

# Uma Experiência de Sincronização de Bases de Dados Relacionais Utilizando SyncML

Alison L. B. Alonso<sup>1</sup>, Cristiano Oliveira<sup>1</sup>, Leonardo Fedalto<sup>1</sup>,  
Fulvio Vilas Boas<sup>1</sup>, Tiago L. G. de Assis<sup>1</sup>, Carmem Satie Hara<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Federal do Paraná (UFPR)  
Caixa Postal 19.081 – 81.531-980 – Curitiba – PR – Brazil

alison@youse.com.br, {cof06, lspf06, fvb08, tlga06, carmem}@inf.ufpr.br

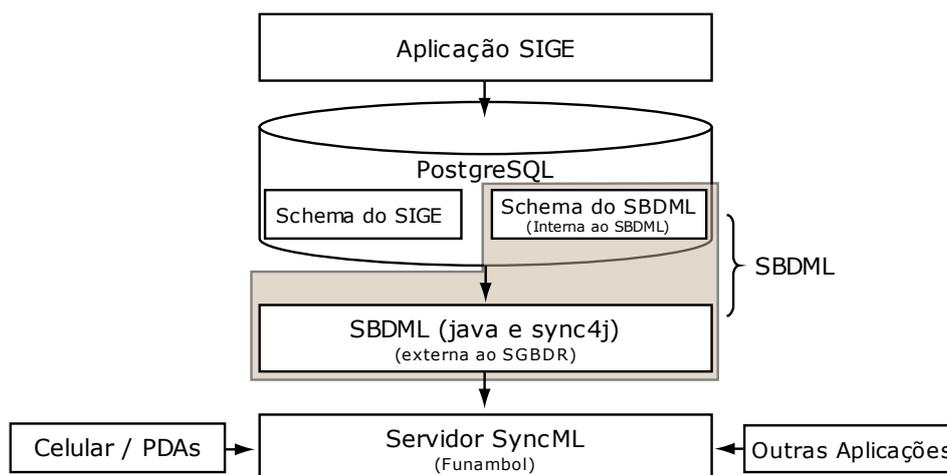
**Abstract.** *This paper describes SBDML, a SyncML-based middleware to replicate and synchronize data between a relational database and either another relational database or mobile devices. We also present some experimental results that determine the overhead of adopting SyncML as a synchronization protocol in this setting.*

**Resumo.** *Este artigo descreve um arcabouço chamado SBDML, baseado em SyncML, desenvolvido para sincronizar dados de um banco de dados relacional com outro banco de dados relacional ou com dispositivos móveis. Resultados de alguns experimentos executados para determinar o desempenho e overhead pela adoção do protocolo SyncML neste contexto são também apresentados.*

## 1. Introdução

O aumento do número de dispositivos móveis, como PDAs, smartphones e celulares, fez com que todos pudessem ter acesso a dados pessoais e de trabalhos em qualquer lugar e a qualquer tempo. Com isso, surge a necessidade de ter os mesmos tipos de dados em diferentes dispositivos, já que por questões de conectividade, disponibilidade e viabilidade, estes dados nem sempre estão disponíveis *online*. A sincronização de dados é a tecnologia usada para atenuar estas questões, mantendo a consistência das cópias distribuídas entre estes dispositivos [Hansmann et al. 2002]. Os serviços de sincronização mais comuns são para informações pessoais, tais como agenda de contato, calendários, tarefas e memorandos. No entanto, a sincronização de dados é importante não apenas para replicar dados entre dispositivos, mas também em aplicações diferentes. Diferentes tipos de aplicações muitas vezes trabalham com o mesmo tipo de informação, e a sincronização entre estes aplicativos seria de grande utilidade. Pode-se imaginar a sincronização de dados entre uma ferramenta de gerenciamento de projetos (GP) com uma ferramenta de agenda pessoal, esta podendo estar tanto em um notebook como em um smartphone. Todas as tarefas delegadas a uma pessoa, dentro da ferramenta de GP, constariam também na agenda pessoal dela, facilitando assim, a visualização dos compromissos a serem cumpridos.

Praticamente todos os Sistemas Integrados de Gestão Empresarial (SIGEs) utilizam um banco de dados relacionais para a persistência de dados. Distribuir os dados dos SIGEs é um desafio a parte, já que estes não possuem suporte para a sincronização com outras ferramentas. Seria necessário então pensar na sincronização em uma camada mais baixa, mais precisamente no banco de dados do SIGE.



**Figura 1. Arquitetura do SBDML**

O artigo *A SyncML Middleware-Based Solutions for Pervasive Relational Data Synchronization* [Yang et al. 2008] apresenta um arcabouço, chamado GSMS, independente de aplicação, para a sincronização de dados entre diferentes sistemas gerenciadores de banco de dados relacionais (SGBDR) proprietários. O GSMS utiliza como protocolo de sincronização o SyncML [OMA] (*Synchronization Markup Language*). O SyncML é um protocolo padrão de sincronização de dados, independente de plataforma, dispositivo e rede. Ele é baseado em tecnologia XML (*Extended Markup Language*) e mantido pela OMA (*Open Mobile Alliance*), uma aliança entre várias empresas para a criação de um protocolo comum para a sincronização de dados.

O objetivo deste trabalho é desenvolver um arcabouço, semelhante ao GSMS, para o SGBDR de código livre Postgres [PostgreSQL], que seja transparente ao aplicativo que utiliza o SGBDR. O arcabouço, chamado do SBDML, também utiliza o protocolo de sincronização SyncML, mas permite sincronização não apenas entre SGBDRs, mas também entre um SGBDR e uma outra aplicação ou dispositivo que utilize o SyncML.

## 2. O Arcabouço SBDML

O SBDML é um arcabouço construído para sincronizar dados armazenados no SGBDR Postgres com dados armazenados em outros dispositivos ou aplicações. Ele foi projetado de forma que o processo de sincronização, baseado em SyncML, seja transparente às aplicações desenvolvidas que utilizem o SGBDR. Em outras palavras, o funcionamento do arcabouço não afeta o funcionamento e não altera a estrutura das tabelas da aplicação.

A Figura 1 apresenta a arquitetura do sistema. A aplicação SIGE altera os dados no SGBDR. O SBDML detecta e propaga as alterações para o servidor SyncML. De forma similar ao GSMS [Yang et al. 2008], o SBDML possui duas partes:

1. uma parte interna ao SGBDR, implementada com gatilhos e procedimentos armazenados, responsável por capturar as alterações;
2. uma parte externa ao SGBDR, responsável por propagar as alterações dos dados e implementada em Java, utilizando a biblioteca Sync4j [FUNAMBOL].

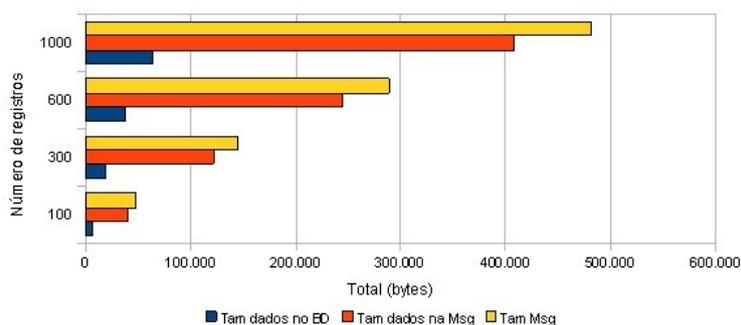
Na parte interna ao SGBDR, construiu-se uma tabela, chamada *change\_log*, para armazenar as mudanças ocorridas nos dados monitorados. Os gatilhos são utilizados para capturar as alterações ocorridas e inseri-las na tabela *change\_log*. A propagação das alterações é realizada pela parte externa do SBDML através de uma consulta à tabela *change\_log*. A detecção de conflitos é responsabilidade do protocolo SyncML. Tenta-se diminuir a ocorrência de conflitos através do aumento da frequência de sincronização. Em geral, quanto mais rápida for a propagação, menor a taxa de conflitos, mas maior será a carga de trabalho, especialmente se a aplicação tiver muitas escritas no banco de dados.

O servidor SyncML utilizado para receber os dados do SBDML e sincronizá-lo com outros dispositivos ou aplicativos foi o Funambol Data Sync Server[FUNAMBOL].

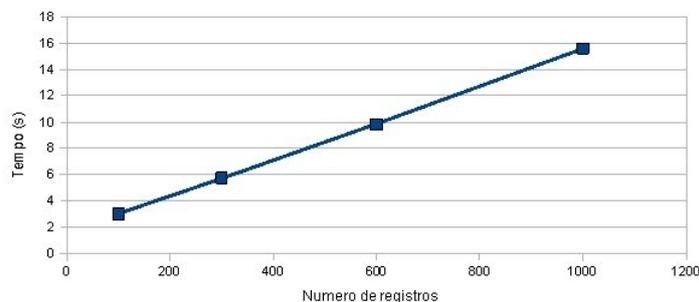
Uma característica positiva desta arquitetura é que os gatilhos e a tabela criada para a captura dos dados alterados não interferem na lógica das aplicações desenvolvidas sobre o SGBDR. Um ponto a ser analisado é o *overhead* que esta abordagem tem sobre o sistema. A próxima seção apresenta resultados de dois experimentos que avaliam esta questão.

### 3. Experimentos

Como cenário dos experimentos, considera-se um Sistema Integrado de Gestão Empresarial (SIGE), no qual deseja-se sincronizar os dados de contatos dos clientes de uma empresa com os dispositivos móveis dos vendedores.



(a) Tamanho dos dados



(b) Tempo de sincronização

**Figura 2. Resultados dos experimentos**

Foram realizados dois experimentos para avaliar o desempenho da sincronização e o *overhead* da utilização do protocolo SyncML. O primeiro experimento tem como objetivo determinar o custo de comunicação imposto pelo protocolo SyncML. Para isso,

foram comparados os tamanhos dos dados sincronizados, representados em três formas: somente o dado no formato armazenado no SGBDR, somente o dado na sua representação na mensagem XML e o tamanho total da mensagem transmitida. A Figura 2(a) exibe os resultados obtidos, no qual o eixo vertical representa a quantidade de registros e o eixo horizontal o tamanho em bytes nas três representações. Pode ser observado o tamanho da mensagem transmitida é aproximadamente 6,38 vezes o tamanho do dado na forma armazenada, sendo que grande parte deste aumento deve-se à transformação da representação do dado no formato XML, codificado em base 64.

O segundo experimento foi realizado para determinar o tempo necessário para executar uma sincronização do SGBDR com o servidor SyncML. Dentro do cenário dos experimentos, o SGBDR da aplicação SIGE, foi o PostgreSQL 8.4 executado em uma máquina Core 2 Duo com 4 GB de RAM. O servidor SyncML foi o Funambol Server 8.0.2 executado em uma máquina Pentium 4 com 1 GB de RAM. A figura 2(b) mostra os resultados obtidos. O tamanho dos dados nos registros foi fixado em 64 bytes. O experimento mostra que são necessários aproximadamente 3 segundos para sincronizar 100 registros, 8 segundos para 500 registros, e 14 segundos para 1000 registros. O resultado destes experimentos mostram que, 1) o tempo de sincronização é linear em relação a quantidade de dados a serem sincronizados; 2) o tamanho das mensagens em XML são significativamente maiores que a quantidade de dados que deseja-se sincronizar.

#### 4. Conclusão

O artigo apresenta o arcabouço SBDML, desenvolvido para estender o SGBDR Postgres com a funcionalidade de sincronização de dados com outras aplicações e dispositivos. Esta funcionalidade é especialmente interessante em sistemas SIGE, nos quais há necessidade de replicação de informações de clientes, que encontram-se no banco de dados, para outros dispositivos, como PDAs. O sistema foi desenvolvido utilizando o protocolo SyncML. Os experimentos realizados mostram que a utilização de XML como protocolo de comunicação é bastante custosa em termos de tamanho. Porém, a utilização de um protocolo padrão para sincronização de dados é muito importante. A conformidade com um padrão tende a ser cada vez mais importante, já que a cada dia surgem novos dispositivos móveis e aplicações. Isto, juntamente com a necessidade de replicação de dados e arquivos pessoais, faz com que o padrão SyncML seja cada vez mais adotado e utilizado. Como trabalho futuro, pretende-se avaliar a viabilidade de estender o sistema com uma ferramenta de compactação de dados.

#### Referências

- FUNAMBOL. Funambol mobile open source project (sync4j). [www.funambol.com/opensource](http://www.funambol.com/opensource).
- Hansmann, U., Mettälä, R., Purakayastha, A., Thompson, P., and Kahn, P. (2002). *SyncML: Synchronizing and Managing Your Mobile Data*. Prentice Hall.
- OMA. Syncml. [www.openmobilealliance.org/syncml](http://www.openmobilealliance.org/syncml).
- PostgreSQL. Postgresql. <http://www.postgresql.org/>.
- Yang, H., Yan, P., Lu, P., and Wang, Z. (2008). A syncml middleware-based solution for pervasive relational data synchronization. *Lecture Notes in Computer Science*, 5245.