PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Universidade Federal do Rio de Janeiro

Tatiane Lima da Silva

HEURÍSTICAS PARA FRAGMENTAÇÃO DE BASES DE DADOS XML

DISSERTAÇÃO DE MESTRADO

RIO DE JANEIRO

2013

Instituto de Matemática

Núcleo de Computação Eletrônica

TATIANE LIMA DA SILVA

HEURÍSTICAS PARA FRAGMENTAÇÃO DE BASES DE DADOS XML

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Orientadoras:
Profa. Vanessa Braganholo Murta, D. Sc.
Profa. Jonice de Oliveira Sampaio, D. Sc.

Rio de Janeiro
2013

S586 Silva, Tatiane Lima da.
Heurísticas para fragmentação de bases de dados XML. / Tatiane Lima da Silva. – 2013.
133 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Núcleo de Computação Eletrônica, Programa de Pós Graduação em Informática

Orientadoras: Vanessa Braganholo Murta
Jonice de Oliveira Sampaio

1. Heurística. 2. Fragmentação de Bases de Dados. 3. XML – Teses. I. Murta, Vanessa Braganholo (Orient.). II. Sampaio, Jonice de Oliveira (Orient.). III. Universidade Federal do Rio de Janeiro Instituto de Matemática, Núcleo de Computação Eletrônica, Programa de Pós -Graduação em Informática. IV. Título

CDD

TATIANE LIMA DA SILVA

HEURÍSTICAS PARA FRAGMENTAÇÃO DE BASES DE DADOS XML

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Aprovada em: Rio de Janeiro, 21 de fevereiro de 2013.

_____

(Profa. Vanessa Braganholo Murta, DSc., UFF)

_____

(Profa. Jonice de Oliveira Sampaio, DSc., UFRJ)

_____

(Profa. Fernanda Araújo Baião, DSc., UNIRIO)

_____

(Profa. Maria Luiza Machado Campos, PhD., UFRJ)

_____

(Prof. Paulo de Figueiredo Pires, DSc., UFRJ)

Aos meus pais Maria e Wilson e ao meu marido Diego.

AGRADECIMENTOS

Agradeço a Deus por ter me dado esta oportunidade de ser o que sou hoje e ter me proporcionado coisas maravilhosas ao longo da minha vida, como a conclusão deste curso.

Aos meus pais, Wilson e Maria, que são os alicerces da minha vida e que não mediram esforços para que eu chegasse aonde eu cheguei.

Ao meu marido, Diego, que esteve ao meu lado em toda esta fase da minha vida dando muita força e trabalhando junto comigo na conclusão desse trabalho. Agradeço o seu apoio nos momentos difíceis e por ter tido paciência durante essa longa jornada.

Às professoras, Fernanda Baião e Marta Mattoso, por terem contribuído com sua participação no artigo que publicamos no Simpósio Brasileiro de Banco de Dados.

À minha orientadora, Vanessa Braganholo, que me ajudou muito nesse trabalho e que contribuiu para que esse trabalho fosse realizado. Agradeço a sua paciência e dedicação em todos os momentos.

À minha orientadora, Jonice Oliveira, por ter aceitado ser minha orientadora junto com a professora Vanessa Braganholo.

Aos professores da banca, por terem aceitado o convite de participação.

Ao PPGI, por ter me dado a oportunidade de alcançar mais um patamar em minha vida profissional.

Ao IC/UFF, por ter me proporcionado a infraestrutura necessária para a execução dos meus experimentos. Um agradecimento em especial ao Carlos, responsável pelo cluster utilizado nesse projeto, que atendeu sempre os meus pedidos de forma rápida e eficiente.

Aos amigos do trabalho, que sempre me apoiaram nessa jornada e que se orgulhavam da minha perseverança em conciliar trabalho e estudo.

Ao CNPq e FAPERJ, que financiaram o meu projeto.

Finalmente, minha gratidão a todos que, direta ou indiretamente, contribuíram para que esse trabalho se realizasse.

RESUMO

SILVA, Tatiane Lima da. **Heurísticas para fragmentação de bases de dados XML**. 2013. 133s. Dissertação (Mestrado em Informática). Programa de Pós-Graduação em Informática, Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.

O volume de dados de coleções de documentos XML e o tempo de resposta do processamento de consultas em sistemas de bancos de dados (SGBD) sobre tais coleções tornaram-se pontos críticos para muitas aplicações. Uma alternativa para melhorar o desempenho de consultas consiste em reduzir o tamanho das coleções de documentos XML através do uso de fragmentação de dados.

No que diz respeito à fragmentação física dos dados (foco do presente trabalho), o potencial de ganho de desempenho é obtido em função do paralelismo no processamento da consulta. A consulta é segmentada em partes e enviada para diferentes nós que as executam em paralelo sobre um volume menor de dados em cada nó. Por outro lado, a fragmentação de uma base de dados também pode degradar o desempenho de uma consulta, quando, por exemplo, sua execução sobre a base fragmentada exige o processamento de junções para reconstruções que não eram necessárias na consulta original, entre outros motivos. Por isso, no projeto de fragmentação da base de dados é preciso analisar as consultas mais frequentes, para que a fragmentação proporcione ganho de desempenho na maioria das consultas realizadas sobre a base de dados distribuída. Além disso, a ausência de avaliações de desempenho de consultas sobre bases fragmentadas dificulta as decisões de um projetista de dados quanto ao projeto de fragmentação.

Na maioria das abordagens encontradas na literatura assume-se que os projetistas já sabem como fragmentar uma base de dados XML de forma eficaz e em muitos casos essa decisão não é trivial. Sendo assim, essa dissertação apresenta um conjunto de heurísticas para fragmentação vertical e horizontal de dados XML a fim de auxiliar o projetista no momento de definição dos fragmentos. As heurísticas propostas foram derivadas das utilizadas nos modelos relacional e orientado a objetos, que já possuem seus conceitos consolidados na literatura. Dessa forma, nessa dissertação realizamos uma séria de experimentos sobre bases de dados XML que nos permitiram fundamentar as heurísticas para a fragmentação de dados XML.

Palavras-Chaves: fragmentação, XML, banco de dados distribuídos, vertical, horizontal

ABSTRACT

SILVA, Tatiane Lima da. **Heurísticas para fragmentação de bases de dados XML**. 2013. 133s. Dissertação (Mestrado em Informática). Programa de Pós-Graduação em Informática, Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.

The large volume of XML data collections and the response time of query processing in database systems (DBMS) over such collections have become critical for many applications. An alternative to improve the performance of XML queries is to reduce the size of those collections through data fragmentation.

In physical fragmentation approaches (focus of this work), the potential performance gain is obtained by the parallelism. The query is segmented into parts and sent to different nodes that run on parallel over a smaller volume of data in each node. On the other hand, the fragmentation of a database can also degrade the performance of a query when, for example, the execution of queries over the fragmented database requires processing several joins, which were not needed in the original query, among other reasons. Therefore, in the fragmentation design of the database one needs to analyze the frequent queries so that it provides performance gains for most queries over the distributed database. Moreover, the absence of performance evaluations of queries over fragmented databases makes it difficult to decide what a good fragmentation design is.

Most of the approaches found in literature assume that designers already know how to efficiently fragment a database and in many cases this decision is not trivial. Thus, this work presents a set of heuristics to vertical and horizontal fragmentation of XML data in order to assist the designer when defining the fragments. The proposed heuristics were derived from the ones used in the relational and object-oriented models. In this dissertation we used knowledge acquired with these two models and applied then to XML through experiments, which in turn allowed us to substantiate heuristics for XML data fragmentation.

Keywords: fragmentation, XML, distributed database, vertical, horizontal

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES

BEA: *Bond Energy Algorithm*

DC: Centrado em Dados (*Data Centric)*

DBMS: *Database Management System*

DTD: *Document Type Definition*

FHP: Fragmentação Horizontal Primária

FHD: Fragmentação Horizontal Derivada

MD: Múltiplos Documentos (*Multiple Document)*

OO: Orientado a Objetos

SD: Documento Único (*Single Document)*

SGBD: Sistema de Gerenciamento de Banco de Dados

XML: *eXtensible Markup Language*

SUMÁRIO

CAPÍTULO 1: INTRODUÇÃO

1.1 MOTIVAÇÃO

Devido ao grande volume de dados predominantemente armazenado em bancos de dados relacionais, há uma grande preocupação com o desempenho no processamento de consultas nestes ambientes e, consequentemente, inúmeros estudos nesta área. Dentre as abordagens utilizadas para resolver o problema de processamento de consultas sobre um grande volume de dados está a fragmentação dos dados e alocação dos fragmentos em diversos nós de uma rede (KOSSMANN, 2000; MIRANDA et al., 2006; OZSU; VALDURIEZ, 2011; PAES et al., 2008). Desta forma, as consultas são executadas em paralelo, sobre porções menores dos dados, o que ocasiona aumento de desempenho (ANDRADE et al., 2006; KOSSMANN, 2000).

O processo que determina como uma base de dados deve ser fragmentada é denominado "projeto de distribuição" (OZSU; VALDURIEZ, 2011). O projeto de distribuição normalmente é subdividido em duas etapas: fragmentação e alocação. A etapa de fragmentação define a forma com que os dados serão fragmentados e realiza a fragmentação propriamente dita. Já a etapa de alocação define a melhor maneira de alocar os fragmentos oriundos da etapa anterior em diferentes nós de uma rede.

A etapa de fragmentação tem como objetivo definir um esquema lógico de fragmentação para as relações do esquema da aplicação (OZSU; VALDURIEZ, 2011). A definição de como fragmentar uma base de dados leva em consideração as propriedades de acesso aos dados da aplicação. A análise desse padrão de acesso é muitas vezes subjetiva e, em geral, é baseada em heurísticas. Dentro do processo de fragmentação são definidas as seguintes etapas (BAIÃO et al., 2004; OZSU; VALDURIEZ, 2011):

i. **Extração de dados relevantes**: Extração dos dados que serão utilizados como insumos para a etapa seguinte (análise), como, por exemplo, principais atributos utilizados nas consultas frequentes;

ii. **Análise**: Avaliação de todas as informações da aplicação e do esquema lógico dos dados para que possa ser escolhida a técnica de fragmentação que melhor se aplica a uma determinada base de dados;

iii. **Fragmentação**: Definição dos algoritmos para a fragmentação. Nessa etapa, a fragmentação efetiva dos dados é realizada.

Baseado nas informações levantadas nas etapas (i) e (ii), a etapa (iii) define a técnica de fragmentação a ser utilizada e o número de fragmentos a serem gerados. As técnicas básicas de fragmentação existentes são: fragmentação horizontal, fragmentação vertical e fragmentação híbrida (BAIÃO et al., 2004; FLORENTINO, 2003; OZSU; VALDURIEZ, 2011).

Nas propostas existentes na literatura para os modelos relacional (OZSU; VALDURIEZ, 2011) e orientado a objetos (BAIÃO et al., 2004), o grande desafio é o desenvolvimento de uma metodologia que contemple a fase de análise (segunda fase do processo de fragmentação), onde uma avaliação é realizada sobre um conjunto de variáveis que permite determinar a forma mais adequada para se fragmentar uma base de dados.

Em contrapartida ao cenário do ambiente relacional, documentos XML deixaram de ser utilizados apenas para troca de dados, e se tornaram um importante formato de representação de dados, permitindo o desenvolvimento de aplicações web flexíveis, integração de dados oriundos de varias fontes, manipulação de dados de múltiplas aplicações, entre outros (MORO et al., 2009).

Este fato faz surgir a necessidade de desenvolvimento de metodologias para processamento eficiente de consultas sobre dados XML (FIGUEIREDO et al., 2010; KLING et al., 2010a). Aproveitando as ideias de fragmentação e distribuição propostas para o modelo relacional e orientado a objetos, vários trabalhos na literatura têm focado em processamento de consultas XML em ambientes distribuídos e também na criação de técnicas de fragmentação no que diz respeito ao formato dos fragmentos e os algoritmos que os formam (ABITEBOUL et al., 2009; ANDRADE et al., 2006; BREMER; GERTZ, 2003). As propostas de projeto de distribuição de dados XML existentes na literatura assumem que o projetista já sabe de que forma a base de dados XML deve ser fragmentada (ABITEBOUL et al., 2009; KLING et al., 2010b).

De fato, nas abordagens de distribuição de dados XML, um dos pontos mais explorados na literatura é justamente o processamento de uma consulta em um ambiente distribuído (ANDRADE et al., 2006; BREMER; GERTZ, 2003; MA; SCHEWE, 2003) enquanto que o projeto de fragmentação de dados XML ainda é um ponto pouco explorado. Conforme discutido por

Figueiredo, Braganholo e Mattoso (2010), de nada adianta uma metodologia para processamento de consultas distribuídas se a base de dados não estiver fragmentada adequadamente, para que as consultas mais frequentes se beneficiem da fragmentação.

Têm-se assim os pontos que definem a problemática que motiva essa dissertação: (i) a realidade do processamento distribuído no ambiente WEB; (ii) o uso intensivo de XML como forma de representação de dados; (iii) a ausência de trabalhos que tratem a fase de análise dentro do processo de fragmentação, visto que essa etapa permite definir de maneira eficaz a forma como a base deve ser fragmentada. A fragmentação deve permitir que as consultas mais frequentes se beneficiem do processo de fragmentação e, consequentemente, acarreta uma melhora no desempenho da aplicação, possibilitando o paralelismo, eliminando acessos desnecessários a dados irrelevantes e, por conseguinte, diminuindo o volume de dados manipulados pelas consultas e transações.

1.2 OBJETIVOS

O principal objetivo dessa dissertação é contribuir com heurísticas que permitam determinar como são os parâmetros relacionados tanto à base de dados quanto às consultas *XQuery* submetidas pelos usuários que devem ser levados em consideração durante o processo de fragmentação de dados XML em ambientes distribuídos, especificamente, na etapa de análise. Essas heurísticas utilizam como fundamentação o que hoje está definido para o modelo relacional (OZSU; VALDURIEZ, 2011) e para o modelo orientado a objetos (BAIÃO et al., 2004; FLORENTINO, 2003).

Para a construção das heurísticas foram utilizados e/ou adaptados conceitos e protótipos já desenvolvidos na literatura, conforme apresentado a seguir.

1. Utilizamos as definições de tipos de fragmentos XML propostas por Andrade et al., (2006). Essas definições incluem regras de correção que garantem a integridade da base de dados XML após a sua fragmentação, e esse foi o principal motivo da escolha delas nesse trabalho. Além disso, as definições de fragmentos Andrade et al., (2006) são muito próximas daquelas para o modelo relacional. Isso nos permite usufruir das informações analíticas da fragmentação nesse modelo. Para o modelo relacional, as questões relativas ao processo de fragmentação de base de dados já está consolidado na literatura.

2. Utilizando as definições de fragmentos propostas por Andrade et al., (2006), Figueiredo, Braganholo e Mattoso (2010) propõem uma metodologia que permite efetuar o processamento de consultas *XQuery* em ambientes distribuídos. A metodologia assume que os fragmentos foram gerados e armazenados nos diversos nós da rede de forma otimizada. Nesse trabalho, utilizamos o protótipo construído por Figueiredo, Braganholo e Mattoso (2007), o qual foi adaptado com o objetivo de melhorar o desempenho de execução das consultas. A escolha dessa metodologia/protótipo justifica-se pelo fato dela suportar os tipos de fragmentos definidos por Andrade et al., (2006).
3. A partir das definições de fragmentos e metodologia de processamento de consultas, foram executados experimentos a fim de derivar heurísticas a partir dos resultados obtidos. Para essas análises foram utilizados os *benchmarks XBench* (YAO et al., 2004) e *XMark* (BUSSE et al., 2002) como base de dados. Os documentos contidos nas bases foram fragmentados de diversas formas, levando em conta os atributos e predicados das consultas do próprio *benchmark*, as quais foram consideradas como consultas frequentes. A definição dos fragmentos foi realizada de acordo com os tipos de fragmentos descritos por Andrade et al*.,* (2006).
4. Após a etapa anterior, as consultas contidas no *benckmark* foram submetidas ao protótipo construído por Figueiredo, Braganholo e Mattoso (2007), que foi adaptado para funcionar em ambiente de cluster. Ao final, os tempos de execução foram coletados e a partir dos critérios já definidos para outros modelos fizemos análises para verificar o que se aplica a bases de dados XML ou não.
5. Após as análises uma lista de parâmetros relevantes foi levantada, e as heurísticas que compõem a fase de análise no processo de distribuição de dados XML em ambientes distribuídos foram definidas. A definição das heurísticas definidas consiste na principal contribuição desse trabalho.

1.3 ORGANIZAÇÃO DO TEXTO

Os estudos e análises realizados no desenvolvimento dessa dissertação estão organizados em capítulos, como segue.

O Capítulo 2 apresenta a fundamentação teórica dessa dissertação. O capítulo descreve o PartiX, que define o conceito de fragmentação de dados XML utilizado nesse

trabalho. Além disso, apresentamos o protótipo utilizado nos experimentos para prover processamento de consultas sobre bases XML distribuídas.

O Capítulo 3 apresenta o estado da arte sobre projetos de distribuição para o modelo relacional, orientado a objetos e, por último, para bases de dados XML. Nesse capítulo, apresentamos os trabalhos relacionados a metodologias para fragmentação de dados XML.

O Capítulo 4 traz todos os passos necessários para a realização dos experimentos dessa dissertação. Nessa etapa do trabalho, o objetivo é observar o comportamento dos resultados obtidos através do uso do protótipo descrito no Capítulo 2. Com base nestas observações, apresentamos todas as regras utilizadas para a composição das heurísticas para a fase de análise em um projeto de fragmentação de dados XML em ambientes distribuídos. O capítulo apresenta a descrição de toda a parte preparativa para a execução do experimento.

O Capítulo 5 apresenta os resultados obtidos com os experimentos executados. A partir desses resultados, as heurísticas para a fragmentação horizontal e vertical propostas nesse trabalho são descritas.

Por fim, o Capítulo 6 apresenta as conclusões desse trabalho, onde são reportadas algumas considerações finais e as contribuições desta dissertação, além de algumas indicações para trabalhos futuros.

CAPÍTULO 2: FRAGMENTAÇÃO DE DADOS XML

Devido ao crescente aumento no volume de dados armazenados nas organizações, surge a necessidade da aplicação de técnicas que permitam a realização de consultas sobre as bases de dados de forma mais eficiente. Para atender essa necessidade, uma alternativa é adotar processamento paralelo das consultas. Para que isso seja possível, é necessário que os dados passem por um processo de fragmentação e alocação sobre diversos nós de uma rede (OZSU; VALDURIEZ, 2011). A fragmentação da base de dados busca obter ganhos de desempenho através do paralelismo da execução da consulta, já que a consulta é distribuída e enviada aos diferentes nós que a executam em paralelo sobre um volume menor de dados em cada nó.

Por outro lado, a fragmentação de uma base de dados também pode degradar o desempenho de uma consulta. Por isso, o projeto da fragmentação da base de dados precisa analisar os padrões de consultas mais frequentes para que a fragmentação apresente ganho de desempenho na maioria das consultas realizadas sobre a base de dados.

Esta dissertação propõe heurísticas para a fase de análise dentro do processo de fragmentação de documentos XML sobre ambientes distribuídos com intuito de minimizar o tempo de resposta das consultas submetidas. Para isso, neste capítulo apresentamos os conceitos necessários para o entendimento da abordagem proposta.

A Seção 2.1 apresenta as definições de fragmentos XML utilizadas nesse trabalho, propostas pelo PartiX (ANDRADE et al., 2006). A metodologia para processamento de consultas sobre bases XML distribuídas proposta pelo PartiX (FIGUEIREDO et al., 2010) é apresentada na Seção 2.2. Por último, Seção 2.3 conclui esse capítulo.

2.1 PARTIX: DEFINIÇÃO DE FRAGMENTAÇÃO DE DADOS XML

O processamento de consultas sobre grandes volumes de dados XML é uma questão que merece destaque, principalmente na Web. O projeto PartiX (ANDRADE et al., 2006) resolve parte do problema propondo uma forma de distribuir dados XML através da criação de fragmentos. Apesar de existirem diversas definições de fragmentos XML na literatura (GERTZ; BREMER, 2003; KLING et al., 2010a; MA; SCHEWE, 2003; PAGNAMENTA, 2005),

nosso trabalho utiliza o conceito proposto por Andrade et al., (2006), pois essa é a que mais se aproxima da definição de fragmentos utilizada no modelo relacional (OZSU; VALDURIEZ, 2011). Essa escolha é essencial, já que desejamos aproveitar as ideias de projeto de fragmentação propostas para o modelo relacional (OZSU; VALDURIEZ, 2011), já consolidadas na literatura, nesse trabalho de mestrado.

Antes de apresentarmos em detalhes a definição de fragmentos para XML proposta pelo PartiX, é preciso apresentar os conceitos de coleção e repositório XML. Uma coleção *C* de documentos XML é um conjunto de árvores. A coleção é dita *homogênea* se todos os documentos da coleção C seguem o mesmo esquema. Caso contrário, a coleção é dita *heterogênea*.

O projeto PartiX utiliza a classificação para repositórios XML definida por Yao e Khandelwal (2004), no qual um repositório composto por vários documentos é denominado de "**Múltiplos Documentos**" (*Multiple Documents*, MD), enquanto que um repositório que contém um único documento é chamado de repositório de "**Documento Único**" (*Single Document*, SD). Para exemplificar, os repositórios do tipo SD abrangem bases de dados que consistem em um único documento com estruturas complexas, como um catálogo, por exemplo. Já no caso de repositórios MD tem-se um conjunto de documentos XML, tais como um arquivo de documentos de notícias ou dados transacionais.

2.1.1 DEFINIÇÃO DE FRAGMENTAÇÃO

O PartiX (ANDRADE et al., 2006) define três tipos de fragmentação: *horizontal*, que divide os documentos em fragmentos distintos através de predicados de seleção; *vertical*, que quebra a estrutura de dados através de projeções; e, por último, a *híbrida*, que combina as operações de seleção e projeção. Esses três tipos de fragmentos são análogos aos existentes no modelo relacional (OZSU; VALDURIEZ, 2011). Apresentamos abaixo, a definição genérica de um fragmento XML definido no PartiX (ANDRADE et al., 2006).

Definição 1: *Um fragmento F de uma coleção homogênea C é uma coleção representada por F:= <C, γ>, onde γ denota uma operação definida sobre C. F é horizontal se γ denota uma seleção (σ); vertical, se o operador de γ é uma projeção (π); ou híbrido, quando ocorre uma composição de operadores de seleção e projeção. Um fragmento só será válido se todos os documentos gerados por γ forem bem-formados*.

Resumidamente, a proposta feita por Andrade et al. (2006) consiste no agrupamento de documentos que atendem algum predicado de seleção na fragmentação horizontal e a quebra de um mesmo documento em diversos segmentos de acordo com algum predicado de projeção no caso da fragmentação vertical. A Figura 1 apresenta os resultados das fragmentações horizontais e verticais aplicadas sobre uma base XML contendo dois documentos. Note que, na figura, ambas as fragmentações foram aplicadas sobre a mesma base de dados, como alternativas.

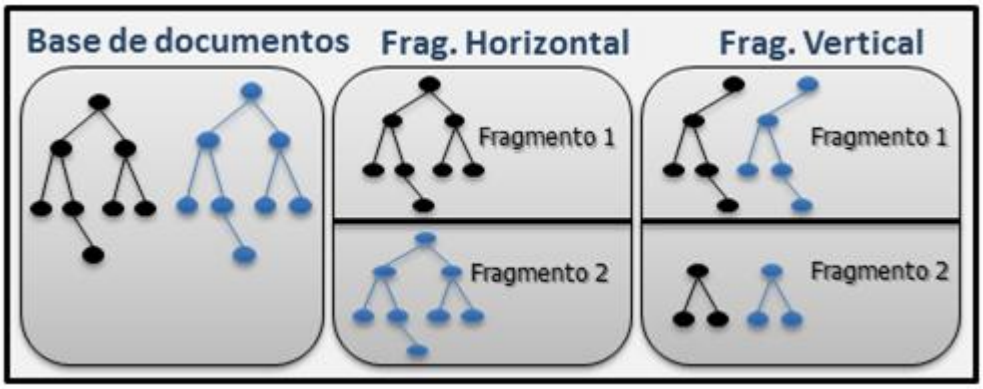


Figura 1: Funcionamento da fragmentação Horizontal e Vertical

Nas próximas subseções são apresentados de forma detalhada os três tipos de fragmentação propostos no PartiX e as regras de correção definidas para garantir a integridade da fragmentação.

2.1.1.1 FRAGMENTAÇÃO HORIZONTAL

Um fragmento horizontal *F* de uma coleção *C* é definido pelo operador de seleção (σ) aplicado sobre os documentos em *C*, onde o predicado de σ é uma expressão booleana contendo um ou mais predicados simples. O fragmento *F* tem o mesmo esquema que *C* (ANDRADE, 2006).

Neste tipo de fragmentação é preciso que a coleção pertença a um repositório de múltiplos documentos, ou seja, a fragmentação horizontal não pode ser aplicada a repositórios de um único documento. A exceção ocorre quando faz-se primeiro uma fragmentação vertical, e, posteriormente, a horizontal.

Definição 2: *Seja μ uma conjunção de predicados simples sobre a coleção C. O fragmento horizontal de C definido por μ é dado pela expressão F:=<C, σ_μ>, onde σ_μ denota a seleção de documentos em C que satisfazem μ, isto é, F contém documentos de C para os quais σ_μ é verdadeiro* (ANDRADE et al., 2006).

Para exemplificar o funcionamento da fragmentação horizontal, a Figura 2 apresenta um esquema referente a contatos de uma agenda e 2 documentos XML que seguem o esquema de Contatos. Suponha que os documentos estão armazenados na Coleção $C_{Contatos}$.

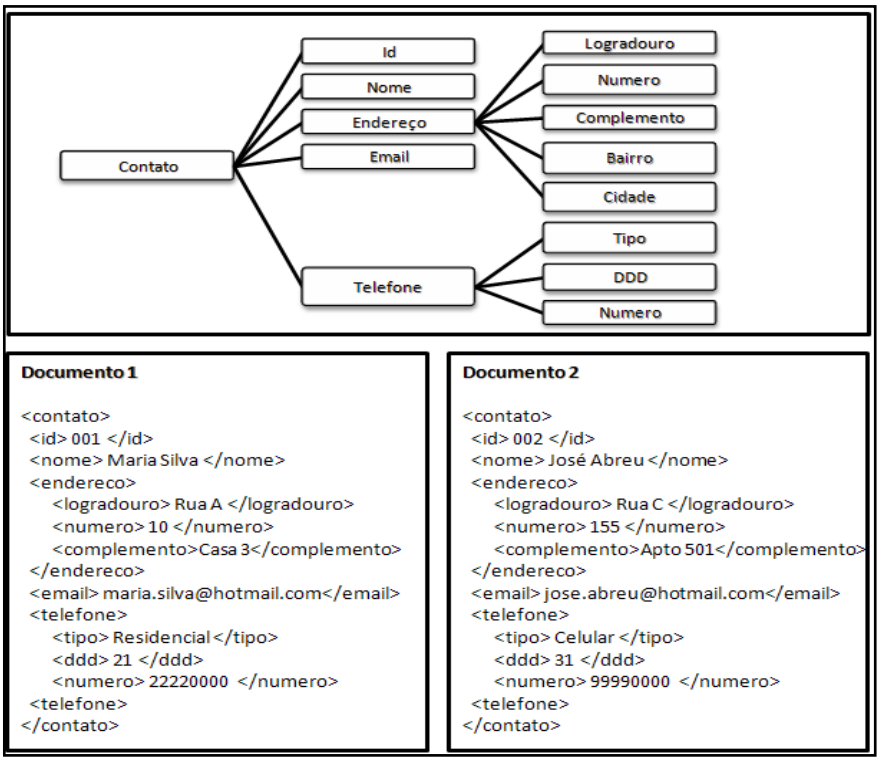


Figura 2: Exemplo de esquema e documentos XML

A Figura 3 apresenta a especificação e os resultados esperados para a fragmentação horizontal da coleção $C_{Contatos}$ da Figura 2. Por exemplo, o fragmento $F1_{Residencial}$ reúne os documentos da coleção $C_{Contatos}$ que possuem conteúdo do nó **tipo** igual a **Residencial**. Por isso, o documento 1 pertence a este fragmento. Já o fragmento $F2_{Residencial}$ agrupa os documentos cujo conteúdo de **tipo** difere de **Residencial**. Logo, o documento 2 que possui tipo igual a **Celular** irá compor o fragmento $F2_{Residencial}$, conforme descrito na Figura 3.

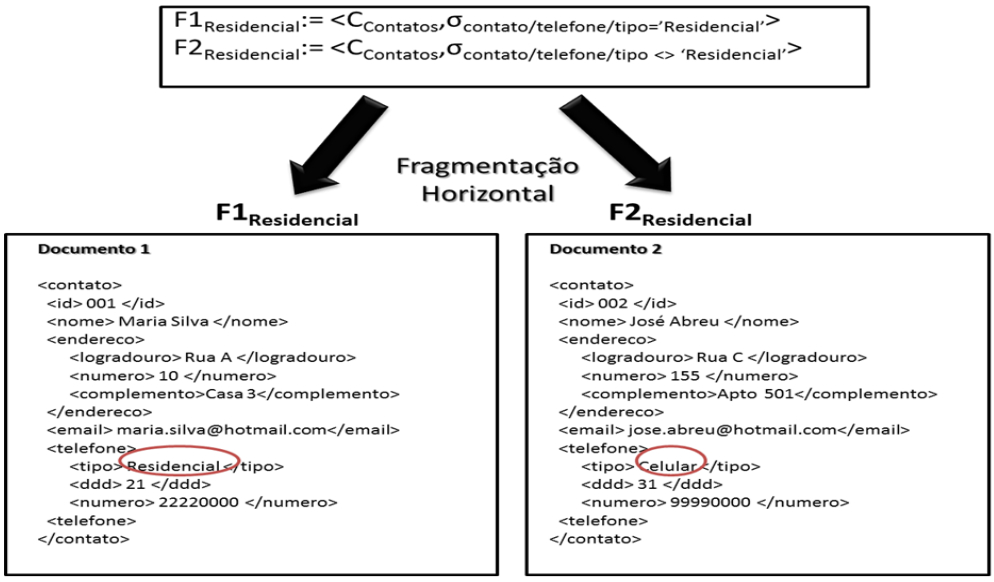


Figura 3: Exemplo de definição de fragmentos sobre a coleção $C_{Contatos}$

2.1.1.2 FRAGMENTAÇÃO VERTICAL

Segundo Andrade et al., (2006), a fragmentação vertical pode ser obtida através da aplicação de uma operação de projeção (π), tendo como objetivo a "quebra" da estrutura de dados em partes menores que são frequentemente acessadas nas consultas. Diferente da fragmentação horizontal, este tipo de fragmentação pode ser executada sobre os repositórios de únicos ou múltiplos documentos.

Definição 3: *Seja P uma expressão de caminho sobre a coleção C, e Γ = {E_1,...,E_x} um conjunto de expressões de caminhos contidos em P (isto é, expressões de caminho em que P é um prefixo). Um fragmento vertical de C definido por P é caracterizado por F:=<C, $\pi_{P,\Gamma}$>, onde $\pi_{P,\Gamma}$ caracteriza a projeção das subárvores enraizadas pelos nós selecionados por P, excluindo do resultado os nós selecionados pelas expressões em Γ. O conjunto Γ é chamado de critério de poda de F* (ANDRADE et al., 2006).

A Figura 4 mostra alguns exemplos de definição de fragmentos verticais da coleção $C_{Contatos.}$ O resultado da fragmentação é apresentado sobre o esquema da coleção apenas para facilitar o entendimento porque, na verdade, o que sofre a fragmentação são os documentos. O fragmento $F1_{Contatos}$ traz como raiz o elemento Contato e efetua a poda da subárvore /Contato/Telefone enquanto o fragmento $F2_{Contatos}$ traz apenas a subárvore Telefone.

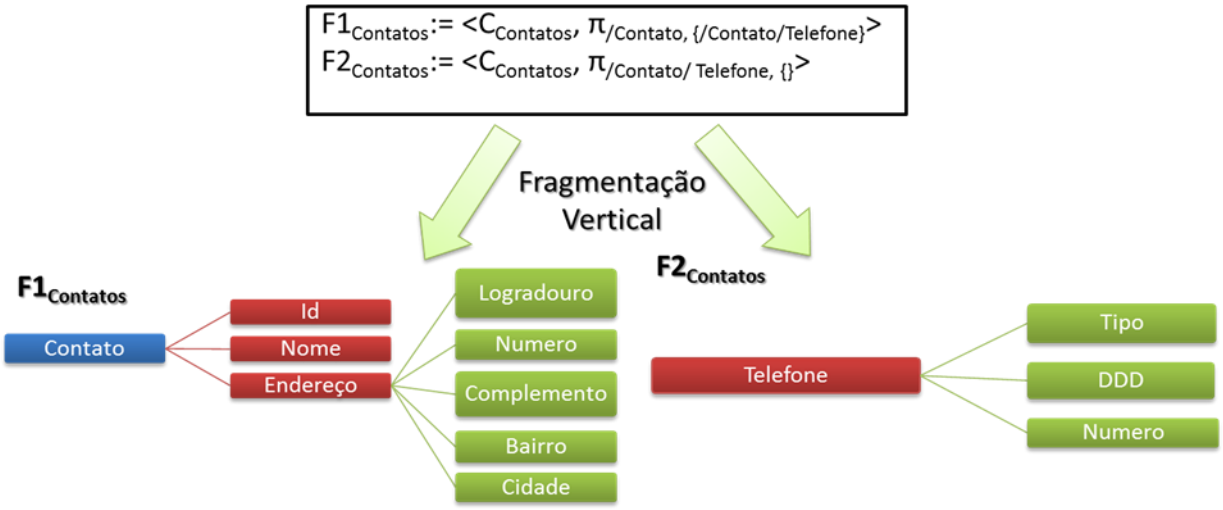


Figura 4: Exemplo de definição de fragmentos verticais sobre a coleção $C_{Contatos}$

É importante ressaltar que, pela definição apresentada por Andrade et al., (2006), não é possível especificar fragmentos verticais sobre um elemento que possui cardinalidade superior a 1. O objetivo dessa restrição é garantir que a fragmentação gere fragmentos bem formados, sem a necessidade de geração de elementos artificiais no momento da reorganização das subárvores projetadas.

2.1.1.3 FRAGMENTAÇÃO HÍBRIDA

A fragmentação híbrida de uma coleção XML pode ser definida pela aplicação da fragmentação vertical seguida pela fragmentação horizontal ou vice-versa. Neste tipo de fragmentação é possível efetuar a fragmentação horizontal sobre esquemas de coleções de repositórios que contenham um único documento, desde que primeiro seja feita uma fragmentação vertical. A ordem da aplicação das operações de seleção e projeção depende do projeto de fragmentação escolhido, uma vez que produzem resultados diferentes.

Definição 4: *Seja σ_μ e π_P operações de seleção e projeção, respectivamente, definidas sobre uma coleção C. Um fragmento híbrido de C é representado por F:=<C, $\pi_{P,\Gamma} \bullet \sigma_\mu$>, onde $\pi_{P,\Gamma} \bullet \sigma_\mu$ representa a seleção das subárvores projetadas por $\pi_{P,\Gamma}$ que satisfazem σ_μ* (ANDRADE et al., 2006).

A Figura 5 apresenta um exemplo de fragmentação híbrida sobre a coleção de Contatos. No exemplo, o fragmento $F1_{Híbrido}$ contém uma projeção de /Contato/Telefone cujo tipo é Residencial. $F2_{Híbrido}$ efetua a mesma projeção, mas com o atributo tipo diferente de Residencial. Por último, a projeção do restante da árvore garante a completude do processo de fragmentação, conforme descrito pelo fragmento $F3_{Híbrido}$.

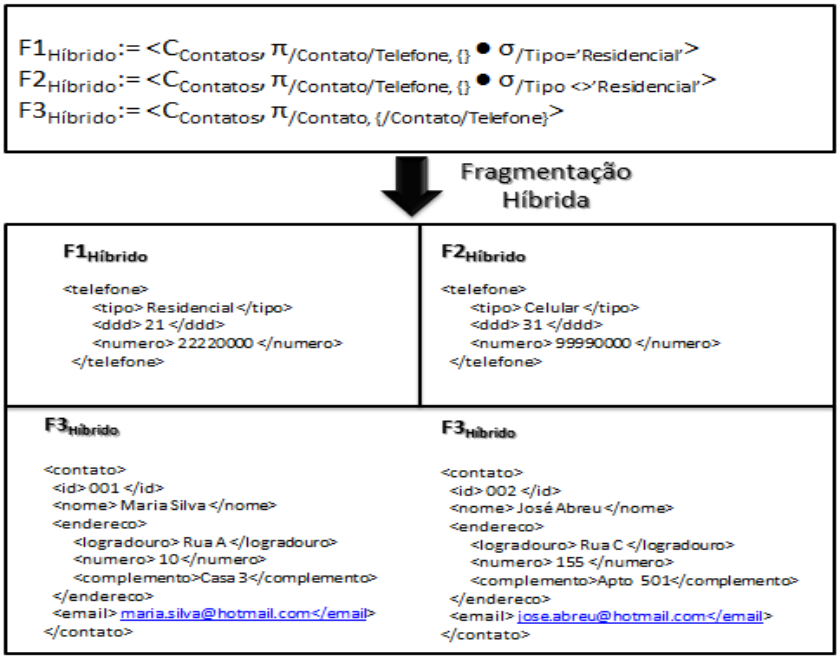


Figura 5: Exemplo de fragmentação híbrida sobre a coleção $C_{Contatos}$

2.1.1.4 REGRAS DE CORREÇÃO

Todo projeto de fragmentação precisa garantir que não há perda de dados. Por isso, o PartiX propõe três regras de correção para fragmentação de bases XML, baseadas nas definições de corretude apresentadas para o modelo relacional (OZSU; VALDURIEZ, 2011). As regras definidas são:

- **Completude**

  Se uma coleção C é decomposta em fragmentos F_1, F_2,...,F_N, cada item de dado em C precisa estar em um dos F_i *(1<i<n)*.

- **Reconstrução**

  Se uma coleção C é decomposta em um conjunto de fragmentos Φ = {F_1,F_2,...,F_N}, deve ser possível definir um operador Δ tal que C = ΔF_i, para todo $F_i \in \Phi$, onde *Δ* é o operador responsável pela reconstrução da coleção C. Esse operador será diferente para cada tipo de fragmentação.

- **Disjunção**

Se uma coleção C é horizontalmente decomposta em um conjunto de fragmentos $\Phi = \{F_1,F_2,...,F_N\}$, e o documento d_i está em F_j, então este não pode estar em nenhum outro fragmento F_k ($k \neq j$).

Se a coleção C é verticalmente decomposta em um conjunto de fragmentos $\Phi = \{F_1,F_2,...,F_N\}$, temos que ter cuidados adicionais. Precisamos, para isso, de uma lista com todas as expressões de caminho P que levem a nós terminais p_i. Então, os fragmentos são disjuntos se a expressão de caminho p_i que está no fragmento F_j não estiver em nenhum outro fragmento F_k ($k \neq j$).

2.2 PROCESSAMENTO DE CONSULTAS SOBRE BASES XML DISTRIBUÍDAS

Para efetuarmos os experimentos em nosso trabalho foi necessário o uso de um protótipo que nos permitisse efetuar consultas sobre bases XML distribuídas, e consequentemente, a visualização do tempo de resposta durante a execução. Por isso, buscamos na literatura uma implementação que nos permitisse efetuar essas análises e escolhemos o projeto de Figueiredo, Braganholo e Mattoso (2010). Nessa seção, apresentamos de forma resumida o funcionamento dessa implementação e a sua fundamentação teórica.

O projeto de Figueiredo, Braganholo e Mattoso (2010), como mencionado anteriormente, consiste em uma metodologia que permite o processamento de consultas *XQuery* sobre bases de dados XML distribuídas onde há a decomposição da consulta principal em subconsultas que são executadas nos nós remotos que contêm os fragmentos de uma coleção global. Assim como na maioria das propostas de processamento de consultas sobre banco de dados relacional (OZSU; VALDURIEZ, 2011), a consulta principal passa por diversas etapas: decomposição, localização, otimização, a criação das subconsultas e a execução das mesmas sobre os fragmentos XML distribuídos.

A decomposição consiste na modificação de uma consulta *XQuery* em uma expressão algébrica sobre coleções globais. Nessa etapa, o processamento das consultas passa pelo mesmo processo que no ambiente centralizado, ou seja, possui as fases de validação sintática e semântica das consultas; simplificação das consultas; e a representação das consultas em forma algébrica como apresentado na Figura 6.

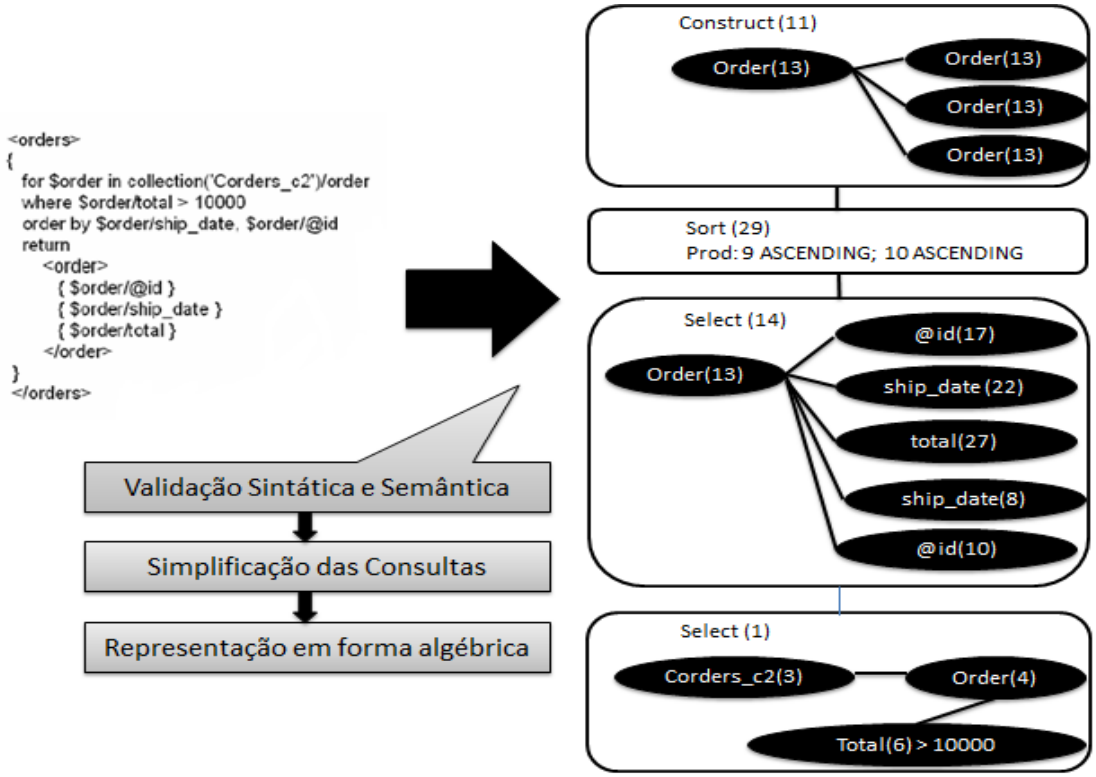


Figura 6: Representação de consulta em sua forma algébrica TLC (FIGUEIREDO, 2007)

A etapa seguinte é a localização, que visa substituir as referências a coleções globais do plano algébrico por referências a fragmentos locais. Essa etapa elimina o acesso aos fragmentos irrelevantes trazendo com isso uma melhora no desempenho das consultas, pois há uma diminuição no volume de dados acessados.

Dentro da etapa de otimização são executados dois passos: a otimização global, que busca a obtenção de um plano algébrico de custo mínimo através de variações sobre o plano algébrico obtido na etapa de localização; e a otimização local, que é aplicada pelo próprio banco de dados que armazena o fragmento XML.

Para finalizar o processo, há a execução da consulta distribuída utilizando o plano algébrico localizado, reduzido e otimizado, onde é possível identificar o local de execução de cada operação algébrica do plano tendo como entrada a localização dos fragmentos da coleção global. Cada grupo de operações corresponde a uma subconsulta a ser executada por um nó remoto ou pelo próprio protótipo, para a composição do resultado final.

A arquitetura proposta por Figueiredo, Braganholo e Mattoso (2010) visa prover um sistema totalmente transparente para o usuário, ocultando os detalhes da localização e da fragmentação da base. As consultas são submetidas sobre um Mediador onde são

decompostas em um conjunto de subconsultas, que então são executadas pelos nós remotos sobre os fragmentos. Os resultados de cada subconsulta são retornados ao mediador para a construção do resultado final.

A Figura 7 apresenta um resumo de como funciona a implementação da arquitetura, que é composta por um mediador e diversos adaptadores. A função do mediador é receber consultas e deixar o ambiente distribuído transparente para o usuário. Já os adaptadores são acoplados a cada uma das bases locais para esconder possíveis heterogeneidades das fontes de dados. Dessa forma, o mediador se comunica com os adaptadores, que são responsáveis por enviar as subconsultas para a base de dados local.

A consulta *XQuery* é submetida ao mediador, que efetua o processamento da consulta dividindo-a em subconsultas. Em seguida, o mediador verifica através do catálogo as configurações do ambiente do distribuído, ou seja, define para quais adaptadores cada uma das subconsultas deve ser submetida. Após isso, os adaptadores executam as suas consultas em bases locais e depois devolvem o resultado ao mediador que por sua vez consolida e retorna o resultado final ao usuário.

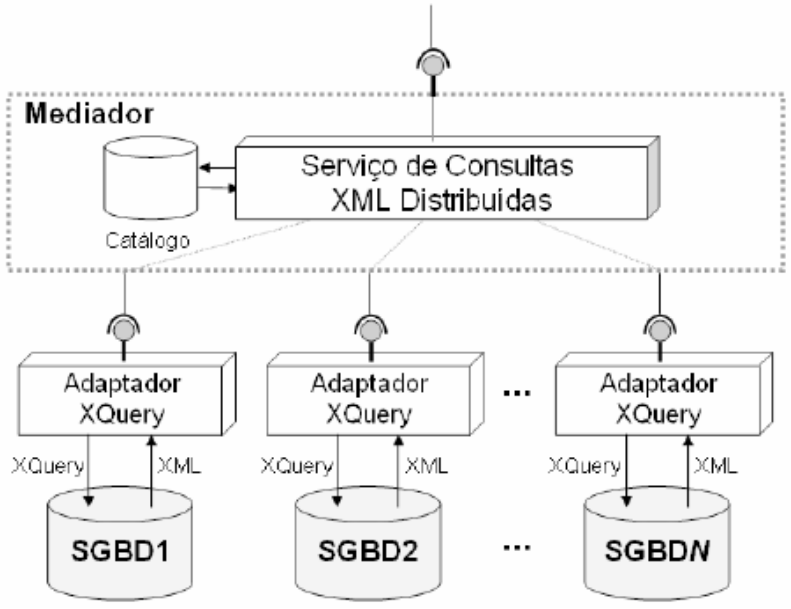


Figura 7: Arquitetura do Mediador (FIGUEIREDO, 2007)

O mediador é o principal componente, pois é responsável pelo processamento da consulta distribuída, realizando as etapas descritas anteriormente (decomposição, localização e a otimização global da consulta). A Figura 8 apresenta os módulos internos do mediador.

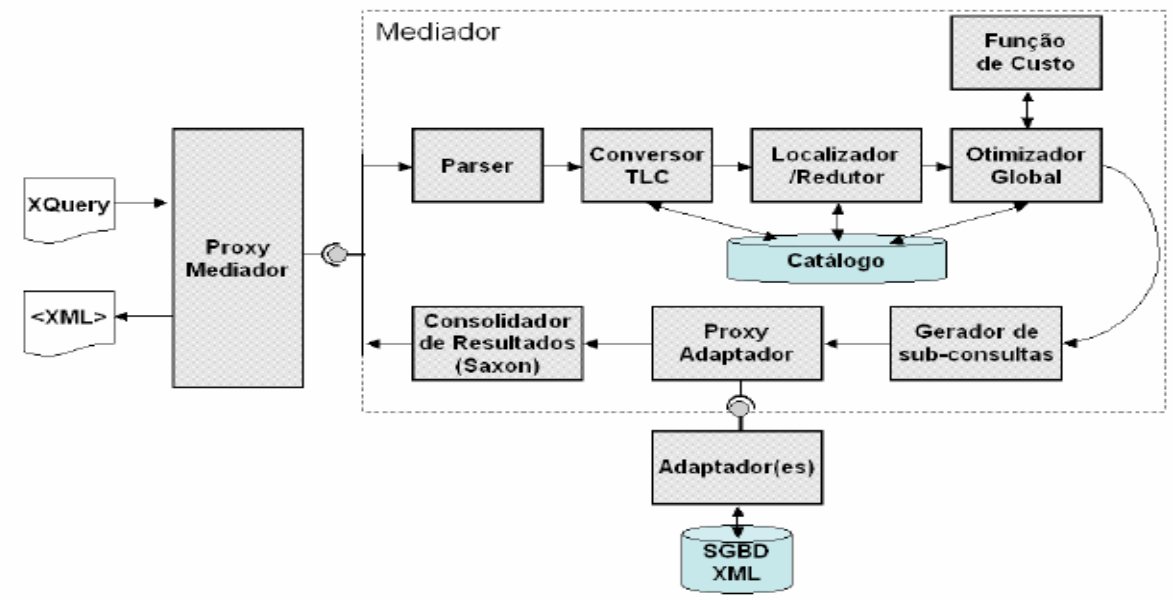


Figura 8: Módulos internos do mediador (FIGUEIREDO, 2007)

Como podemos ver na Figura 8, o catálogo é utilizado pelo mediador para armazenar todas as informações que são necessárias para o processamento das consultas distribuídas. Dentre as informações armazenadas, uma das mais importantes são as definições dos fragmentos, pois a partir da relação dos critérios de seleção ou projeção que formam os fragmentos, o mediador é capaz de efetuar a reconstrução da coleção global a partir dos fragmentos.

Já o adaptador tem como função ser a interface de comunicação entre o mediador e o banco de dados local. Além disso, ele é responsável por atualizar a localização do documento XML para o seu endereço na base de dados local. Para isso, o adaptador possui um arquivo de configuração que contém o mapeamento entre o nome do documento (fragmento) com o seu endereço completo no servidor local.

2.3 CONSIDERAÇÕES FINAIS

Esse capítulo apresentou os conceitos de fragmentação para XML utilizados como fundamentação por esse trabalho. Além disso, o protótipo que implementa a metodologia de processamento de consultas distribuídas é utilizado para efetuarmos os experimentos a fim de obtermos as heurísticas para a fase de análise em um projeto de fragmentação de dados XML. Os detalhes da utilização desse protótipo estão descritos no Capítulo 4 dessa dissertação.

O próximo capítulo descreve as soluções de projeto fragmentação para o modelo relacional e orientado a objetos, que são utilizadas como fundamentação para os trabalhos relacionados à fragmentação de dados XML. Além disso, o capítulo apresenta uma metodologia para a fase analítica para bases de dados orientados a objetos que é usado para a composição das heurísticas propostas nessa dissertação.

CAPÍTULO 3: METOLOGIAS PARA PROJETO DE FRAGMENTAÇÃO DE BASES DE DADOS

O objetivo desse capítulo é apresentar os trabalhos relacionados e suas propostas de fragmentação para os modelos relacionais, orientado a objetos e XML. Resumidamente, esse capítulo está organizado da seguinte forma. A Seção 3.1 apresenta o funcionamento dos projetos de fragmentação consolidados na literatura para os modelos relacional e orientado a objetos. A Seção 3.2 apresenta os trabalhos ligados ao projeto de fragmentação envolvendo especificamente base de dados XML. Por último, é apresentada na Seção 3.3 a conclusão desse capítulo.

3.1 PROJETO DE FRAGMENTAÇÃO DE BANCOS DE DADOS

Dentro de qualquer projeto de fragmentação, a definição dos tipos de fragmentos é essencial, sendo essa questão bem explorada na literatura (ANDRADE et al., 2006; GERTZ; BREMER, 2003; MA; SCHEWE, 2003; OZSU; VALDURIEZ, 2011; PAGNAMENTA, 2005). A proposta de fragmentação de dados XML do PartiX (ANDRADE et al., 2006) é a mais próxima ao modelo relacional (OZSU; VALDURIEZ, 2011), que define três tipos de fragmentação possíveis.

Anteriormente foi mencionado que a maioria das abordagens sobre processamento de consultas e projeto de distribuição para XML são fundamentados no modelo relacional (ANDRADE et al., 2006; GERTZ; BREMER, 2003; KLING et al., 2010a). Por isso, é importante apresentar o funcionamento de projetos de fragmentação e alocação nesse modelo.

Para Ozsu e Valduriez (2011), um projeto de distribuição é dividido em **fragmentação** e **alocação**, onde a etapa de fragmentação é composta pelas seguintes subetapas.

1. Levantamento de informações relevantes tanto da base de dados quanto das consultas submetidas a ela;
2. Definição de fragmentos e suas regras de corretude, e, por último, a fragmentação propriamente dita.

Já na etapa de alocação os fragmentos gerados são alocados nos vários nós da rede, levando em consideração questões como a localização das aplicações, tamanho dos dados, frequências e padrões de acesso.

Nas seções subsequentes são apresentados o funcionamento dos projetos de fragmentação para os modelos relacional (OZSU; VALDURIEZ, 2011) e orientado a objetos (BAIÃO et al., 2004; FLORENTINO, 2003), respectivamente.

3.1.1 PROJETO DE FRAGMENTAÇÃO DE BANCO DE DADOS RELACIONAL

Dentro do projeto de distribuição, a fragmentação é a primeira etapa que objetiva a geração de fragmentos que são alocados de forma distribuída numa etapa posterior. No modelo proposto por Ozsu e Valduriez (2011), as principais questões que precisam ser avaliadas em um projeto de fragmentação são: avaliar as alternativas de fragmentação, definir o grau de fragmentação e descrever as regras de corretude para a fragmentação.

No que diz respeito às alternativas de fragmentação, Ozsu e Valduriez (2011) definem três modos alternativos de fragmentar uma tabela: **horizontal**, no qual os predicados de seleção são aplicados sobre as tabelas para definir os fragmentos; **vertical**, onde a fragmentação é realizada a partir de projeções definidas sobre as tabelas; e **híbrida**, que constitui da aplicação da fragmentação horizontal sobre a vertical ou vice-versa. De maneira simplificada, a fragmentação horizontal fragmenta as tabelas através de tuplas (linhas) enquanto a vertical é aplicada sobre atributos (colunas).

3.1.1.1 FRAGMENTAÇÃO HORIZONTAL

A fragmentação horizontal é subdividida em dois tipos: **primária** (FHP), que usa os predicados de seleção definidos sobre uma relação; **derivada** (FHD), cuja fragmentação de uma relação resulta da definição de predicados pertencentes a outra relação. A Figura 9 apresenta a metodologia de projeto de fragmentação para dados relacionais.

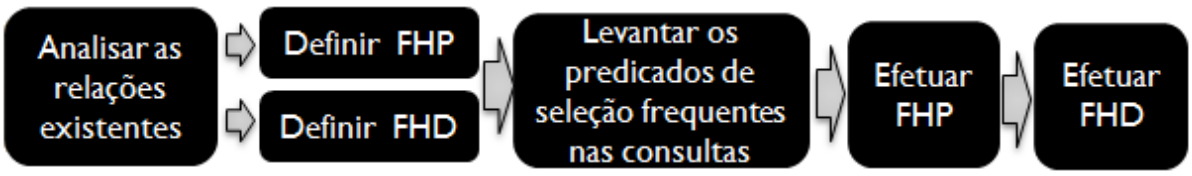


Figura 9: Fragmentação horizontal (ÖZSU; VALDURIEZ, 2011)

Dentro do processo de fragmentação horizontal, inicialmente há a analise das relações existentes a fim de definir qual das duas opções de fragmentação existentes mais se adéqua

a uma determinada relação. Segundo Ozsu e Valduriez (2011), existem alguns requisitos qualitativos e quantitativos a serem avaliados antes do processo de fragmentação horizontal.

As informações qualitativas orientam a atividade de fragmentação enquanto que as quantitativas são incorporadas aos modelos de alocação. Dentre as informações qualitativas que definem a fragmentação em si, temos as informações referentes ao esquema conceitual global, ou seja, os dados que descrevem relacionamentos entre as relações.

A Figura 10 mostra as relações entre as tabelas de uma base de dados. Observe que a orientação de uma ligação denota um relacionamento de um para vários, onde a relação origem da ligação é denominada *dono* (owner) e a outra relação é considerada *membro* (member). Dada a ligação L_1 da Figura 10, as funções *dono* e *membro* têm os seguintes valores: *Proprietário* (L_1) = PAG, *Membro* (L_1) = FUNC. A partir das definições das funções *dono* e *membro* sobre todas as relações que compõem a base de dados, define-se que as relações (tabelas) categorizadas como *dono* sofrerão fragmentação horizontal primária enquanto que as relações classificadas como *membro* sofrerão fragmentação horizontal derivada.

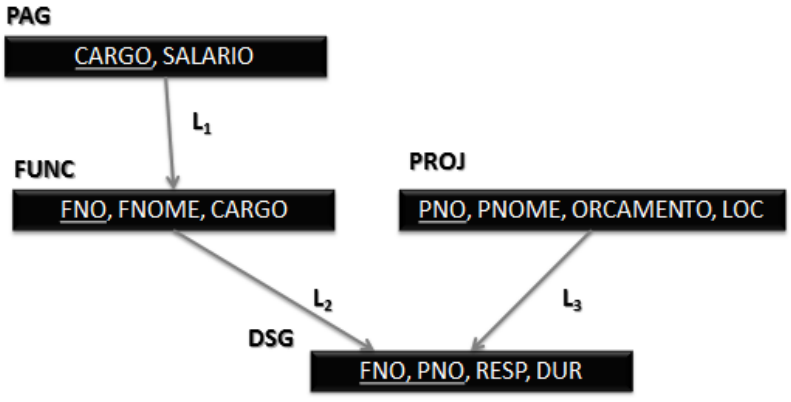


Figura 10: Relacionamentos entre relações (ÖZSU; VALDURIEZ, 2011) – Traduzido

Após a definição do tipo de fragmentação, os predicados utilizados nas consultas frequentes são avaliados para que possa ser definida a melhor forma de fragmentação dos dados. Essas consultas são fragmentadas em consultas simples que posteriormente são utilizadas para a criação do chamado predicado *mintermo*, que constitui uma conjunção de predicados simples. Para exemplificar o funcionamento dos predicados *mintermo,* suponha a relação PAG apresentada na Figura 10, composta dos atributos cargo e salário,

respectivamente. Além disso, suponha que os predicados das consultas mais frequentes sobre essa relação são:

p_1: Cargo = “Engenheiro”

p_2: Cargo = “Analista”

p_3: Cargo = “Programador”

p_4: Salário <= 3000

p_5: Salário > 3000

Dessa forma, alguns dos possíveis predicados *mintermo* construídos a partir dos predicados simples listados anteriormente seriam:

m_1: Cargo = “Engenheiro” ^ Salário <= 3000

m_2: Cargo = “Engenheiro” ^ Salário > 3000

m_3: NOT (Cargo = “Engenheiro”) ^ Salário <= 3000

m_4: NOT (Cargo = “Engenheiro”) ^ Salário > 3000

m_5: Cargo = “Programador” ^ Salário <= 3000

m_6: Cargo = “Programador” ^ Salário > 3000

Após a definição dos predicados *mintermo*, são obtidas também informações quantitativas. Essas informações são as seguintes.

- *Seletividade de mintermo*: quantidade de tuplas da relação que seriam acessadas por uma consulta de acordo com um dado predicado *mintermo*.
- *Frequência de acesso*: é a frequência com que a aplicação do usuário acessa um determinado dado no banco de dados.

Para o algoritmo de fragmentação horizontal apresentado por Ozsu e Valduriez (2011), inicialmente temos como entrada a relação candidata a fragmentação primária e o conjunto de predicados simples. A partir dos predicados simples são determinados o conjunto de *mintermos* e as implicações que serão aplicadas sobre esses *mintermos*, ou seja, regras que determinam a possibilidade de convivência entre os atributos dentro de um mesmo predicado *mintermo*. Por exemplo, suponha que após a geração dos *mintermos* sobre a relação PAG temos a seguinte situação:

m_1: Salário > 3000 ^ Salário < 2000

Por questões semânticas do banco de dados esse tipo de predicado não é válido, uma vez que as condições são contraditórias dentro da conjunção. Feito a eliminação dos

mintermos contraditórios ao conjunto de implicações definidas, os demais são definidos como os fragmentos horizontais a serem aplicados na base de dados.

Já o processo de fragmentação horizontal derivada depende do resultado obtido no processo de fragmentação primária, ou seja, o número de fragmentos derivados estará atrelado ao conjunto de fragmentos primários. Logo, para o processo de fragmentação horizontal derivada é preciso três entradas.

- O conjunto de fragmentos definidos para relações *dono*, ou seja, os fragmentos gerados na fragmentação primária.
- As relações *membro* sobre as quais se deseja aplicar a fragmentação horizontal derivada.
- Conjunto de predicados de semijunção entre as relações *dono* e *membro*.

3.1.1.2 FRAGMENTAÇÃO VERTICAL

A Figura 11 mostra as etapas da fragmentação vertical propostas por Ozsu e Valduriez (2011). Primeiramente é realizada a análise sobre as consultas frequentes submetidas à base de dados. Nessa analise é verificado o uso dos atributos nas diversas consultas executadas, ou seja, dada uma relação *R* que contém os atributos $\{A_1, A_2, ..., A_m\}$ e uma dada consulta q_i, o uso de um atributo é denotado por $uso(q_i, A_j)$, onde:

$uso(q_i, A_j)$ = 1 se o atributo A_j é referenciado pela consulta q_i;

0 em qualquer outro caso

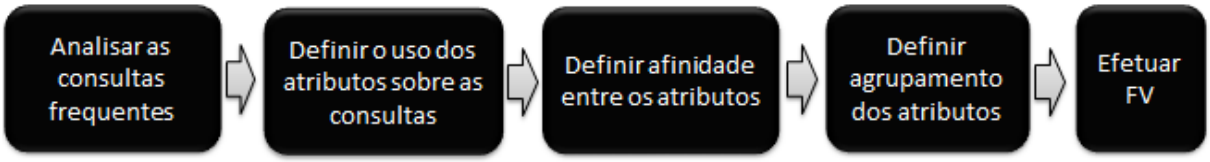


Figura 11: Fragmentação vertical (ÖZSU; VALDURIEZ, 2011)

Os valores de utilização de atributos não são gerais o suficiente para definir como gerar os fragmentos verticais. Isso ocorre porque esses valores não representam a frequência de acesso aos atributos pelos aplicativos. A medida de frequência pode ser incluída na definição da medida de afinidade de atributos $aff(A_i, A_j)$, que mede a ligação entre dois atributos de uma relação de acordo com o modo como eles são acessados pelas consultas. O resultado do cálculo de afinidade é uma matriz *n* x *n*, conhecida como matriz de afinidade de atributos (*Attribute Affinity Matrix*).

A tarefa fundamental no projeto de fragmentação vertical é encontrar algum meio de agrupar (formar *clusters*) os atributos de uma relação com base nos valores de afinidade dos atributos (matriz de afinidade de atributos). Baseado nessa matriz, Ozsu e Valduriez (2011) propõem utilizar o algoritmo de energia de ligação (*BEA Algorithm*), proposto por McCormick, Schweitzer e White (1972), com o objetivo de definir esse agrupamento dos atributos.

Após a obtenção desses grupos de atributos, há a necessidade de definir o melhor ponto de partição desses grupos. Sendo assim, Ozsu e Valduriez (2011) propõem a execução de um algoritmo de particionamento, a fim de aperfeiçoar a fragmentação desses grupos.

Na fragmentação vertical, todos os fragmentos possuem o atributo chave primária, para que a reconstrução da tupla possa ser realizada posteriormente, utilizando junção. Após a etapa de análise de afinidade, é executado o agrupamento dos atributos que visa formar os "grupos" que possuem alguma relação de acordo com as consultas frequentes. Por último, é efetuada a fragmentação da relação obedecendo também às regras de corretude definidas anteriormente.

3.1.1.3 FRAGMENTAÇÃO HÍBRIDA

Em alguns casos, uma fragmentação vertical ou horizontal simples de um determinado esquema de banco de dados não é suficiente para satisfazer aos requisitos de aplicativos do usuário. Nesse caso, uma fragmentação vertical pode ser seguida por uma horizontal ou vice-versa. Esse tipo de fragmentação é conhecido como híbrida, mista ou aninhada (OZSU; VALDURIEZ, 2011). A Figura 12 apresenta um exemplo de fragmentação híbrida, onde a tabela de Produtos foi fragmentada horizontalmente sobre o atributo preço (Frag1: preço >= 3,00 e Frag2 < 3,00). Após a geração dos dois fragmentos horizontais, aplicou-se a fragmentação vertical onde os atributos Nome e Descrição ficaram em um fragmento e os demais atributos (preço e estoque) em outro. Como podemos observar na Figura 12, ao final da fragmentação hibrida apresentada foram gerados 4 fragmentos. Vale lembrar que devido à aplicação da fragmentação vertical, o atributo identificador do produto ficou presente em todos os fragmentos, de forma que a relação possa ser reconstruída através de uma junção, caso alguma consulta exija a recomposição total da tabela, ou até mesmo de parte dela.

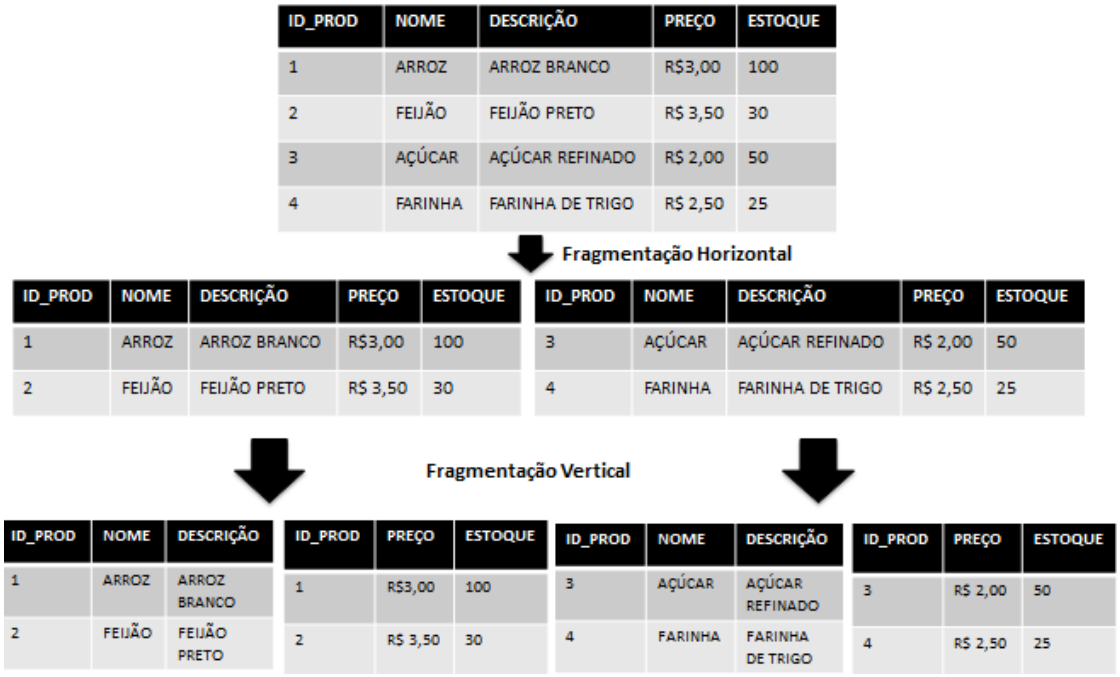


Figura 12: Exemplo de fragmentação híbrida

3.1.1.4 REGRAS DE CORREÇÃO

Tanto na fragmentação horizontal quanto na vertical foi mencionado a necessidade da aplicação das regras de corretude. Segundo Ozsu e Valduriez (2011), as regras de corretude são:

Completeza: dado uma relação *R* decomposta em fragmentos R_1, R_2,...,R_n, cada item de dado que compõe *R* pode ser encontrado em um ou mais fragmentos;

Reconstrução: caso uma relação *R* seja decomposta em fragmentos R_1, R_2,...,R_n, deve ser possível definir um operador relacional que permita a reconstrução de *R* a partir dos fragmentos;

Disjunção: se um item de dado pertence a um determinado fragmento R_1, não pode pertencer a nenhum outro (a exceção é a chave primária da tabela, na fragmentação vertical e híbrida).

3.1.2 PROJETO DE FRAGMENTAÇÃO DE DADOS ORIENTADOS A OBJETOS

Assim como para o modelo relacional, existem trabalhos na literatura (BAIÃO et al., 2004; FLORENTINO, 2003; KARLAPALEM, KAMALAKAR; LI, QING, 2000) que propõem uma forma de realizar projeto de fragmentação para base de dados orientada a objetos, a fim de minimizar o tempo de resposta das consultas. Utilizando o mesmo conceito do modelo relacional (OZSU; VALDURIEZ, 2011), o projeto de fragmentação para modelo de dados

orientados a objetos apresentam duas técnicas em sua composição (BAIÃO et al., 2004): a fragmentação vertical, onde os atributos e os métodos são distribuídos entre os fragmentos; e a fragmentação horizontal onde as instâncias das classes são distribuídas entre os fragmentos. Por ultimo, há a fragmentação hibrida que consiste na aplicação da combinação das duas técnicas apresentadas acima.

O trabalho apresentado por Baião, Mattoso e Zaverucha (2004) aborda a importância da fase de análise dentro do processo de distribuição de dados orientados a objetos e propõe heurísticas para essa etapa, que levam em consideração as seguintes informações:

- Características da aplicação, verificando as operações e sua frequência sobre os dados;
- Semântica do banco de dados, representada pelos atributos, relacionamentos e métodos das classes;
- Informações quantitativas, representadas pelos objetos e seus tamanhos estimados.

A Figura 13 apresenta um resumo da metodologia proposta por Baião, Mattoso e Zaverucha (2004) para fragmentação de banco de dados orientados a objeto.

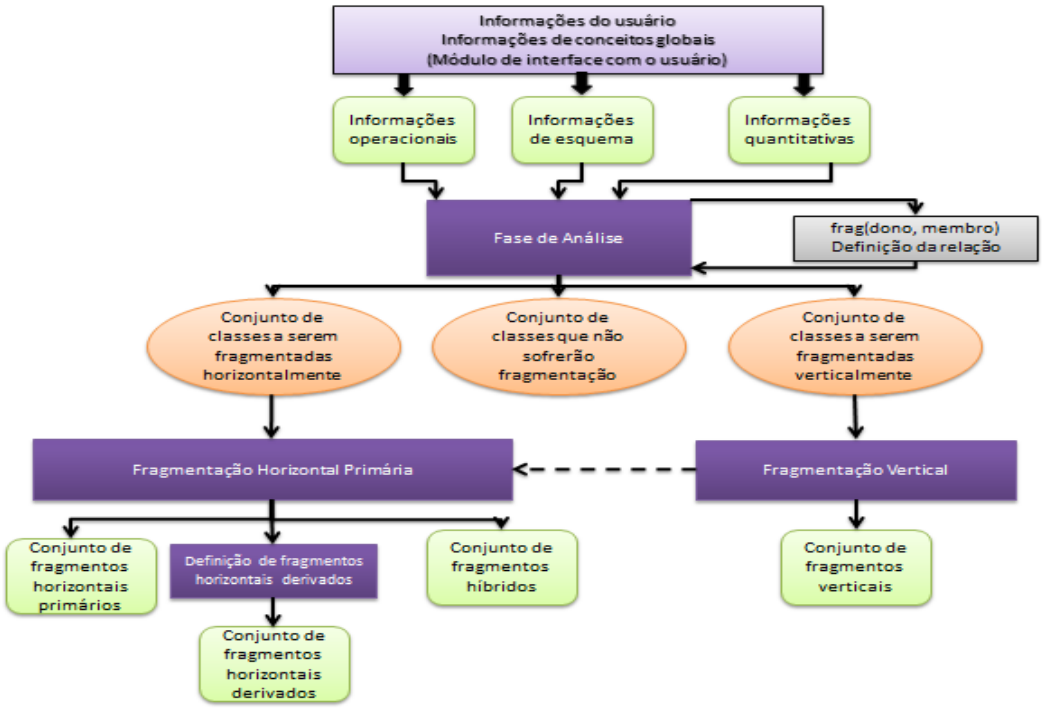


Figura 13: Fragmentação de dados (OO) - Baião, Mattoso e Zaverucha (2004) - Traduzido

Florentino (2003) apresenta uma proposta genérica de uma metodologia para projeto de fragmentação de dados baseado no trabalho apresentado por Baião et al., (2004),

permitindo que seja aplicada a três diferentes tipos de modelos de dados: orientado a objetos, objeto-relacional e relacional.

Assim como a metodologia proposta por Baião, Mattoso e Zaverucha (2004), Florentino (2003) define que os pontos relevantes ao se desenvolver um projeto de fragmentação são: etapa de extração de dados, que consiste na extração das informações consideradas relevantes no processo de definição do tipo de fragmentação mais adequada para uma determinada base; a etapa de análise, que consiste na fase que se define, a partir dos dados disponibilizados na etapa anterior, o tipo de fragmentação que melhor se aplica a cada classe; e, por último, a etapa de fragmentação propriamente dita, que segue a mesma abordagem apresentada para o modelo relacional (OZSU; VALDURIEZ, 2011). A Figura 14 apresenta de forma macro a metodologia proposta por Florentino (2003), conhecida como ODARA. O conceito de coleção apresentado na Figura 14 representa conjuntos de tabelas ou classes.

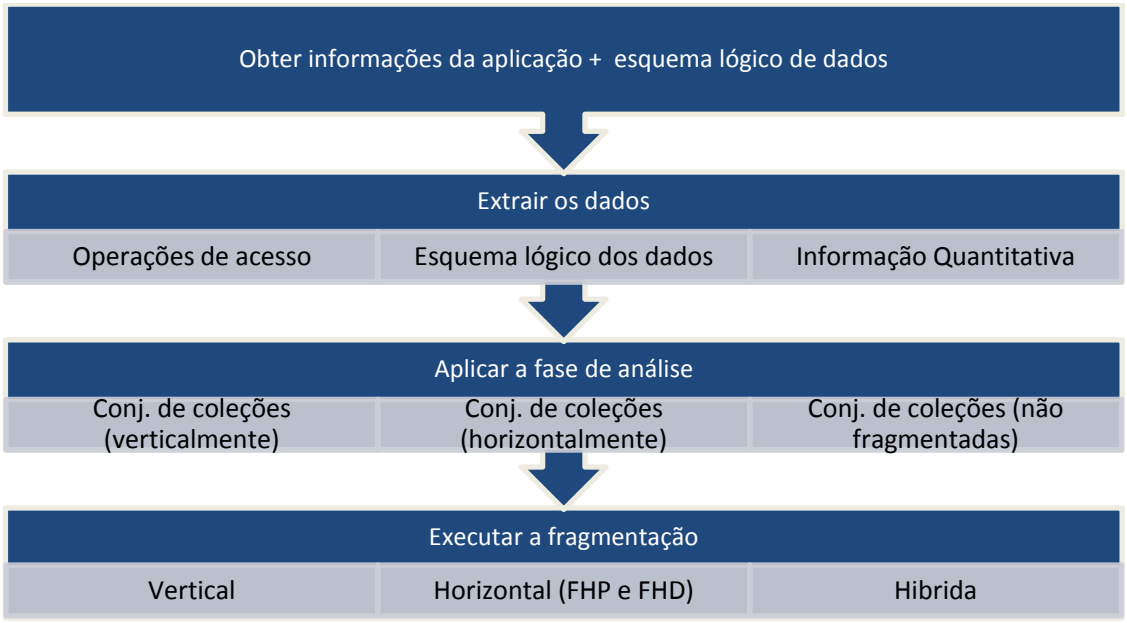


Figura 14: Fragmentação de dados orientados a objetos, objeto-relacional e relacional

A seguir, detalhamos as principais características de cada uma das etapas apresentadas na Figura 14.

Extração dos dados

Para a etapa de extração de dados, as informações da aplicação e o esquema lógico dos dados são extraídos de forma a prover informações importantes que serão utilizadas na

etapa posterior (Fase de Análise). A seguir apresentamos um detalhamento desses dados extraídos.

Informações da aplicação – consiste dos dados referentes às consultas executadas sobre as bases de dados. Após a extração dessas informações as operações presentes são identificadas, classificadas e quantificadas. As operações de acesso são classificadas em três tipos:

- **Projeção** – acessos isolados a atributos utilizados pela aplicação;
- **Seleção** – predicados lógicos aplicados sobre os atributos das coleções;
- **Navegação** – operações realizadas através de expressões de caminhos entre diferentes coleções.

Esquema lógico dos dados – contém os dados referentes ao modelo lógico da base (relacional e orientado a objeto) juntamente com as coleções (tabelas ou classes) e suas cardinalidades, e por último o grafo de dependência das coleções que apresenta os relacionamentos existentes entre as coleções na modelagem lógica.

Fase de Análise

Baseado nas operações de acesso, esquema lógico dos dados e nas informações quantitativas foram definidos as heurísticas que são utilizadas no algoritmo da análise. As heurísticas definidas por trabalhos anteriores (BAIÃO; MATTOSO, MARTA, 1998; TAVARES, 1999) são descritas a seguir e utilizadas para a composição de uma metodologia genérica atendendo o modelo relacional, orientado a objeto e objeto-relacional.

Heurística baseada em semântica – utiliza as informações extraídas do modelo lógico.

Heurística baseada em informações quantitativas - permite designar quais coleções devem ou não ser fragmentadas

- Classes com cardinalidades altas ou médias que possuem instâncias (assim como as suas extensões) devem ser consideradas no processo de fragmentação;
- Classes com baixa cardinalidade não devem ser considerados para a fragmentação vertical;
- Classes abstratas (sem instancias) não devem ser fragmentadas.

O algoritmo de análise utiliza todas as heurísticas definidas acima em sua composição. O algoritmo é composto por três funções: ***mapHierarchy***, que permite que as operações oriundas de acesso a classes abstratas sejam analisadas e mapeadas para as suas subclasses; ***analysisPhase***, onde cada operação é analisada, e de acordo com as heurísticas definidas a técnica de fragmentação mais adequada para cada coleção da base de dados é determinada; ***DefineOwnerMembe****r* que preenche a lista *FragOwnerMember* com base no esquema dos dados, nos relacionamentos Dono-Membro desse esquema e no conjunto de operações ordenadas de acordo com a sua frequência de execução**.** Os detalhes desses algoritmos estão disponíveis no ANEXO I.

Heurística baseada em operações – utiliza as informações extraídas da aplicação, após a classificação, identificação e definição da frequência das operações sobre as consultas.

- Se uma classe possui uma alta ou média cardinalidade e a operação de projeção for uma das frequentes, é indicado fazer a fragmentação vertical;
- Se a operação de seleção for uma das frequentes é indicado que seja feito FHP;
- Se a operação de navegação for uma das frequentes é indicado que seja feito FHD.

Essas heurísticas propostas complementam o que foi apresentado na seção 3.1.1, pois podem ser aplicadas ao modelo relacional conforme a proposta feita por Florentino (2003).

3.2 PROJETO DE FRAGMENTAÇÃO DE DADOS XML

Diversos trabalhos são encontrados na literatura sobre definições de fragmentos XML (ANDRADE, 2006; GERTZ; BREMER, 2003; KLING et al., 2010a; MA; SCHEWE, 2003) e projeto de fragmentação para XML em geral (BONIFATI; CUZZOCREA, 2007; BREMER; GERTZ, 2003; KLING et al., 2010a; KURITA et al., 2007; MA; SCHEWE, 2003). No entanto, especificamente para a etapa de análise do projeto de fragmentação, nenhum trabalho detalha os critérios que precisam ser levados em consideração antes de efetuar a fragmentação de base XML. Esse tipo de deficiência não permite definir um método decisório consistente quanto ao tipo de fragmentação mais aplicável, fazendo com que a fragmentação seja baseada na experiência dos projetistas.

No trabalho apresentado por Pagnamenta (2005), o objetivo principal é a avaliação de aspectos relacionados à distribuição de dados, com ênfase no processamento distribuído das transações executadas, processamento de consultas e a representação dos esquemas globais. Entretanto, dentro desses aspectos a etapa de fragmentação não é abordada. Já a abordagem para distribuição de documentos utiliza noções de fragmentação horizontal e vertical diferentes da proposta convencional do modelo relacional. Além disso, não são apresentadas no trabalho as regras de correção referentes ao modelo de fragmentação aplicado e também não são descritos os critérios que definem quando cada tipo de fragmentação deve ser aplicado. Por último, o modelo de distribuição apresentado não estabelece uma ordem lógica de execução, sendo a forma de fragmentação diretamente ligada à forma de alocação dos fragmentos.

Bremer e Gertz (2003) propõem uma arquitetura para repositórios de dados XML distribuídos e a partir daí um sistema de gerenciamento para esses repositórios. Dentro dessa arquitetura, há também um modelo de alocação dos fragmentos utilizando o conceito de estruturas indexadas. Resumidamente, o trabalho de Bremer e Gertz (2003) aborda:

1. Arquitetura *top-down* para a distribuição de dados XML, seguindo os conceitos já propostos para o modelo relacional e orientado a objetos;
2. Esquema de alocação de dados utilizando os conceitos de estruturas indexadas;
3. Proposta de um modelo de processamento de consultas sobre dados XML distribuídos.

Ma e Schewe (2003) apresentam em seu trabalho técnicas de fragmentação horizontal e vertical para XML como sendo uma generalização do modelo relacional. Segundo os autores, é importante considerar as consultas frequentes no processo de definição dos fragmentos. Além disso, para a escolha de documentos a serem fragmentados verticalmente, os autores propõem a aplicação da matriz de afinidade sobre os atributos assim como é feito para o modelo relacional (OZSU; VALDURIEZ, 2011).

Em outro trabalho, Ma e Schewe (2005) apresentam heurísticas que visam minimizar o custo de consultas para o caso de fragmentação horizontal. Essas heurísticas são baseadas no modelo de custo que leva em consideração as estruturas complexas das consultas XML. Segundo os autores, o custo de transporte é um fator decisório na aceitação ou rejeição da

fragmentação horizontal sob um predicado simples oriundo de uma das consultas frequentes. O foco do trabalho é projetar fragmentos e alocá-los de tal maneira que o desempenho global do sistema da base de dados distribuída seja melhor que o equivalente no centralizado.

Em sua abordagem (MA; SCHEWE, 2005), os tamanhos dos documentos construídos durante a execução da consulta são importantes, uma vez que esses dados devem ser armazenados em disco local e recuperados novamente para daí sim serem enviados pela rede. Sendo assim, os autores utilizam uma estimativa de tamanho que eles mesmos definiram para fragmentação de dados orientados a objetos. Entretanto, os autores não mencionam nenhum resultado experimental que comprove as heurísticas propostas, ficando apenas nas formalizações teóricas.

Já Kling, Ozsu e Daudjee (2011) propõem um projeto de distribuição para XML de forma diferenciada do modelo relacional. O foco do projeto consiste em explorar a aplicação da distribuição dos dados XML como forma de solucionar o problema da eficácia e eficiência no acesso a coleções de dados XML muito grandes. Os autores afirmam que a solução proposta por eles resolve dois problemas que se dão ao processar consultas XML em ambientes distribuídos: a localização e a "poda".

1. A localização consiste na conversão de um plano único de execução da consulta em vários de forma que possam ser executados em um ambiente distribuído;
2. A "poda" seria uma etapa posterior que eliminaria os planos de execução que não contribuiriam com o resultado da consulta, ou seja, somente as bases que contêm fragmentos relevantes à consulta seriam acessadas.

Resumidamente, o projeto proposto pelos autores (KLING et al., 2011) possui o seguinte escopo:

- Apresentar um modelo de distribuição para XML que suporta a fragmentação horizontal e vertical, utilizando o mesmo conceito aplicado ao modelo relacional, onde a fragmentação horizontal baseia-se nas operações de seleção e seus predicados e a fragmentação vertical baseia-se em atributos do esquema. Essas duas formas de fragmentação permitem a formação da fragmentação hibrida. Ao afirmar isso, Kling, Ozsu e Daudjee (2011) contradizem a afirmação realizada que diz que há

diferenças significativas entre os dados e os modelos de consultas XML e relacional, impossibilitando a aplicação das técnicas definidas para o modelo relacional;

- Resolver o problema da localização e da "poda", que segundo eles aumenta o desempenho de forma significativa;
- Baseado na técnica de localização apresentada, propor algoritmos para a etapa de fragmentação.

Para o algoritmo de fragmentação horizontal, Kling, Ozsu e Daudjee (2011) apresentam a mesma técnica aplicada para o modelo relacional. Ou seja, a fragmentação é gerada através da composição de mintermos, da mesma forma que é feita por Ozsu e Valduriez (2011) no modelo relacional. Os mintermos são obtidos através da extração de predicados contidos nas consultas frequentes. Em seguida, há uma decomposição em predicados simples de igualdade e desigualdade. Por último, é realizada a combinação desses predicados simples de forma a obter todas as combinações possíveis. A Figura 15 apresenta o exemplo do funcionamento da extração de predicados simples das consultas e a composição dos mintermos a partir desses predicados.

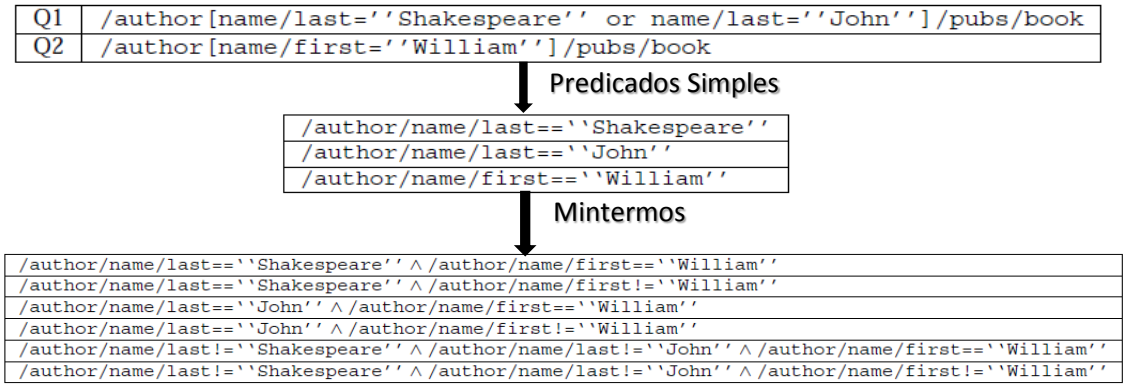


Figura 15: Exemplo de composição de mintermos (KLING; ÖZSU; DAUDJEE, 2011)

Para o algoritmo de fragmentação vertical (KLING et al., 2011) , a consulta é submetida e reformulada em subconsultas a serem executadas sobre os nós que contêm os fragmentos que atendem a essas subconsultas. Após isso, os resultados intermediários são consolidados para enfim obter o resultado completo da consulta. Para esse tipo de fragmentação, Kling, Ozsu e Daudjee (2011) alertam que dependendo da forma como a coleção está fragmentada, os resultados intermediários podem ser tão grandes que inviabilizam a execução em um ambiente distribuído. É preciso então que a fragmentação

esteja bem ajustada ao modelo de consultas executadas. No entanto, os autores não apresentam como esse processo de análise deve ser feito.

Para o projeto de fragmentação vertical, o trabalho de Birhanu, Atnafu e Getahun (2010) apresenta uma proposta sobre duas formas: baseado nas consultas e no tamanho da base de dados. Para a primeira forma é proposta a aplicação do algoritmo de energia de ligação (HOFFER; SEVERANCE, 1975) e do algoritmo gráfico (NAVATHE; RA, 1989). Nesse trabalho, o autor realiza experimentos sobre esses dois algoritmos e compara os resultados com o cenário centralizado. Os resultados apontam que ambas as formas propostas apresentam melhor resultado se comparado ao ambiente centralizado, sendo a aplicação do algoritmo gráfico a que apresentou ganho superior. Entretanto, os autores não apresentam em seu trabalho as consultas utilizadas no experimento e os resultados descritos são muito resumidos, não sendo possível avaliar como foi o comportamento em cada consulta.

Para o projeto de distribuição descrito por Kurita et al., (2007), os autores focam na fragmentação dos dados XML e na realocação desses fragmentos sobre os nós da rede de forma dinâmica. Eles consideram que a construção de um sistema de processamento de distribuição de consultas XML de larga escala deve seguir 4 passos: Fragmentação, Distribuição, Processamento Distribuído das consultas e Alocação Dinâmica. Vale lembrar que o foco do trabalho se restringe apenas à fragmentação vertical. Na fragmentação, os autores não levam em consideração as consultas frequentes e sim o tamanho do fragmento. Ou seja, na fragmentação proposta por eles, o objetivo é obter fragmentos de tamanhos homogêneos. Resumidamente, a fragmentação utiliza a seguinte ideia. Suponha uma base XML com tamanho M e que o número de fragmentos seja N. O tamanho ideal do fragmento é dado por a = M/N. Como nem sempre é possível fragmentar um documento XML com tamanhos iguais, propõe-se a margem a(1-e) <= L <= a(1+e). Na etapa de distribuição, como os fragmentos são basicamente do mesmo tamanho, é realizada a distribuição aleatória.

Já na etapa de processamento distribuído das consultas foi possível observar que a mesma é realizada de forma idêntica ao proposto por Andrade et al., (2005). Resumidamente, há um nó chamado de mediador que recebe a consulta original, submete-a aos nós que processam a consulta e consolida o resultado final. Os demais nós contêm adaptadores que acessam a base de dados local.

Por último, acontece a etapa de alocação dinâmica, que é a maior contribuição do trabalho. Kurita et al., (2007) propõem um algoritmo onde o mediador é responsável por efetuar um controle de estatísticas de execução das consultas sobre os adaptadores e a partir disso tomar a decisão quanto a possível realocação dos fragmentos entre os nós adaptadores.

Para finalizar os trabalhos relacionados, o trabalho de Bonifati e Cuzzocrea (2007) propõe uma fragmentação de dados XML para grandes volumes de dados através da estrutura do documento. Ou seja, questões como tamanho, largura e profundidade das subárvores são levadas em consideração no processo. Os autores dizem propor um conjunto de heurísticas para fragmentação de documentos XML, chamados por eles de SimpleX. Entretanto, no trabalho é apresentado apenas uma delas. Essa heurística busca obter os valores máximos para as variáveis de tamanho, largura e profundidade das subárvores. Para chegar à melhor combinação dessas variáveis, os autores fazem uso de estruturas de histogramas. Esses histogramas foram implementados dentro de um módulo de análise que utiliza algoritmos para otimizar a combinação desses valores.

3.3 CONSIDERAÇÕES FINAIS

Esse capítulo apresenta as propostas de projeto de fragmentação de dados para o modelo relacional e orientado a objetos que são utilizados como fundamentação desse trabalho. Além disso, esse capítulo descreve os trabalhos relacionados à projeto de fragmentação de dados XML. Como podemos observar nos trabalhos relacionados a fragmentação de dados XML avaliados, não foi possível identificar os critérios utilizados para efetuar as fragmentações. Em geral, os trabalhos descrevem o processo de fragmentação e alocação dos fragmentos assumindo que os projetistas já detêm o conhecimento quanto as melhores práticas para a definição dos fragmentos e suas respectivas alocações.

O capítulo seguinte apresenta os experimentos realizados nesse trabalho para obter heurísticas para fragmentação de bases de dados XML. Foram definidos os critérios de fragmentação e alocação dos fragmentos a fim de avaliar o comportamento das formas de fragmentação propostas, quando comparadas ao cenário centralizado. Além disso, é descrito o protótipo que foi utilizado e adaptado para atender os experimentos propostos.

CAPÍTULO 4: AVALIAÇÃO EXPERIMENTAL

O objetivo deste capítulo é apresentar o uso do protótipo de Mediador e Adaptadores apresentado no Capítulo 2, com o intuito de obter através de execuções de consultas sobre bases XML em ambientes distribuídos, heurísticas eficientes para a etapa de análise de projeto de fragmentação de bases de dados XML. Para isso, foram utilizados procedimentos da etapa analítica utilizada em outros modelos de dados a fim de nos beneficiarmos de conceitos já consolidados (BAIÃO et al., 2004; OZSU; VALDURIEZ, 2011).

A Seção 4.1 descreve detalhes sobre a preparação dos experimentos, tais como a descrição das bases de dados e dos servidores utilizados. Na Seção 4.2 é apresentada a metodologia de execução dos experimentos. A metodologia leva em consideração aspectos como definição dos critérios de fragmentação, geração dos fragmentos e os planos de execução do experimento. Finalmente, as considerações finais do capítulo são apresentadas na Seção 4.3.

4.1 PREPARAÇÃO DOS EXPERIMENTOS

Para derivar as heurísticas para fragmentação horizontal e vertical, realizamos uma série de experimentos visando avaliar o comportamento do tempo de resposta de consultas usando diferentes alternativas de fragmentação horizontal e vertical. A execução dos experimentos exige a execução de alguns passos: definir os objetivos dos experimentos; planejar a sua execução, assim como o ambiente onde serão executadas, as bases de dados e fragmentos que serão utilizados; definir a alocação destes fragmentos no ambiente distribuído e construir uma metodologia de execução que será utilizada para permitir uma análise conclusiva dos resultados.

4.1.1 OBJETIVOS

Antes de descrevermos o experimento em si é preciso determinar os objetivos de sua execução para garantir que os seus resultados serão relevantes para o trabalho que está sendo executado. A partir destes objetivos, pode-se elaborar um plano de execução dos experimentos, garantindo, desta forma, que todos serão cumpridos.

Os objetivos definidos para os experimentos deste trabalho são:

1. Comparação do desempenho de consultas *XQuery* sobre um ambiente centralizado com consultas sobre o ambiente distribuído em cenários onde a fragmentação leva em consideração as consultas frequentes ou não.
2. Avaliação de desempenho na execução de consultas que se beneficiam da fragmentação e para consultas que não se beneficiam da fragmentação no ambiente distribuído e fragmentado.
3. Através da avaliação experimental, obter heurísticas que permitam fragmentar qualquer base XML a fim de diminuir o tempo de resposta das consultas submetidas sobre o ambiente distribuído.

Esses objetivos fundamentaram o planejamento dos experimentos, assim como a preparação do ambiente, das bases e dos fragmentos, que são vistos na Seção 4.2.

4.1.2 AMBIENTE

Os experimentos foram executados em um cluster homogêneo composto de 42 máquinas, cada uma com dois processadores Intel Xeon *quadcore* (8 núcleos). Optamos pela arquitetura sem compartilhamento de disco e memória, porque queríamos simular a distribuição dos dados em ambientes o mais genéricos possível. De fato, esse tipo de disposição pode ser aplicado a computadores isolados e não apenas em clusters. Para esse trabalho, usamos nove nós do cluster. Cada nó possui 16GB memória de RAM e disco rígido local de 160GB. Um deles atuou como Mediador, que é responsável pela submissão das consultas, geração das subconsultas e consolidação dos resultados. Uma instância do adaptador executa em cada um dos oito nós restantes, sendo esses nós responsáveis pela execução local das subconsultas. Cada instância do adaptador utilizou o disco local do nó onde foi alocado, evitando desta forma o custo de acesso ao disco compartilhado do cluster. Os fragmentos foram armazenados nos nós num banco de dados XML nativo Sedna (FOMICHEV et al., 2006). A Figura 16 apresenta o ambiente de execução dos experimentos, onde um nó do cluster funciona como mediador e os oito demais nós atuam como adaptadores locais.

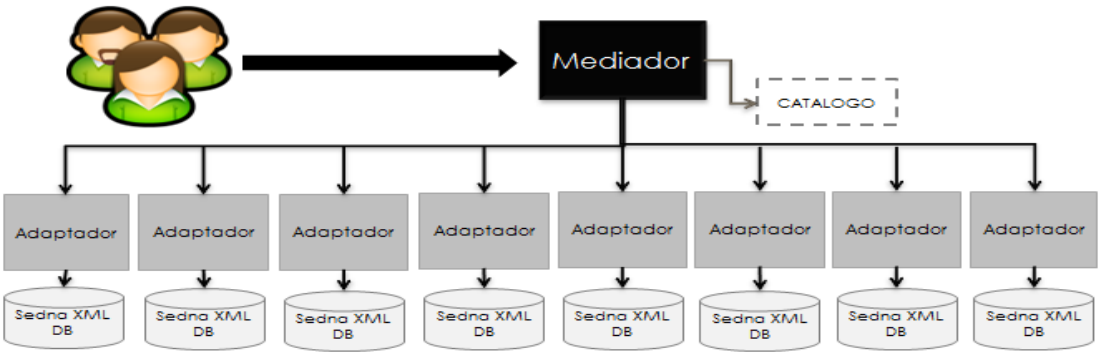


Figura 16: Ambiente para a execução do experimento

É importante destacar que a implementação do Mediador e do Adaptador utilizados nesse experimento é uma alteração da versão do protótipo construído por Figueiredo, Braganholo e Mattoso (2007). A alteração teve como objetivo aperfeiçoar o protótipo para obtenção de melhor desempenho, substituindo a forma de comunicação entre os nós e o SGBD XML Nativo utilizado na versão anterior do protótipo. Na versão original (FIGUEIREDO et al., 2007) a comunicação entre o mediador e os adaptadores era feita via WebService, o que acarretava em perda de desempenho, devido ao tempo gasto com empacotamento e desempacotamento de mensagens. Além disso, o SGBD XML Nativo utilizando era o eXist (MEIER, 2003), que, segundo experimentos realizados pelo grupo, tem baixo desempenho. A configuração da implementação original era um tanto quanto complexa que dificulta o seu funcionamento. Com a adaptação realizada, o mediador e os adaptadores se comunicam via socket e o SGBD XML Nativo utilizado foi o Sedna (FOMICHEV et al., 2006). Essa adaptação permitiu uma facilidade em seu uso além de uma melhora no desempenho.

4.2 METODOLOGIA DE EXECUÇÃO

A metodologia de execução dos experimentos nesse trabalho consiste na execução de consultas *XQuery* repetidamente, num total de 10 vezes cada, sobre as bases de dados em vários cenários que serão apresentados a seguir. Cada cenário possui um objetivo que permitirá a comparação dos resultados de forma a avaliar o desempenho das consultas em diferentes ambientes e configurações. A base de dados utilizada nos experimentos foi obtida através de benchmarks conhecidos para XML. As consultas XQuery disponibilizadas pelos benchmarks foram consideradas as consultas frequentes de uma aplicação, uma vez que os benchmarks não fornecem a frequência de execução de cada consulta.

Para a construção das heurísticas foi definida uma sequência de etapas que são apresentadas na Figura 17.

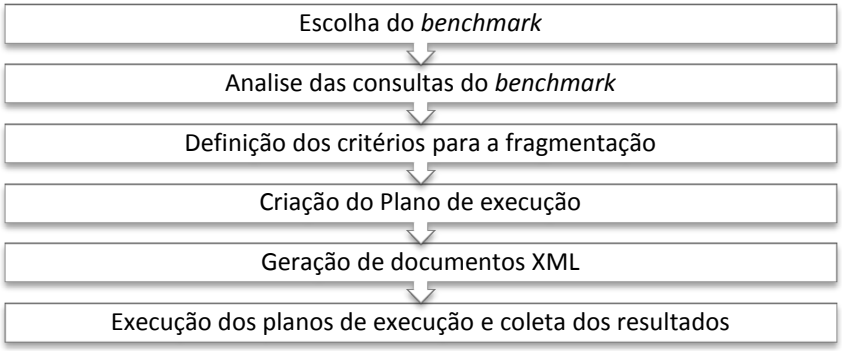


Figura 17: Etapas do processo de definição das heurísticas para fase de análise

As subseções 4.2.1 e 4.2.2 apresentam cada uma das etapas listadas na Figura 17 de acordo com o tipo de base e consequentemente o tipo de fragmentação aplicada nessa base. A subseção 4.2.1 descreve as etapas da metodologia para a base MD de múltiplos documentos (Fragmentação Horizontal), enquanto a subseção 4.2.2 apresenta a metodologia de execução para a base SD de um único documento (Fragmentação Vertical).

4.2.1 FRAGMENTAÇÃO HORIZONTAL

A subseção 4.2.1 visa apresentar toda a metodologia de execução para a base de múltiplos documentos a fim de avaliarmos no experimento proposto os resultados obtidos através da fragmentação horizontal.

4.2.1.1 ESCOLHA DO BENCHMARK

Para analisarmos a fragmentação horizontal, foi escolhido o *benchmark XBench* (YAO et al., 2004) que é bem amplo, possuindo diversos modelos e consultas aos seus dados. Em nosso trabalho foi utilizado um dos quatro tipos de base de dados do *XBench*: *Multiple Document* (DC/MD)[1]. A base DC/MD possui 5 esquemas de documentos XML: Customer (SD), Item (SD), Author (SD), Address (SD), Country (SD) e Orders (MD). As bases do tipo DC simulam aplicações centradas em dados, ou seja, normalmente elas possuem grandes volumes de dados. Além disso, a base de múltiplos documentos foi escolhida devido à restrição de aplicação de fragmentação horizontal em bases de único documento. Portanto, para esse tipo de fragmentação foi utilizado o esquema Orders com variações de 2.592, 25.920 e 259.200 documentos, dependendo do fator de tamanho da base de dados

[1] Bases de dados de múltiplos documentos onde os dados são oriundos de informações transacionais.

(*database size*). Para a geração dessas bases de dados foi utilizado o ToxGene (BARBOSA et al., 2002), um gerador automático de bases de dados XML. A Figura 18 descreve o esquema da base de múltiplos documentos utilizada.

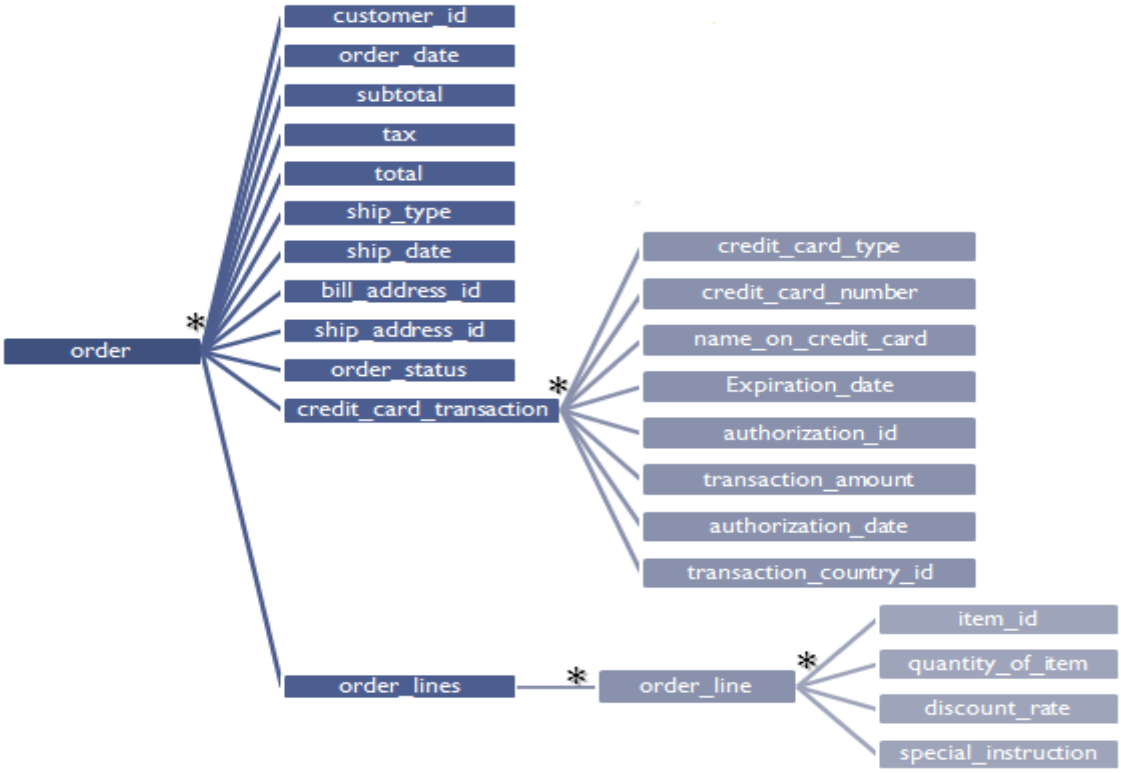


Figura 18: Esquema dos documentos Order (YAO et al., 2004)

4.2.1.2 ANÁLISES DAS CONSULTAS DO BENCHMARK (MD)

Dentre as consultas disponibilizadas no *benchmark XBench*, 19 delas foram consideradas como sendo as consultas mais frequentes de uma aplicação executada sobre as bases de dados. Estas por sua vez foram adaptadas para atender às restrições de consultas permitidas pelo mediador. Ou seja, o mediador só atende consultas que respeitam um certo conjunto de regras de formação gramaticais (FIGUEIREDO, 2007).

Após a escolha da base de dados, as consultas adaptadas foram analisadas para que pudessem ser definidos os critérios de fragmentação, de forma que as consultas fossem beneficiadas pelo processo de fragmentação. Como o enfoque nessa parte do trabalho é avaliar o comportamento da fragmentação horizontal, a Tabela 1 apresenta um resumo dos predicados de seleção presentes em cada uma das 19 consultas utilizadas no experimento de fragmentação horizontal. Vale ressaltar que as consultas C1, C5, C7, C10 e C11 contêm funções agregadoras em seus predicados de seleção e também foram analisados em nosso

experimento. Essas informações são fundamentais para a definição dos fragmentos. As consultas completas estão disponíveis no ANEXO II dessa dissertação.

Tabela 1: Atributos de seleção e projeção das consultas (MD)

Consultas	Predicados de seleção/Predicado de projeção
C1	*count*(/order/order_lines/order_line) >= 5
C2	/ order/@id = "1"
C3	/order/@id = "3"
C4	/order/@id = "5"
C5	*count*(/order/order_lines/order_line) = 1
C6	/order/@id = "6"
C7	/order/total > 7000 and *count*(/order/order_lines/order_line) >= 5
C8	/ order/total > 7000
C9	/ order/total > 7000
C10	/order/total < 2000 and *count*(/order/order_lines/order_line) >= 5
C11	order/total > 11000 and *count*(/order/order_lines/order_line) >= 5
C12	/ order/@id="1"
C13	/order/total > 11000
C14	/order/@id="2"
C15	/order/total > 11000
C16	/order/total > 11000
C17	/order/total > 10000
C18	/order/total > 10000
C19	/order/total > 7000 and /order/total < 8000

4.2.1.3 DEFINIÇÃO DOS CRITÉRIOS PARA A FRAGMENTAÇÃO

A Tabela 2 apresenta um resumo dos predicados de seleção presentes em cada uma das consultas utilizadas no experimento, com o número de ocorrências de cada predicado. Como podemos observar, existe uma frequência maior de consultas que utilizam o atributo order/total. Foi assumida a mesma frequência de execução para todas as consultas. No entanto, quando um mesmo predicado simples aparecia em mais de uma consulta, isto foi refletido no seu número de ocorrências, como pode ser visto na Tabela 2.

Tabela 2: Análise dos predicados de seleção e suas respectivas ocorrências

Predicado Simples	Número de Ocorrências
order/total > 11000	4 vezes
order/total > 10000	2 vezes
order/total > 7000	4 vezes
order/total < 2000	1 vez

order/total < 8000	1 vez
order/@id = “1”	2 vez
order/@id = “2”	1 vez
order/@id = “3”	1 vez
order/@id = “5”	1 vez
order/@id = “6”	1 vez

Após analisar a Tabela 2, foram definidos os critérios para a fragmentação a fim de avaliar o comportamento das execuções e derivar as heurísticas para fragmentação horizontal. Os critérios levados em consideração nos cenários descritos a seguir foram: atributo frequente, domínio do atributo frequente, tamanho da base, predicado de seleção simples mais frequente e quantidade de nós disponíveis para alocação dos fragmentos. Resumidamente, como será apresentado na seção a seguir, os critérios avaliados em cada um dos cenários está dividido da seguinte forma:

Cenário 1: Avalia o uso do predicado de seleção simples mais frequente;

Cenário 2: Utiliza o atributo frequente, domínio do atributo frequente e quantidade de nós disponíveis para alocação dos fragmentos;

Cenário 3: Avalia o resultado da fragmentação ao utilizar um atributo que não aparece em nenhuma das consultas frequentes. As questões de domínio desse atributo e a quantidade de nós disponíveis também são utilizadas nesse cenário.

Já o critério de tamanho da base foi avaliado sobre todos os cenários listados, pois o experimento foi executado sobre bases de 4 MB, 40 MB e 400 MB.·.

4.2.1.4 PLANO DE EXECUÇÃO PARA MÚLTIPLOS DOCUMENTOS

Para o plano de execução sobre o *XBench* (MD) foram definidos 3 cenários baseados nas listas de consultas disponibilizadas que foram considerados no experimento como sendo as consultas frequentes. É importante ressaltar que para cada cenário há a submissão da consulta sobre ambiente centralizado (um único servidor) e também sobre ambientes distribuídos com 2, 4, 6 e 8 servidores. Outro critério que também foi incluso em nosso experimento foi o tamanho da base de dados, permitindo assim avaliar o comportamento do tempo de resposta à medida que o tamanho da base de dados cresce. Por isso, foi inserido na análise bases com 4MB, 40MB e 400MB. Após analisar a Tabela 2, foram definidos 11 subcenários de avaliação, agrupados em três cenários. A definição dos fragmentos utilizados em cada subcenário é descrita na Tabela 3.

Cenário 0: Execução das consultas em ambiente centralizado.

Cenário 1: Execução das consultas utilizando fragmentação horizontal com base nos atributos das consultas frequentes. Como podemos ver na Tabela 2, o atributo total é o que possui maior ocorrência. Outro ponto a ser observado é que o predicado de seleção total > 11000 ocorre em 4 das 19 consultas executadas. Sendo assim, o cenário visa avaliar o comportamento das consultas se fragmentarmos baseado nesse predicado de seleção. Foram definidos dois subcenários:

(1.1.1) três fragmentos distribuídos em dois nós;

(1.1.2) três fragmentos distribuídos em três nós.

Cenário 2: Execução das consultas utilizando fragmentação horizontal, utilizando como base o número total de nós disponíveis para alocação e o domínio dos dados. Avaliando o domínio do atributo de seleção mais frequente (total), verificamos que seu valor varia entre 0 a 15000. Sendo assim, o cenário 2 visa fragmentar a partir do atributo total variando o número de nós disponíveis e levando em conta o domínio de seus valores. Foram definidos quatro subcenários:

(2.1.1) dois fragmentos em dois nós;

(2.1.2) quatro fragmentos em quatro nós;

(2.1.3) seis fragmentos em seis nós;

(2.1.4) oito fragmentos em oito nós.

Cenário 3: Execução das consultas utilizando fragmentação horizontal definida de forma a não utilizar os atributos das consultas frequentes. Se observarmos a Figura 18, podemos verificar que existe um atributo chamado *transaction_country_id* que não aparece em nenhum dos predicados de seleção listados na Tabela 2. Nesse cenário, usamos critérios semelhantes aos do cenário 2 para fragmentarmos, ou seja, analisamos o domínio do campo *transaction_country_id* (1 a 92) e variamos o numero de nós disponíveis. Foram definidos quatro subcenários classificados da seguinte forma:

(3.1.1) dois fragmentos em dois nós;

(3.1.2) quatro fragmentos em quatro nós;

(3.1.3) seis fragmentos em seis nós;

(3.1.4) oito fragmentos em oito nós.

Tabela 3: Resumo do plano de execução do XBench (MD)

Resumo do plano de execução do *XBench*(MD)				
Cenário	***Tipo de Fragmentação***	***Critério de Avaliação***	***Critério de fragmentação***	***Alocação***
0	Não se aplica	Centralizado	Não se aplica	Nó 1
1.1.1	Horizontal	Frequência das Consultas	Frag 1: $<C_{Orders}, \sigma_{order/total >= 11000}>$ Frag 2 : $<C_{Orders}, \sigma_{order/total <= 7000}>$ Frag 3: $<C_{Orders}, \sigma_{order/total > 7000 \wedge order/total < 11000}>$	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 1
1.1.2	Horizontal	Frequência das Consultas e Alocação Distribuída	Frag 1: $<C_{Orders}, \sigma_{order/total >= 11000}>$ Frag 2 : $<C_{Orders}, \sigma_{order/total <= 7000}>$ Frag 3: $<C_{Orders}, \sigma_{order/total > 7000 \wedge order/total < 11000}>$	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3
2.1.1	Horizontal	Alocação e Domínio dos Dados	Frag 1: $<C_{Orders}, \sigma_{total <=1000}>$ Frag 2: $<C_{Orders}, \sigma_{total >1000}>$	Frag 1: Nó 1 Frag 2: Nó 2
2.1.2	Horizontal	Alocação e Domínio dos Dados	Frag 1: $<C_{Orders}, \sigma_{total <=1000}>$ Frag 2: $<C_{Orders}, \sigma_{order/total > 1000 \wedge order/total <= 7000}>$ Frag 3: $<C_{Orders}, \sigma_{order/total > 7000 \wedge order/total <=11000}>$ Frag 4: $<C_{Orders}, \sigma_{order/total > 11000}>$	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3 Frag 4: Nó 4
2.1.3	Horizontal	Alocação e Domínio dos Dados	Frag 1: $<C_{Orders}, \sigma_{total <=250}>$ Frag 2: $<C_{Orders}, \sigma_{order/total > 250 \wedge order/total <= 500}>$ Frag 3: $<C_{Orders}, \sigma_{order/total > 500 \wedge order/total <= 1000}>$ Frag 4: $<C_{Orders}, \sigma_{order/total > 1000 \wedge order/total <= 7000}>$ Frag 5: $<C_{Orders}, \sigma_{order/total > 7000 \wedge order/total <=11000}>$ Frag 6: $<C_{Orders}, \sigma_{order/total > 11000}>$	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3 Frag 4: Nó 4 Frag 5: Nó 5 Frag 6: Nó 6
2.1.4	Horizontal	Alocação e Domínio dos Dados	Frag 1: $<C_{Orders}, \sigma_{total <=250}>$ Frag 2: $<C_{Orders}, \sigma_{order/total > 250 \wedge order/total <= 500}>$ Frag 3: $<C_{Orders}, \sigma_{order/total > 500 \wedge order/total <= 1000}>$ Frag 4: $<C_{Orders}, \sigma_{order/total > 1000 \wedge order/total <= 5000}>$ Frag 5: $<C_{Orders}, \sigma_{order/total > 5000 \wedge order/total <= 7000}>$ Frag 6: $<C_{Orders}, \sigma_{order/total > 7000 \wedge order/total <= 9000}>$ Frag 7: $<C_{Orders}, \sigma_{order/total > 9000 \wedge order/total <= 11000}>$ Frag 8: $<C_{Orders}, \sigma_{order/total > 11000}>$	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3 Frag 4: Nó 4 Frag 5: Nó 5 Frag 6: Nó 6 Frag 7: Nó 7 Frag 8: Nó 8
3.1.1	Horizontal	Sem uso das consultas frequentes	Frag 1 : $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=23}>$ Frag 2 : $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id > 23 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=46}>$ Frag 3 : $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id > 46 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=69}>$ Frag 4 : $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id>69}>$	Frag 1: Nó 1 Frag 2: Nó 1 Frag 3: Nó 2 Frag 4: Nó 2
3.1.2	Horizontal	Sem uso das consultas frequentes Alocação Distribuída	Frag 1 : $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=23}>$ Frag 2 : $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id > 23 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=46}>$ Frag 3 : $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id > 46 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=69}>$ Frag 4: $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id>69}>$	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3 Frag 4: Nó 3
3.1.3	Horizontal	Alocação aleatória dos dados Alocação Distribuída	Frag 1: $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=12}>$ Frag 2: $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id > 12 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=23}>$ Frag 3: $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id > 23 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=46}>$ Frag 4: $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id > 46 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=69}>$ Frag 5: $<C_{Orders}, \sigma_{order/credit\_card\_transaction/transaction\_country\_id > 69 \wedge$	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3 Frag 4: Nó 4 Frag 5: Nó 5 Frag 6: Nó 6

			$\sigma_{order/credit\_card\_transaction/transaction\_country\_id<=81}$> Frag 6: <$C_{Orders}$, $\sigma_{order/credit\_card\_transaction/transaction\_country\_id>81}$>	
3.1.4	Horizontal	Alocação aleatória dos dados Alocação Distribuída	Frag 1: <C_{Orders}, $\sigma_{order/credit\_card\_transaction/transaction\_country\_id<=12}$> Frag 2: <$C_{Orders}$, $\sigma_{order/credit\_card\_transaction/transaction\_country\_id > 12 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=23}$> Frag 3: <$C_{Orders}$, $\sigma_{order/credit\_card\_transaction/transaction\_country\_id > 23 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=39}$> Frag 4: <$C_{Orders}$, $\sigma_{order/credit\_card\_transaction/transaction\_country\_id > 39 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=46}$> Frag 5: <$C_{Orders}$, $\sigma_{order/credit\_card\_transaction/transaction\_country\_id > 46 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=58}$> Frag 6: <$C_{Orders}$, $\sigma_{order/credit\_card\_transaction/transaction\_country\_id > 58 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=69}$> Frag 7: <$C_{Orders}$, $\sigma_{order/credit\_card\_transaction/transaction\_country\_id > 69 \wedge \sigma_{order/credit\_card\_transaction/transaction\_country\_id<=81}$> Frag 8: <$C_{Orders}$, $\sigma_{order/credit\_card\_transaction/transaction\_country\_id>81}$>	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3 Frag 4: Nó 4 Frag 5: Nó 5 Frag 6: Nó 6 Frag 7: Nó 7 Frag 8: Nó 8

4.2.1.5 GERAÇÃO DE DOCUMENTOS XML E FRAGMENTAÇÃO DAS BASES

Para a geração dos documentos da base de múltiplos documentos foi utilizado o ToXgene (BARBOSA et al., 2002), conhecido gerador de dados XML. Conforme descrito anteriormente, os experimentos foram executados sobre bases de 4MB, 40MB e 400MB. Para a base de dados de múltiplos documentos, temos:

1. 4MB: 2592 documentos de 2 KB cada
2. 40MB: 25920 documentos de 2 KB cada
3. 400MB: 259200 documentos de 2 KB cada

Como a massa de dados foi gerada de forma aleatória pelo ToXgene (BARBOSA et al., 2002), a alocação dos fragmentos em cada um dos cenários acima ficou distribuída conforme as Tabela 4, Tabela 5 e Tabela 6, respectivamente. Essa informação é importante para avaliarmos o comportamento da fragmentação a partir da quantidade de documentos alocados em cada nó.

Tabela 4: Número de documentos alocados em cada nó – XBench (4MB)

Cenário	Nó 1	Nó 2	Nó 3	Nó 4	Nó 5	Nó 6	Nó 7	Nó 8
Centralizado	2592							
1.1.1	437	2155						
1.1.2	214	2155	223					
2.1.1	1961	631						
2.1.2	1961	194	223	214				
2.1.3	300	336	925	194	223	214		
2.1.4	300	336	925	139	55	133	90	214
3.1.1	1322	1270						
3.1.2	651	671	632	638				
3.1.3	316	335	671	632	358	280		
3.1.4	316	335	473	198	323	309	358	280

Tabela 5: Número de documentos alocados em cada nó – XBench (40MB)

Cenário	Nó 1	Nó 2	Nó 3	Nó 4	Nó 5	Nó 6	Nó 7	Nó 8
Centralizado	25920							
1.1.1	13828	12092						
1.1.2	6918	12092	6910					
2.1.1	1689	24231						
2.1.2	1689	10403	6910	6918				
2.1.3	428	419	843	10403	6910	6918		
2.1.4	427	419	843	6923	3480	3377	3533	6918
3.1.1	13071	12849						
3.1.2	6539	6532	6464	6385				
3.1.3	3322	3217	6532	6464	3417	2968		
3.1.4	3322	3217	4587	1945	3397	3067	3417	2968

Tabela 6: Número de documentos alocados em cada nó – XBench (400MB)

Cenário	Nó 1	Nó 2	Nó 3	Nó 4	Nó 5	Nó 6	Nó 7	Nó 8
Centralizado	259200							
1.1.1	138252	120948						
1.1.2	69236	120948	69016					
2.1.1	17006	242194						
2.1.2	17006	103942	69016	69236				
2.1.3	4057	4318	8631	103942	69016	69236		
2.1.4	4057	4318	8631	69061	34881	34565	34451	69236
3.1.1	129829	129371						
3.1.2	64074	65755	65499	63872				
3.2.3	32772	31302	65755	65499	34290	29582		
3.2.4	32772	31302	45843	19912	34141	31358	34290	29582

4.2.1.6 EXECUÇÃO DOS PLANOS E COLETA DOS RESULTADOS

Essa etapa consiste na execução dos planos definidos no passo anterior e na coleta dos seus resultados para obtenção das heurísticas. O mediador, após a execução das consultas em cada um dos cenários, carrega os tempos em um arquivo texto que posteriormente é utilizado na análise dos resultados. Através dos resultados obtidos com a execução dos planos de cada cenário, são levantados os critérios que realmente fizeram diferença para melhorar o desempenho das execuções das consultas sobre bases XML distribuídas e fragmentadas se comparado ao cenário centralizado. O detalhamento da análise dos resultados é descrito no Capítulo 5.

4.2.2 FRAGMENTAÇÃO VERTICAL

Conforme descrito na Figura 17, a metodologia proposta descreve sete etapas no processo de execução dos experimentos sugeridos nessa dissertação. Como o enfoque é recomendar heurísticas para fragmentação de dados XML, essa seção apresenta o que foi desenhado para avaliar o comportamento dos resultados no cenário de fragmentação

vertical. As subseções subsequentes detalham como foi realizado todo o processo para efetuarmos a fragmentação vertical e seus respectivos resultados.

4.2.2.1 ESCOLHA DO BENCHMARK

Para efetuar análises das consultas sobre bases SD executadas sobre o protótipo foi escolhido um benchmark conceituado para XML, conhecido como XMark (BUSSE et al., 2002). O XMark possui em seu modelo de dados a representação de um site de leilões cuja base contém um único documento XML, chamado "auction.xml". A Figura 19 apresenta o esquema do documento do benchmark XMark e as consultas disponibilizadas para este benchmark estão disponíveis no ANEXO III. Vale ressaltar que essa base será utilizada no processo de fragmentação vertical porque contém um único documento.

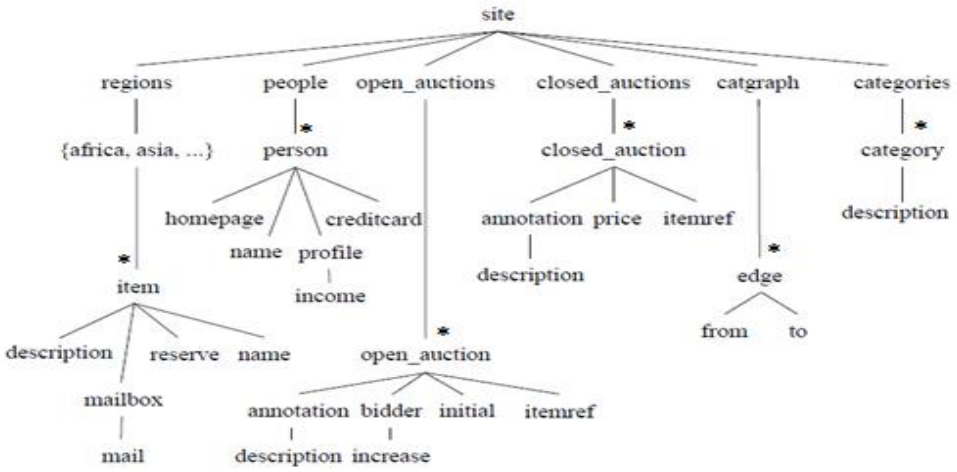


Figura 19: Representação do esquema do modelo de dados do XMark (BUSSE et al., 2002)

4.2.2.2 ANÁLISES DAS CONSULTAS DO BENCHMARK (SD)

Para executar os experimentos foram extraídas 14 consultas do próprio *benchmark XMark* (BUSSE et al., 2002) e também do experimento realizado em (KLING et al., 2010b). Essas consultas sofreram adaptações por conta de restrições gramaticais do protótipo que não permitem a execução de consultas que utilizam certas funções, tais como *contains(), empty(), distinct-values(), exactly-one()*, etc.

A Tabela 7 apresenta um resumo dos predicados de seleção e projeção pertencentes às 14 consultas executadas nos experimentos sobre bases de um único documento. Para avaliar o comportamento dos tempos de resposta das consultas, variamos o fator de escala gerando bases de 3 tamanhos diferentes (10MB, 100MB e 1GB). Essas informações são

fundamentais para a execução da etapa posterior (definição dos critérios para a fragmentação), pois são ela que definem como as bases devem ser fragmentadas. As consultas utilizadas e adaptadas nesse trabalho estão disponíveis no ANEXO III.

Tabela 7: Atributos de seleção e projeção das consultas (SD)

Consultas	Predicados de seleção/Predicado de projeção
C1	/site/people/person/@id = "person0" /site/people/person/name
C2	/site/open_auctions/open_auction/bidder/increase
C3	/site/closed_auctions/closed_auction/price>=40
C4	/site/regions/africa
C5	/site/regions/australia/item/name /site/regions/australia/item/description
C6	/site/closed_auctions/closed_auction/annotation/description
C7	/site/closed_auctions/closed_auction/seller
C8	/site/people/person/homepage
C9	/site/regions/asia/item/name
C10	/site/people/person/profile/@income = "100000" /site/people/person
C11	/site/closed_auctions/closed_auction/price > 600 /site/closed_auctions/closed_auction/price
C12	/site/regions/namerica/ item/mailbox/mail
C13	/site/open_auctions/open_auction/annotation/description /site/closed_auctions/closed_auction/annotation/description /site/categories/category/annotation
C14	/site/closed_auctions/closed_auction/buyer /site/people/person/name

4.2.2.3 DEFINIÇÃO DOS CRITÉRIOS DE FRAGMENTAÇÃO (SD)

Como mencionado anteriormente, para a definição dos critérios de fragmentação inicialmente foram utilizadas as recomendações que hoje são aplicadas no projeto de fragmentação para o modelo relacional (OZSU; VALDURIEZ, 2011) e orientado a objetos (BAIÃO et al., 2004). Sendo assim, foram analisados os atributos de projeção e predicados de seleção que mais ocorrências tinham nas consultas.

Para a base de um único documento há a possibilidade de efetuarmos uma fragmentação vertical e hibrida (vertical seguida da horizontal). Entretanto, avaliamos as consultas para que pudéssemos efetuar apenas fragmentações verticais. É importante ressaltar que para as projeções temos uma avaliação chamada de “Agrupamento” cujo objetivo é avaliar se um determinado atributo aparece agrupado a outro(s) no momento da consulta. Esse critério de avaliação é utilizado por Ozsu e Valduriez (2011) com o objetivo de maximizar o agrupamento dos atributos que normalmente ocorrem em uma mesma consulta. A Tabela 8 apresenta as ocorrências dos atributos de seleção e projeção que aparecem em cada uma das 14 consultas utilizadas nesse trabalho. Vale ressaltar que no

caso de um único documento consideramos a ocorrência de um dado atributo dentro da subárvore. Ou seja, a Tabela 8 mostra que, por exemplo, na consulta C1 houve a ocorrência de um ou mais atributos pertencentes à subárvore /site/people.

Tabela 8: Ocorrências dos atributos dentro das consultas utilizadas

Consultas X Atributos	regions	people	open_auctions	closed_auctions	catgraph	categories
C1	-	X	-	-	-	-
C2	-	-	X	-	-	-
C3	-	-	-	X	-	-
C4	X	-	-	-	-	-
C5	X	-	-	-	-	-
C6	-	-	-	X	-	-
C7	-	-	-	X	-	-
C8	-	X	-	-	-	-
C9	X	-	-	-	-	-
C10	-	X	-	-	-	-
C11	-	-	-	X	-	-
C12	X	-	-	-	-	-
C13	X	-	X	X	-	X
C14	-	X	-	X	-	-

Os critérios levados em consideração nos cenários descritos a seguir foram: distribuição total dos fragmentos, ou seja, não há mais de uma subárvore em um mesmo fragmento; tamanho dos fragmentos, número de nós disponíveis, agrupamento dos atributos com afinidades nas consultas frequentes.

4.2.2.4 PLANO DE EXECUÇÃO PARA UM ÚNICO DOCUMENTO

Para o plano de execução *XMark* (SD) foram definidos 3 cenários baseados nas listas de consultas disponibilizadas que serão considerados no experimento como sendo as consultas frequentes. É importante ressaltar que para cada cenário há a submissão da consulta sobre ambiente centralizado (um único servidor) e também sobre ambientes distribuídos com variações de numero de servidores entre 2 a 6. Outro critério que também foi incluso em nosso experimento foi o tamanho da base de dados, permitindo assim avaliar o comportamento do tempo de resposta à medida que o tamanho da base de dados cresce. Por isso, foi inserido na análise bases com 10MB, 100MB e 1GB. Como último critério de análise, aplicamos o algoritmo de fragmentação vertical proposto por Ozsu e Valduriez (2011), onde é gerada uma matriz de uso dos atributos sobre as consultas frequentes. Após isso, é calculada uma matriz de afinidade agrupada entre os atributos e por último a partição das subárvores, gerando por fim os fragmentos.

Antes de detalharmos cada um dos cenários propostos, há um ponto importante referente à definição de fragmento vertical definido por Andrade et al., (2006), que diz que a

fragmentação não pode ser aplicada sobre subárvores de cardinalidades maiores que 1. Por exemplo, no esquema apresentado na Figura 19, se for criado um fragmento onde somente exista o caminho **/category/description**, teríamos um documento sem raiz. Podemos considerar que cada caminho selecionado para o fragmento gere um novo documento, e assim, possua uma raiz única com um identificador. Além disso, neste caso, não seria possível garantir a reconstrução do documento na ordem correta. Então, esta restrição assegura que a fragmentação irá resultar em documentos bem-formados, sem a necessidade de gerar elementos artificiais para reorganizar as subárvores projetadas no fragmento.

Tendo em vista definição dada anteriormente, há algumas restrições no processo de fragmentação dos dados. Ou seja, para o caso do esquema do documento auction.xml apresentado na Figura 19, as fragmentações verticais realizadas obedeceram aos seguintes critérios:

1. A fragmentação vertical será feita sobre as subárvores superiores (**regions, people, open_auctions, closed_auctions, catgraph, categories**) garantindo a cardinalidade da raiz do fragmento igual a 1;
2. Apenas um dos fragmentos conterá em sua composição o nó raiz (site). Os demais fragmentos usam subelementos de site como sendo o nó raiz. Esse ponto traz consigo a seguinte restrição: o fragmento que não contiver o nó raiz (site) não pode ter mais de uma subárvore superior. Se isso ocorrer, o novo documento XML gerado pelo fragmento será mal formado.

A partir dos pontos apresentados anteriormente, os cenários propostos são apresentados a seguir. A Tabela 19 apresenta um resumo do plano de execução do *XMark* (SD).

Cenário 0: Execução em ambiente centralizado

Esse cenário consiste no armazenamento do documento original (auction.xml) em apenas um nó, sem que ele sofra qualquer tipo de fragmentação. O objetivo desse cenário é nos permitir comparar os tempos de respostas com os cenários fragmentados propostos.

Cenário 1: Distribuição total dos fragmentos

Esse cenário visa avaliar os tempos de resposta das consultas visto que dentre as 14 consultas, 12 delas acessam apenas uma das subárvores do documento. Com essa distribuição total, a consulta seria submetida a um único fragmento, não havendo

necessidade de junções com outros fragmentos, exceto as consultas 13 e 14 que fazem junções entre os dados de diferentes subárvores. É claro que nesse caso de distribuição total foi levado em consideração o número total de nós disponíveis, que nesse caso são efetivamente seis nós. Vale ressaltar que a alocação desses fragmentos foi feita sem replicação dos dados. Sendo assim, como a estrutura de dados do documento auction.xml possui seis subárvores, foram gerados 6 fragmentos conforme a Tabela 9.

Tabela 9: Distribuição Total dos Fragmentos (Cenário 1)

Fragmento	Descrição do Fragmento
Fragmento 1:	$<C_{auction}$, π/site, {{/site/people}, {/site/open_auctions}, {/site/closed_auctions}, {/site/catgraph}, {/site/categories}}>
Fragmento 2:	$<C_{auction}$, π/site/people>
Fragmento 3:	$<C_{auction}$, π/site/open_auctions>
Fragmento 4:	$<C_{auction}$, π/site/ closed_auctions>
Fragmento 5:	$<C_{auction}$, π/site/catgraph>
Fragmento 6:	$<C_{auction}$, π/site/categories>

Cenário 2: Distribuição dos fragmentos por tamanho e número de nós disponíveis

Esse cenário nos permite avaliar o comportamento dos resultados ao variarmos os tamanhos de fragmentos e o numero de nós utilizados para a alocação desses fragmentos. Na Tabela 10 podemos verificar o tamanho de cada subárvore do documento em cada tamanho de base.

Tabela 10: Subárvores de dados base SD (10MB, 100MB e 1GB)

Tamanho da Base de dados	regions	people	open_auctions	closed_auctions	catgraph	categories
10MB	6.0M	1.4M	3.3M	1.8M	4.5K	173K
100MB	60M	14M	33M	18M	47K	1.5M
1GB	598M	135M	327M	177M	487K	15M

Cenário 2.1: Distribuição em dois fragmentos

Ao analisarmos as consultas para esse cenário específico, podemos perceber que apenas quatro delas (C4, C5, C9 e C12) acessam apenas a subárvore /site/regions, nove delas (C1, C2, C3, C6, C7, C8, C10, C11 e C14) acessam alguma outra e apenas uma (C13) delas acessa /site/regions e alguma outra subárvore. Baseado nisso, nesse cenário optou-se por fragmentar a base em dois fragmentos: um que contém a subárvore /site/regions, e outro

que contém as demais subárvores. A Tabela 11 apresenta a distribuição dos fragmentos e seus respectivos tamanhos para o cenário 2.1.

Tabela 11: Distribuição dos Fragmentos (Cenário 2.1)

Fragmento	Descrição do Fragmento	Base 10MB	Base de 100MB	Base de 1GB
Fragmento 1:	<$C_{auction}$, π/site/ regions >	6.0 M	60 M	598 M
Fragmento 2:	<$C_{auction}$,π/site, {/site/regions}>	6.6 M	66 M	655 M

Cenário 2.2: Distribuição em três fragmentos

Para o cenário 2.2, há a divisão do documento em três fragmentos, conforme a Tabela 12. Entretanto, podemos observar que o fragmento 1 tem quase o dobro do tamanho dos demais. Para diminuir o tamanho desse fragmento seria necessário particionar a subárvore /site/regions em subárvores menores. Contudo, por definição da fragmentação vertical (ANDRADE et al., 2006), esse tipo de fragmentação dentro de uma mesma subárvore não é permitido devido à cardinalidade dos subelementos dessa subárvore. Por isso, mantivemos toda a subárvore /site/regions no mesmo fragmento. Essa observação vale também para as demais subárvores. Os demais fragmentos foram gerados levando em consideração o tamanho do fragmento gerado e nesse caso os fragmentos 2 e 3 possuem tamanhos relativamente semelhantes.

Ao analisarmos as consultas para esse cenário, é possível concluir que assim como no cenário 2.1, quatro consultas (C4, C5, C9 e C12) acessam apenas a subárvore /site/regions (Fragmento 1), oito delas (C1, C3, C6, C7, C8, C10, C11 e C14) acessam outra pertencente ao Fragmento 2, a consulta (C2) acessa o fragmento 1 e, por último, temos C13 acessando os 3 fragmentos.

A diferença entre o cenário 2.1 e 2.2 é que para o 2.2 as consultas (C1, C2, C3, C6, Q7, C8, C10, C11 e C14) acessam um fragmento com quase a metade do tamanho do fragmento 2 do cenário 2.1. Entretanto, a consulta Q13 acessa os três fragmentos no cenário 2.2.

Tabela 12: Distribuição dos Fragmentos (Cenário 2.2)

Fragmento	Descrição do Fragmento	Base 10MB	Base de 100MB	Base de 1GB
Fragmento 1:	<$C_{auction}$, π/site/ regions >	6.0 M	60 M	598 M
Fragmento 2:	<$C_{auction}$,π/site, {{/site/regions}, {/site/open_auctions}}>	3.4 M	34 M	327 M
Fragmento 3:	<$C_{auction}$, π/site/open_auctions>	3.3 M	33 M	327 M

Cenário 2.3: Distribuição em quatro fragmentos

Para esse cenário, o documento foi dividido em 4 fragmentos, conforme descrito na Tabela 13. A diferença entre esse cenário e o cenário 2.2 é a redução do tamanho do fragmento 2 para dois novos fragmentos menores. As consultas C3, C6, C7 e C11 acessam exclusivamente o fragmento 3 que é bem menor que o fragmento 2 do cenário 2.2. Entretanto, a consulta C13 acessa os quatro fragmentos no cenário 2.3.

Tabela 13: Distribuição dos Fragmentos (Cenário 2.3)

Fragmento	Descrição do Fragmento	Base 10MB	Base de 100MB	Base de 1GB
Fragmento 1:	$<C_{auction}$, π/site/ regions >	6.0 M	60 M	598 M
Fragmento 2:	$<C_{auction}$,π/site, {{/site/regions}, {/site/open_auctions}, {/site/closed_auctions}}>	1.5 M	15.5 M	150 M
Fragmento 3:	$<C_{auction}$, π/site/closed_auctions>	1.8M	18M	177 M
Fragmento 4:	$<C_{auction}$, π/site/open_auctions>	3.3 M	33 M	327 M

Cenário 3: Agrupamento

A ideia desse cenário é analisar o tempo de resposta nos casos onde são realizados agrupamentos parciais e por afinidades entre as subárvores do documento, avaliando as suas ocorrências nas consultas frequentes.

Cenário 3.1: Agrupamento parcial

Para a proposta de agrupamento parcial, apenas duas consultas do experimento fazem uso de mais de uma subárvore do documento, e iremos fazer o agrupamento parcial sobre essas subárvores. A ideia nesse cenário é avaliar se essas consultas se beneficiaram dessa fragmentação e o impacto que esse tipo de fragmentação traz para as demais consultas. A Tabela 14 mostra como ficou a definição dos fragmentos. Um ponto importante a ser avaliado nesse cenário é que a subárvore /site/categories não pertence ao fragmento 1. Isso se deu pelo fato dessa subárvore só ser acessada em uma única consulta e pelo menos uma das demais subárvores ocorrerem nas consultas executadas no experimento. Nesse cenário não estamos levando em consideração o tamanho dos fragmentos, pois essa variável está sendo analisada no cenário 2.

Tabela 14: Agrupamento Parcial (Cenário 3)

Fragmento	Descrição do Fragmento
Fragmento 1:	<$C_{auction}$, π/site,{{/site/categories}, {/site/catgraph}}>
Fragmento 2:	<$C_{auction}$, π/site/categories >
Fragmento 3:	<$C_{auction}$, π/site/catgraph >

Cenário 3.2: Agrupamento por afinidade

Conforme dito anteriormente, a proposta do agrupamento por afinidade é fundamentada pelo seu uso no processo de fragmentação vertical de dados no modelo relacional (OZSU; VALDURIEZ, 2011). Para a obtenção desse agrupamento por afinidade existem algumas etapas, conforme descrito no Capitulo 3 dessa dissertação, que precisam ser executadas. Primeiramente, é gerada uma matriz de uso onde uma célula *M[i,j]* tem o valor 1 caso a consulta *i* utilize o atributo *j,* e 0 (zero) caso contrário. A Tabela 15 apresenta a matriz de uso utilizando como entrada as 14 consultas frequentes e os atributos pertencentes às subárvores superiores do documento XML (auction.xml).

Tabela 15: Matriz de Uso Gerada para o Experimento

Consultas/Atributos	regions	people	open_auctions	closed_auctions	catgraph	categories
C1	0	1	0	0	0	0
C2	0	0	1	0	0	0
C3	0	0	0	1	0	0
C4	1	0	0	0	0	0
C5	1	0	0	0	0	0
C6	0	0	0	1	0	0
C7	0	0	0	1	0	0
C8	0	1	0	0	0	0
C9	1	0	0	0	0	0
C10	0	1	0	0	0	0
C11	0	0	0	1	0	0
C12	1	0	0	0	0	0
C13	1	0	1	1	0	1
C14	0	1	0	1	0	0

A partir da matriz de uso e de frequência de acesso (matriz contendo o peso de cada consulta para as aplicações do usuário) é possível obter a matriz de afinidade entre os atributos, que mostra o peso das relações entre os atributos nas consultas avaliadas. Na nossa análise experimental, decidimos eliminar a influência do peso das consultas, e por isso foi atribuído um peso de 10 para cada uma das 14 consultas avaliadas.

O algoritmo de geração dessa matriz de afinidade foi implementado em Java, e tomou como base o algoritmo apresentado em Ozsu e Valduriez (2011). A Tabela 16

apresenta o resultado obtido para a matriz de afinidade entre os atributos. Por exemplo, a afinidade da subárvore regions com as subárvores open_auctions, closed_auctions e categories é igual a 5.

Tabela 16: Matriz de Afinidade entre os Atributos

Afinidade	regions	people	open_auctions	closed_auctions	catgraph	categories
regions	50	0	10	10	0	10
people	0	40	0	10	0	0
open_auctions	10	0	20	10	0	10
closed_auctions	10	10	10	60	0	10
catgraph	0	0	0	0	0	0
categories	10	0	10	10	0	10

O objetivo dessa forma de realizar a fragmentação vertical é encontrar um meio de efetuar agrupamentos a partir dos valores da matriz de afinidade de atributos. No Capitulo 3 foi sugerido o uso do algoritmo de energia de ligação para esse propósito. Sendo assim, foi implementado em Java um algoritmo que permite a geração da chamada matriz de afinidade agrupada, conforme descrito na Tabela 17. O objetivo da matriz de afinidade agrupada é agrupar os atributos em clusters reunindo os que têm maiores valores de afinidade em um cluster e aqueles que têm menores valores de afinidade em outro cluster.

Tabela 17: Matriz de Afinidade Agrupada

Afinidade Agrupada	**catgraph**	**categories**	**open_auctions**	**regions**	**closed_auctions**	**people**
catgraph	0	0	0	0	0	0
categories	0	10	10	10	10	0
open_auctions	0	10	20	10	10	0
regions	0	10	10	50	10	0
closed_auctions	0	10	10	10	60	10
people	0	0	0	0	10	40

Por último, a matriz de afinidade agrupada é particionada de forma a separar atributos que possuem alta afinidade em fragmentos diferentes. Dessa forma, o fragmento 1 é composto pelas subárvores *catgraph, categories, open_auctions* e *regions* enquanto que o fragmento 2 contém as subárvores *closed_auctions* e *people*. Por conta das restrições descritas anteriormente, o fragmento 2 possui apenas a subárvore *closed_auctions* e criamos um terceiro fragmento para receber a subárvore *people*, conforme descrito na Tabela 18.

Tabela 18: Agrupamento por afinidade

Fragmento	Descrição do Fragmento
Fragmento 1:	<$C_{auction}$, π/site,{{/site/ people}, {/site/closed_auctions }}>
Fragmento 2:	<$C_{auction}$, π/site/ closed_auctions >
Fragmento 3:	<$C_{auction}$, π/site/ people >

Para concluir a etapa do plano de execução a Tabela 19 apresenta um resumo de todos os cenários discutidos anteriormente e que serão utilizados no experimento da fragmentação vertical. Na Tabela 19 descrevemos cada um desses cenários e seus critérios de avaliação, assim como a alocação desses fragmentos dentro do grupo de 6 nós.

Tabela 19: Resumo do plano de execução do XMark (SD)

Resumo do plano de execução do *XMark* (SD)				
Cenário	***Tipo de Fragmentação***	***Critério de Avaliação***	***Critério de fragmentação***	***Alocação***
0	Não se aplica	Centralizado	Não se aplica.	Nó 1
1.1	Vertical	Distribuição Total	Frag 1: <$C_{auction}$, π/site,{ {/site/people}, {/site/open_auctions}, {/site/closed_auctions}, {/site/catgraph}, {/site/categories}}> Frag 2 : <$C_{auction}$, π/site/people> Frag 3 : <$C_{auction}$, π/site/open_auctions > Frag 4 : <$C_{auction}$, π/site/ closed_auctions > Frag 5 : <$C_{auction}$, π/site/catgraph > Frag 6 : <$C_{auction}$, π/site/categories >	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3 Frag 4: Nó 4 Frag 5: Nó 5 Frag 6: Nó 6
2.1	Vertical	Alocação e Tamanho da Base	Frag 1: <$C_{auction}$, π/site/ regions > Frag 2: <$C_{auction}$, π/site,{/site/regions}>	Frag 1: Nó 1 Frag 2: Nó 2
2.2	Vertical	Alocação e Tamanho da Base	Frag 1: <$C_{auction}$, π/site/ regions > Frag 2: <$C_{auction}$, π/site,{ {/site/regions}, {/site/open_auctions}}> Frag 3: <$C_{auction}$, π/site/open_auctions>	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3
2.3	Vertical	Alocação e Tamanho da Base	Frag 1: <$C_{auction}$, π/site/ regions > Frag 2: <$C_{auction}$, π/site,{ {/site/regions}, {/site/open_auctions}, {/site/closed_auctions}}> Frag 3: <$C_{auction}$, π/site/closed_auctions> Frag 4: <$C_{auction}$, π/site/open_auctions>	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3 Frag 4: Nó 4
3.1	Vertical	Agrupamento Parcial	Frag 1: <$C_{auction}$, π/site,{ {/site/ categories}, {/site/catgraph }}> Frag 2: <$C_{auction}$, π/site/categories > Frag 3: <$C_{auction}$, π/site/catgraph >	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3
3.2	Vertical	Agrupamento por afinidade	Frag 1: <$C_{auction}$, π/site,{ {/site/ people}, {/site/ closed_auctions }}> Frag 2: <$C_{auction}$, π/site/closed_auctions > Frag 3: <$C_{auction}$, π/site/people >	Frag 1: Nó 1 Frag 2: Nó 2 Frag 3: Nó 3

4.2.2.5 GERAÇÃO DE DOCUMENTOS XML E FRAGMENTAÇÃO DAS BASES

Os documentos para os experimentos que envolvem a base de único documento foram gerados com o auxílio do site XQBench (FISCHER et al., 2012). Como o objetivo é avaliar o comportamento sobre vários tamanhos de bases, foram escolhidos documentos

(auction.xml) com tamanhos de 10MB, 100MB e 1GB. Para automatizar o processo de geração dos fragmentos foi desenvolvida em Java uma aplicação que permite efetuar essa fragmentação a partir das regras de fragmentação definidas para cada um dos cenários apresentados na Tabela 19.

Assim como na fragmentação horizontal, a próxima etapa consiste na execução dos planos definidos na etapa “Plano de execução” a fim de obter um conjunto de resultados que serão utilizados na definição das heurísticas para fragmentação vertical.

4.3 CONSIDERAÇÕES FINAIS

Esse capítulo apresentou a metodologia que foi utilizada nesse trabalho para derivar as conclusões obtidas com as fragmentações propostas tanto na base de múltiplos documentos (fragmentação horizontal) quanto para a base de único documento (fragmentação vertical).

Outro ponto importante a ser ressaltado é que, teoricamente, os ganhos de desempenho com a fragmentação tendem a ser maiores nas consultas que se beneficiam da fragmentação realizada, ou seja, as consultas que não se beneficiam da fragmentação podem ter o seu desempenho reduzido devido ao custo da composição do resultado final a partir dos resultados de cada nó.

O Capítulo 5 apresenta os resultados das avaliações que foram obtidas com a metodologia proposta nesse capítulo e as heurísticas propostas baseadas nesses resultados.

CAPÍTULO 5: HEURISTÍCAS PARA FASE DE ANÁLISE

O objetivo desse capítulo é apresentar a análise dos resultados obtidos com a execução dos experimentos para fragmentação horizontal e vertical a fim de derivar heurísticas para fragmentação de dados XML. Nesse capítulo também são descritas as heurísticas obtidas com base no que foi apresentado nesse trabalho e os resultados dos experimentos executados. Vale ressaltar as heurísticas apresentadas nesse capítulo atendem aplicações que usam a linguagem de consulta *XQuery*.

A Seção 5.1 apresenta os resultados das análises para as fragmentações horizontais e verticais. A partir dos resultados dessas análises a Seção 5.2 descreve as heurísticas propostas nesse trabalho. Por último na Seção 5.3 são apresentadas as considerações finais desse capítulo.

5.1 ANÁLISE DOS RESULTADOS

Conforme dito no capítulo anterior, os experimentos foram executados em um cluster, onde os diversos nós foram utilizados como servidores distribuídos totalmente dedicados para os testes, conforme Figura 16. Os resultados são analisados nesta seção, a partir dos dados coletados na execução dos experimentos.

A comparação dos tempos médios de resposta das consultas nos diferentes cenários foi feita por meio de um gráfico que apresenta os tempos de execução das mesmas consultas em relação ao cenário centralizado (Cenário 0). O objetivo desses experimentos é verificar se a fragmentação de dados permite melhores resultados se comparada com o cenário centralizado. Em vários momentos ao logo desse capitulo, utilizamos a palavra “benefício” para denotar esse comportamento. Assumimos que houve benefício quando o tempo médio de resposta de uma consulta no ambiente distribuído foi inferior ao tempo obtido no ambiente centralizado.

A análise dos resultados foi feita a partir da comparação dos tempos totais médios de resposta das consultas entre os diferentes cenários. Como o *benchmark* não fornecia a frequência de execução das consultas optou-se for efetuar a execução de 10 vezes cada

consulta e para o cálculo do tempo médio de resposta foi desconsiderado o tempo total referente à primeira execução. Todas as execuções dos experimentos são gravadas em um arquivo texto contendo as informações descritas na Tabela 20.

Tabela 20: Informações obtidas nos experimentos

Campo	Descrição do campo
Nome da consulta	Nome da consulta que foi executada
Cenário executado	Número atribuído a cada cenário
Número da rodada	Número da rodada de uma determinada consulta
Resultado da consulta	Informa se a consulta foi executada com sucesso ou erro
Tempo de *parser* do mediador	Corresponde ao tempo que o mediador leva para validar a consulta e gerar possíveis subconsultas
Tempo de compilação do mediador	Tempo que o mediador leva para consolidar, processar e exibir os resultados das subconsultas
Tempo de comunicação	Maior tempo de comunicação entre o mediador e o adaptador na execução de uma determinada subconsulta
Tempo de resposta do adaptador	Maior tempo de resposta dentre os adaptadores envolvidos na execução de uma determinada consulta
Tempo de resposta total	Consiste no somatório de todos os outros tempos a fim de informar o tempo de resposta final da consulta

Para analisarmos os resultados para cada tipo de fragmentação, as subseções sequentes foram estruturadas da seguinte forma. Na subseção 5.1.1 são apresentados os resultados obtidos com a fragmentação horizontal em cada um dos três experimentos realizados: 4MB, 40MB e 400MB. Já os resultados da fragmentação vertical são apresentados na subseção 5.1.2 com experimentos de 10MB, 100MB e 1GB. Vale ressaltar que as análises feitas sobre os resultados dos experimentos consistem na fundamentação para as heurísticas propostas nessa dissertação.

5.1.1 ANÁLISE DE RESULTADOS DA FRAGMENTAÇÃO HORIZONTAL

A comparação dos tempos médios de resposta das consultas nos diferentes cenários foi feita através de gráficos que apresentam os tempos médios de respostas em cada cenário versus o tempo médio de resposta no cenário centralizado.

A subseção 5.1.1.1 descreve os resultados obtidos com a fragmentação horizontal sobre uma base de dados de múltiplos documentos de tamanho igual a 4MB. O detalhamento dos resultados do experimento para base de 40MB é apresentado na subseção 5.1.1.2. Já a subseção 5.1.1.3 mostra os resultados do experimento sobre a base de 400MB. Por último, as comparações entre os três experimentos são realizados na subseção 5.1.1.4.

5.1.1.1 BASE DE 4MB

O objetivo dessa subseção é apresentar os resultados obtidos ao executarmos 19 consultas sobre 11 cenários em base de dados contendo 2592 documentos XML totalizando um tamanho da base de 4MB. No ANEXO IV é apresentado um resumo dos tempos médios de resposta (em milissegundos) obtidos com a execução dos experimentos nos diversos cenários sobre a base de 4MB.

Ao analisarmos o comportamento da fragmentação da base de múltiplos documentos sobre a base de 4MB observamos que dentre as dezenove consultas executadas, oito (C1, C2, C3, C4, C5, C6, C12, C14) delas obtiveram melhor desempenho no cenário centralizado, conforme descrito na Figura 20. Todas essas consultas não possuem em seu predicado de seleção o atributo de seleção que foi utilizado como critério de fragmentação. Desta forma, ao executar a consulta foi necessário que a mesma fosse executada em todos os fragmentos visto que não era possível identificar em qual fragmento estaria a informação. Para essas consultas, havia cenários onde o maior fragmento era tão grande quanto a base do cenário centralizado. Em outros cenários, embora possuíssem um fragmento menor que a base do cenário centralizado, era necessário acessar um número expressivo de nós.

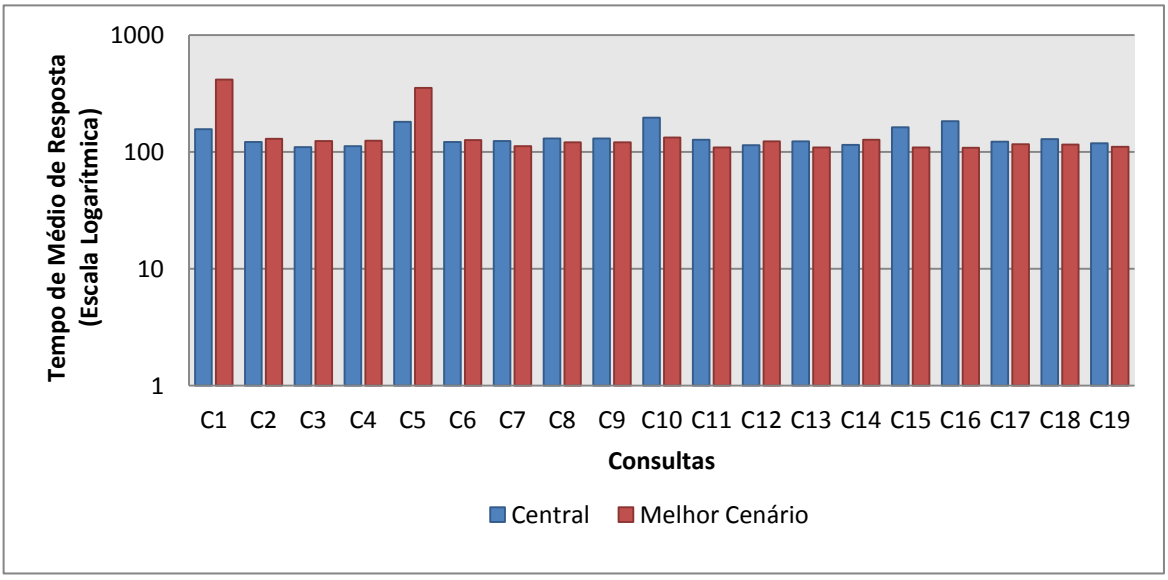


Figura 20: Tempo médio de resposta de 4MB (Centralizado versus Melhor Cenário)

Apresentamos na Figura 21, o gráfico de comparação entre o tempo de execução das consultas sobre o cenário centralizado e o cenário 1. Como podemos perceber, o cenário 1.1.1 foi melhor que o centralizado em 11 consultas (C7, C8, C9, C10, C11, C13, C15, C16, C17, C18 e C19). Vale lembrar que todas essas consultas possuíam em seu predicado de

seleção o atributo “total” utilizado como critério de fragmentação para o cenário 1.1.1. Outro fato importante é que em todas essas consultas, um único fragmento foi acessado. Os ganhos percentuais nesse cenário (1.1.1) variaram entre 3% a 57%.

Além do cenário 1.1.1, podemos observar na Figura 21 o cenário 1.1.2 onde se vê que as consultas C9, C10, C11, C13, C15, C16 e C19 se beneficiaram da fragmentação se comparados ao centralizado. Para essas consultas e nesse cenário, apenas um único fragmento foi acessado. Consequentemente, foi acessado um fragmento com tamanho menor que a base inteira e, além disso, por acessar um único nó, poupou-se tempo do mediador na composição dos resultados obtidos. Já para as consultas C7, C8, C17 e C18, que não se beneficiaram da fragmentação proposta nesse cenário, observou-se que foi necessário acessar mais de um fragmento. Vale lembrar que um dos fragmentos do cenário 1.1.2 possui 2155 documentos, tamanho esse muito próximo ao da base inteira (2592 documentos). Os ganhos percentuais no cenário 1.1.2 variaram entre 2% a 65%.

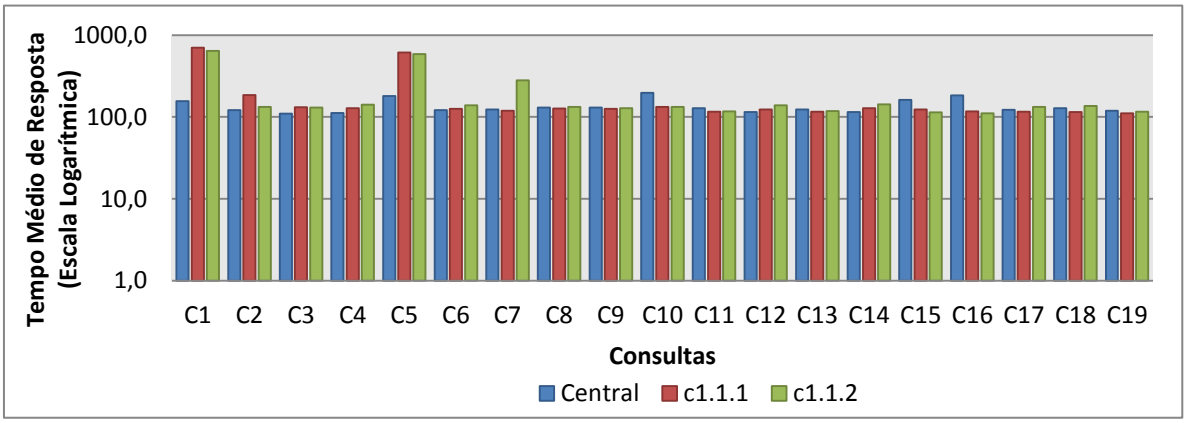


Figura 21: Tempo médio de resposta sobre a base de 4MB (Cenário 1)

O gráfico comparativo entre o cenário centralizado e o cenário 2 é descrito na Figura 22. O cenário 2.1.1 apresentou um melhor tempo que o centralizado em 10 consultas (C7, C8, C9, C11, C13, C15, C16, C17, C18 e C19), com ganhos de tempo médio de resposta variando de 3% a 61%. Assim como os demais cenários apresentados anteriormente, essas 10 consultas acessaram apenas um fragmento e pelos mesmos motivos já descritos se beneficiaram da fragmentação. A consulta C10 não se beneficiou da fragmentação visto que foi necessário efetuar acessos a mais de um fragmento. Lembrando que nesse cenário só tínhamos dois fragmentos, ou seja, foi necessário acessar um fragmento que possuía 1961 documentos.

No cenário 2.1.2, nove consultas (C8, C9, C11, C13, C15, C16, C17, C18 e C19) beneficiaram-se da fragmentação proposta se compararmos com o cenário centralizado. As consultas C11, C13, C15, C16 e C19 acessam um único fragmento de tamanho máximo 223 documentos. Já as consultas C8, C9, C17 e C18, embora acessem dois fragmentos (223 e 214 documentos) apresentaram um tempo médio de resposta local no pior nó inferior ao centralizado. As consultas C7 e C10 não se beneficiaram da fragmentação proposta no cenário 2.1.2 se comparado ao centralizado. A consulta C7 acessa nesse cenário 2 fragmentos e por conta de duas funções agregadoras tanto no predicado de seleção quanto no retorno da consulta, o tempo de processamento local no pior nó foi maior do que no centralizado. Além do tempo médio de resposta local, o mediador ao receber os retornos dos dois nós precisou realizar uma consolidação e compor o resultado. Já a consulta C10 acessou 2 fragmentos onde um deles possuía um tamanho de 1961 documentos. Essa consulta possui a mesma estrutura da consulta C7 onde há duas funções agregadoras em sua composição, e, portanto teve comportamento análogo.

As mesmas nove consultas que se beneficiaram da fragmentação no cenário 2.1.2 também obtiveram um tempo médio de resposta melhor nos cenários 2.1.3 e 2.1.4 se comparado com o centralizado. A justificativa para as consultas que não se beneficiaram da fragmentação é a mesma do cenário 2.1.2.

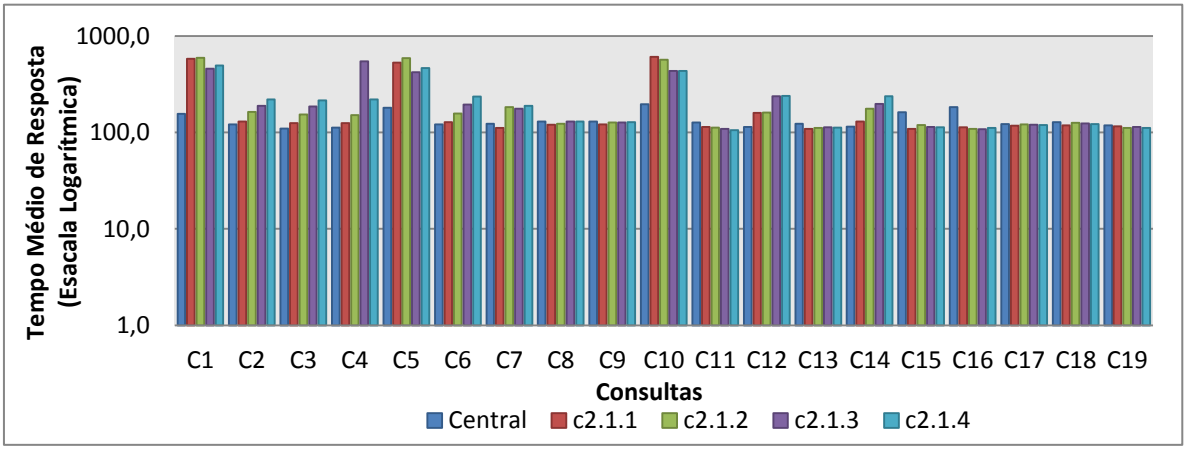


Figura 22: Tempo médio de resposta sobre a base de 4MB (Cenário 2)

Por último, a Figura 23 descreve o comportamento dos tempos de resposta do cenário 3, que, como dito anteriormente, visa analisar os tempos de execução de uma fragmentação realizada utilizando um predicado de seleção que não aparece em nenhuma das consultas frequentes. Ao avaliarmos os dados da seção de ANEXO IV podemos perceber

que o tempo médio de resposta das consultas aumentou à medida que fomos aumentando o número de fragmentos acessados. Isso nos mostra que embora os fragmentos fossem menores, o tempo médio de resposta das consultas foi pior que no ambiente centralizado. Apenas nos cenários 3.1.1 e 3.1.2 tivemos algumas consultas (C8, C9, C15, C16 e C18 no cenário 3.1.1 e C15 e C16 no cenário 3.1.2) com o seu tempo melhor que o centralizado.

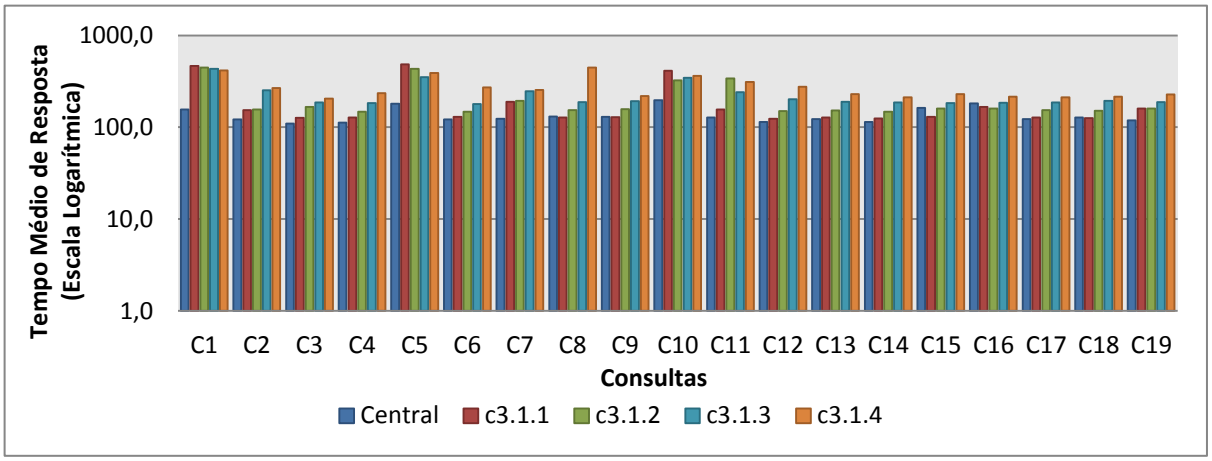


Figura 23: Tempo médio de resposta sobre a base de 4MB (Cenário 3)

5.1.1.2 BASE DE 40MB

O objetivo dessa subseção é apresentar os resultados obtidos ao executarmos 19 consultas sobre 11 cenários em base de dados contendo 25920 documentos XML totalizando um tamanho da base de 40MB. No ANEXO IV apresenta um resumo com todos os tempos médios de respostas (em milissegundos) das 19 consultas sobre os 11 cenários.

Para a base de 40MB foi realizado o mesmo experimento realizado para a base 4MB sobre o mesmo ambiente e com as mesmas consultas. Dentre as 19 consultas, apenas 8 delas não se beneficiaram de nenhuma das opções propostas (Cenário 1, Cenário 2 e Cenário 3), da mesma forma que no experimento de 4MB. A Figura 24 descreve o resultado do tempo médio de resposta das consultas executadas na base de 40MB no cenário centralizado e o melhor tempo dentre os demais cenários. Como podemos observar para as consultas C1 e C5, os tempos de execução foram melhores no ambiente centralizado do que no fragmentado. Nessas duas consultas havia cálculos de agregação que não utilizam os atributos de seleção nos predicados. Ou seja, houve um tempo alto na comunicação remota entre o mediador e os nós (adaptadores) uma vez que as consultas tiveram que ser submetidas a todos os nós. Além do tempo remoto, tivemos um tempo alto na consolidação desses resultados pelo mediador, pois ele é o responsável por calcular a agregação a partir

dos resultados das subconsultas. Podemos perceber na Figura 24 que a consulta C10 se beneficiou de forma expressiva da fragmentação. Isso ocorre pelo fato do tempo de processamento local no fragmentado ter sido menor que no centralizado visto que em dois cenários (1.1.1 e 1.1.2) foram acessados apenas um fragmento com a metade do número de documentos.

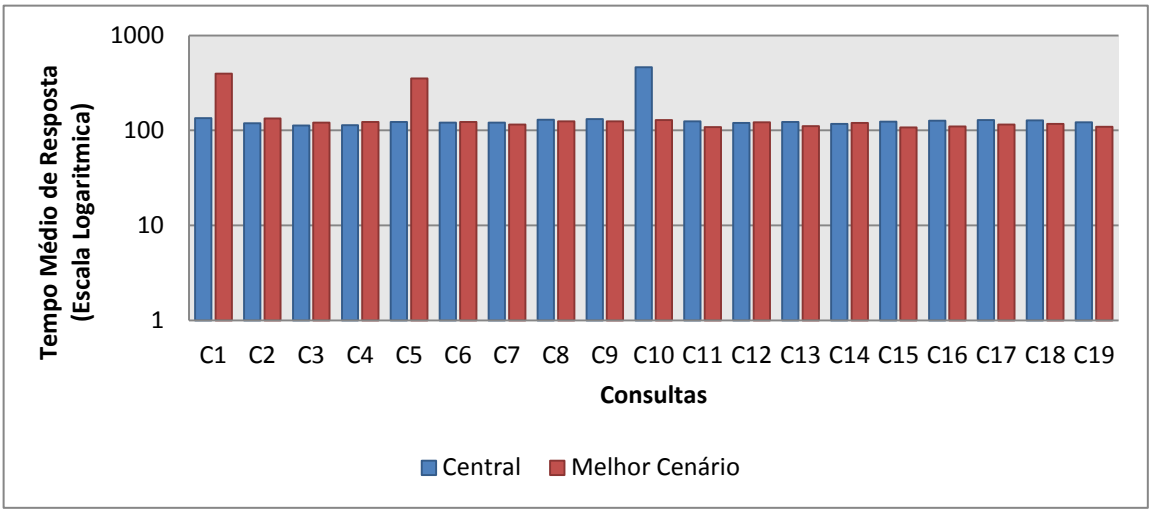


Figura 24: Tempo médio de resposta de 40MB (Centralizado versus Melhor Cenário)

Assim como apresentamos para o experimento de 4MB, a Figura 25 descreve os tempos médios de resposta da consulta no primeiro cenário. Como podemos ver, 11 consultas tiveram seus tempos médios de resposta das consultas inferiores ao centralizado no cenário 1.1.1, com ganhos percentuais variando de 2% a 260%. Todas as 11 consultas que se beneficiaram da fragmentação proposta possuíam em seu atributo de seleção o atributo “total” utilizado no critério de fragmentação do cenário 1.1.1.

Para o cenário 1.1.2 também descrito na Figura 25 os resultados foram idênticos aos do cenário 1.1.1. Ou seja, as 11 consultas listadas anteriormente também se beneficiaram da fragmentação proposta se comparados ao centralizado. Até em termos de ganhos percentuais os resultados foram parecidos, com valores variando entre 2% a 260%. Era esperado que o cenário 1.1.2 apresentasse um resultado melhor do que o cenário 1.1.1, visto que haviam fragmentos menores. Entretanto, foi possível avaliar que essa diferença de tamanho do fragmento não foi significante a ponto de influenciar os resultados obtidos.

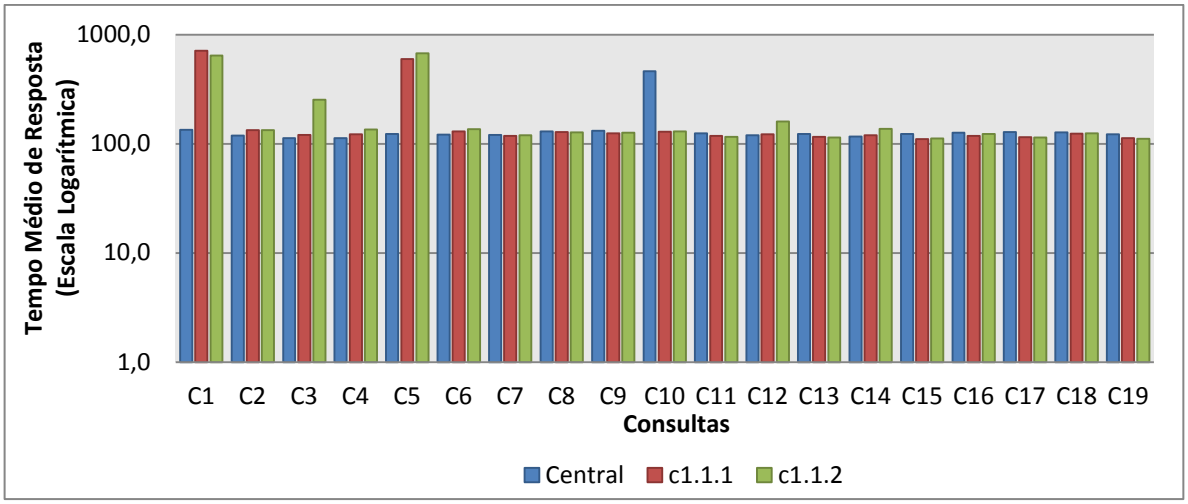


Figura 25: Tempo médio de resposta sobre a base de 40MB (Cenário 1)

Na Figura 26, no gráfico para o cenário 2 podemos perceber que o cenário 2.1.1 apresentou um tempo médio de resposta que o centralizado nas consultas C7, C8, C9, C11, C13, C15, C16, C17, C18 e C19, com ganhos de desempenho variando entre 5% a 14%. Das 11 consultas que se beneficiaram da fragmentação, apenas a consulta C10 não se beneficiou da forma de fragmentação proposta no cenário 2.1.1. Ao analisarmos os resultados foi possível perceber que para obter o resultado da consulta foi necessário submeter subconsultas aos dois fragmentos do cenário 2.1.1 e que um desses fragmentos possuía um tamanho quase igual ao do ambiente centralizado, contendo um total de 24231 documentos. Essa consulta possui duas funções *count* em sua composição, uma na clausula *where* e outra na composição do resultado. Sendo assim, as subconsultas geradas pelo mediador traziam os dados de forma *full*, ou seja, apenas o filtro (total < 2000) foi aplicado em cada subconsulta, pois os cálculos de agregação só poderiam ser realizados após o mediador recebê-los. Esses fatores acabaram influenciando o tempo de comunicação entre o mediador e os adaptadores (havia muitos dados a serem trafegados), e o tempo de composição dos resultados.

O cenário 2.1.2 teve o seu tempo médio de resposta inferior ao centralizado nas consultas C11, C13, C15, C16 e C19 com ganho percentual variando de 10% a 15%. As consultas que não se beneficiaram da fragmentação (C7, C8, C9, C10, C17 e C18) precisaram acessar mais de um fragmento para a obtenção do resultado. Para as consultas C7 e C10 há uma necessidade de composição dos resultados no mediador devido às funções agregadoras. Sendo assim, conforme dito anteriormente, a consulta é executada em cada

adaptador de forma *full* e somente o mediador processa os retornos para a composição do resultado final. Por isso, os tempos ofensores para o resultado do tempo médio de respostas foram o tempo de comunicação entre o mediador e o adaptador que foi maior, o tempo de decomposição da consulta e de processamento do mediador para a composição do resultado final. Por ultimo, as consultas C8, C9, C17 e C18 tiveram os seus tempos médios de respostas muito próximos ao centralizado, tendo como ofensor apenas o tempo de compilação e parse da consulta no mediador, pois para os demais tempos os resultados foram melhores do que no ambiente centralizado.

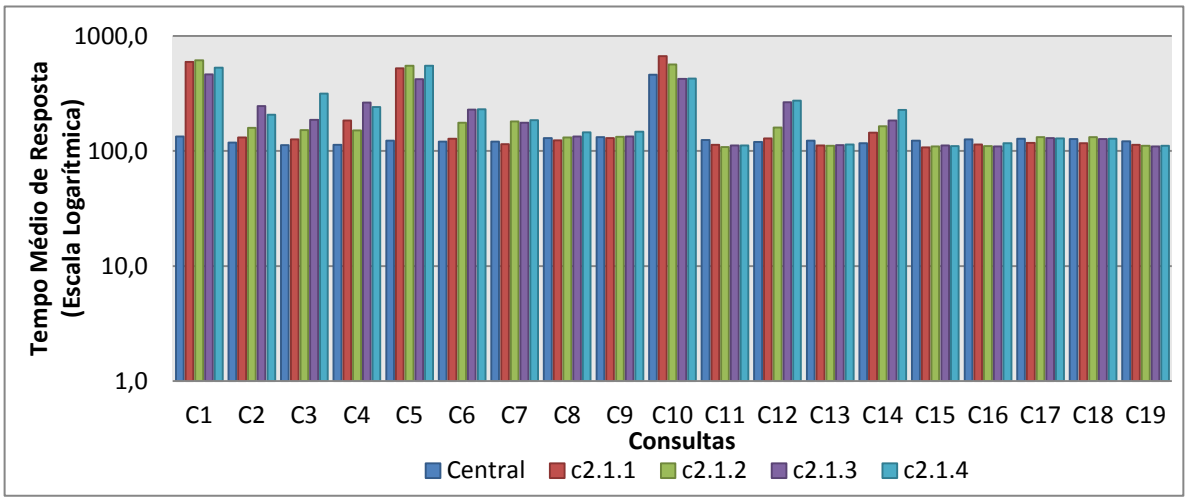


Figura 26: Tempo médio de resposta sobre a base de 40MB (Cenário 2)

O cenário 2.1.3 teve o seu resultado parecido com o 2.1.2. Isso já era esperado, uma vez que a mudança foi apenas para que o fragmento 1 do cenário 2.1.2 se transformasse em 3 fragmentos no cenário 2.1.3. Com esse tipo alteração dos fragmentos esperava-se que apenas a consulta C10 se beneficiasse da fragmentação se comparado ao cenário 2.1.2, e foi isso que ocorreu. Em resumo, no cenário 2.1.3 as consultas C10, C11, C13, C15, C16 e C19 se beneficiaram da fragmentação proposta. Os ganhos percentuais obtidos variaram entre 9% a 15%.

O cenário 2.1.4 também é descrito no gráfico da Figura 26. Nesse cenário, podemos perceber que o resultado foi idêntico ao do cenário 2.1.3. Em outras palavras, as mesmas consultas que se beneficiaram da fragmentação no cenário 2.1.4 também aparecem no cenário 2.1.3 como beneficiadas. Isso também era esperado uma vez que a estrutura dos fragmentos não mudou muito do cenário 2.1.3 para o cenário 2.1.4.

Por último, temos a comparação do cenário centralizado com o cenário 3. A Figura 27 apresenta essa comparação dos tempos médios de resposta entre os dois cenários. Podemos observar que apenas a consulta C10 teve melhor tempo no cenário 3 se comparada com o tempo médio de resposta do ambiente centralizado. Isso ocorre pelo fato do processamento local no fragmentado ter sido menor que no centralizado, embora o tempo de consolidação dos resultados tenha sido maior nessa consulta do que no ambiente centralizado. Um ponto importante observado no cenário 3 foi que as consultas que não se beneficiaram da fragmentação em nenhum dos cenários propostos tiveram seus tempos médios de resposta nesse último cenário inferiores que os demais, ou seja, devido ao equilíbrio no tamanho dos fragmentos desse cenário podemos observar uma melhora nos tempos obtidos.

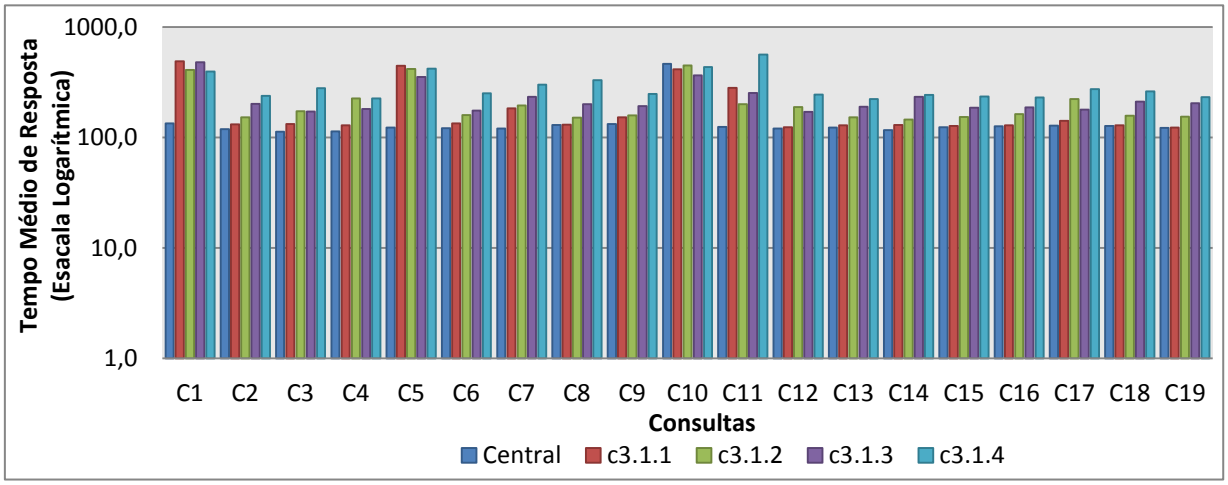


Figura 27: Tempo médio de resposta sobre a base de 40MB (Cenário 3)

5.1.1.3 BASE DE 400MB

O objetivo dessa subseção é apresentar os resultados obtidos ao executarmos 19 consultas sobre 11 cenários em base de dados contendo 259200 documentos XML, totalizando um tamanho da base de 400MB.

Dentre as dezenove consultas executadas nesse experimento, apenas duas delas (C1 e C5) não se beneficiaram de nenhuma das opções propostas (Cenário 1, Cenário 2 e Cenário 3), conforme descrito na Figura 28. O ANEXO IV apresenta os tempos médios de resposta em milissegundos para o experimento de 400MB.

Nas dezessete consultas (C2, C3, C4, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18 e C19) que se beneficiaram de pelo menos uma das propostas de fragmentação, conforme descrito na Figura 28, o ganho percentual entre o cenário centralizado e o melhor dos cenários distribuídos variou entre 55% a 1654%. A Figura 28 apresenta o gráfico do tempo de resposta entre o cenário centralizado e o melhor entre os cenários distribuídos. Note que o gráfico está representado em escala logarítmica.

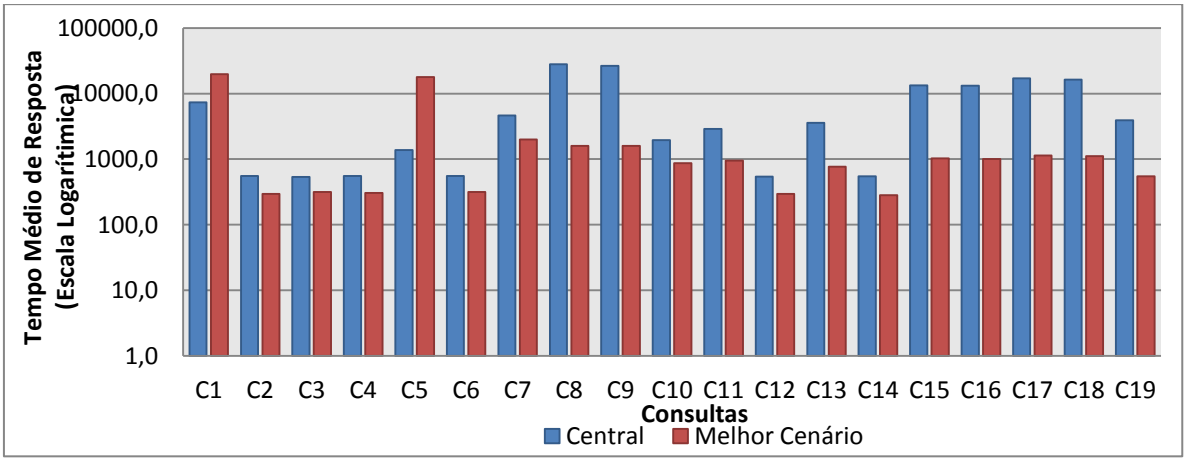


Figura 28: Tempo médio de resposta de 400MB (Centralizado *versus* Melhor Cenário)

A Figura 29 apresenta os tempos médios de resposta das consultas no cenário 1 se comparados com o ambiente centralizado. Como podemos ver, o cenário 1.1.1 foi melhor que o centralizado na execução de 17 das 19 consultas. As únicas consultas que tiveram seus tempos médios de resposta superiores ao ambiente centralizado foram as consultas C1 e C5. Vale ressaltar que essas consultas (C1 e C5) tiverem melhor desempenho no ambiente centralizado em todos os cenários testados. Os ganhos percentuais nesse cenário para as consultas que obtiveram ganhos de desempenho variam entre 13% a 315%.

No cenário 1.1.2, 16 consultas obtiveram os tempos médios de respostas inferiores ao centralizados, são elas: C2, C3, C4, C6, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18 e C19. A consulta C7 teve o seu tempo médio de resposta superior no cenário de execução no ambiente centralizado devido ao tempo gasto na execução local do adaptador, na comunicação com o mediador e na consolidação dos resultados. Nesse cenário, a consulta C7 acessou dois fragmentos (Frag1 e Frag3) com tamanhos semelhantes. Além disso, essa consulta em especial possui em sua cláusula *where* dois predicados simples (total > 7000 e

count($order/order_lines/order_line)>= 5). Entretanto, o mediador ao gerar as subconsultas apenas inclui o predicado “total>7000” para ser calculado nos nós, pois o outro predicado só pode ser aplicado após os resultados parciais serem obtidos nas execuções locais. Esse tempo de consolidação dos resultados foi influenciado por esse fato visto que o mediador precisou efetuar cálculos para obter o resultado final. Para as consultas que se beneficiaram da fragmentação, os ganhos percentuais variaram entre 16% a 781%.

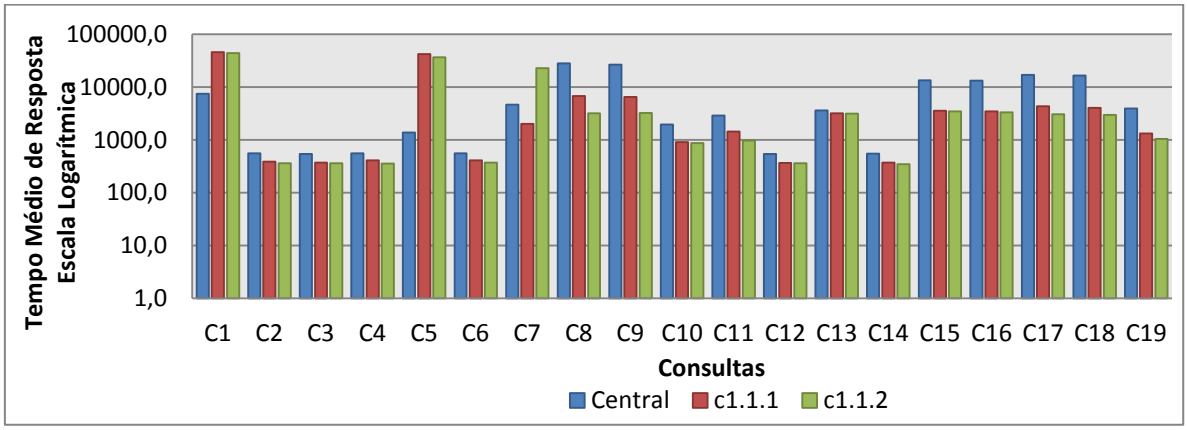


Figura 29: Tempo médio de resposta sobre a base de 400MB (Cenário 1)

Na Figura 30 apresentamos os resultados obtidos no cenário 2 onde foram feitas avaliações em quatro subcenários. No cenário 2.1.1, 11 consultas se beneficiaram da fragmentação proposta se comparadas com o cenário centralizado. Nesse caso, os ganhos variaram entre 1% a 18%. Apenas as consultas C1, C2, C3, C4, C5, C6, C10 e C14 não se beneficiaram da fragmentação. As consultas C2, C3, C4, C6 e C14 são as mesmas que não se beneficiaram da fragmentação nos experimentos de 4MB e 40MB. Além disso, não possuíam em seu predicado de seleção os atributos utilizados na fragmentação desse cenário. Essas consultas precisaram ser submetidas aos dois fragmentos desse cenário. Vale lembrar que um dos fragmentos possuía 242194 documentos, o que é muito próximo ao tamanho da base do ambiente centralizado, que possui 259200 documentos. A consulta C10 também precisou acessar o fragmento com 242194 documentos.

Embora tenhamos consultas que acessem o menor fragmento nesse cenário (2.1.1), a maioria das outras teve que acessar um fragmento com tamanho muito próximo ao da base inteira, o que acabou influenciando o tempo de *parser*, tempo de composição dos resultados pelo mediador e tempo de comunicação. Ou seja, os tempos do cenário 2.1.1 ficaram muito próximos aos do ambiente centralizado, como descrito na Figura 30.

Avaliando o cenário 2.1.2, 15 consultas que se beneficiaram da fragmentação proposta quando comparadas ao cenário centralizado. A diferença percentual do desempenho das consultas executadas variou entre 17% e 760%. Nesse cenário, além das consultas C1 e C5, as consultas C7 e C10 também tiveram melhor desempenho no cenário centralizado se comparado com o 2.1.2. Essas duas consultas possuem características semelhantes: ambas possuem um agregador no predicado de seleção e outro agregador dentro dos resultados.

No cenário 2.1.3, o resultado foi semelhante ao anterior, e as mesmas consultas se beneficiaram da fragmentação proposta se comparada com o centralizado. A diferença ficou por conta do ganho percentual, que nesse cenário variou entre 18% a 786%.

Por último, no cenário 2.1.4 obtivemos os mesmos resultados dos cenários 2.1.2 e 2.1.3 no que diz respeito às consultas que se beneficiaram da fragmentação. O ganho de desempenho nesse cenário ficou entre 18% e 816%. Como conclusão da análise, pode-se perceber que os cenários 2.1.2, 2.1.3 e 2.1.4 obtiveram tempos de resposta muito próximos. Isso já era esperado, uma vez que vários fragmentos eram idênticos dentro desses três cenários.

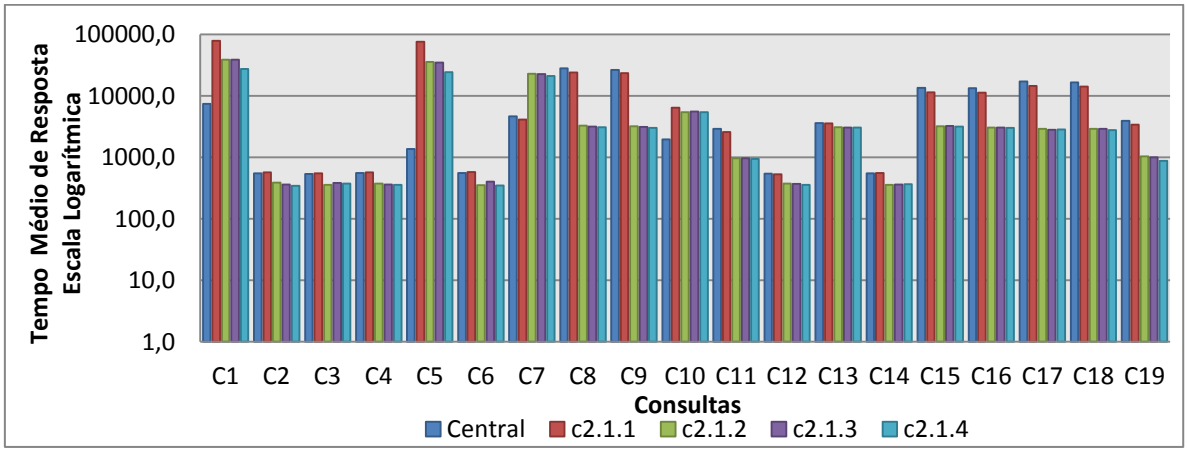


Figura 30: Tempo médio de resposta sobre a base de 400MB (Cenário 2)

A Figura 31 apresenta o ultimo cenário analisado no experimento de 400MB onde visamos avaliar o comportamento dos tempos de respostas das consultas sobre uma base fragmentada sem utilizar os atributos frequentes como critérios de fragmentação. Além disso, é importante lembrar que nesse cenário especificamente há uma distribuição homogênea dos fragmentos, ou seja, os fragmentos possuem tamanhos semelhantes, conforme apresentado na Tabela 6.

Iniciamos a comparação com cenário 3.1.1 e podemos observar que quatorze consultas se beneficiaram desse tipo de fragmentação (C2, C3, C4, C6, C8, C9, C12, C13, C14, C15, C16, C17, C18 e C19) se comparada com o centralizado. As consultas C7, C10 e C11 exigiam um tempo maior do mediador para composição dos resultados por conta da função *count*. O envio dos resultados por todos os nós, visto que a consulta não se beneficiava da fragmentação por atributos, fez com que o tempo médio de resposta fosse superior ao do centralizado. Os ganhos percentuais para as consultas que se beneficiaram da fragmentação nesse cenário variaram de 30% a 729%.

O cenário 3.1.2 apresentou resultado semelhante ao cenário anterior (3.1.1). Entretanto, os tempos médios de respostas para as consultas que não se beneficiaram da fragmentação (C7, C10 e C11) foram inferiores aos do cenário 3.1.1. Além disso, o ganho percentual nesse cenário foi superior ao anterior, com valores variando entre 49% a 1207%. Para esse cenário, cada um dos 4 fragmentos possuía em média um total de aproximadamente 65 mil documentos em cada nó.

Os cenários 3.1.3 e 3.1.4 também apresentaram ganhos nas mesmas consultas dos cenários 3.1.1 e 3.1.2. No cenário 3 foi possível observar que à medida que o numero de nós foi aumentado, e, consequentemente o tamanho dos fragmentos diminuindo, os tempos de resposta foram decrescendo. Se fizermos uma análise entre os quatros subcenários executados, chegamos à conclusão de que o cenário 3.1.4 foi o melhor no geral. A Figura 31 nos permite observar esse comportamento. O cenário 3.1.4 alcançou ganhos percentuais que alternam entre 67% a 1389%.

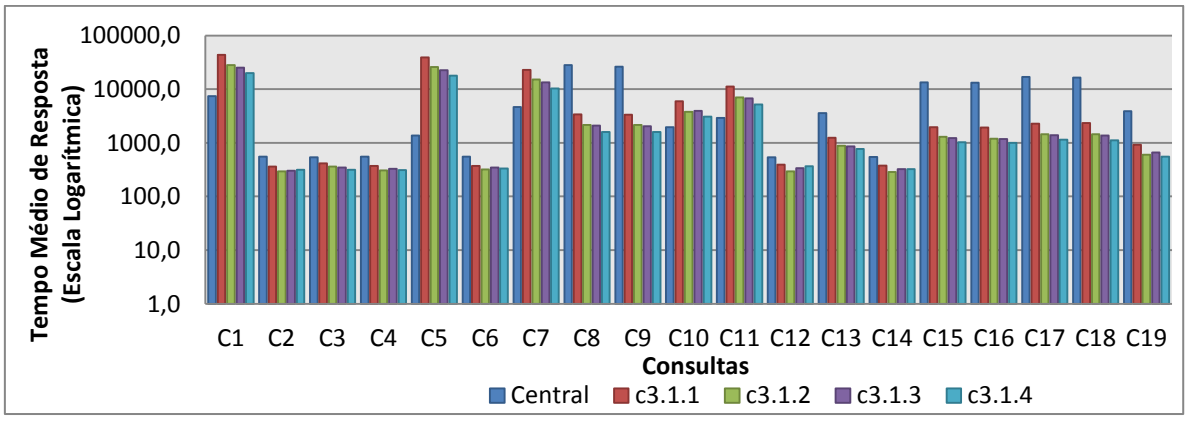


Figura 31: Tempo médio de resposta sobre a base de 400MB (Cenário 3)

Conforme descrito na Figura 28, o experimento de 400MB obteve um ganho sobre dezessete consultas executadas em um total de dezenove consultas. Se efetuarmos uma comparação entre os cenários 1, 2 e 3 para as 17 consultas que se beneficiaram da fragmentação, 6% tiveram suas consultas com melhor tempo de resposta no cenário 1, 6% no cenário 2 e 88% obtiveram seu melhor tempo no cenário 3.

Para melhores esclarecimentos, os gráficos referentes aos tempos médios de processamento remoto estão disponíveis na seção de ANEXO V dessa dissertação.

5.1.1.4 COMPARAÇÃO DOS RESULTADOS

Para complementar as analises realizadas nas subseções 5.1.1.1, 5.1.1.2 e 5.1.2.3, respectivamente, essa subseção apresenta um comparativo entre os três experimentos executados a fim de nos permitir avaliar melhor as situações em que houve ganho de desempenho.

Para iniciarmos as comparações, é importante relembrar que nos experimentos de 4MB e 40MB, onze consultas se beneficiaram da fragmentação enquanto que no experimento de 400MB 17 consultas se beneficiaram de alguma das propostas apresentadas. Segue abaixo as principais observações feitas sobre os três experimentos realizados:

1. As consultas C1 e C5 não se beneficiaram da fragmentação em nenhum dos três experimentos propostos. Para essas consultas, em todos os cenários, todos os fragmentos precisaram ser acessados. Além disso, basicamente foi necessário trazer para o mediador a estrutura de cada documento armazenado em cada fragmento e isso onerou o tempo de processamento local em cada nó e o tempo de comunicação e é claro o tempo de consolidação dos resultados que precisou executar a partir dos resultados as funções agregadoras presentes na consulta. Para ilustrar, abaixo apresentamos a consulta C1.

```
for $a in view('Orders_c7_fh24.xml')/order
return
 <order>
     { $a/@id }
 <order_lines>
        { $a/order_lines/order_line }
 </order_lines>
     { $a/ship_date }
     { $a/total }
```

</order>

2. As consultas C2, C3, C4, C6, C12 e C14 não se beneficiaram das fragmentações propostas nos experimentos de 4MB e 40MB. As diferenças percentuais de desempenho do melhor cenário (cenário que obteve o melhor tempo para essas consultas) comparado com o ambiente centralizado ficaram entre 4 % a 12 % e 1% a 9 %, respectivamente. Esses valores nos mostram que as diferenças dos tempos de execução nesses cenários são muito próximas dos obtidos no ambiente centralizado, e que à medida que o tamanho da base aumentou essa diferença percentual foi diminuindo.

   Já no experimento de 400MB, essas consultas de beneficiaram da fragmentação. Essas 6 consultas em todos os 11 cenários avaliados precisaram acessar todos os fragmentos, pois o predicado de seleção contido nelas não se beneficiaria de nenhuma das formas de fragmentação realizada sobre as bases. Entretanto, foi possível observar que no experimento de 400MB, embora elas tivessem que acessar todos os fragmentos, o tamanho dos fragmentos em cada nó era muito menor se comparado com a base centralizada. Outro ponto interessante observado no experimento de 400MB é que os melhores tempos para essas consultas se deram no cenário 3 (subcenário 3.1.2) no qual a base não estava fragmentada por nenhum dos atributos frequentes encontrados nas consultas. O cenário 3.1.2 foi melhor que os demais cenários pelo fato dos fragmentos gerados nesse cenário terem basicamente os mesmos números de documentos, ou seja, havia uma distribuição uniforme dos dados. Isso acabou acarretando certo balanceamento de carga.

3. Para as consultas que usavam o atributo de seleção frequente “total” a Figura 32 apresenta o número de cenários onde houve benefício da fragmentação, em cada um dos experimentos executados. Como podemos observar o experimento de 400MB obteve, em geral, uma maior abrangência de cenários com ganho de desempenho em cada uma das consultas.

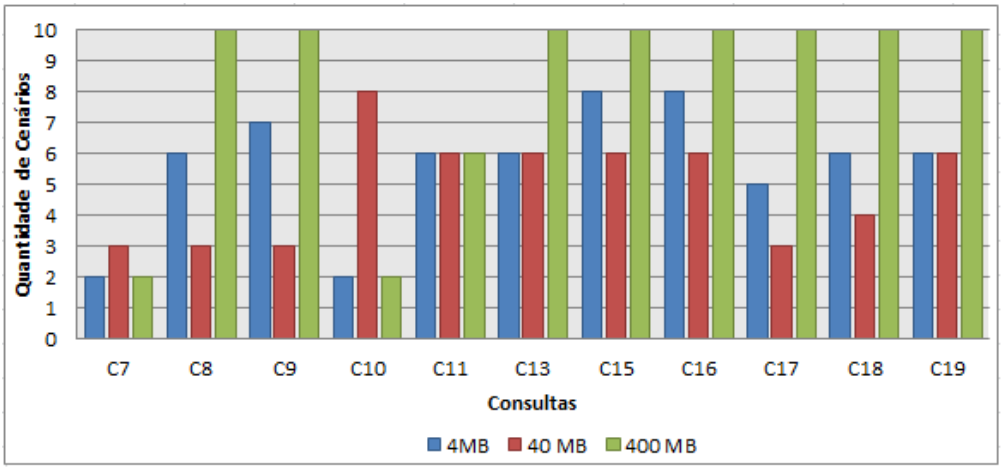


Figura 32: Número de cenários versus consultas (fragmentação horizontal)

Como conclusão, podemos perceber que para os experimentos com bases de 4MB e 40MB, os ganhos foram apenas sobre as consultas que possuíam em seu predicado de seleção o atributo frequente utilizado para efetuar a fragmentação na maioria dos cenários testados. Pelo fato das bases de dados não serem de tamanhos consideráveis questões como tempo de busca, tempo de latência e taxa de transferência interna podem ter contribuído para que os resultados obtidos não tenham sido tão significativos se comparados com o experimento de 400MB. Fazendo uma extrapolação do resultado, podemos supor que se aumentarmos o tamanho da base de 400MB o ganho com a fragmentação tende aumentar gradativamente. Entretanto, há um limite onde se chega à saturação do processo. Quanto a esse aumento gradativo da base não foi realizado nenhum experimento que comprovasse essa afirmação.

Para os cenários 2 e 3 uma das variáveis analisadas foi o número de nós disponíveis. Para essa variável, realizamos variações de 2, 4, 6 e 8 nós. Sendo assim, uma análise importante é a validação se essas alternâncias de números de nós/fragmentos nos fornece alguma informação relevante. Com objetivo de analisar essas questões foram gerados gráficos de tendências que são apresentados a seguir. Inicialmente, avaliando o experimento de 4 MB na Figura 33, percebe-se para o cenário 2 os tempos médios de respostas dos cenários 2.1.2, 2.1.3 e 2.1.4 ficaram muito próximos e isso já era esperado uma vez que a diferença de tamanho dos fragmentos entre esses cenários não era grande. Com isso, as consultas não sofreram tanto impacto com os aumentos dos fragmentos. Já no cenário 3, pode-se perceber que os tempos médios foram aumentando à medida que o número de fragmentos/nós foram aumentando.

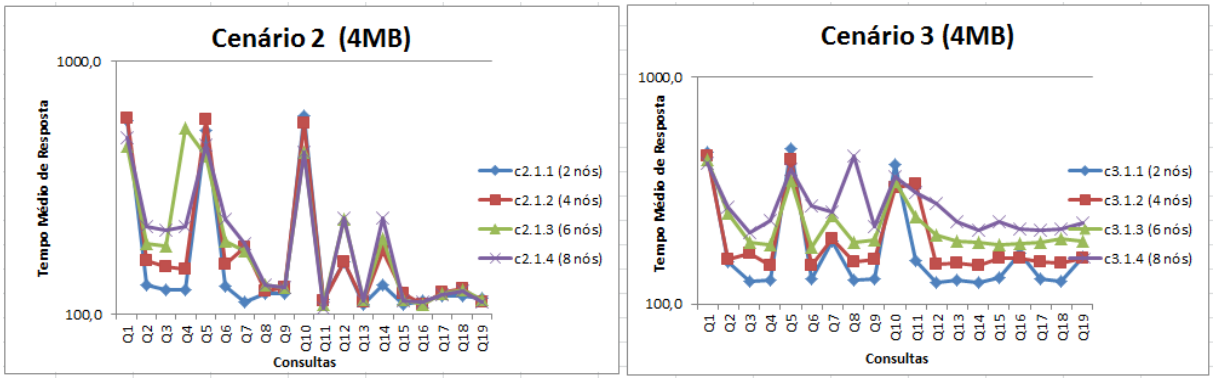


Figura 33: Gráfico de Tendência (Cenário 2 e 3) – Fragmentação Horizontal (4MB)

Para o experimento de 40 MB apresentado na Figura 34, o resultado foi parecido ao do experimento de 4 MB, onde no cenário 2 tivemos resultados similares nos cenários 2.1.2, 2.1.3 e 2.1.4. No cenário 3, os tempos médios de resposta continuaram aumentando à medida que o número de fragmentos/nós aumentaram.

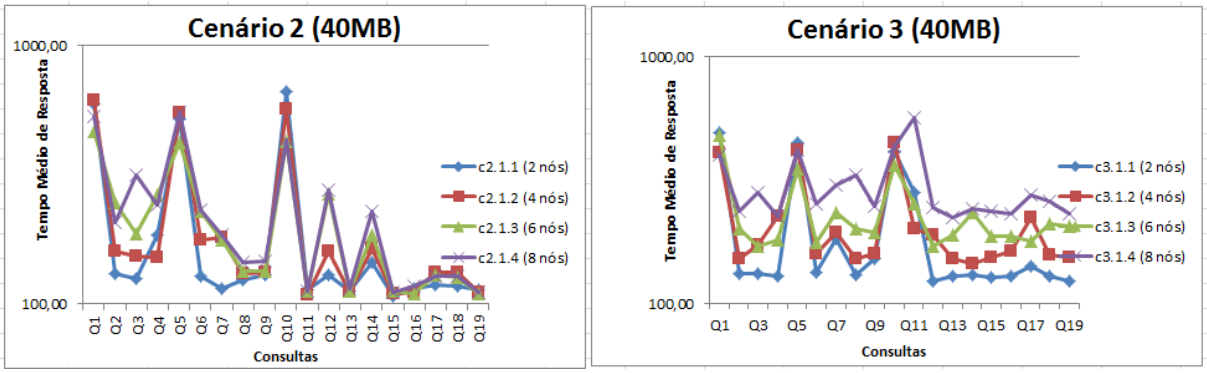


Figura 34: Gráfico de Tendência (Cenário 2 e 3) – Fragmentação Horizontal (40MB)

Por último, no experimento de 400 MB descrito na Figura 35 há uma inversão dos resultados apresentados nos experimentos de 4 e 40MB. No cenário 2, percebe-se que o cenário 2.1.1 tornou-se o pior dentre os 4 cenários avaliados. Isso é esperado uma vez que um dos fragmentos contém basicamente o tamanho da base inteira. Já a semelhança nos resultados dos demais 3 cenários (2.1.2, 2.1.3 e 2.1.4) permanecem. Entretanto, o cenário 2.1.4 aparece como sendo o melhor entre eles. Já para o cenário 3 percebe-se que o cenário 3.1.4 tornou-se o melhor entre os demais (3.1.1, 3.1.2 e 3.1.4). Além disso, é possível observar que há uma linearidade entre os resultados dos 4 cenários.

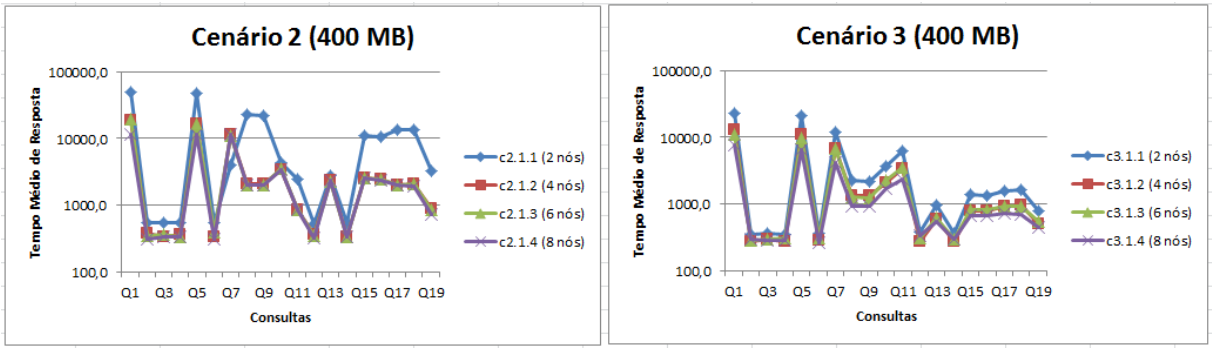


Figura 35: Gráfico de Tendência (Cenário 2 e 3) – Fragmentação Horizontal (40MB)

Por fim, avaliamos, para cada experimento, o percentual de consultas que se beneficiou da fragmentação. Essa análise é mostrada na Tabela 21, e mostra que para os três experimentos executados, o critério de fragmentação proposto no cenário 1 foi o que mais beneficiou as consultas.

Tabela 21: Benefícios da Fragmentação Horizontal

Experimento	Critério de Fragmentação	% de Consultas Beneficiadas
4MB	• Frequência das Consultas	57.89%
40MB	• Frequência das Consultas • Frequência das Consultas e Alocação Distribuída	57.89%
400MB	• Frequência das Consultas	89.47%

Como complemento aos pontos descritos anteriormente, temos:

- Para bases com cardinalidades pequenas e médias, o tamanho do fragmento não é um fator determinante no tempo médio de resposta das consultas;
- Para bases com cardinalidades pequenas e médias, o tempo médio de resposta sobre as consultas é diretamente proporcional ao número de fragmentos acessados;

5.1.2 ANÁLISE DE RESULTADOS DA FRAGMENTAÇÃO VERTICAL

Assim como foi feito para a fragmentação horizontal, nosso trabalho nessa subseção apresenta as análises dos resultados obtidos com a fragmentação vertical. Para esse tipo de fragmentação também foram realizados experimentos sobre 3 tamanhos de bases, são elas: 10MB, 100MB e 1GB, respectivamente. Essas bases foram fragmentadas sobre 6 cenários que tiveram seus resultados comparados com o resultado obtido no ambiente centralizado.

Baseado nas fragmentações realizadas, espera-se que as consultas se beneficiem de pelo menos um dos cenários propostos. Consequentemente, nós realizamos um quadro com as expectativas dos resultados sobre os diversos cenários. A Tabela 22 descreve em quais

cenários é esperado que uma dada consulta se beneficie da fragmentação. Por exemplo, para a consulta C1 espera-se que os melhores tempos ocorram no cenário 1.1 ou cenário 3.2, pois nesses cenários será acessado um único fragmento, cujo tamanho é o menor possível para essa consulta.

Tabela 22: Resultado Esperado na Fragmentação Vertical

Consulta	Cenário 1.1	Cenário 2.1	Cenário 2.2	Cenário 2.3	Cenário 3.1	Cenário 3.2
C1	X					X
C2	X		X	X		
C3	X			X		X
C4	X	X	X	X		
C5	X	X	X	X		
C6	X			X		X
C7	X			X		X
C8	X					X
C9	X	X	X	X		
C10	X					X
C11	X			X		X
C12	X	X	X	X		
C13		X				
C14		X	X			

A subseção 5.1.2.1 descreve os resultados obtidos com a fragmentação vertical sobre uma base de dados de um único documento de tamanho igual a 10MB. O detalhamento dos resultados do experimento para base de 100MB é apresentado na subseção 5.1.2.2. Já a subseção 5.1.2.3 mostra os resultados do experimento sobre a base de 1GB. Por último, as comparações entre os três experimentos são realizados na subseção 5.1.2.4.

5.1.2.1 BASE DE 10MB

Conforme dito anteriormente, os experimentos de fragmentação vertical foram realizados sobre 14 consultas em 6 cenários. O ANEXO IV apresenta um resumo dos tempos médios de resposta obtidos no experimento de 10MB. Os tempos estão representados em milissegundos.

Para detalharmos os resultados obtidos nos tempos médios de resposta para o experimento de 10MB, a Figura 36 descreve o tempo do cenário centralizado com o melhor dos tempos dos cenários fragmentados (escala logarítmica). Para as consultas que se beneficiaram da fragmentação vertical, os ganhos percentuais variaram entre 1% a 88%. Como podemos observar, dentre as 14 consultas executadas apenas uma delas (C3) não se

beneficiou de nenhuma das 6 opções de fragmentação proposta. Entretanto, a diferença no tempo médio entre o ambiente centralizado e o melhor dos cenários analisados foi de apenas 3 segundos. A consulta C3 possui uma função de agregação e um predicado de seleção em sua composição. Ao analisarmos os tempos de resposta dessa consulta verificamos que o ofensor do resultado foi o tempo de processamento local no nó onde estava o fragmento e o tempo de compilação pelo mediador para a geração da subconsulta. Como a base de dados é relativamente pequena, questões relacionadas a tempo de latência e de busca em disco podem ter influenciado nos resultados.

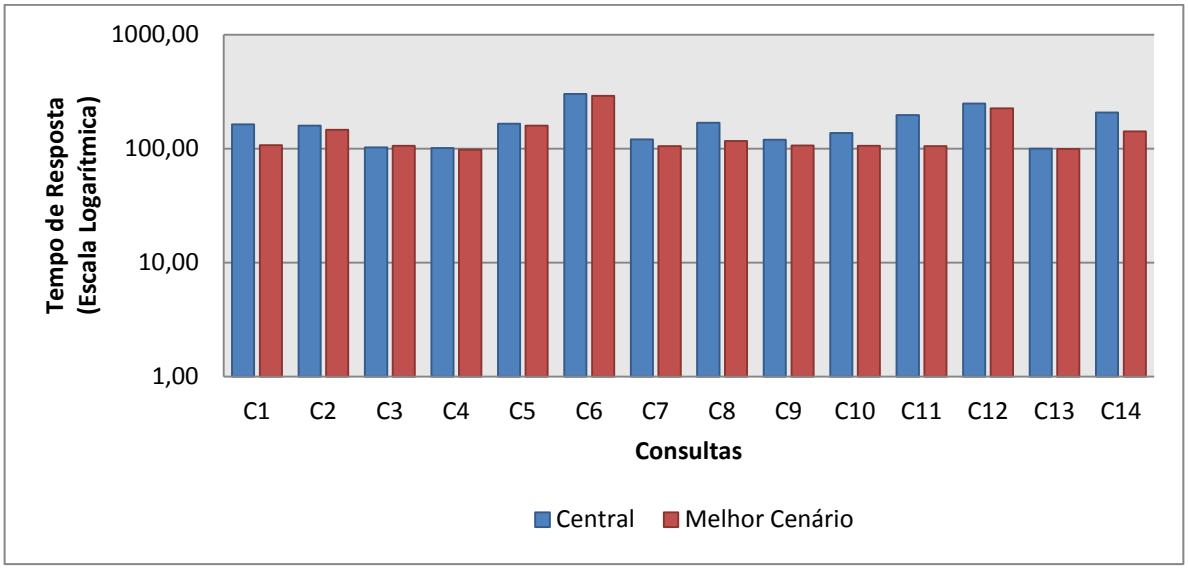


Figura 36: Tempo médio de resposta de 10MB (Centralizado *versus* Melhor Cenário)

Na Figura 37 apresentamos a comparação entre o tempo médio de resposta do cenário centralizado e o obtido no cenário 1. No cenário 1, o objetivo é avaliar o comportamento dos resultados se a base for fragmentada através da distribuição total dos fragmentos. Ao analisarmos a Figura 37 podemos perceber que os tempos entre os dois cenários, em geral, foram muito próximos. Entretanto, apenas as consultas C6, C7, C10, C11 e C12 se beneficiaram efetivamente da fragmentação. Todas essas consultas acessaram um único fragmento para a composição do resultado. Além dessas consultas, as consultas C1, C2, C3, C4, C5, C8 e C9 também fazem acesso a um único fragmento e não se beneficiaram da fragmentação proposta no cenário 1. Para essas consultas verificamos que na maioria dos casos, o tempo médio de processamento local no nó foi superior ao tempo no ambiente centralizado, embora o fragmento fosse menor no cenário 1. Os maiores fragmentos foram acessados pelas consultas C4, C5 e C9, pois acessavam a subárvore /site/regions que contém

basicamente a metade de toda a base de 10MB. Já as consultas C13 e C14 acessaram mais de um fragmento e possuíam funções de agregação, o que acabou acarretando em um maior tempo de processamento pelo mediador e pelo próprio nó.

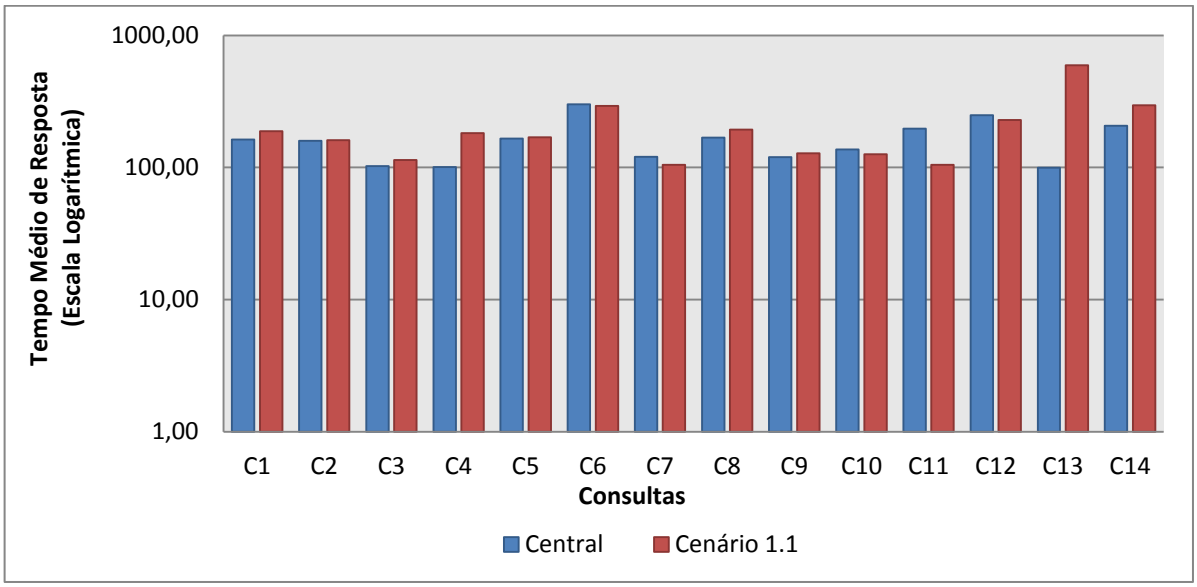


Figura 37: Tempo médio de resposta sobre a base de 10MB (Cenário 1)

Para o cenário 2, a Figura 38 descreve os resultados obtidos se comparado ao ambiente centralizado. No cenário 2.1 todas as consultas acessavam apenas um dos dois fragmentos. A fragmentação proposta apresentou melhor desempenho se comparado ao ambiente centralizado em 9 consultas (C1, C2, C5, C6, C8, C11, C12, C13 e C14), onde os ganhos percentuais ficaram entre 1% a 81%. Para as consultas que não se beneficiaram da fragmentação proposta (C3, C4, C7, C9 e C10) verificamos que os ofensores foram o tempo de processamento remoto pelo nó, que pode ter sofrido influencia pelo tempo de latência e busca, embora o fragmento tivesse um tamanho inferior à base original.

No cenário 2.2, 10 consultas se beneficiaram da fragmentação proposta (C1, C2, C4, C6, C7, C8, C10, C11, C12 e C14) se comparadas com o centralizado. Os ganhos nessas consultas variaram de 3% a 46%. As consultas C3, C5, C9 e C13 não se beneficiaram da fragmentação tendo os seus tempos médios de respostas superiores neste cenário. Na consulta C13 foi necessário acessar mais de um fragmento, o que acabou afetando o tempo do processamento local, tempo de comunicação, compilação e composição dos resultados do mediador. Por fim, para as consultas C3, C5 e C9, o tempo de processamento local influenciou o tempo médio de resposta.

O cenário 2.3 obteve ganhos na fragmentação proposta em 8 consultas (C1, C2, C6, C7, C8, C9, C11 e C12), com variações de ganhos entre 3% a 79% se compararmos com os resultados obtidos no cenário centralizado. Para as consultas C3, C4, C5 e C10, cabem as mesmas observações quanto ao tempo de processamento do nó local como sendo o ofensor do tempo de processamento final. Além delas, as consultas C13 e C14 não se beneficiaram pelo fato de ter que acessar mais de um fragmento.

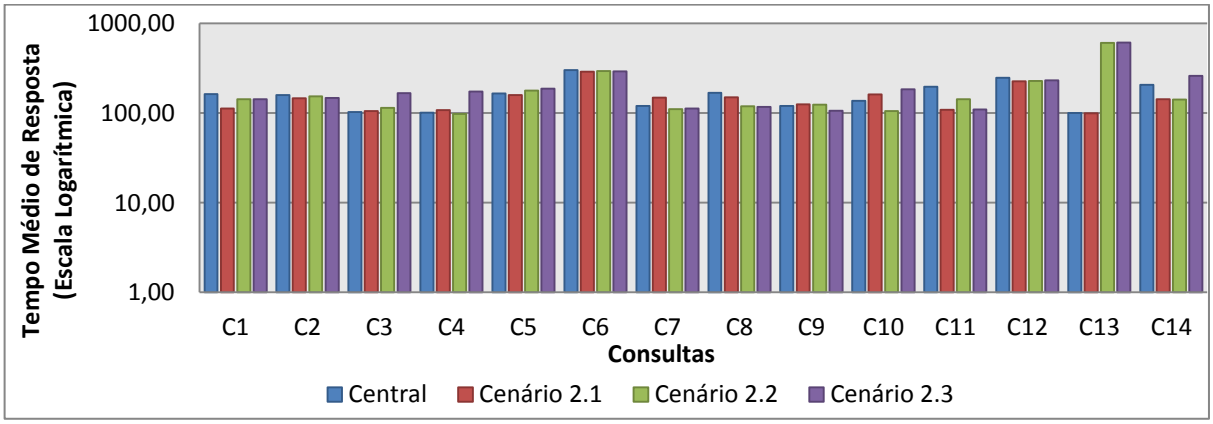


Figura 38: Tempo médio de resposta sobre a base de 10MB (Cenário 2)

A Figura 39 descreve o resultado obtido no cenário 3. Nesse caso, as consultas C1, C2, C6, C7, C8, C10, C11 e C12 se beneficiaram da fragmentação proposta no cenário 3.1. Entretanto, as consultas que possuíam funções agregadoras (C3, C4, C13 e C14) não se beneficiaram da fragmentação. Além dessas consultas, outras duas consultas (C5 e C9) não se beneficiaram da fragmentação proposta no cenário 3.1. Na consulta C5, o tempo médio de resposta foi idêntico ao do ambiente centralizado. No entanto, o que influenciou o tempo dessa consulta foi o tempo de comunicação entre o mediador e o nó que processou a subconsulta, que nesse caso é idêntica à consulta original. Já na consulta C9, o tempo de processamento local no nó acabou sendo superior, o que influenciou no resultado final. Os ganhos percentuais nesse cenário ficaram entre 2% e 67%. Nesse cenário, apenas a consulta C13 acessou mais de um fragmento. Vale ressaltar que na proposta de fragmentação por agrupamento parcial, o fragmento 1 acessado por todas as consultas possui um tamanho próximo ao da base original. Mesmo assim, foi possível tirar benefícios dessa fragmentação descartando as subárvores que praticamente não são acessadas pelas consultas.

Finalmente, no cenário 3.2 as consultas que se beneficiaram da fragmentação foram as mesmas apresentadas no cenário 3.1. Entretanto, os ganhos percentuais na comparação

com o ambiente centralizado foram superiores aos do cenário 3.1. Isso era esperado visto que os tamanhos dos fragmentos nesse cenário eram menores. Os ganhos ficaram entre 3% e 79%.

As fragmentações nesses dois cenários foram bem diferentes. No primeiro (3.1), todos os atributos utilizados em projeções nas consultas frequentes foram colocados em um mesmo fragmento, exceto o /site/categories que aparece em uma única consulta e por isso não foi agrupado no mesmo fragmento. O segundo cenário (3.2) visou agrupar os atributos por afinidade. De acordo com os algoritmos aplicados, era esperado que as subárvores /site/closed_auctions e /site/people ficassem em um mesmo fragmento. Entretanto, por questões de definição da fragmentação vertical, não foi possível agrupar essas subárvores em um só fragmento, o que não permitiu obter maiores vantagens sobre essa forma de fragmentação.

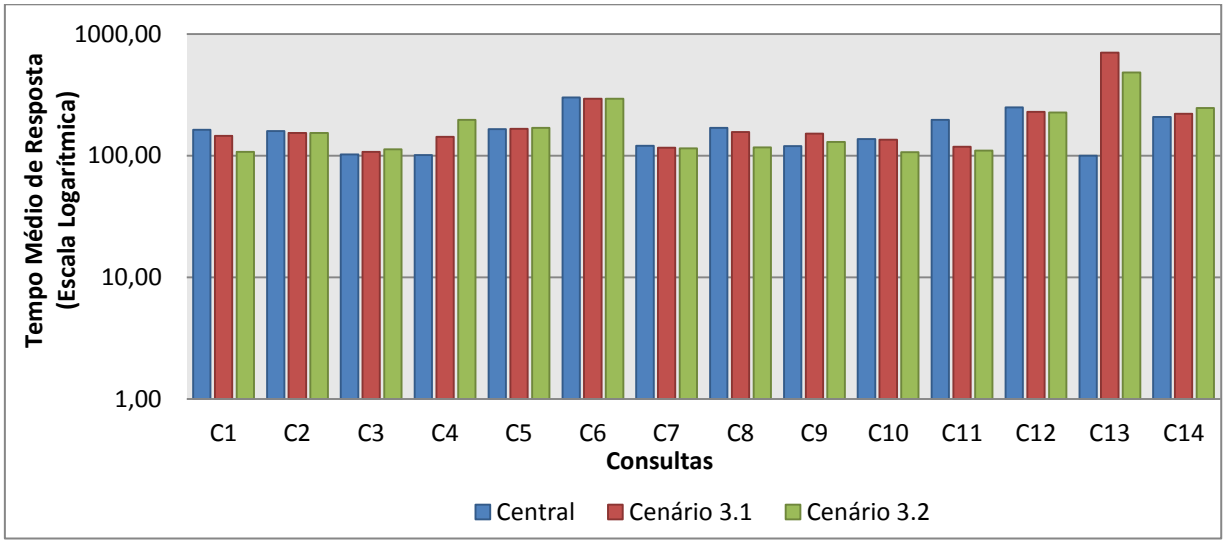


Figura 39: Tempo médio de resposta sobre a base de 10MB (Cenário 3)

Para concluirmos a analise do experimento de 10MB, avaliamos se os resultados obtidos atenderam as expectativas apresentadas na Tabela 22. Aproximadamente, 85% das consultas obtiveram um tempo inferior ao centralizado em um dos cenários descritos na Tabela 22 e isso mostra que há vantagens se fragmentarmos uma base de 10MB.

5.1.2.2 BASE DE 100MB

Ao analisarmos os resultados obtidos na execução do experimento sobre a base de 100MB podemos perceber que dentre as 14 consultas executadas, apenas 2 delas não se beneficiaram de nenhuma das propostas de fragmentação vertical definidas nesse trabalho.

Entretanto, para essas consultas as diferenças dos tempos médios de respostas no ambiente centralizado e o melhor dos cenários do ambiente fragmentado foi muito pequena, com diferenças percentuais de 1% para a consulta C14 e 5% para a consulta C9. Ou seja, para essas consultas houve praticamente um empate nos tempos médios de resposta e não consideramos o ganho do cenário como sendo significativo. Para a consulta C9 o cenário centralizado obteve tempo de resposta inferior aos demais, embora a consulta fosse realizada em uma base de tamanho superior em relação aos demais cenários. Conforme dito anteriormente, essas diferenças de tempo foram muito pequenas. A seção ANEXO IV apresenta um resumo dos resultados dos tempos médios de resposta obtidos nesse experimento.

Para resumir os resultados obtidos no experimento de 100MB, a Figura 40 descreve o tempo do cenário centralizado com o melhor dos tempos dos cenários fragmentados, em escala logarítmica. Para as consultas que se beneficiaram de alguma das propostas de fragmentação vertical, os ganhos percentuais variaram entre 2% a 711%.

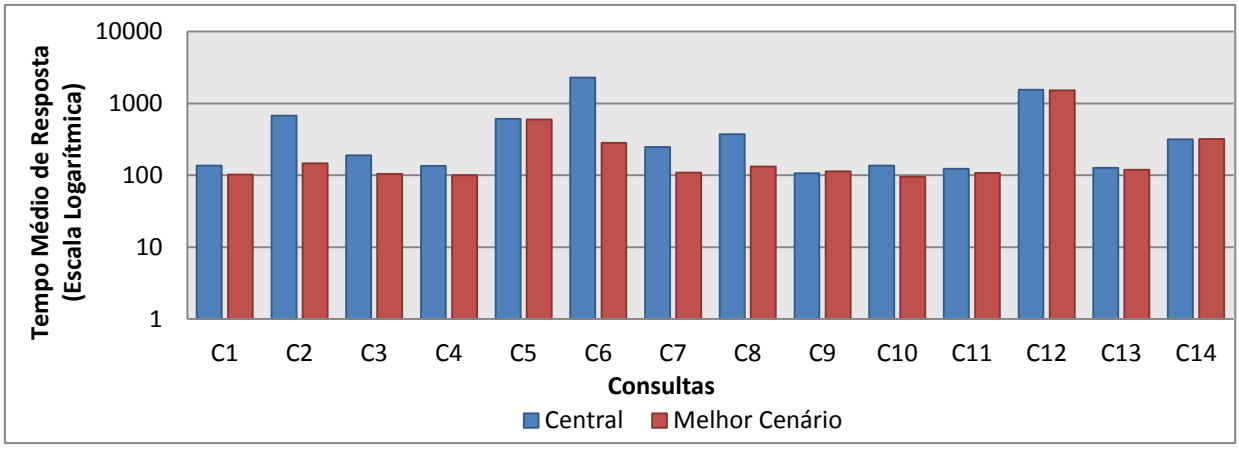


Figura 40: Tempo médio de resposta de 100MB (Centralizado *versus* Melhor Cenário)

No cenário 1 o objetivo era avaliar os benefícios de uma fragmentação onde houvesse a distribuição total dos fragmentos. Ao compararmos com o centralizado, as consultas C2, C3, C4, C5, C6, C7, C10, C11 e C12 se beneficiaram da fragmentação proposta. Além das consultas C9 e C14 já mencionadas anteriormente, a consulta C1 não se beneficiou dessa fragmentação vertical. Nessa consulta, o principal ofensor foi o tempo de *parsing* no mediador. Já as consultas C8 e C13 também tiveram seus tempos médios de resposta inferiores no ambiente centralizado. Isso ocorreu pelo fato da consulta C13 acessar três subárvores para a composição do resultado final e, além disso, a consulta exigia três cálculos

de agregação sobre as três subárvores. A Figura 41 apresenta os tempos de resposta para o cenário 1 se comparado ao centralizado. Para as consultas que se beneficiaram da fragmentação nesse cenário (Cenário 1), os ganhos ficaram entre 1% e 690%.

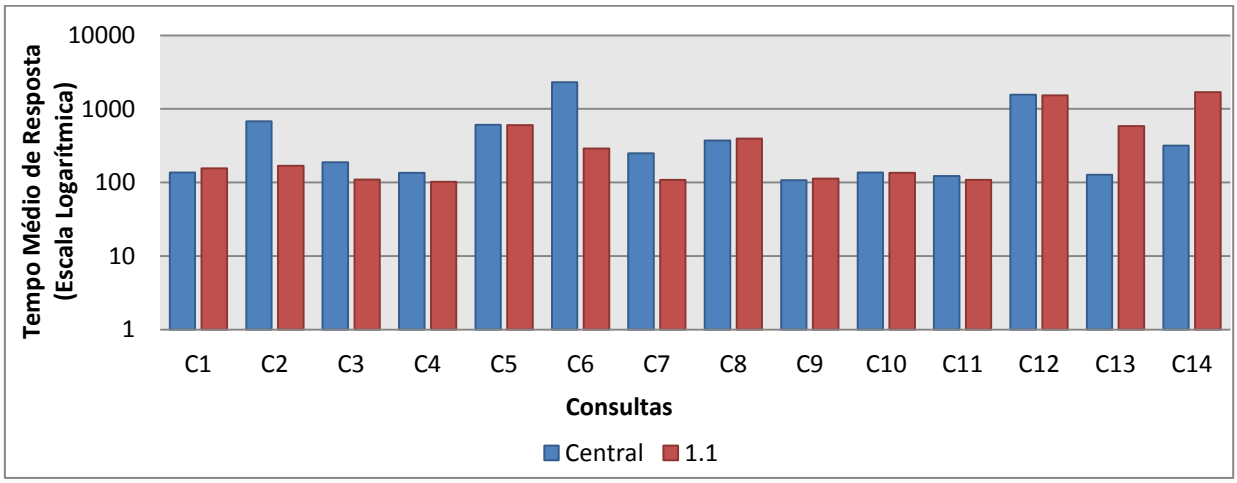


Figura 41: Tempo médio de resposta sobre a base de 100MB (Cenário 1)

Já o cenário 2 propõe avaliar os tempos de resposta para as consultas onde há uma distribuição dos fragmentos por tamanho. Esse cenário foi dividido em 3 outros cenários, o que nos permitiu aumentar o número de fragmentos baseado no número de nós disponíveis para alocação. Os resultados obtidos nesse cenário são apresentados na Figura 42. No cenário 2.1, onde havia apenas dois fragmentos de tamanhos próximos, apenas as consultas C4, C11, C12 e C13 se beneficiaram, com ganhos percentuais de 35%, 4%, 1% e 6%, respectivamente. Para as consultas C2, C6, C7 e C14, as diferenças dos tempos do cenário centralizado e o cenário 2.1 são de apenas 1%, ou seja, pode-se dizer que houve um empate nos tempos de resposta para essas consultas. Para as demais consultas que não se beneficiaram da fragmentação proposta (C1, C3, C5, C8, C9, C10), as diferenças percentuais variaram de 3% a 13%.

Já no cenário 2.2, as consultas C2, C4, C5 e C6 se beneficiaram da fragmentação vertical proposta. Entretanto, apenas a consulta C2 obteve um desempenho significativo, apresentando um ganho de 360% se comparado com o ambiente centralizado. Essa relevância no resultado da consulta C2 se deve pelo fato da mesma acessar um único fragmento de 33MB. Nas demais consultas, os ganhos de desempenho não são relevantes se comparados ao ambiente centralizado.

Por último, na Figura 42 é apresentado o cenário 2.3 onde a fragmentação vertical gerou 4 fragmentos com tamanhos 60MB, 15.5MB, 18MB e 33MB, respectivamente.

Nesse cenário, as consultas C2, C3, C6, C7, C11 e C12 se beneficiaram da fragmentação com ganhos variando entre 13% a 711%. Esse resultado é importante visto que as consultas C3, C6, C7 e C11 acessaram um único fragmento com 18MB de tamanho, e C2 acessou um fragmento de 33MB. A consulta C12 não apresentou diferenças de desempenho nos cenários analisados. Essa consulta acessa a subárvore /site/regions nos três cenários (2.1, 2.2 e 2.3). Em todos os cenários, essa subárvore está num fragmento de mesmo tamanho (60MB). A mesma observação vale para as consultas C5 e C9 que acessam a subárvore /site/regions e obtiveram um resultado muito próximo ao apresentando no ambiente centralizado.

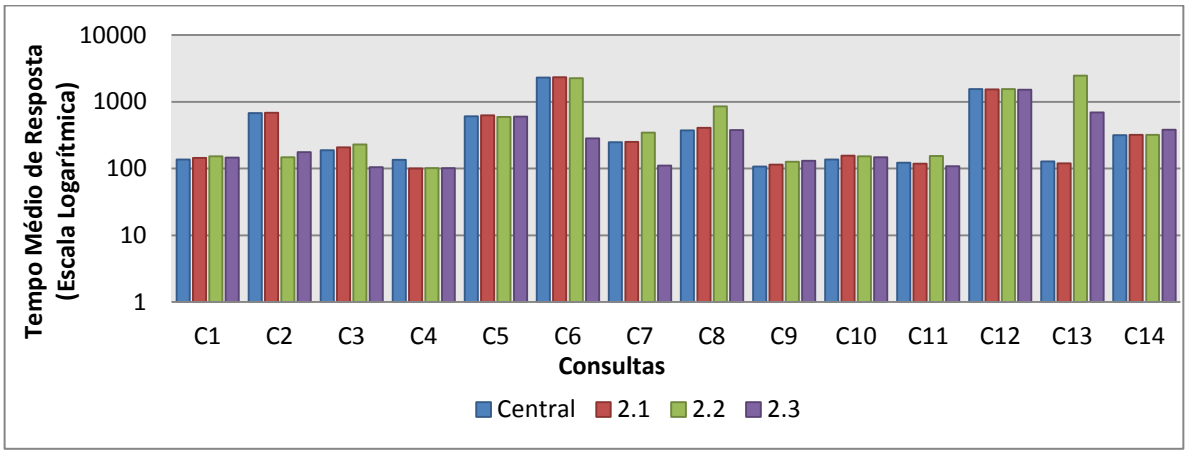


Figura 42: Tempo médio de resposta sobre a base de 100MB (Cenário 2)

O cenário 3 visou analisar os tempos médios de resposta das consultas ao fragmentar a base através de um agrupamento parcial (cenário 3.1) e um agrupamento por afinidade (cenário 3.2). A Figura 43 descreve os tempos de resposta desses dois cenários se comparados com o ambiente centralizado. Para o cenário 3.1 apenas as consultas C2, C4, C11 e C12 se beneficiaram da fragmentação proposta, com ganhos percentuais pouco significativos (2%, 34%, 3% e 1% respectivamente). Esse resultado era esperado visto que o fragmento 1 que é acessado por todas as consultas executadas possui um tamanho 64.5MB (mais da metade do tamanho da base centralizada).

Já o cenário 3.2, que avalia o agrupamento por afinidade, obteve um melhor resultado se comparado com o cenário 3.1. No cenário 3.2, as consultas C1, C5, C6, C8, C10, C11 e C12 se beneficiaram desse tipo de fragmentação vertical com ganhos percentuais variando entre 1% a 182%. As consultas C1, C6 e C8 acessam apenas a subárvore /site/people e pelo fato do fragmento 3 nesse tipo de fragmentação só conter essa subárvore as mesmas se

beneficiaram da fragmentação. Já as consultas C5 e C12 acessam a subárvore /site/regions e nesse cenário o acesso é feito sobre o fragmento 1 que possui tamanho de 61.5MB. Já as consultas C4 e C9 que também acessam apenas a subárvore /site/regions tiveram um resultado diferente, pois nesse caso o ambiente centralizado apresentou um tempo menor no que diz respeito ao tempo de resposta do adaptador. A diferença dos tempos de resposta para essas duas consultas foi de aproximadamente 10%. As consultas C3, C6 e C7 tiveram um resultado muito próximo no ambiente centralizado e nos dois subcenários apresentados na Figura 43. Vale ressaltar todas essas 3 consultas acessam a subárvore contendo /site/closed_auctions.

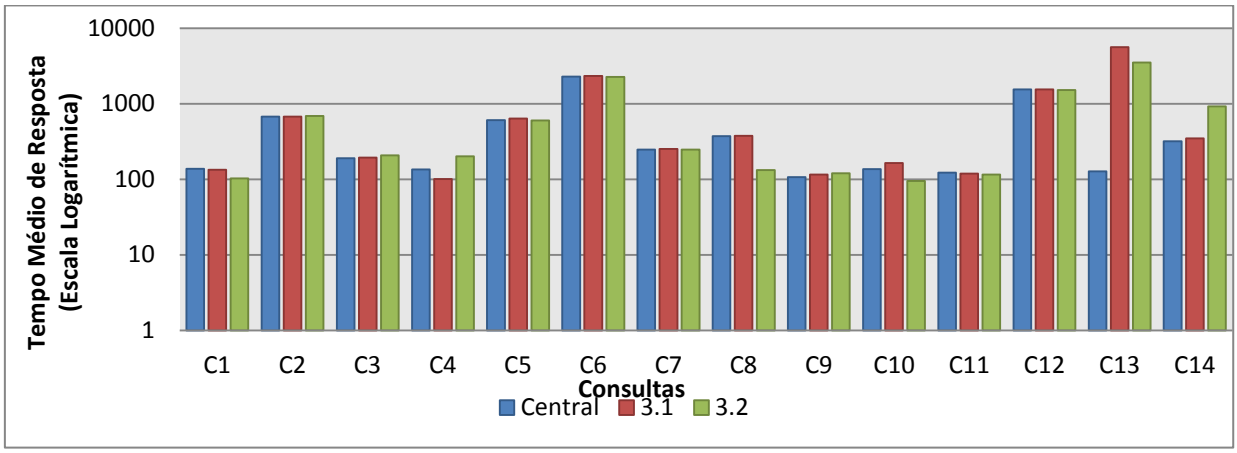


Figura 43: Tempo médio de resposta sobre a base de 100MB (Cenário 3)

Para concluirmos a analise do experimento de 100MB, avaliamos se os resultados obtidos atenderam as expectativas apresentadas na Tabela 22. Aproximadamente, 85% das consultas obtiveram um tempo inferior ao centralizado em um dos cenários descritos na Tabela 22 e isso mostra que houve sim ganhos ao fragmentarmos uma base de 100MB. Entretanto, era esperado que com uma base média os ganhos fossem maiores do que com a base pequena (10MB), mas as análises mostraram justamente o contrário.

5.1.2.3 BASE DE 1GB

Para analisarmos o comportamento das fragmentações verticais propostas nos experimentos executados sobre a base de 1GB, no ANEXO IV apresenta os resultados dos tempos médios de respostas das 14 consultas. Como podemos perceber, todas as consultas se beneficiaram da fragmentação em pelo menos um dos cenários de fragmentação avaliados. A Figura 44 nos permite visualizar de forma gráfica os resultados obtidos. Esse

gráfico descreve o resultado obtido no cenário centralizado com o melhor tempo dos cenários experimentados.

Na maioria das consultas, os ganhos da fragmentação foram expressivos, chegando a quase 10000% na consulta C6, por exemplo. As Figuras (Figura 45, Figura 46 e Figura 47) apresentam as comparações dos tempos médios de resposta em cada um dos cenários avaliados, permitindo avaliar quais foram as consultas que se beneficiaram da fragmentação proposta no cenário se comparados com o ambiente centralizado.

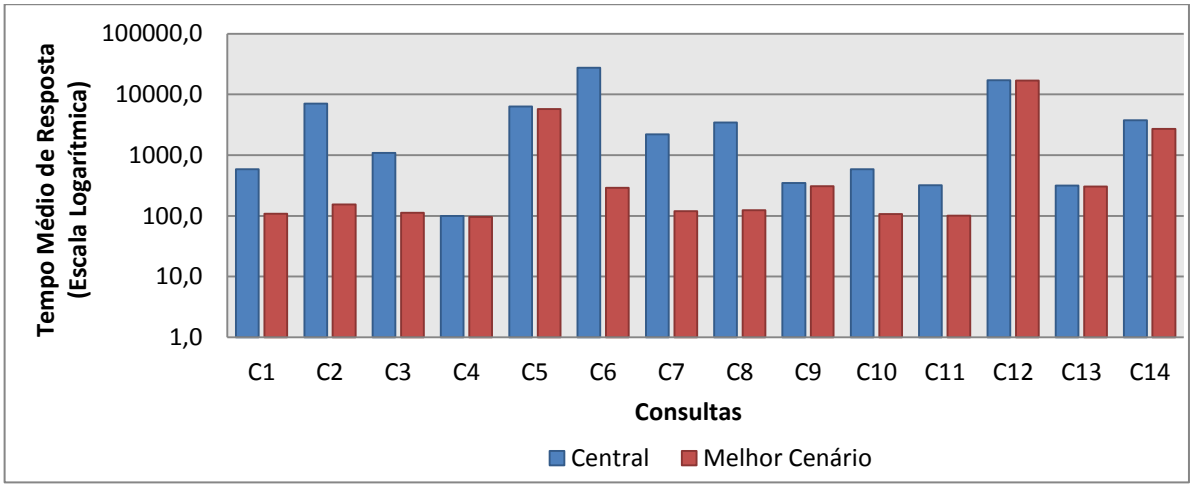


Figura 44: Tempo médio de resposta de 1GB (Centralizado *versus* Melhor Cenário)

Ao analisarmos a Figura 45 podemos perceber que dentre as 14 consultas executadas, 10 delas se beneficiaram da fragmentação proposta no cenário 1.1. Já as consultas C4, C12, C13 e C14 não se beneficiaram dessa fragmentação. Para essas consultas que não se beneficiaram podemos perceber que 3 delas (C4, C13 e C14) possuem uma função de agregação em sua composição e duas delas fazem acesso a dois fragmentos (C13 e C14). A consulta C4 acessa o fragmento 1 com um tamanho de aproximadamente 600MB e efetua a função *count* na geração de seu resultado. Ao compararmos com o ambiente centralizado podemos perceber que embora o tamanho do fragmento seja mais da metade do tamanho da base centralizada o processamento local foi maior que no cenário 1.1. O cenário 1.1 apresentou um tempo de comunicação menor e isso se deve ao fato do tamanho do fragmento ser menor, mas não foi tão significante ao ponto do tempo total de processamento ser melhor que no ambiente centralizado. Além disso, o mediador obteve um tempo de processamento para compilação e geração da subconsulta maior no cenário 1.1.

Para a consulta C12, que também só acessou o fragmento 1, o tempo de processamento local no cenário 1.1 foi menor que no ambiente centralizado. Entretanto, o tempo de comunicação remota nesse cenário foi o ofensor no resultado final. Embora, assim como na consulta C4, o mediador tenha apresentado um tempo de processamento para compilação e geração da subconsulta maior nesse cenário.

As consultas C13 e C14 apresentaram razões semelhantes para o desempenho apresentado no ambiente fragmentado. Em ambos os casos, foram acessados mais de um fragmento. Além disso, essas consultas possuíam a função *count*() na composição dos resultados. Nessas duas consultas, o tempo de processamento local no pior fragmento foi superior ao ambiente centralizado. Além disso, tivemos um tempo de comunicação remota e compilação do mediador superior nesse cenário.

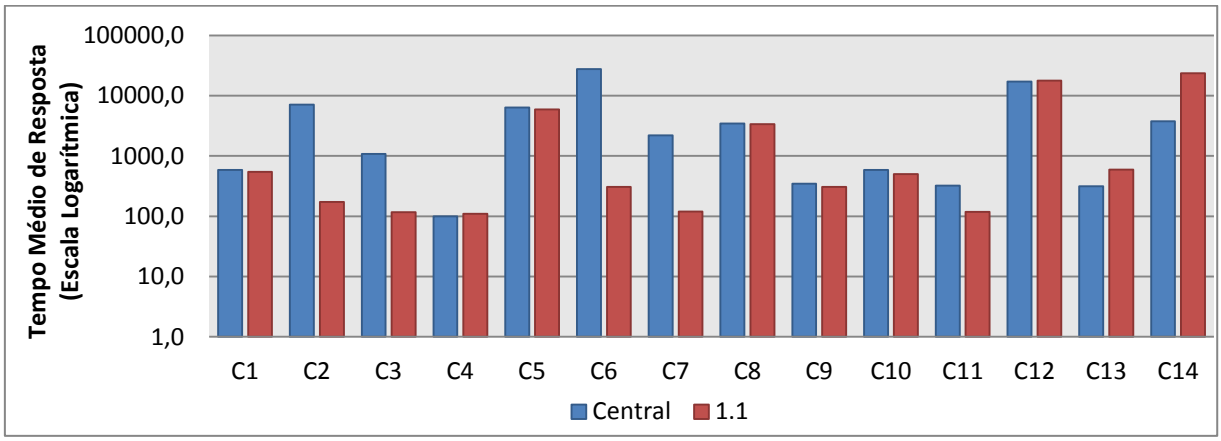


Figura 45: Tempo médio de resposta sobre a base de 1GB (Cenário 1)

Já no cenário 2.1, que se encontra descrito na Figura 46 podemos perceber que 5 consultas não se beneficiaram da fragmentação proposta (C1, C4, C11, C12 e C14). Para esse cenário, as consultas C4 e C12 possuem o mesmo comportamento apresentado no cenário 1.1, pois a consulta acessa o fragmento 1 que é igual em ambos os cenários. Para as consultas C1, C11 e C14, o tempo de processamento local foi superior ao apresentado no ambiente centralizado, embora o fragmento do cenário 2.1 tivesse um tamanho inferior ao da base do centralizado.

A Figura 46 mostra que para o cenário 2.2, tivemos 5 consultas que não se beneficiaram da fragmentação (C4, C6, C11, C12 e C13). Para as consultas C4 e C12, o comportamento foi o mesmo apresentado nos cenários 1.1 e 2.1. Já a consulta C13 precisou acessar dois fragmentos (fragmento 2 e 3) para a composição de seus resultados. Essa consulta

apresentou um tempo maior na compilação do mediador, geração da subconsultas e consolidação dos resultados do mediador, visto que havia uma função *count* no resultado final da consulta.

O cenário 2.3 teve um resultado melhor que os demais cenários, tendo apenas 3 consultas (C4, C13 e C14) que não se beneficiaram da fragmentação. As justificativas para as três consultas não terem se beneficiado da fragmentação foram a função *count,* que aparece nas três consultas e o acesso a mais de um fragmento para as consultas C13 e C14.

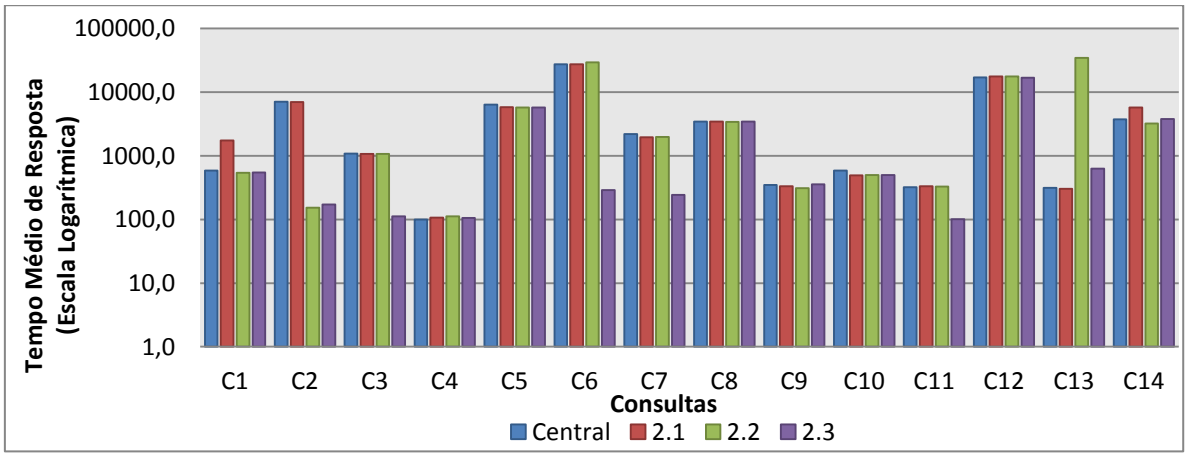


Figura 46: Tempo médio de resposta sobre a base de 1GB (Cenário 2)

Por último, temos o cenário 3, que está representando na Figura 47. O cenário 3.1 é um cenário muito próximo ao centralizado visto possui três fragmentos, sendo que o maior deles só não contém dados das subárvores /site/categories e /site/catgraph que são os menores fragmentos gerados. Nesse cenário, 6 consultas (C2, C6, C8, C11, C13 e C14) não se beneficiaram da fragmentação. Se analisarmos o gráfico podemos perceber que os tempos desse cenário se comparado com centralizado na maioria das consultas são muito próximos. Isso já era esperado visto que o tamanho do fragmento acessado pelas consultas é praticamente igual ao tamanho da base completa (cenário centralizado).

O cenário 3.2 visa gerar fragmentos de acordo com a afinidade dos atributos. Nesse caso, percebe-se um resultado bem semelhante ao apresentando no cenário 3.1, onde cinco consultas (C2, C6, C11, C13 e C14) não se beneficiaram dessa forma de fragmentação vertical. Para maiores esclarecimentos dos resultados, os gráficos referentes aos tempos médios de processamento remoto estão disponíveis no ANEXO V dessa dissertação.

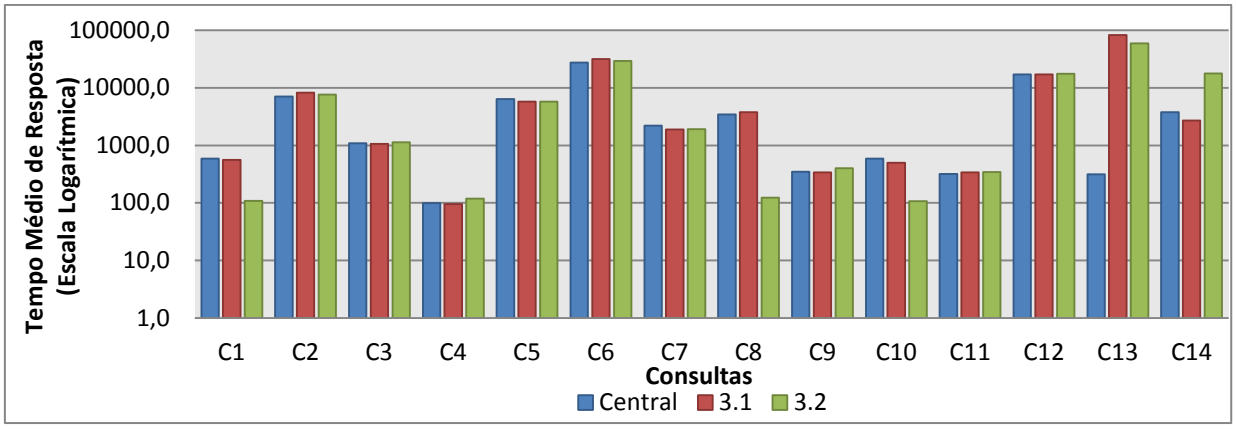


Figura 47: Tempo médio de resposta sobre a base de 1GB (Cenário 3)

5.1.2.4 COMPARAÇÃO DOS RESULTADOS

No cenário 1 (distribuição total dos fragmentos), na base 10MB obtivemos 5 consultas que se beneficiaram da fragmentação proposta. Em contrapartida, para o experimento de 100MB, o número de consultas que se beneficiou passou para 9 consultas e no experimento de 1GB, tivemos 10 consultas que se beneficiaram desse cenário.

Para o cenário 2 (distribuição por tamanho), no experimento de 10MB, o melhor entre os cenários foi o cenário 2.2 com 10 consultas se beneficiando dessa fragmentação. Já no experimento de 100MB, o melhor resultado se deu no cenário 2.3. Por ultimo, no experimento de 1GB, o cenário 2.3 foi o melhor se comparado com 2.1 e 2.2, com 11 consultas beneficiadas com a fragmentação. A Figura 48 mostra o gráfico de tendência do cenário 2 no decorrer dos 3 experimentos.

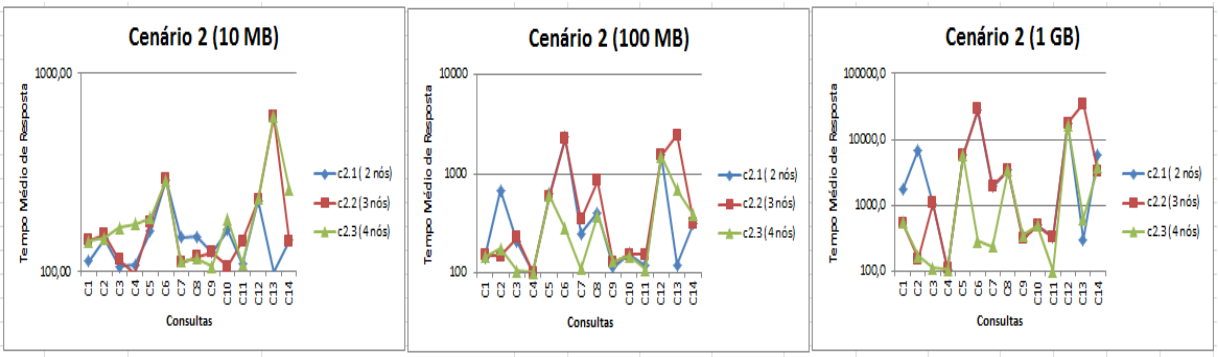


Figura 48: Gráfico de Tendência (Cenário 2) – Fragmentação Vertical

Por último, no cenário 3 (agrupamento), no experimento de 10MB, as quantidades de consultas que se beneficiaram da fragmentação proposta nos cenário 3.1 e 3.2 foram as mesmas. Já no experimento de 100MB, o cenário 3.2 foi superior ao cenário 3.1, com um total de 7 consultas beneficiadas pela fragmentação.

Ao analisarmos de forma geral os três experimentos, percebe-se que as consultas se beneficiaram das fragmentações principalmente sobre as bases de 10MB e 1GB, conforme a Figura 49. Como podemos perceber as consultas que mais se beneficiaram das propostas de fragmentação descritas nessa dissertação são: 10MB (C1, C2, C6, C7, C8, C10, C11, C12 e C14); 100MB (C2, C4, C5, C6, C10, C11 e C12); 1GB (C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 E C12).

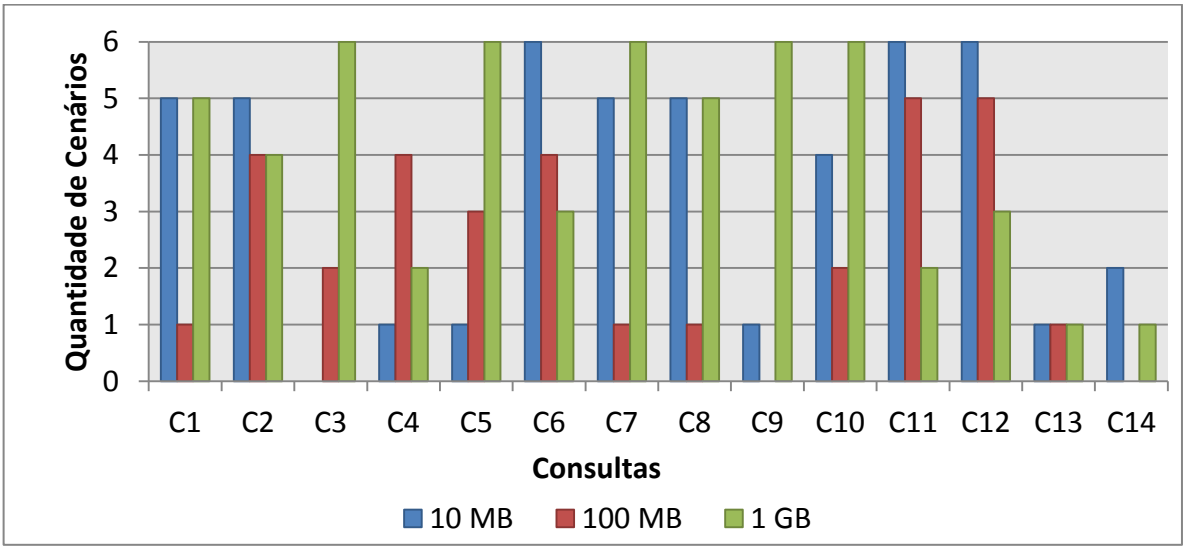


Figura 49: Número de cenários versus consultas (fragmentação vertical)

Por fim, a Tabela 23 mostra o percentual de consultas que se beneficiou da fragmentação nos três experimentos executados. De acordo com a tabela, o critério de fragmentação proposto no cenário 2 foi o que mais beneficiou as consultas.

Tabela 23: Benefícios da Fragmentação Vertical

Experimento	Critério de Fragmentação	% de Consultas Beneficiadas
10MB	Alocação e Tamanho da Base	71.4%
100MB	Distribuição Total	64.2%
1GB	Alocação e Tamanho da Base	79.5%

5.2 HEURÍSTICAS

O projeto de fragmentação de dados XML deve considerar diversos critérios, de forma análoga ao que foi feito para outros modelos de dados que o antecederam. Essa seção apresenta alguns destes critérios, que foram inspirados em seus equivalentes no modelo relacional (OZSU; VALDURIEZ, 2011) e orientado a objetos (BAIÃO et al., 2004; FLORENTINO,

2003), e cuja influência no processamento de consultas sobre as bases XML distribuídas foi comprovada empiricamente durante nossos experimentos.

Existem importantes questões que devem ser avaliadas em um projeto de distribuição na escolha do processo de fragmentação sobre cada coleção. Este trabalho concentra-se em três pontos citados que têm grande influencia sobre a qualidade da distribuição dos dados:

Características da aplicação: operações que são executadas sobre os dados;

Semântica da base de dados: representada pelos atributos e seus relacionamentos.

Informações quantitativas: instâncias das coleções e seus respectivos tamanhos.

As informações referentes à aplicação são fundamentais para entender o seu comportamento, ou seja, permitem identificar, por exemplo, a real utilização da aplicação pelo usuário. Com isso, é necessário avaliar a forma como a mesma está sendo utilizada e fragmentar a base de forma a trazer respostas mais rápidas ao usuário. A etapa de extração dos dados é dividida em três partes: ***operações de acesso***, que é composta pelas informações referentes às operações aplicadas sobre as consultas; ***esquema lógico dos dados***, onde temos a apresentação do esquema XML e seus respectivos dados, ou seja, documentos XML; por ultimo, ***informações quantitativas***, que consistem de informações referentes ao volume de dados e o tipo de base dados na qual será aplicada a fragmentação. A seguir há o detalhamento dos tipos de dados extraídos nesta etapa.

- **Consultas frequentes**

  Ao disponibilizar as informações das consultas frequentes é possível avaliar as operações aplicadas e com isso obter estatísticas sobre o acesso aos dados. Dentre as principais estatísticas que podem ser obtidas das consultas frequentes estão:

  - **Classificação das operações**

    Classificar as operações (seleção e projeção) presentes nas consultas.

  - **Frequências das operações:**

    Analisar de forma quantitativa as operações presentes nas consultas.

  - **Cardinalidade do atributo:**

    Avaliar a cardinalidade do atributo ao qual esta sendo aplicada a operação. Essa avaliação é importante principalmente na decisão para a fragmentação vertical, visto

que este tipo de fragmentação não pode ser aplicada sobre atributos com cardinalidade maior que 1, pois isso compromete a reconstrução dos fragmentos.

- **Frequência dos atributos:**

  Analisar a frequência dos atributos sobre as consultas frequentes. Além disso, verificar as afinidades desses atributos, ou seja, comparar todas as ocorrências dos atributos em uma mesma consulta.

- **Tipo da base de dados**:

Consiste na avaliação da base que sofrerá a fragmentação. Dependendo da base existem limitações quanto ao tipo de fragmentação a ser aplicada. Uma base com um único documento só pode sofrer fragmentação vertical ou hibrida.

- **Volume de dados**:

Consiste em aferir o tamanho da base tanto em quantidade de documentos quanto em espaço em disco ocupado. Este tipo de análise permite verificar a viabilidade da fragmentação sobre a base. Para esse trabalho classificamos as bases em três tamanhos, são eles:

1. Baixa: < 10MB;
2. Média: 10MB a 100MB;
3. Alta: > 100MB.

. A subseção 5.2.1 descreve as heurísticas propostas para a fragmentação horizontal baseado nos experimentos executados e as heurísticas para a fragmentação vertical são apresentadas na subseção 5.2.2.

5.2.1 HEURÍSTICAS PARA FRAGMENTAÇÃO HORIZONTAL

Para que seja tomada a decisão de qual distribuição dos fragmentos é a mais apropriada para um determinado cenário de aplicação, esse trabalho se baseou nas variáveis analisadas e nos experimentos executados para a geração das heurísticas para fragmentação horizontal. A elaboração de ferramentas e algoritmos de apoio ao projeto de fragmentação de dados não é escopo desse trabalho. Entretanto, os resultados obtidos nessa dissertação sugerem heurísticas que podem ser usadas em algoritmos de fragmentação de dados XML. Dentre as heurísticas, podemos citar:

- A fragmentação horizontal só pode ser aplicada sobre uma coleção que esteja em uma base de dados do tipo múltiplos documentos, em função da definição de fragmento horizontal (ANDRADE et al., 2006);
- A fragmentação horizontal só deve ser aplicada sobre uma coleção C quando a maioria das consultas sobre C contiver predicados de seleção sobre C;
- A fragmentação só deve ser aplicada sobre uma coleção C de cardinalidade baixa ou média quando a maioria das consultas sobre C contiver predicados simples sobre um mesmo atributo;
- A fragmentação deve ser aplicada sobre uma coleção C de cardinalidade alta;
- Para bases com cardinalidades altas, os fragmentos resultantes da fragmentação horizontal de uma coleção C devem ter tamanhos homogêneos;
- A fragmentação só deve ser aplicada sobre uma coleção C quando as consultas sobre C não contiverem funções agregadoras.

5.2.2 HEURÍSTICAS PARA FRAGMENTAÇÃO VERTICAL

Assim como foi realizado na fragmentação horizontal, experimentos foram realizados com o objetivo de avaliar as variáveis descritas na seção anterior com o objetivo de derivar heurísticas para fragmentação vertical de bases de dados XML. Dentre as conclusões que foram possíveis chegar a partir dos resultados dos experimentos, são elas:

- A fragmentação vertical só deve ser aplicada sobre uma coleção C quando a maioria das consultas sobre C contiver projeções sobre C;
- A fragmentação com distribuição total dos fragmentos não deve ser aplicada a coleções de cardinalidade baixa;
- Subárvores que não aparecem nas consultas frequentes devem ser retiradas dos fragmentos mais acessados;
- Se as consultas sobre C possuem mais de um atributo de projeção que não pertence à mesma subárvore, deve-se manter essas subárvores no mesmo fragmento;
- A fragmentação com distribuição total dos fragmentos ou por agrupamento de afinidades deve ser aplicada sobre coleções de cardinalidade média.

5.3 CONSIDERAÇÕES FINAIS

O objetivo desse capitulo foi apresentar os resultados dos experimentos executados para fragmentação horizontal e vertical de dados XML. Através desses resultados, derivamos um conjunto de heurísticas que permitem auxiliar o projetista de um projeto de distribuição de dados a otimizar a fragmentação e distribuição dos documentos XML.

CAPÍTULO 6: CONCLUSÃO

A grande quantidade de dados XML disponíveis na Web e dentro das organizações traz consigo um grande desafio no processamento de consultas sobre ambientes distribuídos. Surge então a necessidade da aplicação de técnicas que permitam um processamento de consultas mais eficiente. Neste sentido, técnicas de fragmentação de dados e processamento paralelo de consultas sobre bases de dados distribuídas têm sido adotadas.

No entanto, a forma adequada para a geração de fragmentos XML não está bem definida na literatura. Há muitas definições de fragmentos XML, mas poucas propostas são concentradas em como usar essas definições para realmente fragmentar uma base de dados (isso é chamado de projeto de fragmentação). Definir critérios de fragmentação de dados XML em um ambiente distribuído de forma eficiente é uma tarefa complexa. Os projetistas da aplicação se deparam com várias opções de técnicas de fragmentação possíveis. Com o objetivo de apoiar os projetistas nessa etapa tão importante dentro de um projeto de distribuição, essa dissertação apresenta derivações de heurísticas a partir de uma análise experimental, a fim de aumentar o desempenho do processamento de consultas. A proposta inicial desse trabalho foi analisar o comportamento de uma aplicação com as diversas alternativas de fragmentação e distribuição desses fragmentos nos bancos de dados.

Esse trabalho contribui para a caracterização das dificuldades associadas à definição das melhores formas de se fragmentar dados XML utilizando como insumo conhecimentos prévios de outros modelos de bancos de dados. A inspiração foi retirada dos modelos relacional e orientado a objetos, que têm metodologias sólidas para o projeto de fragmentação de bases de dados. Foram realizados experimentos que permitiram não só avaliar o desempenho das consultas sobre um banco de dados XML fragmentado e distribuído, como também fornecer insumos que contribuem para uma melhor definição dos fragmentos a serem gerados em um projeto de fragmentação.

Para efetuar a etapa experimental dessa dissertação, foi utilizado um protótipo que permite executar consultas sobre uma base de dados XML distribuída. Esse protótipo foi desenvolvido por Figueiredo, Braganholo e Mattoso (2010) e sofreu alterações na forma de

comunicação entre o mediador e os adaptadores nessa dissertação. Além disso, o banco de dados nativos XML utilizado na versão original foi substituído pelo Sedna (FOMICHEV et al., 2006) que possui um desempenho superior ao eXist (MEIER, 2003), utilizado na versão anterior.

Para dar suporte a todos os experimentos, alguns programas em Java foram desenvolvidos, permitindo automatizar algumas das etapas que antecedem a execução dos experimentos em si. Dentre os programas desenvolvidos merecem destaque: Geradores Automáticos de Fragmentos Verticais e Horizontais, Alocação de Fragmentos no Banco de Dados Sedna no cluster, Geração Automática de Matriz de Afinidade Agrupada para definição de fragmentos verticais.

Os experimentos permitiram avaliar em quais cenários é recomendável ou não efetuar a fragmentação da base de dados. Os resultados desse trabalho destacam a importância de uma avaliação prévia das consultas frequentes no processo de definição de fragmentos dos dados.

Na fragmentação horizontal foi possível perceber que ao aumentarmos o tamanho da base, os benefícios da fragmentação também cresciam. Para as bases pequenas e médias, apenas as consultas que possuíam o predicado de seleção que foi utilizado na fragmentação se beneficiaram de pelo menos uma das fragmentações propostas. Em contrapartida, para o experimento de base maior, foi possível perceber que as consultas que não possuíam esse predicado de seleção também se beneficiaram da fragmentação.

Já na fragmentação vertical, os experimentos apresentaram um resultado melhor sobre as bases pequenas e grandes. Na base pequena, tivemos benefício em pelo menos uma das fragmentações propostas em 13 consultas de um total de 14. Na base grande, todas as consultas se beneficiaram de pelo menos um dos cenários avaliados. Para essa base, os ganhos percentuais se comparados ao centralizado chegaram a quase 10000%.

A principal contribuição desse trabalho é um conjunto de heurísticas que podem ser aproveitadas para projeto de distribuição de banco de dados XML nativos em geral. Resumidamente, as heurísticas foram divididas de forma a levar em conta o tamanho da base de dados.

Como resultado desse trabalho, obtivemos as seguintes publicações:

- Workshop de Teses e Dissertações em Banco de Dados - (SILVA et al., 2010)
    - Titulo: Metodologia para Projeto de Fragmentação de Dados XML sobre Bases Distribuídas -
    - Autores: Tatiane Silva, Vanessa Braganholo, Marta Mattoso
    - Congresso: SBBD
    - Ano: Outubro 2010
- Short Paper no Simpósio Brasileiro de Banco de Dados (SBBD) - (SILVA et al., 2012)
    - Titulo: Recomendações para fragmentação horizontal de bases de dados XML
    - Autores: Tatiane Lima da Silva, Fernanda Baião, Jonice de Oliveira Sampaio, Marta Mattoso, Vanessa Braganholo
    - Congresso: SBBD
    - Ano: Outubro 2012
    - Esse Short Paper foi convidado para uma edição especial do Journal of Information and Data Management (JIDM). O artigo foi submetido em janeiro de 2013 e está aguardando julgamento.

Como trabalhos futuros, sugerimos a expansão das heurísticas apresentadas para a fragmentação híbrida, que não foi contemplada nesse trabalho. Além disso, a alocação com redundância não foi explorada nesse trabalho. Seria interessante elaborar uma proposta onde os principais fragmentos fossem replicados em mais de um nó. Outra consideração importante para trabalhos futuros seria a aplicação do plano de execução proposto nessa dissertação para fragmentação horizontal e vertical sobre outro protótipo, a fim de comparar os resultados obtidos. Além disso, principalmente para a fragmentação horizontal executar possíveis experimentos com bases maiores de 1GB para analisarmos se o beneficio da fragmentação horizontal aumenta à medida que a base aumenta. É claro que o fator de alocação máximo de memória das máquinas utilizadas no experimento é um fator que também pode influenciar o resultado.

Outro possível trabalho seria a criação de um algoritmo que incorporasse as heurísticas propostas, permitindo a automatização da fase de análise dentro do projeto de distribuição de dados XML.

REFERÊNCIAS BIBLIOGRÁFICAS

ABITEBOUL, S. ; GOTTLOB, G.; MANNA, M. Distributed XML Design. In: ACM SIMOD-SIGACT-SIGART SYMPOSIUM ON PRINCIPLES OF DATABASE SYSTEMS, 28., 2009. Providence, RI. **Proceedings …** New York: ACM, 2990. p. 247–257.

ANDRADE, A. **PARTIX**: projeto de fragmentação de dados XML. Dissertação 2006. (Mestrado em) - Programa de Engenharia de Sistema e Computação, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

ANDRADE, A. et al. **PartiX:** processing XQuery queries over fragmented XML repositories. Rio de Janeiro: COPPE/UFRJ, 2005. (Technical Report, ES 691).

ANDRADE, A. et al. Efficiently processing XML queries over fragmented repositories with PartiX. In: GRUST, T. (Eds). **Current trends in database technology – EDBT** - EDBT 2006 Workshop PhD, DataX, IIDB, IIHA, ICSNW, QLQP, PIM, PaRMa, and Reactivity on the Web, Munich, 2006. Berlin: Springer, 2006. p. 150–163.

BAIÃO, F. ; MATTOSO, M. A Mixed fragmentation algorithm for distributed object oriented databases. In: INTERNATIONAL CONFERENCE ON COMPUTING AND INFORMATION, 9., 1998, Winnipeg, Canadá, 1998. **Proceedings …** Winnipeg: University of Manitiba/IEEE, 1998. p. 141-148.

BAIÃO, F. ; MATTOSO, M. ; ZAVERUCHA, G. A Distribution design methodology for object DBMS. **Distributed and Parallel Databases**, Boston, v. 16, n. 1, p. 45–90, Jul. 2004.

BARBOSA, D. et al. ToXgene: a template-based data generator for XML. In: SIGMOD 02 – ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 2002, . Madison, Wisconsin. **Proceedings …** New York: ACM, 2002.

BIRHANU, L. ; ATNAFU, S. ; GETAHUN, F. Native XML document fragmentation model. In: SITIS - INTERNATIONAL CONFERENCE ON SIGNAL-IMAGE TECHNOLOGY AND INTERNET-BASED SYSTEMS, 6., 2010, Kuala Lumpur. **Proceedings …** Los Alamitos: IEEE, 2010. p. 233-240.

BONIFATI, A. ; CUZZOCREA, A. Efficient fragmentation of large XML documents. In: WAGNER, R. ; REVELL, N. ; PERNUL, G. (Eds.). **Database and Expert Systems Applications** - 18th International Conference DEXA 2007, Regensburg. Germany, Berlin: Springer, 2007. (Lecture Notes in Computer Science, v.4653).

BREMER, J. ; GERTZ, M. On Distributing XML repositories. In: INTERNATIONAL WORKSHOP ON WEB AND DATABASES, 6., 2003, San Diego. **Proceedings …** New York: ACM, 2003. p.73–78.

BUSSE, R. et al. XMark: a benchmark for XML data management. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASE, 28., 2002, Hong Kong. **Proceedings …** Oelando: Morgan Kaufmann, 2002. p.974–985.

FIGUEIREDO, G. **Processamento de consultas sobre bases XML distribuídas**. 2007. Dissertação (Mestrado em Engenharia de Sistemas) – COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.

FIGUEIREDO, G. ; BRAGANHOLO, V. ; MATTOSO, M. Um mediador para o processamento de consultas sobre bases XML distribuídas. In: BRAZILIAN SYMPOSIUM ON DATABASES, 22., 2007, João Pessoa. **Anais ...** Demos Sesseion. João Pessoa: SBC, 2007. p.21–26.

_____. Processing queries over distributed XML databases. **Journal of Information and Data Management,** São Paulo, v. 1, n. 3, p. 455–470, 2010.

FISCHER, P. ; SINGH, A. ; GRAF, D. **XQBench** - XQuery Benchmark environment. 2012. Disponível em: http://xqbench.org/, Acesso em: 2012.

FLORENTINO, P. **ODARA**: metodologia para projeto de fragmentação de bases de dados. 2003. Dissertação (Mestrado em Engenharia de Sistemas) – CPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003.

FOMICHEV, A. ; GRINEV, M. ; KUZNETSOV, S. Sedna: a native XML DBMS. In: WIEDERMANN, J. et al. (Eds.). **SOFSEM 2006**: theory and practice of computer science. 32nd Conference on Current Trends in Theory and Practice of Computer Science., Merin Czech Republic, 2006 Proceedings. Berlin: Springer, 2006. p. 272–281, (Lecture Notes in Computer Science, 3831.

GERTZ, M. ; BREMER, J. **Distributed XML repositories: top-down design and transparent query processing**. Santa Barbara, CA: University California, Department of Computer Science, 2003. (Technical Report)

HOFFER, A. ; SEVERANCE, D. The use of cluster analysis in physical data base design. In: VLDB - INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 1975. Framingham, MA. **Proceedings …** New York: ACM, 1975. p.69 – 86.

KARLAPALEM, K. ; LI, Q. A Framework for class partitioning in object-oriented databases. **Journal Distributed and Paralell Databases**. Hingham, MA, v. 8, n.3, Jul. 2000.

KLING, P. ; OZSU, M. T. ; DAUDJEE, K. **Distributed XML query processing**: fragmentation, localization and pruning. Waterloo, CA: University of Waterloo, 2010. (Technical Report, CS-2010-02).

_____. Generating efficient execution plans for vertically partitioned XML databases. **Proceedings of the VLDB Endowment**, [S. l.], v. 4, n. 1, p. 1–11, Oct. 2010.

_____. Scaling XML query processing: distribution, localization and pruning. **Distributed and Parallel Databases**, Boston, v. 29, n. 5, p. 445–490, 2011.

KOSSMANN, D. The state of the art in distributed query processing. **ACM Computing Surveys**, New York, v. 32, n. 4, p. 422–469, Dec. 2000.

KURITA, H. et al. Efficient query processing for large XML data in distributed environments. In: INTERNATIONAL CONFERENCE ON ADVANCED NETWORKING AND APPLICATIONS, 21., 2007, Niagara Falls, CA. **Proceedings ...** Washington: IEEE, 2007. p.317–322.

MA, H. ; SCHEWE, K.-D. Fragmentation of XML documents. In: SBBD - SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 18., 2003, Manaus. **Anais …** Manaus: SBC, 2003. p. 200–214.

MA, H. ; SCHEWE, K.-D. Heuristic horizontal XML fragmentation. In: CAiSE - CONFERENCE ON ADVANCED INFORMATION SYSTEMS ENGINEERING, 17., 2005, Porto. **CAiSE Short Paper Proceedings**. Porto: CEUR, 2005. p.131–136. (CEUR Workshop Proceedings 161 CEUR-WS.org 2005).

MCCORMICK, W. ; SCHWEITZER, P. ; WHITE, T. Problem decomposition and data reorganization by a clustering technique. **Operations Research,** Amsterdam, v. 20, n.5, p. 993–1009, Sept./Oct. 1972.

MEIER, W. eXist: an open source native XML database. In: CHAUDHRI, A. et al. (Eds). **Web, web-services, and database systems**, NODe 2002 - Web and Database-Related Workshop , Erfurt, Germany, 2002. London: Springer-Verlag, 2003. p. 169-183, (Lecture Notes in Computer Science, v. 2593).

MIRANDA, B. et al. M. Apuama: combining intra-query and inter-query parallelism in a database cluster. GRUST, T. et al (Eds.). **Current trenin database Technology-EDBT 2006.** EDBT 2006 Workshops PhD, DataX, IIDB, IIHA, ICSNW, QLQP, PIM, PaRMA, and Reactivity on the Web, Munich, Germany, Revised Selected Papers Berlin: Springer-Verlag, 2006. p.649–661. (Lecture Notes in Computer Science, v. 4254).

MORO, M. M. ; BRAGANHOLO, V. ; DORNELES, C. F. et al. XML: some papers in a haystack. **SIGMOD Record**, New York, v. 38, n. 2, p. 29–34, jun. 2009.

NAVATHE, S. ; RA, M. Vertical partitioning for database design: a graphical algorithm. **SIGMOD Record**, New York, v. 18, n. 2, p.440 – 450. Jun. 1989.

OZSU, M. T. ; VALDURIEZ, P. **Principles of distributed database systems**. 3. ed. New York: Springer-Verlag, 2011.

PAES, M. et al. High performance query processing of a real-world OLAP database with ParGRES. In: PALMA, J. M. L. M. et al (Eds). **High performance computing for computational science – VECPAR 2008**. 8th International Conference Toulouse, France, June 2008, Revised selected papers. Berlin: Springer-Verlag, 2008. p. 188-200. (Lecture Notes in Computer Science, v. 5336).

PAGNAMENTA, F. **Design and initial implementation of a distributed xml database**. 2005. Thesis (Master of Science) – Trinity College Dublin, Universidade de Dublin, Dublin, 2005.

SILVA, T. et al. Recomendações para fragmentação horizontal de bases de dados XML. In: SBBD 2012.- SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 2012, São Paulo. **Anais …** São Paulo: SBC, 2012. p.145–152.

SILVA, T. et al. Metodologia para projeto de fragmentação de dados XML sobre bases distribuídas. In: WTDBD 2010 - IX WORKSHOP DE TESES E DISSERTACÕES EM BANCO DE DADOS, 2010, Belo Horizonte. **Anais …** Belo Horizonte: SBC, 2010.

TAVARES, F. **Avaliação do processamento de paralelo de consultas no modelo orientado a objetos.** 1999. Dissertação (Mestrado em Engenharia de Sistema e Computação) -- COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 1999.

YAO, B. B. ; OZSU, M. T. ; KHANDELWAL, N. XBench benchmark and performance testing of XML DBMSs. In: ICDE - IEEE INTERNATIONAL CONFERENCE ON DATA ENGINEERING, 20., 2004, Boston. Proceedings … Los Alamitos: IEEE, 2004. p.621–632.

ANEXOS

ANEXO I: ALGORITMOS

Algoritmo da função mapHierarchy (FLORENTINO, 2003)

```
Function MapHierarchy(
HC: class,
O: set of (OperationId, operationFrequency, operationClassification,
accessedClass),
C: set of (className, classcardinality),)
begin
for each Ci in C do
      if Ci is a subclass of HC then
            for each Oi where C(Oi) = HC do
                  O +=( O.MaxId +1, freqOi, classificationOi, Ci)
            end for
      end if
end for
end
```

Algoritmo da função DefineOwnerMember (FLORENTINO, 2003)

```
Function DefineOwnerMember (
Oi: current operation,
X, Y: classes in the schema accessed by Oi,
Cv: current set of classes to be vertically fragmented,
CDG: class dependency graph)
return (own,mem) : pair of classes in the frag(owner, member) relation

begin
      (own, mem) = null;
      Li = the link (X, Y, name, card) in CDG through which Oi
      navigates from X to Y
      case card of
      "1:1", "1:N":
            if Y is not in Cv then (own, mem) = (X, Y) end if
      "N:1":
            if X is not in Cv then (own, mem) = (Y, X) end if
      "M:N":
            (own, mem) = null
end
```

Algoritmo da função AnalysisPhase (FLORENTINO, 2003)

```
function AnalysisPhase (
C: set of (className, classcardinality),
O: set of (operationId, operationFrequency, operationClassification,
accessedClass),
CDG: class dependency graph)
return Ch : set of pairs (owner, member) of classes to be horizontally
fragmented
Cv: set of classes to be vertically fragmented
Cn: set of classes not to be fragmented
var
fragOwnerMember: set of pairs (owner, member) of classes with no replicas
begin
      Cn = C; fragOwnerMember = empty set
      sort O in descending order according to the operation frequency
      for each (Oi, freqOi, classificationOi, C(Oi)) in O do
            if (Input is not Relational)
                  if ( C(Oi) is an abstract class) then
                        MapHierarchy( C(Oi), O, C) (I)
                  end if
            end if
      case OiOperationClassification of
      "projection":
      if ( C(Oi) is not a member in fragOwnerMember ) and
      ( ((C(Oi), "large") ∈ C) or ((C(Oi), "medium") ∈ C) then
            fragOwnerMember += (C(oi), null)
            Cv += C(Oi) ; Cn -= (C(Oi))
      end if
      "selection":
      if (C(Oi) is not a member in fragOwnerMember ) and
      (Atrib(Oi) is not a Identifier) and (II)
      (Operator(Oi) is not "=" or "≠" ) then (III)
            fragOwnerMember += (C(oi), null)
            Ch += (C(Oi), null) ; Cn -= (C(Oi), cardOi)
      end if
      "navigation":
      for each pair (X,Y) of consecutive classes in C(Oi) do
            if (X is not an abstract class) and (Y is not an abstract
      class) then
                  (own, mem) = DefineOwnerMember (Oi, X, Y, Cv, CDG)
                  if (own, mem) is not null then
                        fragOwnerMember += (own, mem)
                        Ch += (own, mem); Cn -= (own, cardOi); Cn -= (mem,
                  cardM)
                  end if
            end if
      end for
      end case
      end for
      return Ch, Cv, Cn
end
```

ANEXO II: CONSULTAS UTILIZADAS NO EXPERIMENTO (MD)

C1	<results>
	{
	for $order in collection('Orders_c?.xml')/order
	let $l := $order/order_lines/order_line
	where count($l) >= 5
	order by $order/ship_date, $order/@id
	return
	<order>
	{ $order/@id }
	{ $order/ship_date }
	{ $order/total }
	<total_items>
	{ count($l) }
	</total_items>
	</order>
	}
	</results>

C2	<results>
	{
	for $order in collection('Orders_c?.xml')/order
	where $order/@id = "1"
	return
	<order>
	{ $order }
	</order>
	}
	</results>

C3	<results>
	{
	for $a in collection('Orders_c?.xml')/order
	where $a/@id = "3"
	return
	<items>
	{ $a//order_line/item_id }
	</items>
	}
	</results>

C4	<results> { for $a in collection('Orders_c?.xml')/order where $a/@id = "5" return <Output> {$a/order_lines}

	</Output> } </results>

C5	<results>
	{
	for $a in collection('Orders_c?.xml')/order
	where count($a/order_lines/order_line) = 1
	order by $a/@id
	return
	<Output>
	{$a/@id}
	</Output>
	}
	</results>

C6	<results>
	{
	for $a in collection('Orders_c?.xml')/order
	where $a/@id = "6"
	return
	<Output>
	{$a}
	</Output>
	}
	</results>

C7	<results>
	{
	for $order in collection('Orders_c?.xml')/order
	let $l := $order/order_lines/order_line
	where $order/total > 7000
	and count($l) >= 5
	order by $order/ship_date, $order/@id
	return
	<order>
	{ $order/@id }
	{ $order/ship_date }
	{ $order/total }
	<total_items>
	{ count($l) }
	</total_items>
	</order>
	}
	</results>

C8	<results> { for $order in collection('Orders_c?.xml')/order where $order/total > 7000

	order by $order/ship_date, $order/@id return <order> { $order/@id } { $order/ship_date } { $order/total } </order> } </results>

C9	<results>
	{
	for $order in collection('Orders_c?.xml')/order
	where $order/total > 7000
	order by $order/@id
	return
	<order>
	{ $order/@id }
	{ $order/ship_date }
	{ $order/total }
	</order>
	}
	</results>

C10	<results>
	{
	for $order in collection('Orders_c?.xml')/order
	let $l := $order/order_lines/order_line
	where $order/total < 2000
	and count($l) >= 5
	order by $order/ship_date, $order/@id
	return
	<order>
	{ $order/@id }
	{ $order/ship_date }
	{ $order/total }
	<total_items>
	{ count($l) }
	</total_items>
	</order>
	}
	</results>

C11	<results> { for $order in collection('Orders_c?.xml')/order let $l := $order/order_lines/order_line where $order/total > 11000 and count($l) >= 5 order by $order/ship_date, $order/@id

	return <order> { $order/@id } { $order/ship_date } { $order/total } <total_items> { count($l) } </total_items> </order> } </results>

C12	<results>
	{
	for $order in collection('Orders_c?.xml')/order
	where $order/@id="1"
	return
	<customer_id>
	{$order/customer_id}
	</customer_id>
	
	}
	</results>

C13	<results>
	{
	for $a in collection ('Orders_c?.xml')/order
	where $a/total > 11000
	return
	<Output>
	<CustKey>{$a/customer_id}</CustKey>
	<NumberOfOrders>{count($a)}</NumberOfOrders>
	</Output>
	}
	</results>

C14	<results>
	{
	for $a in collection('Orders_c?.xml')/order
	where $a/@id="2"
	return
	<order_line>
	{$a/order_lines/order_line}
	</order_line>
	}
	</results>

C15	<results> { for $a in collection('Orders_c?.xml')/order where $a/total > 11000

	order by $a/ship_type, $a/@id return <Output> {$a/@id} {$a/order_date} {$a/ship_type} </Output> }

C16	<results>
	{
	for $a in collection('Orders_c?.xml')/order
	where $a/total > 11000
	order by $a/total descending, $a/@id
	return
	<Output>
	{$a/@id}
	{$a/order_date}
	{$a/total}
	</Output>
	}
	</results>

C17	<results>
	{
	for $order in collection('Orders_c?.xml')/order
	where $order/total > 10000
	order by $order/ship_date, $order/@id
	return
	<order>
	{ $order/@id }
	{ $order/ship_date }
	{ $order/total }
	</order>
	}
	</results>

C18	<results>
	{
	for $order in collection('Orders_c?.xml')/order
	where $order/total > 10000
	order by $order/@id
	return
	<order>
	{ $order/@id }
	{ $order/ship_date }
	{ $order/total }
	</order>
	}
	</results>

C19	<results>
	{
	for $order in collection('Orders_c?.xml')/order
	where $order/total > 7000
	and $order/total < 8000
	order by $order/@id
	return
	<order>
	{ $order/@id }
	{ $order/ship_date }
	{ $order/total }
	</order>
	}
	</results>

ANEXO III: CONSULTAS UTILIZADAS NO EXPERIMENTO (SD)

C1	<results>
	{
	for $b in collection('auction_c?.xml')/site/people/person
	where $b/@id = "person0"
	return <a>{$b/name}
	}
	</results>

C2	<results>
	{
	for $i in collection('auction_c?.xml')/site/open_auctions/open_auction
	return
	<increase>{$i/bidder/increase}</increase>
	}
	</results>

C3	<results>
	{
	for $b in collection('auction_c?.xml')/site/closed_auctions/closed_auction
	where $b/price >=40
	return <a>{count($b/price)}
	}
	</results>

C4	<results>
	{
	for $b in collection('auction_c?.xml')/site/regions/africa return
	<a>{count($b/item)}
	}
	</results>

C5	<results>
	{
	for $b in collection('auction_c?.xml')/site/regions/australia/item return
	<a>{$b/name} {$b/description}
	}
	</results>

C6	<results>
	{
	for $b in collection('auction_c?.xml')/site
	return <a>{$b/closed_auctions/closed_auction/annotation/description}
	}
	</results>

C7	<results>
	{
	for $b in collection('auction_c?.xml')/site/closed_auctions/closed_auction return
	<a>{$b/seller}
	}

	</results>

C8	<results>
	{
	for $b in collection('auction_c?.xml')/site/people/person return <a>{$b
	/homepage}
	}
	</results>

C9	<results>
	{
	for $b in collection('auction_c?.xml')/site/regions/asia return
	<a>{$b/item/name}
	}
	</results>

C10	<results>
	{
	for $b in collection('auction_c?.xml')/site/people/person
	where $b/profile/@income = "100000"
	return <a>{$b}
	}
	</results>

C11	<results>
	{
	for $b in collection('auction_c?.xml')/site/closed_auctions/closed_auction
	where $b/price > 600
	return <a>{$b/price}
	}
	</results>

C12	<results>
	{
	for $b in collection('auction_c?.xml')/site/regions/namerica
	return <a>{$b/item/mailbox/mail}
	}
	</results>

C13	<results>
	{
	for $b in collection('auction_c?.xml')/site
	return
	<order>
	{count($b/open_auctions/open_auction/annotation/description)}
	{count($b/closed_auctions/closed_auction/annotation/description)}
	{count($b/categories/category/annotation)}
	</order>
	}
	</results>

C14	<results> { for $p in collection('auction_c?.xml')/site

	return
	<item>
	  {$p/people/person/name}
	  {count
	($p/closed_auctions/closed_auction/buyer)}
	</item>
	}

ANEXO IV: TEMPO DE MÉDIO DE RESPOSTA

Resultado dos tempos médios de respostas (4MB) – Múltiplos Documentos

Consulta	Central	c1.1.1	c1.1.2	c2.1.1	c2.1.2	c2.1.3	c2.1.4	c3.1.1	c3.1.2	c3.1.3	c3.1.4
C1	155,78	699,89	641,67	580,11	593,00	457,56	494,33	466,33	447,33	434,00	415,89
C2	121,33	185,56	132,00	129,56	162,78	189,33	220,22	154,00	156,56	252,56	267,89
C3	109,89	131,22	129,89	124,67	153,11	186,33	214,44	126,22	166,44	185,78	204,89
C4	112,11	128,11	140,56	124,78	151,00	544,22	220,33	127,56	147,44	183,22	234,56
C5	180,22	610,00	583,78	530,89	588,67	420,11	466,56	482,67	430,89	351,78	390,11
C6	121,44	126,00	138,44	127,89	157,44	194,11	235,56	129,44	147,22	178,78	271,44
C7	123,44	118,67	280,56	111,78	183,67	176,44	189,44	189,11	193,89	246,22	254,22
C8	130,33	126,44	132,78	120,56	122,78	129,56	129,50	127,89	153,00	187,78	448,11
C9	129,78	125,00	127,67	120,89	126,56	126,78	127,40	128,60	157,11	192,44	218,78
C10	196,22	133,00	132,78	607,78	568,00	433,44	433,78	411,56	323,56	344,11	361,67
C11	127,22	115,40	117,20	114,11	112,00	109,33	105,40	155,89	338,78	241,67	310,89
C12	114,22	123,33	138,89	159,67	160,44	236,89	238,44	124,11	149,78	202,00	275,78
C13	123,11	116,00	118,22	109,22	111,67	113,44	112,40	127,78	152,33	188,89	230,11
C14	114,56	127,22	142,89	130,11	176,44	198,00	238,00	124,56	148,11	185,67	211,33
C15	162,00	123,67	113,89	109,00	119,67	114,11	112,89	129,78	159,89	182,78	229,33
C16	182,44	116,44	110,78	113,11	109,22	108,33	111,44	166,78	160,00	184,89	214,44
C17	122,44	116,11	131,89	117,33	121,67	120,30	119,50	128,22	153,00	185,78	212,56
C18	128,22	115,11	136,00	118,33	126,33	124,44	122,50	125,44	151,44	193,33	214,78
C19	118,78	110,33	115,33	115,44	111,11	113,78	111,67	160,33	159,44	188,44	227,11

Resultado dos tempos médios de respostas (40MB) - Múltiplos Documentos

Consulta	Central	c1.1.1	c1.1.2	c2.1.1	c2.1.2	c2.1.3	c2.1.4	c3.1.1	c3.1.2	c3.1.3	c3.1.4
C1	134,0	710,8	642,2	594,6	617,0	464,3	531,6	488,3	407,3	478,1	395,4
C2	118,6	133,8	133,6	130,9	159,4	246,3	207,1	131,4	151,7	200,6	237,0
C3	112,6	120,2	253,0	126,0	152,4	187,2	316,1	132,0	173,0	171,1	279,7
C4	113,0	122,1	135,4	184,4	151,1	264,3	242,0	128,6	224,7	180,9	224,7
C5	123,0	595,1	672,7	523,2	550,7	421,3	551,0	445,3	416,1	351,7	418,4
C6	121,1	129,4	135,9	127,8	176,2	228,7	231,0	133,6	158,8	175,4	251,4
C7	120,8	118,0	119,4	114,9	181,4	176,4	185,7	183,3	194,0	233,2	299,8
C8	129,9	127,9	127,4	124,2	130,9	133,8	145,7	130,3	150,9	199,7	329,1
C9	131,8	124,7	126,8	129,3	132,7	133,4	147,2	152,0	158,3	192,6	247,0
C10	461,7	128,8	129,4	667,1	567,4	424	428,4	413,6	449,0	363,2	432,1
C11	124,7	118,0	115,9	113,3	108,2	111,8	112,1	280,0	199,9	251,8	562,3
C12	120,1	121,8	159,7	129,1	160,3	266,6	274,8	123,7	188,4	170,8	243,6
C13	122,9	116,1	114,2	112,1	110,9	112,3	114,1	128,7	151,6	189,9	222,6
C14	116,7	119,4	136,7	144,3	164,4	184,3	227,3	129,9	145,6	232,9	242,3
C15	123,3	110,3	111,9	107,8	109,7	112,1	110,7	127,3	152,9	186,0	235,1
C16	126,1	118,3	123,1	114,1	110,4	109,8	117,1	128,4	162,1	186,3	230,3
C17	128,1	115,1	114,1	118,0	132,0	129,8	128,6	141,2	222,1	178,7	272,7
C18	127,1	123,8	124,6	117,0	132,1	127,0	128,0	128,7	157,4	210,4	260,1

C19	121,9	112,3	111,4	113,6	110,9	109,4	110,9	122,9	154,0	204,2	231,3

Resultado dos tempos médios de respostas (400MB) - Múltiplos Documentos

Consulta	Central	c1.1.1	c1.1.2	c2.1.1	c2.1.2	c2.1.3	c2.1.4	c3.1.1	c3.1.2	c3.1.3	c3.1.4
C1	7383,6	45396,7	43806,3	78586,4	38384,6	38709,1	27401,6	43508,1	27960,7	25119,8	19855,0
C2	551,1	387,6	361,0	567,2	387,9	359,2	344,1	362,1	294,0	301,3	314,3
C3	538,6	367,8	357,9	551,1	355,2	381,3	374,6	413,9	347,4	361,2	316,2
C4	555,0	409,7	353,3	571,1	371,7	361,9	357,0	369,1	304,2	326,3	310,0
C5	1371,6	41899,1	36273,0	75237,1	35655,4	34452,2	24177,7	39213,6	25855,0	22657,9	17856,9
C6	552,9	407,8	369,9	577,9	354,2	403,8	349,3	372,4	316,6	345,7	330,7
C7	4628,8	2000,6	22680,1	4116,3	22812,6	22457,2	20917,9	22938,2	15223,4	13365,6	10360,2
C8	28039,4	6754,0	3184,4	23895,0	3259,7	3163,8	3061,7	3383,8	2144,8	2104,2	1598,6
C9	26317,9	6481,3	3221,9	23202,4	3194,3	3118,2	2991,2	3324,4	2153,3	2021,1	1594,4
C10	1958,0	901,4	868,0	6387,0	5395,1	5546,8	5422,2	5915,7	3758,0	3948,0	3078,3
C11	2889,6	1421,7	971,0	2579,2	961,4	958,2	945,9	11197,2	7053,1	6680,8	5193,4
C12	538,8	366,3	359,8	528,2	372,9	370,3	355,1	392,7	294,1	334,7	366,4
C13	3596,4	3190,9	3107,7	3547,8	3073,4	3038,9	3058,3	1248,7	880,4	860,1	763,9
C14	545,8	369,6	345,3	554,9	356,1	359,1	365,6	375,0	283,6	323,2	321,2
C15	13403,1	3567,0	3452,2	11416,6	3204,3	3244,7	3164,8	1964,7	1287,7	1232,4	1030,7
C16	13206,6	3443,4	3312,4	11267,1	3045,7	3053,7	3003,9	1921,9	1195,9	1180,9	1000,2
C17	16990,6	4321,2	3023,1	14429,6	2888,8	2808,6	2843,9	2264,9	1438,3	1382,2	1141,1
C18	16386,1	4037,0	2955,0	14125,7	2897,3	2886,0	2767,0	2332,4	1440,7	1361,7	1110,8
C19	3903,3	1307,7	1041,4	3391,6	1037,1	992,7	871,1	918,9	599,9	655,3	550,4

Resultado dos tempos médios de respostas (10MB) – Único Documento

Consulta	Central	c1.1	c2.1	c2.2	c2.3	c3.1	c3.2
C1	162,78	187,67	112,56	143,00	142,78	145,00	107,00
C2	159,33	160,78	146,33	153,89	147,44	153,78	153,00
C3	102,44	114,11	105,67	114,11	166,67	107,11	112,44
C4	100,89	181,78	107,89	97,67	173,67	142,56	196,33
C5	165,33	169,33	159,22	178,33	186,56	165,67	168,44
C6	300,56	291,89	289,78	293,22	290,89	293,00	292,78
C7	120,44	104,89	149,22	110,78	112,44	116,33	114,78
C8	168,56	193,00	150,00	118,78	116,67	156,00	116,44
C9	119,67	127,89	124,67	124,44	106,11	151,56	129,44
C10	137,00	126,00	161,89	105,56	183,44	134,78	106,44
C11	196,67	104,78	108,56	142,22	109,89	118,11	110,00
C12	248,22	228,44	226,78	228,56	232,44	228,22	226,00
C13	99,78	592,67	99,22	605,22	611,56	703,67	481,33
C14	207,11	296,00	143,00	141,44	259,89	220,11	245,44

Resultado dos tempos médios de respostas (100MB) – Único Documento

Consulta	Central	c11	c21	c22	c23	c31	c32
C1	137,11	155,56	144,22	152,00	146,11	133,89	102,33
C2	678,56	169,56	682,89	147,33	176,22	676,89	692,56
C3	189,22	110,11	207,11	230,00	105,11	194,00	207,00
C4	135,44	102,44	100,56	101,44	101,22	101,22	201,22

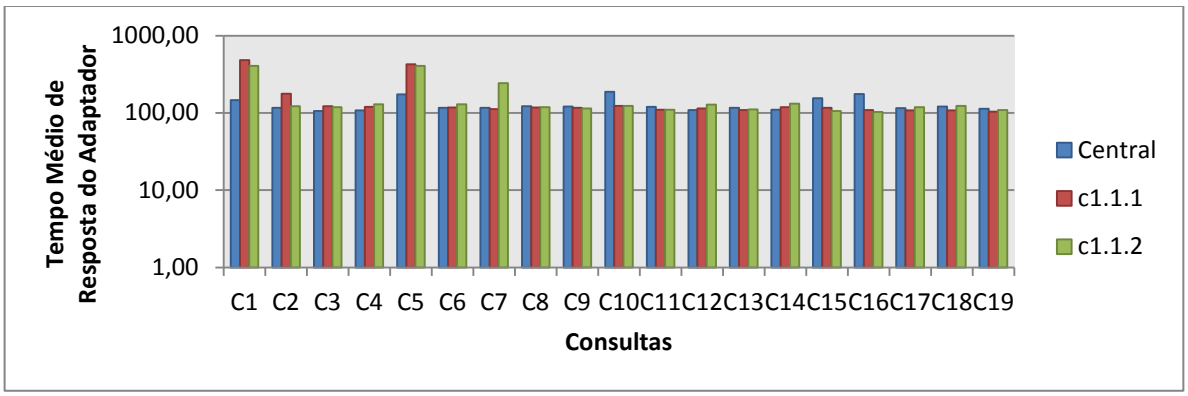
C5	608,67	604,78	629,67	595,89	600,67	637,22	603,67
C6	2301,11	291,33	2326,33	2269,11	283,89	2339,56	2271,44
C7	248,44	109,00	250,89	344,33	110,89	251,78	248,00
C8	373,22	394,89	408,44	859,44	378,56	376,11	132,22
C9	107,33	113,00	115,11	127,11	130,78	115,56	119,67
C10	136,33	135,56	156,78	153,22	148,33	164,33	95,33
C11	122,67	108,33	118,33	154,67	108,78	119,00	115,22
C12	1557,33	1531,33	1540,33	1553,33	1522,11	1546,33	1522,56
C13	127,44	584,22	119,89	2456,11	691,33	5640,22	3529,22
C14	317,33	1686,44	319,89	319,11	383,22	348,44	926,78

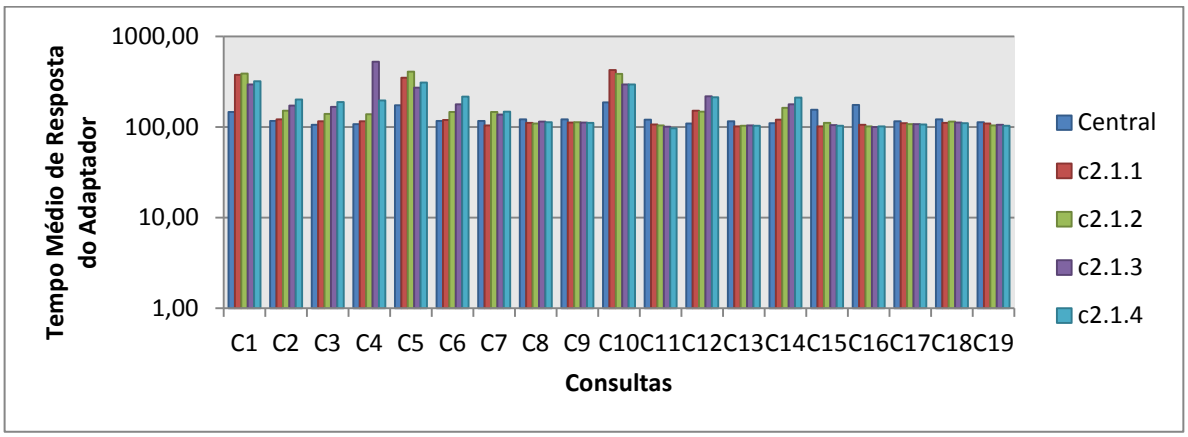
Resultado dos tempos médios de respostas (1GB) – Único Documento

Consulta	Central	1.1	2.1	2.2	2.3	3.1	3.2
C1	587,06	541,11	1742,72	538,22	548,33	555,83	108,78
C2	7053,00	172,06	6986,78	153,39	170,94	8187,17	7556,72
C3	1083,11	117,00	1068,33	1075,56	112,11	1055,67	1128,78
C4	99,94	110,00	106,61	111,28	105,89	95,94	119,00
C5	6350,06	5917,33	5834,50	5755,00	5731,56	5701,06	5722,72
C6	27377,22	306,11	27282,00	29222,94	289,61	31280,11	29291,11
C7	2201,78	120,06	1946,28	1968,72	242,72	1882,00	1898,22
C8	3451,83	3373,94	3428,44	3425,39	3428,56	3751,50	123,44
C9	347,39	307,06	330,67	309,94	357,06	341,44	402,28
C10	585,83	502,06	494,22	500,17	498,39	497,78	106,61
C11	319,78	118,39	334,28	327,17	100,67	338,17	342,00
C12	17030,39	17654,78	17554,06	17563,22	16870,78	16913,33	17415,00
C13	314,78	593,83	302,10	34552,61	624,94	82180,11	59251,17
C14	3744,11	23392,17	5716,67	3230,39	3783,22	2700,83	17782,00

ANEXO V: TEMPO DE RESPOSTA DO ADAPTADOR

GRÁFICO 4MB: TEMPO MÉDIO DE RESPOSTA DO ADAPTADOR (FRAGMENTAÇÃO HORIZONTAL) – Escala Logarítmica





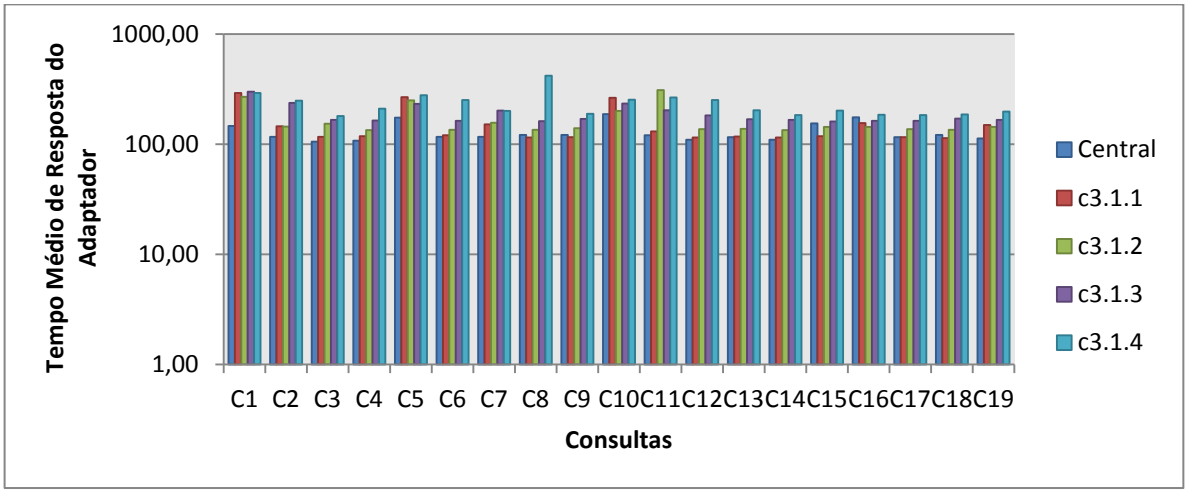
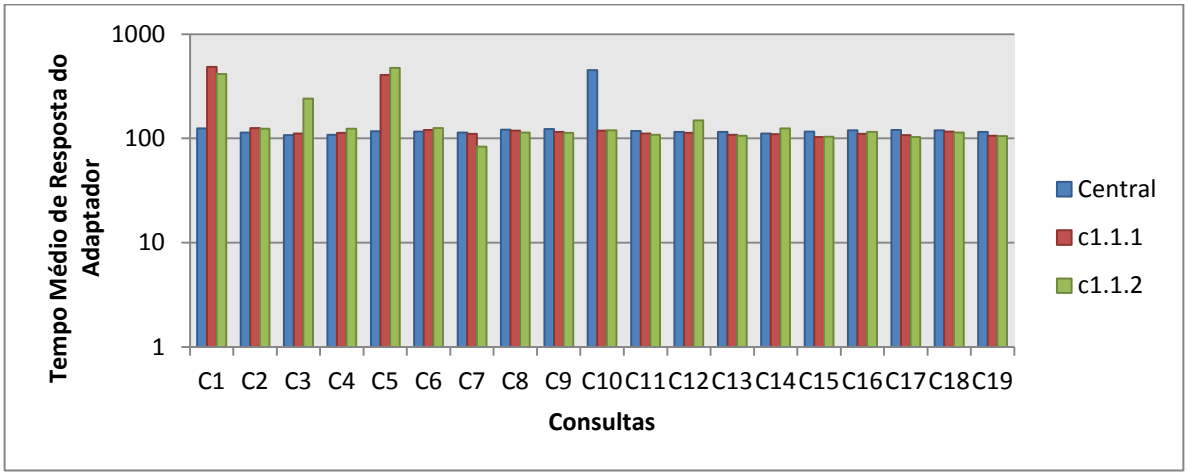
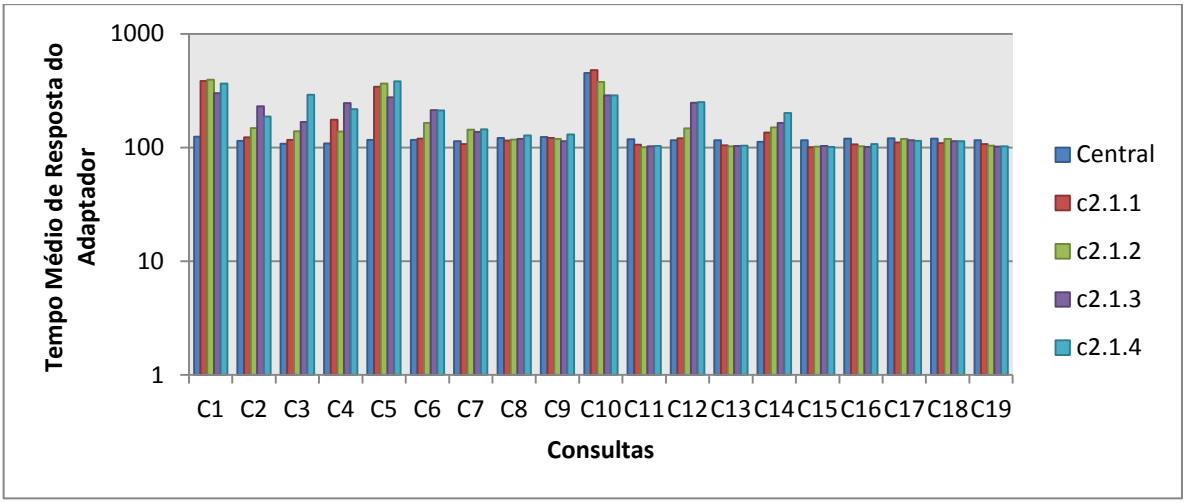


GRÁFICO 40MB: TEMPO MÉDIO DE RESPOSTA DO ADAPTADOR (FRAGMENTAÇÃO HORIZONTAL) – Escala Logarítmica





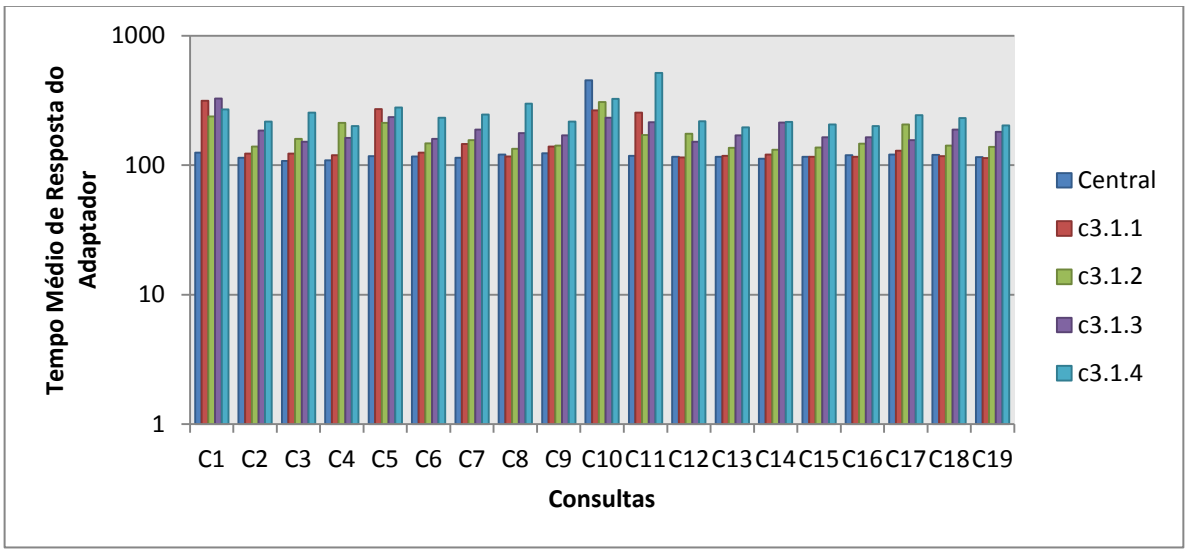
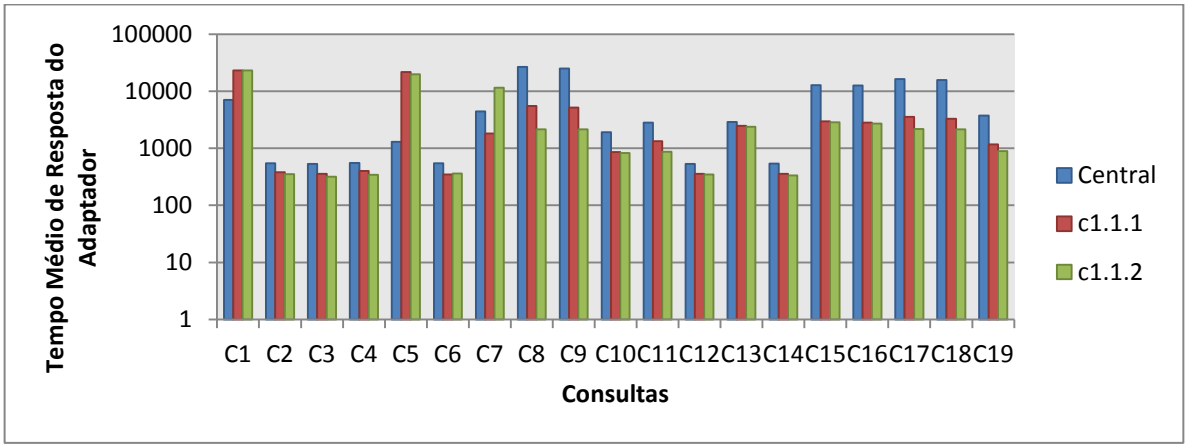
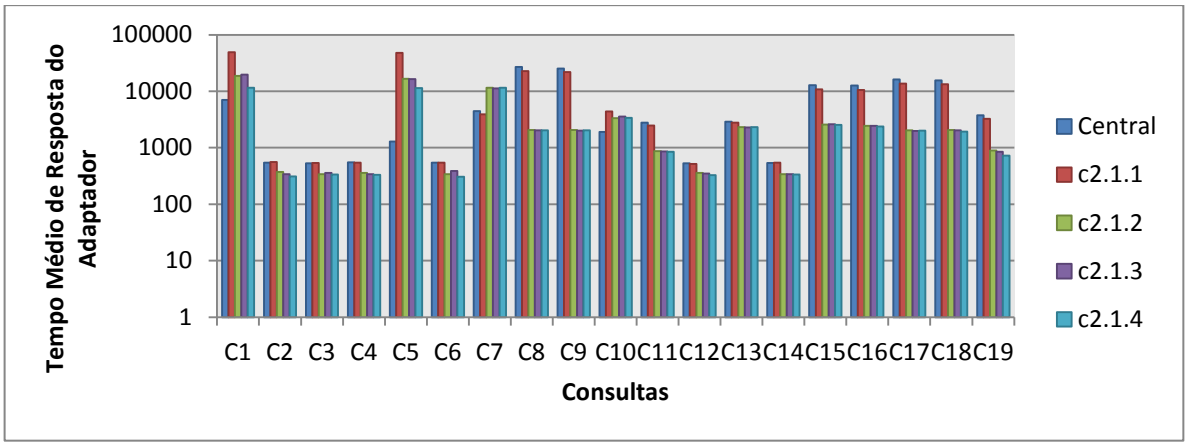


GRÁFICO 400MB: TEMPO MÉDIO DE RESPOSTA DO ADAPTADOR (FRAGMENTAÇÃO HORIZONTAL) – Escala Logarítmica





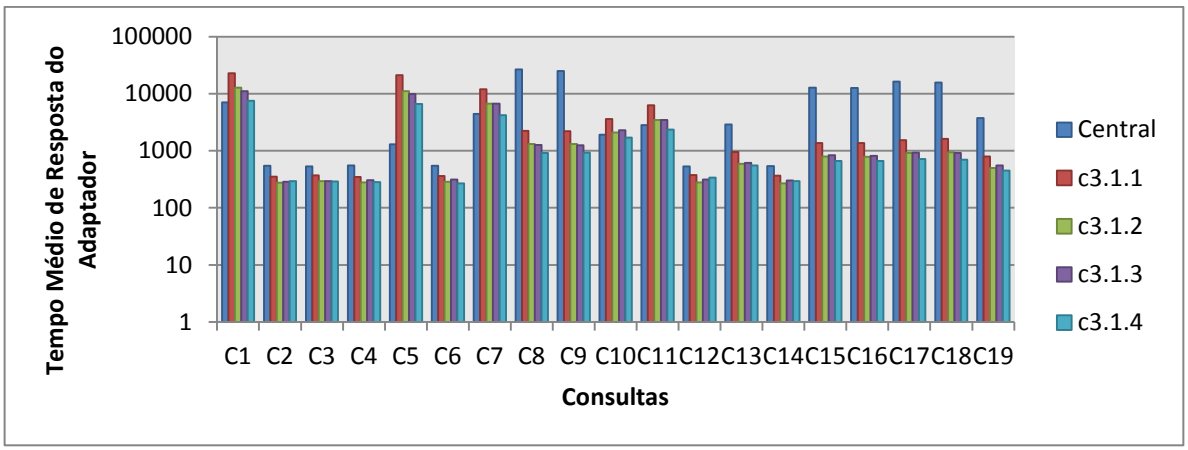
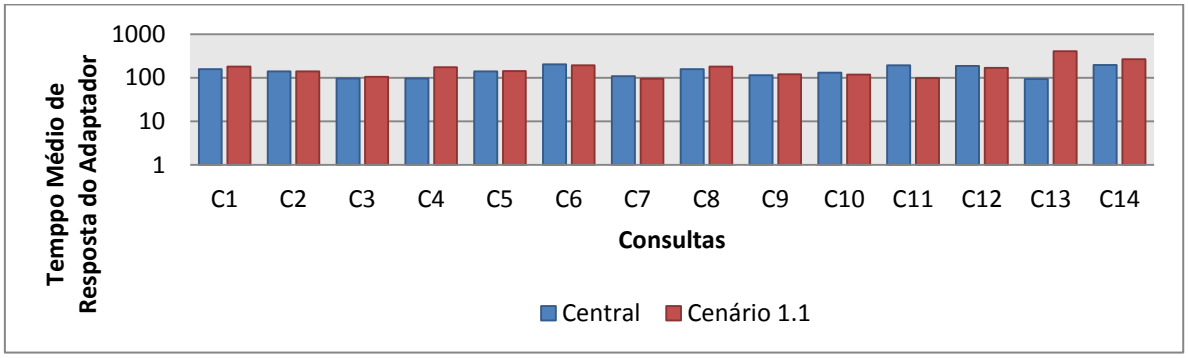
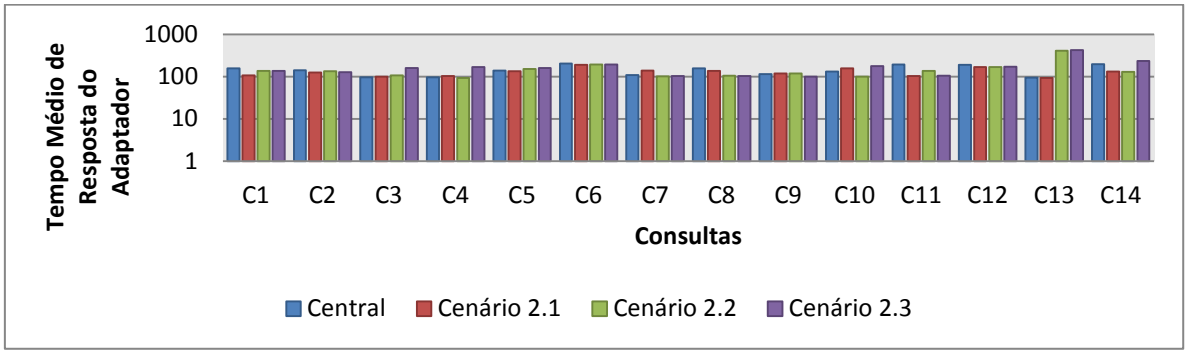


GRÁFICO 10MB: TEMPO MÉDIO DE RESPOSTA DO ADAPTADOR (FRAGMENTAÇÃO VERTICAL) – Escala Logarítmica





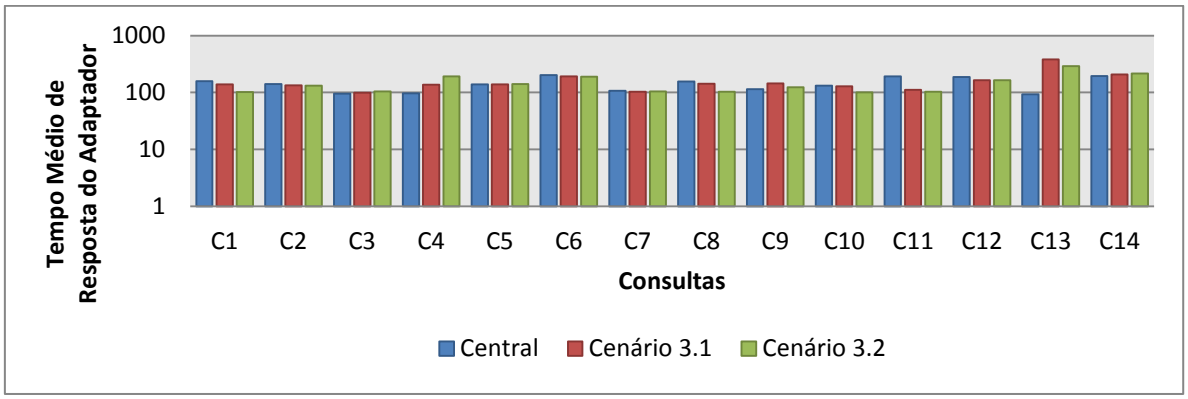
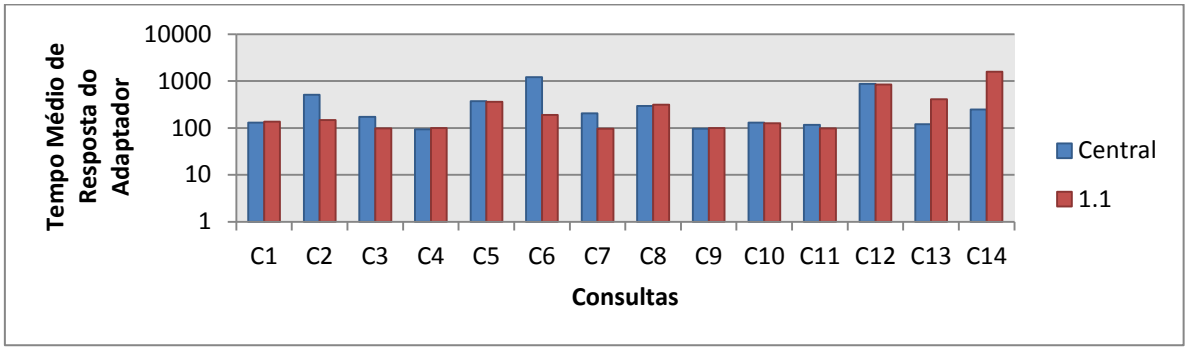
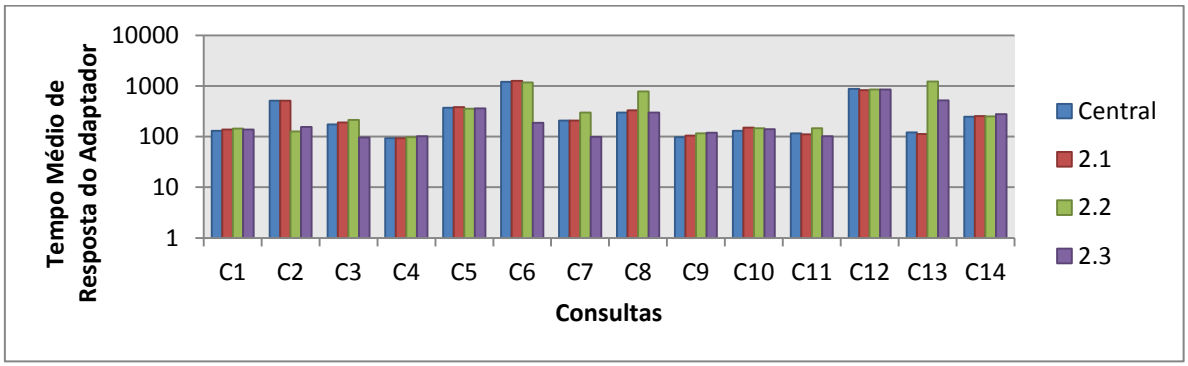


GRÁFICO 100MB: TEMPO MÉDIO DE RESPOSTA DO ADAPTADOR (FRAGMENTAÇÃO VERTICAL) – Escala Logarítmica





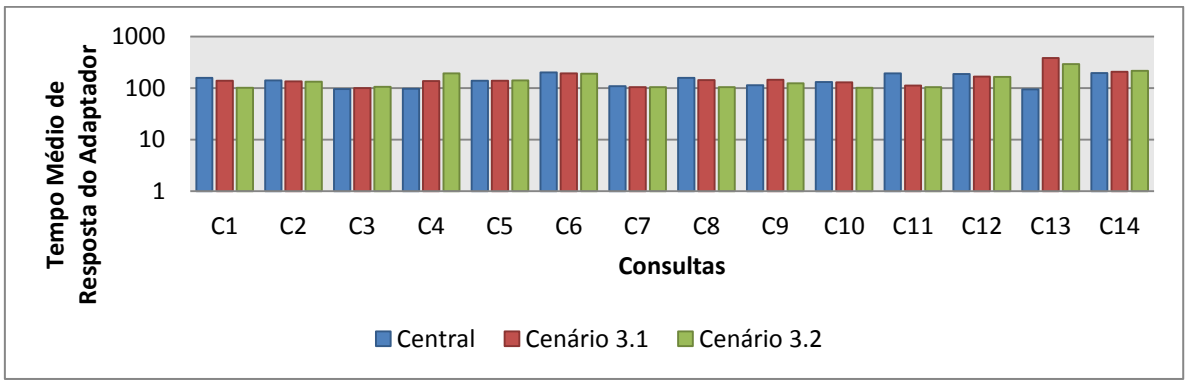
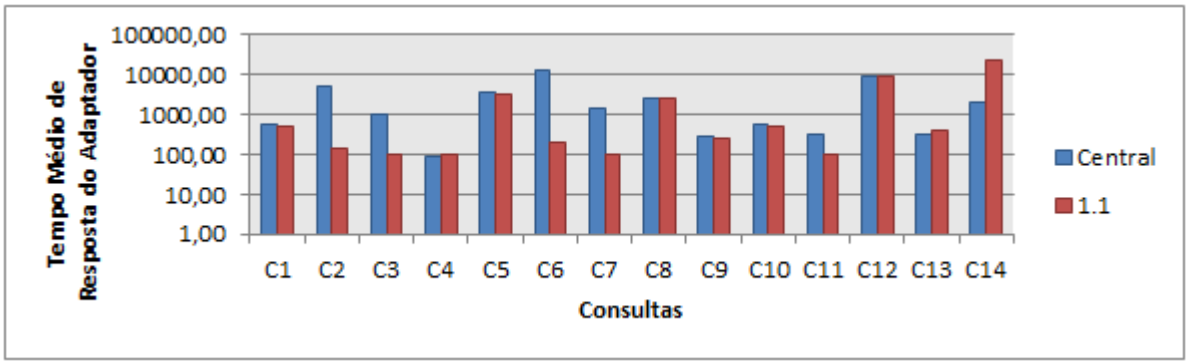
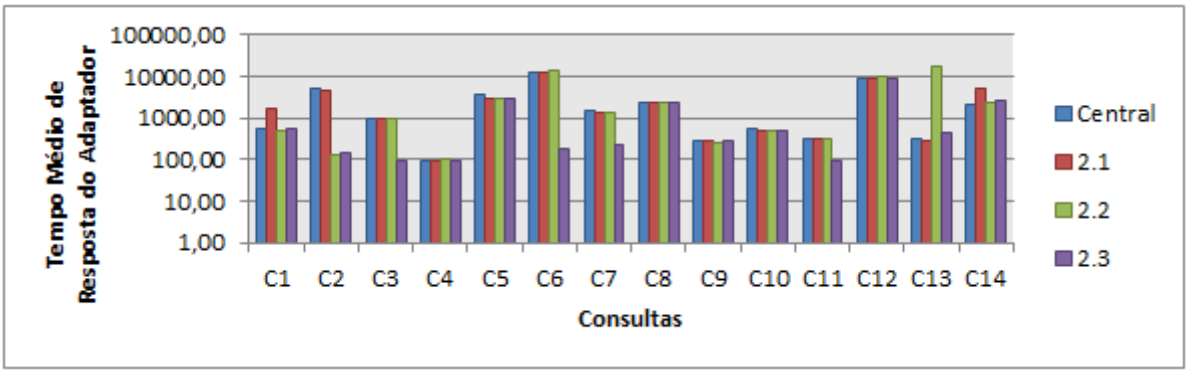
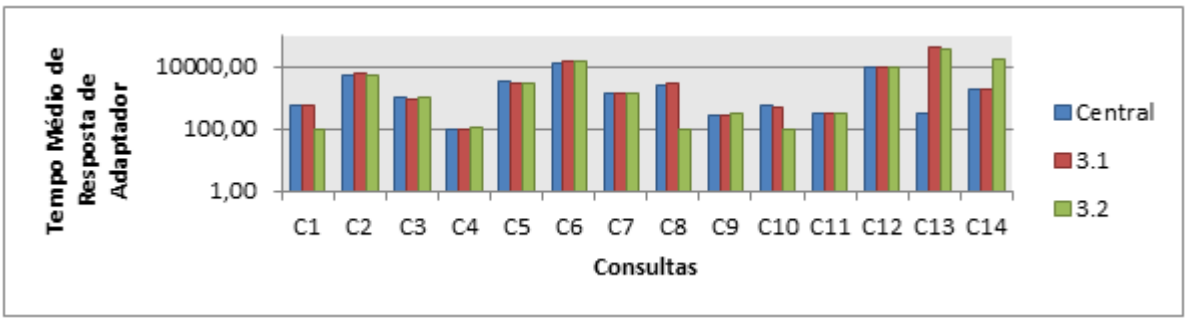


GRÁFICO 1GB: TEMPO MÉDIO DE RESPOSTA DO ADAPTADOR (FRAGMENTAÇÃO VERTICAL) – Escala Logarítmica







UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

CCMN - Bloco C - Cidade Universitária - Ilha do Fundão
Rio de Janeiro - RJ CEP: 21941-916
www.ppgi.ufrj.br