Tárik de Melo e Silva Rocha

GeoCube: Representação, Computação, e Visualização de Cubos Espaciais

Ouro Preto
2012

Tárik de Melo e Silva Rocha

GeoCube: Representação, Computação, e Visualização de Cubos Espaciais

Dissertação apresentada ao Departamento de Computação da Universidade Federal de Ouro Preto, para a obtenção de Título de Mestre em Ciência da Computação, na área de Sistemas de Computação.

Orientador: Joubert de Castro Lima

Ouro Preto
2012

R672g Rocha, Tárik de Melo e Silva.
GeoCube [manuscrito] : representação e visualização de cubos espaciais / Tárik de Melo e Silva Rocha. – 2012.
x, 79 f.: il. color.; grafs.; tabs.; mapas.

Orientador: Prof. Dr. Joubert de Casotro Lima.
Coorientador: Prof. Dr. Tiago Garcia de Senna Carneiro.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Departamento de Computação. Programa de Pós-graduação em Ciência da Computação.
Área de concentração: Ciência da Computação

1. Banco de dados - Modelo OLAP - Modelo SOLAP - Teses. 2. Cubo de dados - Teses. 3. Programação paralela (Computação) - Teses. 4. Sistemas de informação geográfica (SIG) - Teses. I. Universidade Federal de Ouro Preto. II. Título.

CDU: 004.652:910.26

Catalogação: sisbin@sisbin.ufop.br

Ata da Defesa Pública de Dissertação de Mestrado

Aos 14 dias do mês de dezembro de 2012, às 14 horas na Sala de Seminários do Departamento de Computação do Instituto de Ciências Exatas e Biológicas (ICEB), reuniram-se os membros da banca examinadora composta pelos professores: **Prof. Dr. Joubert de Castro Lima (presidente e orientador), Prof. Dr. Tiago Garcia de Senna Carneiro, Prof. Dr. Álvaro Rodrigues Pereira Júnior e Prof. Dr. Jugurta Lisboa Filho**, aprovada pelo Colegiado do Programa de Pós-Graduação em Ciência da Computação, a fim de arguirem o mestrando **Tárik de Melo e Silva Rocha**, com o título **"GeoCube: Representação, Computação e Visualização de Cubos Espaciais"**. Aberta a sessão pelo presidente, coube ao candidato, na forma regimental, expor o tema de sua dissertação, dentro do tempo regulamentar, sendo em seguida questionado pelos membros da banca examinadora, tendo dado as explicações que foram necessárias.

Recomendações da Banca:

( ) Aprovada sem recomendações

( ) Reprovada

(X) Aprovada com recomendações: Alterações sugeridas pela BANCA (estruturais e conceituais)

Banca Examinadora:

Prof. Dr. Joubert de Castro Lima

Prof. Dr. Tiago Garcia de Senna Carneiro

Prof. Dr. Álvaro Rodrigues Pereira Júnior

Prof. Dr. Jugurta Lisboa Filho

Prof. Dr. Haroldo Gambini Santos
Coordenador do Programa de Pós-Graduação em Ciência da Computação
DECOM/ICEB/UFOP

Ouro Preto, 14 de dezembro de 2012.

"Devemos julgar um homem mais pelas suas perguntas
que pelas respostas." (Voltaire).

.....

Agradecimentos

Aos meus pais por todo apoio, sem eles este trabalho não seria possível. A Fundação de Amparo à Pesquisa do estado de Minas Gerais pela bolsa de estudos. Ao professor Joubert de Castro Lima meus sinceros agradecimentos pela orientação, confiança, oportunidade de pesquisa e principalmente pela paciência. Aos amigos de turma e de pesquisa que me ajudaram muito e compartilharam os momentos difíceis e ao professor co-orientador Tiago Garcia de Senna Carneiro. Agradeço também a todos aqueles que contribuíram para a realização deste trabalho.

Resumo

A tecnologia SOLAP (*Spatial On-Line Analytical Processing*) oferece a junção das funcionalidades OLAP (*Online Analytical Processing*) e SIG (Sistema de Informação Geográfica) e abre caminho para uma nova categoria de aplicações que fornecem suporte a manipulação, processamento e navegação em dados espaço-temporais organizados hierarquicamente. Os dados em um sistema OLAP são armazenados como cubos e estes são organizados segundo o conceito de dimensões, medidas e hierarquias. A materialização de um cubo de dados possui ordem de complexidade exponencial em relação ao consumo de memória e tempo de execução. Quando associamos informações espaciais ao cubo, a demanda de memória e processamento aumenta, tornando mais difícil a tarefa de oferecer respostas rápidas ao usuário. Atualmente, poucos trabalhos foram publicados na representação, computação e consulta de cubos espaço-temporais completos. As técnicas apresentadas como materialização seletiva, materialização baseada em aproximações e estruturas para indexação de regiões não oferecem soluções para computação de cubos completos e muitas vezes não podem ser aplicadas para uma grande variedade de funções de agregação espaciais. Arquiteturas de computadores baseadas em endereçamento compartilhado também não são utilizadas nos trabalhos correlatos. Nos trabalhos correlatos as hierarquias são especificadas manualmente por especialistas do domínio e muitas vezes as soluções limitam a quantidade e a variedade de medidas espaciais e não espaciais no cubo. Diante de tal cenário, é proposta uma abordagem para representação, computação e

consulta de cubos espaço-temporais completos ou parciais, chamada GeoCube. A abordagem GeoCube pode ser executada em máquinas com múltiplos núcleos de processamento. Múltiplas funções de agregação espacial e estatística podem ser combinadas. Também é proposta uma abordagem para formação automática de hierarquias através de regras de vizinhança entre seus objetos espaciais. As regras de vizinhança podem ser definidas pelo usuário e executadas pelo GeoCube. O GeoCube já possui vizinhança nxn, onde n>1. Testes comparativos com um sistema implementado usando PostGIS e vistas materializadas mostram que a GeoCube consegue computar cubos de dados espaciais em até 1/6 do tempo gasto pela tecnologia PostGis em uma máquina com 8 núcleos de processamento.

Palavras-chave: OLAP, SOLAP, Cubo de Dados

Abstract

SOLAP (Spatial On-Line Analytical Processing) technology offers the junction of OLAP (online analytical processing) and GIS (Geographic Information System) features, enabling the development of new applications designed for hierarchical spatio-temporal data. OLAP systems index data as data cubes. Data cube is a relational operator organized according to the concept of dimensions, hierarchies and measures. The data cube materialization has exponential complexity in terms of memory consumption and runtime. When we associate spatial information in data cube, memory and processing demand increases, turning fast responses to the user a even hard task to solve. Currently, few studies address solutions to spatio-temporal cubes representation, computation and query. Techniques as selective materialization and materialisation-based approaches used to index regions do not offer solutions for computing complete data cubes and often can not be applied to a wide variety of spatial aggregation functions. Computer architectures based on shared address are not used in related work. Faced with this scenario, we propose an approach to represent, compute and query spatio-temporal cubes, called GeoCube. GeoCube can run on machines with multi-core processors. Multiple functions, including many statistical and spatial functions, can be computed in GeoCube. GeoCube also provides a new method to automatic generate spatial hierarchies using any neighborhood calculus. Comparative tests with a system implemented using PostGIS and materialized views show that GeoCube can compute spatial data cubes six times faster on a machine with

eight cores.

Keywords: OLAP, SOLAP, Data Cubes

Lista de Figuras

Lista de Tabelas

Sumário

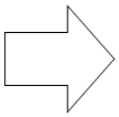
1 *Introdução*

Data Warehouse (DW) é um repositório ou coleção de dados integrado, orientado por assunto, variável com o tempo e não-volátil que oferece suporte a processos de tomada de decisão (Inmon and Hackathorn, 1994). Servidores OLAP (*Online Analytical Processing*) normalmente computam cubos a partir de DWs. Os dados em um sistema OLAP são armazenados como cubos e estes são organizados segundo o conceito de dimensões, fatos, medidas e hierarquias.

O operador cubo de dados, ou simplesmente cubo, foi proposto por Jim Gray em (Gray et al., 1997). Um cubo é a generalização do operador *Group-by* sobre todas as combinações de dimensões, sendo as dimensões compostas de múltiplas hierarquias. A granularidade é o nível de detalhe em que os dados são mantidos no cubo. Cada *Group-by* é chamado de cubóide e corresponde a um conjunto de células ou tuplas sobre as dimensões. Para gerar todas as sumarizações o algoritmo faz o uso do valor ALL. A Figura 1 ilustra o operador cubo implementado como uma tabela composta de tuplas. Neste exemplo há apenas três dimensões e uma medida.

No exemplo da Figura 1 foi utilizado as dimensões Modelo, Ano e Cor. A medida, sempre associada a uma função estatística, é a frequência (COUNT) de

Modelo	Ano	Cor	Venda
Chevy	1990	R	5
Chevy	1990	B	87
Ford	1990	G	64
Ford	1990	B	99
Ford	1991	R	8
Ford	1991	B	7



Modelo	Ano	Cor	Venda
Chevy	1990	B	87
Chevy	1990	R	5
Chevy	1990	ALL	92
Chevy	ALL	B	87
Chevy	ALL	R	5
Chevy	ALL	ALL	92
Ford	1990	B	99
Ford	1990	G	64
Ford	1990	ALL	163
Ford	1991	B	7
Ford	1991	R	8
Ford	1991	ALL	15
Ford	ALL	B	106
Ford	ALL	G	64
Ford	ALL	R	8
ALL	1990	B	186
ALL	1990	G	64
ALL	1991	B	7
ALL	1991	R	8
Ford	ALL	ALL	178
ALL	1990	ALL	255
ALL	1991	ALL	15
ALL	ALL	B	193
ALL	ALL	G	64
ALL	ALL	R	13
ALL	ALL	ALL	270

Figura 1: Operador Cubo de Dados.

veículos vendidos, representada pela coluna Venda. Cubos de dados completos produzem as agregações possíveis sobre o conjunto de tuplas inicial. No exemplo da Figura 1, um cubo completo foi criado a partir de um conjunto de seis tuplas de entrada. O cubo completo resultante possui vinte e seis tuplas, ou seja, enquanto o número de colunas ou dimensões cresce linearmente o número de tuplas ou células em um cubo cresce exponencialmente.

Os sistemas de informações geográficas ou SIG não são desenvolvidos com o objetivo de prover resultados agregados, hierarquizados e rápidos sobre tendências e correlações entre grandes quantidade de dados espaciais. As ferramentas comumente usadas para fazer análise de medidas e dimensões para dados alfanuméricos não suportam medidas e dimensões espaciais. Por isso surgiu uma categoria de ferramentas que combinam dados alfanuméricos e dados espaciais. As ferramentas que oferecem suporte a análise destes tipos de dados são chamadas

de ferramentas SOLAP(*Spatial On-Line Analytical Processing*).

Ferramentas SOLAP geram cubos espaciais. Cubos espaciais possuem medidas, dimensões e hierarquias espaciais e não-espaciais e possuem suporte para agregação de objetos espaciais e não espaciais. Infelizmente, as ferramentas SOLAP mais comuns, descritas no capítulo 3, possuem limitações quanto ao número e tipo de operadores espaciais. A combinação de medidas estatísticas e espaciais é outra limitação dos trabalhos correlatos. A geração de hierarquias espaciais de forma automática, usando para isto regras de vizinhança espaciais variadas, é muito útil ao tomador de decisão, porém não são oferecidas nas abordagens estudadas neste trabalho. Por fim o uso de computadores com múltiplos núcleos toma cada vez mais importância, porém os trabalhos correlatos simplesmente não foram desenhados para tais arquiteturas de computador.

Diante de tal cenário, é proposto o GeoCube, um sistema SOLAP com suporte a múltiplas medidas espaciais e estatísticas, suporte a hierarquização automática de espaços regulares e não regulares segundo quaisquer regra de vizinhança espacial. O GeoCube se mostra muito promissor também em termos de desempenho. Testes comparativos com um sistema implementado usando PostGIS e vistas materializadas mostram que o GeoCube consegue computar cubos de dados espaciais em até 1/6 do tempo gasto pela tecnologia PostGIS.

O restante deste trabalho encontra-se organizado como descrito a seguir: no capítulo 2 é realizada a conceitualização, onde são descritos os conceitos básicos para o correto entendimento do tema abordado neste trabalho. No capítulo 3, são descritos os trabalhos relacionados, sendo apresentadas as principais ferramentas SOLAP existentes, incluindo suas principais características, contribuições e

limitações. No capítulo 4 a ferramenta GeoCube é detalhada. No capítulo 5 são apresentados a experimentação e os resultados obtidos. Por fim, no capítulo 6 são apresentados os trabalhos futuros e as conclusões.

2 Conceitos Básicos

2.1 Armazém de Dados

Data Warehouse ou armazém de dados é uma coleção de dados orientado por assunto, não volátil, variável com o tempo e que pode integrar fontes de dados heterogêneas. Estas características tornam os armazéns de dados úteis para ferramentas de suporte a tomadas de decisão, como OLAP (*Online Analytical Processing*) e Mineração de Dados. Frequentemente, os armazém de dados são mantidos separadamente dos bancos de dados operacionais das organizações que exigem diferentes requisitos de desempenho e funcionalidade. As transações operacionais do dia-a-dia das organizações são suportadas pelas ferramentas OLTP (*Online Transaction Processing*). Ao contrário dos bancos de dados OLTP, o armazém de dados prioriza a persistência de dados históricos, sumarizados e consolidados ao invés de transações individuais. Como estes armazéns possuem dados de longos períodos e de diferentes fontes, seu tamanho é geralmente maior do que o banco de dados operacional. Por fim, os armazéns de dados dão suporte a consultas ad-hoc complexas que podem acessar milhões de registros e executar várias varreduras, junções e agregações. ((Chaudhuri and Dayal, 1997))

Normalmente, um armazém de dados integra dados de fontes heterogêneas

como arquivos binários, objetos serializados, arquivos xml e tabelas relacionais. Técnicas de limpeza e integração de dados são aplicadas para garantir a consistência dos nomes, estruturas das dimensões, medidas, entre outros.

A exploração e organização dos dados de um armazém de dados podem ser feitas por ferramentas OLAP que sumarizam e organizam grandes quantidades de dados em estruturas multidimensionais onde múltiplas hierarquias facilitam a navegação e as estruturas multidimensionais indexam eficientemente os dados.

Dimensões são perspectivas do processo de decisão que descrevem índices, chamados medidas, que desejamos analisar. Estas dimensões possuem pelo menos um atributo que as descrevem. Como exemplo, a dimensão cidade pode possuir os atributos população, data de fundação, nome, sigla, entre outros. Quando os dados são analisados em uma perspectiva multidimensional é assumido um tema central. Este tema central é representado por fatos. Fatos são as medidas que analisamos através das relações entre as dimensões. Os fatos representam o desempenho de um indicador e está sempre associado a uma ou várias dimensões. São exemplos de fatos: temperatura, índice pluvial, preço, entre outros.

O modelo de dados multidimensional pode ser classificado segundo diferentes esquemas. No esquema estrela (*Star-Schema*) há uma tabela de fato no centro, e as tabelas de dimensões ao seu redor. Em esquemas como ilustrado na Figura 2 os fatos são altamente normalizados enquanto as dimensões são desnormalizadas. No exemplo da Figura 2, as tabelas produto, loja e tempo são dimensões enquanto o relacionamento destas três dimensões em torno da medida vendas é a tabela fato. Este esquema pode consumir mais espaço de armazenamento, porém garante uma indexação mais eficiente quando comparado ao esquema flocos de

neve.

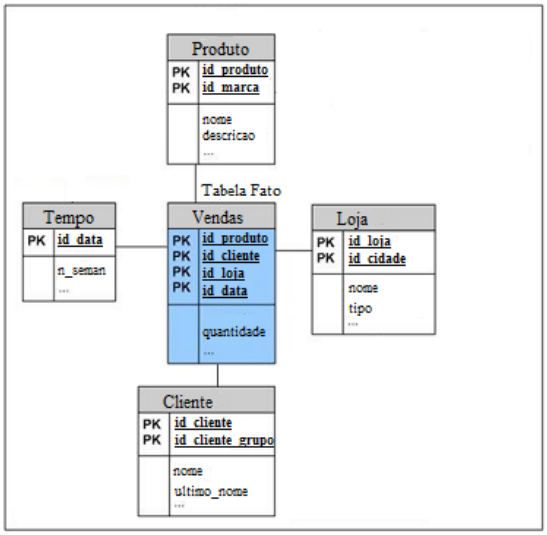


Figura 2: Dimensões e Fatos.

O esquema flocos de neve (*Snowflake*) é uma variação do esquema estrela onde as dimensões são normalizadas e hierarquizadas. Pode ser visto um exemplo deste esquema na Figura 3. As dimensões tempo, endereço, cliente e produto são normalizadas para redução do espaço de armazenamento, porém tais esquemas exigem múltiplas leituras para se obter uma informação.

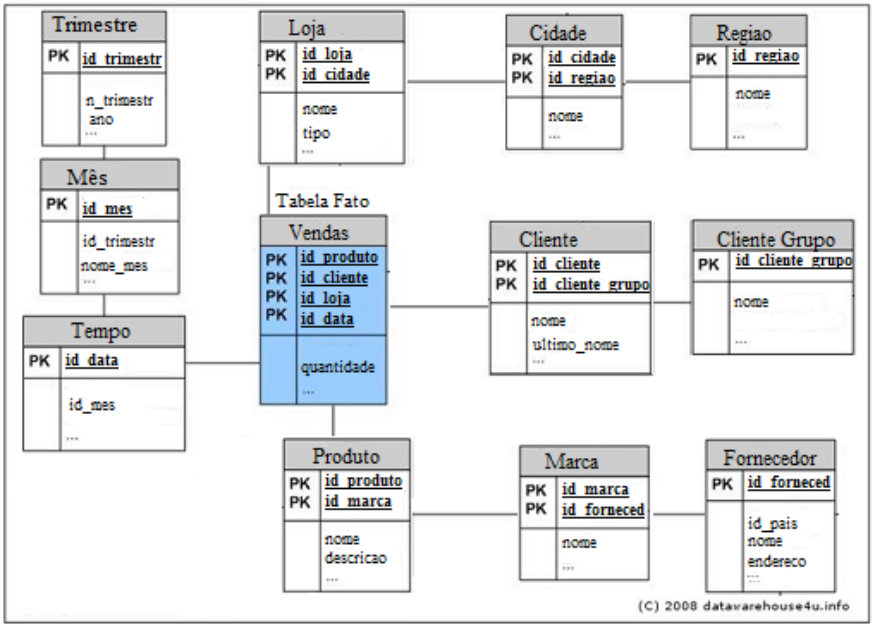


Figura 3: Modelo SnowFlake.

O esquema constelação (*constellation model*) é o mais complexo e é usada por aplicações mais sofisticadas. Este esquema possui várias tabelas de fatos que compartilham as tabelas de dimensão. Este esquema pode ser visto como um conjunto de esquemas estrelas, por isso o nome constelação (Figura 4).

2.2 Cubo de Dados

O modelo multidimensional organiza os dados conceitualmente como um cubo de dados que é uma generalização do operador *Group-By* sobre todas possíveis combinações de dimensões. Cada combinação de dimensões agrega sobre uma medida, portanto um cubo agrega segundo várias granularidades. Cada *Group-by*, chamado cuboide, corresponde a um conjunto de tuplas.

As medidas ou funções agregadas são classificadas em distributiva, algébrica

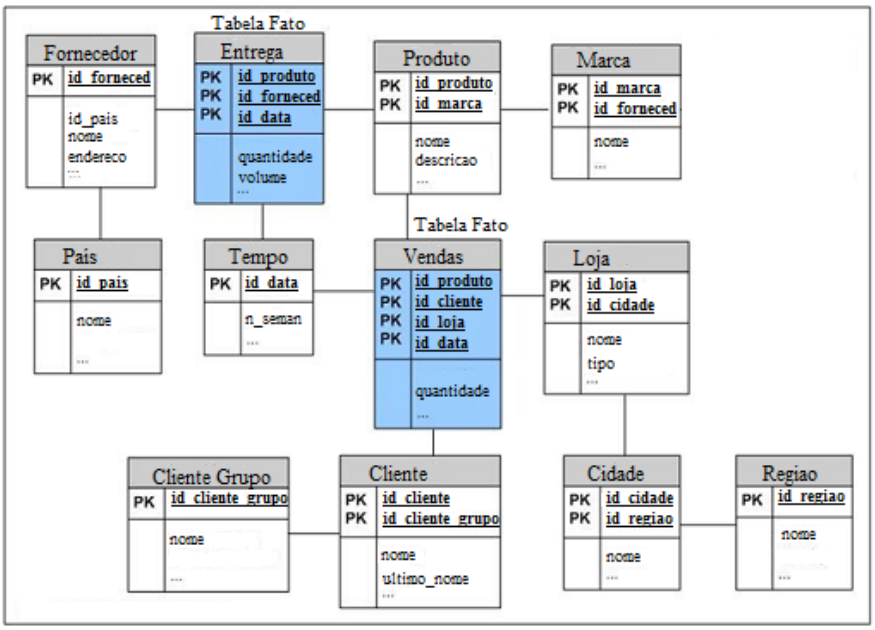


Figura 4: Modelo Constelação.

e holística. Um cubo normalmente suporta múltiplas medidas.

- **Distributiva:** Suponha que os dados estão particionados em n conjuntos. A função de agregação é aplicada em cada partição resultando em n resultados agregados. Se o resultado derivado nas n partições for igual ao resultado da função aplicada ao dado sem particionamento, pode-se dizer que a função pode ser computada de uma maneira distributiva. Neste grupo estão as funções de agregação mais simples de serem calculadas, como count(), sum(), min() e max();

- **Algébricas:** As funções de agregação algébricas podem ser computadas a partir de funções de agregação distributivas. Como, por exemplo, temos a função avg() que pode ser computada a partir das funções sum() e count(), ou seja, é possível reduzir qualquer agregação algébrica em agregações dis-

tributivas. As funções de agregação que pertencem a este grupo são: avg(), min-N() e max-N();

- **Holística:** Não é possível derivar uma função holística em funções algébricas ou distributivas. Trata-se da classe de funções mais complexas de serem computadas. É o tipo de função de agregação mais difícil de ser computada. São exemplos de funções de agregação holísticas: mediana(), moda() e rank().

O operador cubo pode ser usado junto com outros operadores a fim de satisfazer as diferentes necessidades de visualização, navegação ou mesmo reduzir o tamanho do cubo a ser computado. Os principais operadores estão ilustrados na Figura 5 e descritos a seguir:

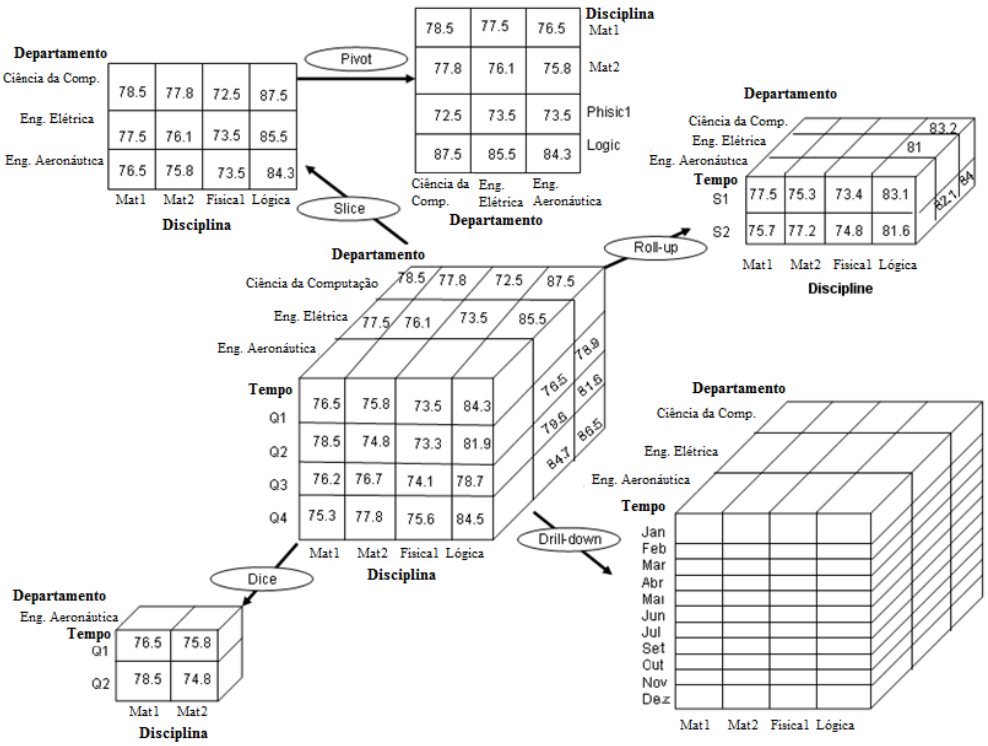


Figura 5: Principais operadores. Fonte: (de Castro Lima, 2009).

- ***Slicing-Dicing:*** Esta operação seleciona uma porção do cubo. Para fazer esta operação, é necessário escolher um valor fixo de uma dimensão e relacionar este valor com todos os valores das outras dimensões;

- ***Pivoting:*** Quando é feito o pivoteamento de um cubo, é feito a rotação das dimensões. Em um cubo com duas dimensões, podemos dizer que ocorre a troca de uma linha por uma coluna;

- ***Roll-Up:*** Algumas dimensões possuem diferentes hierarquias (ou diferentes níveis de granularidade). A operação de *Roll-up* realiza as agregações no cubo de dados subindo no conceito de hierarquia através da remoção lógica de uma ou mais dimensões. Na Figura 5 o resultado da operação de *Roll-up* é a subida no conceito da hierarquia de tempo do nível de trimestre

(Q1, Q2, Q3 e Q4) para semestre (S1 e S2).

- ***Drill-Down:*** Essa operação ocorre em direção oposta a operação de *Roll Up*. Na Figura 5 o resultado da operação de *Drill-Down* temos a descida no conceito de hierarquia de tempo do nível de trimestre (Q1, Q2, Q3 e Q4) para mês (Jan, Fev, Mar, Abr, Mai, Jun, Jul, Ago, Set, Out, Nov e Dez).

2.3 Computação de Cubos

A criação de um cubo é um problema de complexidade exponencial em função do número de dimensões. Sendo assim a materialização de um cubo envolve grande quantidade de memória e tempo de execução. Ao computar cubos espaciais este gasto de tempo e memória aumenta, pois a materialização do cubo envolve o armazenamento de dados espaciais e a computação de operadores espaciais como toca, união, interseção, entre outros.

Um dos primeiros algoritmos na computação de cubos usando estruturas em memória primaria é a abordagem *Multi-way* (Zhao et al., 1998). O algoritmo carrega os dados em vetores e não utiliza tabelas de bancos de dados relacionais. Os vetores são particionados em páginas, onde cada página consegue ser armazenada em memória principal.

Uma vez que os vetores estão particionados em páginas, é possível realizar a compressão dos vetores. Com o auxílio de uma árvore geradora mínima (*minimum memory spanning tree*), o algoritmo usa a estratégia *top-down* para processar o cubo. Em um cenário ideal, a memória deve ser grande o suficiente

para o algoritmo calcular todos as agregações em uma só varredura, evitando múltiplos e custosos acessos aos dados de entrada.

Na Figura 6 ilustra-se a computação do cubo usando a estratégia *top-down*. Inicialmente, se computa a tupla da base de entrada, neste exemplo composta por quatro atributos ABCD. Uma vez gerado a agregação ABCD, pode-se gerar a agregação BCD removendo o atributo A. Este procedimento se repete removendo-se B, C e D. Ao gerar a agregação BCD, também é possível gerar a agregação BD e logo e seguida B. A partir de BCD também é possível gerar BC. De uma forma geral, esta estratégia se beneficia do fato que as agregações mais gerais, ou seja, com menos atributos, podem ser derivadas facilmente a partir das agregações mais específicas, ou seja, com maior número de atributos.

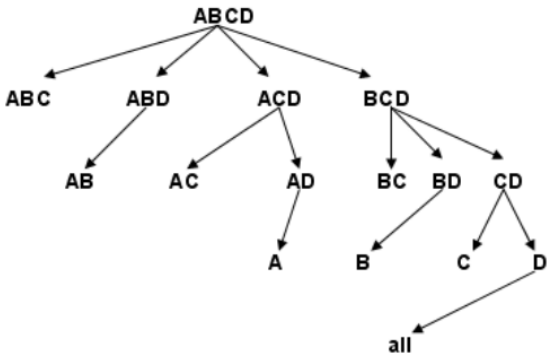


Figura 6: Abordagem *Top-Down* para computação de cubos.

O operador cubo de dados foi estendido em (Beyer and Ramakrishnan, 1999) com a introdução do operador *Iceberg Cube*. *Iceberg Cubes* são cubos que computam somente as porções de *Group-bys* com valores agregados acima de algum limiar mínimo definido pelo usuário. De uma forma geral *Iceberg Cube* computa *Group Bys* que satisfazem a condição de agregação da cláusula SQL HAVING. Na figura 7 o *Iceberg Cube* computado possui agregações com frequência mínima

de 2.

Parte	Local	Cliente
P1	Vancouver	Vance
P1	Calgary	Bob
P1	Toronto	Richard
P2	Toronto	Allison
P2	Toronto	Allison
P2	Toronto	Tom
P2	Ottawa	Allison
P3	Montreal	Anne

Combinação	*Count*
{P1, ANY, ANY}	3
{P2, ANY, ANY}	4
{ANY, Toronto, ANY}	4
{ANY, ANY, Allison}	3
{P2, Toronto, ANY}	3
{P2, ANY, Allison}	3
{ANY, Toronto, Allison}	2
{P2, Toronto, Allison}	2

Iceberg-Cube

Figura 7: Cubo Iceberg. Fonte: (Findlater and Hamilton, 2003)

O algoritmo proposto recebeu o nome de BUC (*Botton-Up Cube*), pois adota a estratégia *Bottom-up*, onde se processa o cubo a partir dos *Group-bys* mais agregados para os *Group-bys* menos agregados, diferente do algoritmo *Multi-way* previamente explicado. A principal vantagem deste algoritmo é realizar a poda das agregações que violam o limiar mínimo o mais cedo possível, sem ter que gastar processamento gerando agregações desnecessárias ao *Iceberg Cube*. A propriedade antimonotônica, descrita em (Agrawal and Srikant, 1994), garante que se um dado conjunto de atributos S é infrequente, todos os ancestrais também são. Os ancestrais de S devem possuir S-n atributos iguais a S, onde $n > 1$. No exemplo da Figura 8 geramos as agregações A, B, C e D separadamente. Depois é gerado a agregação AB. Note que A não fez uso de AB para sua geração, porém se A não é frequente é garantido que não é necessário computar nenhuma agregação que comece com A, diminuindo assim o custo com a computação desnecessária para *Iceberg Cubes*. Infelizmente, nem todos os tipos de medidas obedecem a propriedade antimonotônica. Medidas algébricas e holísticas normalmente não obedecem a propriedade antimonotônica. Outra desvantagem

é que as agregações infrequentes podem se tornar frequentes num servidor OLAP à medida que atualizações são computadas, porém os *Iceberg Cube* não permitem detectar automaticamente tal cenário.

A abordagem Star proposta em (Xin et al., 2007) é a primeira que integra as vantagens das estratégias *bottom-up* e *top-down*. Uma nova representação de cubos é proposta e esta permite que agregações sejam armazenadas em nós intermediários de uma grafo acíclico direcionado (DAG) chamado *star-tree*. Outras abordagens, como Dwarf, C-Cubing, MCG e MDAG, investiram em reduzir o tamanho do grafo, suprindo nós desnecessários da representação do cubo. Neste trabalho é adotado a abordagem Star para geração de agregações de um cubo de dados. A abordagem Star permite a computação de cubos completos ou parciais, incluindo *Iceberg Cubes*. Maiores detalhes da representação Star-Tree no capítulo 4.2.2.

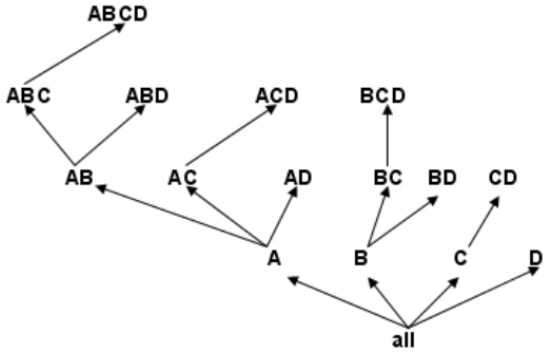


Figura 8: Abordagem Bottom-Up para computação de cubos.

2.4 Representação de dados Espaciais Geográficos

Para entender como é feita a computação de cubos espaciais primeiro é essencial entender como é feita a representação de mapas em sistemas computacionais. Um mapa é representado em camadas onde cada camada representa um tipo de informação. Abaixo há uma breve descrição dos tipos de camadas que são usadas pelos principais Sistemas de Informações Geográficas:

- **Vetorial:** É a camada onde ocorre as agregações dos cubos espaciais, sendo formada por um conjunto de vetores que podem representar pontos, linhas e polígonos. Cada elemento desta camada é chamado geo-objeto e está associado a atributos não espaciais que o descrevem. Cada geo-objeto possui um identificador único que serve como chave de navegabilidade entre a porção espacial e alfanúmerica de polígono, ponto ou linha;

- **Matricial:** Representa imagens que geralmente são coletadas por sensores remotos à bordo de satélites ou aeronaves. Nesta camada o espaço é dividido em células que formam uma matriz. Cada célula pode estar associada a uma classificação prévia que a descreve. Classificar as células exige técnicas avançadas de processamentos de imagens;

- **Espaço Celular Regular :** Pode ser visto como uma generalização de uma estrutura matricial, com a vantagem de que cada célula pode armazenar vários atributos o que faz o manuseio dos dados ser mais simples (Casanova et al., 2005).

A escala de um fenômeno é determinada pela sua resolução ou granularidade e pela extensão em que o fenômeno é caracterizado ao longo do tempo e espaço. Em suma, extensão se refere a área em que o fenômeno acontece enquanto a resolução se refere a precisão usada nas medidas realizadas sobre o fenônemo (White and Running, 1994).

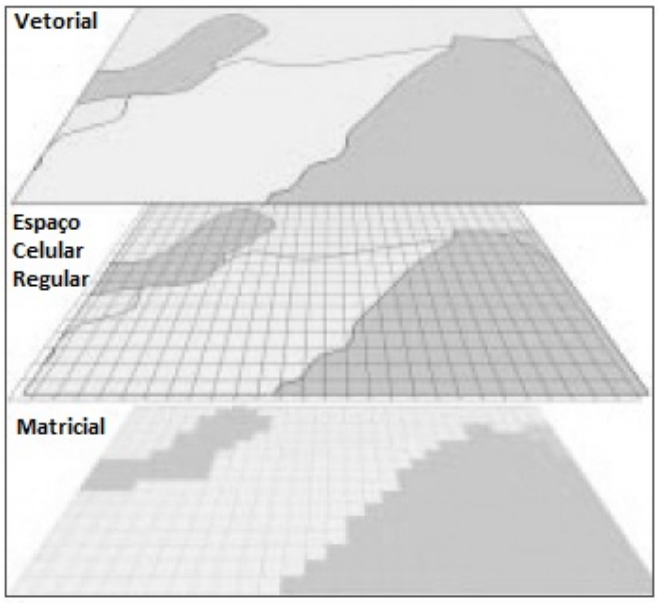


Figura 9: Camadas de um mapa. Fonte: (http://geoportal.icimod.org, 2012).

Na figura 9 está um exemplo dos três tipos de representações de mapas em um sistema de informação geográfica (Vetorial, Matriz e Espaço Celular).

2.5 Medidas e Dimensões Espaciais

O que caracteriza um cubo de dados como espacial é a utilização de geo-objetos, seja como uma medida ou como atributo em uma dimensão. Na Figura 10 temos o exemplo de uma dimensão espacial em um DW que usa o esquema flocos de neve. A localidade é composta por vários atributos alfanuméricos (cep, nome, população, N. de Hospitais) e um atributo geométrico (geometria). A entidade Região serve para definir uma região segundo sua hierarquia (Região $\prec$ Localização).

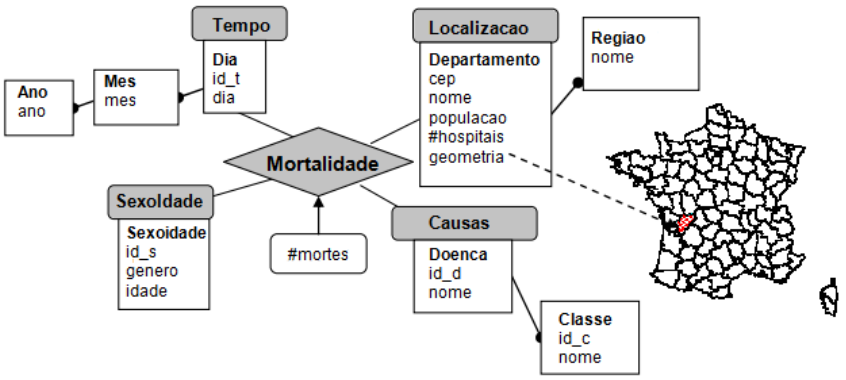


Figura 10: DW mortalidade com a dimensão espacial Localizacao. Fonte: Figura adaptada de (Bimonte S., 2005).

As dimensões espaciais são classificadas em três tipos, segundo (Rivest et al., 2003) (Han et al., 1998a). O primeiro tipo é o não-geométrico. Neste tipo a informação espacial é representada por um dado não-espacial(nome, cep...) em todos os níveis de sua hierarquia. No segundo tipo, todos os níveis de sua hierarquia são

representados por objetos geográficos possibilitando visualização cartográfica das consultas geradas pelo usuário. O terceiro tipo é a mista que é uma combinação dos outros dois tipos. A representação mista varia entre a representação não-espacial em alguns níveis de hierarquia e a representação cartográfica em outros níveis. Os três diferentes tipos de representação podem ser vistos na Figura 11.

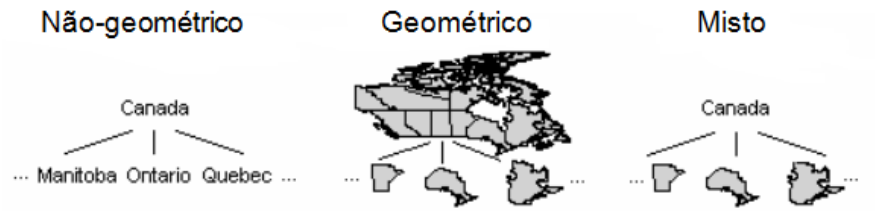


Figura 11: Os tipos de dados suportados em uma dimensão espacial. Fonte: (Rivest et al., 2005).

Uma medida espacial possui um ou mais ponteiros para objetos geográficos (Han et al., 1998b) dependendo do nível de agregação. Na figura 12 está um exemplo da medida espacial 'geometria' que é um ponteiro para um objeto espacial.

Muitos autores consideram uma medida espacial como somente sua parte geográfica, porém outros trabalhos incluem como medida geográfica objetos espaciais complexos. Os objetos espaciais complexos são objetos formados por atributos descritivos e\ou atributos geométricos (nome, população, geometria, etc..) e buscam representar todas as informações necessárias de um objeto real como mostrado na tabela de fatos *departamento* da figura 12 em que a entidade *departamento* é composta por medidas alfanuméricas e uma medida geográfica. Quando é usado objetos espaciais complexos como medida surge um novo problema para a computação das agregações que é chamado dependência

entre atributos, pois no momento da agregação estes atributos podem possuir dependências entre si (Bimonte S., 2005).

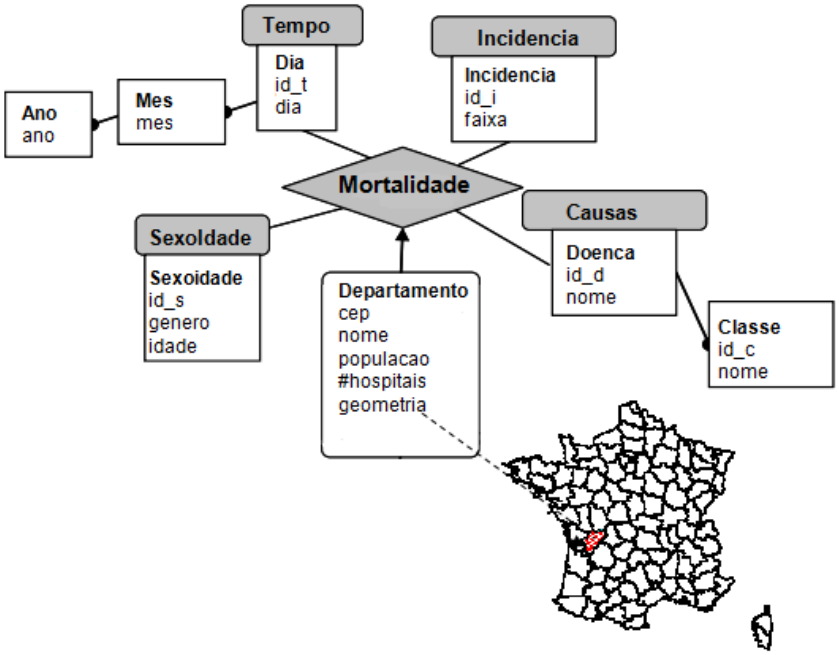


Figura 12: DW mortalidade com a medida espacial departamento. Fone: (Bimonte S., 2005).

Como resultado da agregação de dois objetos espaciais complexos é gerado um novo objeto espacial complexo com os mesmos atributos. Ao realizarmos a agregação de dois atributos espaciais deve-se calcular o valor das agregações de todos os outros atributos. Para isso deve-se aplicar uma função de agregação equivalente a que foi aplicada no atributo espacial. Vamos ao exemplo:

Ao agregar dois objetos espaciais complexos do tipo Departamento e realizar a união de suas geometrias, deve-se somar os valores de população e o de número de hospitais. Se ao invés de unir as geometrias, realizássemos interseção não seria feito a soma do número de hospitais e da população, mas sim a diferença

destes atributos. Definir as dependências entre os atributos para diferentes tipos de agregações espaciais e para diferentes tipos de atributos é um obstáculo para geração de novos objetos espaciais a partir da agregação de objetos hierarquicamente inferiores. Na abordagem GeoCube o usuário é responsável por definir estas dependências.

2.6 Hierarquia Espacial

Assim como no cubo alfa-numérico o cubo espacial também é hierarquizado a fim de possibilitar operações como *roll-up*, *drill-down*, *slice* e *dice*.

A hierarquia de uma dimensão espacial ocorre por inserção de novos atributos que classificam o objeto nos diferentes níveis de hierarquia ou por adição de dimensões. Cada nível de hierarquia possui um identificador para o objeto geográfico correspondente, o que permite diferentes tipos de representações para os diferentes níveis. A medida que mudamos o nível da hierarquia mudamos sua visualização. Este tipo de abordagem facilita a navegação no cubo de dados e também evita que seja feita operações complexas sobre polígonos.

Na Figura 13 ilustra-se uma representação de hierarquia temporal com a seguinte ordem (Ano $\prec$ Semestre $\prec$ Mes $\prec$ Dia) (Ano $\prec$ Semana $\prec$ Dia). Na mesma Figura há uma hierarquia espacial com a seguinte ordem (Pais $\prec$ Estado $\prec$ Cidade).

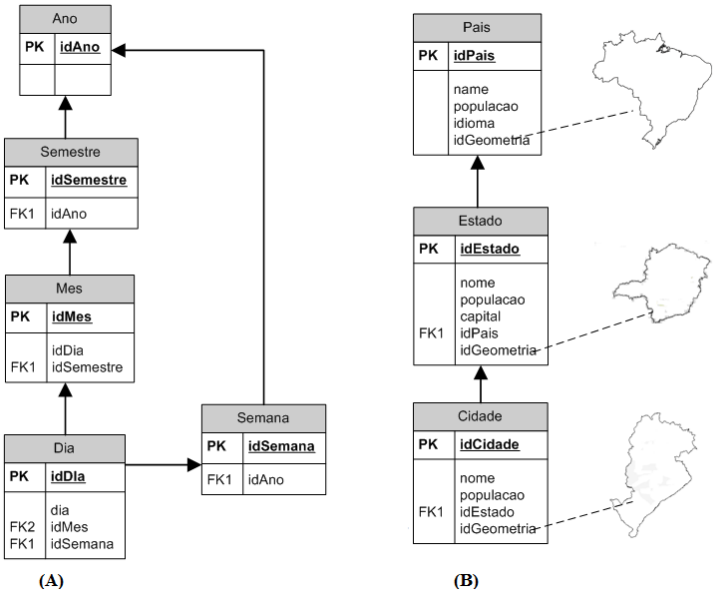


Figura 13: Hierarquia temporal (A) e hierarquia espacial (B).

2.7 Medidas Espaciais

Assim como é classificada as medidas não espaciais também é classificada as medidas espaciais:

- **Distributiva:** Neste grupo estão as funções de agregação mais simples de serem calculadas, como união geométrica, interseção geométrica e *Minimal Orthogonal Bounding Box*. É possível calcular este tipo de função de agregação a partir do nível mais baixo da hierarquia. Note que só é preciso o uso dos níveis imediatamente abaixo na hierarquia e não todos os níveis inferiores da hierarquia. Conforme já explicado, esta propriedade otimiza a computação de cubos;

- **Algébrica:** É possível reduzir qualquer agregação algébrica em agregações distributivas. As funções de agregação espaciais que pertencem a este grupo

são: centroide, centro de massa e centro de gravidade;

- **Holística:** Neste grupo estão as funções mais difíceis de serem computadas, e seus valores não possuem correlação na hierarquia e nem podem ser derivadas em agregações distributivas. Entre as funções holísticas estão: equipartição e vizinho mais próximo.

3 Trabalhos Relacionados

Os trabalhos relacionados foram avaliados segundo as seguintes propriedades:

- **Visualização:** Visualização 3D;
- **Recursos Espaciais:** Medidas e Funções de agregação espaciais diversas, Dimensão espacial, Medidas Hibridas, Hierarquização Automática, múltiplas medidas espaciais, não espacias ou combinações;
- **Desempenho:** Abordagem Paralela, Atualização Incremental.

3.1 PostGis

O sistema PostGis[1] é uma extensão espacial do SGBD PostGreSQL[2] que oferece suporte a objetos geográficos, medidas e funções de agregação espaciais. Normalmente, é utilizado como repositório de dados para sistemas SOLAP. O PostGis usa a biblioteca JTS[3] para realizar a manipulação dos dados espaciais. O PostGis também oferece a indexação dos objetos geográficos através dos índices

[1] http://postgis.refractions.net/
[2] http://www.postgresql.org/
[3] http://tsusiatsoftware.net/jts/main.html

GiST (*Generalized Search Tree*), R-Tree e B-Tree o que torna o processamento de consultas mais eficiente.

Como alternativa para computação de cubos o PostGis oferece a pré-computação de consultas através das vistas materializadas. O propósito das vistas materializadas é diminuir o tempo de resposta de consultas que processam grandes quantidades de dados, agregações e relacionamentos. Para isso, a consulta é pré-computada e seu resultado é persistido no repositório de dados evitando que esta seja re-computada toda vez que for requisitada pelo usuário. Infelizmente o PostGis não oferece suporte nativo a computação de cubos, portanto não possui otimizações ao gerar as agregações. Estratégias de atualização devem ser definidas para se modificar as views materializadas. O PostGis não possui versão desenhada para arquiteturas de computadores com múltiplos núcleos de processamento ou múltiplos computadores, portanto as operações de carga e consulta usufruem somente do paralelismo implícito, o que muitas vezes não consome todo o recurso computacional disponível. PostGis também não oferece a visualização dos mapas. O PostGIS é um SGBD relacional com extensões espaciais, portanto há alguns visualizadores 2D disponíveis para visualizar esquemas espaciais postGIS, dentre eles o mais usado é o PostGIS Viewer [4].

3.2 GeoMondrian

O GeoMondrian[5] é um sistema SOLAP de código aberto escrito em Java que oferece funções como: exploração dos dados de forma interativa, consultas ad-hoc

[4] http://geotux.tuxfamily.org/index.php/en/component/k2/item/293-consola-sql-para-plugin-pgadmin-postgis-viewer

[5] http://www.spatialytics.org/projects/geomondrian/

em mapas espaciais, funções de agregação e medidas espacias e suporte a dados geométricos vetoriais. O GeoMondrian é a versão espacial do servidor OLAP Mondrian e, assim como o PostGis, usa a biblioteca JTS para manipulação dos dados espaciais.

A manipulação do cubo utiliza a linguagem MDX, desenvolvida pela Microsoft. As definições do cubo utilizam esquemas XML. O GeoMondriam carrega os dados persistidos no SGBD PostGis e armazena o cubo resultante em memória primária, portanto cubos computados a partir de bases massivas requerem muita quantidade de memória RAM. O GeoCube também possui tal limitação, sendo necessário versões que fazem uso eficiente de memória externa. Atualizações incrementais dos dados são feitas por meio de notificações, ou seja, o SGBD notifica o GeoMondriam quando algum dado sofre um modificação, o GeoMondrian identifica as células afetadas e refaz as agregações. O GeoCube não implementa o serviço de atualização até o momento.

O GeoMondriam também não suporta computação paralela do cubo de dados. As hierarquias espaciais são obrigatoriamente definidas pelo usuário, não cabendo hierarquias automatizadas a partir de regras de vizinhança entre células de uma grade espacial. O GeoMondriam não oferece a visualização dos mapas gerados, os mapas devem ser visualizados através de outros sistemas. O sistema suporta medidas espaciais e medidas híbridas.

3.3 GeWOlaP

Em (Bimonte et al., 2007) os autores citam as principais características dos sistemas SOLAP e a dificuldade em implementar tais sistemas. Entre os principais desafios estão: a definição de medidas espaciais e suas funções de agregação, dimensões espaciais e suas hierarquias e extensão dos operadores SIG para os sistemas SOLAP.

Os autores também propõem o sistema GewOlap que é um sistema SOLAP Web baseado no modelo multidimensional apresentado anteriormente em (Sandro Bimonte, 2006). O GewOlap é construído em 3 camadas. A primeira camada é um banco de dados ORACLE que suporta operações espaciais, a segunda é o servidor OLAP Mondrian que realiza as agregações e persistência dos dados e a terceira é a camada de visualização representada por um cliente SIG Web composta pelos componentes JPivot, que possibilita a visualização dos dados de forma tabular e do MapXtreme, que fornece a visualização espacial. Os componentes MapXtreme e o SGBD Oracle usado pela solução não possuem código aberto.

O sistema integra todas as tecnologias usadas e customiza o servidor OLAP Mondrian para manipular objetos espaciais. A solução também mescla funcionalidades tradicionais de sistemas OLAP e de sistemas SIG como a exportação de mapas, múltiplas camadas, pivoteamento, drill-downs e roll-up em dados alfanuméricos, entre outros. Este sistema suporta consultas geográficas em diferentes hierarquias, possibilitando as operações de *drill-down* e o *roll-up* espaciais.

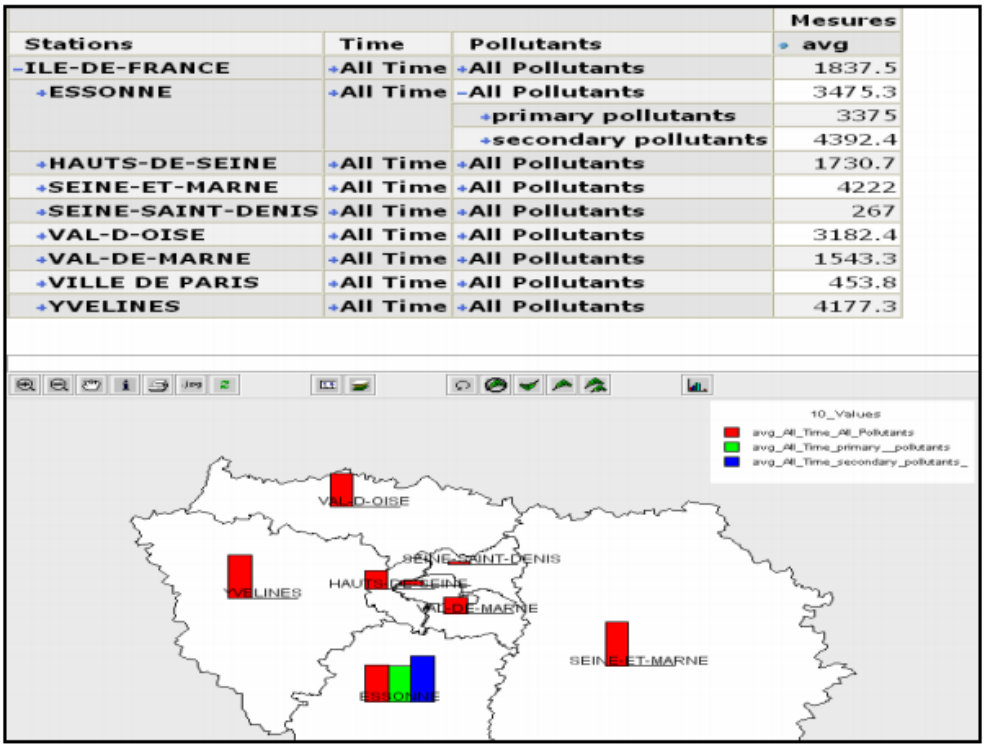


Figura 14: Interface GeWOlap. Fonte: (Bimonte et al., 2007).

O GeWOalp sofre das mesmas limitações do servidor GeoMondrian, pois este é uma extensão do servidor Mondrian. Entre as principais limitações está a falta de uma abordagem paralela para a computação do cubo. A estratégia de atualização incremental é definida pelo Mondrian. Os autores não sugerem nenhuma abordagem para otimização na computação e consulta de cubos espaciais e também não é feita nenhuma análise de desempenho. Para validar o sistema foi realizado algumas consultas e operações sobre um cubo que usa dimensões espaciais e medidas numéricas. Não é mostrado detalhes de sua implementação ou da integração das diversas tecnologias usadas. Na Figura 14 pode ser visto a interface do sistema GeWOlaP.

3.4 SOVAT

Em (Scotch and Parmanto, 2005) é proposta a aplicação SOLAP chamada SOVAT(*Spatial OLAP Visualization and Analysis Tool*) que tem como principal foco servir como ferramenta de análise multidimensional de bases massivas na área de saúde. Entre as principais qualidades do SOVAT estão a sua interface simples, apresentada na Figura 15, que combina a visualização geográfica com visualizações tabulares, permitindo que o usuário associe os dados numéricos ao contexto espacial. O sistema ainda oferece a navegação dos dados utilizando mapas e/ou gráficos.

O estudo de caso realizado para validar o sistema primou pela análise de diferentes fontes de dados. O estudo foi conduzido na região rural da Pennsylvania. Entre as informações selecionadas estão dados populacionais, dados socioeconômicos, incidência de câncer e outros. As dimensões escolhidas para o cubo são: idade, sexo, raça, nível de escolaridade, peso ao nascer, diagnóstico, ano, região e geografia. Cada dimensão possui seus próprios atributos. O cubo geográfico contém apenas uma dimensão espacial chamada geografia. Todas as medidas testadas não são espaciais, portanto não é possível afirmar se tal solução possui suporte e como o implementa. Também não é citada nenhuma estratégia para atualização incremental das agregações.

O trabalho não apresenta detalhes de sua implementação e não propõe técnicas para otimizar as consultas ou a computação de cubos espaciais. O autor apenas cita que o sistema é capaz de manipular grande quantidade de dados sem fazer uma análise quantitativa do mesmo. O SOVAT não oferece a visualização

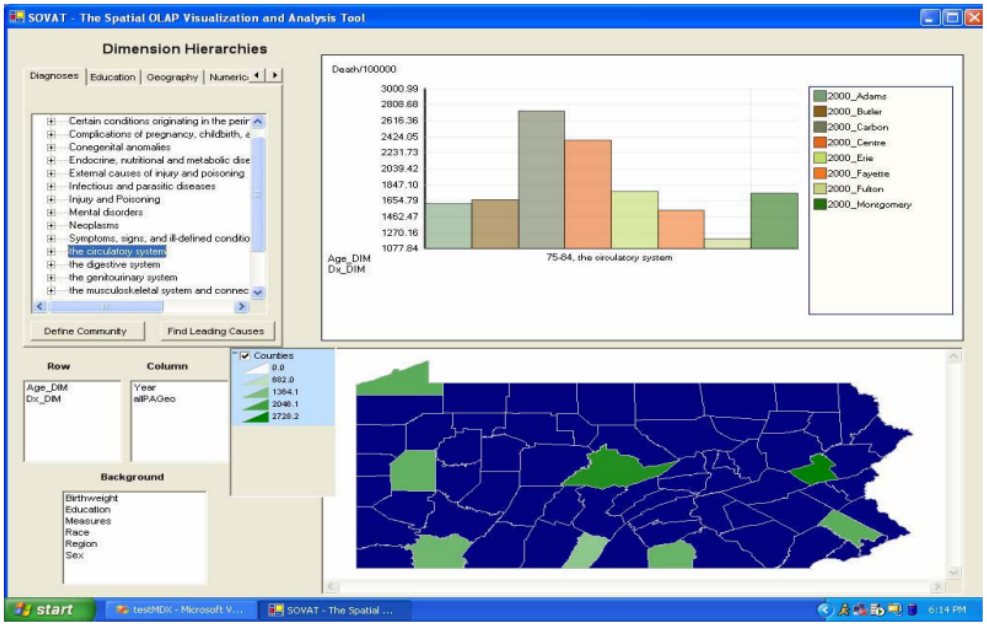


Figura 15: Interface SOVAT. Fonte: (Scotch and Parmanto, 2005).

dos dados geográficos em três dimensões.

3.5 MapCube

Em (Shekhar et al., 2001) é apresentado o operador *MapCube* que é uma extensão do operador tradicional cubo de dados proposto por (Gray et al., 1997) . Este operador é capaz de apresentar os resultados das agregações tanto de forma tabular quanto em forma de mapas. O sistema organiza a coleção de mapas gerados possibilitando operações de *drill-down*, *roll-up*, *slice* e *dice*.

O operador MapCube recebe como parâmetros de entrada o mapa base, tabelas base e preferências cartográficas e gera uma coleção de mapas para comparação e análise. MapCube é um cubo de dados tradicional com visualização cartográfica onde cada dimensão gera um mapa relacionado. O MapCube permite a formação

de hierarquias e a navegação no cubo de dados espacial.

O cubo geográfico é computado da seguinte forma. Primeiro, computa-se a porção alfanumérica dos dados e depois a porção espacial do mesmo. O SGBD processa 2^n *group bys* onde n é o número de atributos alfanuméricos da consulta e a biblioteca geométrica gera os mapas correspondentes à porção alfanumérica. O GeoCube adota estratégia similar ao MapCube, porém a extende com uma versão paralela.

Para validar o sistema foram usados os dados do senso dos EUA como estudo de caso. O cubo gerado possui quatro dimensões e uma medida. As dimensões são: local, faixa etária, raça, faixa de renda e uma dimensão espacial. A medida é a população. Neste trabalho não foi desenvolvido um sistema e sim um operador, portanto não é possível avaliar visualização 3D. Além disto, não é feita nenhuma análise de desempenho apesar de ser proposto um novo operador. O trabalho também propõe uma linguagem de consulta, porém não é descrito como tal linguagem incorpora os operadores espaciais como: interseção, toca, distância, entre outros. O custo de atualização das agregações não é mencionado, portanto não pode-se afirmar que o MapCube possui atualizações incrementais de um cubo espacial.

Em (Moreno et al., 2009) o operador MapCube foi extendido para implementar múltiplas medidas espaciais, porém ainda não especifica como computar medidas espaciais à partir de objetos complexos, ou seja, objetos onde a porção alfanumérica dos dados está associada a porção geométrica. Hierarquização automática a partir de regras de vizinhança entre geo objetos não é possível com o operador MapCube.

3.6 GOLAPA-GWD

Em (do Nascimento Fidalgo et al., 2004) os autores propõem uma arquitetura para integração de um sistema SIG e um sistema OLAP. O sistema atribui identificadores e gera metadados que relacionam os objetos geográficos aos não geográficos. Esse relacionamento não altera as demais estruturas do SIG e dos SGBD's usados. Isto possibilita que o servidor OLAP agregue dados que estão em um SGBD relacional enquanto o SIG realiza as operações espaciais, uma vez que não há perda das referências dos objetos geográficos. A ferramenta GOLAPA funciona como um comunicador dos sistemas SIG e OLAP. Os metadados desempenham um importante papel para manutenção da semântica, consistência entre os dados e integração das funcionalidades.

O sistema GOLAP é capaz de gerar visualizações em Mapas e/ou Tabelas, como é mostrado na Figura 16. Entre as consultas que o sistema pode computar estão as estritamente espaciais(quais lojas ficam mais próximas da matriz), analíticas (qual a quantidade dos produtos vendidos em 2000) ou analítico-espaciais (quais são as categorias de produtos mais vendidas por cada loja do conjunto de lojas próxima da matriz). O foco do trabalho é a arquitetura do sistema e as técnicas usadas para integração das diferentes camadas. O trabalho não faz nenhuma análise de desempenho e não apresenta como as agregações são computadas.

O trabalho não cita nenhuma estratégia para computação paralela do cubo de dados nem para atualização incremental do cubo. Estratégias para formação automática das hierarquias também não são citadas. A visualização dos mapas

é feita em duas dimensões e o trabalho não cita o uso de medidas espaciais.

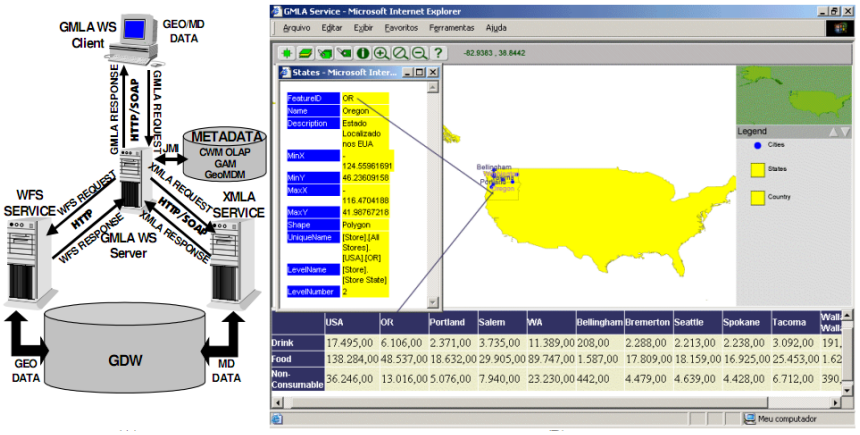


Figura 16: Interface Golapa. Fonte: (do Nascimento Fidalgo et al., 2004).

3.7 JMap

Em (Technologies, 2005) é apresentado o sistema JMap, atualmente disponibilizado como um sistema comercial [6]. Este sistema é feito na linguagem de programação Java e contempla as funções básicas de um sistema SOLAP como *drill-down*, *roll-up* e visualização de dimensões espaciais. O sistema foi construído a fim de ser um sistema Web simples de usar e com uma interface intuitiva, permitindo que usuários sem muito conhecimento técnico possam visualizar e analisar cubos espaciais com poucos cliques.

O sistema JMap não suporta medidas e funções de agregações espaciais. O trabalho também não apresenta nenhuma análise de desempenho. O JMap não suporta a computação paralela do cubo de dados e a atualização incremental das agregações e oferece somente visualização em duas dimensões dos dados geográficos.

[6] http://www.webcitation.org/getfile?fileid=c58181daed52ad95c9081cb273c34927b172bd10

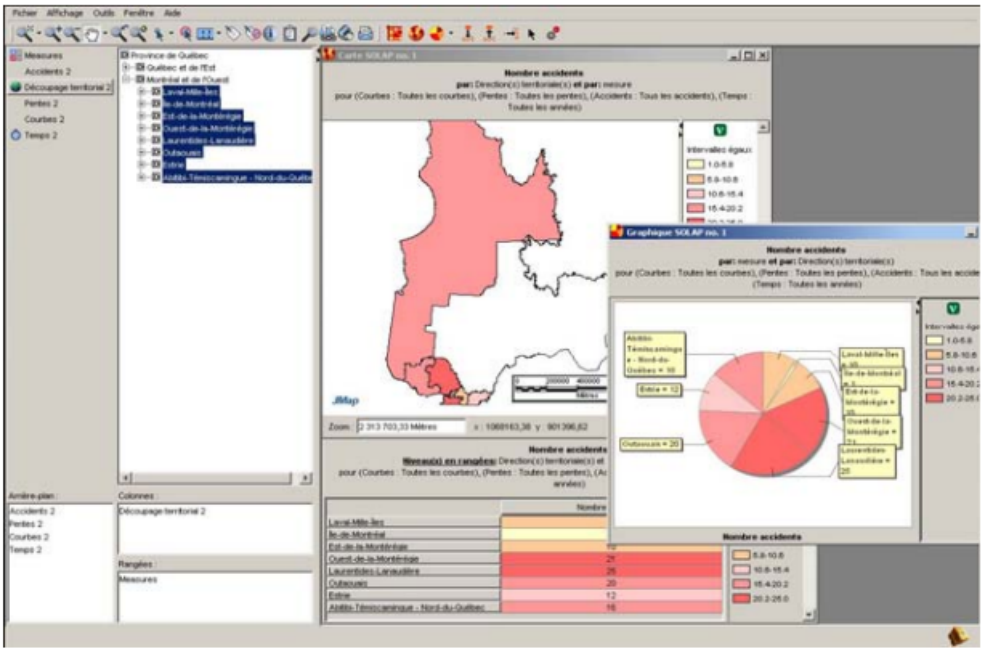


Figura 17: Sistema JMap. Fonte: (Technologies, 2005).

3.8 GlobeOlap

Em (Ferraz and Santos, 2010) é apresentado um sistema SOLAP especializado na visualização de mapas em 3D, como é apresentado na Figura 18. O trabalho não mostra detalhes de implementação e nem detalhes da integração das diversas tecnologias. Como estudo de caso é usado uma base de dados referente aos indicadores do ensino básico no Brasil. Os dados são organizados no esquema estrela. O sistema não é capaz de usar medidas e funções de agregações espaciais e também não apresenta nenhuma otimização para manipulação dos objetos espaciais. Não é feita uma análise de desempenho no sistema GlobeOlap. Não é mencionado versões para máquinas com múltiplos núcleos de processamento.

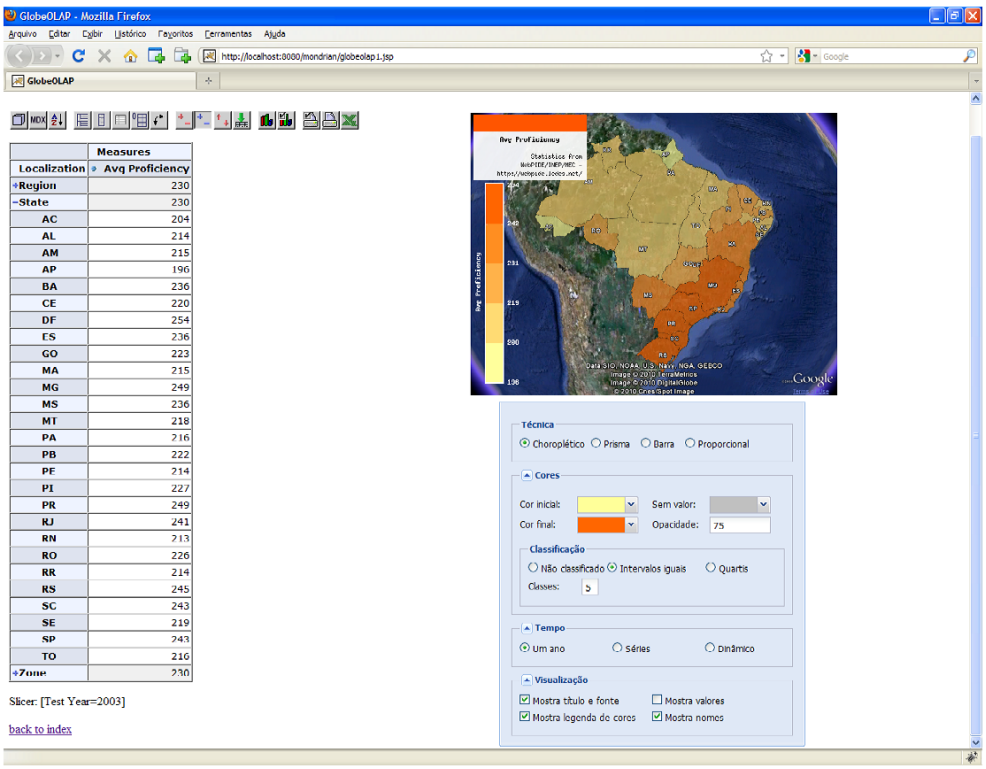


Figura 18: Sistema GlobeOlap. Fonte: (Ferraz and Santos, 2010).

3.9 PostGeoOlap/GeoOlap

Em (Colonese et al., 2008) o autor descreve uma ferramenta livre chamada Post-GeoOlap. Trata-se de um sistema SOLAP de código aberto que tem como finalidade servir como ferramenta de análise para uso em pequenas e médias empresas e unidades públicas de pequeno porte no Brasil. O sistema utiliza o SGBD PostGis para processar consultas geográficas e analíticas e também para persistir o conjunto de dados agregados. O sistema computa partes do cubo quando o desempenho de uma certa consulta está abaixo de um limiar. Quando este cenário ocorre, é criada uma nova tabela para armazenar o resultado das agregações. A nova tabela utiliza duas estruturas de indexação oferecidas pelo PostGreSQL: Os

atributos não-geométricos são indexados por uma árvore-B e os geométricos são indexados por uma GIST. Para visualização dos dados geográficos foi utilizado o *framework* OpenJUMP[7] que é uma solução livre e extensível para visualização de dados espaciais. O sistema integra várias soluções livres a fim de disponibilizar um sistema SOLAP com o maior número de funcionalidades possível. Não é apresentado nenhuma interface para criação de consultas. O trabalho foca na explicação da arquitetura e na modelagem do banco de dados (definição de dimensões, medidas e relação entres estes dados) e não é feita nenhuma análise de desempenho. Não é feita menção a utilização de arquiteturas de computadores de alto desempenho e nem testes com bases massivas. O sistema não suporta a atualização incremental do cubo de dados. Sua interface pode ser vista na Figura 19.

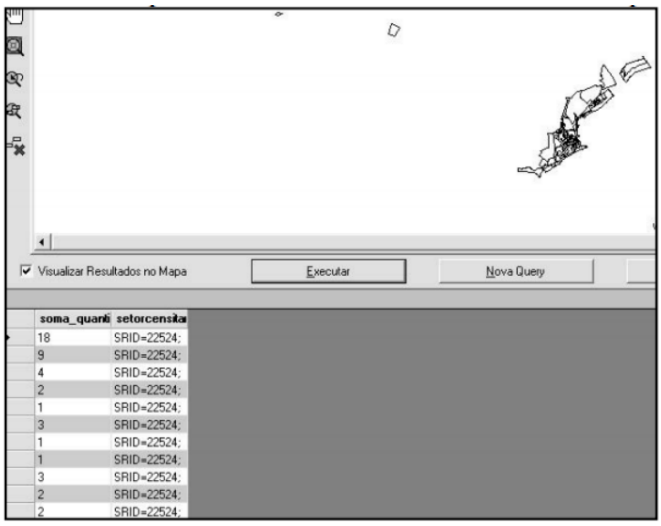


Figura 19: Interface PostGeoOlap. Fonte: (Colonese et al., 2008).

[7]http://www.openjump.org

3.10 MapWharehouse

Em (Sousa, 2007) é apresentado um sistema chamado MapWharehouse que é uma ferramenta Web que oferece as principais funções de um sistema SOLAP (dimensões, medidas e funções de agregação espaciais e também operações de *Drill-Down, Roll-Up, Slice* e *Dice* espaciais).

O sistema é dividido em três camadas e utiliza um SGBD objeto relacional.

- **Camada de visualização:** é composta por um *framework* chamado IGIS que além de propiciar a visualização dos resultados também é responsável pela interface de criação de consultas;
- **Camada de aplicação:** é responsável pela computação do cubo e pelo processamento das consultas SOLAP e foi implementada em Java;
- **Camada operacional:** é composta pelo DW espacial.

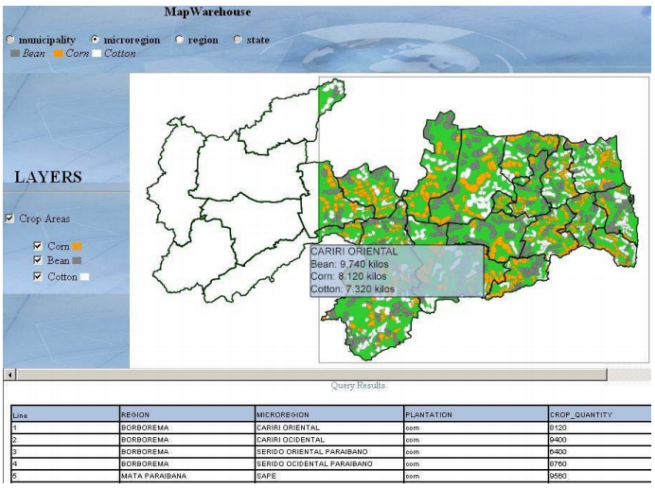


Figura 20: Interface do MapWharehouse. Fonte: (Sousa, 2007).

No trabalho também são propostas algumas técnicas de otimização para consultas espaciais. As técnicas se baseiam na utilização de pre-agregação em diferentes níveis de hierarquia. O desafio de tal solução é determinar a quantidade ideal de dados e hierarquias a serem pré-computadas a fim de responder a consulta do usuário em um tempo aceitável. Assim como em outros trabalhos, a agregação pode ser computada somente quanto consultada, gerando enorme perda de desempenho. As consultas são automaticamente reescritas para acessar os dados agregados espaciais pré-armazenados de forma apropriada. Uma vez reescritas, as consultas são otimizadas pelo SGBD Oracle que explora índices espaciais como R-trees , entre outras técnicas de otimização física. Os mapas são apresentados em 2D e não possui uma abordagem paralela para computação dos cubos.

Para validar o sistema é usado o estudo de caso sobre plantações agrícolas no Brasil, que integra informações analíticas e espaciais. Não é feito uma análise de desempenho comparativa com outros sistemas. A análise de desempenho é feita apenas comparando consultas que usam ou não as técnicas de otimização propostas. O autor cita que técnicas para implementação de materialização seletiva e atualização incremental serão implementadas em trabalhos futuros. Sua interface pode ser vista na Figura 20.

3.11 Resumo

O desenvolvimento de um sistema SOLAP envolve diversos desafios no que diz respeito a seu desempenho, confiabilidade dos dados, visualização, navegação, funcionalidades SIG e outros. Assim, os trabalhos que abrangem todos os

aspectos de um sistema GIS não são comuns. O trabalho (Sousa, 2007) é o mais abrangente os outros trabalhos se destacam em áreas específicas. O GeoMondrian e (Shekhar et al., 2001) dão ênfase a computação do cubo de dados, outros trabalhos como (Ferraz and Santos, 2010) (Scotch and Parmanto, 2005) dão ênfase na visualização e navegação do cubo. Os trabalhos (do Nascimento Fidalgo et al., 2004) e (Colonese et al., 2008) procuram integrar tecnologias nas diferentes camadas com algoritmos e metadados. Na tabela 1 é apresentado um resumo das principais características dos trabalhos relacionados.

Sistema/Abordagem/Operador	Medidas Espaciais	Múltiplas Medidas Espaciais	Dimensão Espacial	Visual. 3D	Paralelismo	Medidas Híbridas	Hierarquia Automática	Atualização Incremental
GewOlap (2007)	X		X			X		X
SOVAT (2005)			X					
MapCube (2001)	X	X	X			X		
Jmap (2005)			X					
GlobeOLAP (2010)			X	X				
MapWharehouse (2007)	X		X					
GOLAPA/GDW (2004)			X					
PostGeoOlap/GeoOlap (2008)			X					
PostGis (2012)	X	X	X			X		
GeoMondrian (2012)	X	X	X			X		X

Tabela 1: Tabela comparativa dos sistemas existentes

4 GeoCube

Neste capítulo é apresentado o sistema SOLAP GeoCube. O sistema GeoCube é um sistema SOLAP capaz de computar cubos espaciais completos ou parciais. Cubos GeoCube podem possuir múltiplas medidas, sejam estas numéricas ou espaciais. Dimensões alfanuméricas, espaciais ou híbridas são implementadas. Hierarquias espaciais, temporais ou a partir de dimensões específicas são implementadas. A geração automática de hierarquias espaciais a partir de qualquer regra de vizinhança espacial é outra inovação do GeoCube. Uma vez que o GeoCube não é baseado em vista seletiva e sim na computação de todas as agregações, o tempo de consulta se torna a principal vantagem na visualização do GeoCube. É utilizado a abordagem *Star-Cubing* para garantir a computação eficiente de cubos completos ou iceberg. O algoritmo *Star-Cubing* foi escolhido por ser um algoritmo eficiente e de fácil implementação. É integrado a abordagem *Star* à biblioteca SIG Geotools, garantindo que operações espaciais possam ser realizadas eficientemente. Por fim, é implementado uma versão paralela que intercala múltiplas gerações de agregações alfanuméricas e operações espaciais, garantindo resultados promissores, mesmo na computação de cubos massivos.

A Figura 21 ilustra o sistema GeoCube. O sistema oferece a visualização

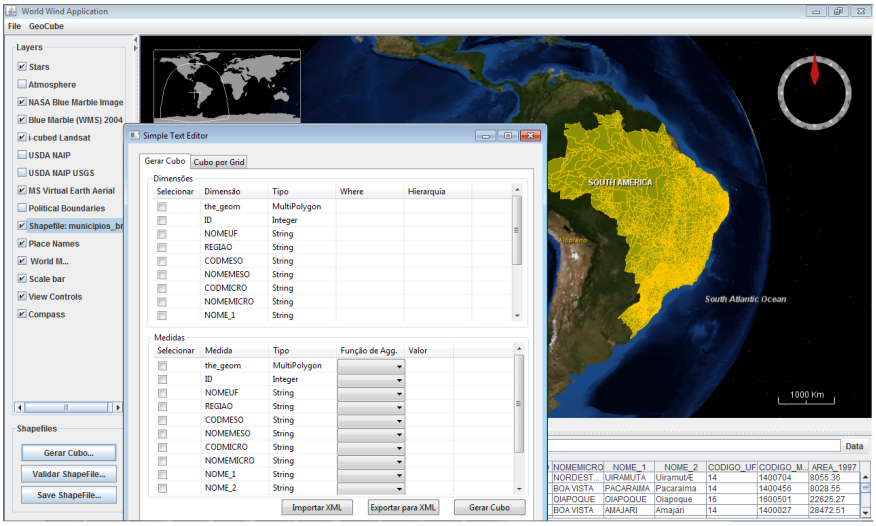


Figura 21: Sistema GeoCube.

em três dimensões (3D) dos dados espaciais e a visualização tabular dos dados alfanuméricos. Além de associar as informações alfanuméricas e espaciais no mapa, o GeoCube permite consultas SQL e também possui uma interface intuitiva para formação do cubo.

Os dados espaciais são renderizados em forma de geometrias 3D independentes, portanto houve problemas gráficos ao tentar renderizar mais do que cinco mil geometrias em um computador com Sistema Operacional windows 7 64 bits. JVM(*Java Virtual Machine*) 6, 1 HDs de 320GB 7200rpm, 4GB RAM DDR2 667, 2 núcleos de processamento com 2.2GHz e placa gráfica AMD Radeon HD 6470M. Diante de tal obstáculo, o GeoCube permite renderização em duas dimensões (2D), renderização raster ao invés dos custosos polígonos e é possível limitar o número de geometrias a serem renderizadas por nível de zoom. O cubo resultante pode ser armazenado em memória primária ou secundária, possibilitando futuras consultas sem que seja necessária a re-computação do cubo. O

GeoCube não suporta a atualização incremental dos dados.

4.1 Arquitetura do Sistema

A arquitetura do sistema desenvolvido é composta por três camadas (Figura 22) e utiliza a arquitetura integrada onde os dados alfanuméricos estão dispostos na mesma base de dados que os dados espaciais. Este tipo de arquitetura suporta funções de sistemas SIG e de sistemas OLAP. Além de unir as funcionalidades SIG e OLAP, esta abordagem tem outra vantagem que é a implementação sem o uso excessivo de metadados. Para isso é feito o uso de tecnologias abertas que têm como finalidade a manipulação e visualização de informações geográficas. Estas tecnologias são integradas ao algoritmo de computação de cubos, formando assim um sistema SOLAP. Múltiplas medidas espaciais, objetos espaciais complexos como medidas são implementados à partir de extensões na abordagem *Star*, adicionando etapa de construção de índices espaciais, e sua estrutura *Star-Node*, adicionando ponteiros para geo objetos.

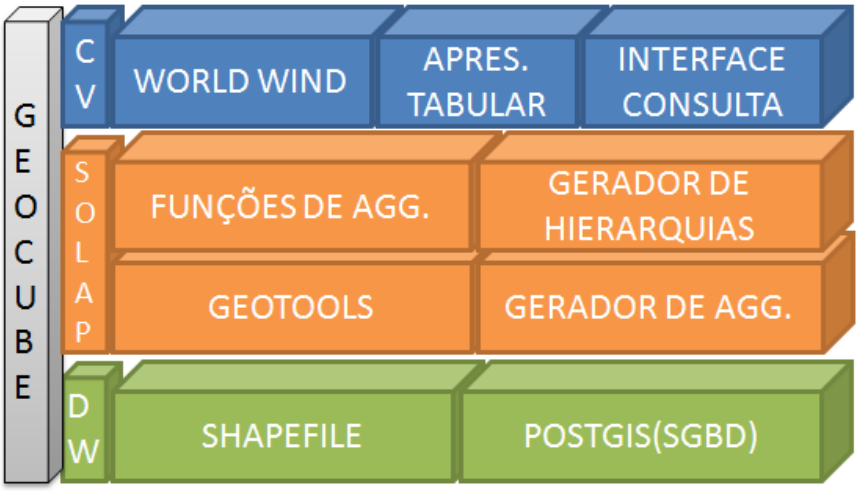


Figura 22: Arquitetura do Sistema GeoCube em detalhes.

A terceira camada é a responsável pela visualização. A segunda camada possui o algoritmo de computação do cubo geográfico, chamado também de GeoCube. A primeira camada é composta pelo repositório de dados. Todos os dados gerados pelo algoritmo GeoCube são persistidos pela primeira camada, seja um SGBD, um shapefile, um arquivo texto ou a combinação dos mesmos. Os mapas gerados são persistidos em memória secundária para futuras consultas.

Os componentes de cada camada são detalhados a seguir:

4.1.1 DW

Está camada é composta por um repositório de dados que deve ter suporte para armazenar objetos geográficos. O GeoCube pode usar como repositório de dados arquivos do tipo shapefile ou o sistema gerenciador de banco de dados PostGis. Estas duas tecnologias foram escolhidas devido a sua ampla utilização

[0]http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf
[0]http://postgis.refractions.net/

para persistir dados geográficos e alfanuméricos.

4.1.2 SOLAP

A segunda camada possui o algoritmo de computação do cubo geográfico. O algoritmo responsável por gerar as agregações, assim como os algoritmos que paralelizam a solução são implementados na linguagem Java e utilizam a biblioteca GeoTools para manipulação de objetos geográficos. Todos os dados gerados pelo algoritmo de computação de cubos espaciais GeoCube podem ser persistidos no repositório de dados.

Em resumo esta camada realiza os seguintes passos:

1. Na primeira etapa as tuplas são separadas por hierarquia, o que possibilita as operações de *drill-down* e *roll-up*;

2. Na próxima etapa é feita a agregação das medidas sendo elas alfanuméricas ou espaciais através de uma adaptação do algoritmo *Star-Cubing* que gera as agregações de um cubo completo ou parcial;

3. Nesta etapa são aplicadas as funções de agregação sobre as medidas do passo anterior, sendo que as funções de agregação espaciais são aplicadas usando a biblioteca GeoTools;

4. Por último é criado os *layers* das diferentes hierarquias. Sendo que para cada *layer* existe um mapa relacionado. O algoritmo também relaciona dimensões e medidas alfanuméricas para cada região vetorial do mapa.

[0]http://www.geotools.org/

Os detalhes desta camada assim como o algoritmo para computação do cubo de dados espacial é descrito na seção 4.2.

4.1.3 Camada de Visualização

O cubo resultante com informações geográficas e alfanuméricas é exibido para o usuário na camada de visualização. Os atributos geográficos são mostrados em um globo terrestre 3D que permite a fácil navegação dos mapas, como mostrado na Figura 23 em que cada ponto representa uma unidade habitacional da cidade de Ouro Preto, além de outros recursos que torna a exploração e análise dos dados geográficos mais intuitiva. Foi implementada uma biblioteca para integrar os dados resultante da computação do cubo e o componente WorldWind. O WorldWind permite a visualização 3D do globo terrestre. Além disto, existe um componente tabular para visualização dos dados alfanuméricos e uma interface de consulta que permite o usuário gerar cubos de maneira mais intuitiva. O sistema também oferece a opção de exportar o cubo resultante para ser visualizado através do Google Maps [1] ou Google Earth [2]. O GeoCube também permite a consulta dos atributos alfanuméricos de forma tabular, porém ainda não há gráficos estatísticos como barra, pizza, linha, entre outros.

O cubo pode ser montado pelo usuário através da interface gráfica mostrada na Figura 24. Através desta interface o usuário pode selecionar quais atributos farão parte das dimensões, das medidas e quais funções de agregação serão usadas, além de definir as hierarquias que serão geradas.

[0] http://worldwind.arc.nasa.gov/java/
[1] https://maps.google.com.br/
[2] http://www.google.com/earth/index.html

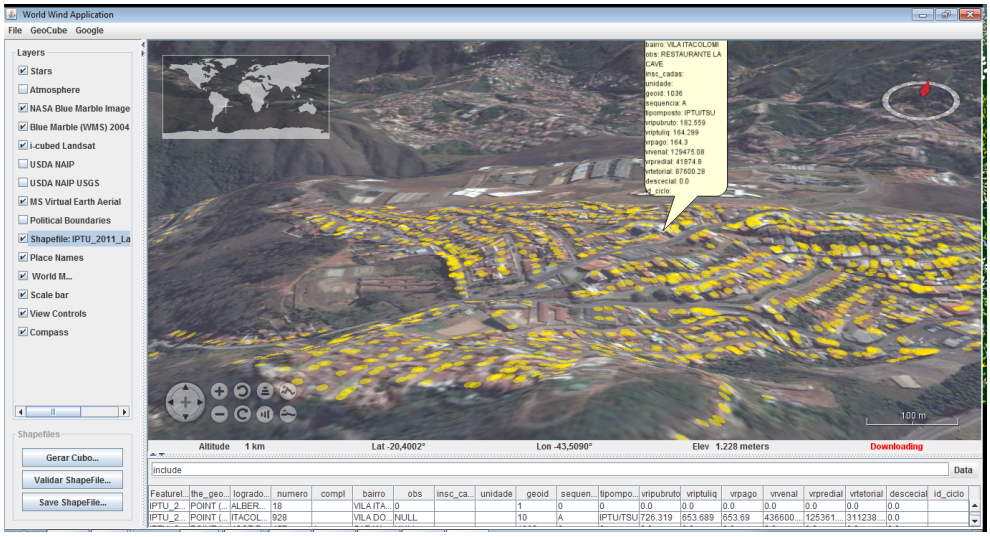


Figura 23: Visualização das unidades habitacionais de Ouro Preto/MG.

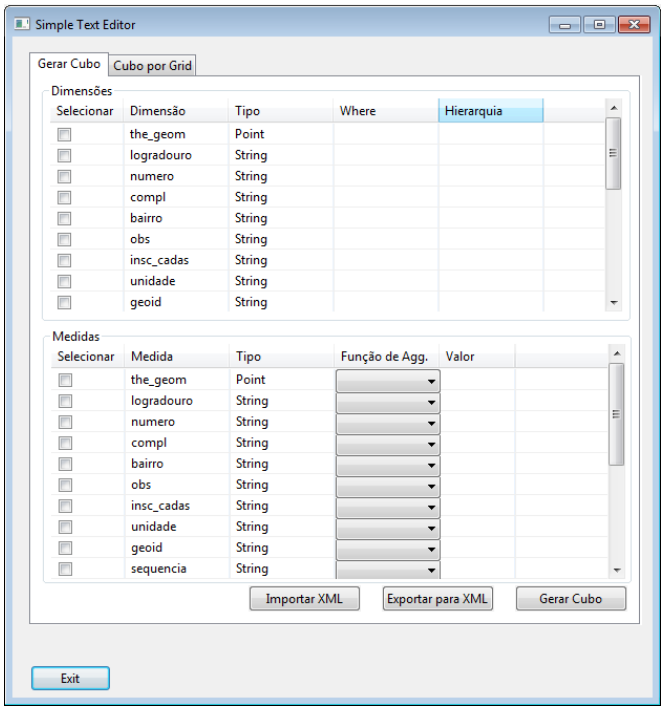


Figura 24: Interface para criação do cubo de dados.

4.2 Detalhamento da camada SOLAP

Nesta seção são descritos os algoritmos usados pelo GeoCube. A seção é dividida em subseções que são as principais etapas da camada SOLAP para computação do cubo completo espacial.

4.2.1 Formação das Hierarquias e do Cubo Base

Nesta seção é apresentada a abordagem usada para gerar as hierarquias e o cubo de dados base a partir de um mapa. A solução proposta permite que o usuário possa definir a hierarquia manualmente ou de forma automática. A hierarquização automática se baseia na ideia de vizinhança espacial e desta forma permite que qualquer regra de vizinhança possa ser aplicada a um conjunto de polígonos para geração de outros polígonos em níveis hierárquicos distintos. A Figura 25 ilustra a geração de novas agregações em níveis hierárquicos distintos e criados de forma automática pelo GeoCube. No exemplo da Figura 25 a regra de vizinhança varia de 3x3 no primeiro nível de agregação para 2x2 no segundo nível de agregação.

A hierarquização automática é representada através de uma nova dimensão que indica a qual hierarquia a região pertence. De acordo com a regra de vizinhança cada região é classificada e sua respectiva hierarquia é atribuída. No caso da hierarquia não ter sido definida através de regras de vizinhança, ela é indicada pelo usuário ao definir as dimensões.

Ao definir uma hierarquia espacial é criada a possibilidade de calcular o valor das funções de agregação alfanuméricas fazendo apenas a leitura da hierarquia

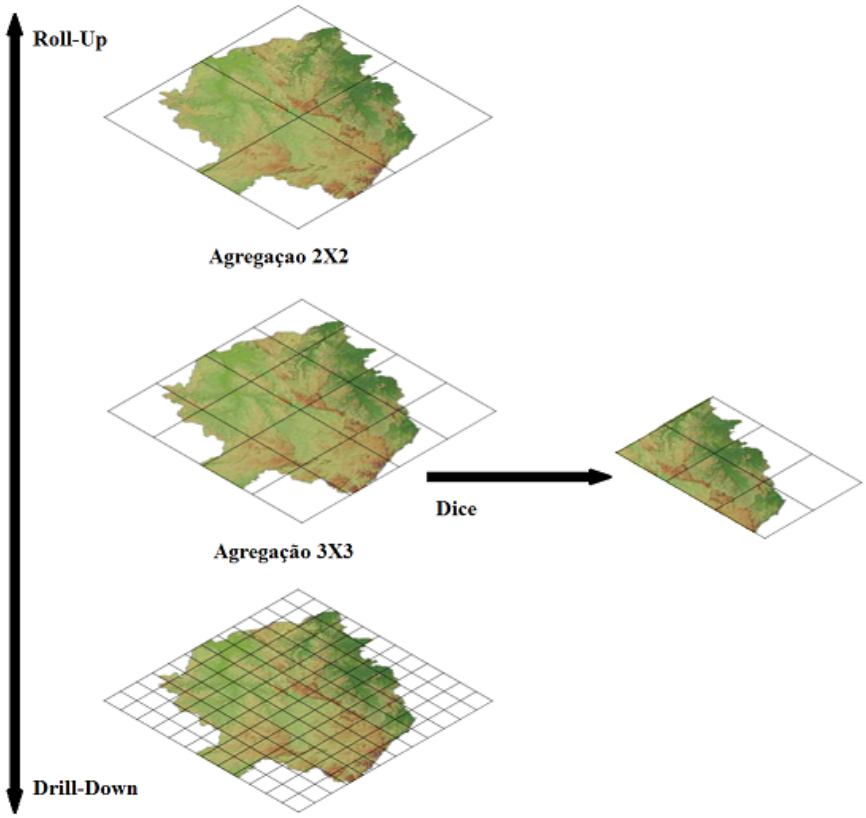


Figura 25: Operações de *drill-down*, *roll-up* e *dice* em espaços celulares regulares.

inferior. Para medidas algébricas e distributivas, sejam espaciais ou numéricas, é usada a estratégia *top-down* conforme descrita na seção 2.3.

O uso de diferentes funções estatísticas ou espaciais para agregar os valores entre diferentes hierarquias possibilita de maneira mais precisa a modelagem de fenômenos não-lineares.

A seguir está um exemplo de como é criado o cubo base, usando a relação de tuplas da tabela 2.

Primeiro é formado o cubo base inserindo uma tupla por vez a partir do nodo raiz reutilizando os prefixos e atualizando as medidas, sejam elas espaciais ou não.

Dimensões			Medida
Doença	Sexo	Mês	Região
Hepatite	M	Dez	1
Chagas	F	Mar	5
Chagas	M	Dez	3
Chagas	F	Set	4

Tabela 2: Relação de tuplas não agregadas com uma medida

O processo é mostrado no algoritmo 4.1 e seus passos são representados na figura 26. Na etapa (a) é inserido a primeira tupla. Na etapa (b) é inserido a segunda tupla, porém como os prefixos são distintos é criado um novo ramo na árvore a partir da raiz. Na etapa (c) é inserido a terceira tupla e como o prefixo 'Chagas' já existe na árvore é criado um novo ramo a partir do nível um da árvore. Ao inserir a última tupla os prefixos 'Chagas' e 'F' já existem na árvore, portanto o novo ramo será criado no nível dois no nodo identificado pelo valor 'F'. Ao final deste processo vamos ter uma estrutura chamada *base cuboid tree* (composto por tuplas sem a formação de todos os subconjuntos das colunas agregadas) com as hierarquias definidas.

O próximo passo é a computação de todas as agregações que será descrita na próxima seção.

```
1 Algorithm 1 SST Base Cuboid
2 Input: A base relation R
3 Output: A base cuboid
4 for each tuple[] in R do
5   call SST_Base_Cuboid(tuple[]);
6
7 procedure SST_Base_Cuboid(tuple[])
8 {
```

```
9    var: SSTtuple[];
10   traverse SST cube from root to leaf, creating new nodes and
         updating the last traversed node measure value if necessary;
11 }
```

Programa 4.1: Algoritmo para computação do cubo base

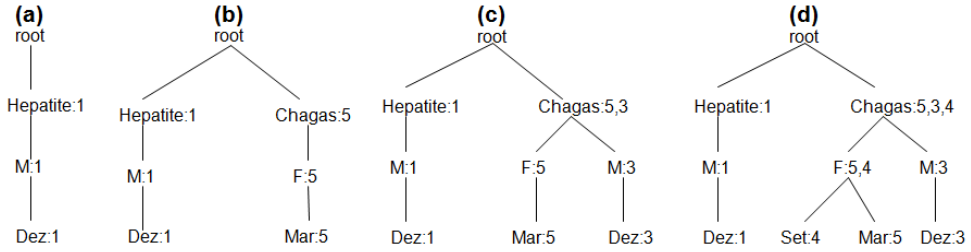


Figura 26: Formação do cubo base.

4.2.2 Computação das Agregações

Nesta etapa são geradas todas as agregações possíveis percorrendo a *base cuboid tree* de maneira *top-down*. O *Star-cubing* manipula uma estrutura de dados chamada *Star-Tree* que realiza a compactação do cubo de dados através da remoção das redundâncias dos prefixos das tuplas. Cada nível da *Star-Tree* representa uma dimensão e cada nodo representa um atributo. A *Star-Tree* possui altura máxima igual ao número de dimensões do cubo. Os nós da *Star-tree* implementada pelo GeoCube possuem as seguintes informações: (i)um conjunto de medidas espaciais; (ii)um conjunto de medidas não-espaciais; (iii)ponteiros para nós descendentes ou para nós irmãos; (iv)conjunto de funções de agregação. O uso desta estrutura possibilita a redução do tamanho do cubo e também possibilita a geração das agregações mais rapidamente. Como o algoritmo permite

a atualização de nodos internos é possível realizar a poda de células que estão abaixo de uma limiar mínimo, se o usuário quiser computar um cubo somente com as doenças que ocorrem em somente uma região, as ramificações da *Star-Tree*, estrutura usada pelo algoritmo *Star-Cubing* para gerar as agregações, que possuem doença de chagas como atributo seriam podadas já na leitura da terceira tupla, diminuindo o tamanho da *Star-Tree* gerada e o número de agregações feitas pelo algoritmo.

As agregações são formadas percorrendo os nós da árvore e inserindo seus descendentes na raiz. Ao inserir os descentes é feita a agregação das medidas. Assim, as medidas de um nodo pai serão compostas pelas medidas de todas as folhas descendentes. No algoritmo 4.2 é apresentado os passos para formação das agregações. Na figura 27 é mostrada a árvore resultante das agregações.

```
1  Algorithm 2 SST Aggregation
2  Input: A base cuboid
3  Output: A complete full cube
4  call SST_Aggr(null, root);
5
6  procedure SST_Aggr(currentAncestral, currentNode)
7  {
8    if ((currentNode has not ben visited) and (currentNode has
         descendants) and (currentAncestral != null)
9      copy the currentNode descendants to currentAncestral;
10     compute currentNode measure value using its non aggregated
           descendants;
11
12   if (currentNode has descendants)
13   {
14     for each currentNode descendant do call SST_Aggr(currentNode,
           currentDescendant);
15     if (currentNode continues having aggregated descendants that
           have not been visited)
16       call SST_Aggr(currentAncestral, currentNode);
17   }
18 }
```

Programa 4.2: Algoritmo para computação das agregações

Ao final da criação da estrutura, a árvore é percorrida e uma chamada a uma biblioteca espacial é feita para cada nodo a fim de executar a função de agregação nas medidas espaciais. Esse processo é feito de forma paralela e explicado com mais detalhes na seção 4.2.4.

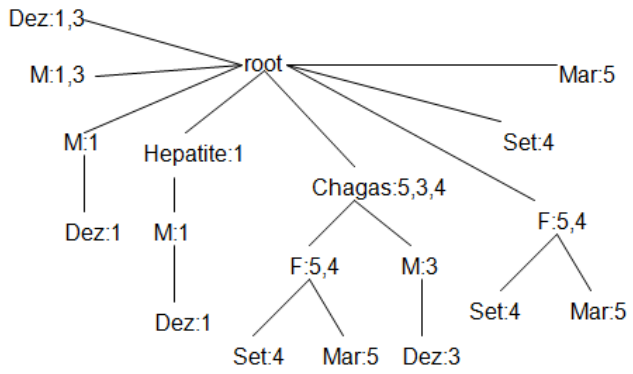


Figura 27: Formação das agregações.

Ao selecionar um resultado agregado o sistema busca a tupla correspondente através de um caminho único formado da raiz até qualquer nó da Star-tree e não somente os nós folha.

Outra vantagem desta abordagem é a possibilidade de armazenarmos os valores de quantas medidas forem necessárias para futuras funções de agregação em uma única varredura. Assim, é possível usar diversas medidas e funções de agregação combinadas sejam elas numéricas ou não.

Ao aplicar o algoritmo sobre a base de dados da Tabela 3 pode-se aplicar funções de agregação sobre objetos geográficos(Região) e medidas numéricas(Idade). No exemplo abaixo é realizada a união das regiões e também é obtida a média de idade das pessoas nas regiões agrupadas por tipo de doença.

O algoritmo constrói a árvore seguindo os mesmos passos para gerar as agregações de uma só medida, porém ao invés de concatenar somente uma medida será concatenado todas as medidas da consulta, formando duas listas distintas como visto na figura 28.

O cálculo simultâneo de medidas também possibilita a adição de novas me-

Dimensões			Medidas	
Doença	Sexo	Mês	Região	Idade
Hepatite	M	Dez	1	23
Chagas	F	Mar	5	35
Chagas	M	Dez	3	26
Chagas	F	Set	4	40

Tabela 3: Relação de tuplas não agregadas com uma medida espacial e uma numérica

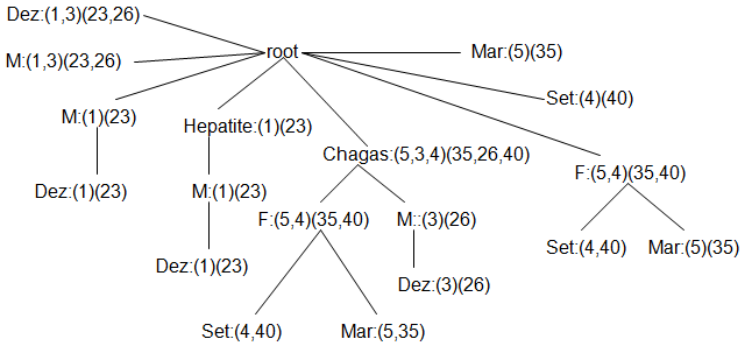


Figura 28: Formação das agregações para uma medida espacial e uma medida numérica.

didas calculadas a partir de medidas existentes na estrutura.

4.2.3 Estratégias de Otimização

Assim como sugerido no trabalho (Han et al., 1998a) ao computar as agregações do cubo é armazenado somente os ponteiros para os objetos geográficos, ou seja, regiões agregadas são conjuntos de ponteiros como é mostrado na figura 29. Uma nova região só é criada ao aplicar as funções de agregação. Esta técnica é usada pelo GeoCube, pois traz grandes benefícios na computação de um cubo espacial completo uma vez que evita que o cubo seja recomputado para cada função de agregação espacial. As funções são aplicadas sob-demanda sobre

o conjunto de ponteiros agregados da tupla, assim pode-se dizer que primeiro é feita as agregações dos ponteiros e depois é aplicada as funções de agregação espaciais sobre estes ponteiros. Outra técnica utilizada neste trabalho é a paralelização da computação do cubo, que será explicada com mais detalhes na próxima seção.

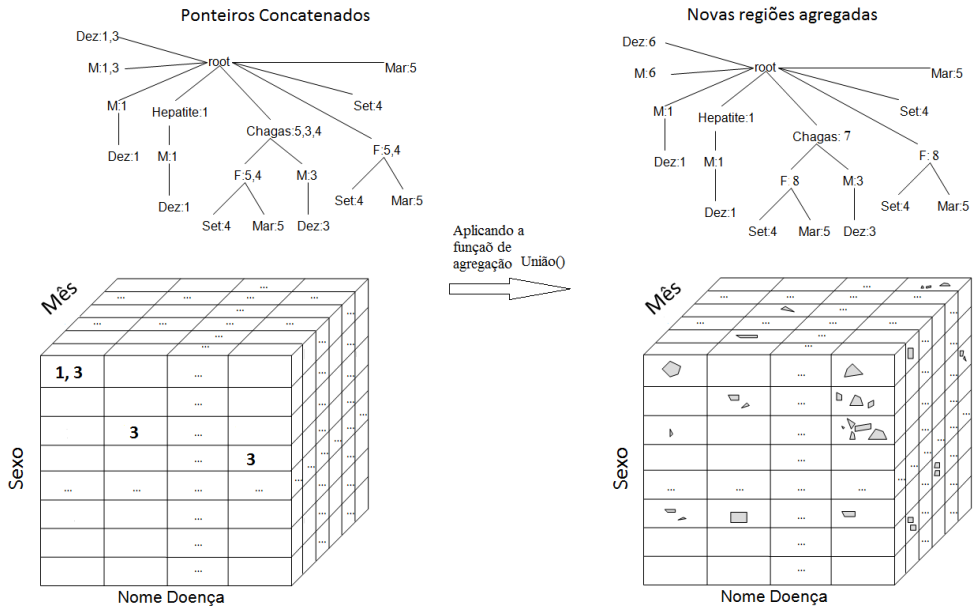


Figura 29: Novas regiões em um cubo com medidas espaciais.

4.2.4 Paralelização

Através da divisão de tarefas para diferentes *threads* pode-se tirar melhor proveito da arquitetura de computadores com múltiplos núcleos. Algumas tarefas na computação de cubos SOLAP possuem fraco acoplamento ou dependabilidade, o que torna simples a prototipação de soluções paralelas escaláveis. A computação do cubo espacial é basicamente divida em três principais etapas:

- **Etapa 1:** Geração das hierarquias ou especificação das regras de vizinhança

para geração automática das hierarquias(Etapa paralelizada);

- **Etapa 2:** Geração das agregações persistindo as medidas através do *Star-Cubing*(Etapa sequencial);
- **Etapa 3:** Aplicação das funções de agregações alfanuméricas e espaciais sobre as medidas persistidas no passo anterior e geração da visualização(Etapa paralelizada);

Dentre as etapas do GeoCube a etapa que consome mais tempo de processamento e memória é a etapa 3. Este passo possui paralelização trivial, portanto a implementamos. Cada *Thread* cuida de um conjunto de nós *Star-Tree* de tamanho similar. Caso não seja similar em tamanho perde-se o balanceamento de carga. De uma forma geral, é simples garantir que cada *thread* receba o mesmo número de nós *Star-Tree* uma vez que é conhecido o número total de nós gerados. As funções de agregação são aplicadas usando a biblioteca GeoTools simultaneamente em máquinas multicore.

Para mapas representados por espaços celulares regulares também pode-se paralelizar de forma eficiente a etapa 1. De uma forma geral, pode-se fazer uma divisão prévia das tuplas sem que ocorra problemas de balanceamento. Outro fator que beneficia a paralelização da computação das hierarquias é o fato de que as regiões agregadas são disjuntas o que evita pontos de sincronização.

4.2.4.1 Paralelização da etapa 1

A estratégia de paralelização para formação das hierarquias é mostrada na figura 30. Tem-se como entrada o mapa base. Este mapa é dividido em regiões

disjuntas, com o mesmo número de células, entre as *threads*. Cada *thread* gera as hierarquias recursivamente unindo regiões da hierarquia inferior de acordo com as regras de vizinhança determinadas pelo usuário. Ao final do processo o resultado é todas as hierarquias definidas pelo usuário já computadas. A computação de todas hierarquias de uma certa porção de células é feita por uma única *thread* usando a hierarquia inferior para o cálculo da hierarquia imediatamente superior. Assim, não é necessário redistribuir as tuplas na computação de cada hierarquia fazendo o melhor uso do princípio da localização. O próximo passo é a geração das agregações, feita de forma sequencial pela etapa 2 através do algoritmo star-cubing.

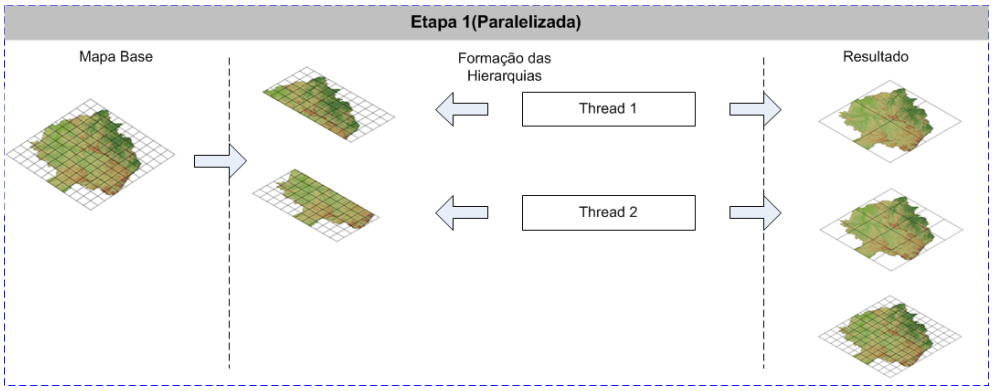


Figura 30: Formando hierarquias paralelamente.

4.2.4.2 Paralelização da etapa 3

A terceira etapa é implementada da mesma maneira para os dois tipos de mapas (vetoriais e espaços regulares). A estratégia de escalonamento escolhida é a produtor-consumidor, que permite que as *threads* sejam reutilizadas, evitando o custo de criação de *threads*, assim como melhorando o balanceamento de carga entre as *threads*.

O algoritmo de paralelização usa como recurso as tuplas obtidas através da *Star-Tree* que foi gerada na etapa anterior. A *thread* tem a tarefa de consumir a tupla que está no recurso aplicando a função de agregação, determinada pelo usuário, sobre o conjunto de medidas da tupla retirada do recurso.

Em um computador *multicore* com dois núcleos, a paralelização ocorre da seguinte maneira: A primeira tupla é enviada para a *thread 1*. A *thread 1* consulta as medidas e aplica a função de agregação. No caso da função de agregação ser espacial a *thread* aplica a função de agregação através da biblioteca GeoTools e cria a visualização da região. A segunda tupla é enviada para a *thread 2* que executa os mesmos passos. Como no exemplo só há duas *threads* a próxima tupla será requisitada pela *thread* que ficar livre primeiro e assim por diante até que não haja mais nenhuma tupla no recurso a ser consumida. Todo o processo pode ser feito em memória primária. Depois desta etapa o cubo está computado. Na figura 31 está um exemplo de como esta paralelização acontece:

Ao criar o GeoCube houve a preocupação de desenvolver um sistema SOLAP abrangente e com desempenho, usabilidade e diferentes funcionalidades SOLAP. Alguns dos trabalhos relacionados focam em características específicas e não ofe-

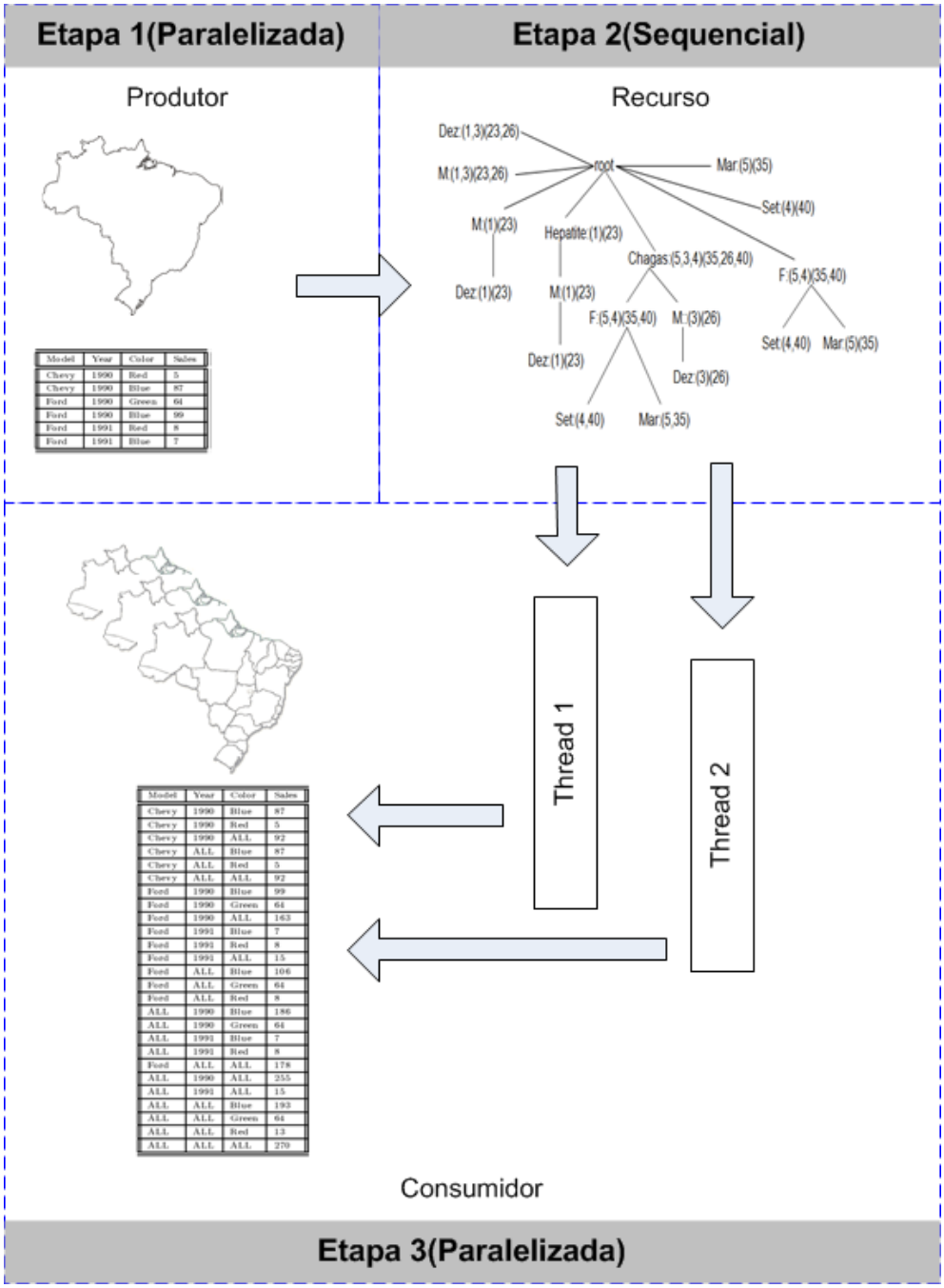


Figura 31: Paralelização do GeoCube.

recem soluções abrangentes. Na tabela 4 ilustra-se a comparação do sistema GeoCube com os trabalhos relacionados.

Sistema/Abordagem/Operador	Medidas Espaciais	Múltiplas Medidas Espaciais	Dimensão Espacial	Visual. 3D	Paralelismo	Medidas Híbridas	Hierarquia Automática	Atualização Incremental
GewOlap (2007)	X		X			X		X
SOVAT (2005)			X					
MapCube (2001)	X	X	X			X		
Jmap (2005)			X					
GlobeOLAP (2010)			X	X				
MapWharehouse (2007)	X		X					
GOLAPA/GDW (2004)			X					
PostGeoOlap/GeoOlap (2008)			X					
PostGis (2012)	X	X	X			X		
GeoMondrian (2012)	X	X	X			X		X
GeoCube	X	X	X	X	X	X	X	

Tabela 4: Tabela comparativa dos sistemas existentes e o GeoCube

Na próxima seção é mostrado os testes de desempenho que é uma grande lacuna nos trabalhos relacionados.

5 *Testes de Desempenho*

Nesta seção são apresentados os testes de desempenho do sistema GeoCube. Os testes mostram o *Speed-Up* (Tempo exeução sequencial / Tempo execução paralelo) e a Eficiência (Tempo execução sequencial / (Tempo execução paralelo $*$ Número de *threads* usado na execução paralela)) do código paralelizado, assim como a comparação com o sistema *PostGis* gerando todos os *group-by´s* necessários e os armazenando numa vista materializada única.

Para os testes desta seção foi usada uma base de dados que representa os 5566 municípios brasileiros obtidos através do site do IBGE[1]. Foram adicionadas outras dimensões com atributos numéricos como a população de cada município no ano de 2010 [2] e o número de casos de dengue por município no ano de 2010 [3].

O computador usado para os testes possui a seguinte configuração: Sistema Operacional windows server 2003 64 bits. JVM(*Java Virtual Machine*) 6, 2 HDs de 320GB 7200rpm, 16GB RAM DDR2 667, oito núcleos de processamento com 2.2GHz.

[1] ftp://geoftp.ibge.gov.br/malhas_digitais/municipio_2007/escala_2500mil/proj_geografica_sad69/brasil/

[2] http://www.sidra.ibge.gov.br/bda/tabela/listabl.asp?z=t&o=25&i=P&c=3145

[3] http://dtr2004.saude.gov.br/sinanweb/tabnet/dh?sinannet/dengue/bases/denguebrnet.def

Consulta	Hierarquia	Dimensão	Medida\Função de Agregação
C1	1	Região	Polígono\União, Dengue\Soma

Tabela 5: Número de casos de dengue por região. Consulta C1.

Cada teste foi repetido cinco vezes, sendo que os resultados com o menor e o com o maior tempo foram removidos. O resultado usado foi a média dos três experimentos restantes.

5.1 Testes Multithread

Nesta seção são apresentados os testes de desempenho usando múltiplas *threads* na computação de diferentes cubos de dados. Não são feitos testes comparativos de desempenho usando múltiplas *threads*, pois não foram encontrados outros sistemas SOLAP que também computem os cubos paralelamente.

5.1.1 Consulta 1

A primeira consulta retorna o número de casos de dengue por região a partir do mapa base composto pelos 5566 municípios brasileiros, Tabela 5.

Através do gráfico 32 é possível visualizar que a medida que aumentamos o número de *thread* o ganho no tempo de execução, que pode ser visto através da inclinação da curva, vai diminuindo. Este comportamento é confirmado através dos gráficos de Speed-Up e Eficiência da consulta C1.

Como mostrado no gráfico 33 a medida que o número de *threads* aumenta a eficiência diminui, desta forma pode-se atribuir este comportamento às seguintes

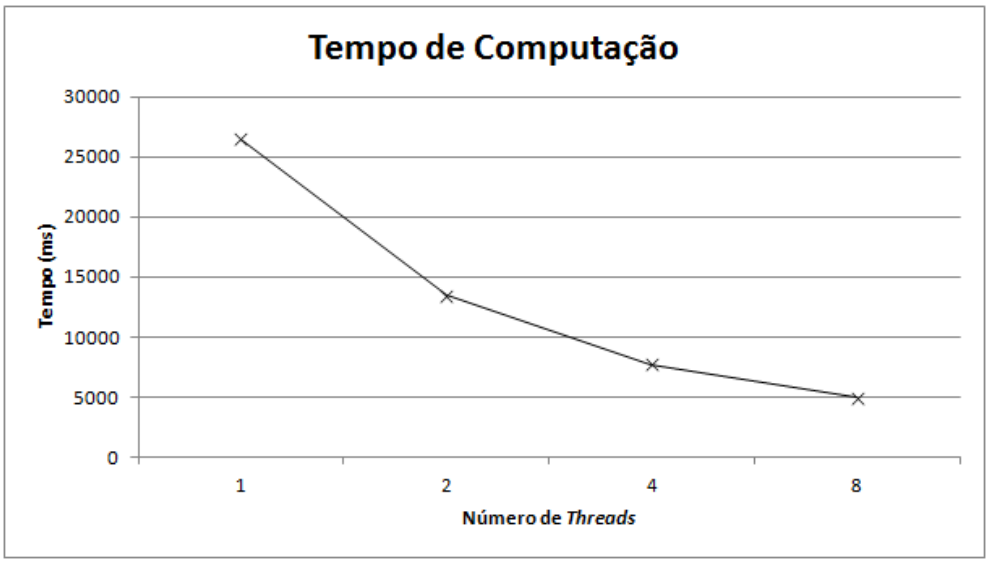


Figura 32: Tempo de Computação para a consulta C1.

razões:

- Não existe solução 100% paralela. Sempre há pelo menos início e fim sequenciais;
- A medida que o número de *threads* aumenta, a concorrência pelo sistema de memória compartilhado também aumenta. A máquina utilizada possui acesso usando barramentos e não *switches*, portanto há um represamento no sistema de memória da máquina;
- O sistema operacional pode a qualquer momento realizar a troca de contexto, portanto nunca é possível garantir 100% do uso de CPU.

5.1.2 Consulta 2

A próxima consulta usada tem como intuito criar hierarquias a partir de um mapa base possibilitando operações de *drill-down* e *roll-up*, ou seja, a partir do

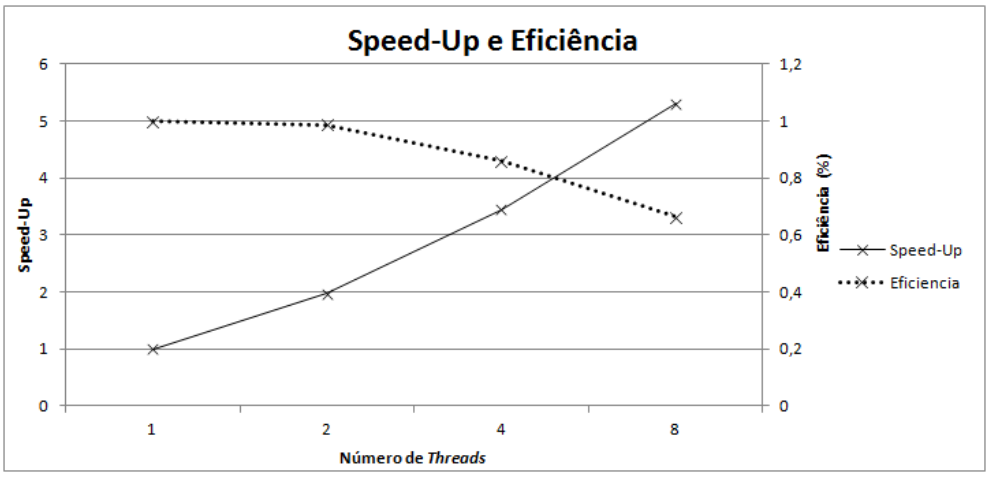


Figura 33: Speed-Up e Eficiência na computação da consulta C1.

mapa base formado pelos municípios do Brasil é criado quatro novas camadas com diferentes níveis de granularidade. Nos testes não foi usado o recurso de hierarquização automática do GeoCube pois, o sistema PostGis não possui este recurso. As hierarquias foram formadas com base nas dimensões Micro-Região , Meso-Região, Estado e Região agregando as regiões, população e o número de casos de dengue. Esta consulta foi escolhida pois ela possibilita a formação das hierarquias e a aplicação das funções de agregação espaciais paralelamente. A Tabela 6 exibe os parâmetros da consulta. Note que o *SpeedUp* e eficiência são similares em C1 e C2. Os tempos de execução são distintos, uma vez que C2 é mais complexa do que C1. Os gráficos das figuras 32 e 34 apenas ilustram quanto tempo efetivamente leva uma consulta.

Consulta	Hierarquia	Dimensão	Medida\Função de Agregação
C2	1	Município	Polígono\União, População\Soma, Dengue\Soma
C2	2	Micro-Região	Polígono\União, População\Soma, Dengue\Soma
C2	3	Meso-Região	Polígono\União, População\Soma, Dengue\Soma
C2	4	Estado	Polígono\União, População\Soma, Dengue\Soma
C2	5	Região	Polígono\União, População\Soma, Dengue\Soma

Tabela 6: Formando hierarquias agregando as regiões a população e o número de casos de dengue

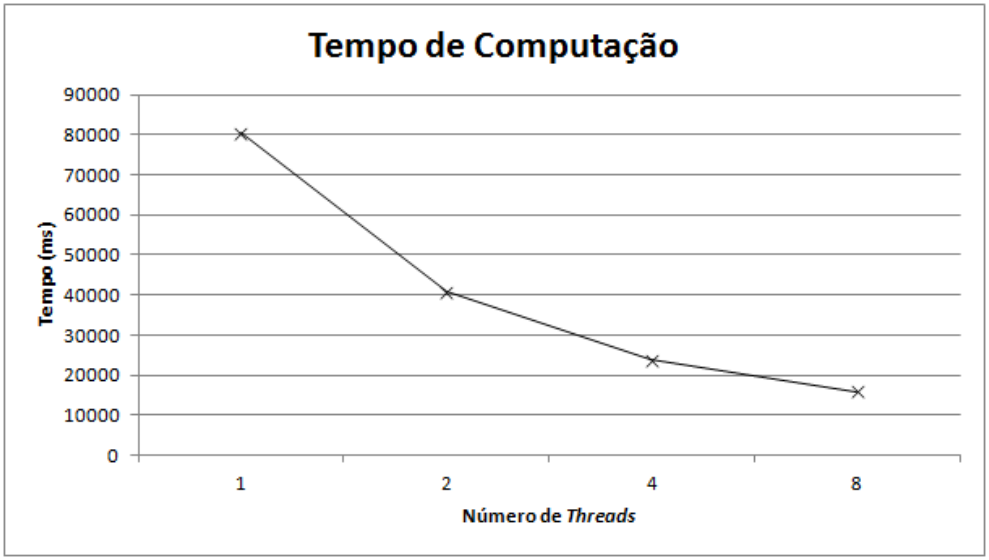


Figura 34: Tempo de Computação para a consulta C2.

5.2 Teste Comparativo

Nesta seção é mostrada os testes comparativos com o sistema PostGis. O sistema PostGis foi escolhido pois é amplamente usado para manipulação de agregações de objetos espaciais.

Os cubos computados tem como finalidade obter as regiões que tocam em cada região somando o número de casos de dengue, população e fazendo a união

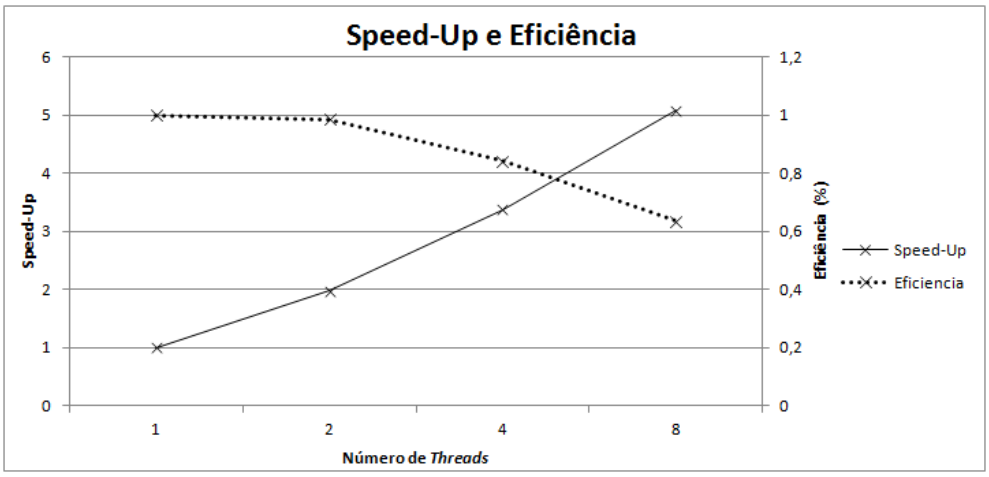


Figura 35: Speed-Up e Eficiência para a consulta C2.

Cubo	Hierarquia	Dimensão	Medida\Função de Agregação
C3	1	Município	Polígono\Toca, Polígono\União, População\Soma, Dengue\Soma
C4	1	Micro-Região	Polígono\Toca, Polígono\União, População\Soma, Dengue\Soma
C5	1	Meso-Região	Polígono\Toca, Polígono\União, População\Soma, Dengue\Soma
C6	1	Estado	Polígono\Toca, Polígono\União, População\Soma, Dengue\Soma
C7	1	Região	Polígono\Toca, Polígono\União, População\Soma, Dengue\Soma

Tabela 7: Parâmetros usados para computação dos cubos usados para comparação dos sistemas GeoCube e PostGis.

das regiões. Ou seja, agregar as regiões, população e número de casos de dengue das regiões vizinhas. O resumo dos cubos é mostrado na Tabela 7. Como o sistema PostGis não possui uma versão paralela os testes comparativos com o GeoCube foram feitos usando apenas uma *Thread* para computar o cubo nos dois sistemas.

Para cubos de pequeno porte (C5, C6 e C7) o número de polígonos gerados pelas bibliotecas espaciais é baixo, portanto os sistemas GeoCube e PostGis apresentaram tempo de computação semelhantes. Para a computação de cubos de dados maiores o sistema GeoCube possui tempo de computação por volta de 15%

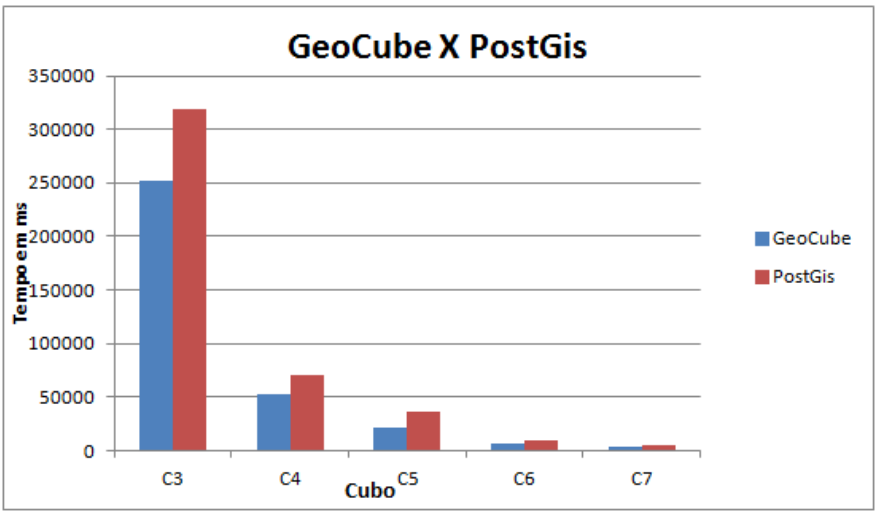


Figura 36: Comparação do tempo de computação dos sistemas GeoCube e PostGis.

mais rápido. O resultado reinforça adoção da representação e computação de cubos específicos, sem adoção de SGBDs, pois cubos maiores ou mais complexos são comumente requeridos.

6 Conclusão e Trabalhos Futuros

Neste trabalho foi possível verificar que existem poucos algoritmos para a computação de cubo de dados espaciais que ofereçam suporte a medidas, dimensões e funções de agregação espaciais, e os existentes não possuem um desempenho satisfatório. Sendo assim foi proposto o GeoCube, que mostrou ser bastante promissor, apresentando desempenhos satisfatórios se comparado com o PostGis, sendo por volta de 15% mais eficiente que o PostGIS.

Além destes avanços, é necessário deixar claro que o GeoCube é o único até o momento a utilizar o benefício da paralelização. Além disso o GeoCube oferece interface para criação de consultas e visualização 3D dos mapas o que torna sua utilização mais intuitiva. Outra contribuição foi a formação de hierarquias baseadas em regras de vizinhança. Este recurso possibilita operações de *drill-down* e *roll-up* sem que tenhamos informações das hierarquias das regiões nas dimensões.

Como trabalho futuro os autores propõem abordagens para atualização do cubo sem que haja necessidade de re-computação do cubo de dados e a utilização de otimizações eficientes para redução de sub-grafos, propostas na abordagem

MCG (de Castro Lima, 2009). Serão feitos testes avaliando o impacto que a multiplicidades de medidas espaciais traz ao GeoCube, assim como o número de dependências para cada medida espacial. Também serão fetios testes comparativos com o sistema MapCube, pois desta forma pode-se comparar o GeoCube com um sistema especializado na computação de cubos.

Referências Bibliográficas

Agrawal and Srikant, 1994 Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Beyer and Ramakrishnan, 1999 Beyer, K. and Ramakrishnan, R. (1999). Bottom-up computation of sparse and iceberg cube. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, SIGMOD '99, pages 359–370, New York, NY, USA. ACM.

Bimonte et al., 2007 Bimonte, S., Tchounikine, A., and Miquel, M. (2007). Spatial olap: Open issues and a web based prototype. In *10th AGILE International Conference on Geographic Information Science*, pages 1–11.

Bimonte S., 2005 Bimonte S., Tchounikine A., M. M. (2005). Towards a spatial multidimensional model. In *Proceedings of the 8th ACM international workshop on Data warehousing and OLAP*, DOLAP '05, pages 39–46, New York, NY, USA. ACM.

Casanova et al., 2005 Casanova, M., Camara, G., Davis, C., Vinhas, L., and de Queiroz, G. R., editors (2005). *Bancos de Dados GeogrÃ¡ficos*. MundoGEO.

Chaudhuri and Dayal, 1997 Chaudhuri, S. and Dayal, U. (1997). An overview of data warehousing and olap technology. *SIGMOD Rec.*, 26(1):65–74.

Colonese et al., 2008 Colonese, G., Manhães, R. S., González, S. M., Candido, U., and Campos, M. (2008). Uma ferramenta em software livre para o desenvolvimento de sistemas de suporte a decisão. *Search*.

de Castro Lima, 2009 de Castro Lima, J. (2009). Sequential and parallel approaches to reduce the data cube size. São Jose dos Campos, Brasil.

do Nascimento Fidalgo et al., 2004 do Nascimento Fidalgo, R., Times, V. C., and da Fonseca de Souza, F. (2004). Golapa: Uma arquitetura aberta e extensivel para integracão entre sig e olap.

Ferraz and Santos, 2010 Ferraz, V. R. T. and Santos, M. T. P. (2010). Globeolap - improving the geospatial realism in multidimensional analysis environment. In *ICEIS (5)'10*, pages 99–107.

Findlater and Hamilton, 2003 Findlater, L. and Hamilton, H. J. (2003). Iceberg-cube algorithms: An empirical evaluation on synthetic and real data. *Intell. Data Anal.*, 7(2):77–97.

Gray et al., 1997 Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., and Pirahesh, H. (1997). Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Min. Knowl. Discov.*, 1(1):29–53.

Han et al., 1998a Han, J., Stefanovic, N., and Koperski, K. (1998a). Selective materialization: An efficient method for spatial data cube construction. In *In*

Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'98, pages 144–158.

Han et al., 1998b Han, J., Stefanovic, N., and Koperski, K. (1998b). Selective materialization: An efficient method for spatial data cube construction. In *In Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDDa-pos;98*, pages 144–158.

http://geoportal.icimod.org, 2012 http://geoportal.icimod.org (2012).

Inmon and Hackathorn, 1994 Inmon, W. H. and Hackathorn, R. D. (1994). *Using the data warehouse*. Wiley-QED Publishing, Somerset, NJ, USA.

Moreno et al., 2009 Moreno, F., Arango, F., and Fileto, R. (2009). Extending the map cube operator with multiple spatial aggregate functions and map overlay. In *Geoinformatics, 2009 17th International Conference on*, pages 1 –7.

Rivest et al., 2003 Rivest, S., Bedard, Y., Proulx, M.-J., and Nadeau., M. (2003). Solap: a new type of user interface to support spatio-temporal multidimensional data exploration and analysis. In *Paper presented at the ISPRS Joint ghop of WG II/5, II/6, IV/1 and IV/2 on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis*.

Rivest et al., 2005 Rivest, S., Bedard, Y., Proulx, M.-J., Nadeau, M., Hubert, F., and Pastor, J. (2005). Solap technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. *Journal of Photogrammetry and Remote Sensing (ISPRS)*, 60(1):17 – 33. Including Theme Section: - Advances in Spatio-temporal Analysis and Representation.

Sandro Bimonte, 2006 Sandro Bimonte, Anne Tchounikine, M. M. (2006). Geocube, a multidimensional model and navigation operators handling complex measures: Application in spatial olap. *Advances in Information Systems*, 4243/2006:100–109.

Scotch and Parmanto, 2005 Scotch, M. and Parmanto, B. (2005). Sovat: Spatial olap visualization and analysis tool. *Hawaii International Conference on System Sciences*, 6:142b.

Shekhar et al., 2001 Shekhar, S., Lu, C., Tan, X., and Chawla, S. (2001). *Map Cube: A Visualization Tool for Spatial Data Warehouses*, pages 74–109. Harvey j. miller and jiawei han (eds.) edition.

Sousa, 2007 Sousa, A. G. D. (2007). Concepcão e validacão de um modelo multidimensional para data warehouse espacial. Master's thesis, Universidade Federal de Campina Grande.

Technologies, 2005 Technologies, K. (2005). (2005) jmap spatial olap: On-line analytical processing for spatial databases.

White and Running, 1994 White, J. D. and Running, S. W. (1994). Testing scale dependent assumptions in regional ecosystem simulations. *Journal of Vegetation Science*, 5(5):687–702.

Xin et al., 2007 Xin, D., Han, J., Li, X., Shao, Z., and Wah, B. W. (2007). Computing iceberg cubes by top-down and bottom-up integration: The starcubing approach. *IEEE Transactions on Knowledge and Data Engineering*, 19:111–126.

Zhao et al., 1998 Zhao, Y., Deshpande, P. M., and Naughton, J. F. (1998). Readings in database systems (3rd ed.). chapter An array-based algorithm for simultaneous multidimensional aggregates, pages 568–579. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.