

Definição de diretrizes para fragmentação horizontal de bases de dados XML

Tatiane Lima da Silva¹, Fernanda Baião², Jonice de Oliveira Sampaio¹, Marta Mattoso³, Vanessa Braganholo⁴

¹PPGI/Federal University of Rio de Janeiro, Brazil

²NP2Tec/Federal University of the State of Rio de Janeiro (UNIRIO), Brazil

³COPPE/Federal University of Rio de Janeiro, Brazil

⁴Fluminense Federal University, Brazil

tatiane.lima@ppgi.ufrj.br, fernanda.baiao@uniriotec.br, jonice@dcc.ufrj.br
marta@cos.ufrj.br, vanessa@ic.uff.br

Resumo. 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, manipulação de dados de múltiplas aplicações, entre outros. Este fato faz surgir a necessidade de desenvolvimento de metodologias para processamento eficiente de consultas sobre dados XML. 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. Sendo assim, o objetivo desse trabalho é apresentar experimentos realizados para definição de futuras diretrizes para o processo de fragmentação horizontal de dados XML.

Categories and Subject Descriptors: H. Information Systems [H.2. Database Management]: H.2.4 Systems — Distributed Databases

Keywords: heuristics, database fragmentation design, horizontal fragmentation, XML

1. INTRODUÇÃO

Devido ao grande volume de dados predominantemente armazenado em bancos de dados XML, há uma grande preocupação com o desempenho no processamento de consultas em tais ambientes e, consequentemente, inúmeros estudos nesta área [Andrade *et al.* 2006; Figueiredo *et al.* 2010; Kling *et al.* 2011]. Surge então a necessidade da aplicação de técnicas que permitam consultas em bancos de dados de forma mais eficiente. Neste sentido, técnicas de distribuição de dados e processamento paralelo de consultas sobre bases de dados têm sido, há muito tempo, adotadas com muito sucesso. Nesta abordagem, os dados devem ser distribuídos pelos diferentes nós de uma rede segundo técnicas de fragmentação e de alocação [Kling *et al.* 2010, 2011; Ozsu and Valduriez 2011].

Existem duas formas distintas de fragmentação: fragmentação física e fragmentação virtual. A fragmentação física [Ozsu and Valduriez 2011] particiona os dados fisicamente, e os aloca em diferentes nós. Já a fragmentação virtual [Lima *et al.* 2010; Rodrigues *et al.* 2011] exige que os dados sejam replicados nos nós da rede, exigindo muito espaço em disco. No que diz respeito à fragmentação física dos dados (foco do presente trabalho), o potencial de ganho de desempenho é obtido em função da localidade dos dados, quando 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 que não existiam na consulta original, entre outros motivos [Figueiredo *et al.* 2010]. Por isso, o projeto da fragmentação da base de dados precisa analisar as consultas mais frequentes para que a fragmentação apresente ganho de desempenho na maioria das consultas realizadas sobre a base de dados.

Aproveitando as ideias de fragmentação e distribuição propostas para o modelo relacional [Ozsu and Valduriez 2011] e orientado a objetos [Baião *et al.* 2004], vários trabalhos na literatura têm focado em processamento de consultas XML em ambientes distribuídos e na criação de técnicas de fragmentação, endereçando aspectos específicos como o formato dos fragmentos e os algoritmos que os geram [Gertz and Bremer 2003; Abiteboul *et al.* 2009]. No entanto, não existe na literatura nenhuma metodologia para projeto de distribuição de dados XML que analise quais técnicas de fragmentação devem ser aplicadas em cada cenário, o que impacta de forma determinante o desempenho das aplicações sobre a base de dados distribuída. As propostas existentes assumem que o projetista já sabe de que forma a base deve ser fragmentada [Gertz and Bremer 2003; Abiteboul *et al.* 2009; Ozsu and Valduriez 2011].

De fato, no panorama dos modelos de distribuição em XML, um dos pontos mais explorados na literatura é justamente a definição do que é um fragmento XML [Bremer and Gertz 2003; Ma and Schewe 2003; Andrade *et al.* 2006] e como consultas podem ser processadas sobre bases XML distribuídas e fragmentadas [Figueiredo *et al.* 2010], enquanto que o projeto de fragmentação de dados XML ainda é um ponto pouco explorado. Conforme discutido por Figueiredo *et al.* [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.

O projeto de fragmentação pode ser dividido em três etapas [Ozsu and Valduriez 2011]: (i) análise, onde são avaliadas informações da aplicação (consultas frequentes) e do esquema do banco de dados para decidir qual o tipo de fragmentação a ser aplicada; (ii) extração de dados relevantes; e (iii) fragmentação propriamente dita. Desta forma, o objetivo deste trabalho é auxiliar no projeto de fragmentação de bases de dados XML. Em especial, nosso foco está na etapa de análise. Para isso, este artigo define diretrizes para fragmentação horizontal para bases de dados XML. As diretrizes são derivadas a partir de uma análise experimental que avaliou diversas alternativas de fragmentação. O foco em fragmentação horizontal é um primeiro passo para solucionar a problemática geral da fase de análise em projeto de fragmentação de dados XML.

O restante desse relatório está estruturado como segue. A seção 2 apresenta os conceitos relacionados à fragmentação horizontal de dados XML que são utilizados em nossa análise, além de uma discussão de metodologias e algoritmos que existem atualmente na literatura para projeto de fragmentação horizontal de dados XML. A seção 3 descreve detalhes sobre a preparação dos experimentos, tais como a descrição das bases, os servidores utilizados, definição dos fragmentos e os planos de execução do experimento. Na seção 4, são apresentadas as consultas executadas sobre as bases fragmentadas e os resultados das análises. As diretrizes para o projeto de fragmentação horizontal de dados XML são apresentadas na seção 5 e, por último, as conclusões obtidas a partir dos experimentos.

2. FRAGMENTAÇÃO HORIZONTAL DE BASES DE DADOS XML

Existem diversos trabalhos na literatura que apresentam definições de fragmentos XML [Bremer and Gertz 2003; Andrade *et al.* 2006; Kling *et al.* 2010] e para o projeto de fragmentação para XML em geral [Gertz and Bremer 2003; Ma and Schewe 2003; Pagnamenta 2005]. 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. Esse tipo de deficiência não permite definir um método decisório consistente quanto ao tipo de fragmentação mais aplicável em cada cenário, fazendo com que a fragmentação seja *ad-hoc*, baseada tipicamente na experiência dos projetistas.

Na arquitetura para banco de dados XML distribuídos proposta por Pagnamenta [2005], a abordagem para distribuição de documentos utiliza noções de fragmentação horizontal e vertical. No entanto, 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.

O discurso apresentado por Ma e Schewe [2003] ressalta a importância da consideração das consultas frequentes na definição dos fragmentos. Além disso, em seu trabalho que descreve as heurísticas para fragmentação de dados horizontal, Ma e Schewe [2003] apresentam uma solução baseada em um modelo de custos, onde o maior ofensor da fragmentação horizontal de dados XML é o tempo de transporte dos resultados locais. Entretanto, em sua abordagem não foi mencionado nenhum resultado experimental que comprovasse a eficiência das heurísticas propostas, ficando apenas nas formalizações teóricas.

Na definição de fragmentação de dados XML, nosso trabalho utiliza o conceito proposto por Andrade et al. [2006], pois das definições encontradas na literatura [Bremer and Gertz 2003; Ma and Schewe 2003; Pagnamenta 2005; Kling *et al.* 2011] essa é a que mais se aproxima da definição de fragmentos do modelo relacional [Ozsu and Valduriez 2011]. Essa escolha é essencial, já que desejamos aproveitar as ideias de projeto de fragmentação propostas para o modelo relacional [Ozsu and Valduriez 2011], pelo fato do seu conceito estar bem consolidado na literatura. Andrade et al. [2006] definem três tipos de fragmentos: horizontal, usando predicados de seleção para permitir a separação de documentos em diferentes fragmentos; vertical, que altera a estrutura de dados através de projeções, e, finalmente; um híbrido que combina as operações de seleção e projeção.

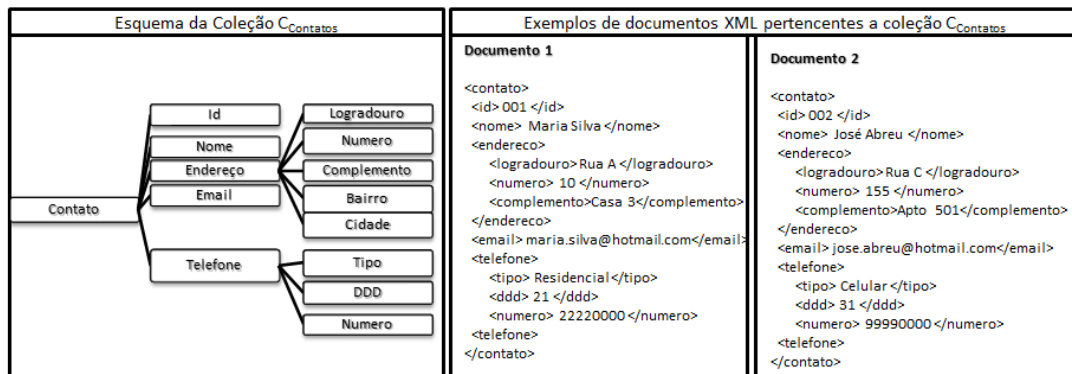


Fig. 1. Exemplo de esquema e documentos XML

Andrade et al. [2006] definem um fragmento horizontal da seguinte forma. Seja μ uma conjunção de predicados simples sobre a coleção C . O fragmento horizontal F de C definido por μ é dado pela expressão $F := \langle C, \sigma_\mu \rangle$, 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. Neste tipo de fragmentação é preciso que a coleção seja do tipo MD (múltiplos documentos), ou seja, a fragmentação horizontal não pode ser aplicada a coleções de um único documento. Este tipo de coleção pode sofrer uma fragmentação híbrida, onde primeiro se aplica uma fragmentação vertical e em seguida uma horizontal.

Para exemplificar o funcionamento da fragmentação horizontal, suponha que temos um esquema referente a contatos de uma agenda e dois documentos XML que seguem o esquema de Contatos. Os dois documentos estão armazenados na Coleção C_{Contatos} . A Figura 1 mostra essas especificações. A Figura 2 apresenta a especificação de uma possível fragmentação horizontal da coleção C_{Contatos} da Figura 1, supondo que as consultas frequentemente usam o elemento *tipo*. O fragmento $F1_{\text{Residencial}}$ reúne os documentos da coleção C_{Contatos} que possuem conteúdo do elemento *tipo* igual a *Residencial*. Por isso, o documento 1 pertence a este fragmento. Já o fragmento $F2_{\text{Residencial}}$ agrupa os documentos cujo conteúdo de *tipo* difere de *Residencial*. Logo, o documento 2 que possui *tipo* igual a *Celular* irá compor o fragmento $F2_{\text{Residencial}}$.

$$\begin{aligned}
 F1_{\text{Residencial}} &:= \langle C_{\text{Contatos}}, \sigma_{\text{contato/telefone/tipo}='Residencial'} \rangle \\
 F2_{\text{Residencial}} &:= \langle C_{\text{Contatos}}, \sigma_{\text{contato/telefone/tipo} \neq 'Residencial'} \rangle
 \end{aligned}$$

Fig. 2. Exemplo de definição de fragmentos sobre a coleção C_{Contatos}

Com base na definição de fragmentos proposta por Andrade et al.[2006], Figueiredo et al. [2010] desenvolveram uma metodologia para processamento de consultas sobre bases de dados XML fragmentadas e distribuídas. O protótipo desenvolvido por Figueiredo et al. [2010] se encarrega de distribuir a consulta aos fragmentos relevantes. Resumidamente, o protótipo inclui um mediador, que é responsável por todo o processamento da consulta, desde a decomposição até a consolidação dos resultados. Cada nó da rede, por sua vez, possui um Componente, que recebe as subconsultas enviadas pelo mediador e as executa no nó local. Um exemplo do protótipo utilizado é descrito na Figura 3, onde temos um nó atuando como mediador e os demais como componentes.

3. PREPARAÇÃO DOS EXPERIMENTOS

Para derivar as diretrizes para fragmentação horizontal, realizamos uma série de experimentos visando avaliar o comportamento do tempo de execução de consultas usando diferentes alternativas de fragmentação horizontal. 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 executados, 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.

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 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 diretrizes que permitam fragmentar qualquer base XML a fim de diminuir o tempo de respostas 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 subseção 2.1 e a elaboração da metodologia de execução apresentada na seção 2.2.

3.1 Ambiente

Os experimentos foram executados em um cluster homogêneo composto de 42 máquinas, cada uma com dois processadores Intel Xeon quadcore (8 cores). No experimento, usamos nove nós do cluster. Cada nó possui 16 GB memória de RAM e disco rígido local de 160 GB. Um deles atuou como Mediador, que é responsável pela submissão das consultas, geração das subconsultas e consolidação dos resultados. Uma instância do componente 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. Essas execuções são realizadas sobre um banco de dados XML nativo Sedna [Fomichev *et al.*

2006]. A Figura 3 apresenta a distribuição das aplicações nos nós do cluster, onde 1 nó funciona como mediador e os demais 8 nós atuam como Adaptadores locais.

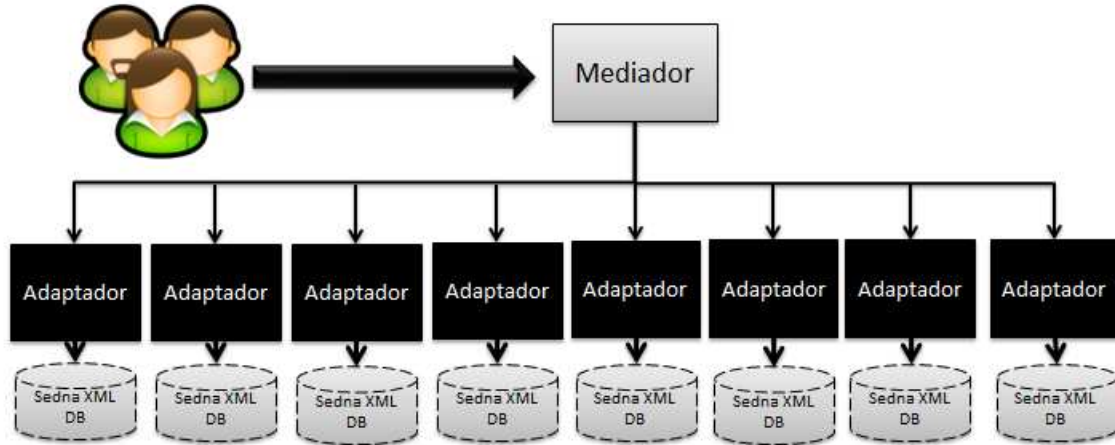


Fig. 3. Ambiente para a execução do experimento

3.2 Metodologia de execução

A metodologia de execução dos experimentos neste trabalho consiste na execução de consultas *XQuery* repetidamente, num total de 10 vezes cada, sobre as bases de dados em três cenários 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.

Para construção das diretrizes foi definida uma sequência de etapas que são apresentadas na Figura 4. A seguir temos o detalhamento de cada uma dessas etapas.

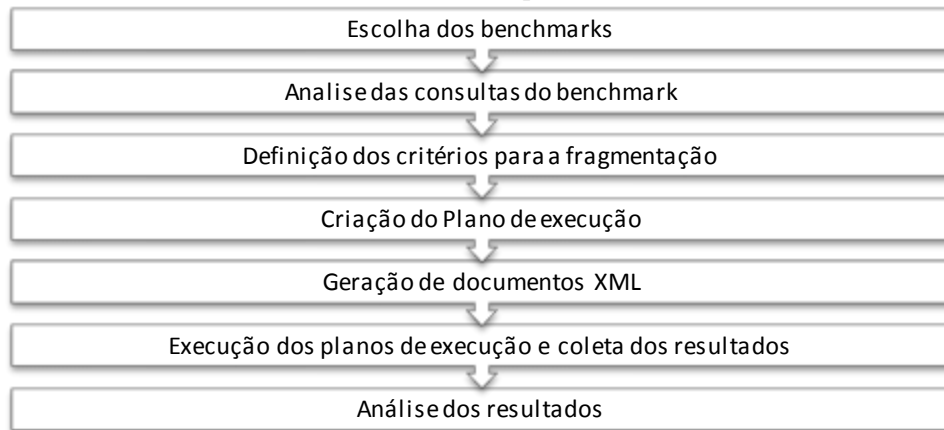


Fig. 4. Etapas do processo de definição da diretrizes para fase de análise

Escolha dos benchmarks. Para efetuar análise sobre consultas executadas em nosso experimento foi escolhido o XBench [Yao *et al.* 2004], um dos *benchmarks* mais conceituados para XML. O *benchmark* XBench é bem amplo, possuindo diversos modelos e consultas aos seus dados. Nesse experimento utilizamos um dos seus tipos de base de dados [Yao *et al.* 2004]: *Multiple Document* (MD). A base MD possui 5 esquemas de documento XML: *Customer* (Single Document), *Item* (SD), *Author* (SD), *Address* (SD), *Country* (SD) e *Orders* (MD). Como o objetivo desse experimento é analisar o comportamento para a fragmentação horizontal, utilizamos o esquema *Orders* que possui variações de 2.592, 25.920,

259.200 e 2.592.000 documentos, dependendo do fator de tamanho da base de dados escolhido no momento da geração da base de dados. A Figura 5 apresenta o esquema de dados da base *Orders* (MD) que é utilizado nesse trabalho.

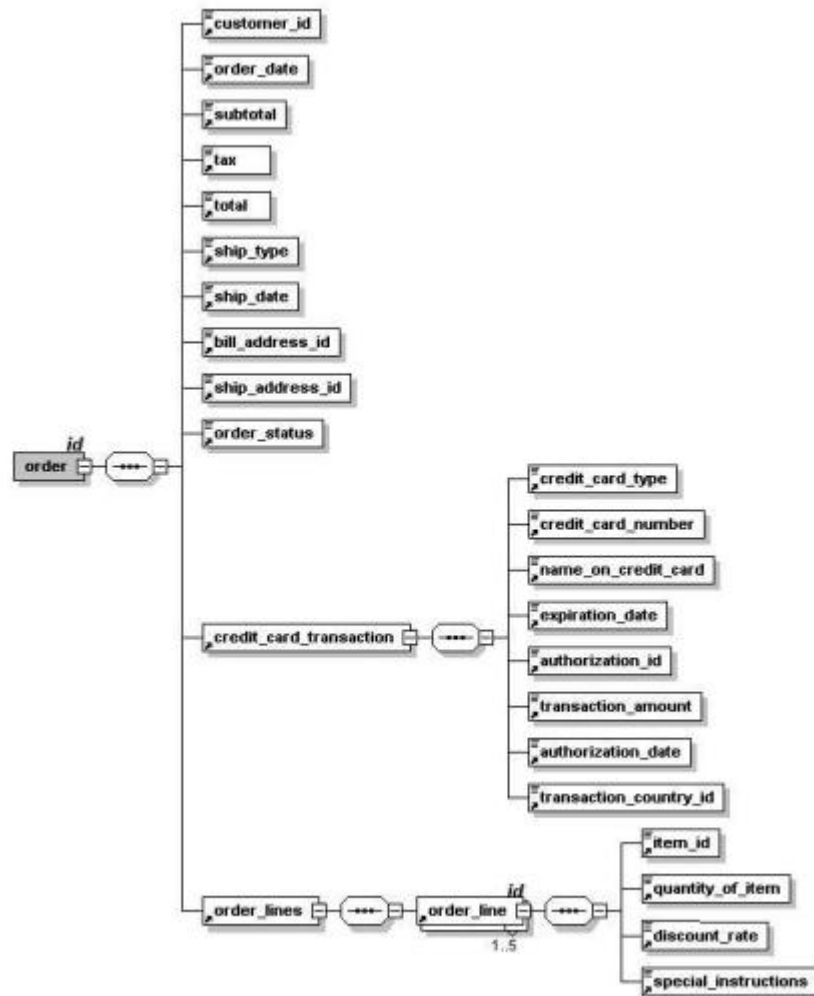


Fig. 5. Esquema *Orders* – Base de múltiplos documentos (MD) [Yao *et al.* 2004]

Análise das consultas do benchmark. Dentre as consultas disponibilizadas no *benchmark*, 19 delas foram consideradas como sendo as consultas mais frequentes executadas sobre as bases de dados. Estas por sua vez foram adaptadas para atender às restrições de consultas do mediador. Ou seja, o mediador só atende a um conjunto de regras de formação gramaticais conforme descrito em [Figueiredo *et al.* 2010]. 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. A Figura 6 apresenta um resumo dos predicados de seleção presentes em cada uma das consultas utilizadas no experimento.

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

Fig. 6. Consultas executadas nos experimentos e seus respectivos atributos de seleção

Definição dos critérios para a fragmentação. Para a definição dos critérios de fragmentação, inicialmente, foram utilizadas as diretrizes que são aplicadas no projeto de fragmentação para o modelo relacional [Ozsu and Valduriez 2011] e orientado a objetos [Baião *et al.* 2004]. Sendo assim, foram analisados os atributos e predicados de seleção que mais ocorrências tinham nas consultas assim como hoje é feito para o modelo relacional [Ozsu and Valduriez 2011] e orientado a objetos [Baião *et al.* 2004]. 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 Figura 7. Como podemos observar, existe uma frequência maior de consultas que utilizam o atributo order/total.

Predicados simples	Ocorrências
order/total > 11000	6 vezes
order/total > 10000	2 vezes
order/total > 7000	4 vezes
order/total < 2000	1 vez
order/total < 8000	1 vez
order/@id = "1"	1 vez
order/@id = "2"	1 vez
order/@id = "3"	1 vez
order/@id = "4"	1 vez
order/@id = "5"	1 vez
order/@id = "6"	1 vez

Fig. 7. Análise das consultas frequentes e a frequência dos predicados simples de seleção

Plano de execução. Após analisar a Figura 7, foram definidos 11 subcenários de avaliação, agrupados em três cenários. Os critérios de fragmentação de cada fragmento em cada subcenário são descritos na Figura 8 :

Cenário 0: Execução em ambiente centralizado.

Cenário 1: Execução da fragmentação horizontal, utilizando atributos das consultas frequentes. Como podemos ver na Figura 7, o atributo total é o que possui maior ocorrência. Outro ponto a ser observado é que o predicado de seleção total > 11000 ocorre em 6 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 da fragmentação horizontal, utilizando número total de nós disponíveis para alocação e 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:

(1.2.1) dois fragmentos em dois nós;

(1.2.2) quatro fragmentos em quatro nós;

(1.2.3) seis fragmentos em seis nós;

(1.2.4) oito fragmentos em oito nós.

Cenário 3: Execução da fragmentação horizontal, não utilizando os atributos das consultas frequentes. Se observarmos a Figura 5, podemos verificar que existe um atributo chamado transaction_country_id que não aparece em nenhum dos predicados de seleção listados na Figura 7. Avaliamos o domínio desse atributo e verificamos que ele varia entre 1 a 92. 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.

A Figura 8 apresenta um resumo dos critérios de fragmentação e alocação desses fragmentos em nosso experimento. Ou seja, a Figura 8 mostra os critérios de seleção utilizados na definição de cada fragmento baseado nas definições de fragmentos definidos por Andrade *et al.* [2006].

<u>Cenário</u>	<u>Critério de Fragmentação</u>	<u>Alocação</u>
0	Não se aplica	Nó 1
1.1.1	Frag 1: $\langle C_{Orders}, \sigma_{order/total} \geq 11000 \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{order/total} \leq 7000 \rangle$ Frag 3: $\langle C_{Orders}, \sigma_{order/total} > 7000 \wedge order/total < 11000 \rangle$	Frag 1 : Nó 1 Frag 2 : Nó 2 Frag 3: Nó 1
1.1.2	Frag 1: $\langle C_{Orders}, \sigma_{order/total} \geq 11000 \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{order/total} \leq 7000 \rangle$ Frag 3: $\langle C_{Orders}, \sigma_{order/total} > 7000 \wedge order/total < 11000 \rangle$	Frag 1 : Nó 1 Frag 2 : Nó 2 Frag 3: Nó 3
1.2.1	Frag 1: $\langle C_{Orders}, \sigma_{total} \leq 1000 \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{total} > 1000 \rangle$	Frag 1 : Nó 1 Frag 2 : Nó 2
1.2.2	Frag 1: $\langle C_{Orders}, \sigma_{total} \leq 1000 \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{order/total} > 1000 \wedge order/total \leq 7000 \rangle$ Frag 3: $\langle C_{Orders}, \sigma_{order/total} > 7000 \wedge order/total \leq 11000 \rangle$ Frag 4: $\langle C_{Orders}, \sigma_{order/total} > 11000 \rangle$	Frag 1 : Nó 1 Frag 2 : Nó 2 Frag 3 : Nó 3 Frag 4 : Nó 4
1.2.3	Frag 1: $\langle C_{Orders}, \sigma_{total} \leq 250 \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{order/total} > 250 \wedge order/total \leq 500 \rangle$ Frag 3: $\langle C_{Orders}, \sigma_{order/total} > 500 \wedge order/total \leq 1000 \rangle$ Frag 4: $\langle C_{Orders}, \sigma_{order/total} > 1000 \wedge order/total \leq 7000 \rangle$ Frag 5: $\langle C_{Orders}, \sigma_{order/total} > 7000 \wedge order/total \leq 11000 \rangle$ Frag 6: $\langle C_{Orders}, \sigma_{order/total} > 11000 \rangle$	Frag 1 : Nó 1 Frag 2 : Nó 2 Frag 3 : Nó 3 Frag 4 : Nó 4 Frag 5 : Nó 5 Frag 6 : Nó 6
1.2.4	Frag 1: $\langle C_{Orders}, \sigma_{total} \leq 250 \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{order/total} > 250 \wedge order/total \leq 500 \rangle$ Frag 3: $\langle C_{Orders}, \sigma_{order/total} > 500 \wedge order/total \leq 1000 \rangle$ Frag 4: $\langle C_{Orders}, \sigma_{order/total} > 1000 \wedge order/total \leq 5000 \rangle$ Frag 5: $\langle C_{Orders}, \sigma_{order/total} > 5000 \wedge order/total \leq 7000 \rangle$ Frag 6: $\langle C_{Orders}, \sigma_{order/total} > 7000 \wedge order/total \leq 9000 \rangle$ Frag 7: $\langle C_{Orders}, \sigma_{order/total} > 9000 \wedge order/total \leq 11000 \rangle$	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: $\langle C_{Orders}, \sigma_{order/total > 11000} \rangle$	Frag 8 : Nó 8
3.1.1	Frag 1: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id \leq 23} \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 23 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 46}} \rangle$ Frag 3: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 46 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 69}} \rangle$ Frag 4: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 69} \rangle$	Frag 1 : Nó 1 Frag 2 : Nó 1 Frag 3: Nó 2 Frag 4: Nó 2
3.1.2	Frag 1: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id \leq 23} \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 23 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 46}} \rangle$ Frag 3: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 46 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 69}} \rangle$ Frag 4: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 69} \rangle$	Frag 1 : Nó 1 Frag 2 : Nó 2 Frag 3: Nó 3 Frag 4: Nó 4
3.1.3	Frag 1: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id \leq 12} \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 12 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 23}} \rangle$ Frag 3: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 23 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 46}} \rangle$ Frag 4: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 46 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 69}} \rangle$ Frag 5: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 69 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 81}} \rangle$ Frag 6: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 81} \rangle$	Frag 1 : Nó 1 Frag 2 : Nó 2 Frag 3 : Nó 3 Frag 4 : Nó 4 Frag 5 : Nó 5 Frag 6 : Nó 6
3.1.4	Frag 1: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id \leq 12} \rangle$ Frag 2: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 12 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 23}} \rangle$ Frag 3: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 23 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 39}} \rangle$ Frag 4: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 39 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 46}} \rangle$ Frag 5: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 46 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 58}} \rangle$ Frag 6: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 58 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 69}} \rangle$ Frag 7: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 69 \wedge \sigma_{order/credit_card_transaction/transaction_country_id \leq 81}} \rangle$ Frag 8: $\langle C_{Orders}, \sigma_{order/credit_card_transaction/transaction_country_id > 81} \rangle$	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

Fig.8. Resumo dos critérios de fragmentação e alocação dos fragmentos em cada cenário

Baseado no plano de execução, a etapa de geração dos documentos XML, e, em seguida, da geração e alocação dos fragmentos. Para a geração desses documentos foi utilizado o ToXgene [Barbosa *et al.* 2002].

A geração dos fragmentos seguiu os critérios definidos pela etapa anterior. A alocação dos fragmentos pelos nós do cluster deu origem à distribuição detalhada nas Tabelas I e II. Os experimentos foram executados sobre bases de 4MB e 40 MB a fim de avaliarmos o comportamento sobre volumes de dados diferentes. Para a base de dados de múltiplos documentos, temos a seguinte composição das bases:

- 4 MB: 2592 documentos de 2 KB cada
- 40 MB: 25920 documentos de 2 KB cada

Como a massa de dados foi gerada de forma aleatória pelo ToXgene [2002], a alocação dos fragmentos em cada um dos cenários acima ficou distribuída conforme as Tabela I e II. Essa informação é importante para avaliarmos o comportamento das consultas a partir da quantidade de documentos alocados em cada nó.

Tabela I. Distribuição dos fragmentos alocados em cada nó – XBench (4 MB)

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					
1.2.1	1961	631						
1.2.2	1961	194	223	214				
1.2.3	300	336	925	194	223	214		
1.2.4	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 II. Distribuição dos fragmentos alocados em cada nó – XBench (40 MB)

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					
1.2.1	1689	24231						
1.2.2	1689	10403	6910	6918				
1.2.3	428	419	843	10403	6910	6918		
1.2.4	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

Execução dos planos e coleta dos resultados. Essa etapa consiste na execução dos planos definidos no item anterior e a coleta dos tempos de execução de cada consulta.

Análise dos resultados. Através dos resultados obtidos com a execução dos planos de execução foram levantados os critérios que mais influenciaram o desempenho das consultas.

Após a execução das etapas do processo de definição das diretrizes para fase de análise definidas anteriormente podem dizer quais são apresentando os critérios que devem ser levados em consideração na etapa de análise em um projeto de fragmentação em XML.

4. ANÁLISE DOS RESULTADOS

Os experimentos foram executados em um cluster, onde os diversos nós foram utilizados como servidores distribuídos totalmente dedicados para os testes. Os resultados são analisados nesta seção, a partir dos dados coletados na execução dos experimentos.

A análise dos resultados foi feita a partir da comparação dos tempos totais médios de execução a quente das consultas entre os diferentes cenários. Cada consulta foi executada 10 vezes e para o cálculo do tempo total médio foi desconsiderado o tempo total referente à primeira execução. Posteriormente, analisamos também os tempos totais médios desconsiderando o tempo gasto com a comunicação entre o Mediador e os Componentes. Por fim, analisamos, para cada consulta em cada cenário, o tempo médio de compilação pelo Mediador a fim de verificar individualmente o custo da decomposição da consulta distribuída.

É importante destacar que a implementação do Mediador e do Componente utilizada neste experimento é uma alteração da versão do protótipo construído em [Figueiredo *et al.* 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. Entretanto, os resultados que serão apresentados podem ser melhorados em futuras versões ou em outras implementações.

A seção 4.1 descreve os resultados obtidos com a fragmentação horizontal sobre uma base de dados de múltiplos documentos de tamanho igual a 4 MB. O detalhamento dos resultados do experimento para base de 40 MB é apresentada na seção 4.2.

4.1 Análise de Resultados: Base de 4 MB

Ao analisarmos, o comportamento da fragmentação da base de múltiplos documentos sobre a base de dados C_{Orders} de 4 MB observamos que dentre as dezenove consultas executadas, sete (C1, C2, C3, C4, C5, C12, C14) consultas obtiveram melhor desempenho no ambiente centralizado, conforme descrito na Figura 9. Todas essas consultas não possuem em seu predicado de seleção o atributo de seleção que foi utilizado em nossa fragmentação, conforme Figuras 6 e 8. Desta forma, a fragmentação não beneficiou o tempo de execução dessas consultas.

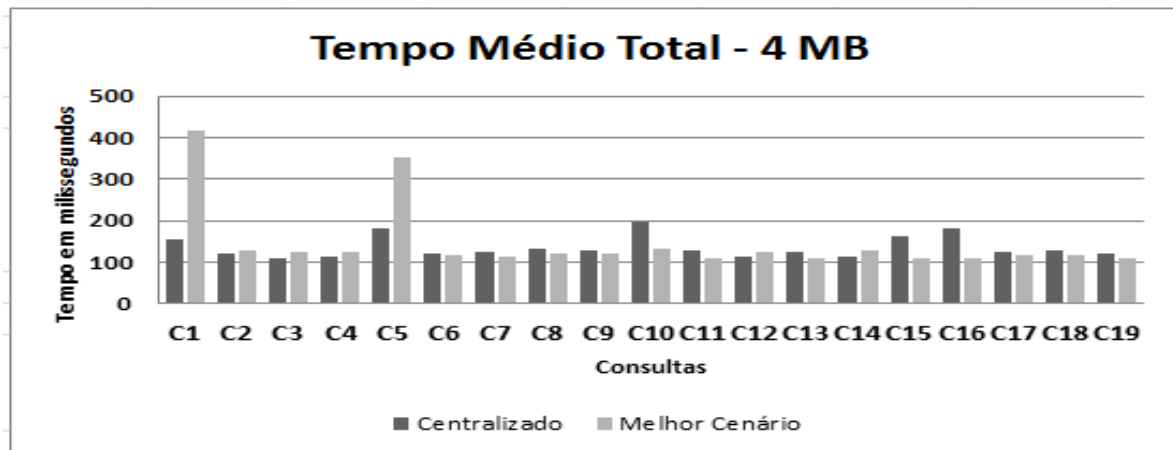


Fig.9. Comparação do Tempo Médio Total de execução das consultas sobre as bases C_{Orders} de 4 MB nos cenários centralizado e o melhor dos cenários analisados

Apresentamos nas Figuras 10 e 11, os gráficos de comparação entre o tempo de execução das consultas no ambiente centralizado e nos três cenários apresentados. A Figura 10 apresenta os tempos de execução total da consulta nos dois primeiros cenários (1 e 2). Como podemos perceber no primeiro cenário, o cenário 1.1.1 foi melhor que o centralizado nas consultas C7, C8, C9, C13, C15, C16, C17, C18 e C19. Esses resultados se devem pelo tempo de processamento das subconsultas em paralelo foram menor que no cenário centralizado. Já o cenário 1.1.2 obteve o melhor resultado dentre os demais apenas na consulta C10 com um ganho de aproximadamente cinquenta por cento se comparado ao centralizado, isso ocorreu pelo mesmo fato do cenário 1.1.1. Para o cenário 2 também apresentado na Figura 10, o cenário 1.2.1 apresentou um melhor tempo que o centralizado em 6 consultas. São elas: C7, C8, C9, C13, C15 e C18, com ganhos de tempo comparado ao centralizado variando de 10% a 50%. Para o cenário 1.2.2, seis consultas (C9, C11, C13, C15, C16, C19) se beneficiaram dessa fragmentação se compararmos com o centralizado. Já no cenário 1.2.3, sete consultas (C8, C9, C11, C13, C15, C16, C19) se beneficiaram dessa fragmentação com vantagens de tempo total nas consultas (C11 e C16) variando de 16% a 68%. Por último, temos no cenário 1.2.4, as consultas C15, C16 e C19 que obtiveram um melhor tempo de execução nesse cenário se comparado ao centralizado. Na análise foi possível observar que em todos os cenários onde o fragmentado obteve um melhor resultado que o centralizado, isso se deu pelo fato do tempo de processamento distribuído e paralelo das subconsultas foi menor do que no cenário centralizado.

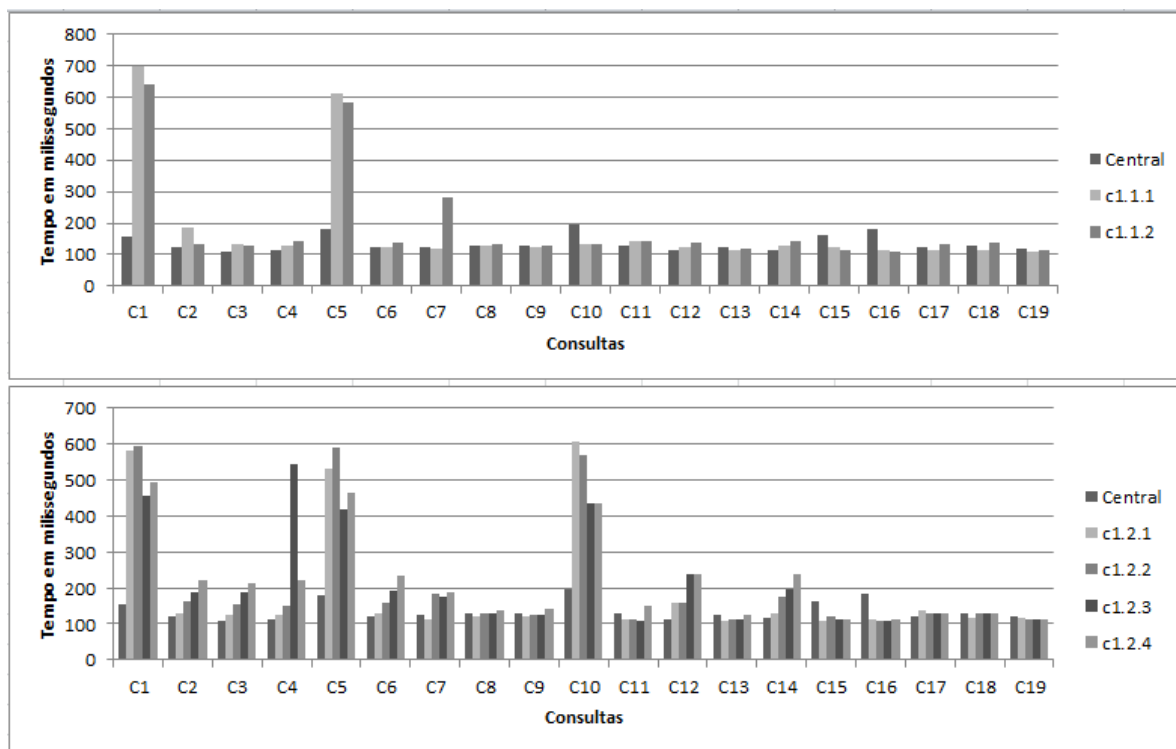


Fig. 10. Comparação do Tempo Médio Total da execução das consultas sobre as bases C_{Orders} de 4 MB nos cenários centralizado, 1 e 2

Por último, a Figura 11 descreve o comportamento dos tempos de execução 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 Figura 11 podemos perceber que as consultas C6, C8, C9, C15, C16 e C18 foram melhores em seu tempo total que a execução no ambiente centralizado. No processamento realizado pela aplicação, as subconsultas são executadas em paralelo. Ou seja, mesmo a consulta não tendo o predicado que foi usado na fragmentação, só a diminuição do volume de dados que cada subconsulta tem que processar já reduz o tempo de processamento. Por isso, o resultado das consultas listadas anteriormente, o tempo de processamento local no pior nó conseguiu ser menor do que em único nó.

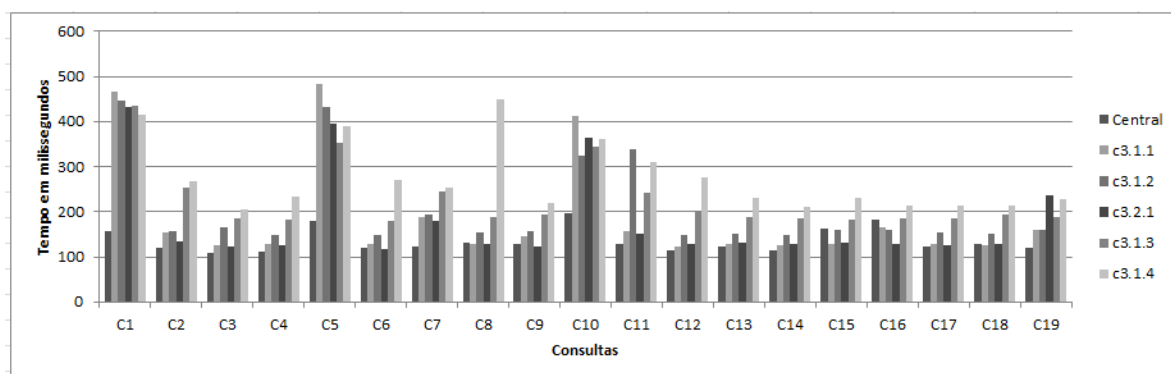


Fig. 11. Comparação do Tempo Médio Total da execução das consultas sobre as bases C_{Orders} de 4 MB nos cenários centralizado e no cenário (Cenário 3) fragmentado sem utilizar os atributos frequentes.

4.2 Análise de Resultados: Base de 40 MB

Para a base de 40 MB foi realizado o mesmo experimento realizado para a base 4 MB sobre o mesmo ambiente e com as mesmas consultas. Dentre as dezenove consultas, apenas sete delas não se beneficiaram assim como o experimento de 4MB. Entretanto, dentre as sete consultas (C1, C2, C3, C4, C5, C6 e C14) que não obtiveram ganhos com a fragmentação sobre a base de 40 MB, apenas as consultas C1, C2, C3, C4, C5, C14 tiveram este comportamento em ambos os experimentos. Ou seja, a consulta C6 teve um melhor resultado na base menor enquanto que a consulta C12 apresentou um melhor resultado na base maior. A Figura 12 apresenta o resultado do tempo médio de execução das consultas executadas na base de 40 MB 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 centralizado do que no fragmentado. Nessas duas consultas haviam 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 Componentes uma vez que as consultas tiveram que ser submetidas a todos os nós. Além do tempo remoto, tivemos um resultado 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 12 que as consultas C10 e C12 se beneficiaram de forma expressiva da fragmentação. Isso ocorrer pelo fato do processamento local no fragmentado ter sido menor que no centralizado, embora o tempo de consolidação dos resultados terem sido maiores nesses cenários do que o centralizado.

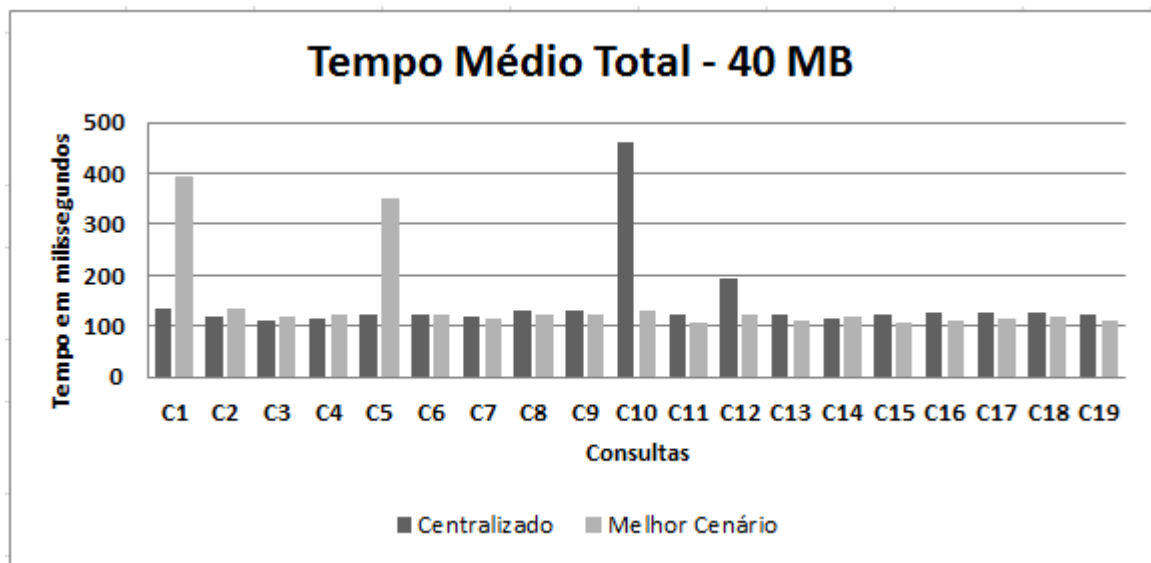


Fig.12. Comparação do Tempo Médio Total de execução das consultas sobre as bases C_{Orders} de 40 MB nos cenários centralizado e o melhor dos cenários analisados

Assim como apresentamos para o experimento de 4 MB, a Figura 13 apresenta os tempos de execução total da consulta nos dois primeiros cenários (1 e 2). Como podemos ver no primeiro gráfico, os tempos de execução das consultas C9, C10, C12 e C17 foram melhores no ambiente distribuído, com ganhos de tempo de execução variando entre 6% a 260% em relação ao ambiente centralizado. O cenário 1.1.1 foi melhor nas consultas C9, C12 e C17 e apenas a consulta C12 se beneficiou no cenário 1.1.2.

Para o segundo gráfico (Cenário 2) descrito na Figura 14, o cenário 1.2.1 apresentou um tempo melhor que o centralizado nas consultas C7, C8, C15 e C18, com tempos inferiores variando entre 5% a 15%. O cenário 1.2.2 foi melhor nas consultas C11 e C13 com ganho percentual variando de 11% a 15%. O ganho é aceitável uma vez que para essas consultas é acessado o fragmento (total > 11000) contendo

apenas 6918 documentos. Já o cenário 1.2.3 obteve um melhor tempo nas consultas C16 e C19, com ganhos percentuais idênticos ao cenário 1.2.4. Esse resultado semelhante entre os cenários 1.2.3 e 1.2.4 para as consultas C16 e C19 se deu pelo fato que em ambos os casos os fragmentos acessados possuem o mesmo tamanho (6918 fragmentos).

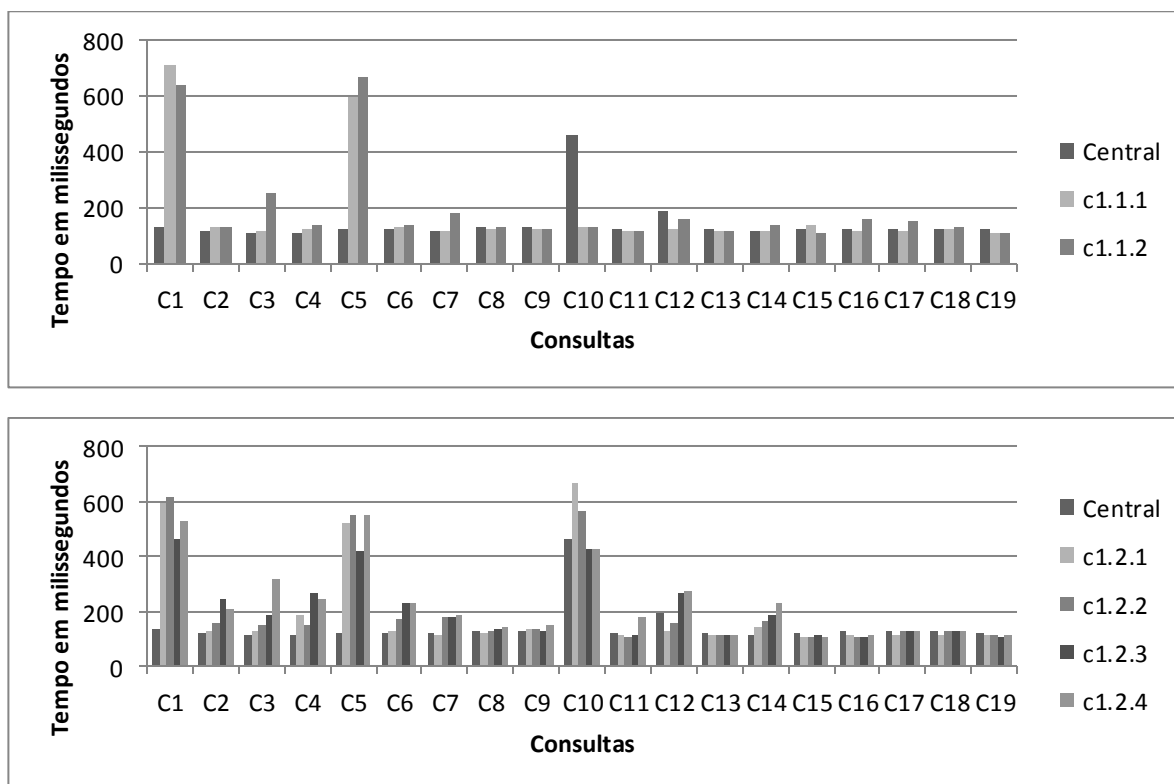


Fig. 13. Comparação do Tempo Médio Total da execução das consultas sobre as bases C_{Orders} de 40 MB nos cenários centralizado e no cenário 1e 2

Por último, temos a comparação do cenário centralizado com o cenário 3, a Figura 14 apresenta essa comparação dos tempos de execução entre os dois cenários. Na Figura 14 podemos observar que apenas as consultas C10, C12 e C19 tiveram melhor tempo no cenário 3 se comparadas com o tempo do centralizado. Isso ocorrer pelo fato do processamento local no fragmentado ter sido menor que no centralizado, embora o tempo de consolidação dos resultados terem sido maiores nesses cenários do que o centralizado.

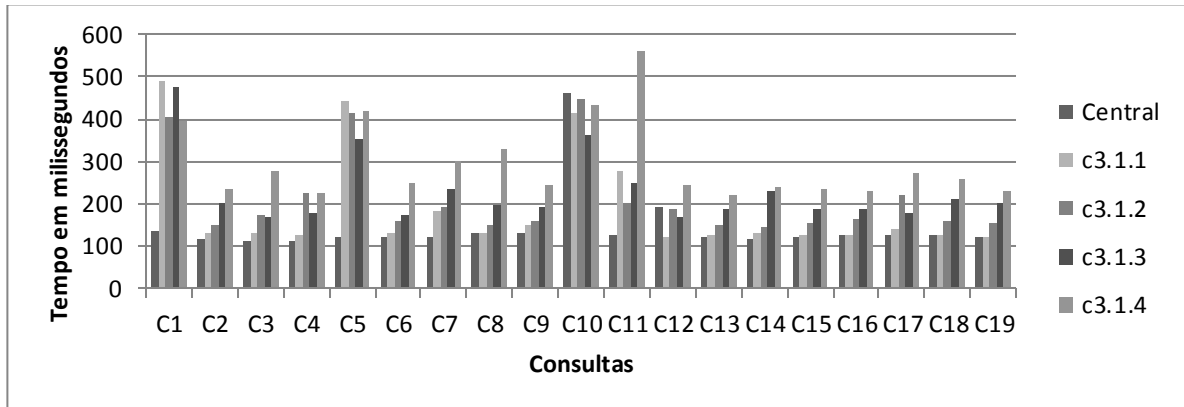


Fig. 14. Comparação do Tempo Médio Total da execução das consultas sobre as bases C_{Orders} de 40 MB nos cenários centralizado e no cenário (Cenário 3) fragmentado sem utilizar os atributos frequentes.

5. DIRETRIZES PARA FRAGMENTAÇÃO HORIZONTAL

O projeto de fragmentação de dados XML deve considerar diversos critérios, de forma análoga a outros modelos de dados que o antecederam. Essa seção apresenta alguns destes critérios, que foram inspirados em seus equivalentes no modelo relacional [Ozsu and Valduriez 2011] e orientado a objetos [Baião *et al.* 2004], e cuja influência no processamento de consultas sobre a base XML distribuída foi comprovada empiricamente durante nossos experimentos. Tais critérios são listados a seguir, e se mostraram necessários para a definição de diretrizes para fragmentação horizontal de dados XML.

Frequência de predicado de seleção: Ao analisar as consultas frequentes é importante verificar a frequência de predicados de seleção dentro do conjunto de consultas para que seja possível obter melhor desempenho se realizarmos a fragmentação horizontal. Este critério foi determinante no cenário 1, onde tínhamos dentro o conjunto de consultas, 4 consultas que utilizavam o mesmo predicado de seleção (“total > 11000”), conforme Figura 3. Nesse cenário, foi gerado um fragmento que atendia esse predicado de seleção. É importante ressaltar que as demais consultas que não utilizam o atributo de seleção podem não se beneficiar dessa fragmentação. Outro parâmetro importante é utilizar a frequência de execução das consultas junto com a análise de predicado de seleção, pois isso pode ou não viabilizar a fragmentação baseada na frequência de predicado de seleção. Entretanto, nos experimentos utilizados assumimos que todas as consultas tinham a mesma frequência de execução.

Domínio dos dados: O domínio dos dados é outro critério importante para a fragmentação horizontal, após a análise do atributo mais utilizado. Tal critério foi determinante no cenário 2, onde foi analisado o domínio do atributo “total”. Esse atributo variava entre 0 a 15000.

Número de nós disponíveis: Essa análise considera a alocação dos dados de forma a ocupar o máximo de nós disponíveis, sendo assim, ao agregarmos essa análise às duas outras anteriores podemos obter um conjunto de critérios para fragmentação. Entretanto, é importante avaliar o tamanho dos fragmentos gerados, pois podem ocorrer casos onde o fragmento fica com um volume de dados muito grande, o que pode acabar degradando os benefícios da fragmentação. Para obtermos uma avaliação do comportamento das consultas nos cenários 1 e 2 foram realizadas fragmentações variando a alocação entre 2 a 8 fragmentos. Ao analisarmos os resultados, dentre as consultas que se beneficiaram da fragmentação, no experimento de 4 MB, 7 consultas obtiveram um melhor tempo no cenário 2. Já no experimento de 40 MB, tivemos um total de 8 consultas. Lembrando que o cenário 2

foram considerado o domínio dos dados e a variação de número de nós disponíveis para alocação. Essa forma de fragmentar permite gerar fragmentos menores distribuídos pelos diversos nós.

Para exemplificar as diretrizes propostas nesse artigo, apresentamos na Figura 15 as conclusões retiradas a partir dos experimentos sobre bases de múltiplos documentos XML. Como podemos ver a entrada para definição das diretrizes são justamente todos os atributos de seleção e projeção existentes nas consultas existentes e o tamanho da base que se deseja fragmentar. Com base nisso, se nas consultas frequentes tivermos uma quantidade de atributos de seleção maior que a projeção, sugere-se aplicar a fragmentação horizontal. Entretanto, poderíamos fazer outra análise nesses casos, pois nessa situação talvez fosse interessante uma fragmentação híbrida. Isso permanece em aberto, e será tratado em trabalhos futuros. Após a escolha pela fragmentação horizontal e tendo as informações dos domínios dos atributos mais frequentes das consultas e a quantidade de nós disponíveis para alocação dos fragmentos, a fragmentação pode ser aplicada levando em consideração esses critérios. Todavia, se for observado que após a definição dos fragmentos, alguns deles ficaram com tamanhos muito superiores aos demais talvez seja necessário a diminuição do número de nós para alocação dos fragmentos até que o equilíbrio dos tamanhos dos fragmentos seja obtido. Outro ponto importante é que nos experimentos realizados não foi possível analisar se o tamanho da base faria diferença nos resultados uma vez que não tivemos muitas discrepâncias. Provavelmente, em bases muito maiores seja possível incluir a variável do tamanho da base no critério de escolha do melhor tipo de fragmentação a ser aplicada.

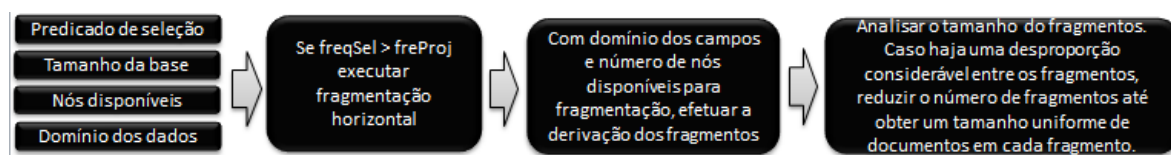


Fig.15. Definição das diretrizes para fragmentação horizontal de dados XML

CONCLUSÃO

Esse trabalho apresenta um conjunto de experimentos que abrange vários critérios que nos permitiu derivar diretrizes para fragmentação horizontal de dados XML baseados em seus resultados. Essas diretrizes visam obter ganhos de desempenho na execução de consultas sobre banco de dados distribuídos. Para chegar nessas diretrizes foram efetuados experimentos sobre bases XML de 4 e 40 MB buscando analisar o comportamento dos tempos de execução sobre vários cenários e a partir desses resultados obter diretrizes para o projeto de fragmentação de dados XML. Todos os cenários desenvolvidos nesse trabalho para a obtenção dessas diretrizes são oriundas dos modelos orientados a objetos [Baião *et al.* 2004] e relacional [Ozsu and Valduriez 2011]. Nesses modelos, as consultas mais frequentes são analisadas com objetivo de derivar a fragmentação dos dados.

Diante dos resultados obtidos podemos concluir que as consultas que possuíam em seu predicado de seleção o mesmo predicado utilizado na fragmentação se beneficiaram do processo de fragmentação realizado. Além disso, a fragmentação realizada sobre o predicado de seleção que não aparecia nas consultas frequentes apresentou ganhos em algumas consultas se comparado com o cenário centralizado. Nosso experimento apresentou um ganho de desempenho para as consultas frequentes que se beneficiaram do processo de fragmentação, se compararmos com os resultados obtidos com ambiente centralizado. A partir desses resultados foi possível a definição de diretrizes para fragmentação horizontal de dados XML que contribuem na escolha do melhor tipo de fragmentação horizontal a ser aplicada a uma determinada base de dados.

Como trabalhos futuros, propõe-se a execução desses mesmos cenários sobre bases de dados maiores a fim de avaliar se o aumento do volume de dados implica em resultados diferentes dos que já

obtivemos até o momento. Além disso, expandir as diretrizes desenhadas nesse trabalho para os demais tipos de fragmentação de dado XML: vertical e híbrida. Isso nos permitirá avaliar qual tipo de fragmentação melhor se adequa a uma determinada base e suas consultas frequentes.

REFERÊNCIAS

- ABITEBOUL, S., GOTTLOB, G., MANNA, M. Distributed XML Design. In *ACM PODS*, Providence, USA, pp. 247–258, 2009.
- ANDRADE, A., RUBERG, G., BAIÃO, F., BRAGANHOLO, V., MATTOSO, M. Efficiently Processing XML Queries over Fragmented Repositories with PartiX. In *International Workshop on Database Technologies for Handling XML Information on the Web*, Munich, Germany, pp. 150–163, 2006.
- BAIÃO, F., MATTOSO, M., ZAVERUCHA, G. A Distribution Design Methodology for Object DBMS. *Distributed Parallel Databases* 16(1): 45–90, 2004.
- BARBOSA, D., MENDELZON, A., KEENLEYSIDE, J., LYONS, K. ToXgene: a template-based data generator for XML. In, Madison, Wisconsin, 2002.
- BREMER, J.-M., GERTZ, M. On Distributing XML Repositories. In *Workshop on Web and Databases (WebDB)*, San Diego, United States, pp. 73–78, 2003.
- FIGUEIREDO, G., BRAGANHOLO, V., MATTOSO, M. Um Mediador para o Processamento de Consultas sobre Bases XML Distribuídas. In *Proceedings of the Demos Session of the Brazilian Symposium on Databases*, João Pessoa, Brazil, pp. 21–26, 2007.
- FIGUEIREDO, G., BRAGANHOLO, V. P., MATTOSO, M. Processing Queries over Distributed XML Databases. *Journal of Information and Data Management (JIDM)* 1(3): 455–470, 2010.
- FOMICHEV, A., GRINEV, M., KUZNETSOV, S. Sedna: A native XML DBMS. 3831(SOFSEM 2006: Theory and Practice of Computer Science, J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, and J. Stuller (Eds.)): 272–281, 2006.
- GERTZ, M., BREMER, J.-M. *Distributed XML Repositories: Top-down Design and Transparent Query Processing*. Technical Report T.R.CSE-2003-20, Department of Computer Science, 2003.
- KLING, P., OZSU, T., DAUDJEE, K. *Distributed XML Query Processing: Fragmentation, Localization and Pruning*. Technical Report CS-2010-02, University of Waterloo, 2010.
- KLING, P., ÖZSU, M., DAUDJEE, K. Scaling XML query processing: distribution, localization and pruning. *Distributed and Parallel Databases* 29(5): 445–490, 2011.
- LIMA, A., MATTOSO, M., VALDURIEZ, P. Adaptive Virtual Partitioning for OLAP Query Processing in a Database Cluster. *JIDM* 1(1): 75–88, 2010.
- MA, H., SCHEWE, K.-D. Fragmentation of XML documents. In *Proceedings of the Brazilian Symposium on Databases*, Manaus, Brazil, pp. 200–214, 2003.
- OZSU, M. T., VALDURIEZ, P. *Principles of Distributed Database Systems*. Prentice Hall, 2011.
- PAGNAMENTA, F. Design and initial implementation of a distributed xml database. Master Thesis, Universidade de Dublin, Irlanda, 2005.

- RODRIGUES, C., BRAGANHOLO, V., MATTOSO, M. Virtual Partitioning ad-hoc Queries over Distributed XML Databases. *Journal of Information and Data Management* 2(3): 495–510, 2011.
- YAO, B. B., OZSU, M. T., KHANDELWAL, N. XBench benchmark and performance testing of XML DBMSs. In *IEEE International Conference on Data Engineering (ICDE)*, Boston, United States, pp. 621–632, 2004.