



Análise de desempenho para o armazenamento versionado de conteúdos binários com acesso eficiente via Web

Corneli Gomes Furtado Júnior – DETI/CT – Universidade Federal do Ceará,
junior.corneli83@gmail.com

Danielo Gonçalves Gomes – DETI/CT – Universidade Federal do Ceará,
danielo@ufc.br

José Marques Soares - DETI/CT - Universidade Federal do Ceará,
marques@ufc.br

Giovanni Cordeiro Barroso – Depto de Física – Universidade Federal do Ceará,
gcb@fisica.ufc.br

Abstract. *The free availability of e-learning objects (ELOs) is clearly growing on the Web. A good ethical conduct requires that modified versions be made available along with the original, keeping the proper credit of authorship. It is not clear, however, repositories that provide appropriate tools for version control. Aiming to specify a repository that allows not only the reuse, but also efficient access to ELOs, it presents a comparative analysis of systems performance Subversion (SVN), MySQL and PostgreSQL according to several metrics. To evaluate the positives and negatives of different techniques for versioning the Web, presents a comparative analysis of performance between the SVN, MySQL and PostgreSQL. The best results are obtained in hybrid solutions, with the release of the latest in comics and earlier versions in SVN.*

Keywords: *distance education, e-learning objects (ELOs), version control, authorship.*

Resumo: *A livre disponibilização de conteúdos didáticos digitais (CDDs) vem crescendo de maneira expressiva na Web. A boa conduta ética requer que versões modificadas sejam disponibilizadas juntamente com a original, guardando os devidos créditos de autoria. Não se percebe, entretanto, repositórios que forneçam instrumentos apropriados para o controle de versões. Visando especificar um repositório que permita não só o reuso, mas também o acesso eficiente a CDDs, são avaliados os aspectos positivos e negativos de técnicas distintas para versionamento na Web, apresentando-se uma análise comparativa de desempenho entre o Subversion (SVN), MySQL e PostgreSQL. Os melhores resultados são obtidos em soluções híbridas, com a disponibilização da última versão em BDs e as versões anteriores no SVN.*

Palavras-chave: *educação à distância, conteúdos didáticos digitais (CDDs), controle de versão, autoria.*

1. Introdução

A idéia da criação e disponibilização de conteúdos de apoio ao processo de ensino e aprendizagem a partir da combinação de componentes reutilizáveis armazenados em bases de dados vem sendo discutida há tempo (**Erro! Fonte de referência não encontrada.**). Nos últimos anos, o Ministério da Educação (MEC) têm financiado tanto o desenvolvimento de ferramentas como o trabalho de catalogação e disponibilização de conteúdos através de



editais específicos. Dentre os projetos financiados pelo MEC podemos destacar: o Rived (**Erro! Fonte de referência não encontrada.**); Portal do Professor (**Erro! Fonte de referência não encontrada.**); Fabrica Virtual (**Erro! Fonte de referência não encontrada.**) e o InterRed (**Erro! Fonte de referência não encontrada.**).

Com o objetivo de evitar trabalhos redundantes, muitas vezes desenvolvidos dentro das mesmas instituições de ensino, conteúdos didáticos disponibilizados podem ser reaproveitados e associados a outros elementos para compor um novo conteúdo, permitindo aos autores maior produtividade e economia de tempo (**Erro! Fonte de referência não encontrada.**).

O versionamento de conteúdo é uma atividade de grande utilidade em diversas áreas do conhecimento. É útil, por exemplo, para acompanhar a evolução de uma pesquisa, o desenvolvimento de um código fonte ou as modificações de um conteúdo didático digital¹ (CDD). Versionamento, no âmbito dos CDDs, deve ser entendido como a operação de criar novas versões de um material didático a partir de um conteúdo pré-existente. Ferramentas que provêem essas funcionalidades são conhecidas por sistemas de controle de versão (SCV). Dentre os SCV mais populares podemos citar o Subversion (**Erro! Fonte de referência não encontrada.**) e o CVS (**Erro! Fonte de referência não encontrada.**).

Uma solução de baixo custo para proporcionar versionamento de CDDs é a utilização de um SCV já existente integrado aos tradicionais sistemas de armazenamento utilizados em repositório de conteúdos.

Visando especificar um repositório que permita não só o reuso, mas também o acesso eficiente a CDDs, neste trabalho são avaliados os aspectos positivos e negativos de técnicas distintas para versionamento na Web, apresentando-se uma análise comparativa de desempenho entre o SVN, MySQL e PostgreSQL. Usando cenários de simulação para *upload* e *download* de CDDs e utilizando um *benchmark* desenvolvido especificamente para este fim, a análise realizada considera como métricas o tempo de inserção e de resgate de conteúdos, o consumo de memória e de CPU e o total de espaço em disco utilizado.

Este documento é organizado da seguinte maneira: após a introdução realizada nesta seção, a motivação é discutida na seção **Erro! Fonte de referência não encontrada.**. Na seção **Erro! Fonte de referência não encontrada.**, apresentam-se trabalhos relacionados ao reuso e versionamento de conteúdos e referentes à análise de desempenho em sistemas de armazenamento. A metodologia e os experimentos realizados são documentados na seção **Erro! Fonte de referência não encontrada.**. Os resultados e as conclusões são, respectivamente, mostrados nas seções **Erro! Fonte de referência não encontrada.** e **Erro! Fonte de referência não encontrada.**.

2. Motivação

A criação e disponibilização de conteúdos a partir de versões preexistentes prevêm o enriquecimento dos conteúdos postados, através da reutilização de materiais de qualidade, visando extinguir a duplicação de esforços e possibilitando o incremento da produção de conhecimento humano.

Com forte inspiração neste princípio, foi concebido o InterRed, um repositório que visa o incentivo ao compartilhamento e ao reuso CDDs gerados por professores dos

¹Objeto composto por texto, imagem, som, vídeo, animação ou aplicação digitalizados contendo conteúdos pertinentes e úteis na educação presencial ou à distância



Institutos Federais de Educação Ciência e Tecnologia (IF). O projeto, financiado pela SETEC-MEC teve início com os IFs da região nordeste, sendo, em seguida, expandido para outras regiões do país (**Erro! Fonte de referência não encontrada.**).

Um dos requisitos do InterRed é que os conteúdos postados, quando modificados, dêem origem a novas versões do mesmo conteúdo e que estes sejam retornados ao repositório, de maneira que seja possível a manutenção do histórico da evolução e reutilização desse material. Na primeira versão do InterRed, a postagem de uma nova versão do conteúdo era controlada pelo próprio usuário e inserida no repositório independentemente da anterior, sendo armazenado sempre em um banco de dados.

Devido ao potencial crescimento do espaço requerido para armazenamento em disco devido a disponibilização de novas versões de CDDs, uma nova arquitetura foi elaborada para o InterRed visando o controle de versões como uma funcionalidade da própria ferramenta. Na nova implementação do InterRed, o Sistema Gerenciador de Banco de Dados (SGBD) foi substituído pelo SVN, visando a otimização do espaço requerido para conteúdos versionados. Entretanto, tal substituição não levou em conta o desempenho do SVN, mostrando-se ineficiente quanto ao tempo de utilização de CPU e consumo de memória RAM, impactando, assim, no tempo de resposta para recuperação dos conteúdos armazenados, funcionalidade mais executada pelos usuários do InterRed.

Dessa maneira, o estudo realizado neste trabalho permitiu estabelecer as características de configuração mais adequadas para conceber uma nova arquitetura para o InterRed, podendo ser generalizada para qualquer sistema que possua como requisito o armazenamento versionado de conteúdos binários.

3. Trabalhos Relacionados

Sendo a produção de conteúdos e de recursos didáticos de qualidade para apoio ao processo de ensino e aprendizagem uma atividade de elevado custo, recomenda-se que estes valorizem o reaproveitamento e a interoperabilidade entre diferentes plataformas (**Erro! Fonte de referência não encontrada.**).

Algumas equipes do Fábrica Virtual utilizam um arquivo externo ao objeto para representar sua arquitetura de informação, que é constituída de metadados. Metadados de Objetos de Aprendizagem (OAs), de acordo com Girardi (2004), são informações que definem características como: conteúdo, objetivos, autor(es), idioma, data, versão, etc. Os metadados podem ser descritos em XML (Extensible Markup Language), que é uma linguagem de marcação extensível, permitindo definir e compor elementos específicos de um domínio. Por ser extensível, todo documento XML deve obedecer a regras pré-estabelecidas.

A importância de metadados para a descrição de OAs está na possibilidade de localização, armazenamento e (re-)utilização dos mesmos. Os metadados também facilitam o compartilhamento e o intercâmbio de OAs. Para isso, existem padrões que facilitam a troca de dados entre os OAs que utilizam o mesmo padrão, permitindo o compartilhamento de dados.

Muzio (**Erro! Fonte de referência não encontrada.**) descreve suas experiências práticas na criação, armazenamento, uso e reuso de conteúdos didáticos no CEDAR². Cada conteúdo produzido recebe um conjunto de metadados, definidos através de palavras-chave, e que não

² Centre for Economic Development and Applied Research

seguem um padrão único. Essa abordagem permite o desenvolvimento de conteúdos reutilizáveis, que darão origem a novos conteúdos. CEDAR reconhece a necessidade da criação de um padrão para descrever o material produzido, estando consciente da necessidade de uma nova catalogação dos objetos posteriormente.

O LOM (Learning Object Metadata) é um padrão que segue os propósitos genéricos de metadados, possibilitando que objetos educacionais desenvolvidos, organizados e armazenados nesse padrão possam ser recuperados (**Erro! Fonte de referência não encontrada.**). Cabe salientar que o LOM é utilizado em padrões mais modernos e abrangentes, como SCORM (**Erro! Fonte de referência não encontrada.**), além de recomendado pelo RIVED. Apesar disso, nem todas as equipes seguem ou adotam um padrão de metadados.

Os conteúdos postados no InterRed podem ser reaproveitados, modificados ou adaptados a propósitos particulares, sem necessariamente precisar passar por um novo processo de desenvolvimento integral. Nesse sentido, o versionamento configura-se como uma técnica adequada ao desenvolvimento de revisões diversas de um conteúdo didático. Cada revisão pode ser uma cópia da anterior, adaptada a públicos específicos ou desenvolvida com tecnologia diferente, permitindo a sistematização do mecanismo de compartilhamento e devolução ao repositório das modificações propostas, contribuindo com a democratização do acesso às informações (**Erro! Fonte de referência não encontrada.**).

Nos trabalhos relacionados precedentemente, nota-se que o reuso é mais fortemente centrado na representação do conteúdo digital do que em aspectos relacionados à sua composição em termos de estrutura de dados. Neste trabalho, privilegia-se o tratamento dado ao conteúdo binário (vídeos, apresentações em forma de slides eletrônicos ou quaisquer outros tipos de documentos digitais) a fim de evitar a redundância no armazenamento de novas versões, buscando, entretanto, soluções que não comprometam o desempenho de acesso aos conteúdos disponibilizados. Na perspectiva do versionamento, foram analisados o SVN e os dois SGBDs livres mais utilizados em aplicações Web.

O SVN oferece dois subsistemas *backend* para armazenamento de dados versionados. O primeiro utiliza um ambiente de banco de dados chamado Berkeley DB (BDB). Repositórios baseados nesse ambiente também são conhecidos como BDB-*backed*. O segundo usa o próprio sistema de arquivos do Sistema Operacional, em um mecanismo conhecido como FSFS. O *backend* de armazenamento utilizado pelo repositório SVN influencia diretamente em seu desempenho (**Erro! Fonte de referência não encontrada.**) e, devido a isso, foram realizados testes com as duas configurações.

No contexto de SGBDs, poucos trabalhos acadêmicos foram publicados sobre a criação e o aperfeiçoamento de *benchmarks*. É possível que isso se deva à restrição imposta pela cláusula *DeWitt* (**Erro! Fonte de referência não encontrada.**), que proíbe a realização de testes em bancos de dados comerciais sem a prévia autorização do fabricante. Em razão disso, alguns *benchmarks* foram construídos para comparar o desempenho de SGBD de código aberto: TPCC-UVA (**Erro! Fonte de referência não encontrada.**) e OSDB (**Erro! Fonte de referência não encontrada.**). Entretanto, nenhum dos *benchmarks* apresentados acima é aplicado à análise de desempenho do SVN.

A maioria dos resultados comparando o desempenho de SGBDs de código aberto são muitas vezes contraditórios e tendenciosos. Os estudos apresentados em (**Erro! Fonte de referência não encontrada.**) e (**Erro! Fonte de referência não encontrada.**) comparam o desempenho dos SGBD PostgreSQL e MySQL, exibindo resultados contrários. Em (**Erro! Fonte de referência não encontrada.**), o PostgreSQL mostra-se superior, porém por estar ligado a uma fornecedora de soluções para o PostgreSQL o resultado é contestado (**Erro! Fonte de**

referência não encontrada.). O comportamento repete-se em favor do MySQL no trabalho (Erro! Fonte de referência não encontrada.).

O trabalho de Mabanza (Erro! Fonte de referência não encontrada.) comparou o desempenho de quatro bases de dados de código fonte aberto: Berkeley BD XML, Xindice, eXist e dbXML. Estes foram submetidos às operações de inserção, exclusão, busca e alteração de conteúdos. O fator de desempenho analisado foi o tempo de execução de cada uma destas tarefas.

Wang (Erro! Fonte de referência não encontrada.) descreve e faz uso de uma ferramenta para avaliar o desempenho e a escalabilidade de um sistema de armazenamento de conteúdos baseados em LDAP. As métricas assumidas como medidas de qualidade foram: tempos de realização das tarefas de adição, exclusão, busca e alteração de conteúdos presentes na base, taxa de utilização de CPU, taxa de utilização de memória, dentre outros.

As publicações (Erro! Fonte de referência não encontrada.) e (Erro! Fonte de referência não encontrada.) são bases para a construção dos *benchmarks* utilizados neste trabalho. A metodologia de experimentos será apresentada na próxima seção.

4. Metodologia de Experimentos

A fim de avaliar o desempenho do SVN, considerando suas duas opções de armazenamento (BDB e FSFS), e dos SGBDs MySQL e PostgreSQL, com vistas ao versionamento de conteúdos, foram propostas duas metodologias de análise: A primeira explora o comportamento dos repositórios submetidos à tarefa de versionamento de CDDs de um usuário único. A segunda analisa o comportamento dos mesmos submetidos ao acesso concorrente de um número variável de usuários.

Em ambos os casos, cada um dos repositórios foi submetido às operações de inserção (*upload*) e resgate (*download*) de CDDs. As métricas tempo de inserção e de resgate de conteúdos, consumo de memória, tempo de utilização de CPU e total de espaço em disco utilizado foram coletadas e assumidas como parâmetros de desempenho.

O tempo de inserção de um CDD é período entre a submissão do pedido de inserção (no cliente) até o tempo em que este foi completamente armazenado no repositório (no servidor). O tempo de resgate de conteúdo é o intervalo entre a solicitação de resgate (realizada no cliente) e o completo armazenamento deste na máquina que originou o pedido.

As taxas de consumo de memória, tempo de utilização de CPU e espaço em disco utilizados para armazenamento de conteúdos foram medidas nos hospedeiros servidores e coletadas através comando *time*, nativo do kernel Linux.

O ambiente de realização dos experimentos é definido na seção Erro! Fonte de referência não encontrada. e os cenários avaliados estão detalhados na seção Erro! Fonte de referência não encontrada..

4.1. O ambiente de realização dos Experimentos

O ambiente montado para a realização dos experimentos é composto por dois computadores, um cliente (C) e um servidor (S), de configurações semelhantes interligadas através de uma rede Gibabit (Gb) Ethernet. Na máquina C é executado um navegador (Mozilla Firefox), sendo este responsável por enviar requisições (de inserção e resgate de



conteúdos) para a máquina **S**. Esta, por sua vez, é responsável por receber as requisições do cliente e processá-las.

A máquina **S** hospeda os sistemas de armazenamento definido para cada cenário e uma aplicação Web, desenvolvida em PHP, que é responsável pela interface entre o cliente e os repositórios de conteúdos. O servidor Web utilizado nesta experimentação foi o Apache versão 2.2. A conexão entre os hospedeiros **C** e **S** ocorre a uma taxa de um gigabit por segundo (Gbps) e o protocolo para transporte de conteúdo utilizado é o HTTP.

As máquinas **C** e **S** apresentam a seguinte configuração: Processador Dual Core 2.2 GHz, Memória RAM de três GB, disco rígido de 360 GB padrão SATA 7200 RPM e placa de rede Ethernet 10/100/1000. O sistema operacional instalado em ambas é o Linux, distribuição Ubuntu 10.04 com kernel versão 2.6.32-25-generic.

4.2. Descrição dos Cenários

Os cenários A, B, C e D, descritos em detalhes através da Tabela **Erro! Fonte de referência não encontrada.**, foram estabelecidos com o objetivo de analisar o desempenho dos repositórios PostgreSQL, MySQL, SVN *FSFS-backed* e SVN *BDB-backed* em ambientes Web e submetidos às operações, de um único usuário, de inserção e resgate de CDDs.

O CDDs utilizados em ambas as fase de experimentações foram previamente elaborados com o uso da ferramenta *split*. Uma imagem ISO (.iso) de quatro gigabytes foi segmentada dando origem aos CDDs de prova.

Em um primeiro momento, foram realizadas experimentações em cenários que não incluem concorrência. Em cada desses cenários, foram realizadas vinte operações de inserção e vinte operações de resgate de conteúdos. Para cada operação, considera-se um CDD de tamanho inicial de 10 megabytes (MB) e final de 200 MB, acrescidos de 10 MB a cada nova operação.

Cenário	Sistema de Armazenamento	Operação
A	PostgreSQL	Inserção de Conteúdo
		Resgate de Conteúdo
B	MySQL	Inserção de Conteúdo
		Resgate de Conteúdo
C	SVN (<i>FSFS-backed</i>)	Inserção de Conteúdo
		Resgate de Conteúdo
D	SVN (<i>BDB-backed</i>)	Inserção de Conteúdo
		Resgate de Conteúdo

Tabela 1. Cenários de teste

Para analisar o comportamento dos sistemas avaliados em situação de concorrência, novos cenários foram propostos para os repositórios SVN *FSFS-backed*, SVN *BDB-backed*, PostgreSQL, MySQL também para inserção e resgate de CDDs. Nestes cenários, cada *thread* gerada no computador **C** é responsável por realizar uma operação de inserção ou resgate de conteúdos para cada um dos sistemas de armazenamento, uma de cada vez, a cada nova operação. O número inicial e final de *threads* é, respectivamente, 15 e 300, acrescidos de 15 a cada nova operação.

Os CDDs de prova têm tamanhos variáveis. A cada operação, 50% das *threads* manipulam CDDs de 1MB, 30% operam CDDs de 1,5 MB e 20% delas inserem ou resgatam CDDs de 5MB. Visando a utilização de um cenário de testes baseado em parâmetros reais, o percentual e o tamanho dos CDDs utilizados neste cenário refletem a situação atual da última versão estável do repositório InterRed (**Erro! Fonte de referência não encontrada.**).

Na metodologia adotada, cada operação é executada trinta e três vezes para cada cenário, calculado-se, em seguida, a média, o desvio padrão e o intervalos de confiança (IC). Os resultados são apresentados na seção **Erro! Fonte de referência não encontrada.**

5. Resultados Obtidos

A figura 1 apresenta os tempos de inserção e resgate de conteúdos, obtidos para os repositórios SVN *BDB-backed*, SVN *FSFS-backed*, MySQL e PostgreSQL, em função do tamanho do CDD.

O SVN *BDB-backed* descreve qualquer atualização, inicialmente, em arquivos de *log* específicos para, em seguida, realizar a operação. Isto garante a recuperação da base de dados, em casos de falha. Esse comportamento juntamente com o processo de identificação e armazenamento de deltas são responsáveis pelo maior consumo de tempo observado para a realização de operações neste repositório.

A inserção de conteúdos em campos BLOB³ no PostgreSQL, segundo o cenário A, é dividida em duas etapas: primeiramente, o CDD é transferido de **C** para **S**; em seguida, o caminho do objeto em **S** é passado como parâmetro para a função *lo_import()*. Esta operação armazena o conteúdo binário na tabela de sistema *pg_large_object* e devolve como resultado um OID (Identificador de Objeto) que deverá ser armazenado e utilizado como referência ao conteúdo, na tabela criada pelo usuário.

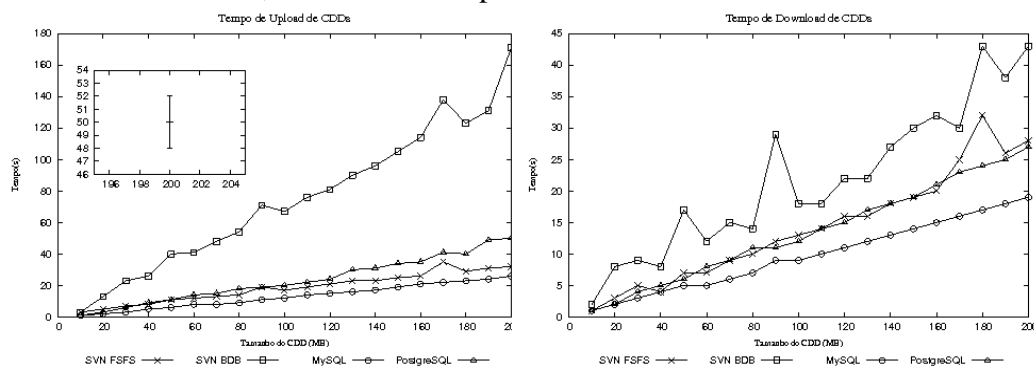


Figura 1: Tempo de (a) Inserção de Conteúdos e (b) Resgate de Conteúdos.

A inserção em campos BLOB no MySQL não precisa, como no PostgreSQL, passar pela etapa intermediária de transferência do conteúdo do cliente para o servidor. A maneira como os diferentes SGBDs realizam o armazenamento e resgate de conteúdos binários justifica o menor tempo de inserção e resgate de conteúdos apresentado pelo MySQL quando comparado ao PostgreSQL.

³Binary Large Object

O Subversion emprega uma técnica de *skip-deltas* que constroi os deltas das revisões utilizando informações das revisões passadas. A quantidade de revisões usadas para construção do delta depende do número da revisão atual. Em especial, as revisões de número (2^n+1) utilizam todas as revisões anteriores no processo de deltificação, por isso tendem a ocupar mais espaço de armazenamento (**Erro! Fonte de referência não encontrada.**), o que justifica os picos observados nos tempos de inserção e resgate das versões três, cinco, nove e dezessete dos CDDs de prova.

A figura 2 apresenta os resultados para o (a) tempo de utilização de CPU em operações de inserção e (b) consumo de memória em operações de resgate de conteúdos em função do tamanho do CDD. Podemos observar uma taxa de utilização excessiva de CPU e Memória pelos repositórios SVN *BDB-backed* e *FSFS-backed*. Isso é devido ao processamento inerente à realização da deltificação. Os menores consumos de memória no resgate das versões três, cinco, nove e dezessete dos CDDs de prova, quando comparados a revisões imediatamente anteriores se deve a não necessidade de processamento de deltas para obtenção dessas revisões (**Erro! Fonte de referência não encontrada.**).

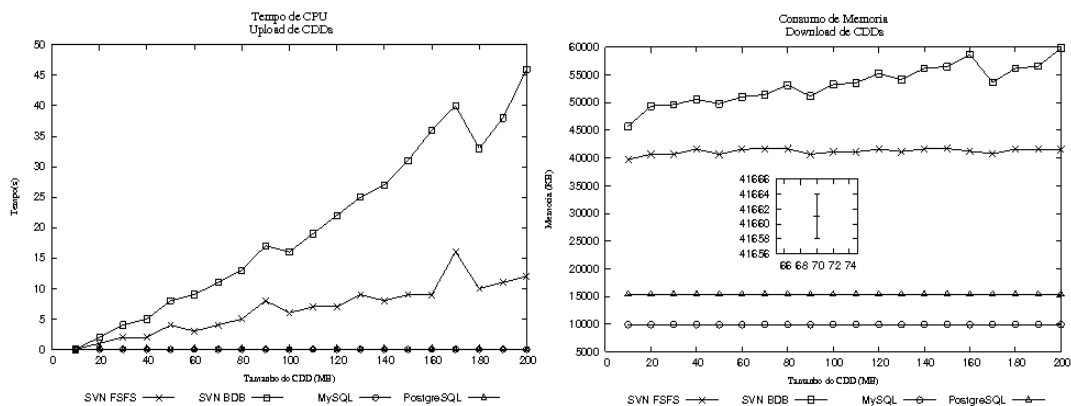


Figura 2: (a) Tempo de utilização de CPU em operações de inserção de conteúdos e (b) consumo de memória em operações de resgate de CDDs.

A figura 3 apresenta o (a) total de espaço em disco utilizado para armazenar as revisões de um mesmo conteúdo e o (b) tempo de resgate concorrente de conteúdos para os repositórios PostgreSQL e MySQL. O SVN não apresentou resultados suficientes para serem apresentados através de gráficos.

A menor demanda por espaço em disco pelos repositórios SVN está ligado ao processo de deltificação, enquanto que o maior consumo de espaço em discos pelo SVN *BDB-backed* quando comparado ao SVN *FSFS-backed* está ligado ao modelo de armazenamento de *skip-deltas* utilizados por cada *backend* de armazenamento (**Erro! Fonte de referência não encontrada.**).

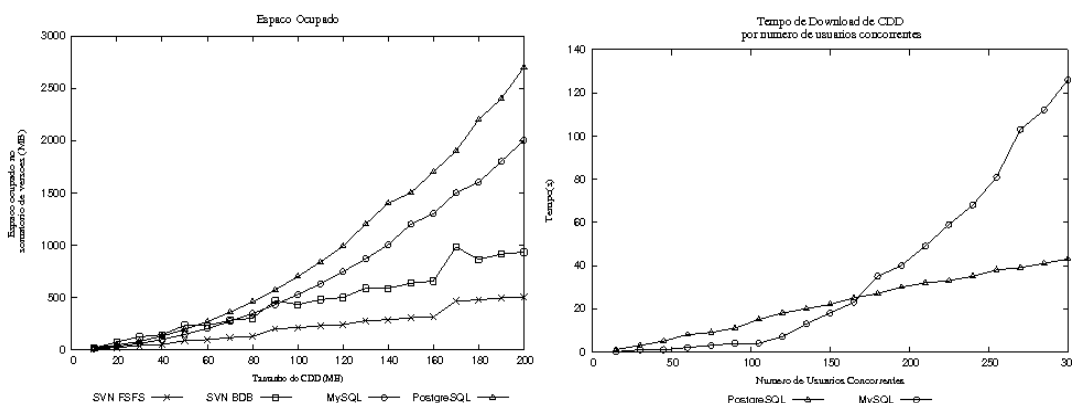


Figura 3: (a) Espaço ocupado em disco e (b) Tempo de resgate em ambientes concorrentes.

O MySQL apresenta melhor desempenho em operações concorrentes de inserção de conteúdos quando comparados ao PostgreSQL; Em operações de resgate de conteúdos, o repositório MySQL apresenta, novamente, melhor desempenho até o processamento de cento e sessenta e cinco requisições simultâneas, onde ocorre o saturamento do repositório, e a partir daí o PostgreSQL apresenta melhores resultados.

Para o acesso concorrente, os piores resultados ficaram com repositórios SVN *BDB-backed* e SVN *FSFS-backed* devido a dificuldade em fundir as mudanças conflitantes em arquivos binários. Nessas situações, é realmente necessário que o arquivo seja alterado por um usuário de cada vez. (**Erro! Fonte de referência não encontrada.**)

Os consumos de memória e CPU em ambientes concorrentes acompanham os resultados alcançados em ambientes com um único usuário. As considerações finais serão apresentadas na seção **Erro! Fonte de referência não encontrada.**

6. Conclusões

Neste trabalho foi apresentado um estudo sobre o desempenho de repositórios com vistas ao versionamento de conteúdos de arquivos binários variando entre 10 e 200 MB. Foram analisados os sistemas SVN *BDB-backed*, SVN *FSFS-backed*, MySQL e PostgreSQL com o uso de ferramentas de testes desenvolvidas especificamente para reproduzir as funções de inserção e resgate de conteúdos binários via HTTP. A análise visa a especificação de um repositório de conteúdos didáticos digitais (CDDs) cujos requisitos são: o acesso eficiente para consulta via Web; o reuso dos arquivos disponibilizados e, conseqüentemente, a geração de novas versões e; por último, a disponibilidade permanente de acesso às versões anteriores.

Após as experimentações e testes realizados, foi constatado que os melhores tempos para inserção e resgate de conteúdos, bem como o menor consumo de memória e de CPU foram obtidos com o uso do MySQL, seguido pelos sistemas PostgreSQL, SVN *FSFS-backed* e SVN *BDB-backed*, nesta ordem. Observando-se a métrica ocupação de espaço em disco, devido ao modelo de armazenamento que considera apenas as diferenças entre os conteúdos em diferentes versões (deltas), os repositórios SVN *FSFS-backed* e SVN *BDB-*



backed apresentaram os melhores resultados. Nota-se, entretanto, que o SVN *BDB-backed* não apresenta desempenho que justifique sua substituição pelo SVN *FSFS-backed*.

Tendo em vista a construção de um repositório Web que ofereça um acesso eficiente aos conteúdos nele armazenados, os resultados apresentados, por um lado, remetem a uma solução cujos objetos mais frequentemente utilizados sejam mantidos em SGBDs tradicionais, notando-se vantagem em tempo de resposta no uso do MySQL. Todavia, considera-se que o PostgreSQL, embora tenha apresentado desvantagem relativamente aos tempos de resposta do MySQL, pode ser utilizado em aplicações com fortes requisitos de segurança e robustez, visto que ainda mantém melhores índices do que o SVN.

Supondo, por outro lado, que versões mais antigas dos CDDs tendam a ser requeridas com menor frequência, ou apenas em situações particulares, a análise realizada sugere ser mais eficiente, do ponto de vista do espaço físico para armazenamento, o uso de ferramentas especificamente concebidas para realizar a deltificação.

Os resultados apontam para a concepção de um ambiente de armazenamento híbrido, cujas últimas versões sejam mantidas em SGBD e as mais antigas no sistema SVN. Este ambiente se justifica devido ao fato de os SGBDs não terem sido concebidos com funcionalidades de deltificação, o que exigiria um investimento em desenvolvimento que pode ser evitado com o uso de ferramentas especializadas neste tipo de operação.

Referências

APACHE (2010). Skip-deltas in subversion. Disponível em <http://svn.apache.org/repos/asf/subversion/trunk/notes/skip-deltas>. Acesso em Janeiro de 2010.

BOHL, O., SCHEUHASE, J., SENGLER, R., and WINAND, U. (2002). The sharable content object reference model (scorm) - a critical review. *Computers in Education: Proceedings. International Conference on*, pages 950–951. Dec. 2002.

COLLINS-SUSSMAN, B., FITZPATRICK, B. W., and PILATO, C. M. (2007). *Controle de Versão com Subversion*. O'Reilly.

EPCT Virtual (2010). Online: disponível em <http://www.redenet.edu.br/geral/siep1.php>. Acesso em Novembro de 2010.

FURTADO, C., BONETTI, A., HONÓRIO Filho, P., CUNHA, R., SIQUEIRA, R., SOARES, J., BEZERRA, F., and Bezerra, E. (2009). Integração entre ferramentas e instituições: um modelo de colaboração para o desenvolvimento da educação profissional e tecnológica. *XX Simpósio Brasileiro de Informática na Educação*. Florianópolis.

GIRARDI, R. A. D. (2004). Framework para coordenação e mediação de web services modelados como learning objects para ambientes de aprendizado na web. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio. páginas 14-26.

GreatBridge (2000). Greatbridge performance comparison. Online: disponível em <http://www.angelfire.com/country/aldev0/pgsqli/GreatBridge.html>. Acesso em Janeiro de 2009.



HERNÁNDEZ, P. and GONZALO, J. (2002). Implementación en c del benchmark de transacciones distribuidas tpc-c. *Thesis. Escuela Universitaria Politécnica de Valladolid.*

Mabanza, N., Chadwick, J., and Krishna Rao, G. (2006). Performance evaluation of open source native xml databases - a case study. In *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, volume 3, pages 1861–1865.

MORAN, B. (2003). The Devils in the DeWitt Clause. Online: disponível em <http://www.sqlmag.com/article/sql-server/the-devil-s-in-the-dewitt-clause>. Acesso em Janeiro de 2011.

Muzio, J. A., Heins, T., and Mundell, R. (2002). Experiences with reusable e-learning objects: From theory to practice. *The Internet e Higher Education*, Vol. 5:21–34.

MySQL (2005). Performance comparison by mysql group. Disponível em <http://www.wikivs.com/wiki/MySQLvsPostgreSQL>. Acesso em Janeiro de 2010.

OSDB (2000). The open source database benchmark. Online: disponível em <http://osdb.sourceforge.net>. Acesso em Janeiro de 2011.

RIVED (2010). Rede internacional virtual de educação. Online: disponível em <http://www.rived.mec.gov.br>. Acesso em Janeiro de 2010.

SCHEER, S. and GAMA, C. L. G. (2004). Construção de um repositório para objetos educacionais hiperídia. *Congresso Nacional de Ambientes Hiperídia para Aprendizagem – CONAHPA*, page 8 p. Florianópolis - SC.

SEED-MEC (2010). Secretaria de educação a distância - ministério da educação (mec). Online: disponível em <http://portal.mec.gov.br>. Acesso em Dezembro de 2010.

SUBVERSION (2011). Online: disponível em <http://subversion.tigris.org>. Acesso em Janeiro de 2011.

Wang, X., Schulzrinne, H., Kelur, D., and Verma, D. (2008). Measurement e analysis of ldap performance. *Networking, IEEE/ACM Transactions on*, 16(1):232–243.

Widenius, M. (2000). Mysql developer contests postgresql benchmarks. Disponível em <http://www.devshed.com/c/a/BrainDump/MySQL-Developer-Contests-PostgreSQL-Benchmarks>. Acesso em Janeiro de 2011.