MyGFT: um Módulo de Integração entre MySQL e Google Fusion Tables

Alexandre Savaris[1,2], Carmem Satie Hara[1], Aldo von Wangenheim[1,2]

[1]Universidade Federal do Paraná (UFPR) – Departamento de Informática
Caixa Postal 19.081 – 81.531-980 – Curitiba – PR – Brasil

[2]INCoD – Instituto Nacional para Convergência Digital
Universidade Federal de Santa Catarina (UFSC) – Departamento de Informática e Estatística – Sala 320 – 88.040-970 – Florianópolis – SC – Brasil

{asavaris,carmem}@inf.ufpr.br, awangenh@inf.ufsc.br

Abstract. *This work presents MyGFT, a storage engine for integrating MySQL DBMS and Google Fusion Tables, a cloud-based data management service. The module is described in terms of architecture, its integration to MySQL and how its use can be a viable alternative to data retrieval for controlled and structured vocabularies used in healthcare applications.*

Resumo. *Este trabalho apresenta o MyGFT[1], um storage engine para a integração do SGBD MySQL ao serviço de gerenciamento de dados em nuvem Google Fusion Tables. O módulo é descrito em termos de arquitetura, integração ao MySQL e de como sua utilização pode se tornar uma alternativa viável à recuperação de dados pertencentes a vocabulários controlados e estruturados utilizados em aplicações na área da saúde.*

1. Introdução

A extensão SQL/MED (*Management of External Data*), definida em meados de 2000, passou a integrar o padrão SQL com o objetivo de estabelecer uma metodologia de acesso a fontes de dados externas às instâncias relacionais. Pela sua utilização, é possível complementar os dados normalizados disponíveis em instâncias de bancos de dados relacionais com dados provenientes de origens diversas (como arquivos armazenados em sistemas de arquivos convencionais, instâncias de bancos de dados de rede ou hierárquicos, outras instâncias de bancos de dados relacionais, dentre outras). Esse acesso a fontes de dados heterogêneas visa simplificar atividades que envolvam diferentes conjuntos estruturalmente distintos, provendo uma interface unificada (baseada na linguagem SQL) que permita a seleção, a manipulação e o estabelecimento de relações diretas entre os conjuntos disponíveis [Melton 2001], [Melton 2002].

A gravação de dados externos em arquivos ou mesmo em bancos de dados com arquiteturas diversas é uma prática conhecida. Como estratégia alternativa, serviços orientados a dados em nuvem (DaaS – *Data as a Service*) têm se apresentado como uma opção viável ao armazenamento convencional [Zhou 2010], [Dikaiakos 2009]. Além de

[1] Este trabalho é parcialmente financiado pela Fundação Araucária projeto 22.741 e CNPq processo 484366/2011-4.

tornarem o acesso aos dados ubíquo, esses serviços podem auxiliar na redução da redundância de dados entre diferentes sistemas de informação. Neste cenário, consideram-se diferentes sistemas clientes acessando os mesmos serviços em nuvem via APIs oferecidas pelos próprios serviços.

Visando prover conjuntos de dados comuns a diferentes sistemas de informação de forma a centralizar o seu armazenamento, este trabalho apresenta o MyGFT – um *storage engine* que objetiva a integração entre instâncias do banco de dados MySQL com o serviço de gerenciamento de dados em nuvem Google Fusion Tables (GFT). O *storage engine* é desenvolvido como uma extensão modular (o que permite a sua instalação e utilização em diferentes cenários de uso do SGBD), de forma a facilitar o processo de recuperação de dados a partir do GFT e sua posterior integração a conjuntos de dados locais.

O trabalho é organizado como segue. Na seção dois são apresentados detalhes técnicos sobre o GFT, com foco em sua arquitetura, API e políticas de acesso; a seção três descreve o processo de desenvolvimento do MyGFT, partindo da sua integração com a arquitetura modular do MySQL até o processo de utilização durante a criação de tabelas; a seção quatro apresenta e discute um possível cenário de uso para o módulo desenvolvido; a seção cinco relaciona trabalhos correlatos, e a seção seis conclui o trabalho com as primeiras impressões sobre a utilização do módulo e com possíveis trabalhos futuros.

2. Google Fusion Tables

Disponibilizado em junho de 2009, o Google Fusion Tables (GFT) é um serviço de gerenciamento de dados em nuvem que objetiva facilitar o compartilhamento de conjuntos de dados e a execução de atividades colaborativas sobre esses conjuntos em uma arquitetura *Web* [Gonzalez 2010a]. O serviço possibilita a criação de tabelas de dados pela execução de comandos em uma interface *Web* ou pela importação de arquivos nos formatos CSV (*Comma Separated Values*), KML (*Keyhole Markup Language*) ou planilhas, limitados a um tamanho máximo de 100MB. Uma vez criadas, essas tabelas de dados podem ser definidas como públicas, privadas ou compartilhadas entre usuários chamados *colaboradores*; além disso, seu conteúdo pode ser relacionado ao conteúdo de outras tabelas via equijunções, sendo possível também a construção de visões para a integração de diferentes tabelas visando um acesso unificado.

Estruturalmente, o GFT é organizado sobre uma pilha de serviços de armazenamento, com destaque para a estrutura utilizada na persistência de pares chave/valor e para a biblioteca de primitivas utilizada na criação de índices secundários, gerenciamento de transações e replicação [Gonzalez 2010b]. O armazenamento efetivo dos dados do GFT é feito em estruturas do tipo *Bigtable*, implementadas como mapas multidimensionais ordenados altamente escaláveis. Essas estruturas são utilizadas também para o armazenamento dos esquemas das tabelas, índices, *logs* de transação e comentários feitos pelos usuários com acesso às tabelas. A biblioteca *Megastore*, por sua vez, é responsável pela indexação secundária dos atributos das tabelas, pela implementação e gerenciamento de transações ACID e pela replicação de dados, esquemas, índices, *logs* e comentários em diferentes servidores.

A API[2] do GFT é organizada de forma a permitir que diferentes aplicações clientes possam usufruir do serviço de armazenamento, utilizando-se de um conjunto bem definido de operações DDL (criação e exclusão de tabelas) e DML (seleção, projeção, agrupamento e agregação, inserção, atualização e exclusão de dados). As operações a serem executadas são enviadas pelas aplicações clientes via HTTP/RPC para o GFT, onde são interceptadas e avaliadas pelo módulo despachante. Esse módulo converte a requisição recebida para uma representação interna do serviço (consulta), e a encaminha para o módulo de otimização, responsável pela geração do respectivo plano de execução. Esse plano de execução é enviado ao processador de consulta, módulo responsável pela execução do plano, pelo recebimento dos resultados e pelo encaminhamento desses resultados aos módulos de nível superior.

3. O *storage engine* MyGFT

O SGBD MySQL é caracterizado por uma arquitetura modular, extensível pelo desenvolvimento de *plugins* responsáveis por encapsular funcionalidades como busca textual e autenticação [Golubchik e Hutchings 2010]. Especificamente com relação ao armazenamento de dados, é possível definir e implementar diferentes formatos e suas respectivas formas de acesso pelo atendimento às especificações da *Storage Engine API*[3]. O *storage engine* MyGFT é construído de acordo com os preceitos dessa API, permitindo sua integração à arquitetura do MySQL conforme exibido na figura 1.

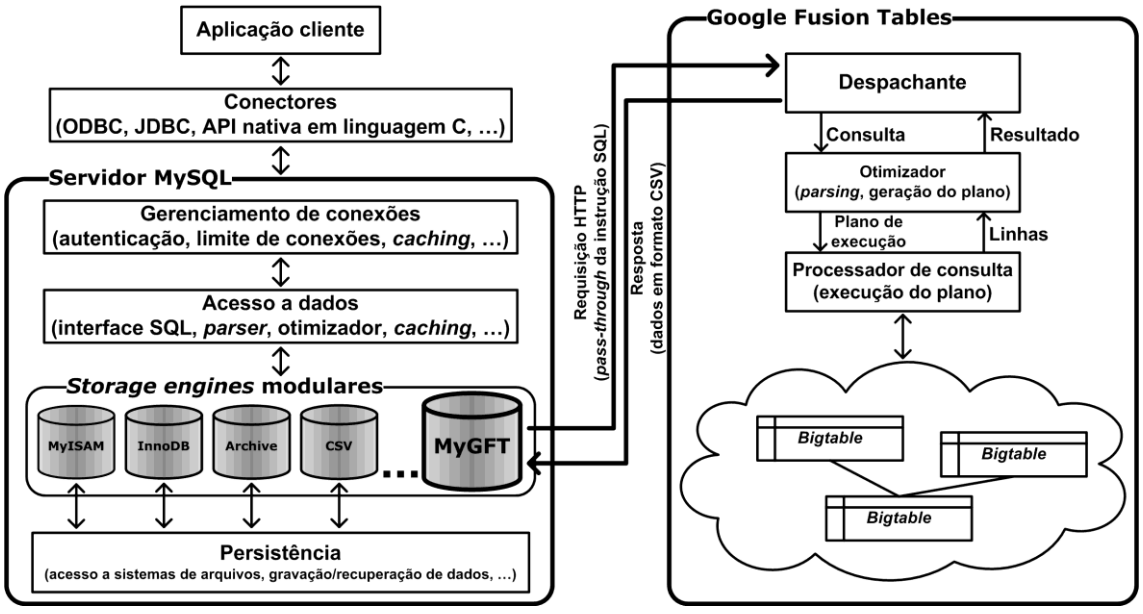

Figura 1. O *storage engine* MyGFT – integração à arquitetura MySQL e acesso ao GFT (adaptação).

Visando atender ao cenário de uso descrito na seção quatro, o MyGFT foi projetado como um *storage engine* somente de leitura; essa restrição de funcionalidade é possível graças à organização da *Storage Engine API*, que mapeia instruções DDL e DML para funções em linguagem C++ nas quais as ações são efetivamente implementadas.

[2] https://developers.google.com/fusiontables/
[3] http://dev.mysql.com/doc/refman/5.6/en/pluggable-storage-overview.html

Como exemplo, uma instrução SELECT é mapeada para um conjunto de funções (`open()`, `rnd_init()`, `rnd_next()`, `rnd_end()`, `close()`) invocadas pelo servidor na ordem definida pela API.

Uma vez compilado, o módulo MyGFT pode ser disponibilizado para utilização em um determinado servidor MySQL seguindo o método padrão de instalação de *plugins*, como segue:

```
mysql> install plugin MYGFT soname 'ha_mygft.so';
```

Com o módulo devidamente instalado, é possível criar tabelas locais e usá-las para recuperar dados a partir de tabelas armazenadas no GFT. A criação dessas tabelas requer a especificação de um padrão de codificação de caracteres para os dados recuperados, bem como a especificação do identificador único da tabela no GFT:

```
mysql> CREATE TABLE cid10(codigo_subcategoria text,
         nome_subcategoria text) DEFAULT CHARSET=utf8
         CONNECTION='4371448' engine=mygft;
```

No exemplo, o *default charset* para os dados recuperados é definido como UTF-8 (padrão do GFT), e o identificador único da tabela (4371448) corresponde ao identificador atribuído pelo GFT no momento da criação da tabela original. Além disso, a relação de campos deve ser equivalente à relação de colunas da tabela original.

Instruções de seleção (SELECTs) repassadas pelo servidor ao módulo MyGFT não são interpretadas pelo módulo. Graças à API SQL do GFT, é possível encaminhar essas instruções na forma de requisições HTTP diretamente ao serviço, provendo assim uma implementação para a modalidade *pass-through* prevista no padrão SQL/MED. A construção das requisições HTTP baseada na URL padrão para execução de consultas no GFT, bem como no identificador único da tabela no serviço, é feita com o auxílio das funções disponibilizadas pela biblioteca libcurl[4], que assume o papel de um cliente *web* integrado ao módulo. Essa abordagem simplifica a implementação do MyGFT, que repassa toda a responsabilidade pela manutenção dos dados pesquisados – incluindo a indexação – ao GFT.

Os dados encontrados pelo GFT após a execução das consultas são encaminhados ao MyGFT no formato CSV. O módulo, então, executa um *parsing* sobre esses dados convertendo-os no formato interno utilizado pelo MySQL para a representação de campos e registros, repassando-os sem seguida à camada de acesso a dados do servidor. Uma vez nessa camada, os registros recuperados podem ser usados de forma integrada a registros armazenados em tabelas locais (via junções ou subconsultas), como no exemplo a seguir: uma tabela local (*laudo_exame*) é relacionada à tabela *cid10* armazenada no GFT, objetivando recuperar o identificador único do laudo e o termo relacionado ao vocabulário controlado/estruturado (vide seção 4).

```
mysql> SELECT le.id, c.nome_subcategoria FROM laudo_exame le, cid10 c
        WHERE le.codigo_subcategoria = c.codigo_subcategoria;
+----+-------------------------------------------------------+
| id | nome_subcategoria                                     |
+----+-------------------------------------------------------+
|  1 | Cólera devida a Vibrio cholerae 01, biótipo cholerae  |
+----+-------------------------------------------------------+
```

[4] http://curl.haxx.se/libcurl/

4. Cenário de uso – acesso a vocabulários controlados

Sistemas de informação desenvolvidos para a área da saúde costumam utilizar um ou mais vocabulários controlados/estruturados. Esses vocabulários são compostos por conjuntos de termos normalizados que objetivam padronizar a nomenclatura usada na área de forma a facilitar a indexação de conteúdo e diminuir a utilização de texto livre. Vocabulários como DeCS (Descritores em Ciências da Saúde)[5] e CID-10 (Classificação Estatística Internacional de Doenças e Problemas Relacionados com a Saúde)[6] são exemplos desses vocabulários; sua estruturação hierárquica permite representar não apenas doenças, mas também termos pertinentes a áreas como anatomia, compostos químicos, equipamentos, dentre outras.

Para que o conteúdo desses vocabulários seja disponibilizado às aplicações, é uma prática conhecida que suas estruturas hierárquicas sejam modeladas de maneira *ad hoc* em bancos de dados relacionais, resultando em diversos esquemas com particularidades específicas a cada aplicação. Essa multiplicidade de esquemas pode influenciar negativamente o processo de integração de dados entre sistemas (operação comum na área da saúde), bem como a busca de dados executada entre sistemas.

A utilização do *storage engine* MyGFT provê uma alternativa viável à criação de repositórios individuais (em cada sistema) para o armazenamento dos vocabulários controlados/estruturados. Sua característica somente de leitura permite que diferentes sistemas possam acessar o conteúdo de tabelas/visões criadas diretamente no GFT, centralizando o acesso a conjuntos de dados comuns e que, tais como esses vocabulários, não recebem atualizações constantes; de forma complementar, a restrição de acesso imposta ao módulo garante a integridade das informações disponibilizadas, evitando que diferentes aplicações atualizem de forma indiscriminada o seu conteúdo e comprometam a semântica inerente a dados utilizados previamente. Outro aspecto favorável à utilização do módulo é a garantia de que, em caso de atualização de conteúdo, o mesmo seja disponibilizado imediatamente a todas as instâncias locais em MySQL relacionadas às tabelas no GFT; com isso, garante-se que todas as aplicações que compartilham os dados dos vocabulários passem a usufruir de uma visão atualizada, provendo uma integração de dados consistente.

5. Trabalhos relacionados

Módulos customizados têm sido usados como forma de integração entre o modelo relacional e outros modelos de armazenamento, visando o aproveitamento das suas melhores características. O trabalho de [Ribas 2010] apresenta um módulo de armazenamento para a integração do MySQL com Tabelas de Espalhamento Distribuídas (DHT); diferentemente deste trabalho (que foca no acesso a dados centralizados utilizando clientes distribuídos), os autores visam o desenvolvimento de um sistema caracterizado pela escalabilidade, descentralização, tolerância a falhas e facilidade de uso, no qual os dados são distribuídos. O trabalho de [Atwood 2007], por sua vez, objetiva o armazenamento de grandes volumes de dados em nuvem, visando escalabilidade. No presente trabalho,

[5] http://decs.bvs.br/P/decsweb2012.htm
[6] http://www.datasus.gov.br/cid10/v2008/cid10.htm

busca-se a centralização e o compartilhamento de dados pouco mutáveis, com volumes bem definidos, cuja escalabilidade não é um fator determinante.

6. Conclusões e trabalhos futuros

Este trabalho apresentou o *storage engine* MyGFT, um módulo desenvolvido com o objetivo de integrar o SGBD MySQL ao serviço de gerenciamento de dados em nuvem Google Fusion Tables. Essa integração visa permitir a recuperação de conjuntos de dados armazenados em um repositório centralizado, de forma que os mesmos não precisem estar fisicamente replicados em diferentes sistemas de informação, estruturados em esquemas definidos de maneira *ad hoc*.

Os testes prévios executados com o módulo, em um cenário de uso envolvendo os vocabulários controlados/estruturados DeCS e CID-10, atestam a possibilidade de recuperação de termos desses vocabulários via MyGFT pela execução de consultas cujos predicados intersectem os conjuntos de predicados suportados pelo GFT e pelo MySQL. O desempenho de busca, conforme esperado, é inferior ao desempenho de acesso a dados locais; isso é justificável pelo fato do acesso ao GFT via MyGFT envolver, além da execução da consulta em si e do *parsing* sobre os dados encontrados, a construção de uma requisição HTTP, o envio dessa requisição ao GFT, a construção de uma resposta HTTP e o envio do conjunto de dados resultante ao MyGFT. Otimizações deste processo podem ser relacionadas como trabalhos futuros, englobando testes mais abrangentes envolvendo um maior número de predicados e a implementação de *caches* locais para a minimização do volume de dados trafegado.

Referências

Atwood, M. (2007) “A Storage Engine for Amazon S3”. Em: MySQL Conference & Expo 2007.

Dikaiakos, M. D., et al. (2009) “Cloud Computing: Distributed Internet Computing for IT and Scientific Research”, Em: IEEE Internet Computing, v. 13(5), p. 10-13.

Golubchik, S. e Hutchings, A. (2010), MySQL 5.1 Plugin Development, Packt Publishing Ltd.

Gonzalez, H., et al. (2010a) “Google Fusion Tables: Web-Centered Data Management and Collaboration”, Em: Proceedings of the 2010 International Conference on Management of Data, p. 1061-1066.

Gonzalez, H., et al. (2010b) “Google Fusion Tables: Data Management, Integration and Collaboration in the Cloud”, Em: Proceedings of the 1st ACM Symposium on Cloud Computing, p. 175-180.

Melton, J., et al. (2001) “SQL and Management of External Data”, Em: ACM SIGMOD Record, v. 30(1), p. 70-77.

Melton, J., et al. (2002) “SQL/MED – A Status Report”, Em: ACM SIGMOD Record, v. 31(3), p. 81-89.

Ribas, E. A., et al. (2010) “Um SGBD com Armazenamento Distribuído de Dados Baseado em DHT”, Em: Anais do XXV Simpósio Brasileiro de Banco de Dados.

Zhou, M., et al. (2010) “Services in the Cloud Computing Era: A Survey”, Em: Proceedings of the 4th International Universal Communication Symposium, p. 40-46.