Recomendações para fragmentação horizontal de bases de dados XML

Tatiane Lima da Silva[1], Fernanda Baião[2], Jonice de Oliveira Sampaio[1], Marta Mattoso[3], Vanessa Braganholo[4]

[1]PPGI/Universidade Federal do Rio de Janeiro, Brasil
[2] NP2Tec/Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Brasil
[3]COPPE/Universidade Federal do Rio de Janeiro, Brasil
[4]Universidade Federal Fluminense, Brasil
tatiane.lima@ppgi.ufrj.br, fernanda.baiao@uniriotec.br, jonice@dcc.ufrj.br
marta@cos.ufrj.br, vanessa@ic.uff.br

Resumo. 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). Inspirado pelos modelos relacionais e orientado a objetos, que têm metodologias sólidas para o projeto de fragmentação de bases de dados, o objetivo principal deste trabalho é estudar e propor recomendações, baseadas em experimentos, que poderiam ser usados em projetos de distribuição de bancos de dados XML, a fim de aumentar o desempenho do processamento de consultas.

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; Gang e Rada 2007; Kling *et al.* 2009; Moro *et al.* 2009; Figueiredo *et al.* 2010]. 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 grande sucesso. Nesta abordagem, os dados devem ser distribuídos pelos diferentes nós de uma rede segundo técnicas de fragmentação e de alocação de dados [Kling *et al.* 2010, 2011; Ozsu e Valduriez 2011].

Existem duas formas distintas de fragmentação: fragmentação física e fragmentação virtual. A fragmentação física [Ozsu e Valduriez 2011] fragmenta os dados fisicamente, e os aloca em diferentes nós. Já a fragmentação virtual [Rodrigues *et al.* 2011] exige que os dados sejam replicados nos nós da rede, exigindo mais 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 (proximidade) 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

[Figueiredo *et al.* 2010] 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, o projeto de fragmentação da base de dados precisa analisar as consultas mais frequentes, para que a fragmentação proporcione ganho de desempenho na maioria das consultas realizadas sobre a base de dados distribuída.

Aproveitando as ideias de fragmentação e distribuição propostas para o modelo relacional [Ozsu e 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 e Bremer 2003; Ma e Schewe 2003; Andrade *et al.* 2006; Abiteboul *et al.* 2009]. No entanto, não existe na literatura nenhuma metodologia para o 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 e Bremer 2003; Abiteboul *et al.* 2009; Ozsu e 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 e Gertz 2003; Ma e 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 em 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 e 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 recomendações para fragmentação horizontal de bases de dados XML. 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. Nesse artigo, essas recomendações são derivadas de análises de resultados experimentais sobre duas bases de dados XML de tamanhos distintos.

O restante deste artigo está estruturado da seguinte forma. 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 projetos de fragmentação horizontal de dados XML. A Seção 3 apresenta uma avaliação experimental, que foi usada como base para derivar recomendações para fragmentação horizontal, que são apresentadas na Seção 4. Finalmente, as conclusões e trabalhos futuros são apresentados na Seção 5.

2. FRAGMENTAÇÃO HORIZONTAL DE BASES DE DADOS XML

Existem diversos trabalhos na literatura que apresentam definições de fragmentos XML [Bremer e Gertz 2003; Andrade *et al.* 2006; Kling *et al.* 2010] e para o projeto de fragmentação para XML em geral [Gertz e Bremer 2003; Ma e 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 e Gertz 2003; Ma e Schewe 2003; Pagnamenta 2005; Kling *et al.* 2011] essa é a que mais se aproxima da definição de fragmentos do modelo relacional [Ozsu e Valduriez 2011]. Essa escolha é essencial, já que desejamos aproveitar as ideias de projeto de fragmentação propostas para o modelo relacional [Ozsu e 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.

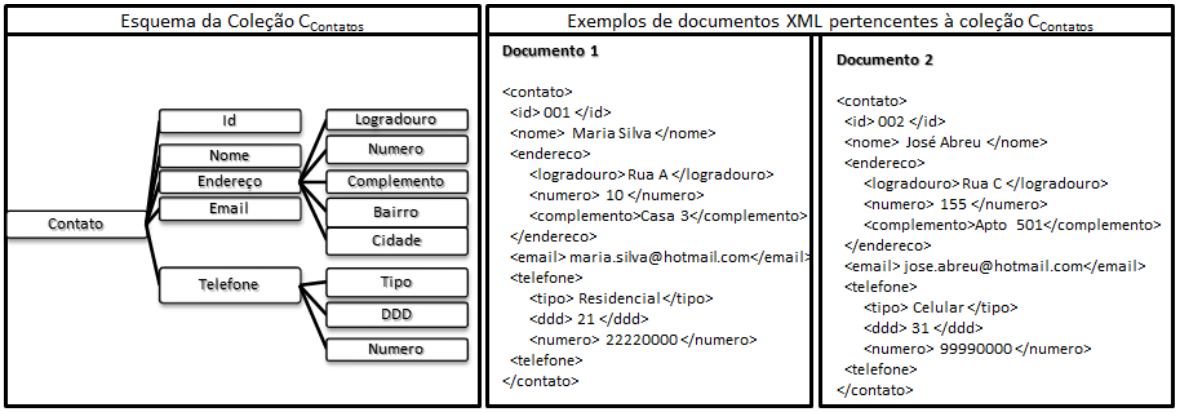

Fig. 1. Exemplo de esquema e documentos XML

Andrade et al. [2006] definem um fragmento horizontal da seguinte forma. Seja μ uma conjunção de predicados simples sobre uma coleção de dados XML *C*. O fragmento horizontal *F* de *C* definido por μ é dado pela expressão $F:=<C, \sigma\mu>$, onde σμ denota a seleção de documentos em *C* que satisfazem μ, isto é, *F* contém documentos de *C* para os quais σμ é verdadeiro. 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. Este tipo de repositório 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 $F1_{Residencial}$ 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 $F2_{Residencial}$ agrupa os documentos

cujo conteúdo de tipo difere de Residencial. Logo, o documento 2 que possui *tipo* igual a Celular irá compor o fragmento $F2_{Residencial}$.

$$F1_{Residencial} := \langle C_{Contatos}, \sigma_{contato/telefone/tipo='Residencial'} \rangle$$
$$F2_{Residencial} := \langle C_{Contatos}, \sigma_{contato/telefone/tipo <> 'Residencial'} \rangle$$

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

Com base na definição de fragmentos definida 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 adaptador, que recebe as subconsultas enviadas pelo mediador e as executa no nó local.

3. AVALIAÇÃO EXPERIMENTAL

Para obter recomendações que possam ser usadas na etapa de análise de projeto de fragmentação de bases XML, executamos uma série de experimentos, utilizando um benchmark. Os dados do benchmark foram sistematicamente fragmentados de acordo com várias estratégias, inspiradas na literatura [Baião *et al.* 2004; Ozsu e Valduriez 2011], e o comportamento das consultas foi avaliado em cada cenário. A análise dos resultados foi feita a partir da comparação dos tempos totais médios de execução 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 rodada. 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]. Abaixo, temos o detalhamento de cada um dos cenários executados. A Tabela I apresenta um resumo dos critérios de fragmentação e alocação desses fragmentos em nosso experimento.

Cenário 0: Execução em ambiente centralizado, utilizando apenas 1 nó para executar todas as consultas.

Cenário 1: Execução da fragmentação horizontal, utilizando atributos das consultas frequentes. Foram utilizados 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 de nós disponíveis para alocação e domínio dos dados. Avaliando o domínio do atributo de seleção mais frequente, o objetivo é fragmentar a partir do domínio dos dados em 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 da fragmentação horizontal, não utilizando os atributos das consultas frequentes. Foram executados 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 I. Resumo dos critérios de fragmentação e alocação dos fragmentos em cada cenário

Cenário	Critério de Fragmentação	Alocação	Cenário	Critério de Fragmentação	Alocação
1.1.1	Frag 1: total >= 11000 Frag 2 : total <= 7000 Frag 3: total > 7000 e total < 11000	Frag 1 : Node 1 Frag 2 : Node 2 Frag 3: Node 1	1.2.4	Frag 1: total <=250 Frag 2: total >250 e total <=500 Frag 3: total >500 e total <=1000 Frag 4: total > 1000 e total <=5000 Frag 5: total > 5000 e total <=7000 Frag 6: total >7000 e total <=9000 Frag 7: total >9000 e total <=11000 Frag 8: total >11000	Frag 1 : Node 1 Frag 2 : Node 2 Frag 3 : Node 3 Frag 4 : Node 4 Frag 5 : Node 5 Frag 6 : Node 6 Frag 7 : Node 7 Frag 8 : Node 8
1.1.2	Frag 1: total >= 11000 Frag 2 : total <= 7000 Frag 3: total > 7000 e total < 11000	Frag 1 : Node 1 Frag 2 : Node 2 Frag 3: Node 3	3.1.1	Frag 1 : transaction_country_id <=23 Frag 2 : transaction_country_id > 23 e transaction_country_id <=46 Frag 3 : transaction_country_id > 46 e transaction_country_id <=69 Frag 4 :transaction_country_id > 69	Frag 1 : Node 1 Frag 2 : Node 1 Frag 3: Node 2 Frag 4: Node 2
1.2.1	Frag 1: total <=1000 Frag 2: total >1000	Frag 1 : Node 1 Frag 2 : Node 2	3.1.2	Frag 1 : transaction_country_id <=23 Frag 2 : transaction_country_id > 23 e transaction_country_id <=46 Frag 3 : transaction_country_id > 46 e transaction_country_id <=69 Frag 4: transaction_country_id > 69	Frag 1 : Node 1 Frag 2 : Node 2 Frag 3: Node 3 Frag 4: Node 4
1.2.2	Frag 1: total <=1000 Frag 2: total > 1000 e total <=7000 Frag 3: total >7000 e total <=11000 Frag 4: total >11000	Frag 1 : Node 1 Frag 2 : Node 2 Frag 3 : Node 3 Frag 4 : Node 4	3.1.3	Frag 1: transaction_country_id <=12 Frag 2: transaction_country_id > 12 e transaction_country_id <=23 Frag 3: transaction_country_id > 23 e transaction_country_id <=46 Frag 4: transaction_country_id > 46 e transaction_country_id <=69 Frag 5: transaction_country_id > 69 e transaction_country_id <=81 Frag 6: transaction_country_id > 81	Frag 1 : Node 1 Frag 2 : Node 2 Frag 3 : Node 3 Frag 4 : Node 4 Frag 5 : Node 5 Frag 6 : Node 6
1.2.3	Frag 1: total <=250 Frag 2: total >250 e total <=500 Frag 3: total >500 e total <=1000 Frag 4: total > 1000 e total <=7000 Frag 5: total >7000 e total <=11000 Frag 6: total >11000	Frag 1 : Node 1 Frag 2 : Node 2 Frag 3 : Node 3 Frag 4 : Node 4 Frag 5 : Node 5 Frag 6 : Node 6	3.1.4	Frag 1: transaction_country_id <=12 Frag 2: transaction_country_id > 12 e transaction_country_id <=23 Frag 3: transaction_country_id > 23 e transaction_country_id <=39 Frag 4: transaction_country_id > 39 e transaction_country_id <=46 Frag 5: transaction_country_id > 46 e transaction_country_id <=69 Frag 6: transaction_country_id > 69 e transaction_country_id <=81 Frag 7: transaction_country_id > 81	Frag 1 : Node 1 Frag 2 : Node 2 Frag 3 : Node 3 Frag 4 : Node 4 Frag 5 : Node 5 Frag 6 : Node 6 Frag 7 : Node 7 Frag 8 : Node 8

Para avaliarmos o comportamento dos cenários foram executadas 19 consultas pertencentes ao benchmark XBench [Yao *et al.* 2004]. Cada consulta foi executada em bases de dados de múltiplos documentos pertencentes ao XBench. Além disso, avaliamos o comportamento das consultas em duas bases de tamanhos distintos (4 MB e 40 MB). O resumo dessas consultas e os predicados de seleção utilizados em cada uma delas são apresentados na Tabela II. Ao analisarmos essas consultas é possível observar que o atributo de seleção que mais aparece nas consultas é "total". Por esse motivo, o cenário 1 visa avaliar os resultados quando fragmentamos a partir do predicado de seleção que mais aparece nas consultas e o cenário 2 propõe uma análise sobre o comportamento dos tempos de execução quando se utiliza o predicado de seleção com maior ocorrência juntamente com a análise sobre o domínio (distribuição de valores) desse atributo.

Tabela II. Consultas executadas nos experimentos e seus respectivos atributos de seleção

Consulta	Predicado de seleção	Consulta	Predicado de seleção	Consulta	Predicado de seleção
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				

A comparação dos tempos médios de execução 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. Outro ponto importante a destacar é que a implementação do Mediador e do Adaptador utilizada neste experimento foi uma alteração da versão do protótipo construído por Figueiredo et al. [2010], tendo como objetivo aperfeiçoar o protótipo para obtenção de melhor desempenho. A Figura 3 apresenta uma análise comparativa dos tempos médios de execução entre o ambiente centralizado (Cenário 0) e o melhor cenário analisado nos experimentos com bases de dois tamanhos: 4 MB e 40 MB, respectivamente.

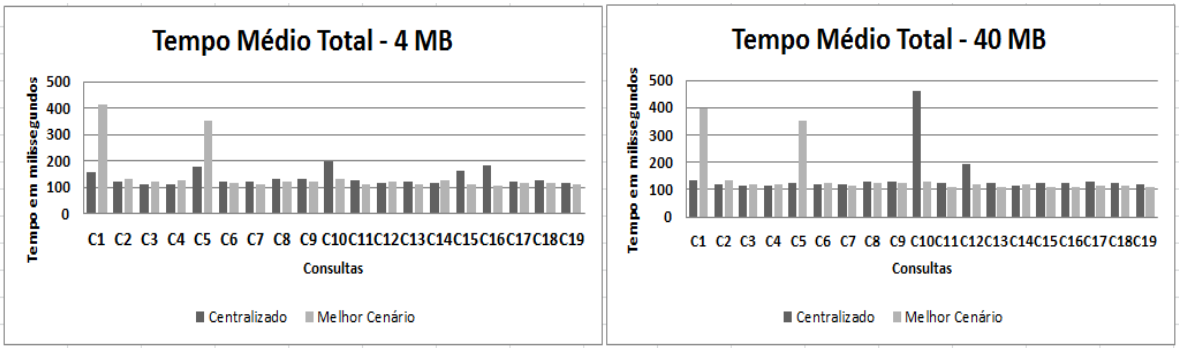

Fig.3. Comparação do Tempo Médio Total (em milissegundos) de execução das consultas sobre as bases C_{Orders} de 4 e 40 MB nos cenários centralizado (Cenário 0) e o melhor dos cenários analisados.

Ao analisarmos a Tabela II e a Figura 3, podemos observar que nos experimentos executados tivemos 12 consultas que se beneficiaram da fragmentação se compararmos com o ambiente centralizado. No experimento de 4 MB, essas consultas são: C6, C7, C8, C9, C10, C11, C13, C15, C16, C17, C18, C19. Já para o experimento de 40 MB, a consulta C6 não se beneficiou da fragmentação e em contrapartida, a consulta C12 se beneficiou. Dentre as consultas que não se beneficiaram da fragmentação no experimento de 4 MB apenas as consultas que não possuíam o atributo de seleção "total" não se beneficiaram da fragmentação (C1, C2, C3, C4, C5, C12 e C14). Para o experimento de 40 MB obtivemos o mesmo comportamento havendo apenas uma variação nas consultas que não se beneficiaram. São elas: C1, C2, C3, C4, C5, C6 e C14. Resumidamente, para o experimento de 4 MB obtivemos um ganho de quase 70% na consulta C16 e no experimento de 40 MB foi observado um ganho de aproximadamente 260% na consulta C10. As estatísticas entre os dois experimentos não aparentam muitas divergências em seus resultados. Todavia, a consulta C10 teve um ganho bem maior com a base de 40 MB, por outro lado, as consultas C15 e C16 reduziram o seu ganho nessa base. Além disso, foi possível identificar que as consultas C12 (Experimento 4MB) e C6 (Experimento 40 MB), embora não tivessem o atributo total em seu predicado de seleção, se beneficiaram da fragmentação. Isso se explica pelo fato de a fragmentação realizada no cenário 3.1.2 ter gerado fragmentos de tamanho uniforme, possibilitando assim um grau maior de paralelismo durante a realização da consulta. Por questões de restrições de espaço, os experimentos são apresentados em detalhes em um relatório técnico [Silva *et al.* 2012].

4. RECOMENDAÇÕES PARA FRAGMENTAÇÃO HORIZONTAL

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 e 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 recomendações 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 dentre o conjunto de consultas, 4 consultas que utilizavam o mesmo predicado de seleção ("total > 11000"), conforme Tabela I. 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 (universo de valores) 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 ao máximo os 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.

Para exemplificar as recomendações propostas nesse artigo, apresentamos na Figura 4 as conclusões retiradas a partir dos experimentos sobre bases de múltiplos documentos XML. Como podemos ver a entrada para definição das recomendações 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 analise 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.

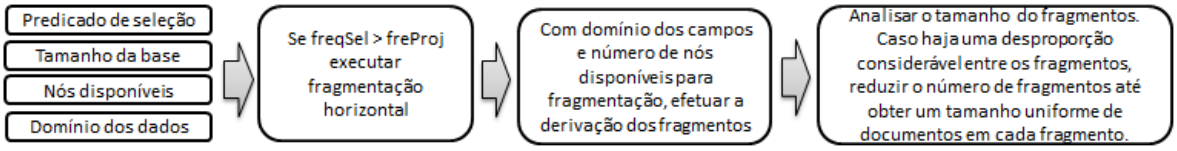

Fig.4. Definição das recomendações para fragmentação horizontal de dados XML

5. CONCLUSÃO

Esse trabalho apresenta recomendações para fragmentação horizontal de dados XML baseados em um conjunto de experimentos que abrange vários critérios. Essas recomendações visam a obter ganhos de desempenho na execução de consultas sobre bancos de dados distribuídos. Para chegar nessas recomendações foram efetuados experimentos sobre bases XML de 4 e 40 MB buscando analisar o comportamento dos tempos de execução em vários cenários. Esses tamanhos de bases visam a analisar o crescimento dos dados e estamos trabalhando em experimentos adicionais, com mais dados, que nos permitirão uma análise mais assertiva. Isso nos favorecerá avaliar o tipo de fragmentação que traz mais ganho em desempenho para as consultas mais frequentes sobre uma determinada base.

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 recomendações 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 resultados diferentes dos que já obtivemos até o momento. Além disso, expandir as recomendações desenhadas nesse trabalho para os demais tipos de fragmentação de dado XML: vertical e hibrida. Isso nos permitirá avaliar qual tipo de fragmentação mais se adequa a uma determinada base e suas consultas frequentes.

Agradecimentos. Os autores gostariam de agradecer ao CNPq e FAPERJ pelo financiamento parcial desse trabalho.

REFERÊNCIAS

ABITEBOUL, S., GOTTLOB, G., MANNA, M. Distributed XML Design. In *ACM PODS*, Providence, USA, p. 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, p. 150–163, 2006.

BAIÃO, F., MATTOSO, M., ZAVERUCHA, G. A Distribution Design Methodology for Object DBMS. *Distributed Parallel Databases* 16(1): 45–90, 2004.

BREMER, J.-M., GERTZ, M. On Distributing XML Repositories. In *Workshop on Web and Databases (WebDB)*, San Diego, United States, p. 73–78, 2003.

FIGUEIREDO, G., BRAGANHOLO, V., 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.

GANG, G., RADA, C. Efficiently Querying Large XML Data Repositories: A Survey. (IEEE Trans. Knowl. Data Eng. (TKDE)): 1381-1403, 2007.

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. *Optimizing distributed XML queries through localization and pruning*. Technical Report CS-2009-13, University of Waterloo, 2009.

KLING, P., OZSU, M. T., DAUDJEE, K. Generating efficient execution plans for vertically partitioned XML databases. *PVLDB* 4(1): 1-11, 2010.

KLING, P., ÖZSU, M., DAUDJEE, K. Scaling XML query processing: distribution, localization and pruning. *Distributed and Parallel Databases* 29(5): 445-490, 2011.

MA, H., SCHEWE, K.-D. Fragmentation of XML documents. In *Proceedings of the Brazilian Symposium on Databases*, Manaus, Brazil, p. 200–214, 2003.

MORO, M. M., BRAGANHOLO, V., DORNELES, C. F., DUARTE, D., GALANTE, R., MELLO, R. S. XML: some papers in a haystack. *SIGMOD Rec.* 38(2): 29–34, 2009.

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.

SILVA, T., BAIÃO, F., SAMPAIO, J., MATTOSO, M., BRAGANHOLO, V. *Definição de recomendações para fragmentação horizontal de bases de dados XML*. Technical Report, Universidade Federal do Rio de Janeiro, 2012.

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, p. 621–632, 2004.