Um Mediador para o Processamento de Consultas sobre Bases XML Distribuídas

Guilherme Figueiredo[1], Vanessa Braganholo[2], Marta Mattoso[1]

[1]Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ
[2]Departamento de Ciência da Computação – IM/UFRJ

{g.coelho, marta}@cos.ufrj.br, braganholo@dcc.ufrj.br

Abstract. *This paper describes a tool that implements an architecture to the query processing of XQueries over distributed and fragmented XML databases. This architecture, based on a Mediator with Adapters attached to the remote databases, implements a query processing methodology where the Mediator publishes a global XML view of the distributed data which can be queried transparently.*

Resumo. *Este artigo descreve uma ferramenta que implementa uma arquitetura para o processamento de consultas XQuery sobre bases de dados XML distribuídas e fragmentadas. Esta arquitetura, baseada em um Mediador com Adaptadores acoplados aos bancos de dados remotos, implementa uma metodologia de processamento de consultas distribuídas, onde o Mediador fornece uma visão XML global dos dados distribuídos que pode ser consultada de forma transparente.*

1. Introdução

O processamento distribuído de consultas XML ganhou importância com a popularidade do uso de documentos XML pela *Web*. Neste contexto, surgiram uma série de sistemas para processamento de consultas sobre bases XML distribuídas na *Web*, como [Suciu 2002, Gertz,Bremer 2003, Re et al. 2004]. Outros trabalhos enfocam o processamento de consultas XML sobre bases heterogêneas distribuídas, como [Baru et al. 1999, Gardarin et al. 2002, Lee et al. 2002]. Entretanto, nenhuma dessas propostas utiliza técnicas de fragmentação de bases XML.

Técnicas de fragmentação de bases XML nativas começaram a ser exploradas com o aumento crescente do volume de dados XML armazenado. Algumas técnicas já foram elaboradas [Bremer,Gertz 2003, Andrade et al. 2006]. Em especial, as definições de fragmentos propostas por Andrade et al. fazem uso de uma álgebra XML (TLC [Paparizos et al. 2004]). Neste trabalho, são também definidas regras formais de reconstrução de documentos XML a partir de seus fragmentos. Experimentos que mostram o potencial de ganho em desempenho para consultas realizadas sobre bases XML fragmentadas também foram explorados [Andrade et al. 2006].

Para que o processamento de uma consulta sobre uma base XML nativa distribuída seja automático e genérico, se faz necessário o uso de um formalismo XML. Através de uma álgebra XML, uma consulta XQuery pode ser reescrita através de expressões algébricas. Ao usar fragmentos definidos com operações dessa mesma álgebra, regras formais de equivalência entre um documento XML e seus fragmentos

podem ser usadas. Assim, torna-se possível substituir a referência a um documento XML, numa expressão algébrica de uma consulta, por referências aos fragmentos que compõem esse documento a partir da sua regra algébrica de reconstrução. Essa substituição pode ser realizada de forma correta e automática, quando os fragmentos são definidos por operações da mesma álgebra usada na reescrita da consulta XQuery. Na nossa proposta utilizamos a álgebra TLC [Paparizos et al. 2004].

Entretanto, esse formalismo não é suficiente. Um processamento automático distribuído necessita de uma metodologia que defina etapas que possibilitem o processamento de consultas XQuery distribuídas, de forma análoga à metodologia existente para o modelo relacional [Özsu,Valduriez 1999]. Esta metodologia foi definida em nosso trabalho anterior [Figueiredo et al. 2007] e implementada na forma de uma arquitetura baseada em um Mediador com Adaptadores acoplados às bases remotas, que será descrita neste artigo. O uso desta arquitetura torna o processamento de consultas XQuery distribuídas totalmente transparente para o usuário final. Através da ferramenta construída, é possível visualizar graficamente todas as etapas de processamento da consulta distribuída: decomposição, localização dos dados, e reconstrução do resultado final. Deste modo, a ferramenta pode também ser utilizada para fins didáticos.

A arquitetura proposta para o processamento de consultas distribuídas é apresentada na seção 2. A seção 3 apresenta a interface para testes do Mediador, enquanto que a seção 4 contém as considerações finais.

2. Arquitetura do Sistema

Em nossa arquitetura, gostaríamos que uma base de dados pudesse ser vista de duas maneiras distintas: (i) quando uma base centralizada é fragmentada, deve-se ter uma visão centralizada dos fragmentos, para que a fragmentação fique transparente para o usuário; (ii) quando várias bases locais pré-existentes precisam ser acessadas de forma integrada, deve-se poder definir uma visão XML sobre elas, de modo que cada base local possa ser considerado um fragmento desta visão global (exemplo: uma rede de livrarias, onde cada base de filial possa ser vista como um fragmento de uma visão global (virtual)). Deste modo, definimos uma arquitetura que suportasse ambas estas visões, e chamamos a base central (em ambos os casos), de visão global.

Para compor uma visão global e servir de ponto único de acesso, consideraremos o uso de um mediador [Wiederhold 1992] como mostrado na Figura 1, que seria responsável pelo processamento das consultas distribuídas, ocultando dos usuários os detalhes da localização e da fragmentação da base de dados. As consultas submetidas sobre uma visão global seriam decompostas em um conjunto de sub-consultas, que seriam então executadas pelos nodos remotos sobre os fragmentos. Os resultados de cada sub-consulta retornariam ao mediador para construção do resultado final. No contexto de bases XML distribuídas, consultas XQuery seriam submetidas sobre visões globais de fragmentos XML distribuídos.

A arquitetura apresentada na Figura 1, sobre a qual nossa implementação foi totalmente baseada, contempla os seguintes componentes: Mediador, Adaptadores e Catálogo, que serão detalhados nas seções que seguem.

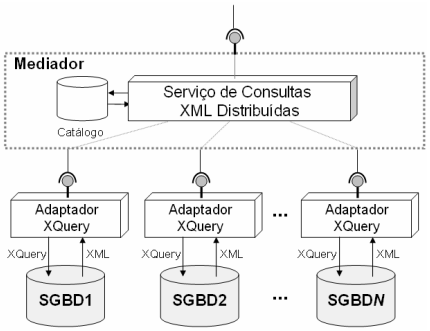

Figura 1: Arquitetura Mediador – Adaptadores

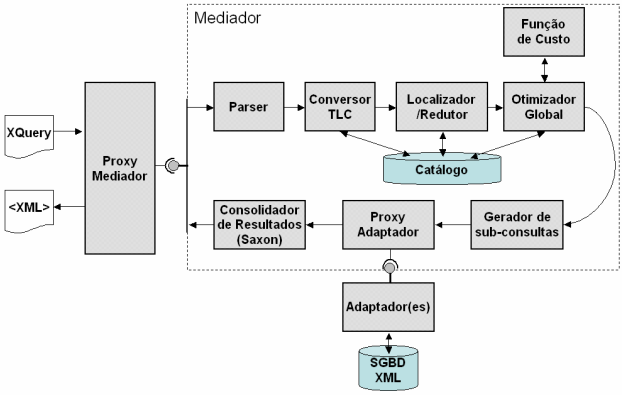

Figura 2: Diagrama de blocos dos componentes do Mediador.

2.1. Mediador

O Mediador é o principal componente da arquitetura, pois ele é responsável pelo processamento da consulta distribuída, realizando as etapas de decomposição, localização e otimização global da consulta, de acordo com a metodologia de processamento de consultas sobre bases XML distribuídas [Figueiredo et al. 2007].

O diagrama da Figura 2 segue a arquitetura básica para processamento de consultas apresentada em [Kossman 2000], estendida para o Mediador. Esta arquitetura possui uma série de módulos, responsáveis por partes isoladas do processamento da consulta distribuída, descritos a seguir.

Parser: responsável pela validação sintática da consulta XQuery.

Conversor TLC: módulo responsável pela representação algébrica TLC da consulta XQuery. Implementa o algoritmo de conversão da XQuery para a TLC descrito em [Paparizos et al. 2004]. Esta representação algébrica em TLC será utilizada pelas próximas fases do processamento da consulta.

Localizador/Redutor: responsável pela localização do plano algébrico da consulta. Essa etapa compreende duas atividades: (i) substituição das referências a coleções globais por referências a fragmentos destas coleções, e (ii) eliminação dos fragmentos irrelevantes ao resultado da consulta, utilizando a abordagem de redução do plano algébrico definida na nossa metodologia [Figueiredo et al. 2007].

Otimizador Global: responsável pela otimização global do plano algébrico. Na implementação do protótipo, desenvolvemos um otimizador que produz um conjunto de planos algébricos equivalentes a partir das réplicas dos fragmentos existentes no ambiente distribuído para descobrir, com o uso de uma função de custo, o plano de menor custo total dentre os planos gerados.

Função de Custo: função para a estimativa do custo de execução de um plano algébrico a partir dos dados estatísticos de cada fragmento, como o número de nodos, tamanho médio em *bytes* de cada nodo, parâmetros de seletividade, peso de leitura em disco, peso de comunicação, etc. Na implementação do protótipo utilizamos apenas o

peso da comunicação e a estimativa do número total de nodos do fragmento para o cálculo do custo do plano algébrico.

Gerador de Sub-consultas: responsável pela extração e composição das sub-expressões (sub-consultas) do plano algébrico otimizado. Cada sub-consulta é transformada em uma representação em XQuery e enviada para o seu Adaptador correspondente. A composição do resultado final será feita também por uma sub-consulta, que é executada pelo próprio Mediador sobre os resultados das sub-consultas remotas. As sub-consultas são criadas a partir de operações da TLC, através de um algoritmo inverso ao algoritmo de conversão da XQuery, produzindo uma consulta textual em XQuery a partir de expressões de operações algébricas.

Este componente merece uma atenção especial, já que ele é o grande responsável pela característica não intrusiva de nossa arquitetura. O SGBD XML nativo não precisa ser modificado para uso de nossa solução. Nossa ferramenta é sempre executada como uma camada acima do SGBD, interceptando as consultas e gerenciando os resultados. Deste modo, como não temos acesso ao processador de consultas interno do SGBD, precisamos transformar as sub-consultas TLC novamente em consultas XQuery, para que elas possam ser enviadas aos adaptadores e executadas por eles.

***Proxy* do Adaptador**: permite a comunicação entre o Mediador e os Adaptadores, através da execução dos protocolos para chamada de serviços *Web*. O *proxy* permite a definição do endereço do Adaptador que será invocado, tornando todo o processo de comunicação com o serviço *Web* transparente para o resto do Mediador.

Consolidador dos Resultados: responsável pela composição do resultado final. Em nossa implementação, a composição do resultado final é realizada através da execução de uma consulta XQuery local sobre os resultados das sub-consultas retornadas pelos Adaptadores. Utilizamos o processador de XQuery Saxon para execução da consulta em memória, sem necessidade de armazenamento dos resultados dos Adaptadores em disco. Se houver apenas uma sub-consulta, não será necessário compor resultados e o resultado final será o próprio resultado desta sub-consulta.

***Proxy* do Mediador**: permite a comunicação de um cliente com o Mediador, através da implementação dos protocolos de comunicação e da configuração de atributos específicos do Mediador.

2.2. Catálogo

O Catálogo armazena todas as informações necessárias para o processamento da consulta distribuída, em especial para a etapa de localização, como o nome e o *schema* das visões globais; os fragmentos que formam a visão global da coleção distribuída; a definição de cada fragmento; o endereço de cada Adaptador remoto que possui uma cópia do fragmento; estatísticas dos fragmentos, como número total de nodos, características de seletividade, etc. O Catálogo foi implementado como um conjunto de objetos em Java que pode ser serializado e desserializado em um documento XML para edição manual.

2.3. Adaptadores

Os Adaptadores são responsáveis pela execução das sub-consultas nos SGBDs XML a eles acoplados. O resultado de cada sub-consulta é retornado ao Mediador para

composição do resultado final. A única responsabilidade do Adaptador, além de implementar a interface de comunicação com o Mediador, é atualizar a localização do documento ou coleção XML sendo consultada, 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 fragmento e seu endereço completo no servidor local. Após a realização deste mapeamento, o Adaptador pode executar a consulta utilizando a interface ou API do banco de dados por ele acoplado ao ambiente. Na nossa implementação utilizamos o banco de dados XML nativo eXist. O diagrama de blocos dos componentes de um Adaptador é apresentado na Figura 3.

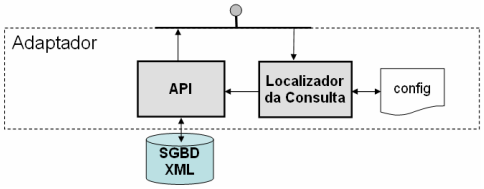

Figura 3: Diagrama de blocos dos componentes do Adaptador

3. Execução de Consultas sobre Bases Distribuídas

Consultas podem ser submetidas sobre as bases distribuídas de duas formas: através do *proxy* do Mediador; ou através de uma interface desenvolvida para testes do sistema e que permite a visualização gráfica da representação algébrica da consulta nas diferentes etapas de processamento da consulta.

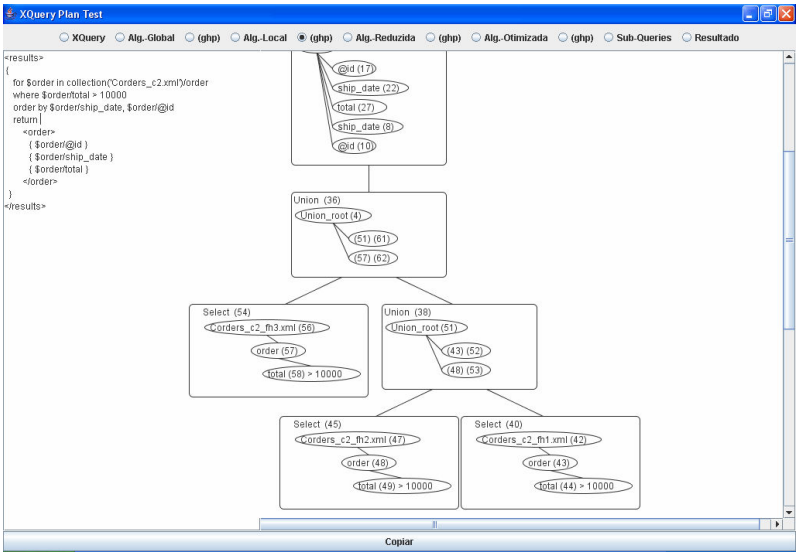

Figura 4: Interface gráfica para execução de consultas no Mediador.

O proxy do Mediador é uma classe que contém métodos que simplificam a comunicação de um cliente com o Mediador, através da implementação do protocolo de comunicação. Já a interface gráfica (Figura 4) para execução de consultas sobre o Mediador foi desenvolvida para que pudéssemos acompanhar a evolução da consulta submetida durante as etapas da metodologia de processamento de consultas. Através da interface, tem-se uma noção muito clara de todas as etapas, o que facilita o entendimento da metodologia. Nossa demonstração será focada nesta interface.

4. Conclusão

Este artigo apresentou um mediador que implementa uma metodologia para o processamento de consultas XQuery sobre bases de dados XML nativas distribuídas e fragmentadas [Figueiredo et al. 2007]. O mediador decompõe uma consulta XQuery em sub-consultas que são destinadas a adaptadores acoplados aos bancos de dados remotos. Estes adaptadores executam a sub-consulta através do SGBD XML a ele acoplado e retornam o resultado ao Mediador, que realiza a composição do resultado final. Consultas podem ser submetidas programaticamente através de uma classe *proxy* que se comunica com o Mediador ou manualmente através de uma interface gráfica que executa a consulta no Mediador e ainda exibe a representação algébrica da consulta submetida nas diferentes etapas de processamento.

Nossa arquitetura torna o processamento de consultas XQuery distribuídas totalmente transparente para o usuário final, além de ser não intrusiva e poder ser utilizada com qualquer SGBD. O fato de utilizarmos adaptadores nas bases locais faz com que seja possível inclusive utilizar bases não XML (relacionais, por exemplo). Basta, para isso, que o adaptador daquela base publique os dados no formato XML esperado por aquele fragmento.

Referências

Andrade, A., Ruberg, G., Baião, F., Braganholo, V., Mattoso, M. (2006) "Efficiently processing XML queries over fragmented repositories with PartiX", In: DATAX, p. 150-163, Munich, Germany.

Baru, C., Gupta, A., Ludaesher, B., Marciano, R., Papakonstantinou, Y., Pavel, V., Chu, V. (1999) "XML-Based Information Mediation with MIX", In: SIGMOD, p. 597-599. ACM Press.

Bremer, J.-M., Gertz, M. (2003) "On Distributing XML Repositories", In: WebDB, p. 73-78, San Diego, California.

Figueiredo, G. C., Braganholo, V., Mattoso, M. (2007) "A Methodology for Query Processing over Distributed XML Databases", In: Technical Report ES-710/07 (http://www.dcc.ufrj.br/~braganholo/artigos/RT-guilherme.pdf), p. 1-24, Rio de Janeiro, Brazil.

Gardarin, G., Mensch, A., Dang-Ngoc, T.-T., Smit, L. (2002) "Integrating Heterogeneous Data Sources with XML and XQuery", In: DEXA, p. 839-846. IEEE Computer Society.

Gertz, M., Bremer, J.-M. (2003) "Distributed XML Repositories: Top-down Design and Transparent Query Processing". Department of Computer Science.

Kossman, D. (2000) "The State of the Art in Distributed Query Processing", In: ACM Computing Surveys, v. 32, p. 422-469.

Lee, K., Min, J., Park, K., Lee, K. (2002) "A Design and Implementation of XML-Based Mediation Framework (XMF) for Integration of Internet Information Resources", In: Hawaii International Conference on System Sciences, v. 7, p. 202-211. IEEE Computer Society.

Özsu, M. T., Valduriez, P. (1999) "Principles of Distributed Database Systems". 2 ed., Prentice Hall

Paparizos, S., Wu, Y., Lakshmanan, L. V. S., Jagadish, H. V. (2004) "Tree Logical Classes for Efficient Evaluation of XQuery", In: SIGMOD, p. 71-82. ACM.

Re, C., Brinkley, J., Hinshaw, K. P., Suciu, D. (2004) "Distributed XQuery", In: IIWeb, p. 116-121, Toronto, Canada.

Suciu, D. (2002) "Distributed Query Evaluation on Semistructured Data", ACM TODS, v. 27, 1, p. 1-62.

Wiederhold, G. (1992) "Mediators in the Architecture of Future Information Systems", In Michael N.Huhns and Munindar P.Singh, Readings in AgentsMorgan Kaufmann