nos experimentos.

A análise dos dados será feita de três maneiras: i) análise individual de cada opção de processamento para cada uma das configurações de seletividade espaço-temporal e tipo de dado espacial; ii) análise da influência da seletividade espaço-temporal para cada tipo de dado espacial; iii) análise da influência do tipo de dado espacial para cada seletividade espaço-temporal.

6.1 Análise das Opções de Processamento em Função da Seletividade Espaço-Temporal e do Tipo de Dado Espacial

Esta seção trata da análise individual de cada opção de processamento para cada uma das 16 configurações resultantes da combinação das seletividades espaço-temporais (i.e. muito baixa, baixa, média e alta) e dos tipos de dados espaciais (i.e., pontos de clientes, pontos de fornecedores, linhas de ruas e polígonos de cidades) e que estão sumarizadas na Tabela 3. Para facilitar o entendimento dos dados, recomenda-se ter em mãos a Figura 28 que contém a codificação utilizada para cada opção de processamento.

6.1.1 Objetivo da Análise

O objetivo das análises a seguir é identificar comportamentos e padrões dentro das opções de processamento e de suas respectivas classes quando

comparadas contra o *baseline* de visão materializada e contra a opção de processamento que apresentou melhor desempenho. Esta identificação dará subsídios para se refutar ou confirmar as hipóteses formuladas na Seção 5.2.

6.1.2 Análise da Configuração S0-caddress

Os dados de análise da configuração utilizando uma seletividade muito baixa (S0) para pontos originários da tabela de clientes (caddress) podem ser observados na Tabela 7. Nela vemos que a opção de processamento vencedora foi a pipeTEhstb com melhoria de apenas 2,65% no desempenho com relação ao *baseline* de visão materializada, o qual destaca-se aqui por ter ficado em segundo lugar sendo este o melhor resultado do *baseline* para este tipo de dado espacial. Outra opção que merece destaque é a intmTEhstb, com 7,71% de perda para o *baseline* de visão materializada e apenas 10,15% de perda para a opção vencedora. Já a quarta opção (pipeEhsbTepCf) apresenta um salto de perda de desempenho e cai para 31,98% de perda com relação ao *baseline* de visão materializada e 33,78% para a opção vencedora. Desta forma, somente estas três opções de processamento alcançaram um desempenho próximo (i.e., menor que 15%) ao desempenho obtido pela opção vencedora.

O pior desempenho ficou com a opção indpTfEhsbCf cujas perdas ficaram em torno de 98,03% sobre o *baseline* de visão materializada e de 98,09% sobre a opção vencedora. Pode-se também notar de acordo com os resultados mostrados na Tabela 7 que o restante das opções de processamento obteve perdas expressivas (i.e. acima de 40%) quando comparadas ao *baseline* de visão materializada. Tal agrupamento começa com a opção de processamento intmTEstb (54,16%) na quinta colocação da tabela e termina com a última opção de processamento (indpTfEsbCf) com 99,07% de perda.

Quando se observa o agrupamento das opções de processamento em suas respectivas classes de processamento, nota-se a seguinte ordem decrescente de desempenho (i.e. melhor para pior desempenho): um das opções de processamento da classe *Pipeline* TE-C, *baseline* de visão materializada, ET Intercalado (FTRC) com uma das opções da classe *Pipeline* E-T-C entre elas, a segunda opção de processamento da classe *Pipeline* TE-C, *Pipeline* E-T-C e *Pipeline* T-E-C se alternando nas posições intermediárias e, mais para o final da tabela *Pipeline* T-E-C

alternando com as opções da classe Independente TE/C e, por último, a classe Independente T/E/C. Mais especificamente, todas as opções que tiveram um desempenho próximo (ganho ou perda de até 40%) sobre o *baseline* de visão materializada são opções que tentam utilizar o processamento em *pipeline*. Por outro lado, opções baseadas em processamento independente necessitam processar de forma duplicada o predicado convencional, ocasionando um alto custo adicional. Desta forma, opções baseadas nas classes de processamento independente, proporcionaram pior desempenho quando comparado ao *baseline* de visão materializada.

Tabela 7: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora – S0-caddress

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/perda sobre baseline	Porc perda para o vencedor
pipeTEhstb	Pipeline TE-C	HSTB	200052454,92	0,2001	2,646	0,000
VMA	VM indexada	BTree	205488680,00	0,2055	0,000	2,646
intmTEhstb	ET Intercalado (FTRC)	HSTB	222645614,52	0,2227	-7,706	10,148
pipeEhsbTepCf	Pipeline E-T-C	HSB	302099907,28	0,3021	-31,980	33,779
intmTEstb	ET Intercalado (FTRC)	STB	448307245,20	0,4483	-54,163	55,376
pipeTEstb	Pipeline TE-C	STB	451803873,16	0,4518	-54,518	55,721
pipeEsbTepCf	Pipeline E-T-C	SB	512108425,68	0,5121	-59,874	60,936
pipeTpEsbCf	Pipeline T-E-C	SB	3419625560,76	3,4196	-93,991	94,150
pipeEhsbTefCf	Pipeline E-T-C	HSB	5165417140,56	5,1654	-96,022	96,127
pipeTfEhsbCf	Pipeline T-E-C	HSB	5339445534,80	5,3395	-96,152	96,253
indpTEhstb	Independente TE/C	HSTB	5559320979,36	5,5593	-96,304	96,401
pipeTfEsbCf	Pipeline T-E-C	SB	5568135440,36	5,5681	-96,310	96,407
indpTEstb	Independente TE/C	STB	5781776860,52	5,7818	-96,446	96,540
pipeTpEhsbCf	Pipeline T-E-C	HSB	6077794914,00	6,0778	-96,619	96,708
pipeEsbTefCf	Pipeline E-T-C	SB	6865567344,44	6,8656	-97,007	97,086
indpTpEsbCf	Independente T/E/C	SB	7107168537,64	7,1072	-97,109	97,185
indpTpEhsbCf	Independente T/E/C	HSB	7810912956,08	7,8109	-97,369	97,439
indpTfEsbCf	Independente T/E/C	SB	9868703426,64	9,8687	-97,918	97,973
indpTfEhsbCf	Independente T/E/C	HSB	10452732645,32	10,4527	-98,034	98,086

Cores usadas

E-T Intercalado
 Pipeline TE-C
 Pipeline T-E-C
 Pipeline E-T-C
 Independente T/E/C
 Independente TE/C



STB-index
 HSTB-index
 SB-index
 HSB-index



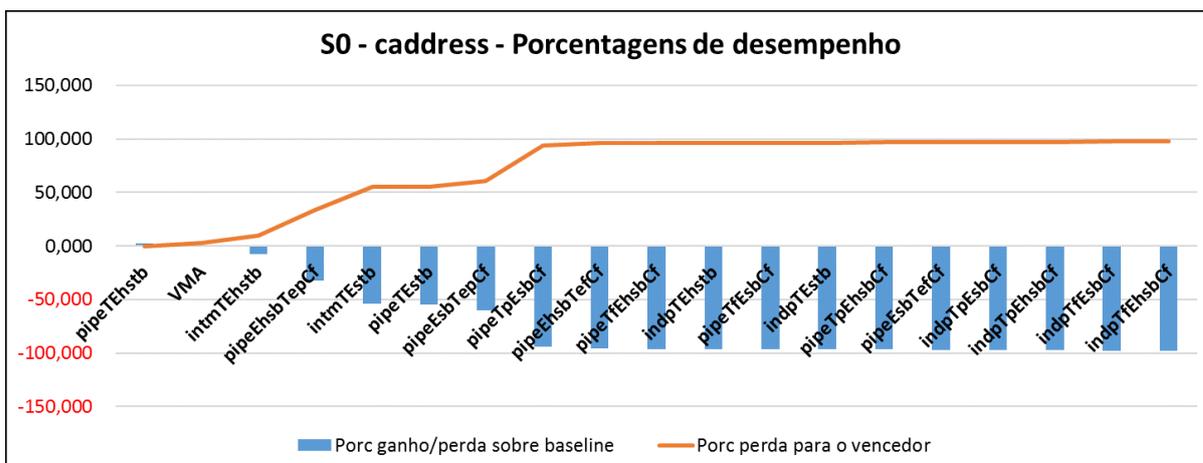


Figura 38: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S0-address

Quanto à maneira de limitação do predicado temporal, pode-se realizar a comparação entre as opções de processamento que utilizam a técnica Tp (consulta ao PostgreSQL somente da parte temporal) em relação à técnica Tf (consulta ao *software* de indexação bitmap somente da parte temporal) e a técnica Tep (consulta ao PostgreSQL com limitação de candidatos após a avaliação espacial) em relação à técnica Tef (consulta ao *software* de indexação bitmap com limitação de candidatos após a avaliação espacial). Para as opções de processamento em *pipeline*, três das opções de processamento baseadas na técnica Tp e Tep (pipeTpEsbCf, pipeEsbTepCf e pipeEhsbTepCf) levaram vantagem sobre suas correspondentes baseadas na técnica Tf e Tef (pipeTfEsbCf, pipeEsbTefCf e pipeEhsbTefCf). A exceção ficou por conta da opção pipeTfEhsbCf que teve um desempenho um pouco melhor que a opção pipeTpEhsbCf. A mesma comparação quando feita entre as opções de processamento da classe Independente T/E/C mostra que as opções baseadas na técnica Tp (indpTpEsbCf e indpTpEhsbCf) obtiveram melhor desempenho que suas respectivas opções baseadas na técnica Tf (indpTfEsbCf e indpTfEhsbCf). Os resultados acima se explicam pela consulta direta ao banco de dados com processamento em memória das técnicas Tp e Tep estar prevalecendo em sua maioria sobre a execução das técnicas Tf e Tef que envolvem a leitura do arquivo texto de resposta para carrega-las em uma estrutura de lista na linguagem de programação alvo.

Comparando-se as opções de processamento mais próximas da opção de processamento vencedora, tem-se uma predominância das opções de processamento baseadas no índice HSTB-index (pipeTEhstb e intmTEhstb) sobre suas relativas baseadas no STB-index (pipeTEstb e intmTEstb). Essas opções de processamento tiveram a interferência da opção de processamento pipeEhsbTepCf, devido à maneira de preparação da seletividade dos dados onde foi fixada a parte temporal com seletividade alta (31,26%) e variada a parte espacial para atingir a seletividade esperada (neste caso, muito baixa de 0,002%) e ao fato desta fazer a limitação espacial a priori facilitando a eliminação de candidatos a cada predicado processado. Destaca-se também a força da indexação em árvore para seletividades muito baixas que fez não só as opções baseadas nos índices HSTB-index, mas também o próprio *baseline* de visão materializada (com índices baseados em árvores binárias), ficarem com os melhores desempenhos.

Observa-se também que os resultados encontrados por Siqueira et. al. (2012) ao comparar o desempenho do SB-index contra a sua versão hierarquizada (o HSB-index) se manteve nesta seletividade muito baixa de registros, ou seja, dentre opções de processamento similares, as opções de processamento que usaram o HSB-index como índice base tiveram um melhor desempenho que as opções de processamento que usaram o SB-index. Um exemplo é a opção de processamento pipeTpEhsbCf que utilizou como índice base o HSB-index e teve melhor desempenho que a opção de processamento equivalente utilizando o SB-index como base, a pipeTpEsbCf. A única exceção à regra acima foi a opção pipeTpEsbCf (baseada no SB-index) que obteve melhor desempenho que a opção pipeTpEhsbCf (baseada no HSB-index). Tal observação ainda se verificou quando os índices tiveram a parte temporal incluída na estrutura dos nós, como é o caso das opções baseadas no STB-index (intmTEstb e pipeTEstb) e das baseadas no HSTB-index (intmTEhstb e pipeTEhstb). Desta forma, a opção baseada no HSTB-index obteve um melhor desempenho que a opção baseada no STB-index, nesta seletividade muito baixa.

6.1.3 Análise da Configuração S0-address

Para esta configuração foi utilizada uma seletividade espaço-temporal muito baixa (S0) tendo o tipo de dado ponto indicando os endereços de fornecedores

(caddress) como fonte. De acordo com a Tabela 8, podemos observar que a opção de processamento vencedora foi a *intmTEhstb* com 31,30% de melhoria no desempenho em relação ao *baseline* de visão materializada. Bem próxima em segundo lugar, ficou a opção *pipeTEhstb* que obteve 31,07% de melhoria sobre o *baseline* de visão materializada e apenas 0,33% de perda em comparação com a opção vencedora. Dentre as demais opções de processamento, somente a opção *pipeEhsbTepCf*, baseada no *Pipeline* E-T-C, conseguiu um desempenho próximo à opção vencedora (i.e., perda de até 15%), com perda de desempenho de 8,09%.

As demais opções tiveram perda de desempenho com relação à opção vencedora variando entre 22,97% a 93,15%. A pior opção foi a *indpTfEsbCf* que apresentou 90,04% de perda para o *baseline* de visão materializada e 93,15% de perda para a opção vencedora. Várias configurações tiveram melhoria de desempenho contra o *baseline* de visão materializada, a saber: *pipeTEstb* (10,80%), *intmTEstb* (10,70%), *pipeEsbTepCf* (9,73%) e *pipeTpEsbCf* (1,65%). Dentre as opções que tiveram perda de desempenho em relação ao *baseline* de visão materializada, somente a opção *pipeTpEhsbCf* ficou mais próxima com 16,17% de perda com o restante das opções obtendo perdas de desempenho acima de 80%.

O agrupamento das opções de processamento em suas respectivas classes quando ordenadas do melhor para o pior desempenho sobre o *baseline* de visão materializada seguiu a seguinte ordem: ET Intercalado (FTRC) com opções das classes *Pipeline* TE-C e *Pipeline* E-T-C entre elas, *Pipeline* E-T-C e *Pipeline* T-E-C mescladas e com o *baseline* de visão materializada entre eles, Independente T/E/C e Independente TE/C intercalados do meio para o final da tabela. Da quarta à décima segunda posição as classes estão misturadas quase que duas a duas e todas as classes que utilizam processamento em *Pipeline* ou Independente possuem opções de processamento com desempenho inferior ao *baseline* de visão materializada. Tal comportamento adiciona importância às opções que utilizam processamento em *pipeline* e que possuem os dados para limitação espaço-temporal na estrutura do índice (opções de processamento baseadas no STB-index ou HSTB-index). Desafiando a tendência temos a opção *pipeEhsbTepCf* que ficou em terceiro lugar, mas tal fato é explicado pela maneira de construção das seletividades (neste caso fixando-se a seletividade temporal em 33,65% e variando-se a janela espacial até atingirmos a seletividade espaço-temporal desejada de 0,027%) o que tende a favorecer as opções de processamento que fazem a

limitação espacial a priori. Além do observado acima, há mais um fator que contribuiu para a vantagem obtida pela opção de processamento pipeEhsbTepCf que foi o número reduzido de registros na tabela “supplier” (7.375 contra 95.967 da tabela “customer”), o que facilitou a filtragem temporal no PostgreSQL e fez a opção ficar entre as melhores.

Tabela 8: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S0-address

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/per da sobre baseline	Porc perda para o vencedor
intmTEhstb	ET Intercalado (FTRC)	HSTB	745490963,56	0,7455	31,297	0,000
pipeTEhstb	Pipeline TE-C	HSTB	747946096,16	0,7480	31,070	0,328
pipeEhsbTepCf	Pipeline E-T-C	HSB	811098568,00	0,8111	25,250	8,089
pipeTEstb	Pipeline TE-C	STB	967838115,68	0,9678	10,805	22,974
intmTEstb	ET Intercalado (FTRC)	STB	968929233,40	0,9689	10,705	23,060
pipeEsbTepCf	Pipeline E-T-C	SB	979475486,68	0,9795	9,733	23,889
pipeTpEsbCf	Pipeline T-E-C	SB	1067165372,36	1,0672	1,652	30,143
VMB	VM básica	N/A	1085086040,00	1,0851	0,000	31,297
pipeTpEhsbCf	Pipeline T-E-C	HSB	1294419541,28	1,2944	-16,172	42,407
pipeEhsbTefCf	Pipeline E-T-C	HSB	5485418640,48	5,4854	-80,219	86,410
indpTpEhsbCf	Independente T/E/C	HSB	5582051215,48	5,5821	-80,561	86,645
indpTEhstb	Independente TE/C	HSTB	5615261227,92	5,6153	-80,676	86,724
pipeTfEhsbCf	Pipeline T-E-C	HSB	5682582737,76	5,6826	-80,905	86,881
pipeTfEsbCf	Pipeline T-E-C	SB	6036724578,76	6,0367	-82,025	87,651
indpTpEsbCf	Independente T/E/C	SB	6172040490,08	6,1720	-82,419	87,921
indpTEstb	Independente TE/C	STB	6226219475,76	6,2262	-82,572	88,027
pipeEsbTefCf	Pipeline E-T-C	SB	8953304715,08	8,9533	-87,881	91,674
indpTfEhsbCf	Independente T/E/C	HSB	9937679363,36	9,9377	-89,081	92,498
indpTfEsbCf	Independente T/E/C	SB	10891054279,72	10,8911	-90,037	93,155

Cores usadas

- ET Intercalado (FTRC)
- Pipeline TE-C
- Pipeline T-E-C
- Pipeline E-T-C
- Independente T/E/C
- Independente TE/C



- STB-index
- HSTB-index
- SB-index
- HSB-index



Outro comportamento a ser notado é a maneira de limitação do predicado temporal entre as opções de processamento que utilizam a técnica Tp (consulta ao PostgreSQL somente da parte temporal) em relação à técnica Tf (consulta ao software de indexação bitmap somente da parte temporal) e a técnica Tep (consulta ao PostgreSQL com limitação de candidatos após a avaliação espacial) em relação à técnica Tef (consulta ao software de indexação bitmap com limitação de

candidatos após a avaliação espacial). Para as opções de processamento em *pipeline*, aquelas baseadas na técnica Tp e Tep (pipeTpEsbCf, pipeTpEhsbCf, pipeEsbTepCf e pipeEhsbTepCf) levaram vantagem sobre suas correspondentes baseadas na técnica Tf e Tef (pipeTfEsbCf, pipeTfEhsbCf, pipeEsbTefCf e pipeEhsbTefCf). O mesmo comportamento é encontrado quando comparamos as opções de processamento da classe Independente T/E/C onde as opções baseadas na técnica Tp (indpTpEsbCf e indpTpEhsbCf) obtiveram melhor desempenho que suas respectivas opções baseadas na técnica Tf (indpTfEsbCf e indpTfEhsbCf). Isso se explica pela consulta direta ao banco de dados com processamento em memória das técnicas Tp e Tep estarem prevalecendo sobre a execução das técnicas Tf e Tef que envolvem a leitura do arquivo texto de resposta para carrega-las em uma estrutura de lista na linguagem de programação alvo.

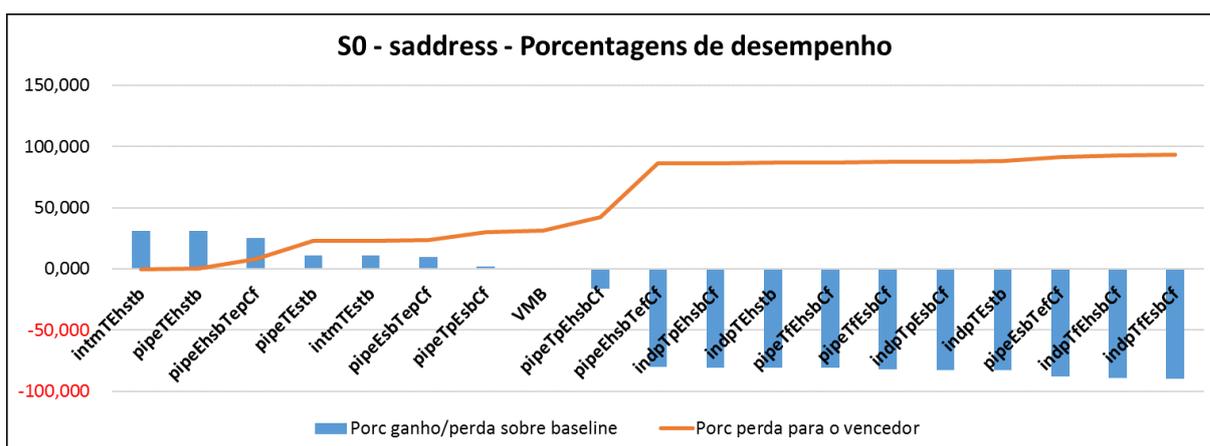


Figura 39: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S0-saddress

A tendência de superação de desempenho das opções baseadas no HSB-index sobre as baseadas no SB-index também se verificou nesta configuração. A própria opção pipeEhsbTepCf citada acima (que ficou em terceiro lugar) é um exemplo quando comparada com a sua opção equivalente utilizando o SB-index como índice base (pipeEsbTepCf). Exceção a essa regra, a opção pipeTpEsbCf obteve melhor desempenho que a sua relativa baseada no HSB-index (a pipeTpEhsbCf).

A comparação das opções vencedoras vem nos mostrar novamente a força das opções de processamento utilizando como base índices hierárquicos (HSTB-index e HSB-index) contra as suas versões utilizando índices baseados em vetor

(STB-index e SB-index) nesta seletividade muito baixa. Nota-se que as opções de processamento *intmTEhstb*, *pipeTEhstb* e *pipeEhstbTepCf* obtiveram melhores desempenhos que suas equivalentes (*intmTEstb*, *pipeTEstb* e *pipeEsbTepCf*) e pode-se ainda observar uma certa distância entre elas onde a porcentagem de perda para a opção vencedora foi de 8,09% da terceira para 22,97% da quarta colocada.

6.1.4 Análise da Configuração S0-street”

Como pode ser observado na Tabela 9, a opção de processamento vencedora para a granularidade muito baixa (S0) com linhas vindas da tabela “street” (*street*) foi o *baseline* de visão materializada na sua versão avançada (altamente indexada) superando a segunda opção de processamento, a *pipeEhstbTepCf*, com 73,57% de diferença no desempenho. Tal superioridade se explica pela correta utilização dos índices por parte do PostgreSQL e da simplicidade do objeto espacial envolvido, uma vez que as ruas da tabela “street” foram construídas utilizando-se linhas horizontais e verticais as quais não geram o fenômeno conhecido como “dead space”, o que incluiria cálculos adicionais para a retirada de falsos candidatos. Da terceira à sexta posição temos as opções de processamento *intmTEhstb*, *pipeTEhstb*, *intmTEstb* e *pipeTEstb* que estão muito próximas entre si com menos de 6% de diferença de desempenho (73,62% a 79,87% de perda para a opção de processamento vencedora).

O pior desempenho ficou com a opção *pipeEsbTefCf* cujas perdas ficaram em torno de 98,32% sobre o *baseline* de visão materializada. Com a melhoria provida ao *baseline* de visão materializada e o seu correto processamento usando os índices da visão materializada, todas as opções de processamento foram ultrapassadas em mais de 40% pelo seu desempenho. Outra diferença que afetou negativamente os tempos das outras opções de processamento foi a necessidade da inclusão de uma fase de refinamento dos dados do tipo linha por se tratar de uma consulta do tipo IRQ (Intersection Range Query) a qual pode introduzir falsos candidatos a serem retirados em uma etapa posterior.

Tabela 9: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S0-street

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/perda sobre baseline	Porc perda para o vencedor
VMA	VM indexada	BTree	225716800,00	0,2257	0,000	0,000
pipeEhsbTepCf	Pipeline E-T-C	HSB	854120340,96	0,8541	-73,573	73,573
intmTEhstb	ET Intercalado (FTRC)	HSTB	855668615,12	0,8557	-73,621	73,621
pipeTEhstb	Pipeline TE-C	HSTB	857329807,40	0,8573	-73,672	73,672
intmTEstb	ET Intercalado (FTRC)	STB	1107614684,44	1,1076	-79,621	79,621
pipeTEstb	Pipeline TE-C	STB	1121552217,08	1,1216	-79,875	79,875
pipeEsbTepCf	Pipeline E-T-C	SB	1310374405,32	1,3104	-82,775	82,775
pipeTpEhsbCf	Pipeline T-E-C	HSB	3042693076,36	3,0427	-92,582	92,582
pipeTpEsbCf	Pipeline T-E-C	SB	3674976922,36	3,6750	-93,858	93,858
pipeTfEsbCf	Pipeline T-E-C	SB	5449827320,28	5,4498	-95,858	95,858
indpTEhstb	Independente TE/C	HSTB	5750761170,48	5,7508	-96,075	96,075
pipeEhsbTefCf	Pipeline E-T-C	HSB	5784547883,40	5,7846	-96,098	96,098
indpTEstb	Independente TE/C	STB	5994092382,04	5,9941	-96,234	96,234
pipeTfEhsbCf	Pipeline T-E-C	HSB	5996075366,56	5,9961	-96,236	96,236
indpTpEhsbCf	Independente T/E/C	HSB	7792578568,84	7,7926	-97,103	97,103
indpTpEsbCf	Independente T/E/C	SB	8542786030,88	8,5428	-97,358	97,358
indpTfEhsbCf	Independente T/E/C	HSB	10283733451,32	10,2837	-97,805	97,805
indpTfEsbCf	Independente T/E/C	SB	11119158716,24	11,1192	-97,970	97,970
pipeEsbTefCf	Pipeline E-T-C	SB	13459700189,72	13,4597	-98,323	98,323

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

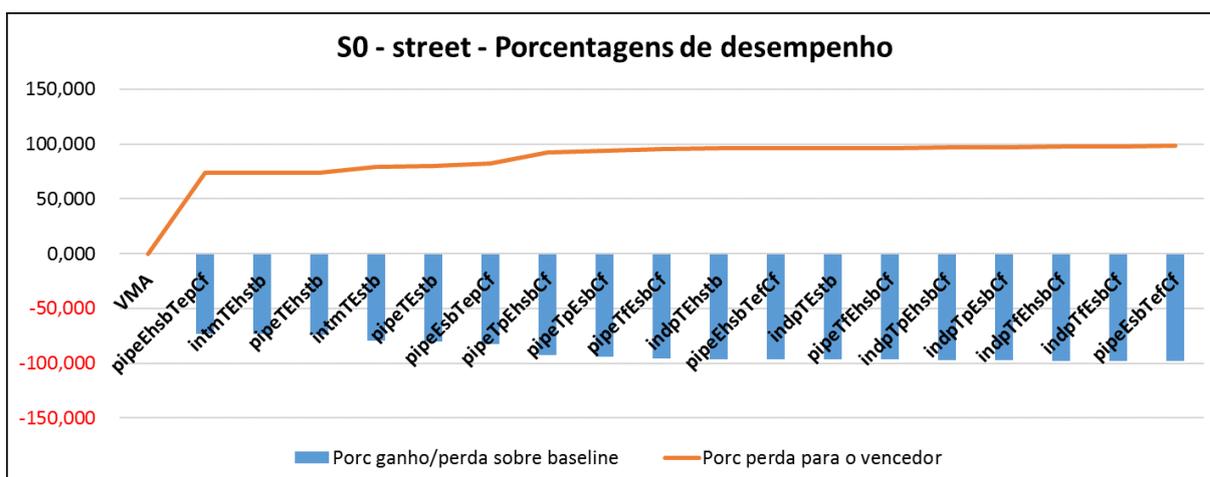


Figura 40: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S0-street

O agrupamento das opções de processamento nas suas respectivas classes resultou na seguinte ordem (do maior para o menor desempenho): *Baseline* de visão materializada, uma das opções de processamento da classe *Pipeline* E-T-C, ET Intercalado (FTRC) e *Pipeline* TE-C alternando-se nas posições, uma segunda opção da classe *Pipeline* E-T-C seguida da maioria das opções de processamento da classe *Pipeline* T-E-C, Independente TE/C e Independente T/E/C. A presença do *baseline* de visão materializada como opção de processamento vencedora demonstra que, para esta opção espaço-temporal, tanto o processamento em *pipeline* quanto a ajuda extra dada pela inclusão da informação espaço-temporal na estrutura dos índices não foram fatores decisivos para que as opções de melhor desempenho pudessem superar também o *baseline* de visão materializada. Desta forma, a proposta de índices espaço-temporais para DW, como feito na proposta do STB-index no trabalho de Tsuruda (2013) e na adaptação da HSTB-index realizada pelo autor desta pesquisa com base na implementação da HSB-index proposta por Siqueira et. al. (2012), ainda não foram suficientes para rivalizar o desempenho desse tipo de visão materializada, mas foram importantes para distanciar as opções de processamento que as utilizam das demais opções.

Também pode-se observar que o processamento extra do predicado convencional requerido pelas opções independentes as colocam nas últimas posições da Tabela 9 com pronunciada separação. Entre as opções de processamento independentes também se observa que as opções de processamento que utilizaram como índice base o STB-index e o HSTB-index (cuja informação espaço-temporal está incluída na estrutura do índice) levaram vantagem sobre as baseadas no SB-index e HSB-index, cuja limitação temporal é feita em separado utilizando-se consultas SQL ao PostgreSQL ou processamento de índices bitmap.

Na comparação entre as técnicas de processamento puramente temporais (T_p contra T_f e T_{ep} contra T_{ef}), notamos um comportamento sem exceções onde as opções de processamento baseadas nas técnicas T_p e T_{ep} superam as suas opções correspondentes utilizando as técnicas T_f e T_{ef} . Um exemplo disso é a opção $pipeEhsbTepCf$ que superou a opção $pipeEhsbTefCf$.

Quando comparamos as cinco melhores opções de processamento depois do *baseline* de visão materializada, observamos que as três primeiras (i.e., do segundo ao quarto lugares) utilizam como índice base o HSB-index e o HSTB-index e que as

duas últimas (quinto e sexto lugares) utilizam o STB-index como base. Além disso, a própria indexação do *baseline* de visão materializada utiliza índices hierárquicos (BTree e GiST) mostrando que a poda de registros das opções hierárquicas nas seletividades mais baixas é uma forte aliada para agilizar o processamento das consultas.

O comportamento das opções de processamento em relação à utilização dos índices SB-index e HSB-index como base se manteve dentro do esperado e no mesmo padrão das opções espaço-temporais baseadas em ponto (S0-caddress e S0-saddress) onde as opções que usaram o SB-index como índice base tiveram desempenho pior quando comparados à sua opção de processamento correspondente utilizando o HSB-index como base. A única exceção foi a opção de processamento pipeTfEsbCf que obteve melhor desempenho frente à sua relativa utilizando o HSB-index como base (a pipeTfEhsbCf).

6.1.5 Análise da Configuração S0-city

Como pode ser observado na Tabela 10, a opção de processamento vencedora para a granularidade muito baixa (S0) com polígonos vindos da tabela de cidades (city) foi o *baseline* de visão materializada na sua versão avançada (VMA), com ganhos de desempenho de 65,64% para a opção de processamento intmTEhstb. Mais uma vez vemos o PostgreSQL utilizando os índices corretamente e temos objetos espaciais simples para a representação das cidades (retângulos), o que facilitou os cálculos pela ausência (ou grande diminuição) do chamado “*dead space*”. A opção de processamento que ficou em segundo lugar foi seguida muito perto pelas outras opções até a nona posição com menos de 10% de diferença de desempenho entre elas. Outra diferença para as outras análises feitas até agora é a presença de mais duas opções de processamento entre as melhores colocadas (a pipeTfEhsbCf e pipeTpEhsbCf) que utilizam um índice hierárquico como base (o HSB-index, no caso). O pior desempenho ficou com a opção de processamento pipeEsbTefCf cujas perdas ficaram em torno de 98,05% sobre o *baseline* de visão materializada que também foi a opção de processamento vencedora.

Tabela 10: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora – S0-city

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/perda sobre baseline	Porc perda para o vencedor
VMA	VM indexada	BTree	259604120,00	0,2596	0,000	0,000
intmTEhstb	ET Intercalado (FTRC)	HSTB	755617983,48	0,7556	-65,643	65,643
pipeTEhstb	Pipeline TE-C	HSTB	778520059,96	0,7785	-66,654	66,654
pipeTfEhscf	Pipeline T-E-C	HSB	802413661,68	0,8024	-67,647	67,647
pipeTpEhscf	Pipeline T-E-C	HSB	817998855,32	0,8180	-68,264	68,264
pipeTEstb	Pipeline TE-C	STB	959104748,32	0,9591	-72,933	72,933
intmTEstb	ET Intercalado (FTRC)	STB	979891015,20	0,9799	-73,507	73,507
pipeTpEsbCf	Pipeline T-E-C	SB	1028846015,16	1,0289	-74,767	74,767
pipeEhscf	Pipeline E-T-C	HSB	1046407624,24	1,0464	-75,191	75,191
pipeEsbTepCf	Pipeline E-T-C	SB	1452311665,24	1,4523	-82,125	82,125
pipeEhscf	Pipeline E-T-C	HSB	5068788097,36	5,0688	-94,878	94,878
indpTEstb	Independente TE/C	STB	5668636170,60	5,6686	-95,420	95,420
indpTpEhscf	Independente T/E/C	HSB	5703219767,16	5,7032	-95,448	95,448
indpTEhstb	Independente TE/C	HSTB	5850248214,80	5,8503	-95,563	95,563
indpTpEsbCf	Independente T/E/C	SB	5892577824,48	5,8926	-95,594	95,594
pipeTfEsbCf	Pipeline T-E-C	SB	6112052761,80	6,1121	-95,753	95,753
indpTfEhscf	Independente T/E/C	HSB	10135476461,88	10,1355	-97,439	97,439
indpTfEsbCf	Independente T/E/C	SB	10146940473,24	10,1469	-97,442	97,442
pipeEsbTefCf	Pipeline E-T-C	SB	13305317396,56	13,3053	-98,049	98,049

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

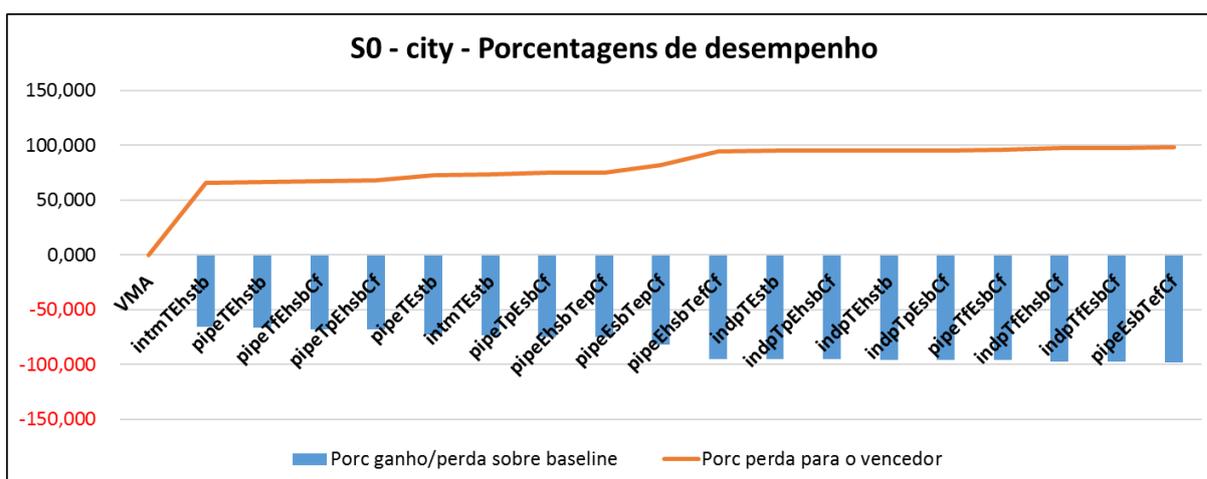


Figura 41: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S0-city

Quando se observa o agrupamento das opções de processamento nas suas respectivas classes, nota-se a seguinte ordem (do melhor para pior desempenho): *Baseline* de visão materializada, ET intercalado (FTRC), *Pipeline* TE-C e *Pipeline* T-E-C se revezando nas primeiras posições, *Pipeline* E-T-C, Independente TE/C e Independente T/E/C. Tal resultado indica que opções utilizando processamento em *pipeline* (onde a limitação de candidatos obtida numa etapa é utilizada na etapa posterior) foram as que mais se aproximaram do desempenho obtido pelo *baseline* de visão materializada. As opções independentes seguem com perdas significativas em relação ao *baseline* de visão materializada por conta da necessidade de duplicação de processamento convencional e da forma de construção das seletividades, onde um dos tipos de processamento sempre irá retornar uma grande quantidade de registros (neste caso a limitação temporal que foi fixada numa seletividade alta retornando 90,28% dos registros da tabela “city”) em um dos predicados de processamento.

Na comparação entre as técnicas de processamento puramente temporais (Tp contra Tf e Tep contra Tef), notamos o mesmo comportamento das opções baseadas em ponto e de mesma seletividade espaço-temporal (S0-caddress e S0-saddress), inclusive com a mesma exceção obtida pela opção de processamento pipeTfEhsbCf sobre sua relativa utilizando a técnica Tp (a pipeTpEhsbCf). As mesmas observações e conclusões se aplicam também a esta configuração espaço-temporal.

Observa-se pela comparação entre as opções de processamento que ficaram mais próximas do *baseline* de visão materializada o mesmo comportamento encontrado na configuração S0-street, onde as opções baseadas em índices hierárquicos (HSTB-index e HSB-index) se sobressaíram sobre suas correspondentes que utilizaram o STB-index ou o SB-index como base. A Tabela 11 detalha os tempos médios obtidos por cada opção em cada tipo de predicado da consulta STOLAP e, como esperado, a limitação espaço-temporal foi muito mais rápida para a opção intmTEhstb (baseada no HSTB-index) pela poda de candidatos. As outras opções de processamento variaram bastante, mas sempre com vantagem para as opções que possuíam ambas as informações para limitação espacial e temporal na estrutura do índice. A fase de Refinamento envolve uma consulta direta ao banco de dados espaço-temporal onde a única diferença encontrada é na API de acesso ao banco PostgreSQL cuja versão em C++ (das opções baseadas no STB-

index) ficou em média compatível com a versão em C (utilizada nas opções baseadas no HSTB-index). As diferenças no processamento do predicado convencional apresentaram uma maior variação e foram decisivas para indicar a opção de processamento vencedora. Vale ressaltar que é executada a mesma consulta final, no mesmo diretório de índices e utilizando-se o mesmo software de implementação dos índices bitmap (fastbit) em todas as opções e é o próprio software que realiza a criação do arquivo de saída com as respostas. Ainda assim podemos encontrar tais diferenças no tempo de processamento.

Tabela 11: Tempos médios em cada predicado e tempo total das opções vencedoras (S0-city)

Opção	Índice base	Limitação espaço-temporal (ns)	Refinamento (ns)	Limitação Convencional (ns)	Média Total (ns)
intmTEhstb	HSTB	165.129,84	122.136.108,40	633.316.745,24	755.617.983,48
pipeTEhstb	HSTB	13.798.205,12	117.406.839,24	647.315.015,60	778.520.059,96
pipeTfEhsbCf	HSB	89.629.277,48	119.055.858,72	593.728.525,48	802.413.661,68
pipeTpEhsbCf	HSB	62.855.505,84	126.897.205,80	628.246.143,68	817.998.855,32
pipeTEstb	STB	524.803,68	113.591.689,40	844.988.255,24	959.104.748,32
intmTEstb	STB	390.395,04	115.567.484,48	863.933.135,68	979.891.015,20
pipeTpEsbCf	SB	55.823.467,28	122.538.345,68	850.106.202,20	1.028.468.015,16

À exceção das opções de processamento discutidas acima, o comportamento das outras opções seguiu o que é observado por Siqueira et al. (2012). Para seletividades muito baixas, as opções cujo índice base é o HSB-index tiveram melhor desempenho que suas opções relativas utilizando o SB-index. Um exemplo disso é a opção de processamento pipeTpEhsbCf que teve um melhor desempenho que a sua relativa, a opção pipeTpEsbCf. Isso se deve à poda de candidatos proporcionada pela estrutura em árvore do HSB-index e que favorece a eliminação precoce de candidatos assim melhorando o tempo de processamento do índice. Tal situação também acontece com as opções de processamento baseadas no STB-index em relação às opções de processamento baseadas no HSTB-index pelos mesmos motivos apresentados anteriormente nas opções espaço-temporais baseadas em pontos (S0-caddress e S0-saddress). A exceção à regra ficou por conta da opção indpTEstb (baseada no STB-index) que teve um melhor desempenho frente à sua opção relativa utilizando o HSTB-index (a indpTEhstb).

6.1.6 Análise da Configuração S1-caddress

Como pode ser observado na Tabela 12, a opção vencedora para a granularidade baixa (S1) com pontos vindos da tabela de clientes (caddress) foi a pipeTEhstb, com 43,45% de melhoria no desempenho em relação ao *baseline* de visão materializada. A segunda melhor opção foi a intmTEhstb que obteve média de ganhos de desempenho de 37,46% sobre o *baseline* de visão materializada e de perda de 9,58% em relação à opção vencedora. Além desta última, apenas a opção de processamento pipeEhsbTepCf conseguiu um desempenho próximo (i.e. menor que 15%) à opção vencedora. O pior desempenho ficou com a opção indpTfEsbCf cuja perda foi de 93,93% sobre o *baseline* de visão materializada e 96,57% sobre a opção de processamento vencedora.

Mais opções se destacaram com ganhos de desempenho representativos (i.e. acima de 15%) sobre o *baseline* de visão materializada. São elas a intmTEstb (15,78%) e pipeTEstb (15,37%). A opção pipeEsbTepCf também apresentou ganho sobre o *baseline* de visão materializada, mas este foi de apenas 7,16% e, portanto, não se destacou frente a esta. Todas as outras opções foram ultrapassadas pelo *baseline* de visão materializada com um mínimo de 88,03% de perda de desempenho (opção pipeTpEsbCf) e máximo de 93,93% de perda de desempenho (opção indpTfEsbCf).

Observando-se o comportamento das opções de processamento através das suas classes, vê-se a seguinte ordenação do melhor para o pior desempenho: *Pipeline* TE-C, intercalado (FTRC) e *Pipeline* E-T-C se revezando nas primeiras posições, o *baseline* de visão materializada, *Pipeline* T-E-C e *Pipeline* E-T-C se revezando na parte intermediária da tabela com a Independente TE/C entre elas, e, finalmente, a classe Independente T/E/C. A característica comum às classes que obtiveram ganho sobre o *baseline* de visão materializada é o uso do resultado de uma das fases de processamento como limitação da fase posterior (*pipeline*). As opções independentes continuaram na parte de baixo da Tabela 12 devido ao processamento adicional demandado para o predicado convencional.

Já na comparação das técnicas de processamento temporal (Tp contra Tf e Tep contra Tef – descrita com mais detalhes na configuração S0-caddress), ainda se observa a maioria das opções de processamento utilizando a interface com o banco de dados (Tp e Tep) tendo vantagem sobre as opções de processamento utilizando

o software de indexação bitmap (Tf e Tef) e há exceção somente para a opção pipeTfEsbCf contra a sua opção de processamento relativa utilizando a técnica Tp (pipeTpEsbCf). Apesar das inversões de posição, as opções de processamento estão muito próximas (somente 0,14% de diferença de desempenho entre elas quando comparadas com a opção vencedora) e a confirmação de uma tendência virá somente se a diferença aumentar nas seletividades mais altas.

Tabela 12: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S1-caddress

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/per da sobre baseline	Porc perda para o vencedor
pipeTEhstb	Pipeline TE-C	HSTB	354261154,32	0,3543	43,452	0,000
intmTEhstb	ET Intercalado (FTRC)	HSTB	391803021,44	0,3918	37,459	9,582
pipeEhsbTepCf	Pipeline E-T-C	HSB	397519956,44	0,3975	36,547	10,882
intmTEstb	ET Intercalado (FTRC)	STB	527611192,68	0,5276	15,781	32,856
pipeTEstb	Pipeline TE-C	STB	530166771,68	0,5302	15,373	33,179
pipeEsbTepCf	Pipeline E-T-C	SB	581599033,04	0,5816	7,164	39,088
VMA	VM indexada	BTree	626477800,00	0,6265	0,000	43,452
pipeTpEsbCf	Pipeline T-E-C	SB	5234124219,12	5,2341	-88,031	93,232
pipeEhsbTefCf	Pipeline E-T-C	HSB	5271941313,96	5,2719	-88,117	93,280
pipeTfEsbCf	Pipeline T-E-C	SB	5346062898,24	5,3461	-88,282	93,373
indpTEhstb	Independente TE/C	HSTB	5398911785,20	5,3989	-88,396	93,438
pipeTfEhsbCf	Pipeline T-E-C	HSB	5477075443,40	5,4771	-88,562	93,532
indpTEstb	Independente TE/C	STB	5580189120,68	5,5802	-88,773	93,651
pipeTpEhsbCf	Pipeline T-E-C	HSB	5887513842,52	5,8875	-89,359	93,983
pipeEsbTefCf	Pipeline E-T-C	SB	7240905663,92	7,2409	-91,348	95,108
indpTpEhsbCf	Independente T/E/C	HSB	8687783363,84	8,6878	-92,789	95,922
indpTpEsbCf	Independente T/E/C	SB	10091867196,84	10,0919	-93,792	96,490
indpTfEhsbCf	Independente T/E/C	HSB	10291803698,36	10,2918	-93,913	96,558
indpTfEsbCf	Independente T/E/C	SB	10327353520,28	10,3274	-93,934	96,570

Cores usadas

ET Intercalado (FTRC)
 Pipeline TE-C
 Pipeline T-E-C
 Pipeline E-T-C
 Independente T/E/C
 Independente TE/C



STB-index
 HSTB-index
 SB-index
 HSB-index



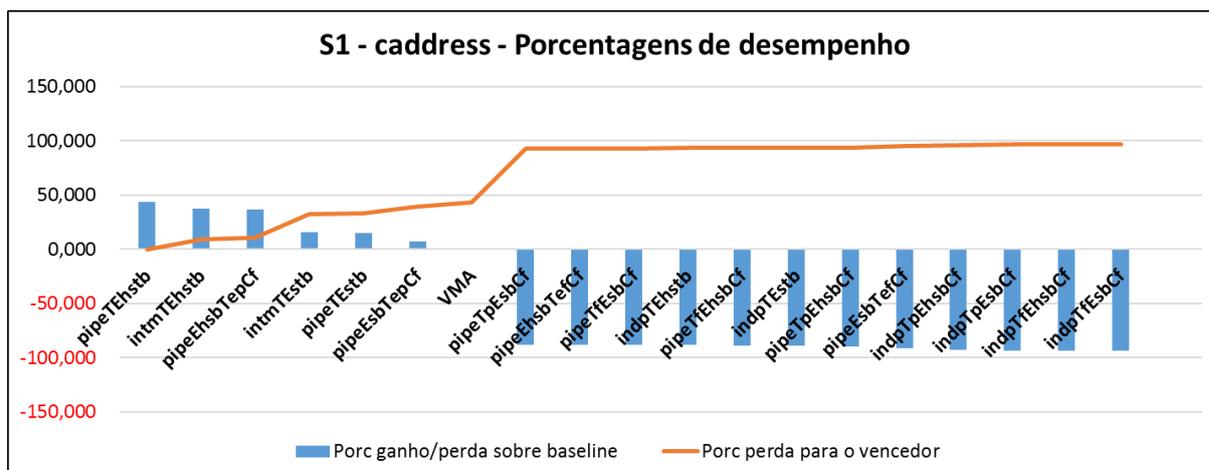


Figura 42: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S1-address

Ao comparar as opções de processamento nas melhores colocações, pode-se notar que as opções de processamento baseadas no HSTB-index e HSB-index tiveram melhor desempenho que suas correspondentes baseadas no STB-index e SB-index e que a opção pipeEhsbTepCf está presente entre as opções vencedoras devido aos mesmos motivos apresentados na comparação das opções vencedoras da opção espaço-temporal S0-caddress à diferença das seletividades envolvidas (68,74% para a seletividade temporal e 0,02% para a seletividade espaço-temporal).

Já na comparação dos índices base, vemos que os resultados das opções baseadas no HSB-index continuaram sendo melhores que as suas opções correspondentes tendo o SB-index como índice base somente para as classes *Pipeline* E-T-C e Independente T/E/C devido aos mesmos motivos apresentados na configuração S0-caddress. A classe *Pipeline* T-E-C foi a exceção com suas duas opções baseadas no índice SB-index (pipeTpEsbCf e pipeTfEsbCf) ganhando em desempenho das suas relativas utilizando o HSB-index como base (pipeTpEhsbCf e pipeTfEhsbCf) mas a diferença entre elas ainda não foi significativa para indicar uma tendência (a maior diferença entre elas foi de 0,75% quando considerada a porcentagem de perda de desempenho destas à opção de processamento vencedora). A comparação entre as opções de processamento que utilizaram o STB-index ou o HSTB-index como índice base deu vantagem para as opções de processamento utilizando a última mostrando que, ainda nesta seletividade baixa, a

poda de registros proporcionada pelo HSTB-index teve uma influência determinante no desempenho.

6.1.7 Análise da Configuração S1-address

Esta configuração espaço-temporal para endereços de fornecedores (saddress) com seletividade baixa (S1) apresentou resultados interessantes pela presença do *baseline* de visão materializada (na sua versão básica sem índices) separando as seis primeiras opções de processamento em dois blocos: da primeira à terceira opções de processamento utilizando o HSTB-index e o HSB-index como índice base e da quarta à sétima posição utilizando os índices STB-index e SB-index como base (Tabela 13). A opção de processamento vencedora foi a pipeTEhstb com 16,30% de ganho sobre o *baseline* de visão materializada seguida de perto das opções intmTEhstb, com 15,61% de ganho sobre o *baseline* de visão materializada e apenas 0,82% de perda para a opção vencedora, e pipeEhsbTepCf, com 9,46% de ganho sobre o *baseline* de visão materializada e 7,55% de perda para a opção de processamento vencedora. Dentre as opções de processamento que apresentaram perdas para o *baseline* de visão materializada, destacamos as três anteriormente citadas e cujas perdas não ultrapassaram 10%: intmTEhstb (6,97%), pipeEsbTepCf (8,94%) e pipeTEstb (9,47%). A pior opção de processamento foi a indpTfEsbCf com 91,64% de perda para o *baseline* de visão materializada e 93% de perda para a opção de processamento vencedora.

Outra observação interessante é a presença de níveis bem diferenciados de desempenho com relação à opção de processamento vencedora. Um primeiro grupo pode ser destacado com as opções de processamento pipeTEhstb, intmTEhstb e pipeEhsbTepCf que estiveram muito perto entre si e apresentaram porcentagens mínimas de perda de desempenho (entre 0% e 7,55%). O segundo grupo apresenta perdas crescentes (como uma transição) com o *baseline* de visão materializada (16,30%) e as opções de processamento intmTEstb (22,13%), pipeEsbTepCf (23,78%), pipeTEstb (24,23%), pipeTpEhsbCf (34,19%) e pipeTpEsbCf (37,08%). Começando com a opção de processamento pipeEhsbTepCf (85,59%) e indo até a opção de processamento indpTfEsbCf (93 %) temos o terceiro grupo com perdas acima de 85% sobre a opção de processamento vencedora. Isso indica uma

diferença de desempenho considerável em favor das opções de processamento vencedoras, onde apenas 3 se destacaram fortemente e 3 medianamente.

Tabela 13: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora – S1-saddress

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/per da sobre <i>baseline</i>	Porc perda para o vencedor
pipeTEhstb	Pipeline TE-C	HSTB	772723539,84	0,7727	16,299	0,000
intmTEhstb	ET Intercalado (FTRC)	HSTB	779094922,96	0,7791	15,609	0,818
pipeEhsbTepCf	Pipeline E-T-C	HSB	835864235,96	0,8359	9,459	7,554
VMB	VM básica	N/A	923191200,00	0,9232	0,000	16,299
intmTEstb	ET Intercalado (FTRC)	STB	992342072,32	0,9923	-6,968	22,131
pipeEsbTepCf	Pipeline E-T-C	SB	1013875015,96	1,0139	-8,944	23,785
pipeTEstb	Pipeline TE-C	STB	1019798891,60	1,0198	-9,473	24,228
pipeTpEhsbCf	Pipeline T-E-C	HSB	1174166064,76	1,1742	-21,375	34,190
pipeTpEsbCf	Pipeline T-E-C	SB	1228152111,72	1,2282	-24,831	37,082
pipeEhsbTefCf	Pipeline E-T-C	HSB	5361676646,96	5,3617	-82,782	85,588
indpTEhstb	Independente TE/C	HSTB	5534567400,08	5,5346	-83,320	86,038
pipeTfEhsbCf	Pipeline T-E-C	HSB	5571064786,56	5,5711	-83,429	86,130
indpTEstb	Independente TE/C	STB	5945991594,56	5,9460	-84,474	87,004
pipeTfEsbCf	Pipeline T-E-C	SB	6115245551,48	6,1153	-84,903	87,364
indpTpEsbCf	Independente T/E/C	SB	6223368362,92	6,2234	-85,166	87,584
indpTpEhsbCf	Independente T/E/C	HSB	6256343683,04	6,2563	-85,244	87,649
pipeEsbTefCf	Pipeline E-T-C	SB	9512728426,84	9,5127	-90,295	91,877
indpTfEhsbCf	Independente T/E/C	HSB	10488052091,08	10,4881	-91,198	92,632
indpTfEsbCf	Independente T/E/C	SB	11043004537,24	11,0430	-91,640	93,003

Cores usadas

ET Intercalado (FTRC)
 Pipeline TE-C
 Pipeline T-E-C
 Pipeline E-T-C
 Independente T/E/C
 Independente TE/C



STB-index
 HSTB-index
 SB-index
 HSB-index



Apesar das diferenças descritas acima, quando se observa o agrupamento das opções de processamento a partir das suas respectivas classes de processamento, vê-se a seguinte ordem do maior para o menor desempenho: *Pipeline* TE-C, ET intercalado (FTRC) e *Pipeline* E-T-C se revezando nas 7 primeiras posições com o *baseline* de visão materializada os separando, *Pipeline* T-E-C e *Pipeline* E-T-C se revezando, Independente TE/C e Independente T/E/C aparecendo na parte final da Tabela 13 com opções em *pipeline* entre elas. As classes de processamento das opções vencedoras têm em comum o processamento em *pipeline* e a utilização de um índice base envolvendo hierarquia de geometrias e, quando possível, a informação de validade destas na estrutura de índice.

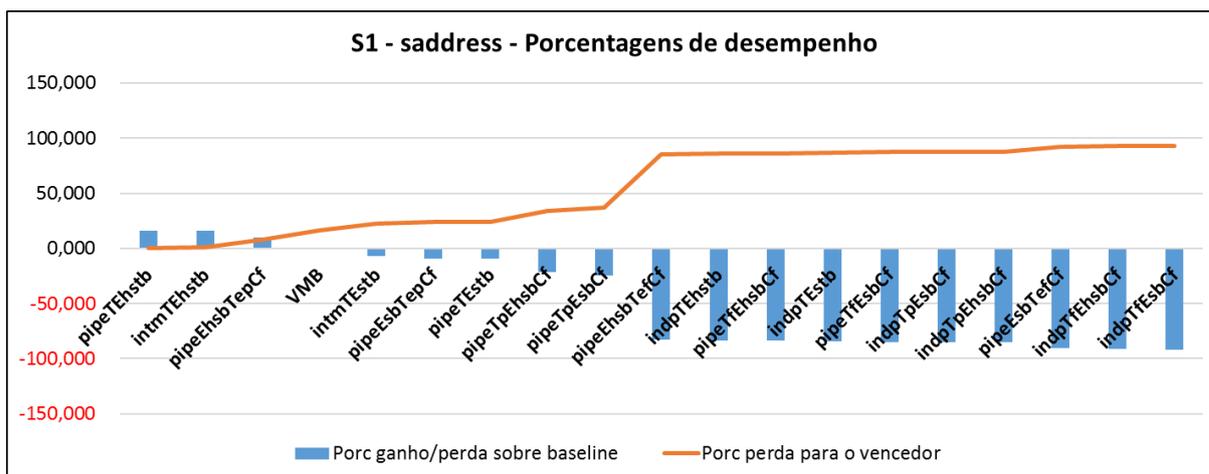


Figura 43: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S1-address

Quanto à comparação das técnicas de processamento temporal (Tp contra Tf e Tep contra Tef – descrita com mais detalhes na opção espaço-temporal S0-address), observa-se que todas as opções de processamento utilizando a interface com o banco de dados (Tp e Tep) levaram vantagem sobre suas opções de processamento relativas utilizando o software de indexação bitmap (Tf e Tef). Cabe aqui a mesma conclusão vista na opção espaço-temporal S0-address onde as opções Tp e Tep tiveram vantagem devido ao processamento em memória da interface com o PostgreSQL.

Observando-se como a ordenação entre as opções de processamento variou quando consideramos os índices base utilizados, vê-se que as opções baseadas no HSTB-index tiveram melhor desempenho sobre as suas opções relativas baseadas no STB-index em todas as classes. A situação é muito similar quando são comparadas as opções de processamento que utilizam o SB-index ou o HSB-index como índices base com este último auxiliando mais as opções de processamento que a utilizam e, assim, perfazendo um melhor desempenho quando comparadas à sua opção de processamento similar baseada no SB-index. A única exceção fica para a opção de processamento indpTpEsbCf que teve melhor desempenho que a sua relacionada utilizando o HSB-index como base (indpTpEhstbCf). Quando comparamos as opções de processamento que utilizam a técnica independente, vemos que as opções utilizando o HSTB-index tiveram melhor desempenho em separado e, conseqüentemente, no resultado final.

A Tabela 14 vem mostrar o desdobramento das opções de processamento vencedoras em cada parte do processamento e pode-se notar um agrupamento forte das duas primeiras opções com tempos muito similares nas duas fases de processamento. Vale notar que há o uso do software de implementação dos índices bitmap na limitação convencional o qual não temos controle sobre os tempos. A distância vai ficando maior a partir da terceira opção com a limitação espaço-temporal ocupando cada vez mais tempo de processamento. Apesar destas diferenças, o software de processamento bitmap (usado para processar o predicado convencional e gerar a resposta da consulta) foi quem realmente aumentou o resultado das opções de processamento listadas e decidiu quais seriam as ganhadoras.

Tabela 14: Tempos obtidos por cada opção de processamento vencedora na opção de processamento S1-saddress

Opção	Classe	Índice base	Limitação espaço-temporal (ns)	Limitação Convencional (ns)	Média Total (ns)
pipeTEhstb	Pipeline TE-C	HSTB	306.442,44	772.417.097,40	772.723.539,84
intmTEhstb	ET Intercalado (FTRC)	HSTB	321.282,64	778.773.640,32	779.094.922,96
pipeEhstbTepCf	Pipeline E-T-C	HSB	11.712.073,28	824.152.162,68	835.864.235,96
intmTEstb	ET Intercalado (FTRC)	STB	1.062.641,88	991.279.430,44	992.342.072,32
pipeEsbTepCf	Pipeline E-T-C	SB	12.858.150,76	1.001.016.865,20	1.013.875.015,96
pipeTEstb	Pipeline TE-C	STB	14.239.124,76	1.005.559.766,84	1.019.798.891,60

Abaixo das opções de processamento vencedoras, a comparação entre os índices base continuou mostrando que opções de processamento baseadas no HSB-index (hierárquico) tiveram vantagem sobre as suas opções relativas utilizando o SB-index (sequencial) como índice base com exceção da opção de processamento indpTpEsbCf e sua relativa utilizando o HSB-index como base (nominalmente, indpTpEhstbCf). Os mesmos motivos apresentados na opção espaço-temporal S0-caddress se aplicam a esta opção espaço-temporal.

6.1.8 Análise da Configuração S1-street

Nesta configuração espaço-temporal para ruas de fornecedores e clientes (tipo de dado linha – street) com seletividade baixa (S1), podemos observar na Tabela 15 o *baseline* de visão materializada na sua versão avançada (altamente indexada) sendo a opção de processamento vencedora com diferença de 70,54% de desempenho para a segunda colocada, a pipeTEhstb. Esta opção de

processamento foi seguida de perto pelas demais até a sétima colocada, a *pipeEsbTepCf*, com uma diferença de desempenho de menos de 10% quando estas foram comparadas à opção de processamento vencedora (70,54% de perda da opção *pipeTEhstb* a 79,92% de perda da opção *pipeEsbTepCf*). Mais uma vez temos a explicação de tal comportamento na simplicidade da construção do banco de dados, no qual linhas horizontais e verticais foram utilizadas para a tabela de ruas e, assim, facilitou os cálculos realizados pelo sistema gerenciador de banco de dados. Outro fator que contribuiu foi a correta utilização dos índices pelo otimizador de consultas do PostgreSQL. A pior opção de processamento foi a *pipeEsbTefCf* com 97,97% de perda para o *baseline* de visão materializada. A exemplo da configuração S0-street, a melhoria provida ao *baseline* de visão materializada e o correto processamento deste fez com que todas as outras opções de processamento fossem ultrapassadas em mais de 70% pelo seu desempenho e podemos citar a necessidade da inclusão de uma fase de refinamento do tipo de dados linha (por se tratar de uma consulta do tipo IRQ) como outro diferencial afetando negativamente o processamento das demais opções.

O agrupamento das opções de processamento nas suas respectivas classes de processamento (do maior para o menor desempenho) resultou na seguinte ordem: *Baseline* de visão materializada, *Pipeline* TE-C, *Pipeline* E-T-C e ET Intercalado (FTRC) alternando-se nas primeiras posições em função do tipo de índice base utilizado, uma das opções pertencentes ao *Pipeline* T-E-C, Independente TE/C e Independente T/E/C. As classes independentes possuem algumas opções de processamento que utilizam processamento em *pipeline* alternando posições entre elas. Tal agrupamento de classes indica que as opções de processamento em *pipeline* possuem em sua maioria melhor desempenho que as opções de processamento independentes e, dentre das opções de processamento que utilizam *pipeline*, as que se baseiam no HSB-index e HSTB-index possuem melhor desempenho que as baseadas no SB-index ou STB-index por conta da poda de registros proporcionada pela hierarquia de MBRs, a qual se mostrou ainda mais interessante quando complementada pela informação espaço-temporal já incluída na estrutura do índice, assim simplificando o processamento do mesmo.

Tabela 15: – Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S1-street

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/perda sobre baseline	Porc perda para o vencedor
VMA	VM indexada	BTree	281047000,00	0,2811	0,000	0,000
pipeTEhstb	Pipeline TE-C	HSTB	953881727,28	0,9539	-70,536	70,536
pipeEhstbTepCf	Pipeline E-T-C	HSB	956689693,00	0,9567	-70,623	70,623
intmTEhstb	ET Intercalado (FTRC)	HSTB	979385488,08	0,9794	-71,304	71,304
intmTEstb	ET Intercalado (FTRC)	STB	1182771637,88	1,1828	-76,238	76,238
pipeTEstb	Pipeline TE-C	STB	1217100376,48	1,2171	-76,908	76,908
pipeEsbTepCf	Pipeline E-T-C	SB	1399564883,60	1,3996	-79,919	79,919
pipeTpEhstbCf	Pipeline T-E-C	HSB	3009065688,64	3,0091	-90,660	90,660
pipeTpEsbCf	Pipeline T-E-C	SB	3308442218,08	3,3084	-91,505	91,505
indpTEhstb	Independente TE/C	HSTB	5624807596,32	5,6248	-95,003	95,003
pipeTfEsbCf	Pipeline T-E-C	SB	5779507510,48	5,7795	-95,137	95,137
indpTEstb	Independente TE/C	STB	5876914772,92	5,8769	-95,218	95,218
pipeEhstbTefCf	Pipeline E-T-C	HSB	5918944772,72	5,9189	-95,252	95,252
pipeTfEhstbCf	Pipeline T-E-C	HSB	6119536166,60	6,1195	-95,407	95,407
indpTpEhstbCf	Independente T/E/C	HSB	7900291182,84	7,9003	-96,443	96,443
indpTpEsbCf	Independente T/E/C	SB	8290062310,76	8,2901	-96,610	96,610
indpTfEhstbCf	Independente T/E/C	HSB	10742008364,28	10,7420	-97,384	97,384
indpTfEsbCf	Independente T/E/C	SB	10999645926,84	10,9997	-97,445	97,445
pipeEsbTefCf	Pipeline E-T-C	SB	13890526665,80	13,8905	-97,977	97,977

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

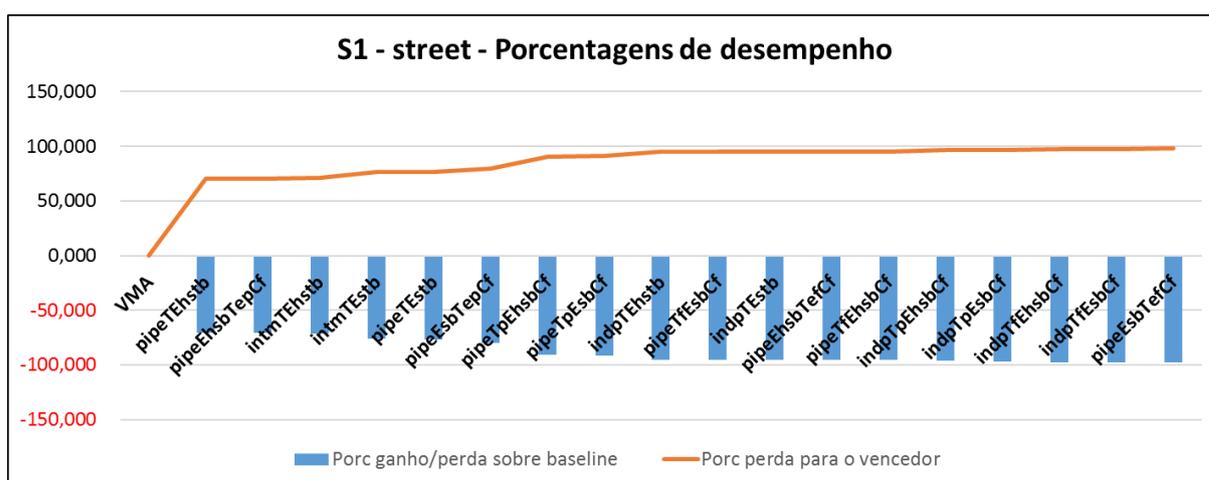


Figura 44: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S1-street

Na comparação entre as técnicas de processamento puramente temporais (T_p contra T_f e T_{ep} contra T_{ef}), notamos o mesmo comportamento da opção espaço-temporal S0-street onde as opções de processamento que utilizaram as técnicas T_p ou T_{ep} tiveram vantagem sobre as opções utilizando as técnicas T_f e T_{ef} . Os mesmos motivos também se aplicam a esta configuração espaço-temporal.

Comparando-se o desempenho de cada opção de processamento que utiliza o SB-index (vetor de MBRs) como base contra a sua respectiva opção de processamento utilizando o HSB-index (hierarquia de MBRs), temos também o mesmo comportamento da opção espaço-temporal S0-street com as opções de processamento baseadas no SB-index sendo sobrepujadas pelo desempenho das opções de processamento baseadas no HSB-index. Há uma única exceção nesta regra com a opção pipeTfEsbCf obtendo um melhor desempenho frente à sua opção relativa utilizando o HSB-index (a pipeTfEhsbCf). A comparação entre as opções de processamento baseadas no STB-index contra as baseadas no HSTB-index também possui as mesmas características da opção S0-street. Os mesmos motivos e comentários se aplicam também a esta configuração espaço-temporal.

6.1.9 Análise da Configuração S1-city

Nesta configuração espaço-temporal para cidades de fornecedores e clientes (tipo de dado polígono – city) com seletividade baixa (S1), o *baseline* de visão materializada obteve um desempenho médio separando nove opções de processamento que tiveram ganhos contra nove que tiveram perdas de desempenho. Observa-se a opção intmTEhstb como a vencedora com 51,80% de ganho sobre o *baseline* de visão materializada. Em segundo lugar tem-se a opção de processamento pipeTEhstb com 51,61% de ganho sobre o *baseline* de visão materializada e apenas 0,38% de perda para a opção vencedora. A opção pipeEhsbTepCf também conseguiu resultados próximos à opção vencedora com apenas 0,42% de perda de desempenho. Além dessas três opções de processamento, destacam-se as opções pipeTfEhsbCf e pipeTpEhsbCf cujas perdas ficaram abaixo dos 15% de perda para a opção de processamento vencedora e puderam rivalizar contra esta.

A pior opção de processamento foi a pipeEsbTefCf com 85,74% de perda para o *baseline* de visão materializada e 93,13% de perda para a opção de

processamento vencedora. As opções que tiveram perda contra o *baseline* de visão materializada apresentaram perdas de, no mínimo, 65,15% (indpTEhstb) e, no máximo, de 85,74% (pipeEsbTefCf).

A observação do agrupamento das classes das opções de processamento nos dá a seguinte ordem (do melhor para o pior desempenho): ET intercalado (FTRC) alternando com as classes *Pipeline* TE-C, *Pipeline* E-T-C e *Pipeline* T-E-C em função do tipo de índice espaço-temporal, Independente TE/C e Independente T/E/C com algumas opções das classes anteriores em seu meio. Desta ordenação, vemos que as opções de processamento em *pipeline* são eficientes para ultrapassar o *baseline* de visão materializada do qual esperava-se um processamento melhor devido à característica dos polígonos utilizados (retângulos relativamente “bem-comportados” e com muito pouco “dead space”) mesmo nesta seletividade baixa. Já as classes de processamento independentes sofrem mais uma vez com a necessidade de processamento duplo do predicado convencional e perdem para o *baseline* de visão materializada consistentemente.

Quanto à comparação das técnicas de processamento temporal (Tp contra Tf e Tep contra Tef – descrita com mais detalhes na configuração espaço-temporal S0-city), observa-se que todas as opções de processamento utilizando a interface com o banco de dados (Tp e Tep) levaram vantagem sobre suas opções de processamento relativas utilizando o software de indexação bitmap (Tf e Tef) inclusive com a mesma exceção para a opção pipeTfEhsbCf contra a sua relativa (a pipeTpEhsbCf). Cabe aqui a mesma conclusão vista na opção espaço-temporal S0-city, onde as opções Tp e Tep tiveram vantagem por conta do processamento em memória da interface com o PostgreSQL.

Dentre as opções de processamento vencedoras, o destaque fica para a diferença da primeira até a quinta opção de apenas 5,26%. Todas utilizam índices baseados em hierarquia como base e, apesar dos tempos muito próximos, vê-se uma ligeira vantagem para as opções que possuem as informações para limitação temporal (datas de validade do registro) e espacial (MBR) na estrutura dos índices.

Tabela 16: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S1-city

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/perda sobre baseline	Porc perda para o vencedor
intmTEhstb	ET Intercalado (FTRC)	HSTB	908030414,04	0,9080	51,797	0,000
pipeTEhstb	Pipeline TE-C	HSTB	911491910,48	0,9115	51,614	0,380
pipeEhstbTepCf	Pipeline E-T-C	HSB	911846561,00	0,9119	51,595	0,419
pipeTfEhstbCf	Pipeline T-E-C	HSB	954724198,76	0,9547	49,319	4,891
pipeTpEhstbCf	Pipeline T-E-C	HSB	958457075,60	0,9585	49,120	5,261
pipeTEstb	Pipeline TE-C	STB	1118650263,12	1,1187	40,617	18,828
intmTEstb	ET Intercalado (FTRC)	STB	1119265224,12	1,1193	40,584	18,873
pipeTpEsbCf	Pipeline T-E-C	SB	1183669519,96	1,1837	37,165	23,287
pipeEsbTepCf	Pipeline E-T-C	SB	1306125782,12	1,3061	30,665	30,479
VMA	VM indexada	BTree	1883778240,00	1,8838	0,000	51,797
indpTEhstb	Independente TE/C	HSTB	5404843834,36	5,4048	-65,146	83,200
pipeEhstbTefCf	Pipeline E-T-C	HSB	5862842900,20	5,8628	-67,869	84,512
indpTpEsbCf	Independente T/E/C	SB	6039552226,40	6,0396	-68,809	84,965
indpTEstb	Independente TE/C	STB	6210534864,52	6,2105	-69,668	85,379
pipeTfEsbCf	Pipeline T-E-C	SB	6263872154,36	6,2639	-69,926	85,504
indpTpEhstbCf	Independente T/E/C	HSB	6298461141,12	6,2985	-70,091	85,583
indpTfEsbCf	Independente T/E/C	SB	10724406419,80	10,7244	-82,435	91,533
indpTfEhstbCf	Independente T/E/C	HSB	11119279564,56	11,1193	-83,058	91,834
pipeEsbTefCf	Pipeline E-T-C	SB	13216309025,44	13,2163	-85,747	93,129

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

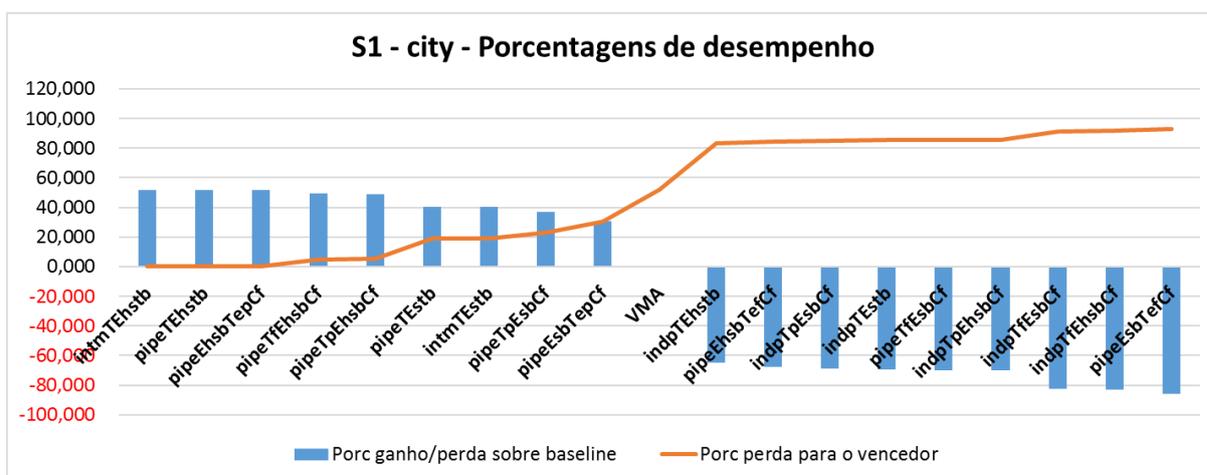


Figura 45: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S1-city

Na comparação dos índices base das opções de processamento, mais especificamente nas opções que utilizam processamento em *pipeline*, a maior limitação e poda de candidatos das opções de processamento baseadas no HSB-index continuou tendo um melhor desempenho quando comparadas às suas opções de processamento relacionadas baseadas no SB-index (pipeTpEhsbCf contra pipeTpEsbCf, por exemplo). As opções de processamento independentes fugiram a essa regra e percebemos a opção indpTpEsbCf (baseada no SB-index) tendo um melhor desempenho contra a opção indpTpEhsbCf (baseada no HSB-index) e a opção indpTfEsbCf com melhor desempenho que a opção indpTfEhsbCf.

A Tabela 17 e a Tabela 18 nos mostram a comparação das opções indpTpEsbCf, indpTpEhsbCf, indpTfEsbCf e indpTfEhsbCf em cada fase de processamento, a saber: Tp corresponde à limitação temporal com consulta ao PostgreSQL; Tf corresponde ao processamento no software de indexação bitmap; as saídas de Tp e Tf são guardadas em um tipo de dados do tipo lista cujos nós estão organizados em ordem crescente; Esb (ou Ehsb) e Refinement correspondem aos passos de processamento para limitação espacial do SB-index (ou HSB-index) e cuja saída é guardada em uma segunda lista ordenada; L1 é a lista resultante do processamento co-sequencial das duas listas anteriores fazendo uma operação de interseção e que vai conter os candidatos espaço-temporais; Cf é a limitação convencional processada com o software de indexação bitmap (fastbit) e cujo arquivo texto de saída (já ordenado) é lido para uma outra estrutura de lista; L2 é a junção co-sequencial da lista de candidatos espaço-temporais com a dos candidatos vindos da limitação convencional também numa operação de interseção; CfResults é a parte de processamento final da consulta no software de indexação bitmap e que vai formar a resposta final da consulta. Além da organização na estrutura do índice (sequencial no SB-index e hierárquico no HSB-index), a principal diferença entre as opções é a linguagem de programação utilizada onde o SB-index é construído em C++ e o HSB-index em C. Isso influi na API do PostgreSQL utilizada, na implementação das listas e na leitura e manipulação de arquivos. Mas a principal (e inesperada) diferença ficou com o processamento dos resultados (CfResults), onde a mesma consulta é realizada utilizando-se o mesmo software de indexação bitmap. O processamento de listas ficou dentro do esperado pois em C++ foi utilizado o tipo "List" nativo da linguagem enquanto em C o tipo de dados precisou ser criado e o foi feito de maneira customizada e otimizada para o índice e, portanto, apresentou

menores tempos. Os processamentos Tp e Refinement dependem do desempenho do *driver* PosgreSQL e este se manteve bem variado entre as opções de processamento mas representou pouca diferença no processamento destas. Na média, as imprevisibilidades e particularidades discutidas penderam a favor das opções baseadas no SB-index.

Tabela 17: Tempos médios de processamento das opções indpTpEsbCf e indpTpEhsbCf

OpçãoTpEsb	Predicado TpEsb	Tempo médio TpEsb (ns)	TpEsb	Diferença(%)	TpEhsb	OpçãoTpEhsb	Predicado TpEhsb	Tempo médio TpEhsb (ns)
indpTpEsbCf	Tp	52.397.193,72		5,65	>	indpTpEhsbCf	Tp	49.434.457,00
indpTpEsbCf	Esb	198.063,40	<	52,10		indpTpEhsbCf	Ehsb	413.528,52
indpTpEsbCf	Refinement	119.291.462,72	<	9,04		indpTpEhsbCf	Refinement	131.140.569,44
indpTpEsbCf	L1	467.791,36		70,52	>	indpTpEhsbCf	L1	137.899,44
indpTpEsbCf	Cf	5.562.758.608,92		0,14	>	indpTpEhsbCf	Cf	5.555.229.636,08
indpTpEsbCf	L2	29.180,84		3,79	>	indpTpEhsbCf	L2	28.074,20
indpTpEsbCf	CfResults	304.409.925,44	<	45,84		indpTpEhsbCf	CfResults	562.076.976,44
indpTpEsbCf	Total	6.039.552.226,40	<	4,11		indpTpEhsbCf	Total	6.298.461.141,12

Tabela 18: Tempos médios de processamento das opções indpTfEsbCf e indpTfEhsbCf

OpçãoTfEsb	Predicado TfEsb	Tempo médio TfEsb (ns)	TfEsb	Diferença(%)	TfEsb2	OpçãoTfEhsb	Predicado TfEhsb	Tempo médio TfEhsb (ns)
indpTfEsbCf	Tf	5.143.801.560,64		0,15	>	indpTfEhsbCf	Tf	5.136.203.452,68
indpTfEsbCf	Esb	912.803,80		47,30	>	indpTfEhsbCf	Ehsb	481.020,88
indpTfEsbCf	Refinement	142.596.011,44		12,95	>	indpTfEhsbCf	Refinement	124.128.389,32
indpTfEsbCf	L1	711.034,16		93,88	>	indpTfEhsbCf	L1	43.525,64
indpTfEsbCf	Cf	5.109.801.164,24	<	3,69		indpTfEhsbCf	Cf	5.305.794.820,56
indpTfEsbCf	L2	29.313,12		11,85	>	indpTfEhsbCf	L2	25.840,80
indpTfEsbCf	CfResults	326.554.532,40	<	40,91		indpTfEhsbCf	CfResults	552.602.514,68
indpTfEsbCf	Total	10.724.406.419,80	<	3,55		indpTfEhsbCf	Total	11.119.279.564,56

A comparação das opções de processamento baseadas no STB-index contra as do HSTB-index continua indicando os mesmos resultados e conclusões da configuração espaço-temporal S0-caddress e S0-saddress, mas sem exceções à regra.

6.1.10 Análise da Configuração S2-caddress

Como pode ser observado na Tabela 19, a opção vencedora para a granularidade média (S2) com pontos vindos da tabela de clientes (caddress) foi a *intmTEhstb* com 35,61% de ganho sobre o *baseline* de visão materializada na sua versão básica (sem indexação). É interessante notar que esta configuração espaço-temporal possui as opções de processamento da segunda à quarta melhores posições muito próximas com apenas 4,72% de perda máxima contra a opção de

processamento vencedora. São elas a *pipeTEhstb*, a *pipeTEstb* e, por fim, a *intmTEstb*. A pior opção de processamento foi a *indpTfEhsbCf* com 91,53% de perda para o *baseline* de visão materializada e 94,54% de perda para a opção vencedora.

A única opção de processamento que conseguiu ficar próxima das quatro opções vencedoras (menos de 15% de perda para a opção vencedora) foi a *pipeEhsbTepCf*, com 26,71% de ganho para o *baseline* de visão materializada e 12,14% de perda para a opção de processamento vencedora. Além das opções citadas acima, nota-se na Tabela 19 que apenas a opção *pipeEsbTepCf* obteve ganho de desempenho sobre o *baseline* de visão materializada (12,54%). Do outro lado, todas as opções de processamento que tiveram perda de desempenho em comparação com o *baseline* de visão materializada o tiveram com mais de 80%, começando com a opção *indpTEhstb* (83,10%) e indo até a pior opção, a *indpTfEhsbCf*, com 91,53% de perda.

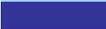
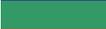
Ao analisar a Figura 46, pode-se observar novamente o comportamento em “plataformas” quando comparamos a perda de desempenho das opções de processamento contra a opção vencedora. As quatro primeiras opções formam a primeira plataforma com perdas até 4,7%. As três opções seguintes formam a segunda plataforma com perdas variando de 12,14% a 35,61%. A terceira e última plataforma contém todas as outras opções de processamento com perdas acima de 89,12% contra a opção vencedora.

O agrupamento das opções de processamento em suas respectivas classes nos mostra a seguinte ordem (do melhor para o pior desempenho): ET intercalado (FTRC) com as duas opções da classe *Pipeline* TE-C entre elas, *Pipeline* E-T-C, *baseline* de visão materializada e a Independente TE/C juntamente com a *Pipeline* T-E-C e *pipeline* E-T-C se revezando nas posições intermediárias, e, por fim, a Independente T/E/C nas últimas posições. As opções em *pipeline* (que utilizam a lista de candidatos de uma fase de processamento como limitador na fase seguinte) se mostraram superiores às opções independentes em sua grande maioria. Opções que possuem as informações para limitação espaço-temporal na estrutura do índice (datas de validade temporal e coordenadas de MBR) levaram vantagem sobre as opções de que não as possuem.

Tabela 19: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S2-address

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/perda sobre baseline	Porc perda para o vencedor
intmTEhstb	ET Intercalado (FTRC)	HSTB	589725834,56	0,5897	35,608	0,000
pipeTEhstb	Pipeline TE-C	HSTB	594174977,56	0,5942	35,122	0,749
pipeTEstb	Pipeline TE-C	STB	616974334,32	0,6170	32,632	4,416
intmTEstb	ET Intercalado (FTRC)	STB	618948982,52	0,6190	32,417	4,721
pipeEhsbTepCf	Pipeline E-T-C	HSB	671181042,88	0,6712	26,714	12,136
pipeEsbTepCf	Pipeline E-T-C	SB	800976733,80	0,8010	12,541	26,374
VMB	VM básica	N/A	915832160,00	0,9158	0,000	35,608
indpTEhstb	Independente TE/C	HSTB	5418257262,56	5,4183	-83,097	89,116
pipeTfEhsbCf	Pipeline T-E-C	HSB	5631147480,60	5,6312	-83,736	89,527
pipeTfEsbCf	Pipeline T-E-C	SB	5642429718,80	5,6424	-83,769	89,548
indpTEstb	Independente TE/C	STB	5660051128,64	5,6601	-83,819	89,581
pipeEhsbTefCf	Pipeline E-T-C	HSB	5663890619,44	5,6639	-83,830	89,588
pipeTpEsbCf	Pipeline T-E-C	SB	6239913665,16	6,2399	-85,323	90,549
pipeTpEhsbCf	Pipeline T-E-C	HSB	7092482833,92	7,0925	-87,087	91,685
pipeEsbTefCf	Pipeline E-T-C	SB	7250361326,24	7,2504	-87,368	91,866
indpTpEhsbCf	Independente T/E/C	HSB	9053110694,60	9,0531	-89,884	93,486
indpTpEsbCf	Independente T/E/C	SB	9908782392,72	9,9088	-90,757	94,048
indpTfEsbCf	Independente T/E/C	SB	10390345889,16	10,3904	-91,186	94,324
indpTfEhsbCf	Independente T/E/C	HSB	10811021898,92	10,8110	-91,529	94,545

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

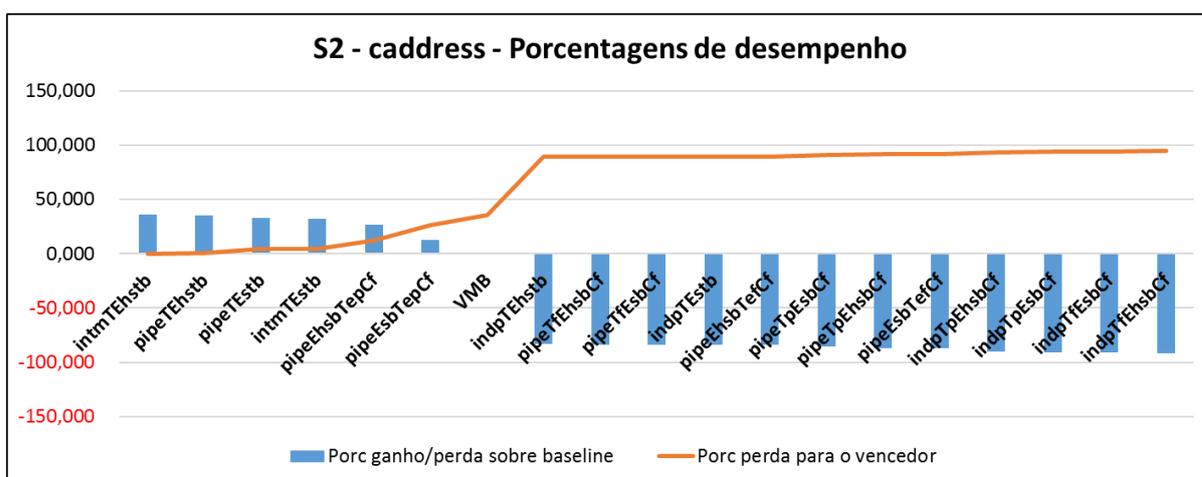


Figura 46: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S2-address

A comparação das técnicas de processamento do predicado temporal (Tp e Tep – baseadas em consultas ao sistema gerenciador de banco de dados – contra Tf e Tef – baseadas no software de indexação bitmap) apresentou o mesmo comportamento encontrado na configuração S0-caddress (item 6.1.2) somente para as classes de processamento *Pipeline* E-T-C e nas independentes. A situação se inverteu para as opções de processamento da classe *Pipeline* T-E-C, onde ambas as opções utilizando a técnica Tf obtiveram melhores desempenhos que a sua relativa baseada na técnica Tp. Analisando-se os tempos parciais das quatro opções vê-se um comportamento natural com as limitações dos predicados espacial, temporal e convencional não sofrendo alterações bruscas para as três primeiras opções (Tabela 20). Já a última opção de processamento apresentou tempo de processamento temporal e convencional compatíveis, mas seu tempo de processamento espacial foi aproximadamente três vezes acima da média e contribuiu negativamente para o seu desempenho. Nas análises anteriores (configurações S0-caddress e S1-caddress) a opção de processamento pipeTpEhsbCf já estava sendo ultrapassada consistentemente pela sua relativa baseada na técnica Tf e, agora, temos as duas opções de processamento sendo ultrapassadas. Tal observação indica a tendência de que, quanto maior a seletividade, melhor o desempenho das opções utilizando a técnica T5 em seu processamento.

Tabela 20: Tempos médios de processamento das opções da classe *Pipeline* T-E-C para a configuração S2-caddress

Opção	Índice base	Limitação Temporal (ns)	Limitação espacial (ns)	Limitação Convencional (ns)	Média Total (ns)
pipeTfEhsbCf	HSB	5.201.637.399,08	84.844.047,64	344.666.033,88	5.631.147.480,60
pipeTfEsbCf	SB	5.272.005.871,48	45.465.782,16	324.958.065,16	5.642.429.718,80
pipeTpEsbCf	SB	5.615.628.838,76	43.610.968,72	580.673.857,68	6.239.913.665,16
pipeTpEhsbCf	HSB	6.368.434.789,20	287.532.869,96	436.515.174,76	7.092.482.833,92

Dentre as opções de processamento vencedoras, o processamento hierárquico das opções de processamento baseadas no HSTB-index (intmTEhstb e pipeTEhstb) ainda representou uma vantagem em relação aos índices baseados em um vetor sequencial de MBRs com suas datas de validade que é a base do STB-index (pipeTEstb e intmTEstb). Quando aliado à limitação espacial a priori feita pela

opção *intmTEhstb*, fez esta levar vantagem sobre sua correspondente em *pipeline* (*pipeTEhstb*). Para alcançar a seletividade alvo desta configuração, foi mantida a seletividade temporal em níveis altos (68,74%) e variou-se a seletividade espacial até encontrar a seletividade espaço-temporal desejada entre 100 e 200 registros (108 registros ou 0,11%).

Na comparação dos índices base, pode-se notar um pequeno aumento das opções de processamento baseadas no SB-index que tiveram melhor desempenho que a sua opção relacionada baseada no HSB-index, a saber: *pipeTpEsbCf* e *pipeTpEhsbCf*; *indpTfEsbCf* e *indpTfEhsbCf*. Já para as opções de processamento baseadas no HSTB-index vemos estas superarem as suas opções relativas baseadas no STB-index (*pipeTEhstb* e *pipeTEstb*, por exemplo) consistentemente mesmo nesta seletividade. Como discutido anteriormente, as diferenças entre elas estão ficando cada vez menores, conforme esperava-se.

6.1.11 Análise da Configuração S2-address

A configuração para o tipo de dado ponto vindo dos endereços de fornecedores (*saddress*) com seletividade média (S2) apresentou a opção de processamento *pipeTEhstb* como vencedora com ganho de desempenho de 36,26% para o *baseline* de visão materializada na sua versão básica (sem indexação). A opção de processamento *intmTEhstb* ficou na segunda colocação com 35,88% de ganho sobre o *baseline* de visão materializada e apenas 0,59% de perda para a opção vencedora. Além destas opções de processamento, apenas a opção *pipeEhsbTpCf* apresentou um desempenho próximo da opção vencedora (perda de menos de 15%) com 12,37% de perda contra esta. Pode-se citar também as outras três opções de processamento que obtiveram ganhos sobre o *baseline* de visão materializada: *intmTEstb* (22,32%), *pipeTEstb* (21,92%) e *pipeEsbTepCf* (20,52%). Duas das opções de processamento merecem menção por terem desempenho pior que o *baseline* de visão materializada mas este ter ficado próximo de 1% de perda (o que não é tão significativo), a saber: *pipeTpEsbCf* (com 1,05% de perda) e *pipeTpEhsbCf* (com 1,08% de perda). A partir da opção de processamento *indpTEhstb*, todas as opções de processamento apresentaram perdas significativas em relação ao *baseline* de visão materializada começando com 75,07% da opção

indpTEhstb e aumentando até os 87,55% de perda apresentado pela última opção de processamento, que foi a indpTfEsbCf.

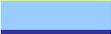
A exemplo de outras configurações espaço-temporais, também pode-se encontrar a presença de níveis bem diferenciados de desempenho com relação à opção de processamento vencedora. Um primeiro grupo pode ser destacado com as opções de processamento pipeTEhstb, intmTEhstb, pipeEhsbTepCf, intmTEstb, pipeTEstb e pipeEsbTepCf que tiveram perda de desempenho abaixo de 20% com relação à opção de processamento vencedora. O segundo grupo apresentou perdas praticamente constantes em 36% com o *baseline* de visão materializada (36,26%) e as opções de processamento pipeTpEsbCf (36,92%) e pipeTpEhsbCf (36,94%). Começando com a opção de processamento indpTEhstb (84,11%) e indo até a opção de processamento indpTfEsbCf (92,07%) temos o terceiro grupo com perdas acima de 80% sobre a opção de processamento vencedora. Isso indica uma diferença de desempenho considerável em favor das opções de processamento vencedoras, onde apenas 6 se destacaram fortemente e 2 medianamente.

Quando se observa o agrupamento das opções de processamento a partir das suas respectivas classes de processamento, vê-se a seguinte ordem do maior para o menor desempenho: *Pipeline* TE-C, ET intercalado (FTRC) e *Pipeline* E-T-C se revezando nas seis primeiras posições em função do tipo de índice espaço-temporal usado, *baseline* de visão materializada, *Pipeline* T-E-C, Independente TE/C e as opções restantes das classes *Pipeline* E-T-C e *Pipeline* T-E-C se revezando com as opções da classe Independente T/E/C na parte final da Tabela 21. O processamento em *pipeline* é presença constante dentre as opções de processamento vencedoras e tal fato não foi diferente para esta configuração, onde as seis primeiras colocações a utilizam. Dentre estas, destacam-se as três primeiras opções que utilizaram versões hierarquizadas de índices como índice base (o HSTB-index ou o HSB-index) e, observando-se as duas primeiras, vê-se que elas têm em comum o fato do MBR e das datas de validade dos mesmos estarem incluídos na estrutura do índice (HSTB-index como índice base).

Tabela 21: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S2-address

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/per da sobre baseline	Porc perda para o vencedor
pipeTEhstb	Pipeline TE-C	HSTB	839209429,16	0,8392	36,257	0,000
intmTEhstb	ET Intercalado (FTRC)	HSTB	844181117,48	0,8442	35,879	0,589
pipeEhstbTepCf	Pipeline E-T-C	HSB	957657622,24	0,9577	27,260	12,369
intmTEstb	ET Intercalado (FTRC)	STB	1022630105,72	1,0226	22,325	17,936
pipeTEstb	Pipeline TE-C	STB	1027942718,64	1,0279	21,922	18,360
pipeEsbTepCf	Pipeline E-T-C	SB	1046382905,00	1,0464	20,521	19,799
VMB	VM básica	N/A	1316550320,00	1,3166	0,000	36,257
pipeTpEsbCf	Pipeline T-E-C	SB	1330480869,00	1,3305	-1,047	36,924
pipeTpEhstbCf	Pipeline T-E-C	HSB	1330863736,68	1,3309	-1,076	36,943
indpTEhstb	Independente TE/C	HSTB	5281599757,20	5,2816	-75,073	84,111
indpTEstb	Independente TE/C	STB	5485312587,80	5,4853	-75,999	84,701
pipeEhstbTefCf	Pipeline E-T-C	HSB	5663927573,52	5,6639	-76,756	85,183
indpTpEhstbCf	Independente T/E/C	HSB	5810875009,08	5,8109	-77,343	85,558
pipeTfEhstbCf	Pipeline T-E-C	HSB	5934732290,40	5,9347	-77,816	85,859
pipeTfEsbCf	Pipeline T-E-C	SB	5987005391,04	5,9870	-78,010	85,983
indpTpEsbCf	Independente T/E/C	SB	6463559676,48	6,4636	-79,631	87,016
pipeEsbTefCf	Pipeline E-T-C	SB	9348093117,60	9,3481	-85,916	91,023
indpTfEhstbCf	Independente T/E/C	HSB	10427979464,68	10,4280	-87,375	91,952
indpTfEsbCf	Independente T/E/C	SB	10577242706,16	10,5772	-87,553	92,066

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

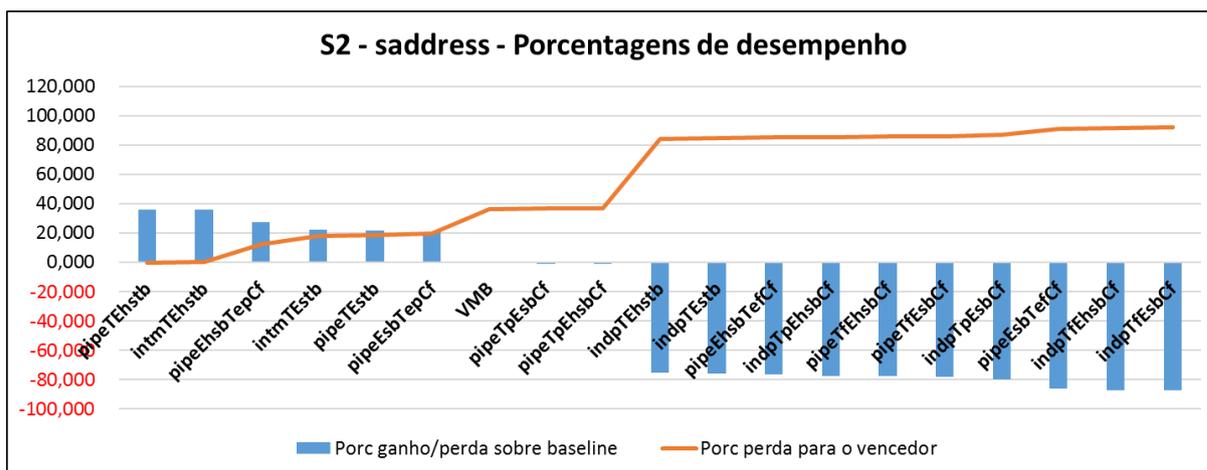


Figura 47: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S2-address

Quanto à comparação das técnicas de processamento temporal (Tp contra Tf e Tep contra Tef – descrita com mais detalhes na opção espaço-temporal S0-

caddress), observa-se que todas as opções de processamento utilizando a interface com o banco de dados (Tp e Tep) levaram vantagem sobre suas opções de processamento relativas utilizando o software de indexação bitmap (Tf e Tef). Cabe aqui a mesma conclusão vista na opção espaço-temporal S0-caddress (item 6.1.2) onde as opções Tp e Tep tiveram vantagem devido ao processamento em memória da interface com o PostgreSQL.

Podemos considerar como opções de processamento vencedoras aquelas que apresentaram ganhos de desempenho frente ao *baseline* de visão materializada, a saber: pipeTEhstb, intmTEhstb, pipeEhsbTepCf, intmTEstb, pipeTEstb e pipeEsbTepCf. Elas repetiram o comportamento apresentado na opção S0-saddress (item 6.1.3) onde as opções de processamento em *pipeline* tiveram melhor desempenho dentre as demais repetindo as suas propriedades ao evitar processamento de registros já fora da resposta final em estágios adiantados do processamento. A ordenação entre as opções de processamento continua com as opções baseadas no HSTB-index como vencedoras sobre as suas opções relativas baseadas no STB-index para as opções de processamento em todas as classes. Ficou mais uma vez confirmado que a propriedade de poda de registros influencia fortemente os resultados desta seletividade espaço-temporal.

Abaixo das opções de processamento vencedoras, a comparação entre os índices base continuou mostrando que opções de processamento baseadas no HSB-index (hierárquico) tiveram vantagem sobre as suas opções relativas utilizando o SB-index (sequencial) como índice base. Mais uma vez exceção à regra, a opção de processamento pipeTpEsbCf continua tendo melhores resultados que a sua relativa utilizando o HSB-index (a pipeTpEhsbCf). Os mesmos motivos apresentados na configuração S0-caddress (item 6.1.3) se aplicam a esta configuração.

6.1.12 Análise da Configuração S2-street

O comportamento do *baseline* de visão materializada nesta configuração de seletividade média (S2) utilizando o tipo de dado linha vindo da tabela *street* (street) seguiu o mesmo padrão da configuração S1-street (item 6.1.8) onde, na sua versão avançada (altamente indexada), obteve o melhor desempenho dentre todas as opções de processamento com 8,91% de ganho sobre a opção pipeTEhstb e 9,51% de ganho sobre a opção terceira colocada (a intmTEhstb). Da opção de

processamento pipeEhsbTepCf até a intmTEstb as perdas para o *baseline* de visão materializada (e também a opção vencedora) ficaram entre 16,82% e 20,77%. A partir da opção de processamento pipeEsbTepCf, as perdas de desempenho aumentam gradativamente até chegarem a seu máximo de 88,87% atingido pela opção de processamento pior colocada (a pipeEsbTefCf). As mesmas ponderações sobre os motivos de tal comportamento do *baseline* de visão materializada feitas na configuração S1-street se aplicam a esta configuração.

Tabela 22: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S2-street

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/per da sobre baseline	Porc perda para o vencedor
VMA	VM indexada	BTree	1575232480,00	1,5752	0,000	0,000
pipeTEhstb	Pipeline TE-C	HSTB	1729313952,56	1,7293	-8,910	8,910
intmTEhstb	ET Intercalado (FTRC)	HSTB	1740779586,24	1,7408	-9,510	9,510
pipeEhsbTepCf	Pipeline E-T-C	HSB	1893883035,84	1,8939	-16,825	16,825
pipeTEstb	Pipeline TE-C	STB	1972611126,68	1,9726	-20,145	20,145
intmTEstb	ET Intercalado (FTRC)	STB	1988189460,92	1,9882	-20,771	20,771
pipeEsbTepCf	Pipeline E-T-C	SB	2825657671,16	2,8257	-44,253	44,253
pipeTpEhsbCf	Pipeline T-E-C	HSB	3843851176,52	3,8439	-59,019	59,019
pipeTpEsbCf	Pipeline T-E-C	SB	4323649438,08	4,3237	-63,567	63,567
pipeEhsbTefCf	Pipeline E-T-C	HSB	6061191644,56	6,0612	-74,011	74,011
indpTEhstb	Independente TE/C	HSTB	6300466408,40	6,3005	-74,998	74,998
indpTEstb	Independente TE/C	STB	6590835647,40	6,5908	-76,100	76,100
pipeTfEsbCf	Pipeline T-E-C	SB	6787492332,56	6,7875	-76,792	76,792
pipeTfEhsbCf	Pipeline T-E-C	HSB	7257068841,96	7,2571	-78,294	78,294
indpTpEhsbCf	Independente T/E/C	HSB	8490511408,68	8,4905	-81,447	81,447
indpTpEsbCf	Independente T/E/C	SB	9075532694,80	9,0755	-82,643	82,643
indpTfEhsbCf	Independente T/E/C	HSB	11635674755,12	11,6357	-86,462	86,462
indpTfEsbCf	Independente T/E/C	SB	11748912621,24	11,7489	-86,593	86,593
pipeEsbTefCf	Pipeline E-T-C	SB	14150793029,84	14,1508	-88,868	88,868

Cores usadas

- ET Intercalado (FTRC)
- Pipeline TE-C
- Pipeline T-E-C
- Pipeline E-T-C
- Independente T/E/C
- Independente TE/C



- STB-index
- HSTB-index
- SB-index
- HSB-index



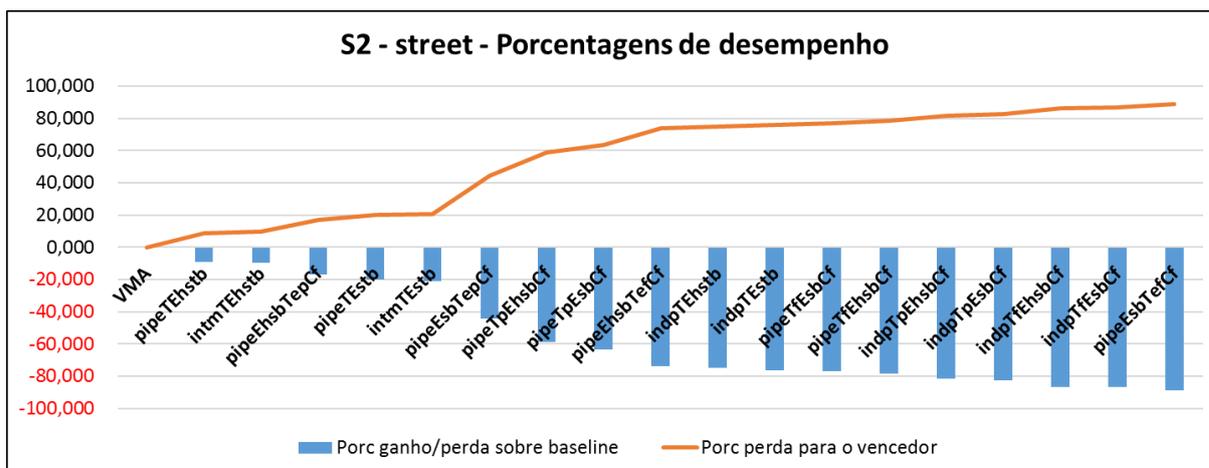


Figura 48: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S2-street

O agrupamento das opções de processamento nas suas respectivas classes de processamento resultou na seguinte ordem (do melhor para o pior desempenho): *Baseline* de visão materializada, *Pipeline* TE-C, se revezando nas primeiras posições com as classes *Pipeline* TE-C Intercalado (FTRC) e *Pipeline* E-T-C, *Pipeline* T-E-C, Independente TE/C e Independente T/E/C. As classes nas posições intermediárias estão agrupadas quase que duas a duas com opções das classes em *pipeline* entre elas. Tal agrupamento de classes ficou bem próximo do que se verificou na configuração S1-street (item 6.1.8) e as mesmas conclusões se aplicam a esta configuração. Na comparação entre as técnicas de processamento puramente temporais (Tp contra Tf e Tep contra Tef), notamos o mesmo comportamento apresentado em configurações anteriores onde as opções Tp e Tep, sem exceções, tiveram vantagem devido ao processamento em memória da interface com o PostgreSQL.

Tendo o *baseline* de visão materializada como opção de processamento melhor colocada, podemos dizer que as opções de processamento que mais se aproximaram do seu desempenho (até aproximadamente 20% de perda) são as melhores depois deste. Isso nos deixa com seis opções de processamento (pipeTEhstb, intmTEhstb, pipeEhstbTepCf, pipeTEstb, intmTEstb e pipeEsbThsbCf) cujo comportamento segue as mesmas tendências anteriores com as opções que utilizaram índices base hierárquicos (HSTB-index e HSB-index) em vantagem sobre as opções utilizando índices base sequenciais (STB-index e SB-index) e, com a

ajuda extra dada pela presença da informação de limitação espaço-temporal na estrutura do índice, as opções baseadas no HSTB-index e STB-index tiveram vantagem sobre suas relativas que possuíam apenas a informação de limitação espacial no índice (HSB-index e SB-index) e a limitação temporal sendo feita à parte.

Comparando-se o desempenho de cada opção de processamento que utiliza o SB-index (vetor de MBRs) como base contra a sua respectiva opção de processamento utilizando o HSB-index (hierarquia de MBRs), temos o mesmo comportamento observado na configuração S1-street (inclusive com a exceção para a opção pipeTfEsbCf) e as mesmas observações e conclusões se aplicam a esta configuração. Na comparação entre as opções de processamento utilizando o STB-index ou o HSTB-index, as opções de processamento utilizando este último estiveram sempre à frente das opções relativas utilizando o SB-index, fruto da poda de candidatos que ainda se apresenta forte nesta seletividade.

6.1.13 Análise da Configuração S2-city

Nesta opção espaço-temporal para cidades de fornecedores e clientes (polígonos na tabela de cidades – city) com seletividade média (S2), o *baseline* de visão materializada ficou bastante aquém em desempenho e muitas opções de processamento obtiveram ganhos contra este. Novamente observa-se a opção intmTEhstb como a vencedora com 78,19% de ganho sobre o *baseline* de visão materializada. Em segundo lugar tem-se a opção de processamento pipeEhsbTepCf com 77,79% de ganho sobre o *baseline* de visão materializada e somente 1,78% de perda para a opção vencedora. Além dela, a opção de processamento pipeTEhstb também obteve desempenho próximo da opção vencedora com apenas 1,80% de perda para esta. A pior opção de processamento foi a pipeEsbTefCf com 57,22% de perda para o *baseline* de visão materializada e 90,67% de perda para a opção de processamento vencedora. Ainda tiveram destaque as opções pipeTpEhsbCf, pipeTfEhsbCf, pipeTEstb, intmTEstb por terem ganhos acima de 70% sobre o *baseline* de visão materializada e menos de 15% de perda de desempenho para a opção de processamento vencedora. Nove opções tiveram perda contra o *baseline* de visão materializada começando com a opção indpTpEhsbCf (7,69%) e indo até a opção de processamento pior colocada, que foi a pipeEsbTefCf com 57,22% de

perda para o *baseline* de visão materializada e 90,67% de perda para a opção vencedora.

Quanto à comparação das técnicas de processamento temporal (Tp contra Tf e Tep contra Tef – descrita com mais detalhes na configuração S0-caddress do item 6.1.2), observa-se que todas as opções de processamento utilizando a interface com o banco de dados (Tp e Tep) levaram vantagem sobre suas opções de processamento relativas utilizando o software de indexação bitmap (Tf e Tef). A mesma conclusão vista na configuração S0-caddress também se aplica nesta configuração onde as opções Tp e Tf tiveram vantagem devido ao processamento em memória da interface com o sistema gerenciador de banco de dados.

Tabela 23: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S2-city

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/per da sobre baseline	Porc perda para o vencedor
intmTEhstb	ET Intercalado (FTRC)	HSTB	1238126237,76	1,2381	78,186	0,000
pipeEhsbTepCf	Pipeline E-T-C	HSB	1260518487,84	1,2605	77,792	1,776
pipeTEhstb	Pipeline TE-C	HSTB	1260885699,60	1,2609	77,785	1,805
pipeTpEhsbCf	Pipeline T-E-C	HSB	1290071589,68	1,2901	77,271	4,027
pipeTfEhsbCf	Pipeline T-E-C	HSB	1306429097,36	1,3064	76,983	5,228
pipeTEstb	Pipeline TE-C	STB	1448174023,72	1,4482	74,485	14,504
intmTEstb	ET Intercalado (FTRC)	STB	1448400443,52	1,4484	74,481	14,518
pipeTpEsbCf	Pipeline T-E-C	SB	1517240505,24	1,5172	73,268	18,396
pipeEsbTepCf	Pipeline E-T-C	SB	1676231155,60	1,6762	70,467	26,136
VMB	VM básica	N/A	5675849480,00	5,6759	0,000	78,186
indpTpEhsbCf	Independente T/E/C	HSB	6148605404,48	6,1486	-7,689	79,863
pipeEhsbTefCf	Pipeline E-T-C	HSB	6285538184,80	6,2855	-9,700	80,302
indpTEstb	Independente TE/C	STB	6315013671,92	6,3150	-10,121	80,394
indpTEhstb	Independente TE/C	HSTB	6321092413,24	6,3211	-10,208	80,413
pipeTfEsbCf	Pipeline T-E-C	SB	6409441542,36	6,4094	-11,445	80,683
indpTpEsbCf	Independente T/E/C	SB	6553827855,72	6,5538	-13,396	81,108
indpTfEsbCf	Independente T/E/C	SB	11041667827,08	11,0417	-48,596	88,787
indpTfEhsbCf	Independente T/E/C	HSB	11216216776,12	11,2162	-49,396	88,961
pipeEsbTefCf	Pipeline E-T-C	SB	13266647128,96	13,2667	-57,217	90,667

Cores usadas

- ET Intercalado (FTRC)
- Pipeline TE-C
- Pipeline T-E-C
- Pipeline E-T-C
- Independente T/E/C
- Independente TE/C



- STB-index
- HSTB-index
- SB-index
- HSB-index



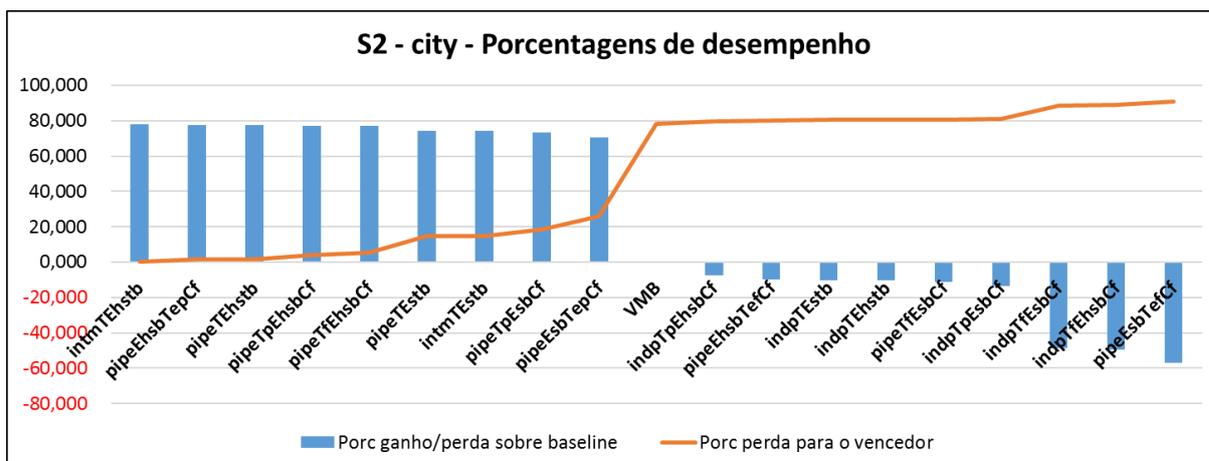


Figura 49: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S2-city

Mais uma vez temos um comportamento em “plataformas” quando comparamos o desempenho das opções de processamento contra a opção de processamento vencedora, mas, desta vez, com dois níveis somente. Como pode ser visto na Figura 49, as nove primeiras opções de processamento formam o primeiro nível da plataforma com até 26,14% de perda de desempenho, depois há uma rápida transição para o segundo degrau que se inicia com o *baseline* de visão materializada (78,19% de perda para a opção vencedora) e continua até a opção com o pior processamento. Tal resultado adiciona importância à utilização da técnica em *pipeline* para diminuir o esforço de processamento a cada predicado processado e, assim, melhorar o desempenho do índice.

Na comparação dos índices base das opções de processamento, mais especificamente nas opções que utilizam processamento em *pipeline*, a maior limitação e poda de candidatos das opções de processamento baseadas no HSB-index continuou tendo um melhor desempenho quando comparadas às suas opções de processamento relacionadas baseadas no SB-index (pipeTpEhstbCf contra pipeTpEsbCf, por exemplo). As opções de processamento independentes tiveram uma exceção a essa regra e percebemos a opção indpTfEsbCf tendo um melhor desempenho contra a opção indpTpEsbCf embora esta seja de apenas 0,17s.

A comparação das opções de processamento baseadas no STB-index contra as do HSTB-index nos mostra que as opções de processamento em *pipeline* (usando intercalamento ou não) tiveram as opções de processamento que usaram o

HSTB-index como base em vantagem contra as que utilizaram o STB-index como índice base. A situação inversa aconteceu com as opções de processamento da classe independente, onde a indpTEstb obteve melhor desempenho que a sua relativa usando o HSTB-index. A força da poda de candidatos proporcionada pela hierarquização da estrutura (HSTB-index) ainda se mostra forte nesta seletividade.

6.1.14 Análise da Configuração S3-caddress

Esta configuração espaço-temporal para endereços de clientes (caddress) com seletividade alta (S3) apresentou resultados interessantes pelo agrupamento quase perfeito das classes de processamento.

Outra observação interessante é a presença de níveis bem diferenciados de desempenho, onde temos as opções de processamento vencedoras – pipeTEstb com 52,83% de ganho sobre o *baseline* de visão materializada; intmTEstbc com 51,51% de ganho sobre o *baseline* de visão materializada e 2,73% de perda para a opção vencedora; intmTEhstb com 44,53% de ganho sobre o *baseline* de visão materializada e 15,24% de perda para a opção vencedora; pipeTEhstb com 43,19% de ganho para o *baseline* de visão materializada e 16,98% de perda para a opção vencedora; pipeEhsbTepCf com 38,39% de ganho sobre o *baseline* de visão materializada e 23,44% de perda para a opção vencedora e, por fim, a pipeEsbTepCf com 31,71% de ganho sobre o *baseline* de visão materializada e 23,44% de perda para a opção vencedora – com uma distância grande para a sétima colocada de 4,58 vezes o tempo desta. Isso significa que as técnicas de processamento em *pipeline* combinadas com a inclusão dos limitadores espaço-temporais (datas de validade e MBRs) diretamente na estrutura do índice são imprescindíveis para o bom desempenho de um índice nesta configuração de processamento. Todas as opções de processamento com perda para o *baseline* de visão materializada a tiveram com valores maiores que 60%, começando com a opção pipeTfEsbCf (68,05%) e indo até a pior colocada, a indpTepEhsbCf com 83,17% (e 92,06% de perda para a opção de processamento vencedora).

Apesar das diferenças descritas acima, quando se observa o agrupamento das opções de processamento a partir das suas respectivas classes de processamento, vê-se a seguinte ordem, do maior para o menor desempenho: *Pipeline* TE-C e ET intercalado (FTRC) se revezando nas quatro primeiras posições

em função do tipo de índice espaço-temporal usado, *Pipeline* E-T-C, *baseline* de visão materializada, *Pipeline* T-E-C e *Pipeline* E-T-C se revezando na segunda parte da tabela com as opções da classe Independente TE/C separando-os em dois blocos e Independente T/E/C na parte final da Tabela 24. O comportamento entre as classes de processamento se manteve parecido de opções espaço-temporais anteriores (como a S2-caddress, Seção 6.1.10), onde o processamento em *pipeline* se mostrou eficiente para superar o *baseline* de visão materializada.

Quanto à comparação das técnicas de processamento temporal (T_p contra T_f e T_{ep} contra T_{ef} – descrita com mais detalhes na configuração S0-caddress do item 6.1.2), observa-se que as opções de processamento utilizando a técnica T_e tiveram desempenho pior que as opções utilizando a técnica T_f enquanto as opções de processamento utilizando a técnica T_{ep} tiveram melhor desempenho que as opções baseadas na técnica T_{ef} . A quantidade de registros envolvidos fez com que o software de indexação bitmap levasse vantagem sobre o processamento do sistema gerenciador de banco de dados, mas, como a opção T_{ep} já se vale da limitação espacial feita a priori para diminuir o número de chaves na resposta, ela ainda conseguiu ser mais vantajosa que sua correspondente baseada no software de indexação bitmap.

Na comparação das opções de processamento que utilizam índices base espaço-temporais (STB-index e HSTB-index), tivemos as opções baseadas no STB-index como vencedoras sobre as suas opções relativas baseadas no HSTB-index quando consideradas as opções de processamento que utilizam a técnica de *pipeline*. A situação se inverte nas opções de processamento que utilizam a técnica independente onde cada um dos predicados sendo processados em separado nas opções utilizando o HSTB-index tiveram melhor desempenho em separado e, conseqüentemente, no resultado final. Tal fato demonstra que o nível de seletividade onde o desempenho dos índices baseados em hierarquia começa a se equiparar ao desempenho dos índices baseados em vetor – observado por Siqueira et. al. (2012) – também se aplica às opções de índices espaço-temporais utilizadas neste trabalho para esta configuração espaço-temporal.

Tabela 24: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S3-address

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/perda sobre baseline	Porc perda para o vencedor
pipeTEstb	Pipeline TE-C	STB	860654196,60	0,8607	52,834	0,000
intmTEstb	ET Intercalado (FTRC)	STB	884834317,20	0,8848	51,509	2,733
intmTEhstb	ET Intercalado (FTRC)	HSTB	1015374748,68	1,0154	44,355	15,238
pipeTEhstb	Pipeline TE-C	HSTB	1036626854,48	1,0366	43,191	16,976
pipeEhsbTepCf	Pipeline E-T-C	HSB	1124197786,92	1,1242	38,392	23,443
pipeEsbTepCf	Pipeline E-T-C	SB	1246090608,24	1,2461	31,712	30,932
VMB	VM básica	N/A	1824751880,00	1,8248	0,000	52,834
pipeTfEsbCf	Pipeline T-E-C	SB	5711813502,24	5,7118	-68,053	84,932
indpTEhstb	Independente TE/C	HSTB	5759925678,84	5,7599	-68,320	85,058
indpTEstb	Independente TE/C	STB	5873524399,16	5,8735	-68,933	85,347
pipeEhsbTefCf	Pipeline E-T-C	HSB	5917534993,32	5,9175	-69,164	85,456
pipeTfEhsbCf	Pipeline T-E-C	HSB	6249458462,20	6,2495	-70,801	86,228
pipeTpEsbCf	Pipeline T-E-C	SB	6608888647,80	6,6089	-72,389	86,977
pipeEsbTefCf	Pipeline E-T-C	SB	7184408572,36	7,1844	-74,601	88,021
pipeTpEhsbCf	Pipeline T-E-C	HSB	8980406635,24	8,9804	-79,681	90,416
indpTpEhsbCf	Independente T/E/C	HSB	10368691782,76	10,3687	-82,401	91,699
indpTfEsbCf	Independente T/E/C	SB	10607368299,92	10,6074	-82,797	91,886
indpTpEsbCf	Independente T/E/C	SB	10665376351,08	10,6654	-82,891	91,930
indpTfEhsbCf	Independente T/E/C	HSB	10841946957,00	10,8420	-83,170	92,062

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

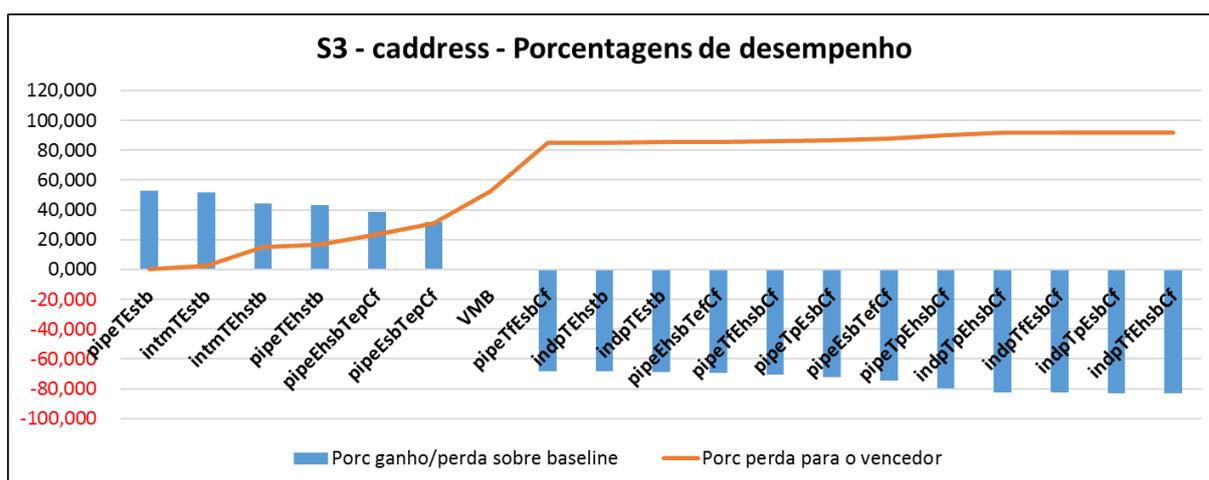


Figura 50: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S3-address

Abaixo das opções de processamento vencedoras, a comparação entre os índices base continuou mostrando que opções de processamento baseadas no

HSB-index (hierárquico) tiveram vantagem sobre as suas opções relativas utilizando o SB-index (sequencial) como índice base. Exceção feita às opções indpTfEsbCf e indpTfEhsbCf, os mesmos motivos e apresentados na opção espaço-temporal S0-address (Seção 6.1.2) se aplicam a esta opção espaço-temporal.

6.1.15 Análise da Configuração S3-address

Como pode ser observado na Tabela 25, a opção de processamento vencedora para a granularidade alta (S3) com pontos vindas de fornecedores (saddress) foi a pipeTEhstb com 63,36% de ganho em relação ao *baseline* de visão materializada. A segunda melhor opção de processamento foi a intmTEhstb obtendo média de ganhos de 62,27% sobre o *baseline* de visão materializada e de perda de apenas 2,90% em relação à opção de processamento vencedora. O pior desempenho ficou com a opção indpTfEsbCf cujas perdas ficaram em torno de 80,44% sobre o *baseline* de visão materializada e 92,83% sobre a opção de processamento vencedora. Além das duas melhores opções de processamento citadas acima, outras 6 conseguiram ganho de desempenho superior ao *baseline* de visão materializada variando de 61,09% (pipeEhsbTepCf) a 40,08% (pipeEsbTepCf). Todas as opções de processamento que foram ultrapassadas pelo desempenho do *baseline* de visão materializada obtiveram perda de desempenho variando entre 56,72% (pipeEhsbTefCf) e 80,44% (indpTfEsbCf).

O agrupamento das opções de processamento nas suas respectivas classes resultou na seguinte ordem (do melhor para o pior desempenho): *Pipeline* TE-C, ET Intercalado (FTRC) e *Pipeline* E-T-C alternando-se nas primeiras posições, *Pipeline* T-E-C, *baseline* de visão materializada, Independente TE/C e Independente T/E/C. Nota-se uma mistura de classes entre as opções de processamento que obtiveram perdas contra o *baseline* de visão materializada. A presença do *baseline* de visão materializada separando quase na metade as opções de processamento e com desempenho inferior à maioria das classes que utilizam o processamento em *pipeline* demonstra que, para esta opção espaço-temporal, o processamento em *pipeline* é uma boa opção para garantir um desempenho superior sendo, desta forma, fundamental para alcançarmos um ganho de desempenho significativo sobre a tecnologia atual de processamento em DW nesta configuração espaço-temporal. Também pode-se observar que o processamento extra do predicado convencional

requerido pelas opções independentes as colocam nas últimas posições da Tabela 25 com pronunciada separação.

Tabela 25: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S3-address

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/perda sobre baseline	Porc perda para o vencedor
pipeTEhstb	Pipeline TE-C	HSTB	798977976,32	0,7990	63,360	0,000
intmTEhstb	ET Intercalado (FTRC)	HSTB	822842460,80	0,8228	62,266	2,900
pipeEhstbTepCf	Pipeline E-T-C	HSB	848479070,00	0,8485	61,090	5,834
intmTEstb	ET Intercalado (FTRC)	STB	953848648,56	0,9539	56,258	16,236
pipeTEstb	Pipeline TE-C	STB	956952472,84	0,9570	56,116	16,508
pipeEsbTepCf	Pipeline E-T-C	SB	960315609,56	0,9603	55,962	16,800
pipeTpEhstbCf	Pipeline T-E-C	HSB	1246163858,36	1,2462	42,853	35,885
pipeTpEsbCf	Pipeline T-E-C	SB	1306635049,96	1,3066	40,080	38,852
VMB	VM básica	N/A	2180637160,00	2,1806	0,000	63,360
pipeEhstbTefCf	Pipeline E-T-C	HSB	5038735877,32	5,0387	-56,723	84,143
indpTEhstb	Independente TE/C	HSTB	5841626966,44	5,8416	-62,671	86,323
indpTpEhstbCf	Independente T/E/C	HSB	5953803521,52	5,9538	-63,374	86,580
pipeTfEhstbCf	Pipeline T-E-C	HSB	6011983418,00	6,0120	-63,728	86,710
indpTEstb	Independente TE/C	STB	6044917126,52	6,0449	-63,926	86,783
indpTpEsbCf	Independente T/E/C	SB	6093079364,56	6,0931	-64,211	86,887
pipeTfEsbCf	Pipeline T-E-C	SB	6128380008,88	6,1284	-64,417	86,963
pipeEsbTefCf	Pipeline E-T-C	SB	9511720024,00	9,5117	-77,074	91,600
indpTfEhstbCf	Independente T/E/C	HSB	9778584860,96	9,7786	-77,700	91,829
indpTfEsbCf	Independente T/E/C	SB	11150069091,04	11,1501	-80,443	92,834

Cores usadas

- ET Intercalado (FTRC)
- Pipeline TE-C
- Pipeline T-E-C
- Pipeline E-T-C
- Independente T/E/C
- Independente TE/C
- STB-index
- HSTB-index
- SB-index
- HSB-index

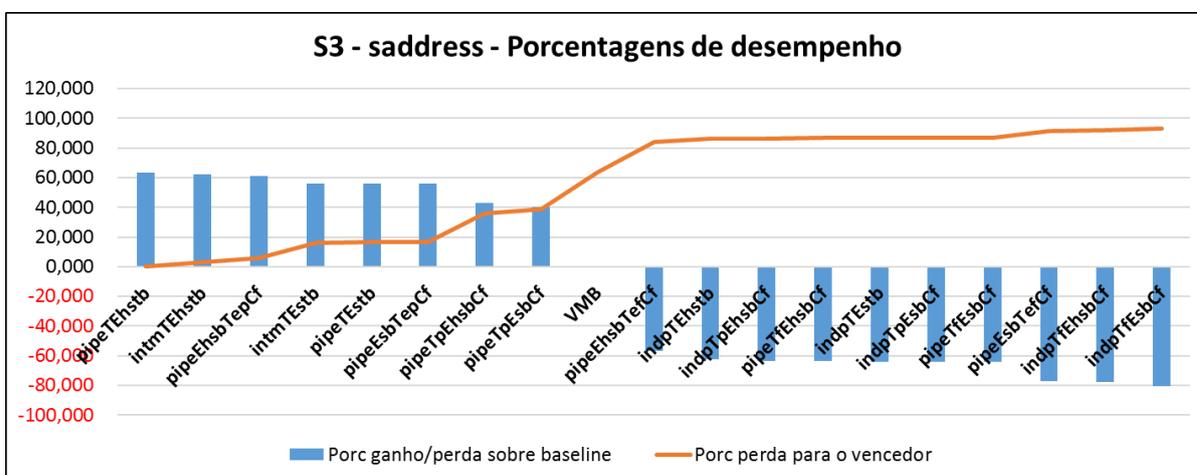


Figura 51: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S3-address

Na comparação entre as técnicas de processamento puramente temporais (Tp contra Tf e Tep contra Tef), notamos o mesmo comportamento das opções baseadas em ponto da seletividade muito baixa (S0-caddress e S0-saddress, Seções 6.1.2 e 6.1.3 respectivamente) onde as opções de processamento utilizando as técnicas Tp e Tep obtiveram melhor desempenho que suas opções de processamento relativas utilizando as técnicas Tf e Tef.

Quando comparamos as três melhores opções de processamento (com perda de desempenho menor do que 15% para a opção vencedora), observamos que as opções que utilizam como índice base o HSTB-index prevaleceram, mesmo que fracamente, sobre a opção que utiliza o HSB-index como índice base. A diferença entre elas está na inclusão de ambas as informações espacial e temporal na estrutura do índice. A proximidade das opções revelou que nesta seletividade alta já se começa a notar uma condição desfavorável para a poda de registros nas opções que utilizaram o HSTB-index como base vindo a privilegiar as opções baseadas no STB-index pela simplicidade de navegação nos registros, como era esperado.

Quando comparamos o desempenho das opções de processamento levando em conta seus índices base, vemos que o comportamento se manteve no mesmo padrão das opções espaço-temporais baseadas em ponto (S0-caddress e S0-saddress, itens 6.1.2 e 6.1.3) onde o desempenho das opções que usaram o SB-index como índice base tiveram desempenho pior quando comparados à sua correspondente utilizando o HSB-index como base. Pode-se notar que as diferenças de tempos diminuíram, mas esperava-se que, a partir desta seletividade, o reduzido número de podas dos registros já fizesse os índices baseados em vetores terem desempenho comparável ou superior aos baseados em hierarquias de MBR. Uma provável explicação está no tempo de processamento do predicado temporal que pode ter influenciado os tempos médios a ponto de mascarar possíveis mudanças nos tempos de processamento do predicado espacial.

6.1.16 Análise da Configuração S3-street

Esta configuração para linhas de ruas (street) com seletividade alta (S3) apresentou a opção pipeTEhstb como vencedora com ganho de desempenho de 16,15% em relação ao *baseline* de visão materializada na sua forma básica (sem índices). A opção que ficou com o segundo melhor desempenho foi a intrmTEhstb

com 11,75% de ganho de desempenho sobre o *baseline* de visão materializada e 4,98% de perda para a opção vencedora. A seguir, já temos o *baseline* de visão materializada na sua versão básica com perda de 16,15% para a opção de processamento vencedora. Outras opções de processamento obtiveram perdas de desempenho frente ao *baseline* de visão materializada com um desempenho muito próximo desta (menos de 15% de perda) e são dignas de menção: pipeTEhstb (com 8,18% de perda) e intmTEstb (com 8,64% de perda). Olhando para a parte de baixo da Tabela 26, vemos opções e processamento que obtiveram perda de desempenho maior que 40% frente ao *baseline* de visão materializada começando pela opção pipeEhsbTefCf (41,94% de perda) e indo até a opção com o pior desempenho, que foi a pipeEsbTefCf (65,24% de perda). Pode-se notar também um “achatamento” no gráfico da Figura 52, uma vez que esta foi a configuração que menos variou em termos de ganhos de desempenho máximo (16,15% da opção vencedora) e de perdas máxima (65,29% da pior opção) e, em consequência disso, tivemos um crescimento mais constante da perda de desempenho das opções de processamento quando comparadas à opção de processamento vencedora.

Quando se observa o agrupamento das opções de processamento a partir das suas respectivas classes de processamento, vê-se a seguinte ordem do maior para o menor desempenho: *Pipeline* TE-C e ET intercalado (FTRC) se revezando nas quatro primeiras posições e separadas exatamente ao meio pelo *baseline* de visão materializada, *Pipeline* E-T-C, *Pipeline* T-E-C, Independente TE/C e Independente T/E/C. As classes que obtiveram perda de desempenho contra o *baseline* materializada estão se alternando nas posições da Tabela 26 sem um comportamento visível entre elas. A presença das classes *Pipeline* TE-C e ET-intercalado (FTRC) como vencedoras vem mostrar que, além da utilização da técnica de processamento em *pipeline*, a ajuda dada pela construção do índice base já contendo as informações para limitação espacial e temporal na estrutura do índice foi fundamental para proporcionar um desempenho melhor que o *baseline* de visão materializada. Vale ressaltar aqui que o *baseline* de visão materializada tinha sido a grande vencedora para este tipo de dados nas seletividades mais baixas (S0-street, S1-street e S2-street) com a sua versão avançada, mas esta perdeu força nesta seletividade inclusive nos outros tipos de dados espaciais.

Tabela 26: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora (melhor desempenho) – S3-street

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/perda sobre baseline	Porc perda para o vencedor
pipeTEhstb	Pipeline TE-C	HSTB	5190745026,44	5,1908	16,146	0,000
intmTEhstb	ET Intercalado (FTRC)	HSTB	5462983501,44	5,4630	11,748	4,983
VMB	VM básica	N/A	6190183000,00	6,1902	0,000	16,146
pipeTEstb	Pipeline TE-C	STB	6742018843,12	6,7420	-8,185	23,009
intmTEstb	ET Intercalado (FTRC)	STB	6775413019,48	6,7754	-8,638	23,389
pipeEhstbTepCf	Pipeline E-T-C	HSB	7956390998,96	7,9564	-22,199	34,760
pipeTpEsbCf	Pipeline T-E-C	SB	8031081231,92	8,0311	-22,922	35,367
indpTEhstb	Independente TE/C	HSTB	8824635163,52	8,8246	-29,853	41,179
indpTEstb	Independente TE/C	STB	9317418841,92	9,3174	-33,563	44,290
indpTpEhstbCf	Independente T/E/C	HSB	10130959374,88	10,1310	-38,898	48,764
pipeEhstbTefCf	Pipeline E-T-C	HSB	10661539366,04	10,6615	-41,939	51,313
pipeTfEsbCf	Pipeline T-E-C	SB	11284641582,88	11,2846	-45,145	54,002
indpTpEsbCf	Independente T/E/C	SB	11785775500,08	11,7858	-47,478	55,958
pipeTfEhstbCf	Pipeline T-E-C	HSB	12091698910,68	12,0917	-48,806	57,072
indpTfEhstbCf	Independente T/E/C	HSB	13873729790,92	13,8737	-55,382	62,586
pipeTpEhstbCf	Pipeline T-E-C	HSB	14121157946,40	14,1212	-56,164	63,241
indpTfEsbCf	Independente T/E/C	SB	14899268472,80	14,8993	-58,453	65,161
pipeEsbTepCf	Pipeline E-T-C	SB	17310750535,84	17,3108	-64,241	70,014
pipeEsbTefCf	Pipeline E-T-C	SB	17834166002,00	17,8342	-65,290	70,894

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

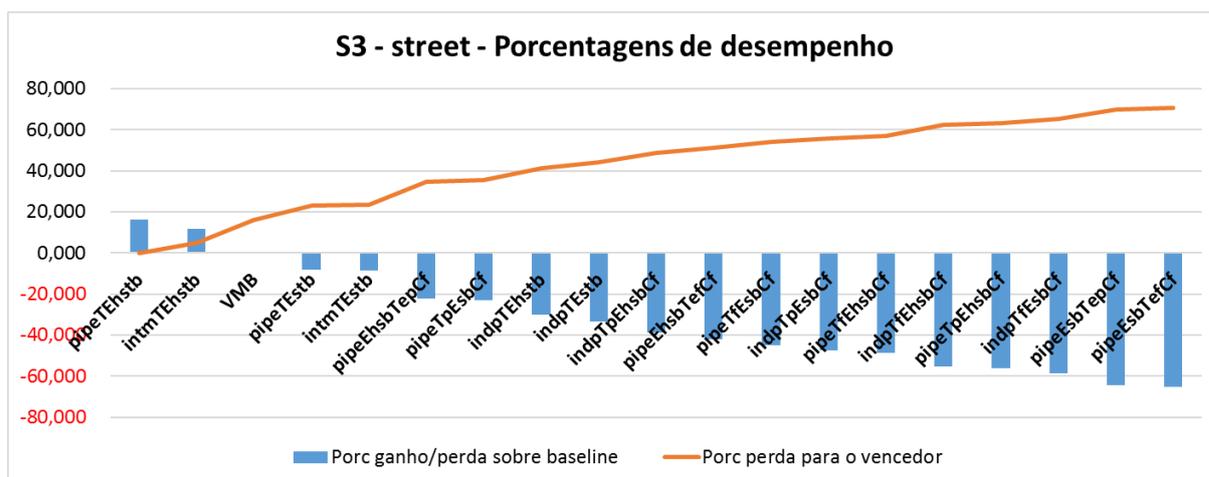


Figura 52: Desempenho das opções de processamento com relação ao *baseline* de visão materializada e com relação à opção de processamento vencedora – S3-street

Quanto à comparação das técnicas de processamento temporal (Tp contra Tf e Tep contra Tef – descrita com mais detalhes na opção espaço-temporal S0-

caddress, Seção 6.1.2), observa-se que todas as opções de processamento utilizando a técnica Tep (interface com o banco de dados após limitação temporal) superaram as suas opções relativas utilizando a técnica Tef (índice bitmap após limitação temporal). Porém, o mesmo não ocorreu em relação às técnicas Tp e Tf, onde as opções de processamento pipeTpEsbCf superou sua correspondente utilizando a técnica Tf (pipeTfEsbCf), com exceção da opção pipeTpEhsbCf que obteve desempenho aquém da sua respectiva opção utilizando a técnica Tf (pipeTfEhsbCf).

Dentre as opções de processamento que ficaram com perda de desempenho abaixo de 30% com relação à opção de processamento vencedora, podemos listar as seguintes opções: pipeTEhstb, intmTEhstb, pipeTEstb e intmTEstb. Mais uma vez vemos as opções baseadas no HSTB-index obtendo melhor desempenho com relação às suas equivalentes utilizando o STB-index, o que não era esperado para seletividades maiores pois temos menos poda de candidatos, que é a principal vantagem desse tipo de índice. A ordenação entre as opções de processamento não variou deixando as opções baseadas no HSTB-index como vencedoras sobre as suas opções relativas baseadas no STB-index tanto para as opções de processamento que utilizam a técnica de *pipeline* quanto para as que usam a técnica independente.

Abaixo das opções de processamento vencedoras, a comparação entre os índices base mostrou mais opções de processamento baseadas no HSB-index (hierárquico) tendo pouca vantagem ou perda sobre as suas opções relativas utilizando o SB-index (sequencial) como índice base (pipeTpEsbCf com relação à opção pipeTpEhsbCf, por exemplo). Tal situação segue o comportamento esperado e a quantidade de registros diminuir a importância da poda destes a priori.

6.1.17 Análise da Configuração S3-city

Esta configuração espaço-temporal para polígonos de cidades (city) com seletividade alta (S3) apresentou resultados muito semelhantes à configuração S2-city no que tange a separação de opções de processamento pelo *baseline* de visão materializada (exatamente na metade) e na distribuição das classes de processamento. A opção de processamento vencedora foi a pipeTEhstb com 64,87% de ganho sobre o *baseline* de visão materializada em sua versão básica (sem índices). Da segunda até a oitava posições, as opções de processamento

estão muito próximas com perda máxima de desempenho para a opção vencedora de 11,78%. A opção de processamento com pior desempenho foi a pipeEsbTefCf com 58,05% de perda para o *baseline* de visão materializada e de 85,26% para a opção vencedora.

Outra observação interessante é a presença de níveis bem diferenciados de desempenho com relação à opção de processamento vencedora (pipeTEhstb). Um primeiro grupo pode ser formado pelas opções que obtiveram menos de 15% de perda para a opção de processamento vencedora, a saber: intmTEhstb (0,59%), pipeTpEhsbCf (0,88%), pipeTfEhsbCf (1,28%), pipeEhsbTepCf (2,55%), intmTEstb (9,82%), pipeTESstb (10,84%) e pipeTpEsbCf (11,78%). O segundo grupo compreende a transição com a opção pipeEsbTepCf (30,62) e o terceiro possui todas as opções de processamento restante iniciando com o *baseline* de visão materializada (64,87%) até a pior opção com 85,26% de perda.

Tabela 27: Comparação das opções de processamento com relação ao *baseline* de visão materializada e à opção de processamento vencedora – S3-city

Opção	Classe	Índice base	Tempo médio da opção (ns)	Tempo médio da opção (seg)	Porc ganho/per da sobre baseline	Porc perda para o vencedor
pipeTEhstb	Pipeline TE-C	HSTB	2002548585,32	2,0026	64,874	0,000
intmTEhstb	ET Intercalado (FTRC)	HSTB	2014377658,60	2,0144	64,666	0,587
pipeTpEhsbCf	Pipeline T-E-C	HSB	2020258186,52	2,0203	64,563	0,877
pipeTfEhsbCf	Pipeline T-E-C	HSB	2028592393,60	2,0286	64,417	1,284
pipeEhsbTepCf	Pipeline E-T-C	HSB	2054857786,44	2,0549	63,956	2,546
intmTEstb	ET Intercalado (FTRC)	STB	2220697291,16	2,2207	61,047	9,823
pipeTEstb	Pipeline TE-C	STB	2246065757,12	2,2461	60,602	10,842
pipeTpEsbCf	Pipeline T-E-C	SB	2269850034,60	2,2699	60,185	11,776
pipeEsbTepCf	Pipeline E-T-C	SB	2886266152,20	2,8863	49,372	30,618
VMB	VM básica	N/A	5700966520,00	5,7010	0,000	64,874
indpTpEhsbCf	Independente T/E/C	HSB	6484859612,48	6,4849	-12,088	69,120
pipeEhsbTefCf	Pipeline E-T-C	HSB	6688694018,60	6,6887	-14,767	70,061
indpTEhstb	Independente TE/C	HSTB	6887596798,72	6,8876	-17,229	70,925
indpTEstb	Independente TE/C	STB	7293205066,64	7,2932	-21,832	72,542
indpTpEsbCf	Independente T/E/C	SB	7310130893,88	7,3101	-22,013	72,606
pipeTfEsbCf	Pipeline T-E-C	SB	7348197676,96	7,3482	-22,417	72,748
indpTfEhsbCf	Independente T/E/C	HSB	11778504753,16	11,7785	-51,599	82,998
indpTfEsbCf	Independente T/E/C	SB	11821242615,88	11,8212	-51,774	83,060
pipeEsbTefCf	Pipeline E-T-C	SB	13588738346,36	13,5887	-58,046	85,263

Cores usadas

- ET Intercalado (FTRC)
- Pipeline TE-C
- Pipeline T-E-C
- Pipeline E-T-C
- Independente T/E/C
- Independente TE/C



- STB-index
- HSTB-index
- SB-index
- HSB-index



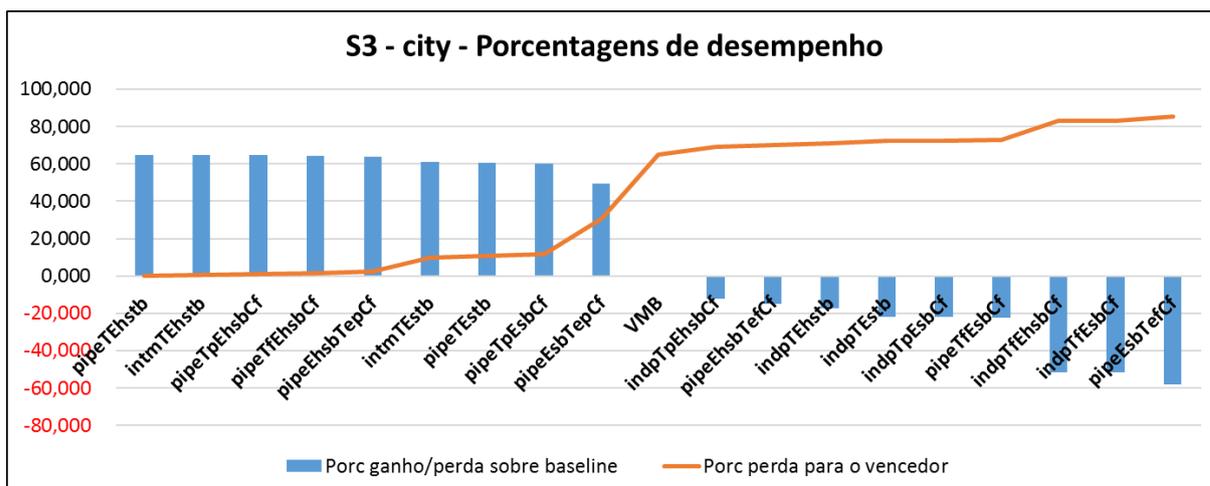


Figura 53: Desempenho das opções de processamento com relação ao *baseline* de opção materializada e com relação à opção de processamento vencedora – S3-city

Quando se observa o agrupamento das opções de processamento a partir das suas respectivas classes de processamento, vê-se a seguinte ordem do maior para o menor desempenho: *Pipeline* TE-C, ET intercalado (FTRC), *Pipeline* T-E-C e *Pipeline* E-T-C se revezando nas primeiras posições, *baseline* de visão materializada, Independente T/E/C e Independente TE/C se revezando na parte final da Tabela 27. Tal ordenação de classes nos mostra que nesta configuração o processamento em *pipeline* já é suficiente para proporcionar um desempenho superior ao *baseline* de visão materializada. Vemos mais uma vez as opções de processamento baseadas nas classes independentes apresentando desempenhos inferiores à demais opções devido ao processamento duplicado da parte convencional.

Quanto à comparação das técnicas de processamento temporal (Tp contra Tf e Tep contra Tef – descrita com mais detalhes na opção espaço-temporal S0-caddress), observa-se que todas as opções de processamento utilizando a interface com o banco de dados (Tp e Tep) levaram vantagem sobre suas opções de processamento relativas utilizando o software de indexação bitmap (Tf e Tef). Tal comportamento não era o esperado devido ao aumento da seletividade, mas pode ser explicado pelo número de registros da tabela “city” que possui 1399 registros, enquanto a tabela “street” possui 42594 registros e, portanto, é mais difícil de processar.

Já a ordenação entre as opções de processamento não variou e ainda se observa que as opções baseadas no HSTB-index sobrepujaram as suas opções

relativas baseadas no STB-index em todas as classes. Com relação à comparação entre os índices base continuou mostrando que opções de processamento baseadas no HSB-index (hierárquico) tiveram vantagem sobre as suas opções relativas utilizando o SB-index (sequencial) como índice base. Os mesmos motivos apresentados na opção espaço-temporal S0-caddress (item 6.1.2) se aplicam a esta opção espaço-temporal.

6.1.18 Conclusões

A análise pormenorizada das configurações mostrou que o uso de índices especializados é imprescindível para se obter um melhor desempenho na execução de consultas OLAP espaço-temporais. Ela também mostrou que uma maior “poda” de registros nos estágios iniciais da consulta melhora o desempenho de um índice consideravelmente principalmente quando é utilizada a técnica de processamento em *pipeline*, onde somente as chaves dos registros candidatos à resposta de um predicado anterior são utilizadas no processamento do predicado posterior. Outro fator que influenciou positivamente foi a inclusão de ambas as informações de verificação espaço-temporal diretamente na estrutura do índice (o vetor ou hierarquia de MBRs em conjunto com as datas de validade inicial e final deste). Nas seletividades mais altas houveram casos onde somente a combinação da técnica em *pipeline* em conjunto com a indexação das informações espaciais e temporais conseguiram ganho de desempenho sobre o *baseline* de visão materializada, mas tal combinação se mostrou a melhor pelas opções vencedoras em todas as configurações.

Passaremos, a seguir para as análises das opções de processamento comparando o comportamento das classes nas diversas seletividades.

6.2 Análise da Influência da Seletividade no Desempenho Relativo das Técnicas de Processamento

As análises a seguir tentarão esclarecer como foi o comportamento das opções de processamento nas várias seletividades para encontrar possíveis tendências que possam ajudar em futuras implementações.

6.2.1 Objetivo da Análise

Como o próprio título desta seção sugere, o objetivo desta análise é identificar possíveis tendências entre as opções de processamento, suas classes e características, para identificar se a variação da seletividade possui influência nos resultados obtidos e analisar se esta influência foi positiva ou negativa de modo a evita-las ou atingi-las nas várias situações de consulta a serem enfrentadas. Para tanto, fixado um determinado tipo de dado, comparou-se o desempenho relativo das opções nas diferentes seletividades.

6.2.2 Análise das Configurações Envolvendo Pontos de Clientes (caddress)

Observando-se o padrão de cores obtidos a partir das opções de processamento e suas respectivas classes pode-se observar a formação de quatro regiões horizontais (Z1, Z2, Z3 e Z4) que chamaremos de zonas de variação.

A primeira delas (Z1) nos mostra que as opções de processamento baseadas nas classes ET Intercalado (FTRC), *Pipeline* TE-C e uma das opções da classe *Pipeline* E-T-C foram as que obtiveram os melhores desempenhos em todas as seletividades. Uma característica em comum destas classes é o processamento em *pipeline*. Mais duas características explicam os resultados encontrados: a utilização de hierarquias (para poda a priori de registros que, com certeza, não fazem parte da solução) e a particularidade na construção das seletividades (com a fixação de uma seletividade temporal alta e variação da janela de consulta espacial até atingir-se a seletividade espaço-temporal desejada). A primeira ajudou as opções de processamento baseadas no HSTB-index e HSB-index a obterem os melhores

desempenhos dentro das seletividades mais baixas (S0, S1 e S2) e tal desempenho das opções que a utilizaram só foi ultrapassado na seletividade S3 – a qual ultrapassa a seletividade observada por Siqueira et. al. (2012) onde a poda de registros não era mais determinante para um melhor desempenho do HSB-index sobre o SB-index. Comprova-se assim que tal seletividade se verifica também para os índices espaço-temporais estudados neste trabalho. A segunda característica (da forma como foi alcançada a seletividade espaço-temporal) auxiliou a opção de processamento *pipeEhsbTepCf* a limitar mais registros nas fases iniciais do processamento dos predicados fazendo-a se sobressair sobre as demais da mesma classe e sempre ficar entre as melhores opções de processamento. Curiosamente, esperava-se que as opções intercaladas também se valessem dessa propriedade e, assim, conseguissem sempre um melhor desempenho sobre suas opções relativas em *pipeline*. Isso se mostrou verdade somente na seletividade S2, indicando que a poda de registros dos índices que utilizaram técnicas hierárquicas foi o principal fator determinando um melhor ou pior desempenho. Também digno de menção nesta zona de variação, temos o *baseline* de visão materializada na sua versão avançada na segunda colocação para a seletividade muito baixa (S0). Isso demonstra o avanço das técnicas de indexação já presentes no PostgreSQL as quais, quando auxiliadas corretamente pelo otimizador de consultas, produzem resultados interessantes rivalizando com índices mais especializados de processamento de consultas STOLAP.

Na zona de variação Z2 temos opções de processamento das classes *Pipeline* T-E-C e *Pipeline* E-T-C sem predominância de uma técnica de processamento específica. Vê-se também a presença do *baseline* de visão materializada nas seletividades S1, S2 e S3 e de opções das classes independentes nas seletividades S2 e S3 aumentando a diversidade das opções de processamento nesta zona de variação. As perdas para a opção vencedora são também dignas de nota por formarem uma fase de transição para as opções de processamento com os piores desempenhos (entre 26,37% a 96,25% de perda para a opção vencedora).

Tabela 28: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para address com todas as seletividades

	S0			S1			S2			S3			
	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	
Z1	pipeTEhstb	HSTB	0,000	pipeTEhstb	HSTB	0,000	intmTEhstb	HSTB	0,000	pipeTEstb	STB	0,000	melhor ↓ pior
	VMA	BTree	2,646	intmTEhstb	HSTB	9,582	pipeTEhstb	HSTB	0,749	intmTEstb	STB	2,733	
	intmTEhstb	HSTB	10,148	pipeEhstbTepCf	HSB	10,882	pipeTEstb	STB	4,416	intmTEhstb	HSTB	15,238	
	pipeEhstbTepCf	HSB	33,779	intmTEstb	STB	32,856	intmTEstb	STB	4,721	pipeTEhstb	HSTB	16,976	
	intmTEstb	STB	55,376	pipeTEstb	STB	33,179	pipeEhstbTepCf	HSB	12,136	pipeEhstbTepCf	HSB	23,443	
Z2	pipeTEstb	STB	55,721	pipeEsbTepCf	SB	39,088	pipeEsbTepCf	SB	26,374	pipeEsbTepCf	SB	30,932	
	pipeEsbTepCf	SB	60,936	VMA	BTree	43,452	VMB	N/A	35,608	VMB	N/A	52,834	
	pipeTpEsbCf	SB	94,150	pipeTpEsbCf	SB	93,232	indpTEhstb	HSTB	89,116	pipeTTEsbCf	SB	84,932	
	pipeEhstbTefCf	HSB	96,127	pipeEhstbTefCf	HSB	93,280	pipeTTEhstbCf	HSB	89,527	indpTEhstb	HSTB	85,058	
	pipeTTEhstbCf	HSB	96,253	pipeTTEsbCf	SB	93,373	pipeTTEsbCf	SB	89,548	indpTEstb	STB	85,347	
Z3	indpTEhstb	HSTB	96,401	indpTEhstb	HSTB	93,438	indpTEstb	STB	89,581	pipeEhstbTefCf	HSB	85,456	
	pipeTTEsbCf	SB	96,407	pipeTTEhstbCf	HSB	93,532	pipeEhstbTefCf	HSB	89,588	pipeTTEhstbCf	HSB	86,228	
	indpTEstb	STB	96,540	indpTEstb	STB	93,651	pipeTpEsbCf	SB	90,549	pipeTpEsbCf	SB	86,977	
	pipeTpEhstbCf	HSB	96,708	pipeTpEhstbCf	HSB	93,983	pipeTpEhstbCf	HSB	91,685	pipeEsbTefCf	SB	88,021	
	pipeEsbTefCf	SB	97,086	pipeEsbTefCf	SB	95,108	pipeEsbTefCf	SB	91,866	pipeTpEhstbCf	HSB	90,416	
Z4	indpTpEsbCf	SB	97,185	indpTpEhstbCf	HSB	95,922	indpTpEhstbCf	HSB	93,486	indpTpEhstbCf	HSB	91,699	
	indpTpEhstbCf	HSB	97,439	indpTpEsbCf	SB	96,490	indpTpEsbCf	SB	94,048	indpTTEsbCf	SB	91,886	
	indpTTEsbCf	SB	97,973	indpTTEhstbCf	HSB	96,558	indpTTEsbCf	SB	94,324	indpTpEsbCf	SB	91,930	
	indpTTEhstbCf	HSB	98,086	indpTTEsbCf	SB	96,570	indpTTEhstbCf	HSB	94,545	indpTTEhstbCf	HSB	92,062	

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

A zona de variação Z3 possui predominantemente as opções das classes Independente TE/C, Pipeline T-E-C e Pipeline E-T-C onde nota-se as opções de processamento da classe independente melhorando o seu desempenho gradativamente a cada aumento de seletividade chegando a mudarem para a zona de variação Z2 na seletividade S3. As opções de processamento que utilizam o conceito de *pipeline* e que aparecem predominantemente nesta zona de variação são baseadas no tipo de processamento Tf e Tef que envolve consulta ao software de processamento de índices bitmap e leitura de resultados em disco, que torna a operação mais custosa que as demais. A variação entre elas é grande quando comparadas entre seletividades e nenhum padrão aparente (além das classes de processamento) pode ser encontrado.

Z4 é a última zona de variação e contém, curiosamente, somente opções de processamento da classe Independente T/E/C. Tal comportamento se explica pela necessidade de duplicação do processamento da parte convencional. Já tendo perdas de desempenho variando entre 91,70% e 98,09%, não é possível observar nenhum outro aspecto do comportamento das opções de processamento dentro da classe.

Concluindo, verifica-se que o aumento da seletividade não afetou o conjunto das opções de processamento vencedoras, no caso as opções pertencentes às classes *Pipeline* TE-C e ET Intercalado (FTRC). Verifica-se também que as opções de processamento baseadas no HSTB-index tendem a ter um desempenho melhor nas seletividades mais baixas (S0, S1 e S2) mas, a partir da seletividade S3, as opções de processamento baseadas no STB-index já devem ser escolhidas. Outro ponto a destacar é o comportamento do *baseline* de visão materializada. Na sua versão avançada (altamente indexada) e seletividade S0, ele se mostrou uma boa opção de processamento, mas, com o aumento da seletividade, seu desempenho piorou e o colocou na segunda zona de variação. Além disso, a porcentagem de perda do *baseline* de visão materializada para a opção de processamento vencedora aumentou entre as seletividades S2 e S3, e, baseando-se na tecnologia característica de tal processamento, a tendência é continuar piorando seu desempenho. Assim, conclui-se que o uso do *baseline* de visão materializada é inadequado para maiores valores de seletividades e a implementação das outras opções de processamento, especialmente das classes ET intercalado (FTRC) e *pipeline* TE-C, é imprescindível para se obter um bom desempenho no processamento de consultas STOLAP.

6.2.3 Análise das Configurações Envolvendo Pontos de Fornecedores (saddress)

Segue-se nesta análise a mesma divisão em zonas de variação feita na análise anterior (Seção 6.2.2).

A exemplo da análise feita para o tipo de dados ponto vindo da tabela de clientes (item 6.2.2), a zona de variação Z1 nos mostra que as opções de processamento baseadas nas classes ET Intercalado (FTRC), *Pipeline* TE-C e, curiosamente, uma das opções de processamento da classe *Pipeline* E-T-C foram as que obtiveram os melhores desempenhos em todas as seletividades. Apesar de curiosa, a presença da opção de processamento *pipeEhstbTepCf* explica-se pela limitação espacial feita a priori (privilegiada em alguns casos pela maneira de formação das seletividades – temporal alta para depois ajustar a janela de consulta), utilização do HSB-index como índice base (hierárquico) e limitação temporal mais rápida pela técnica Tep (como mostrado nas análises pormenorizadas, elas levam

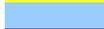
vantagem na maioria dos casos contra sua relativa baseada no índice bitmap – a Tef). Todas essas características somadas ao processamento em *pipeline* e ao menor número de registros da tabela de fornecedores (“supplier”) fizeram com que a opção obtivesse um desempenho muito próximo das opções de processamento baseadas nos índices HSTB-index e STB-index (já espaço-temporais). Outro destaque fica para as opções de processamento pipeTEhstb (que foi a vencedora em três seletividades e segunda melhor opção na seletividade S0) e intmTEhstb (que foi vencedora na seletividade S0 e a segunda colocada nas demais seletividades) que estiveram sempre muito próximas e se revezaram nas primeiras colocações. Ao contrário da análise tipo de dados ponto vindo da tabela de clientes (item 6.2.2), as opções de processamento baseadas no STB-index não ultrapassaram em desempenho as opções baseadas no HSTB-index mesmo na seletividade S3. Uma possível explicação para isso é a quantidade reduzida de linhas da tabela de fornecedores em relação à tabela de clientes (“supplier” com 7.375 linhas e “customer” com 95.967) o que também diminuiu o tamanho dos seus índices e fez com que a navegação e poda na árvore ainda se mostrasse melhor que o vetor de MBRs. À exceção da opção pipeEhstbTepCf (cujo comportamento já foi explicado acima), o processamento em *pipeline* aliado à estrutura espaço-temporal interna ao índice faz com que as classes de processamento utilizando tais características tenham tamanho sucesso.

Na zona de variação Z2 temos opções de processamento das classes *Pipeline* T-E-C, *Pipeline* E-T-C e o *baseline* de visão materializada (em três das seletividades) se revezando nas posições com a interferência de uma das opções de processamento da classe *Pipeline* TE-C na seletividade S1. É interessante notar que a porcentagem de perda para a opção vencedora dá um “salto” entre esta e a próxima zona de variação (diferença mínima de 20,78% na seletividade S3) indicando um possível ponto de ruptura no desempenho para este tipo de dado. Nada pode se afirmar ao compararmos os índices base para esta zona de variação.

Tabela 29: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para saddress com todas as seletividades

	S0			S1			S2			S3			
	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	
Z1	intmTEhstb	HSTB	0,000	pipeTEhstb	HSTB	0,000	pipeTEhstb	HSTB	0,000	pipeTEhstb	HSTB	0,000	melhor ↓ pior
	pipeTEhstb	HSTB	0,328	intmTEhstb	HSTB	0,818	intmTEhstb	HSTB	0,589	intmTEhstb	HSTB	2,900	
	pipeEhstbTepCf	HSB	8,089	pipeEhstbTepCf	HSB	7,554	pipeEhstbTepCf	HSB	12,369	pipeEhstbTepCf	HSB	5,834	
	pipeTEstb	STB	22,974	VMB	N/A	16,299	intmTEstb	STB	17,936	intmTEstb	STB	16,236	
intmTEstb	STB	23,060	intmTEstb	STB	22,131	pipeTEstb	STB	18,360	pipeTEstb	STB	16,508		
Z2	pipeEsbTepCf	SB	23,889	pipeEsbTepCf	SB	23,785	pipeEsbTepCf	SB	19,799	pipeEsbTepCf	SB	16,800	
	pipeTpEsbCf	SB	30,143	pipeTEstb	STB	24,228	VMB	N/A	36,257	pipeTpEhstbCf	HSB	35,885	
	VMB	N/A	31,297	pipeTpEhstbCf	HSB	34,190	pipeTpEsbCf	SB	36,924	pipeTpEsbCf	SB	38,852	
	pipeTpEhstbCf	HSB	42,407	pipeTpEsbCf	SB	37,082	pipeTpEhstbCf	HSB	36,943	VMB	N/A	63,360	
Z3	pipeEhstbTefCf	HSB	86,410	pipeEhstbTefCf	HSB	85,588	indpTEhstb	HSTB	84,111	pipeEhstbTefCf	HSB	84,143	
	indpTEhstbCf	HSB	86,645	indpTEhstb	HSTB	86,038	indpTEstb	STB	84,701	indpTEhstb	HSTB	86,323	
	indpTEhstb	HSTB	86,724	pipeTfEhstbCf	HSB	86,130	pipeEhstbTefCf	HSB	85,183	indpTpEhstbCf	HSB	86,580	
	pipeTfEhstbCf	HSB	86,881	indpTEstb	STB	87,004	indpTpEhstbCf	HSB	85,558	pipeTfEhstbCf	HSB	86,710	
Z4	pipeTfEsbCf	SB	87,651	pipeTfEsbCf	SB	87,364	pipeTfEhstbCf	HSB	85,859	indpTEstb	STB	86,783	
	indpTpEsbCf	SB	87,921	indpTpEsbCf	SB	87,584	pipeTfEsbCf	SB	85,983	indpTpEsbCf	SB	86,887	
	indpTEstb	STB	88,027	indpTpEhstbCf	HSB	87,649	indpTpEsbCf	SB	87,016	pipeTfEsbCf	SB	86,963	
	pipeEsbTefCf	SB	91,674	pipeEsbTefCf	SB	91,877	pipeEsbTefCf	SB	91,023	pipeEsbTefCf	SB	91,600	
	indpTfEhstbCf	HSB	92,498	indpTfEhstbCf	HSB	92,632	indpTfEhstbCf	HSB	91,952	indpTfEhstbCf	HSB	91,829	
	indpTfEsbCf	SB	93,155	indpTfEsbCf	SB	93,003	indpTfEsbCf	SB	92,066	indpTfEsbCf	SB	92,834	

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

A zona de variação Z3 possui predominantemente as opções das classes Independente T/E/C e Independente TE/C (onde cada predicado da consulta é executado separadamente e os resultados intermediários usados em processamento cosequencial de listas) que precisaram processar a parte convencional em duplicidade e, assim, obter desempenhos inferiores. Junto com elas, há a presença de algumas das opções da classe *Pipeline E-T-C* e *Pipeline T-E-C* que utilizaram as técnicas de processamento Tf e Tef e cujos desempenhos foram constantemente inferiores frente às suas técnicas relativas utilizando o sistema gerenciador de banco de dados (Tp e Tep). Mais uma vez, a porcentagem de perda de desempenho frente às opções vencedoras é muito grande (sempre maior que 84%) vindo a reforçar a importância das opções de processamento vencedoras.

A zona Z4 repete o comportamento da zona de variação anterior, mas com presença massiva das opções de processamento da classe Independente T/E/C. A distância para as opções de processamento vencedoras também é grande (mais 85% de perda) tornando indesejável o uso das opções de processamento compreendidos nesta zona de variação em trabalhos futuros quando as mesmas condições de processamento forem utilizadas.

Concluindo, verifica-se que o aumento da seletividade não afetou o conjunto de opções de processamento vencedoras, no caso a *pipeTEhstb*, *intmTEhstb*, *pipeEhsbTepCf*, *pipeTEstb* (esta com exceção no quadrante Z1-S1) e *intmTEstb*, e também não afetou de forma geral o relacionamento entre as classes. Quanto a estas, verifica-se a seguinte ordem decrescente de desempenho (melhor para pior) entre elas: *Pipeline* TE-C alternando posições com a classe E-T intercalado (FTRC) e uma das opções da classe *Pipeline* E-T-C. *Pipeline* T-E-C, *pipeline* E-T-C e processamento independente (com alternância entre processamento independente T/E/C e TE/C). Outro ponto a destacar é o desempenho do *baseline* de visão materializada que esteve na zona de variação superior (Z1) apenas uma vez, mas com perda de desempenho maior que 15% com relação à opção vencedora, e que o aumento da seletividade (quando comparamos as zonas Z2 e Z3) apontou um aumento da perda para a opção de processamento vencedora. Conclui-se, mais uma vez, que o uso do *baseline* de visão materializada é inapropriado para maiores valores de seletividades e a implementação das opções de processamento, especialmente das classes *Pipeline* TE-C e ET intercalado (FTRC), é imprescindível para se obter um bom desempenho no processamento de consultas STOLAP.

6.2.4 Análise das Configurações Envolvendo Linhas de Ruas (street)

Segue-se nesta análise a mesma divisão em zonas de variação feita na análise do item 6.2.2.

Surpresa neste tipo de dado linha vindo da tabela de ruas (*street*), o *baseline* de visão materializada na sua versão avançada (altamente indexada) ficou como a melhor entre três seletividades (S0, S1 e S2), mas depois perdeu em desempenho até para a sua versão básica que foi superado pela opção de processamento *pipeTEhstb* (da classe *Pipeline* TE-C) e *intmTEhstb* (da classe ET Intercalado (FTRC)) na seletividade S3. Tal facilidade pode ser explicada pela simplicidade das linhas utilizadas na construção dos dados (sempre verticais e horizontais aos eixos) que não geram “*dead space*” e, assim, facilitam a execução dos índices espaciais. Mesmo assim, nota-se que a queda de desempenho foi bastante brusca entre as seletividades S1, S2 e S3 indicando dificuldades do sistema gerenciador de banco de dados em manter uma estratégia constante na execução das consultas na presença de índices que as deveriam auxiliar. Constante nesta e em todas as

análises é o desempenho das opções de processamento das classes *Pipeline* TE-C, ET Intercalado (FTRC) e a opção *pipeEhsbTepCf* da classe *Pipeline* E-T-C. Elas se revezaram nas primeiras colocações sempre muito próximas e começaram a ultrapassar o *baseline* de visão materializada na última seletividade analisada. Também surpresa nesta análise, a tendência de os índices baseados em vetor superarem os índices baseados em hierarquia na maior seletividade não se concretizou, inclusive com a diferença da porcentagem de desempenho entre a melhor opção usando o índice STB-index para a opção vencedora nas seletividades S2 e S3 aumentando de 20,14% para 23,01%. Tal comportamento indica que é necessário alcançar seletividades maiores neste tipo de dado linha com as características mencionadas de não ter *dead space* para que a poda de registros dos índices baseados em hierarquia já não seja relevante e as opções de processamento baseadas no índice STB-index possam superar as opções baseadas no HSTB-index.

Nas zonas de variação Z2 e Z3 temos novamente opções de processamento das classes *Pipeline* E-T-C, *Pipeline* T-E-C, Independente TE/C e Independente T/E/C se revezando nas posições intermediárias das zonas de variação. A interferência de opções de processamento baseadas na classe Independente TE/C aconteceu em todas as seletividades e da Independente T/E/C aconteceu somente S3, sendo que nesta última todas as opções de processamento melhoraram de desempenho em relação à opção vencedora. Pode-se notar, inclusive, uma melhora constante do desempenho das opções independentes, mas tais melhoras ainda não conseguiram rivalizar com as opções de processamento vencedoras. Talvez seletividades mais altas possam confirmar a tendência de melhora no desempenho relativo e incluí-las entre as opções vencedoras neste tipo de dado espacial.

A zona de variação Z4 seguiu a mesma disposição das opções de processamento nas seletividades S0, S1 e S2 com opções de processamento pertencentes à classe Independente T/E/C e a opção *pipeEsbTefCf* em último lugar. Já na seletividade S3 (alta) teve a interferência de outras opções em *pipeline* que perderam em desempenho relativo à opção de processamento vencedora. As opções de processamento que utilizam o conceito de *pipeline* e seguem aparecendo nesta zona de variação o estão por serem baseadas no tipo de processamento Tef que envolve consulta ao software de processamento de índices bitmap e leitura de

resultados em disco, que torna a operação mais custosa que as demais. A única exceção a essa regra foi a opção pipeTpEhscCf no quadrante Z4-S3.

Tabela 30: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para street com todas as seletividades

	S0			S1			S2			S3			
	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	
Z1	VMA	BTree	0,000	VMA	BTree	0,000	VMA	BTree	0,000	pipeTEhstb	HSTB	0,000	melhor ↓ pior
	pipeEhscTepCf	HSB	73,573	pipeTEhstb	HSTB	70,536	pipeTEhstb	HSTB	8,910	intmTEhstb	HSTB	4,983	
	intmTEhstb	HSTB	73,621	pipeEhscTepCf	HSB	70,623	intmTEhstb	HSTB	9,510	VMB	N/A	16,146	
	pipeTEhstb	HSTB	73,672	intmTEhstb	HSTB	71,304	pipeEhscTep	HSB	16,825	pipeTEstb	STB	23,009	
	intmTEstb	STB	79,621	intmTEstb	STB	76,238	pipeTEstb	STB	20,145	intmTEstb	STB	23,389	
Z2	pipeTEstb	STB	79,875	pipeTEstb	STB	76,908	intmTEstb	STB	20,771	pipeEhscTepCf	HSB	34,760	
	pipeEsbTepCf	SB	82,775	pipeEsbTepCf	SB	79,919	pipeEsbTepCf	SB	44,253	pipeTpEsbCf	SB	35,367	
	pipeTpEhscCf	HSB	92,582	pipeTpEhscCf	HSB	90,660	pipeTpEhscCf	HSB	59,019	indpTEhstb	HSTB	41,179	
	pipeTpEsbCf	SB	93,858	pipeTpEsbCf	SB	91,505	pipeTpEsbCf	SB	63,567	indpTEstb	STB	44,290	
	pipeTfEsbCf	SB	95,858	indpTEhstb	HSTB	95,003	pipeEhscTef	HSB	74,011	indpTpEhscCf	HSB	48,764	
Z3	indpTEhstb	HSTB	96,075	pipeTfEsbCf	SB	95,137	indpTEhstb	HSTB	74,998	pipeEhscTefCf	HSB	51,313	
	pipeEhscTefCf	HSB	96,098	indpTEstb	STB	95,218	indpTEstb	STB	76,100	pipeTfEsbCf	SB	54,002	
	indpTEstb	STB	96,234	pipeEhscTefCf	HSB	95,252	pipeTfEsbCf	SB	76,792	indpTpEsbCf	SB	55,958	
	pipeTfEhscCf	HSB	96,236	pipeTfEhscCf	HSB	95,407	pipeTfEhscCf	HSB	78,294	pipeTfEhscCf	HSB	57,072	
	indpTpEhscCf	HSB	97,103	indpTpEhscCf	HSB	96,443	indpTpEhscCf	HSB	81,447	indpTfEhscCf	HSB	62,586	
Z4	indpTpEsbCf	SB	97,358	indpTpEsbCf	SB	96,610	indpTpEsbCf	SB	82,643	pipeTpEhscCf	HSB	63,241	
	indpTfEhscCf	HSB	97,805	indpTfEhscCf	HSB	97,384	indpTfEhscCf	HSB	86,462	indpTfEsbCf	SB	65,161	
	indpTfEsbCf	SB	97,970	indpTfEsbCf	SB	97,445	indpTfEsbCf	SB	86,593	pipeEsbTepCf	SB	70,014	
	pipeEsbTepCf	SB	98,323	pipeEsbTepCf	SB	97,977	pipeEsbTepCf	SB	88,868	pipeEsbTepCf	SB	70,894	

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

Chama a atenção a pouca variação entre as opções de processamento pertencentes às zonas de variação Z2 a Z4 nas seletividades S0, S1 e S2, indicando que as seletividades mais baixas não influenciam no resultado das opções para este tipo de dado.

Concluindo, verifica-se que o aumento da seletividade não afetou o conjunto de opções de processamento vencedoras, no caso o *baseline* de visão materializada, pipeTEhstb, intmTEhstb, pipeTEstb e intmTEstb, e também não afetou de forma geral o relacionamento entre as classes. Quanto a estas, verifica-se a seguinte ordem decrescente de desempenho (melhor para pior): *Baseline* de visão materializada, *pipeline* TE-C alternando com opções de processamento da classe E-T intercalado (FTRC) e a opção pipeEhscTepCf (da classe *Pipeline* E-T-C) nas primeiras posições, *pipeline* T-E-C, *pipeline* E-T-C e processamento independente (com alternância entre processamento independente T/E/C e TE/C). Outro ponto a destacar é o desempenho do *baseline* de visão materializada que ficou entre as

opções de processamento da zona de variação Z1 e ganhou em desempenho nas seletividades mais baixas. Conclui-se que para este tipo de dado espacial, o uso do *baseline* de visão materializada é interessante nas seletividades mais baixas (S0 e S1) mas inapropriado para maiores valores de seletividades e a implementação das opções de processamento, especialmente das classes ET intercalado (FTRC) e *pipeline* TE-C, é imprescindível para se obter um bom desempenho no processamento de consultas STOLAP.

6.2.5 Análise das Configurações Envolvendo Polígonos de Cidades (city)

Segue-se nesta análise a mesma divisão em zonas de variação feita na análise do item 6.2.2.

As opções de processamento baseadas nas classes ET Intercalado (FTRC) e *Pipeline* TE-C continuam se destacando na zona de variação Z1 e obtendo os melhores desempenhos na maioria das seletividades. Tivemos mais uma vez a interferência do *baseline* de visão materializada na sua versão avançada (altamente indexada) vindo a ganhar de maneira significativa (mais de 65% de diferença de desempenho para a opção de processamento na segunda colocação) na seletividade S0, mas este não conseguiu repetir o desempenho já na seletividade S1, ficando na zona de variação Z3 daí em diante. As opções de processamento *intmTEhstb* e *pipeTEhstb* estiveram sempre muito próximas em primeiro e segundo lugares com diferença de desempenho variando entre 0,38% e 1,78%, o que pode ser considerado um “empate técnico”, uma vez que qualquer variação de *hardware* ou *software* pode afetar os resultados. Notório mais uma vez é o fato das opções de processamento baseadas no HSTB-index e HSB-index dominarem as comparações dos índices base nessa zona de variação. Inclusive, temos a presença constante das opções de processamento *pipeTfEhsbCf* e *pipeTpEhsbCf* (da classe *Pipeline* T-E-C) devido à força da poda de registros da indexação com hierarquia de MBRs, da simplicidade da forma geométrica usada para formar as cidades (retângulos com muito pouco ou nenhum “*dead space*”) e da quantidade limitada de registros da tabela de cidades não contribuir negativamente no processamento do predicado temporal (como estava fazendo nos outros tipos de dados). Como comentado em análises anteriores o processamento em *pipeline* e a estrutura espaço-temporal

interna ao índice utilizado formam uma base sólida para que as classes de processamento utilizando tais características tenham tamanho sucesso.

Na zona de variação Z2 temos somente opções em *pipeline* e estas estão sempre muito próximas em desempenho em relação à pior opção de processamento da zona de variação Z1. A predominância é das opções de processamento que utilizam o STB-index como índice base e nota-se que a diferença de desempenho destas frente às opções de processamento da zona Z1 vai diminuindo gradativamente sendo provável que comecem a rivalizar com estas nas seletividades mais altas somente.

Tabela 31: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para city com todas as seletividades

	S0			S1			S2			S3			
	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	
Z1	VMA	BTree	0,000	intmTEhstb	HSTB	0,000	intmTEhstb	HSTB	0,000	pipeTEhstb	HSTB	0,000	melhor ↓ pior
	intmTEhstb	HSTB	65,643	pipeTEhstb	HSTB	0,380	pipeEhstbTepCf	HSB	1,776	intmTEhstb	HSTB	0,587	
	pipeTEhstb	HSTB	66,654	pipeEhstbTepCf	HSB	0,419	pipeTEhstb	HSTB	1,805	pipeTpEhstbCf	HSB	0,877	
	pipeTTEhstbCf	HSB	67,647	pipeTTEhstbCf	HSB	4,891	pipeTpEhstbCf	HSB	4,027	pipeTTEhstbCf	HSB	1,284	
	pipeTpEhstbCf	HSB	68,264	pipeTpEhstbCf	HSB	5,261	pipeTTEhstbCf	HSB	5,228	pipeEhstbTepCf	HSB	2,546	
Z2	pipeTEstb	STB	72,933	pipeTEstb	STB	18,828	pipeTEstb	STB	14,504	intmTEstb	STB	9,823	
	intmTEstb	STB	73,507	intmTEstb	STB	18,873	intmTEstb	STB	14,518	pipeTEstb	STB	10,842	
	pipeTpEsbCf	SB	74,767	pipeTpEsbCf	SB	23,287	pipeTpEsbCf	SB	18,396	pipeTpEsbCf	SB	11,776	
	pipeEhstbTepCf	HSB	75,191	pipeEsbTepCf	SB	30,479	pipeEsbTepCf	SB	26,136	pipeEsbTepCf	SB	30,618	
	pipeEsbTepCf	SB	82,125	VMA	BTree	51,797	VMB	N/A	78,186	VMB	N/A	64,874	
Z3	pipeEhstbTefCf	HSB	94,878	indpTEhstb	HSTB	83,200	indpTpEhstbCf	HSB	79,863	indpTpEhstbCf	HSB	69,120	
	indpTEstb	STB	95,420	pipeEhstbTefCf	HSB	84,512	pipeEhstbTefCf	HSB	80,302	pipeEhstbTefCf	HSB	70,061	
	indpTpEhstbCf	HSB	95,448	indpTpEsbCf	SB	84,965	indpTEstb	STB	80,394	indpTEhstb	HSTB	70,925	
	indpTEhstb	HSTB	95,563	indpTEstb	STB	85,379	indpTEhstb	HSTB	80,413	indpTEstb	STB	72,542	
	indpTpEsbCf	SB	95,594	pipeTTEsbCf	SB	85,504	pipeTTEsbCf	SB	80,683	indpTpEsbCf	SB	72,606	
Z4	pipeTTEsbCf	SB	95,753	indpTpEhstbCf	HSB	85,583	indpTpEsbCf	SB	81,108	pipeTTEsbCf	SB	72,748	
	indpTTEhstbCf	HSB	97,439	indpTTEsbCf	SB	91,533	indpTTEsbCf	SB	88,787	indpTTEhstbCf	HSB	82,998	
	indpTTEsbCf	SB	97,442	indpTTEhstbCf	HSB	91,834	indpTTEhstbCf	HSB	88,961	indpTTEsbCf	SB	83,060	
	pipeEsbTefCf	SB	98,049	pipeEsbTefCf	SB	93,129	pipeEsbTefCf	SB	90,667	pipeEsbTefCf	SB	85,263	

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

As zonas de variação Z3 e Z4 possuem a presença das opções de processamento das classes Independente TE/C e Independente T/E/C misturadas com as opções em *pipeline* que utilizam as técnicas Tf e Tef no processamento da cláusula temporal (notoriamente mais custosas que as técnicas Tp e Tep). À exceção da seletividade S0, as zonas são iniciadas pelo *baseline* de visão materializada e a menor perda de desempenho das opções de processamento destas zonas de variação pode ser observada no quadrante Z3-S1 com 51,80%. Ao contrário das análises anteriores, o desempenho das opções de processamento das

classes independentes não teve melhoras significativas a ponto de migrarem de zona de variação.

Mais uma vez fica evidente que o aumento da seletividade não afetou o conjunto de opções de processamento vencedoras, no caso a *intmTEhstb*, *pipeTEhstb*, *pipeEhsbTepCf*, *pipeTpEhsbCf* e *pipeTfEhsbCf* mas temos o comportamento entre as classes afetado pela inclusão de opções de processamento da classe *Pipeline* T-E-C às classes já tidas como grandes vencedoras em outras análises (ET Intercalado (FTRC) e *Pipeline* TE-C). Quanto à comparação entre as classes, verifica-se a seguinte ordem decrescente de desempenho (melhor para pior): E-T intercalado (FTRC) intercalando com opções de processamento da classe *pipeline* TE-C e o *baseline* de visão materializada na seletividade S0, *pipeline* T-E-C e *pipeline* E-T-C. O processamento independente (com alternância entre processamento independente T/E/C e TE/C) ficou relegado às zonas de variação com piores desempenho. O *baseline* de visão materializada mostrou força na seletividade S0, mas caiu drasticamente de desempenho nas seletividades seguintes indicando a imaturidade do sistema gerenciador de banco de dados frente a diferentes seletividades. A exemplo das outras análises realizadas, conclui-se que o uso do *baseline* de visão materializada é inapropriado para maiores valores de seletividades para este tipo de dado espacial e a implementação das opções de processamento, especialmente das classes ET intercalado (FTRC) e *pipeline* TE-C, é imprescindível para se obter um bom desempenho no processamento de consultas STOLAP.

6.2.6 Conclusões

Desta análise podemos concluir que as opções de processamento baseadas nas classes de processamento ET Intercalado (FTRC) e *Pipeline* TE-C possuem os melhores desempenhos em todas as seletividades e são recomendadas para futuras implementações. Elas têm por característica a inclusão da informação para limitação temporal (datas de validade inicial e final do registro) e para limitação espacial (MBRs) inclusas na mesma estrutura de índice e o processamento em *pipeline* dos registros (onde a resposta de um dos predicados é utilizado como limitador no processamento do predicado seguinte). Fica a ressalva para o tipo de dado linha vindo da tabela de ruas (“street”) onde o *baseline* de visão materializada na sua

versão avançada (altamente indexada) obteve desempenhos consistentes nas seletividades S0, S1 e S2 e pode ser considerado como alternativa quando observadas as mesmas condições de testes (quantidade de registros da tabela e linhas verticais e horizontas somente).

A forma da construção das seletividades (onde se fixou a seletividade temporal alta para depois variar a janela de consulta para encontrar a seletividade desejada) não pareceu privilegiar as opções de processamento da classe ET Intercalado (FTRC) (por esta fazer a limitação espacial a priori) como esperado. A principal característica que permeia as opções de processamento vencedoras é a poda de registros proporcionada pela hierarquia de MBRs das opções de processamento baseadas no HSTB-index e HSB-index.

6.3 Análise da Influência do Tipo de Dado Espacial no Desempenho das Opções de Processamento

As análises a seguir tentarão esclarecer como foi o comportamento das opções de processamento nos vários tipos de dados espaciais para encontrar possíveis tendências que possam ajudar em futuras implementações.

6.3.1 Objetivo da Análise

O objetivo desta análise é identificar possíveis tendências entre as opções de processamento, suas classes e características, para identificar se a variação do tipo de dado espacial possui influência nos resultados obtidos e analisar se esta influência foi positiva ou negativa de modo a evitá-las ou atingi-las nas várias situações a serem enfrentadas em futuras pesquisas ou implementações. Para tanto, fixada uma determinada seletividade, comparou-se o desempenho relativo das opções nos diferentes tipos de dados.

6.3.2 Análise das Configurações Envolvendo Seletividade Muito Baixa (S0)

Observando-se o padrão de cores obtidos a partir das opções de processamento e suas respectivas classes pode-se observar a formação de quatro regiões horizontais (Z1 a Z4) que chamaremos de zonas de variação.

A zona de variação Z1 nos indica que temos o *baseline* de visão materializada e as opções de processamento baseadas nas classes ET intercalado (FTRC) e *Pipeline* TE-C como as que obtiveram os melhores desempenhos para os tipos de dados espaciais na seletividade muito baixa (S0), onde a quantidade de registros do conjunto resposta após a limitação espaço-temporal deve chegar entre 1 ou 2 registros. O *baseline* de visão materializada na sua versão avançada (altamente indexada) ganhou maior destaque por que aparece como a melhor opção nos tipos de dados “street” e “city” – linhas e polígonos), seguido pela opção *intmTEhstb* (que é a melhor opção de processamento do tipo de dados “saddress” – pontos vindos da tabela “supplier”) e pela *pipeTEhstb* (melhor opção de processamento para o tipo de dados *caddress* – pontos vindos da tabela “customer”). Nota-se também que, para esta seletividade muito baixa, as opções de processamento que tiveram os melhores desempenhos nesta zona de variação Z1 possuem um índice hierarquizado para a apoiar (HSTB-index, HSB-index ou a própria B-Tree usada pelo sistema gerenciador de banco de dados) indicando que a poda de candidatos proporcionada pelos métodos em árvore possui ampla vantagem. Além dessa característica, o processamento em *pipeline* está presente em todas as opções de processamento.

A segunda zona de variação contém, em sua maioria, opções de processamento das classes *Pipeline* T-E-C e *Pipeline* T-E-C em posições variadas e sem um padrão definido. As porcentagens de perda das opções de processamento desta zona de variação também variaram bastante (de 30,14% a 96,25%), o que aumenta a imprevisibilidade das opções.

Tabela 32: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para a seletividade muito baixa (S0) com todos os tipos de dados

	address			saddress			street			city			
	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	
Z1	pipeTEhstb	HSTB	0,000	intrmTEhstb	HSTB	0,000	VMA	BTree	0,000	VMA	BTree	0,000	melhor ↓ pior
	VMA	BTree	2,646	pipeTEhstb	HSTB	0,328	pipeEhstbTepCf	HSB	73,573	intrmTEhstb	HSTB	65,643	
	intrmTEhstb	HSTB	10,148	pipeEhstbTepCf	HSB	8,089	intrmTEhstb	HSTB	73,621	pipeTEhstb	HSTB	66,654	
	pipeEhstbTepCf	HSB	33,779	pipeTEstb	STB	22,974	pipeTEhstb	HSTB	73,672	pipeTfEhstbCf	HSB	67,647	
	intrmTEstb	STB	55,376	intrmTEstb	STB	23,060	intrmTEstb	STB	79,621	pipeTpEhstbCf	HSB	68,264	
	pipeTEstb	STB	55,721	pipeEsbTepCf	SB	23,889	pipeTEstb	STB	79,875	pipeTEstb	STB	72,933	
Z2	pipeEsbTepCf	SB	60,936	pipeTpEsbCf	SB	30,143	pipeEsbTepCf	SB	82,775	intrmTEstb	STB	73,507	
	pipeTpEsbCf	SB	94,150	VMB	N/A	31,297	pipeTpEhstbCf	HSB	92,582	pipeTpEsbCf	SB	74,767	
	pipeEhstbTefCf	HSB	96,127	pipeTpEhstbCf	HSB	42,407	pipeTpEsbCf	SB	93,858	pipeEhstbTepCf	HSB	75,191	
	pipeTfEhstbCf	HSB	96,253	pipeEhstbTefCf	HSB	86,410	pipeTfEsbCf	SB	95,858	pipeEsbTepCf	SB	82,125	
	indpTEhstb	HSTB	96,401	indpTpEhstbCf	HSB	86,645	indpTEhstb	HSTB	96,075	pipeEhstbTefCf	HSB	94,878	
	pipeTfEsbCf	SB	96,407	indpTEhstb	HSTB	86,724	pipeEhstbTefCf	HSB	96,098	indpTEstb	STB	95,420	
Z3	indpTEstb	STB	96,540	pipeTfEhstbCf	HSB	86,881	indpTEstb	STB	96,234	indpTpEhstbCf	HSB	95,448	
	pipeTpEhstbCf	HSB	96,708	pipeTfEsbCf	SB	87,651	pipeTfEhstbCf	HSB	96,236	indpTEhstb	HSTB	95,563	
	pipeEsbTefCf	SB	97,086	indpTpEsbCf	SB	87,921	indpTpEhstbCf	HSB	97,103	indpTpEsbCf	SB	95,594	
	indpTpEsbCf	SB	97,185	indpTEstb	STB	88,027	indpTpEsbCf	SB	97,358	pipeTfEsbCf	SB	95,753	
	indpTpEhstbCf	HSB	97,439	pipeEsbTefCf	SB	91,674	indpTfEhstbCf	HSB	97,805	indpTfEhstbCf	HSB	97,439	
	indpTfEsbCf	SB	97,973	indpTfEhstbCf	HSB	92,498	indpTfEsbCf	SB	97,970	indpTfEsbCf	SB	97,442	
Z4	indpTfEhstbCf	HSB	98,086	indpTfEsbCf	SB	93,155	pipeEsbTefCf	SB	98,323	pipeEsbTefCf	SB	98,049	

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

As zonas de variação Z3 e Z4 se mostraram misturadas entre as opções de processamento cujas classes foram baseadas em *pipeline* e as independentes. Mais uma vez, as opções de processamento em *pipeline* que aqui aparecem utilizam as técnicas Tf e Tef de processamento do predicado temporal (pipeTpEhstbCf é a única exceção no quadrante Z3-address). As opções de processamento nestas zonas de variação já estão apresentando perdas de mais de 86% em relação à opção de processamento. Portanto, tais opções devem ser evitadas quando do desenvolvimento de índices nesta seletividade.

Concluimos através desta análise que a variação do tipo de dado espacial na seletividade muito baixa (S0) afetou apenas minimamente o conjunto das opções de processamento vencedoras, mas afetou muito o relacionamento entre as classes nas zonas de variação inferiores da Tabela 32. Podemos apenas elencar as classes E-T intercalado (FCTR) e *pipeline* TE-C (cujas opções de processamento aparecem somente na zona de variação Z1) como tendo um comportamento estável frente às alterações de tipo de dados. Apesar de ser o vencedor em dois dos tipos de dados e ter um desempenho muito próximo para outro tipo de dado, o *baseline* de visão materializada não manteve um comportamento estável no tipo de dados saddress e

caiu para a zona de variação Z2 com perda de 31,29% com relação à opção vencedora. Conclui-se que o uso do *baseline* de visão materializada para a seletividade muito baixa é indicado, mas a implementação de opções de processamento das classes que se mantiveram estáveis frente à variação do tipo de dado espacial é importante para se obter sempre o melhor desempenho no processamento de consulta STOLAP.

6.3.3 Análise das Configurações Envolvendo Seletividade Baixa (Seletividade S1)

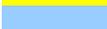
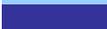
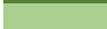
Observando-se o padrão de cores obtidos a partir das opções de processamento e suas respectivas classes pode-se observar a formação de quatro regiões horizontais (Z1 a Z4) que chamaremos de zonas de variação.

A zona de variação Z1 nos indica que mais uma vez temos as opções de processamento baseadas nas classes ET intercalado (FTRC), *Pipeline* TE-C e a opção pipeE2T2C1 da classe *Pipeline* E-T-C como as que obtiveram os melhores desempenhos para todos os tipos de dados espaciais para esta seletividade baixa (S1), onde a quantidade de registros do conjunto resposta após a limitação espaço-temporal deve chegar entre 10 a 20 registros. As opções de processamento que tiveram maior destaque foram a pipeTEhstb (ganhadora nos tipos de dados caddress e saddress e segunda colocada para os tipos de dados street e city), a intrmTEhstb (segunda colocada nos tipos de dados caddress e saddress, quarta colocada no tipo de dado street e vencedora no tipo de dado city) e pipeEhsbTepCf (terceira colocada em todos os tipos de dados). Elas têm em comum o processamento dos predicados em *pipeline* e uma das opções de índice base baseada em hierarquia (HSTB-index e HSB-index). O *baseline* de visão materializada na sua versão avançada se destacou no tipo de dado street ao ganhar com pouco mais de 70% de diferença no desempenho para a opção na segunda colocação – a maior diferença de desempenho entre a primeira e segunda posições. As razões para que as opções dessa zona de variação sejam as melhores e que foram explicadas na análise de tendências para o tipo de dados caddress (endereços na representação de pontos vindos da tabela “customer”) com seletividade variável continuam sendo verdadeiras para esta análise.

Tabela 33: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para a seletividade baixa (S1) com todos os tipos de dados

	caddress			saddress			street			city			
	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	
Z1	pipeTEhstb	HSTB	0,000	pipeTEhstb	HSTB	0,000	VMA	BTree	0,000	intmTEhstb	HSTB	0,000	melhor ↓ pior
	intmTEhstb	HSTB	9,582	intmTEhstb	HSTB	0,818	pipeTEhstb	HSTB	70,536	pipeTEhstb	HSTB	0,380	
	pipeEhstbTepCf	HSB	10,882	pipeEhstbTepCf	HSB	7,554	pipeEhstbTepCf	HSB	70,623	pipeEhstbTepCf	HSB	0,419	
	intmTEstb	STB	32,856	VMB	N/A	16,299	intmTEhstb	HSTB	71,304	pipeTfEhstbCf	HSB	4,891	
	pipeTEstb	STB	33,179	intmTEstb	STB	22,131	intmTEstb	STB	76,238	pipeTpEhstbCf	HSB	5,261	
Z2	pipeEsbTepCf	SB	39,088	pipeEsbTepCf	SB	23,785	pipeTEstb	STB	76,908	pipeTEstb	STB	18,828	
	VMA	BTree	43,452	pipeTEstb	STB	24,228	pipeEsbTepCf	SB	79,919	intmTEstb	STB	18,873	
	pipeTpEsbCf	SB	93,232	pipeTpEhstbCf	HSB	34,190	pipeTpEhstbCf	HSB	90,660	pipeTpEsbCf	SB	23,287	
	pipeEhstbTefCf	HSB	93,280	pipeTpEsbCf	SB	37,082	pipeTpEsbCf	SB	91,505	pipeEsbTepCf	SB	30,479	
	pipeTfEsbCf	SB	93,373	pipeEhstbTefCf	HSB	85,588	indpTEhstb	HSTB	95,003	VMA	BTree	51,797	
Z3	indpTEhstb	HSTB	93,438	indpTEhstb	HSTB	86,038	pipeTfEsbCf	SB	95,137	indpTEhstb	HSTB	83,200	
	pipeTfEhstbCf	HSB	93,532	pipeTfEhstbCf	HSB	86,130	indpTEstb	STB	95,218	pipeEhstbTefCf	HSB	84,512	
	indpTEstb	STB	93,651	indpTEstb	STB	87,004	pipeEhstbTefCf	HSB	95,252	indpTpEsbCf	SB	84,965	
	pipeTpEhstbCf	HSB	93,983	pipeTfEsbCf	SB	87,364	pipeTfEhstbCf	HSB	95,407	indpTEstb	STB	85,379	
	pipeEsbTefCf	SB	95,108	indpTpEsbCf	SB	87,584	indpTpEhstbCf	HSB	96,443	pipeTfEsbCf	SB	85,504	
Z4	indpTpEhstbCf	HSB	95,922	indpTpEhstbCf	HSB	87,649	indpTpEsbCf	SB	96,610	indpTpEhstbCf	HSB	85,583	
	indpTpEsbCf	SB	96,490	pipeEsbTefCf	SB	91,877	indpTfEhstbCf	HSB	97,384	indpTfEsbCf	SB	91,533	
	indpTfEhstbCf	HSB	96,558	indpTfEhstbCf	HSB	92,632	indpTfEsbCf	SB	97,445	indpTfEhstbCf	HSB	91,834	
	indpTfEsbCf	SB	96,570	indpTfEsbCf	SB	93,003	pipeEsbTefCf	SB	97,977	pipeEsbTefCf	SB	93,129	

Cores usadas

ET Intercalado (FTRC)		STB-index	
Pipeline TE-C		HSTB-index	
Pipeline T-E-C		SB-index	
Pipeline E-T-C		HSB-index	
Independente T/E/C			
Independente TE/C			

A segunda zona de variação contém opções de processamento das classes *Pipeline* T-E-C e *Pipeline* E-T-C em posições variadas e sofrendo interferência de opções de processamento das classes ET Intercalado (FCTR) (intmTEstb no quadrante Z2-city), *Pipeline* ET-C (pipeTEstb nos quadrantes Z2-saddress, Z2-street e Z2-city) e do *baseline* de visão materializada no quadrante Z2-caddress. A exemplo da análise da Seção 6.3.2, as porcentagens de perda de desempenho das opções de processamento desta zona de variação também oscilaram bastante (de 18,83% a 93,28%), o que aumenta a imprevisibilidade das opções.

As zonas de variação Z3 e Z4 também tiveram o mesmo comportamento encontrado na análise das variações de tipo de dado para seletividade muito baixa (item 6.3.2) com opções em *pipeline* utilizando as técnicas de processamento Tf e Tef e as opções de processamento das classes Independente TE/C e Independente T/E/C. As porcentagens de perda frente à opção de processamento vencedora já são bastante altas (variando entre 51,80% e 97,97%, com média aritmética 90,82%).

Podemos tirar desta análise para a seletividade baixa (S1) as mesmas conclusões da análise anterior (item 6.3.2), onde a variação do tipo de dado espacial

não afetou as classes das opções de processamento vencedoras (ET Intercalado (FTRC), *Pipeline* TE-C e *Pipeline* E-T-C – com a opção de processamento *pipeEhsbTepCf*), com exceção do tipo de dado linha para cidades com ganho da visão materializada. Além disso, afetou o relacionamento entre as classes principalmente nas zonas de variação inferiores da Tabela 33 (a partir da zona de variação Z2). Da mesma maneira, o *baseline* de visão materializada não manteve um comportamento estável entre os tipos de dados e refletiu a instabilidade das outras opções de processamento nos resultados sendo que a única observação que pode ser feita é sobre seu alto desempenho no tipo de dado linha da tabela “street”. Conclui-se que o uso do *baseline* de visão materializada pode oferecer riscos ao desempenho esperado das consultas e que a implementação de opções de processamento das classes que se mantiveram estáveis frente à variação do tipo de dado espacial é importante para se obter um bom desempenho no processamento de consultas STOLAP.

6.3.4 Análise das Configurações Envolvendo Seletividade Média (S2)

Observando-se o padrão de cores obtidos a partir das opções de processamento e suas respectivas classes pode-se observar a formação de quatro regiões horizontais (Z1 a Z4) que chamaremos de zonas de variação.

A zona de variação Z1 nos indica que mais uma vez temos as opções de processamento baseadas nas classes ET intercalado (FTRC), *Pipeline* TE-C e a opção *pipeEhsbTepCf* (da classe *Pipeline* E-T-C) como as que obtiveram os melhores desempenhos para todos os tipos de dados espaciais nesta seletividade média (S2), onde a quantidade de registros do conjunto resposta após a limitação espaço-temporal deve chegar entre 100 a 200 registros. Pode-se ver também a interferência do *baseline* de visão materializada na sua forma avançada (altamente indexada) que teve o melhor desempenho no tipo de dado “street” e de duas opções da classe *Pipeline* T-E-C no tipo de dado linha para cidades. A opção de processamento que ganhou mais destaque foi a *intmTEhstb* aparecendo como a melhor em dois tipos de dados (*caddress* e *city*), como a segunda melhor no tipo de dado “*saddress*” e como a terceira melhor no tipo de dado “street”. A opção *pipeTEhstb* também merece destaque por ter tido o melhor desempenho no tipo de dado “*saddress*”, ter sido a segunda colocada nos tipos de dado “*caddress*” e “street”

e a terceira colocada no tipo de dado “city”. Quanto à comparação dos índices base, o comportamento das opções de processamento nesta zona de variação demonstrou que as opções de processamento que se basearam em hierarquia de MBRs (HSTB-index, HSB-index e BTree do sistema gerenciador de bancos de dados) possuíram melhor desempenho frente às opções de processamento que se basearam em vetor de MBRs (STB-index) e nota-se uma crescente queda de desempenho deste ao processar dados mais complexos. As razões para que as opções dessa zona de variação sejam as melhores e que foram explicadas na análise de tendências para a seletividade S0 (item 6.3.2) continuam sendo verificadas nesta análise.

Tabela 34: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para a seletividade média (S2) com todos os tipos de dados

	caddress			saddress			street			city			
	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	
Z1	intmTEhstb	HSTB	0,000	pipeTEhstb	HSTB	0,000	VMA	BTree	0,000	intmTEhstb	HSTB	0,000	melhor ↓ pior
	pipeTEhstb	HSTB	0,749	intmTEhstb	HSTB	0,589	pipeTEhstb	HSTB	8,910	pipeEhstbTepCf	HSB	1,776	
	pipeTEstb	STB	4,416	pipeEhstbTepCf	HSB	12,369	intmTEhstb	HSTB	9,510	pipeTEhstb	HSTB	1,805	
	intmTEstb	STB	4,721	intmTEstb	STB	17,936	pipeEhstbTepCf	HSB	16,825	pipeTpEhstbCf	HSB	4,027	
Z2	pipeEhstbTepCf	HSB	12,136	pipeTEstb	STB	18,360	pipeTEstb	STB	20,145	pipeTfEhstbCf	HSB	5,228	
	pipeEsbTepCf	SB	26,374	pipeEsbTepCf	SB	19,799	intmTEstb	STB	20,771	pipeTEstb	STB	14,504	
	VMB	N/A	35,608	VMB	N/A	36,257	pipeEsbTepCf	SB	44,253	intmTEstb	STB	14,518	
	indpTEhstb	HSTB	89,116	pipeTpEsbCf	SB	36,924	pipeTpEhstbCf	HSB	59,019	pipeTpEsbCf	SB	18,396	
Z3	pipeTfEhstbCf	HSB	89,527	pipeTpEhstbCf	HSB	36,943	pipeTpEsbCf	SB	63,567	pipeEsbTepCf	SB	26,136	
	pipeTfEsbCf	SB	89,548	indpTEhstb	HSTB	84,111	pipeEhstbTefCf	HSB	74,011	VMB	N/A	78,186	
	indpTEstb	STB	89,581	indpTEstb	STB	84,701	indpTEhstb	HSTB	74,998	indpTpEhstbCf	HSB	79,863	
	pipeEhstbTefCf	HSB	89,588	pipeEhstbTefCf	HSB	85,183	indpTEstb	STB	76,100	pipeEhstbTefCf	HSB	80,302	
Z4	pipeTpEsbCf	SB	90,549	indpTpEhstbCf	HSB	85,558	pipeTfEsbCf	SB	76,792	indpTEstb	STB	80,394	
	pipeTpEhstbCf	HSB	91,685	pipeTfEhstbCf	HSB	85,859	pipeTfEhstbCf	HSB	78,294	indpTEhstb	HSTB	80,413	
	pipeEsbTefCf	SB	91,866	pipeTfEsbCf	SB	85,983	indpTpEhstbCf	HSB	81,447	pipeTfEsbCf	SB	80,683	
	indpTpEhstbCf	HSB	93,486	indpTpEsbCf	SB	87,016	indpTpEsbCf	SB	82,643	indpTpEsbCf	SB	81,108	
			indpTpEsbCf	SB	91,023	indpTfEhstbCf	HSB	86,462	indpTfEsbCf	SB	88,787		
			indpTfEsbCf	SB	91,952	indpTfEsbCf	SB	86,593	indpTfEhstbCf	HSB	88,961		
			indpTfEhstbCf	HSB	94,545	indpTfEsbCf	SB	92,066	pipeEsbTefCf	SB	90,667		

Cores usadas

- ET Intercalado (FTRC)
- Pipeline TE-C
- Pipeline T-E-C
- Pipeline E-T-C
- Independente T/E/C
- Independente TE/C



- STB-index
- HSTB-index
- SB-index
- HSB-index



A segunda zona de variação contém opções de processamento de todas as classes menos a Independente T/E/C com perda de desempenho frente à opção de processamento vencedora de 14,50% (quadrante Z2-city) a 89,53% (quadrante Z2-caddress). O *baseline* de visão materializada na sua versão básica (não indexada) está presente nesta zona de variação para o tipo de dado ponto (caddress e

saddress). Nenhum comportamento distinto pode ser observado ao analisarmos os índices base desta zona de variação.

As zonas de variação Z3 e Z4 se mostraram misturadas entre as opções de processamento cujas classes foram baseadas em *pipeline* e as independentes. Para todos os tipos de dados espaciais, as opções de processamento já estão apresentando perdas de mais de 74,01% em relação à opção de processamento vencedora. Apesar das opções independentes estarem quase sempre agrupadas, vemos as classes de processamento aparecendo bem misturadas e não se consegue identificar nenhum padrão.

Em vista das observações acima, pode-se concluir que a variação do tipo de dado espacial na seletividade média (S2) afetou pouco as opções de processamento vencedoras (intmTEhstb e pipeTEhstb), com exceção do tipo de dado linha da tabela cidades no qual a visão materializada foi a vencedora, mas afetou o relacionamento entre as classes principalmente nas zonas de variação inferiores da Tabela 34 (a partir da zona de variação Z2). Podemos apenas elencar as classes E-T intercalado (FCTR), *Pipeline* TE-C e a opção pipeEhsbTepCf (da classe *Pipeline* E-T-C) como tendo um comportamento estável frente às alterações de tipo de dados. Da mesma maneira, o *baseline* de visão materializada não manteve um comportamento estável entre os tipos de dados e reflete a instabilidade das outras opções de processamento nos resultados. Conclui-se que o uso do *baseline* de visão materializada pode oferecer riscos ao desempenho esperado das consultas e que a implementação de opções de processamento das classes que se mantiveram estáveis frente à variação do tipo de dado espacial é importante para se obter um bom desempenho no processamento de consultas STOLAP.

6.3.5 Análise das Configurações Envolvendo Seletividade Alta (S3)

Observando-se o padrão de cores obtidos a partir das opções de processamento e suas respectivas classes pode-se observar a formação de quatro regiões horizontais (Z1 a Z4) que chamaremos de zonas de variação.

Temos mais uma vez as opções de processamento baseadas nas classes ET intercalado (FTRC) e *Pipeline* TE-C como as que obtiveram os melhores desempenhos para todos os tipos de dados espaciais nesta seletividade alta (S3), onde a quantidade de registros do conjunto resposta após a limitação espaço-

temporal deve chegar entre 1000 a 2000 registros. A opção de processamento que ganhou maior destaque foi a pipeTEhstb aparecendo como a melhor entre todas as opções de processamento em três dos quatro tipos de dados espaciais e perfazendo ganhos de desempenho entre 0,59% e 4,98% com relação à segunda melhor opção de processamento. A exceção ficou por conta do tipo de dado ponto vindo da tabela de clientes (caddress) onde a opção de processamento pipeTEstb foi a vencedora com 2,73% de diferença de desempenho para a segunda opção (a intmTEstb). Mais uma vez temos a presença constante da opção de processamento pipeEhsbTepCf (da classe Pipeline E-T-C) nesta zona de variação. Ao compararmos os índices base utilizados pelas opções de processamento, vemos que o STB-index já rivaliza com os índices baseados em hierarquia no tipo de dado “caddress” mas ainda está um pouco aquém destes nos outros tipos de dados seja pela quantidade menor de registros na tabela alvo (saddress e city) seja pela necessidade de uma fase extra de refinamento (street e city).

Tabela 35: Tabela de opções de processamento e índices base por ordem de ganho de desempenho para a seletividade alta (S3) com todos os tipos de dados

	caddress			saddress			street			city			
	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	Opção	Índice base	Porc perda para o vencedor	
Z1	pipeTEstb	STB	0,000	pipeTEhstb	HSTB	0,000	pipeTEhstb	HSTB	0,000	pipeTEhstb	HSTB	0,000	melhor ↓ pior
	intmTEstb	STB	2,733	intmTEhstb	HSTB	2,900	intmTEhstb	HSTB	4,983	intmTEhstb	HSTB	0,587	
	intmTEhstb	HSTB	15,238	pipeEhsbTepCf	HSB	5,834	VMB	N/A	16,146	pipeTpEhsbCf	HSB	0,877	
	pipeTEhstb	HSTB	16,976	intmTEstb	STB	16,236	pipeTEstb	STB	23,009	pipeTEhsbCf	HSB	1,284	
	pipeEhsbTepCf	HSB	23,443	pipeTEstb	STB	16,508	intmTEstb	STB	23,389	pipeEhsbTepCf	HSB	2,546	
Z2	pipeEsbTepCf	SB	30,932	pipeEsbTepCf	SB	16,800	pipeEhsbTepCf	HSB	34,760	intmTEstb	STB	9,823	
	VMB	N/A	52,834	pipeTpEhsbCf	HSB	35,885	pipeTpEsbCf	SB	35,367	pipeTEstb	STB	10,842	
	pipeTfEsbCf	SB	84,932	pipeTpEsbCf	SB	38,852	indpTEhstb	HSTB	41,179	pipeTpEsbCf	SB	11,776	
	indpTEhstb	HSTB	85,058	VMB	N/A	63,360	indpTEstb	STB	44,290	pipeEsbTepCf	SB	30,618	
	indpTEstb	STB	85,347	pipeEhsbTefCf	HSB	84,143	indpTpEhsbCf	HSB	48,764	VMB	N/A	64,874	
Z3	pipeEhsbTefCf	HSB	85,456	indpTEhstb	HSTB	86,323	pipeEhsbTefCf	HSB	51,313	indpTpEhsbCf	HSB	69,120	
	pipeTfEhsbCf	HSB	86,228	indpTpEhsbCf	HSB	86,580	pipeTfEsbCf	SB	54,002	pipeEhsbTefCf	HSB	70,061	
	pipeTpEsbCf	SB	86,977	pipeTfEhsbCf	HSB	86,710	indpTpEsbCf	SB	55,958	indpTEhstb	HSTB	70,925	
	pipeEsbTefCf	SB	88,021	indpTEstb	STB	86,783	pipeTfEhsbCf	HSB	57,072	indpTEstb	STB	72,542	
	pipeTpEhsbCf	HSB	90,416	indpTpEsbCf	SB	86,887	indpTfEhsbCf	HSB	62,586	indpTpEsbCf	SB	72,606	
Z4	indpTpEhsbCf	HSB	91,699	pipeTfEsbCf	SB	86,963	pipeTpEhsbCf	HSB	63,241	pipeTfEsbCf	SB	72,748	
	indpTfEsbCf	SB	91,886	pipeEsbTefCf	SB	91,600	indpTfEsbCf	SB	65,161	indpTfEhsbCf	HSB	82,998	
	indpTpEsbCf	SB	91,930	indpTfEhsbCf	HSB	91,829	pipeEsbTepCf	SB	70,014	indpTfEsbCf	SB	83,060	
	indpTfEhsbCf	HSB	92,062	indpTfEsbCf	SB	92,834	pipeEsbTefCf	SB	70,894	pipeEsbTefCf	SB	85,263	

Cores usadas

- ET Intercalado (FTRC)
- Pipeline TE-C
- Pipeline T-E-C
- Pipeline E-T-C
- Independente T/E/C
- Independente TE/C



- STB-index
- HSTB-index
- SB-index
- HSB-index



A zona de variação Z2 contém opções de processamento de todas as classes menos a ET Intercalado (FTRC) e em posições muito variadas tirando a

possibilidade de conclusões mais concretas. As perdas de desempenho das opções de processamento nesta zona de variação variaram entre 10,84% e 85,35%.

As zonas de variação Z3 e Z4 se mostraram misturadas entre as opções de processamento cujas classes foram baseadas em *pipeline* e, no processamento independente dos predicados. Tal mistura não nos permite fazer inferências sobre o comportamento entre as classes e seus índices base. As perdas de desempenho nesta zona de variação ficaram entre 51,31% e 92,83%. Opções de processamento destas zonas de variação devem ser preteridas quando do desenvolvimento de novos índices pois a principal característica delas frente às mudanças de tipos de dado foi a imprevisibilidade.

A variação do tipo de dado espacial na seletividade alta (S3) feita nesta análise não afetou o resultado obtido pelas classes de processamento vencedoras (*Pipeline* TE-C e ET Intercalado (FTRC)) e a opção *pipeEhsbTepCf* (da classe *Pipeline* E-T-C) mas afetou o relacionamento entre as classes principalmente nas zonas de variação inferiores da Tabela 35 (a partir da zona de variação Z2). A exemplo das análises anteriores, podemos apenas elencar as classes (e opção de processamento) citadas como vencedoras neste parágrafo como tendo um comportamento estável frente às alterações de tipo de dados. Da mesma maneira, o *baseline* de visão materializada não manteve um comportamento estável entre os tipos de dados e reflete a instabilidade das outras opções de processamento. A conclusão desta análise é que o uso do *baseline* de visão materializada pode oferecer riscos ao desempenho esperado das consultas e que a implementação de opções de processamento das classes que se mantiveram estáveis frente à variação do tipo de dado espacial é importante para se obter um bom desempenho no processamento de consulta STOLAP.

6.3.6 Conclusões

Esta análise do comportamento das opções de processamento para cada seletividade frente à mudança do tipo de dado espacial nos leva a concluir que as opções de processamento pertencentes à classe de processamento ET Intercalado (FTRC) e *Pipeline* TE-C foram as únicas que se comportaram de maneira estável em todas as seletividades sempre aparecendo na primeira zona de variação. Presença constante na segunda zona de variação, as opções de processamento baseadas

nas classes de processamento *Pipeline* T-E-C e *Pipeline* E-T-C obtiveram um desempenho intermediário, mas seu comportamento não foi tão estável (como esperava-se) devido às diferentes características de processamento do predicado espacial. Já as opções de processamento baseadas nas classes Independente T/E/C e Independente TE/C tiveram poucas exceções aparecendo na segunda zona de variação por apresentarem um baixo desempenho com relação às demais classes e frente ao *baseline* de visão materializada. O uso do *baseline* de visão materializada mostrou-se efetivo em alguns casos, particularmente com o tipo de dado linha da tabela de ruas devido à sua simplicidade com linhas horizontas e verticais somente, mas oscilou bastante para outros tipos de dados e apresentou baixo desempenho em muitas configurações.

CAPÍTULO 7

CONCLUSÕES

Após discorrer sobre a pesquisa em todos os seus detalhes, resultados e análises, nos resta juntar todas as informações e concluir sobre as hipóteses a serem provadas ou refutadas. Além disso, este capítulo vai mostrar as contribuições e limitações desta dissertação para, em seguida, projetar os passos futuros para possível continuidade desta pesquisa.

7.1 Considerações Finais

À luz das análises realizadas no Capítulo 6 - e das hipóteses formuladas na Seção 5.2, pode-se inferir o que se segue:

H1: A seletividade de um predicado (de uma consulta STOLAP) afeta o custo de processamento de uma consulta STOLAP. Quanto maior a seletividade de um predicado, maior a quantidade de tuplas selecionadas em uma tabela de dimensão e conseqüentemente maior o custo da junção requerida para se processar o esquema estrela de um data warehouse, no caso da consulta STOLAP ser processada sem a ajuda de índices. Quando a consulta STOLAP é apoiada com o uso de índices, quanto maior a seletividade de um predicado, maior a quantidade de elementos a serem processados no índice e conseqüentemente maior o custo para se percorrer vários ramos e nós de uma árvore n-ária ou para se processar diferentes entradas em vetores baseados em um índice bitmap (por exemplo, para o índice SB-index).

Conclusão: Como se pode observar pela evolução dos tempos médios obtidos pelas opções de processamento em cada seletividade, a hipótese H1 está confirmada. Pode-se também afirmar que ela também é verdadeira para os resultados parciais, uma vez que as opções de processamento vencedoras utilizaram o processamento em pipeline e, as opções que puderam se privilegiar de índices baseados em hierarquias, conseguiram ainda melhores resultados.

H2: No processamento de uma consulta STOLAP, o uso de um índice é apropriado somente para pequenos valores de seletividade para os predicados que o índice apoia. Uma alta seletividade não favorece o uso de um índice. O valor a ser considerado como baixo ou alto para a seletividade é empírico e varia conforme o tipo de predicado. Por exemplo, para predicados espaciais, Siqueira (2012) considerou consultas retornando menos de 200 registros (dentro um total de 1.000.000 de registros) na fase de filtragem para indicar uma baixa seletividade.

Conclusão: A partir das análises realizadas no Capítulo 6 - pode-se confirmar a hipótese H2. Tal comportamento foi observado mais facilmente no tipo de dado pontos vindos da tabela de clientes (caddress), o qual confirmou a quantidade de registros observada por Siqueira et al. (2012) e mostrou que as opções de processamento baseadas no HSTB-index (baseado em hierarquias de MBRs e que proporcionam maior poda de registros em baixas seletividades) foram ultrapassadas pelas opções baseadas no STB-index (implementadas usando um vetor de MBRs) a partir da seletividade S3 – a primeira a ultrapassar os 200 registros. Já para os outros tipos de dados tivemos outros fatores influenciando o tempo de processamento (como a quantidade de registros das tabelas alvo e o cálculo da validade temporal dos MBRs), o que também mudou o comportamento das opções de processamento baseadas no STB-index e, assim, as fez não ultrapassarem suas opções relativas utilizando hierarquias de MBRs apesar de diminuir sua distância em termos de porcentagens de desempenho. Tal observação confirma a hipótese por mostrar que o valor limite observado anteriormente existe, mas que ele é dependente de outros fatores que influenciam os tempos de processamento e, portanto, vai receber influência da demanda de processamento de cada opção.

H3: A seletividade dos predicados convencional, espacial e temporal na maioria dos casos é diferente e em alguns casos a seletividade dos predicados espacial e temporal é bem menor do que a seletividade do predicado convencional. Portanto, processar primeiro os predicados espacial e temporal e por fim o predicado convencional gerará um melhor desempenho no processamento de consultas STOLAP.

Conclusão: Pela observação do comportamento das opções de processamento da classe independente, pode-se dizer que a hipótese H3 está confirmada. Em face das técnicas de processamento utilizadas neste trabalho, se o processamento do predicado convencional da consulta STOLAP fosse feito a priori

precisaríamos repetir o processamento da parte convencional na formação da resposta final da consulta e, assim, entraríamos na mesma situação relatada para as opções de processamento independentes onde o processamento duplicado do predicado convencional resultou em piores resultados na grande maioria dos casos.

H4: O custo do processamento do predicado espacial tende a ser muito maior do que o custo do processamento do predicado temporal. Isto ocorre em função do alto custo para se processar dados espaciais compostos geralmente por muitos vértices (pontos) e com a execução de operações complexas para se detectar relacionamentos topológicos, mesmo para valores bem menores de seletividade para o predicado espacial. Portanto, processar primeiramente o predicado temporal para atuar como filtro para o processamento do predicado espacial é a melhor ordem de processamento destes dois predicados.

Conclusão: Observando o comportamento das opções em pipeline e considerando a forma de formação das seletividades (privilegiando a seletividade espacial como decisiva para a limitação de candidatos), pode-se dizer que nada podemos concluir com relação à hipótese H4. Podemos dividir as análises das opções de processamento que utilizam pipeline em três categorias: as que utilizam os dados temporais e espaciais diretamente da estrutura do índice (baseadas no HSTB-index e STB-index); as que utilizam o software de indexação bitmap no processamento do predicado temporal e as que utilizam o sistema gerenciador de banco de dados para processamento do predicado temporal. A primeira categoria compreende as opções de processamento pipeTEhstb e pipeTEstb (que fazem a limitação temporal a priori) comparadas contra a suas relativas fazendo a limitação espacial a priori (intmTEhstb e intmTEstb) e onde, das 32 combinações de seletividade e tipo de dado espacial analisadas, 17 que utilizaram a limitação temporal a priori foram mais rápidas contra 15 das que utilizaram a limitação espacial a priori. A segunda categoria compreende as opções pipeTfEhsbC1 e pipeTfEsbC1 comparadas às suas relativas fazendo a limitação espacial a priori (pipeEhsbTefC1 e pipeEsbTefC1) e onde, das 32 combinações analisadas, houve grande vantagem para as opções utilizando a limitação temporal a priori (21 a 11) mas cuja importância ficou diminuída por estas opções estarem sempre entre as opções de processamento com desempenho intermediário para baixo. A terceira e última categoria compreende as opções de processamento pipeTpEhsbCf e pipeTpEsbCf comparadas às suas relativas com processamento espacial a priori

(pipeEhsbTepCf e pipeEsbTepCf) e onde o processamento espacial levou ampla vantagem (7 a 25) mas que refletiu a maneira de formação das seletividades espaço-temporais (privilegiando a limitação espacial) e, assim, não permitiu que se concluísse a favor de nenhum dos predicados.

H5: O processamento independente dos predicados convencional, espacial e temporal com a geração de listas de resultados parciais contendo os IDs das tuplas que satisfazem o predicado e o subsequente processamento das listas de resultados parciais por meio de processamento cosequencial (i.e. interseção e união de listas de IDs para computar operadores AND e OR) possui custo maior do que usar um *pipeline* de processamento para os predicados convencional, espacial e temporal, onde cada predicado atua como um filtro para o próximo predicado do *pipeline*.

Conclusão: Em face dos resultados obtidos e documentados nas análises pode-se dizer que a hipótese H5 está confirmada. As opções de processamento independentes precisaram executar a limitação convencional em duplicidade (uma para a limitação convencional em si e outra para a obtenção da resposta da consulta) e, assim, influenciaram negativamente no tempo de processamento das consultas STOLAP. Outro fator a se considerar é a maneira de formação das seletividades onde o processamento independente terá sempre uma grande quantidade na lista de candidatos do predicado temporal. Se invertermos a maneira de formação da seletividade seria a lista de candidatos espaciais que se degradaria. Ainda que conseguíssemos equilibrar a quantidade de candidatos de modo a termos a mesma seletividade espacial e temporal, o processamento em duplicidade do predicado convencional faria o desempenho das consultas cair consideravelmente a ponto de não conseguirem alcançar as consultas em pipeline.

H6: Processar uma consulta STOLAP considerando o filtro da parte temporal em conjunto com a parte espacial (i.e., já limitando os seus resultados antes da fase de refinamento do predicado espacial) proverá melhor desempenho que o processamento separado dos predicados temporal e espacial.

Conclusão: A partir da análise das opções de processamento vencedoras para as configurações espaço temporais utilizando linhas para ruas e polígonos para cidades (“street” e “city”) pode-se confirmar a hipótese H6. Chamando de opções vencedoras as que tiveram presença constante na zona de variação Z1 das análises com variação de seletividade ou de tipo de dado espacial, quase todas as configurações citadas possuem opções de processamento das classes ET

Intercalado (FCTR) e Pipeline TE-C as quais possuem por característica a limitação dos predicados espaciais e temporais antes da fase de refinamento da consulta.

Após todas as considerações das hipóteses feitas, conclui-se que a utilização de índices especializados para processamento de consultas espaço-temporais é recomendada em detrimento de se utilizar unicamente o processamento com visões materializadas. Conclui-se também que, quanto mais simples de verificar e maior a limitação de registros candidatos no processamento de cada um dos predicados, melhor será o desempenho do processamento da consulta STOLAP sendo processada. É recomendação deste pesquisador que o processamento em pipeline seja utilizado conjuntamente com a implementação de índices especializados cuja informação de validade temporal e limitação espacial (na forma de um MBR ou informação similar para limitação espacial) estejam presentes na estrutura do índice a ser utilizado. Um último passo a se considerar na escolha do índice mais apropriado para uma aplicação é a seletividade alvo para se escolher as melhores opções de processamento na seletividade escolhida.

7.2 Contribuições e Limitações

Este trabalho proporciona três contribuições principais.

A primeira, e mais importante, é a avaliação e recomendação das melhores práticas e opções de processamento dos predicados espacial, temporal e convencional em consultas STOLAP frente a várias configurações de tipos de dados e seletividades de consultas.

A segunda é a extensão do índice HSB-index para uma versão espaço-temporal chamada HSTB-index que veio complementar as técnicas de processamento já utilizadas e desenvolvidas pelo grupo de pesquisa de banco de dados da UFSCar.

A terceira foi a proposta da inversão do processamento espacial e temporal nas opções de processamento que utilizaram o STB-index e o HSTB-index como base dando origem às opções de processamento da classe ET Intercalado (FTRC). Tais opções de processamento foram muito importantes por evidenciar que a

seletividade de cada predicado em separado não tem grande influência no resultado geral do processamento em pipeline.

Pode-se citar cinco como as maiores limitações deste trabalho.

A primeira foi a utilização de apenas uma consulta dentre as várias propostas pelo benchmark Spatial SSB para realizar as análises das configurações espaço-temporais.

A segunda está na maneira de construção das seletividades que serviu para termos uma análise detalhada nos vários tipos de dados espaciais em vários níveis de seletividade, mas limitou as análises pela sua influência nos resultados.

A terceira compreende o formato das geometrias utilizadas no DWET cujas linhas foram sempre horizontais ou verticais e os polígonos possuíam pouca variação de geometria. Tal fato diminuiu o efeito de “dead space” quando da obtenção dos MBRs e favoreceu em certo grau tais tipos de dados quando do processamento do predicado espacial.

A quarta consiste na maneira de implementação do índice HSTB-index, onde a informação de validade temporal dos MBRs foi incluída somente nos nós folha limitando, assim, a poda de candidatos em níveis mais altos da hierarquia. Mesmo assim, a implementação ajudou algumas opções de processamento a obterem bons resultados de desempenho.

Por último temos o fato de que a escalabilidade dos dados não foi testada. Os testes se limitaram ao fator de escala 1 do benchmark SSB (Star Schema Benchmark) que serviu de base para o Spatial SSB e cuja extensão espacial feita por Tsuruda (2013) foi utilizada neste trabalho.

7.3 Trabalhos futuros

Vê-se que as limitações deste trabalho podem ser a semente para os trabalhos futuros.

A análise de outras consultas do benchmark Spatial SSB juntamente com mudanças no fator de escala das consultas pode ser considerada em futuros trabalhos a fim de se confirmar as observações e recomendações aqui propostas.

Um outro possível trabalho é a variação da formação da seletividade espaço temporal de modo a se privilegiar a limitação temporal em detrimento da limitação espacial para avaliar se o mesmo comportamento de independência é observado entre as opções de processamento das consultas.

A construção de um DWET com geometrias mais realísticas pode ser também um dos trabalhos futuros. Tais geometrias compreenderiam retas oblíquas para as ruas e polígonos com maiores reentrâncias para as cidades aumentando o efeito de “dead space” e, conseqüentemente, a confiabilidade dos resultados.

Um quarto trabalho possível é a melhoria do índice HSTB-index o qual pôde ter a informação temporal adicionada somente nos nós folha da árvore. Tal ponto de inclusão limitou a capacidade de poda apenas à parte espacial do índice. A melhoria consiste em incluir a informação de validade temporal dos MBRs também nos nós intermediários para que estes também possam auxiliar na poda dos registros candidatos.

Pensando na recomendação de que as maiores limitações devem ser feitas a priori no processamento pipeline (seja por poda de registros ou para se indicar a seletividade alvo), um quarto trabalho futuro pode ser a avaliação das seletividades espaciais e temporais antes do processamento em si de forma a sempre utilizar o índice com melhor desempenho e se ter uma situação onde sempre se ganha. Por exemplo, se a seletividade alvo for muito baixa, o otimizador de consultas pode se adaptar e utilizar as opções de processamento que melhor se adaptaram a esta.

REFERÊNCIAS

- BENTLEY, J. L. Multidimensional binary search trees used for associative searching. **Communications of the ACM**, v. 18, n. 9, p. 509-517, 1975.
- BIMONTE, S.; TCHOUNIKINE, A.; MIQUEL, M. **AGILE International Conference on Geographic Information Science**. 10th AGILE International Conference on Geographic Information Science. Aalborg (Denmark): [s.n.]. 2007. p. 11.
- CÂMARA, G. et al. **Anatomia de Sistemas de Informação Geográfica**. Campinas, SP: Instituto de Computação - UNICAMP, 1996. 193 p.
- CARÉLO, C. C. M. et al. Slicing the metric space to provide quick indexing of complex data in the main memory. **Information Systems**, v. 36, n. 1, p. 79-98, mar 2011.
- CASANOVA, M. E. A. **Bancos de dados geográficos**. 2a. ed. Curitiba, PR: MundoGEO, 2005.
- CHÁVEZ, E.; GONZALO NAVARRO, R. B.; MARROQUÍN, J. L. Searching in Metric Spaces. **ACM Computing Surveys**, v. 33, p. 273-321, September 2001.
- CIFERRI, R. R. **Análise da influência do fator distribuição espacial dos dados no desempenho de métodos de acesso multidimensionais**. Centro de Informática, Universidade Federal de Pernambuco. Recife, PE, p. 246. 2002.
- EDER, J.; KONCILIA, C.; MORZY, T. **The COMET Metamodel for Temporal Data Warehouses**. INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION SYSTEMS ENGINEERING (CAiSE). Berlin / Heidelberg, Germany: Springer. 2006. p. 83-99.
- ELMASRI, R. E.; NAVATHE, S. **Sistemas de Banco de Dados**. [S.l.]: Addison-Wesley, 2005. 744 p.
- FERREIRA, A. C. F.; CAMPOS, M. L.; TANAKA, A. K. **An Architecture For Dimensional And Spatial Analyses Integration**. WORLD MULTICONFERENCE ON SYSTEMATICS, CYBERNETICS AND INFORMATICS. Orlando, FL: [s.n.]. 2001.
- FOLK, M. J.; ZOELLICK, B. **File Structures: A Conceptual Toolkit**. 2nd. ed. Boston, MA: Addison-Wesley, 1991. 590 p.

GAEDE, V.; GÜNTHER, O. Multidimensional access methods. **ACM Computing Surveys (CSUR)**, New York, NY, v. 30, n. 2, p. 170-231, jun 1998.

GOLFARELLI, M.; RIZZI, S. A Survey on Temporal Data Warehousing. **International Journal of Data Warehousing and Mining (IJDWM)**, v. 5, n. 1, p. 1-17, 2009.

GÜTING, R. H. et al. A foundation for representing and querying moving objects. **ACM Transactions on Database Systems**, v. 25, n. 1, p. 1-42, mar 2000.

GUTTMAN, A. **R-Trees: A Dynamic Index Structure for Spatial Searching**. Annual Meeting ACM SIGMOD. Boston, MA,: [s.n.]. 1984. p. 47-57.

HARINARAYAN, V. . R. A. . U. J. D. **Implementing Data Cubes Efficiently**. Proceedings of the 1996 International Conference on Management of Data. [S.I.]: ACM Press. 1996. p. 205-216.

INMON, W. H. **Building the Data Warehouse**. 4th. ed. Indianapolis, IN: Wiley, 2005. 576 p.

JENSEN, S. C. E. A. A consensus glossary of temporal database concepts. **SIGMOD Record**, New York, NY, v. 23, n. 1, p. 52-64, mar 1994.

KIMBAL, R.; ROSS, M. **The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling**. 2nd. ed. New York, Ny: Wiley, 2002. 464 p.

KLINE, N. An update of the temporal database bibliography. **SIGMOD Record**, New York, NY, v. 22, n. 4, p. 66-80, dez 1993.

KOUBA, Z.; MATOUŠEK, K.; MIKŠOVSKÝ, P. **Novel Knowledge Discovery Tools In Industrial Applications**. INTELLIGENT METHODS FOR QUALITY IMPROVEMENT IN INDUSTRIAL PRACTICE. Praga, República Checa: [s.n.]. 2002. p. 143.

MEDEIROS, V. N. et al. **Especificação e Avaliação de Algoritmos para Processamento Multidimensional-Geográfico**. SIMPÓSIO BRASILEIRO DE BANCO DE DADOS (SBBDD). Campinas, SP: [s.n.]. 2008. p. 61-75.

MENDELZON, A.; VAISMAN, A. **Temporal queries in OLAP**. INTERNATIONAL CONFERENCE ON VERY LARGE DATABASES (VLDB). Cairo, Egipt: [s.n.]. 2000. p. 242-253.

MICROSOFT. Microsoft SQL Server Library – MDX. **Microsoft Technet**, s.d. Disponível em: <[http://technet.microsoft.com/en-us/library/aa216767\(v=sql.80\).aspx](http://technet.microsoft.com/en-us/library/aa216767(v=sql.80).aspx)>. Acesso em: 22 fev. 2014.

NASCIMENTO, S. M. et al. **The Spatial Star Schema Benchmark**. BRAZILIAN SYMPOSIUM ON GEOINFORMATICS (GEOINFO). São José dos Campos, SP: Proceedings. 2011. p. 73-84.

NIEVERGELT, J.; HINTERBERGER, H.; SEVCIK, K. The GRID FILE: An Adaptable, Symmetric Multi-Key File Structure. **ACM Transactions on Database Systems (TODS)**, v. 9, n. 1, p. 38-71, 1984.

O'NEIL, P.; GRAEFE, G. Multi-table joins through bitmapped join indices. **ACM SIGMOD Record**, v. 24, n. 3, p. 8-11, 1995.

O'NEIL, P. A. O. E. A. C. X. A. R. S. The Star Shema Benchmark and Augmented Fact Table Indexing. In: NAMBIAR, R. A. P. M. **Performance Evaluation and Benchmarking**. Berlin, Heidelberg: Springer-Verlag, 2009. p. 237-252.

PAPADIAS, D. et al. **Efficient OLAP Operations in Spatial Data Warehouses**. INTERNATIONAL SYMPOSIUM ON ADVANCES IN SPATIAL AND TEMPORAL DATABASES (SSTD). Redondo Beach, CA.: Springer. 2001. p. 443-459.

PAPADIAS, D. et al. **Indexing Spatio-Temporal Data Warehouses**. INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE). San Jose, CA: IEEE. 2002. p. 166-175.

RAKOTO-RAVALONTSALAMA, N. **Partial and total order relations for supervisory control of hybrid systems**. INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION. Antibes - Juan les Pins, France: IEEE. 2001. p. 675-678.

RIVEST, S. et al. SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. **Journal of International Society for Photogrammetry and Remote Sensing (ISPRS)**, v. 60, p. 17-33, 2005.

RIVEST, S.; BÉDARD, Y.; MARCHAND, P. Toward better support for spatial decision making: Defining the characteristics of spatial on-line analytical processing (SOLAP). **GEOMATICA**, v. 55, n. 4, p. 539-555, 2001.

RODDICK, J. A. Survey of Schema Versioning Issues for Database Systems. **Information and Software Technology**, v. 37, n. 7, p. 383-393, 1995.

SAMET, H. The Quadtree and related hierarchical data structures. **ACM Computing Surveys (CSUR)**, New York, NY, v. 16, n. 2, p. 187-260, jun 1984.

SAMPAIO, M. C.; BAPTISTA, C.; SOUSA, A. . N. F. Enhancing Decision Support Systems With Spatial Capabilities. In: _____ **Intelligent Databases: Technologies and Applications**. Hershey, PA: IGI Global, 2006. Cap. 5, p. 94-116.

SCOTCH, M.; PARMANTO, B. Development of SOVAT: A numerical–spatial decision support system for community health assessment research. **International Journal of Medical Informatics**, v. 75, p. 771-784, 2006.

SHEKHAR, S. et al. **Map Cube: A Visualization Tool for Spatial Data Warehouses**. Geographic Data Mining and Knowledge Discovery. Bristol, PA: Taylor & Francis, Inc. 2001. p. 73.

SIQUEIRA, T. L. L. **SB-INDEX: um índice espacial baseado em bitmap para data warehouse geográfico**. Departamento de Computação, Universidade Federal de São Carlos, São Carlos. São Carlos, SP, p. 118. 2009.

SIQUEIRA, T. L. L. et al. The SB-index and the HSB-index: efficient indices for spatial data warehouses. **Geoinformatica**, Hingham, MA, v. 16, n. 1, p. 165-205, jan 2012.

SNODGRAS, T. TSQL2 and SQL3 Interactions, s.d. Disponível em: <<http://www.cs.arizona.edu/people/rts/sql3.html>>. Acesso em: 19 jan. 2014.

SPACCAPIETRA, S. et al. A conceptual view on trajectories. **Data and Knowledge Engineering**, v. 65, p. 126-146, nov 2007.

STEFANOVIC, N.; HAN, J.; KOPERSKI, K. Object-based selective materialization for efficient implementation of spatial data cubes. **IEEE Transactions on Knowledge and Data Engineering**, v. 12, n. 6, p. 938-958, nov 2000.

TSURUDA, R. M. **STB-Index: Um Índice Baseado em Bitmap para Data Warehouse Espaço-Temporal**. Departamento de Computação, Universidade Federal de São Carlos. São Carlos, p. 81. 2013.

VAISMAN, A.; IZQUIERDO, A.; KTENAS, M. **Web-enabled Temporal OLAP**. FOURTH LATIN AMERICAN WEB CONGRESS (LA-WEB'06). Cholula, México: [s.n.]. 2006. p. 220-229.

VAISMAN, A.; ZIMÁNYI, E. **What is Spatio-Temporal Data Warehousing?** INTERNATIONAL CONFERENCE ON DATA WAREHOUSING AND KNOWLEDGE DISCOVERY (DaWaK). Linz, Austria: [s.n.]. 2009. p. 9-23.

WU, K.; OTOO, E. J.; ARIE, S. Optimizing Bitmap Indices With Efficient Compression. **ACM Transactions on Database Systems**, v. 31, n. 1, p. 1-38, 2006.

WU, K.; STOCKINGER, K.; SHOSHANI. **Breaking the Curse of Cardinality on Bitmap Indexes**. International Conference on Scientific and Statistical Database Management. Hong Kong: Springer-Verlag. 2008. p. 348-365.

WU, M.-C.; BUCHMANN, A. P. **Encoded Bitmap Indexing for Data Warehouses**. INTERNATIONAL CONFERENCE ON DATA ENGINEERING. Orlando, FL: IEEE Computer Society. 1998. p. 220-230.