

DsmDB - Um Sistema de Gerenciamento de Banco de Dados em Memória, Distribuído

Mitchel Felske

Introdução

Propõe-se neste trabalho um gerenciador para armazenamento de dados em memória que execute de forma distribuída em computadores interligados em rede - nodos. Sua finalidade é fornecer uma solução eficiente, na medida em que tenta diminuir as restrições de desempenho e disponibilidade que os Sistemas Gerenciadores de Bancos de Dados (SGBD's) tradicionais impõem às aplicações. Busca-se manter os dados armazenados nas memórias principais dos vários nodos. Com isso, aproveita-se a grande capacidade de memória presente nos ambientes computacionais atuais. Soluções em memória são mais apropriadas do que soluções em disco, as quais normalmente são utilizadas pelos SGBD's tradicionais. Elas possibilitam alto throughput e baixos tempos de resposta, diferentemente do que ocorre com os discos. Dessa forma, as aplicações passam a dispor de um espaço de armazenamento limitado pela soma das capacidades das memórias dos nodos interligados e não somente por um único nodo, conforme [1] e [5]. Para isso é utilizado o paradigma Distributed Shared Memory (DSM), que oferece a abstração de um único espaço de endereços implementado sobre várias memórias físicas, conforme proposto por [2] e [3].

Metodologia

O dsmDB é composto por dois módulos: o Gerenciador de Armazenamento e o Gerenciador de Consistência. O primeiro implementa uma abstração transacional a qual provê o acesso ao conjunto de todos os objetos do banco de dados. Para codificá-lo, ele foi subdividido em duas partes: a primeira consiste em implementar um programa para fazer a manipulação dos dados no BDB em um ambiente local, ou seja, em um servidor centralizado. Já a segunda parte visa expandi-lo, de forma que as aplicações possam acessar os dados remotamente. O segundo módulo consiste na criação de um sistema que irá gerenciar globalmente as informações, tratando os casos de concorrência e condições de disputa e, conseqüentemente, assegurando a consistência dos dados. Para isso, se faz uso de um mecanismo otimista baseado no conceito de transações.

Durante a fase de pesquisa bibliográfica, foi encontrado um sistema centralizado capaz de gerenciar bancos de dados em memória, o Berkeley Data Base (BDB). Conforme descrito em [4], trata-se de um banco de dados embarcado nas aplicações que promove acesso transacional, alta performance, escalabilidade e com alto grau de confiabilidade. Realizou-se uma avaliação das suas funcionalidades, e concluiu-se que ele seria adequado para servir como ponto de partida para a implementação das idéias propostas. Como o armazenamento em memória no BDB é realizado de forma centralizada, foi realizada uma modificação, remodelando-o para torna-se

distribuído entre vários nodos, com base no paradigma DSM, motivo que inspirou sua denominação: dsmDB.

Resultados

Foi desenvolvido o programa referente ao módulo do Gerenciador de Armazenamento. Conforme mencionado anteriormente, a implementação foi realizada em duas partes, com o objetivo de facilitar a codificação. A primeira permite a manipulação dos dados localmente, ou seja, as aplicações que realizam o acesso aos bancos de dados compartilham o mesmo espaço de endereços do programa construído. A segunda permite que as aplicações busquem informações em nodos remotos. Este módulo está em fase de testes, com o objetivo de determinar o seu desempenho e verificar se ele atende adequadamente as necessidades especificadas. O módulo Gerenciador de Consistência está em fase de codificação.

Conclusões

Foi descrito o andamento do trabalho que envolve a construção de um sistema de gerenciamento de bancos de dados distribuídos em memórias. Utilizou-se como ponto de partida um banco de dados centralizado. Resultados parciais mostram que a implementação das idéias propostas será uma boa alternativa para aplicações que demandam alta performance na recuperação e disponibilidade dos dados.

Referências

1. CAMARGOS, L.; PEDONE, F.; WIELOCH, M. **Sprint: A middleware for highperformance transaction processing**. In Eurosys, 2007.
2. Li, K. **Shared virtual memory on loosely coupled multiprocessors**, 1986.
3. Li, K.; HUDAK, P. **Memory coherence in shared virtual memory systems**. ACM Transactions on Computer Systems, 1989.
4. Oracle Berkeley db reference guide, 2006.
<http://www.oracle.com/technology/documentation/berkeley-db/db/ref/toc.html>
5. STONEBRAKER M. **The end of an architectural era (it's time for a complete rewrite)**. In VLDB, 2007.