7ª Conferência da Associação Portuguesa de Sistemas de Informação, Universidade de Aveiro, 17 a 19 de Janeiro de 2007

Benchmark de bases de dados de suporte a serviços de informação

José Novais[1], Leonel Duarte do Santos [2]

1) Universidade do Minho – DSI, Guimarães, Portugal

josenovais@josenovais.com

2) Universidade do Minho – DSI, Guimarães, Portugal
leonel@dsi.uminho.pt

Resumo

A necessidade de fazer chegar informação a uma audiência cada vez mais vasta estimula o aparecimento de serviços com capacidades de recolha, armazenamento e difusão de informação, com o objectivo de facilitar a sua gestão, numa perspectiva de divulgação e de recolha, isto é, serviços de informação *online*. Para o bom funcionamento desses serviços, o armazenamento eficiente de informação assume um papel de grande relevância, pelo que se torna necessária a execução de testes para a selecção dos modelos mais adequados para esta tarefa. Neste trabalho são propostos um modelo e um sistema de testes que deverão ser capazes de permitir tirar conclusões sobre o modelo de armazenamento de informação (relacional ou XML) a adoptar pelo repositório de dados de suporte a serviços de informação *online*.

Palavras chave: Repositórios, *benchmarks*, BDs, serviços de informação *online*

1. Introdução

Actualmente, com o aumento da utilização da Internet e das tecnologias como um meio de comunicação privilegiado e com a diversificação das aplicações que vão sendo colocadas online, tem-se assistido à disponibilização de cada vez maiores quantidades de informação a um cada vez maior número de utilizadores. Nesta perspectiva, as entidades vão disponibilizando serviços com capacidades de recolha, armazenamento e difusão de informação com vista a facilitar a gestão da informação, quer numa perspectiva de divulgação quer de recolha. Estes serviços implicam normalmente a manipulação de grandes quantidades de informação e de sistemas capazes de executar esta tarefa eficazmente, aos quais pode ser dado o nome de repositórios. Desta forma, os repositórios são o que alimenta os serviços de informação. Para o armazenamento desta informação, e assumindo que tal é feito com recurso a bases de dados (BDs), existem actualmente dois modelos bastante divulgados: o relacional e XML. Com este artigo, pretende-se comparar estes modelos para o armazenamento de informação, tendo sido feito para isso um conjunto de testes cujo modelo e resultado são apresentados.

O artigo está organizado da seguinte forma. Inicialmente, é feita uma abordagem ao conceito de repositório, sendo exemplificadas áreas onde este é utilizado. Faz-se também uma análise a formas de armazenamento de informação que poderão ser utilizadas nos repositórios. A secção seguinte (a 3) apresenta ferramentas utilizadas para o teste de performance – *benchmarks*. De seguida, na secção 4 são definidos os cenários a considerar neste trabalho e na secção 5 é apresentado um possível modelo para a execução de testes. Na secção 6 é descrito um sistema no qual serão conduzidos os testes. Por fim, na secção 7 são apresentados resultados da execução dos testes e na secção 8 são delineadas as conclusões e o trabalho futuro.

2. Repositórios e armazenamento de informação

Repositório é um termo utilizado de várias formas em vários contextos. Um repositório pode ser visto de várias perspectivas como por exemplo espaço de armazenamento, simples ficheiros com informação, BDs ou sistemas complexos de gestão de informação. Como exemplos de áreas onde é utilizado o conceito de repositório temos o armazenamento digital de documentos, nomeadamente bibliotecas digitais, e o comércio electrónico. Actualmente, um sistema com bastante notoriedade na área das bibliotecas digitais é o DSpace [DSpace 2004]. É definido como sendo um repositório institucional *open source* que tem como objectivo armazenar e gerir material de investigação e educacional produzido por organizações ou instituições.

O conceito de repositório está também presente no comércio electrónico. Nesta área têm sido criados standards de comunicação baseados em XML [W3C 2004]. Um standard bastante divulgado nesta área é o ebXML [OASIS 2004]. No ebXML os repositórios armazenam vários tipos de informação como perfis de organizações (que são consultados pelas outras organizações com vista a encontrar um possível parceiro de negócio), definições de processos, componentes básicos, etc.

Em suma, o termo repositório pode ter várias utilizações em vários contextos. De uma forma geral, um repositório pode ser visto como um sistema onde a informação é armazenada. Poderá ser uma camada inferior onde assentam todas as outras que constituem as aplicações – camada de armazenamento – ou mesmo uma aplicação completa. Um aspecto importante é o facto de a informação não ser acedida directamente por utilizadores (humanos ou aplicações informáticas). O seu acesso e gestão são feitos exclusivamente através de uma camada superior, a qual abarca toda a lógica de acesso, atribuindo significado à informação armazenada. Desta forma, a abrangência do termo repositório pode variar, podendo ser considerado apenas como um depósito de dados ou conjuntamente com quaisquer outras camadas. Independente das partes constituintes de um repositório, algo que é comum a todas as abordagens de repositório é a sua capacidade básica de armazenar informação, o que implica a utilização de um modelo. Assumindo que isto é feito com recurso a BDs, existem várias opções: utilização de um modelo como o relacional, por objectos, ou outro, bem como a utilização de XML.

O modelo relacional, introduzido por Codd em 1970 [Codd 1970] é o modelo mais divulgado actualmente nos sistemas de gestão de BD. Com a massificação da utilização da Internet, o XML tem-se assumido como um standard com um papel preponderante na troca de informação. Esta tendência actual leva a que actualmente muitos fabricantes de software estejam a incorporar nos seus produtos suporte para XML, como é o caso das BDs. Esta nova forma de representação de informação estruturada apresenta novos desafios para o seu armazenamento exigindo conversões para a utilização com os modelos convencionais.

Existem dois modelos de documentos XML [Bourret 2005b]: centrado nos dados e centrado no documento. Os documentos do primeiro são caracterizados por possuírem uma estrutura regular e normalmente a ordem pela qual os vários elementos com o mesmo nível na estrutura são representados é irrelevante ao passo que os restantes caracterizam-se por possuir uma estrutura irregular e uma maior granulosidade na informação, isto é as unidades mais elementares de informação podem estar ao nível de elementos com conteúdo misto e a ordem com que os elementos são representados é normalmente relevante. Estes modelos influenciam a forma como o XML é armazenado de forma persistente em BD.

A tradução referida anteriormente poderá ser executada internamente na BD. A estas dá-se o nome de BDs com suporte para XML (XML-*enabled*) [Bourret 2005b]. Desta forma, estas são BDs convencionais que incluem internamente um mecanismo de tradução, dispensando assim a utilização componentes de software adicionais. Este tipo de abordagem está normalmente associado à utilização do modelo de documentos XML centrado nos dados. Actualmente, todos os grandes fabricantes de BDs incluem nos seus produtos suporte para o armazenamento e manipulação de XML.

Uma outra perspectiva para o armazenamento de XML em BDs é um novo tipo de BD, denominadas nativas XML. Neste tipo de BDs, a entrada, saída e armazenamento de informação é sempre no formato XML, não existindo qualquer tipo de conversões intermédias como nas soluções descritas anteriormente. Uma das grandes vantagens é o facto de os documentos serem armazenados e recuperados intactos, pelo que este tipo de BDs é apropriado para aplicações centradas nos documentos. No entanto, também são apropriadas para aplicações centradas nos dados, pois permitem a recuperação de fragmentos dos documentos, caso seja necessário. A utilização deste tipo de BDs considerada por alguns autores [Schmidt et al. 2001] como a melhor solução para o armazenamento e manipulação de documentos XML. A utilização deste tipo de BD permite a manipulação de qualquer tipo de documento mas de forma especialmente eficaz os documentos orientados ao documento.

Apesar de ambas manipularem XML, as BDs XML nativas e com suporte para XML apresentam diferenças [Bourret 2005b]:

- As nativas preservam a estrutura física dos documentos o que nem sempre é feito nas com suporte para XML.
- As nativas guardam qualquer ficheiro, mesmo sem conhecer o seu *schema*, o que pode não ser possível nas com suporte para XML.
- Nas nativas, a informação apenas é acessível em formato XML, ao passo nas com suporte para XML poderá existir outra forma de aceder à informação.

3. Benchmarks

Existe uma grande diversidade de sistemas que manipulam e armazenam informação, baseada em diversos modelos como o relacional, orientado por objectos ou XML, com desempenhos e características variados. Desta forma, a necessidade de comparar desempenhos surge de uma forma natural.

Um *benchmark* é um programa utilizado para testar a performance de software, hardware ou um sistema [Collin 2002]. Mais concretamente, um *benchmark* para BDs pode ser visto como um conjunto de instruções utilizadas para medir e comparar o desempenho de dois ou mais sistemas de gestão de base de dados. Isto é feito recorrendo à execução de experiências bem definidas cujas medidas de desempenho serão usadas para prever o desempenho do sistema [Seng et al. 2005]. Desta forma, na especificação de um *benchmark* são considerados 3 componentes principais [Menascé 2002]: o sistema a ser testado (SUT[1]), a carga de trabalho submetida ao SUT (workload), que consiste nas operações de teste, e uma ou mais métricas que são resultantes da monitorização e avaliação do desempenho do SUT o qual inclui a BD de teste (Figura 1). Exemplos de métricas são *throughput*, tempo de resposta, tamanho da BD e relação performance / custos de manutenção.

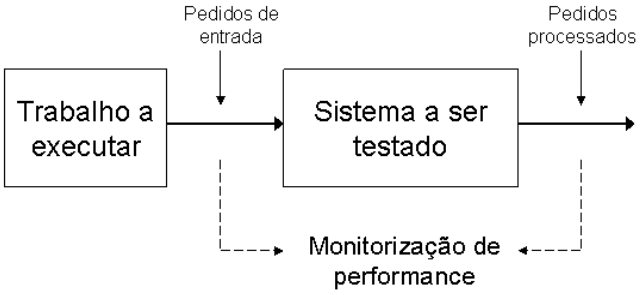

Figura 1 – Estrutura de um *benchmark* (adaptado de [Menascé 2002])

Os sistemas, que serão alvo de teste dos *benchmarks*, são construídos de forma a darem resposta a problemas específicos de determinados domínios de aplicação. A utilização de *benchmarks* genéricos, em sistemas de vários domínios de aplicação, não conduz a resultados fiáveis dada a natureza dos sistemas, já que uns estarão mais aptos para certo tipo de tarefas que outros. Desta forma em [Gray 1993] defende-se o uso de *benchmarks* específicos para os diversos domínios, os quais deverão respeitar os princípios básicos: relevância (deverá capturar as características do sistema a ser medido), portabilidade (deverá facilmente ser implementado em diferentes sistemas), escalabilidade (deverá ter a possibilidade de testar várias BDs em diferentes sistemas) e simplicidade (deverá ser perceptível de forma a ser credível).

Os *benchmarks* podem ainda classificar-se como sintéticos ou empíricos [Seng et al. 2005] ou uma mistura entre ambos. *Benchmarks* sintéticos emulam aplicações típicas de um determinado domínio, tanto a nível de operações como de BDs de teste, ao passo que os empíricos utilizam operações de testes e informação reais. Para sistemas baseados em modelos mais comuns como o relacional, foram desenvolvidos ao longo do tempo diversos testes de performance que estimularam a comparação de performance e consequentemente ao aperfeiçoamento dos sistemas. Com o aparecimento de sistemas capazes de lidar com XML, foi necessário criar testes que tenham em consideração novos desafios colocados por este modelo. Estes, em [Schmidt et al. 2002] são especificados como:

- Preservação de ordem textual das várias estruturas que compõem os documentos manipulados.
- Utilização de strings como tipo de dados básico cujo armazenamento e manipulação podem levantar problemas aos sistemas e podem entrar em conflito com a forma como os tipos de dados são tratados pelas linguagens de *query*.
- *Queries* que envolvem a manipulação de estruturas hierárquicas complexas e a preservação de ordem requerem a execução de operações dispendiosas principalmente quando o XML está armazenado numa estrutura relacional.
- Utilização de *Schemas* pouco rígidos de validação dos documentos. Isto não só não facilita a tarefa de escrita de *queries* complexos por parte dos utilizadores, como também pode levantar problemas a nível de optimização do armazenamento da informação, introduzindo valores NULL que podem fazer aumentar o tamanho da BD.

Existem vários *benchmarks* utilizados em BDs relacionais, tais como Wisconsin [DeWitt 1993], AS3AP [Turbyfill et al. 1989], mais orientados para o teste de aspectos importantes do servidor de BD, sendo constituídos por uma BD de teste e alguns *queries*.

Outros *benchmarks* importantes são os definidos pelo *Transaction Processing Performance Council* (TPC), um consórcio de vários fabricantes de *software* e *hardware* sem fins lucrativos. Este consórcio define *benchmarks* para vários fins, como processamento de transacções online (OLTP), comércio electrónico, apoio à decisão, etc. Os resultados da utilização destes *benchmarks* são publicados regularmente. Dois *benchmarks* definidos pelo TPC são o TPC-C [TPC 2005] e o TPC-W [TPC 2002]. Estes *benchmarks*, que como referem as suas especificações poderiam ser utilizados por qualquer SGBD, não só SGBD relacionais, têm grande aceitação na indústria.

Para BDs XML existem dois tipos de *benchmarks*: micro-*benchmarks* e aplicacionais. Todos eles consistem num conjunto de informação (BD de teste) que pode ter várias versões com tamanhos diferentes e sobre o qual serão executados um conjunto predefinido de *queries*.

Os micro-*benchmarks*, como o Michigan Benchmark [Runapongsa et al. 2003], são desenhados de forma a testarem componentes específicos do sistema com vista a isolar e ajudar a corrigir

[1] Da língua inglesa: *System under test*.

certos problemas. Pretendem explorar o impacto na performance do sistema das características mais importantes do XML, dispondo de uma BD de teste heterogénea, não inspirado numa qualquer aplicação real, sobre o qual são especificados *queries* especialmente desenhados para testar componentes elementares da linguagem de *query* (como selecção, joins, etc). Com um *benchmark* deste tipo torna-se possível aperfeiçoar as operações mais básicas ao nível do SGBD.

Os *benchmarks* aplicacionais, por seu lado funcionam a um nível mais elevado, pretendendo medir a performance do sistema como um todo e não questões específicas. Cada um deles dispõe de uma BD de teste, que pode ser ou não inspirada numa aplicação real, sobre a qual são definidos *queries* que pretendem abranger o maior número possível de características da linguagem de *query*. Exemplos destes *benchmarks* são o XOO7 [Li et al. 2001], Xmark [Schmidt et al. 2002], XBench [Yao et al. 2004] e o XMach-1 [Böhme e Rahm 2001].

4. Cenários

A informação poderá estar modelada segundo o modelo relacional, devidamente inserida em registos de tabelas numa BD. No entanto, poder-se-á optar por a representar não utilizando o modelo relacional mas XML.

A informação representada em XML poderá ser encarada de duas formas distintas, tal como referido anteriormente: centrado nos dados ou centrado nos documentos. Apesar de não existirem regras rígidas, cada uma destas hipóteses poderá estar mais adequada para utilização para diferentes tipos de BDs. Desta forma, uma visão centrada nos dados poderá ser mais adequada para utilização numa BD com suporte para XML (XML *enabled*) ao passo que uma visão centrada no documento poderá ter melhor desempenho numa BD XML nativa [Vakali et al. 2005].

No contexto do presente trabalho, terá interesse estudar cenários onde sejam utilizadas para o armazenamento físico de informação do repositório de suporte a um serviço de informação, BDs relacional (para a representação da informação no modelo relacional), XML *enabled* ou XML nativa (para suportar informação representada em XML). O estudo destes cenários deverá ter como base uma grande quantidade de informação de forma a atribuir o maior realismo possível ao estudo a efectuar. Desta forma, a quantidade de informação será outro aspecto com interesse no estudo, além do tipo de BD, podendo ser considerados cenários com quantidades distintas de informação.

Este trabalho centra-se numa perspectiva de divulgação de informação (isto é, consultas) deixando-se de parte os aspectos de recolha de informação. Desta forma, tomou-se como ponto de partida uma BD (relacional) real com informação respeitante a currículos de investigadores, cuja informação foi posteriormente convertida para XML e inserida nas respectivas BDs.

5. Modelo de testes

Para tirar conclusões sobre os modelos mais adequados para a utilização em repositórios torna-se necessária a execução de testes, isto é de *benchmarking*. Torna-se necessário conceber e implementar um sistema, executar testes e interpretar os resultados.

Trabalhos como [Böhme e Rahm 2001; Li et al. 2001; Schmidt et al. 2002; Yao et al. 2004] abordam questões relacionadas com o processamento de XML nomeadamente o processamento de *queries* que pretendem abranger o maior número possível de características da linguagem de *query*, pretendendo medir a performance do sistema como um todo, existindo um foco exclusivo em BDs de XML ao passo que outros como [Turbyfill et al. 1989; DeWitt 1993] abordam questões de processamento de *queries* em sistemas relacionais. Na presente situação o objectivo não é o mesmo que o dos trabalhos referidos uma vez que não se está particularmente

interessado em explorar questões de processamento de XML ou características das linguagens de *query* ou ainda a capacidade de processamento destes pelos sistemas. A atenção foca-se antes na escolha de um modelo, XML ou relacional, que melhor resposta às necessidades de um serviço de informação online para divulgação de currículos de investigadores. Desta forma, para os testes de performance, parte-se deste cenário real previamente definido e irá tentar-se definir os aspectos principais a testar. É uma abordagem diferente de outros trabalhos (por exemplo [Böhme e Rahm 2001; Schmidt et al. 2002; TPC 2002; TPC 2005] dado que não é feito com vista a simular uma dada aplicação de um determinado domínio com o objectivo de abordar o maior número de aspectos relevantes possível, mas é feito com os condicionalismos inerentes à aplicação definida pelo contexto. O maior destes condicionalismos é, como referido, o facto de se estar a considerar dois modelos distintos, XML e relacional.

Com vista à realização de testes é necessário criar pesquisas que se pretendem o mais simples possível. Cada pesquisa tem 3 versões, uma para cada tipo de BD envolvida, na respectiva linguagem de *query* utilizada. Uma questão importante na criação destas pesquisas é o facto de os modelos a testar serem bastante diferentes. Aspectos como a recuperação de documentos na sua forma original, travessias em documentos estruturados ou a composição de novos documentos com base na informação da BD, apenas fazem sentido quando se trata de XML, pelo que as pesquisas terão de abordar aspectos comuns, como pesquisas em texto integral, manipulação de resultados de grandes dimensões ou a utilização de *joins*.

Ao invés de tentar criar pesquisas que abrangessem uma lista completa de aspectos comuns aos modelos em causa, optou-se por definir um conjunto de requisitos para as pesquisas, as quais ao serem implementadas poderão incluir obrigatoriamente pelo menos uma das características referidas.

Desta forma, podemos considerar que um serviço de informação *online* com um perfil centrado na consulta de informação deverá permitir as seguintes funcionalidades básicas:

1. Acesso directo aos itens por chave (pesquisa por chave)
2. Obtenção de resultados ordenados segundo determinados critérios.
3. Obtenção de indicadores com base na informação armazenada.
4. Suporte eficiente para resultados de grandes dimensões (elevado número de itens).
5. Pesquisas textuais

Com base nestas características básicas foram construídas as pesquisas.

A execução de testes pressupõe a medição de parâmetros relevantes. Um parâmetro importante, e que será medido na execução dos testes, é o *thoughput* do sistema que se pode definir como o número de pesquisas, baseadas nas características propostas, para o qual se obteve uma resposta por segundo.

6. Sistema de testes

Para a execução de testes torna-se necessária a criação de um sistema que tenha em atenção as funcionalidades básicas descritas. A sua arquitectura (Figura 2) é inspirada nas propostas em [Böhme e Rahm 2001; TPC 2002] sendo igualmente baseada numa aplicação *web*. É composta por três componentes: BD, servidor aplicacional e clientes, não existindo a necessidade do componentes adicionais externos aos SUT como o que faz recolha e upload de informação para a BD (loader) [Böhme e Rahm 2001] ou gateway de pagamentos [TPC 2002].

Na presente situação o sistema irá utilizar, em testes distintos, BDs para a manipulação de XML (*enabled* e nativa) e relacional, todas elas contendo informação equivalente representada nos respectivos modelos.

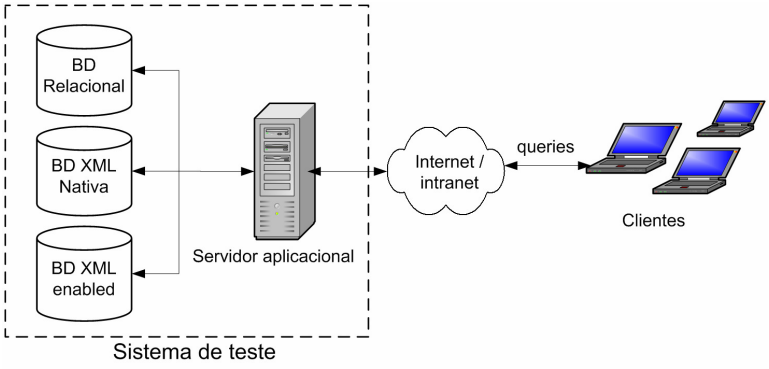

Figura 2 – Arquitectura do sistema de teste

No servidor aplicacional (que tal como o servidor de BD poderia ser na realidade constituído por vários servidores) é executada uma aplicação *web* com base num servidor *web* (http) bem como outros componentes de software necessários para acesso às BDs. O conjunto do servidor aplicacional e de BD constitui o sistema de teste, excluindo os clientes pelo que não se considera nos resultados dos testes atrasos devidos a comunicações e a processamentos feitos a nível dos clientes. Desta forma, os clientes conectam-se ao sistema de teste via o protocolo http e todos os resultados são medidos a nível do servidor aplicacional tendo apenas em conta o intervalo de tempo entre a chegada do pedido e o final do seu processamento (ignorando o tempo que o resultado demoraria a chegar ao cliente). Para evitar atrasos relativos à rede, os servidores aplicacional e de BDs serão os mesmos, com dados armazenados localmente. Com vista a evitar inconsistências nos resultados, não é permitida *cache* ao nível do servidor. Para simular diferentes cenários são utilizadas diferentes BDs com diferente número de registos, bem com é alterado o número de clientes e consequentemente o número de pedidos ao servidor aplicacional. Os valores vão de 25 a 125 em intervalos de 25, tendo sido criada uma aplicação que simula os pedidos simultâneos à aplicação. Ao *schema* da BD original foi feita uma simplificação, mas mantendo a estrutura básica de um currículo. O XML *Schema* foi criado com base neste *schema* relacional simplificado. Apesar de terem sido utilizadas ferramentas automáticas para esta tarefa, a versão final teve uma revisão manual, tendo sido adoptada uma abordagem baseada em atributos para representar as colunas das tabelas e usando os nomes das colunas no nome dos atributos [Williams et al. 2000]. Após a criação do XML *Schema* foi feita a conversão da informação relacional para o formato XML, respeitando este *schema*. Esta conversão foi feita inteiramente com recurso a ferramentas automáticas.

Para responder aos vários cenários idealizados para os testes é necessário que o tamanho da BD de teste seja constituído por um número de registos variável. Desta forma a BD de teste inicial foi completada utilizando ferramentas especialmente desenvolvidas. Os valores são 115.000, 517.000 e 1 milhão de registos.

7. Resultados dos testes

Facilmente se observa nos gráficos da Figura 3, que a BD relacional apresenta um comportamento quase sempre superior nesta carga a que foi submetida. Isto apenas não acontece na BD de maiores dimensões com o menor número de clientes por uma margem pequena relativamente às outras BDs, pelo que se poderá considerar pouco significativo. Os testes realizados revelam assim que existem notoriamente 2 tipos de desempenho: alto para a BD relacional e baixo para as BDs de XML.

No caso das BDs pequena e média, a diferença de desempenho entre a BD relacional e as de XML é significativa chegando a ser mais de 6 vezes superior no caso da BD média. Isto é mais relevante com o aumento de clientes. Nota-se claramente dois grupos de desempenho: alto para a relacional e baixo para as BD de XML. Entre as BDs de XML, nota-se um desempenho que se

pode considerar idêntico no caso das BDs médias. No caso da BD pequena a com suporte para XML leva vantagem chegando a ter um desempenho cerca de 50% superior à nativa XML.

Na BD grande esta diferença de desempenho relativa entre os vários tipos de BDs não é tão acentuada, estando todos os valores relativamente próximos, mas novamente com a relacional em vantagem. Nota-se no entanto uma queda de desempenho bastante acentuada relativamente às BDs média e pequena.

A análise de variância dos resultados revela que os factores que mais influenciaram os resultados foram o tamanho da BD (33,52%) e o tipo (31,97%), ao passo que o número de clientes a influência foi menor (5,3%). Um outro aspecto que é observável nos valores obtidos é uma quebra acentuada de desempenho na BD de maiores dimensões, para todos os tipos. Mesmo assim a relacional está em vantagem. Esta quebra poderá estar relacionada com configurações ao nível do servidor onde foram executados os testes. No entanto, não foi possível executar testes que permitissem comprovar esta hipótese, pelo que estudos adicionais poderão de futuro clarificar este aspecto.

8. Conclusão e trabalho futuro

Neste documento foi introduzido um conceito de repositório bem como diferentes abordagens ao armazenamento de informação. Foi também proposto um modelo e um sistema de testes com os quais se pretende averiguar qual o modelo mais adequado para o armazenamento de informação nos repositórios. Para isto torna-se necessário a execução de testes, isto é, *benchmarking*. Estes testes terão de respeitar duas condições fundamentais para que possam ser credíveis. Deverão ter como base cenários de aplicação reais e não deverão abordar aspectos que não façam sentido em ambos os modelos considerados.

Os resultados obtidos nestes testes demonstram uma superioridade inegável da BD relacional. A escalabilidade de uma solução baseada no modelo relacional é bastante superior comparativamente com uma solução baseada em XML. Isto tem duas faces. Por um lado, as BDs relacionais apresentam décadas de desenvolvimento ao passo que a tecnologia XML é bastante mais recente. Desta forma, as BDs XML ainda não apresentam uma maturidade suficiente para possam ser colocadas ao nível das relacionais e para que o modelo XML possa ser uma alternativa ao modelo relacional. Por outro lado, a informação utilizada neste trabalho inicialmente estava numa BD relacional, tendo sido convertida para XML. Apresenta um perfil que se enquadra perfeitamente no modelo relacional, isto é, representação em tabelas e onde não existe uma noção de hierarquia. Estas situações são portanto mais favoráveis à utilização do modelo relacional [Lapis 2005], o que terá penalizado a BD de XML. É importante ter em atenção que existem situações onde uma BD de XML é mais apropriada, como gestão de informação orientada a documentos, integração de informação (integração entre aplicações, com informação obtida de diversas fontes), gestão de informação não estruturada ou evolução do *schema* [Bourret 2005a]. Desta forma, seria bastante interessante repetir os testes deste trabalho mas com um tipo de informação mais apropriada ao modelo XML (com características como a possibilidade de alterações ao seu *schema* ou com hierarquias). Uma solução que adopte ambos os modelos aproveita o melhor de cada um deles: a rapidez associada ao modelo relacional (como demonstrado nos testes efectuados) e a capacidade de manipulação de documentos e lidar com *schemas* complexos ou que mudam frequentemente. Neste caso, parte da informação está armazenada em tabelas relacionais e outra parte em documentos XML, por exemplo a informação não estruturada cujo *schema* é pouco rígido e que poderá mudar mais frequentemente. Nesta perspectiva, o uso dos modelos relacional ou XML não pode ser visto numa perspectiva competitiva, mas sim complementar.

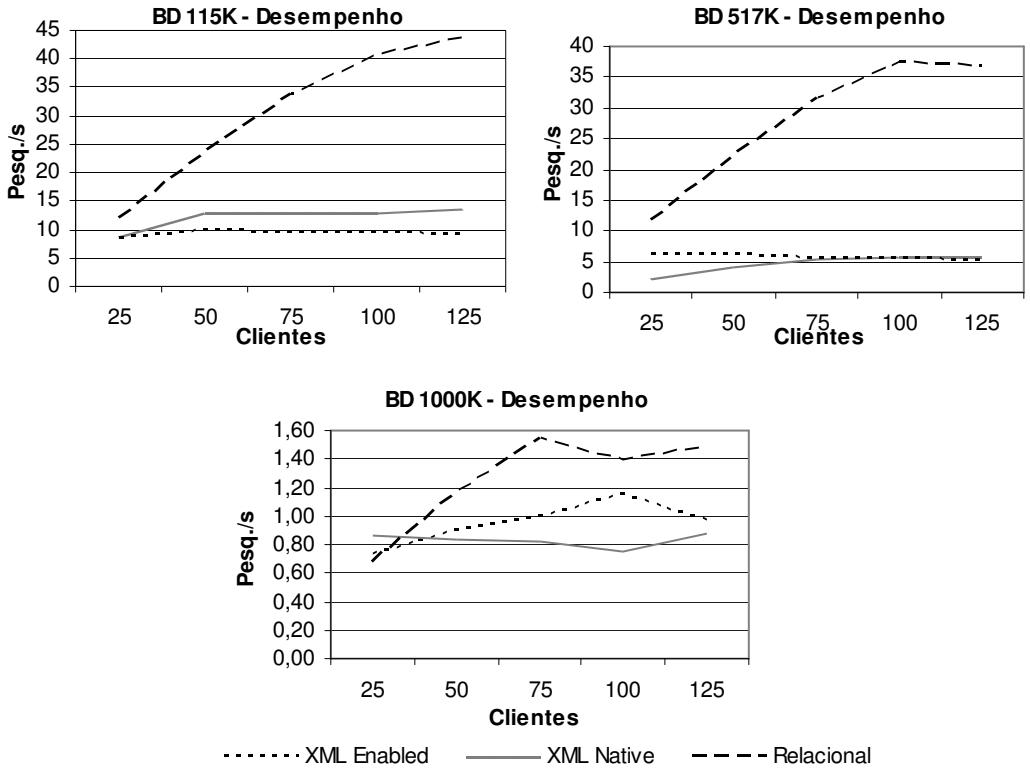

Figura 3 – Resultados dos testes

9. Referências

Böhme, T. e E. Rahm (2001). XMach-1: A Benchmark for XML Data Management. Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), Oldenburg, Germany, Springer-Verlag.

Bourret, R. (2005a). Native XML Databases in the Real World. XML 2005 Conference & Exposition, Atlanta, USA.

Bourret, R., XML and Databases, http://www.rpbourret.com/xml/XMLAndDatabases.htm, (15/01/2006), 2005b

Codd, E. F. (1970). "A relational model of data for large shared data banks." Communications of the ACM 13(6): 377 - 387.

Collin, S. M. H. (2002). Dictionary of Information Technology, Third Edition, Peter Collin Publishing.

DeWitt, D. J. (1993). The Wisconsin Benchmark: Past, Present, and Future. The Benchmark Handbook. J. Gray (Eds.), Morgan Kaufmann Publishers, Inc.

DSpace, DSpace Federation, http://www.dspace.org/, (01/02/2004), 2004

Gray, J. (1993). Database and Transaction Processing Performance Handbook. The Benchmark Handbook, Morgan Kaufmann Publishers, Inc.

Lapis, G. (2005). XML and Relational Storage - Are they mutually exclusive? XTech 2005, Amsterdam, The Netherlands.

Li, Y. G., S. Bressan, G. Dobbie, Z. Lacroix, M. L. Lee, U. Nambiar e B. Wadhwa (2001). XOO7: Applying OO7 Benchmark to XML Query Processing Tools. Conference on Information and Knowledge Management, Atlanta, Georgia, USA, ACM Press, New York, NY, USA.

Menascé, D. A. (2002). TPC-W: A Benchmark for E-Commerce. IEEE Internet Computing. 6**:** 83 - 87.

OASIS, ebXML, http://www.ebxml.org/, (01/02/2004), 2004

Runapongsa, K., J. M. Patel, H. V. Jagadish, Y. Chen e S. Al-Khalifa (2003). The Michigan Benchmark: Towards XML Query Performance Diagnostics. 29th VLDB Conference, Berlin, Germany.

Schmidt, A., F. Waas, M. Kersten, M. J. Carey, I. Manolescu e R. Busse (2002). XMark: A benchmark for XML Data Management. 28th VLDB Conference, Hong Kong, China.

Schmidt, A., F. Waas, M. Kersten, D. Florescu, M. J. Carey, I. Manolescu e R. Busse (2001). "Why and how to benchmark XML databases." ACM SIGMOD Record 30(3): 27 - 32.

Seng, J.-L., S. B. Yao e A. R. Hevner (2005). "Requirements-driven database systems benchmark method." Decision Suppot Systems 38(4): 629 - 648.

TPC (2002) Transaction Processing Performance Council - TPC Benchmark W, ver. 1.8,

TPC (2005) Transaction Processing Performance Council - TPC Benchmark C, ver. 5.4,

Turbyfill, C., C. Orji e D. Bitton (1989). AS3AP:a comparative relational database benchmark. 34th IEEE Computer Society International Conference, San Francisco, CA.

Vakali, A., B. Catania e A. Maddalena (2005). "XML Data Stores: Emerging Practices." Internet Computing, IEEE 9(2): 62-69.

W3C, Extensible Markup Language (XML) 1.0 (Third Edition), http://www.w3.org/TR/REC-xml, (13/02/2005), 2004

Williams, K., P. Dengler, J. Gabriel, A. Hoskinson, M. Kay, M. Brundage, T. Maxwell, M. Ochoa, J. Papa e M. Vanmane (2000). Professional XML Databases, Wrox Press Ltd.

Yao, B. B., M. T. Özsu e N. Khandelwal (2004). XBench Benchmark and Performance Testing of XML DBMSs. 20th International Conference on Data Engineering, Boston, Massachusetts, U.S.A., IEEE Computer Society.