

Hibernate em um ambiente de larga escala

Rogério Manente - rogerio@manente.com.br

Orientador: Prof. Dr. João Eduardo Ferreira - jef@ime.usp.br

1. Introdução e Objetivos

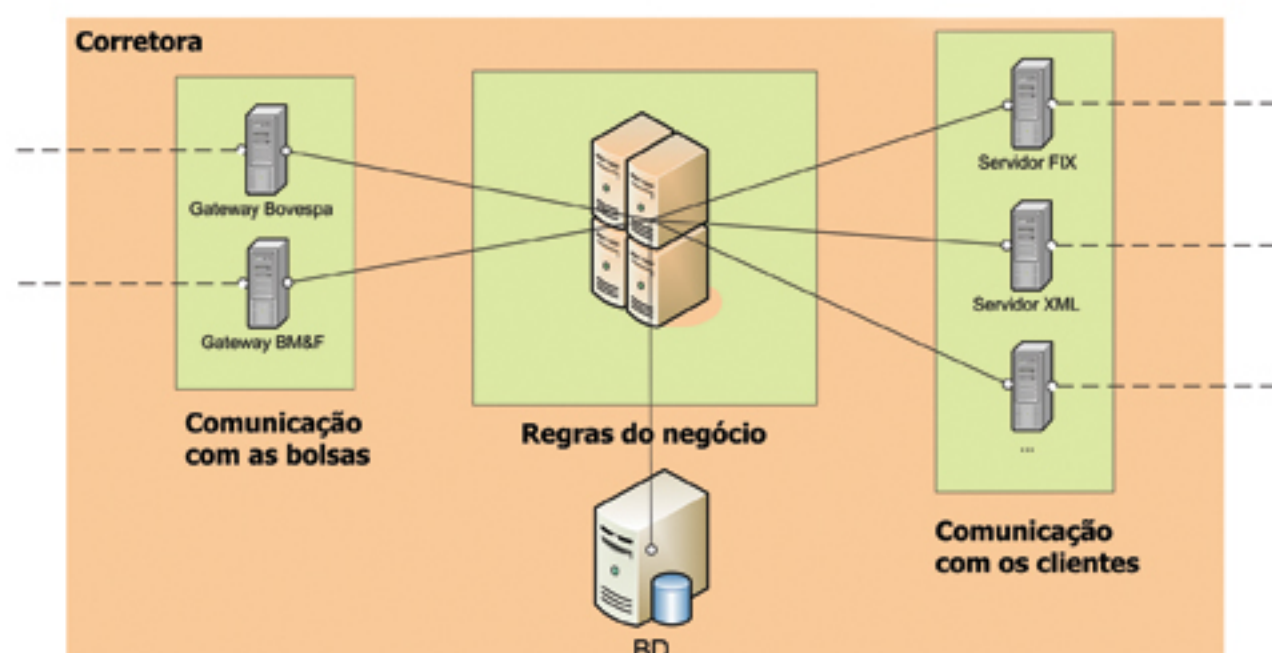
Uma das soluções mais utilizadas para o problema da persistência de dados é o **Hibernate**. O poder da ferramenta e sua agilidade no desenvolvimento de aplicações a tornam tão popular.

No entanto, é intuitivo que a adição de uma camada extra gera perda de performance. O objetivo deste trabalho é analisar os limites do **Hibernate** num ambiente de produção real de larga escala.

2. Ambiente

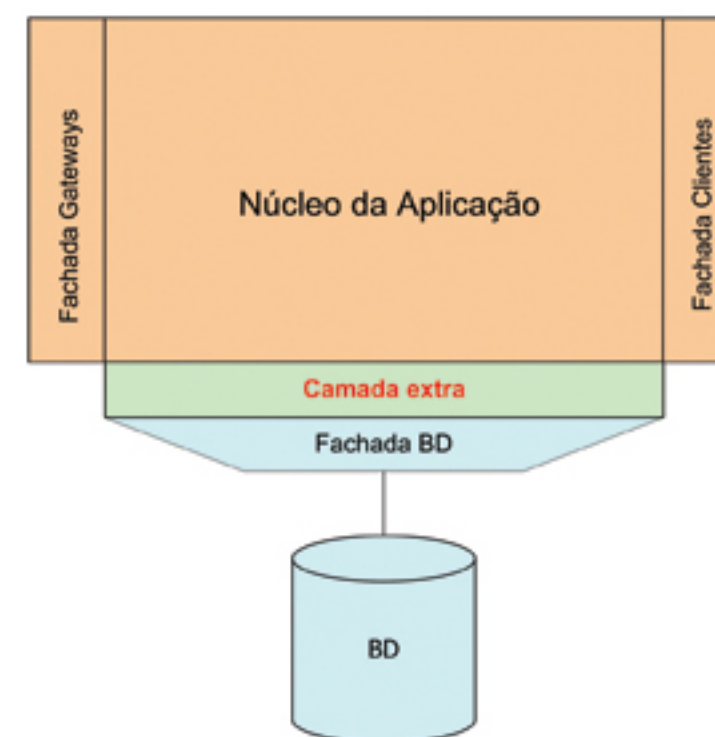
A aplicação na qual os testes foram feitos faz parte de um sistema de roteamento de ordens desenvolvido para uma importante corretora de valores. No Brasil, todo envio de ordem às bolsas de valores (Bovespa e BM&F) é feito através de uma corretora de valores. Os sistemas de roteamento de ordens consistem na transmissão de ordens de clientes às bolsas e vice versa.

O sistema de roteamento de ordens no qual os testes de persistência foram feitos é dividido em três partes. Servidores fazem a comunicação com os clientes, outros fazem a comunicação com as bolsas e um servidor central faz a integração, validações etc. O sistema foi desenvolvido em *Microsoft C#* e o gerenciador de banco de dados *Microsoft SQL Server*.



3. Análise

Foi criada uma camada extra na aplicação de forma que fosse possível fazer a substituição do **Hibernate** por outra solução, neste caso, a persistência direta por stored procedures, armazenadas e pré-compiladas no gerenciador de banco de dados.

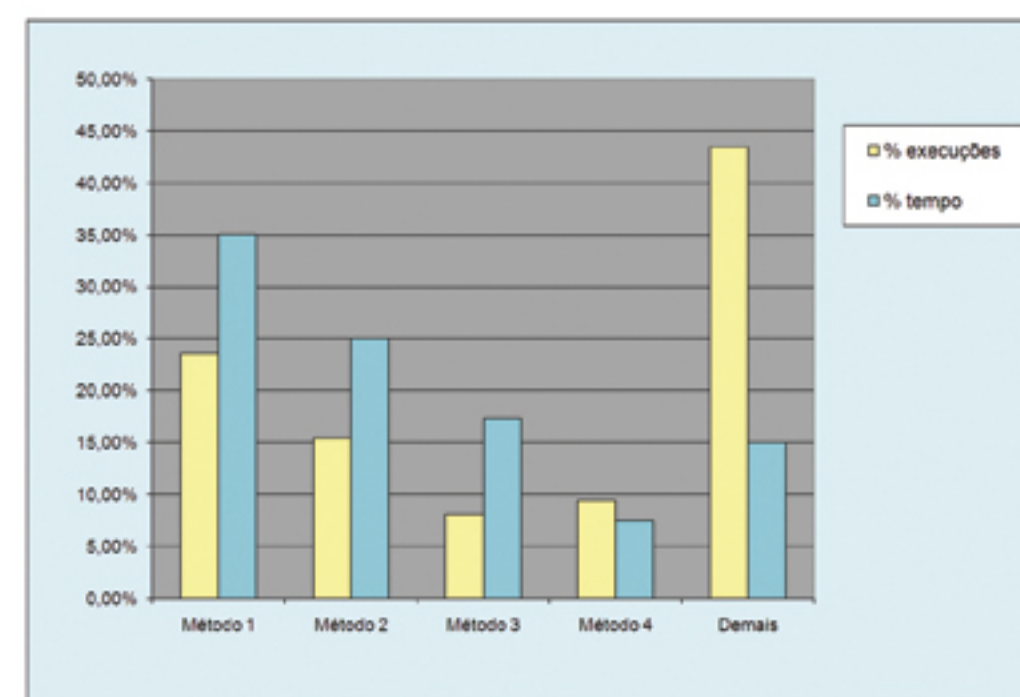


A camada extra foi criada através de uma Interface. Da seguinte forma.

```
interface IPersistencia
{
    Conta getConta(long idConta);
    Oferta getOferta(long idOferta);
    void saveOferta(Oferta oferta);
    /* (...) */
}
```

Foram coletadas estatísticas de quais métodos eram responsáveis pelo maior tempo gasto na camada de persistência.

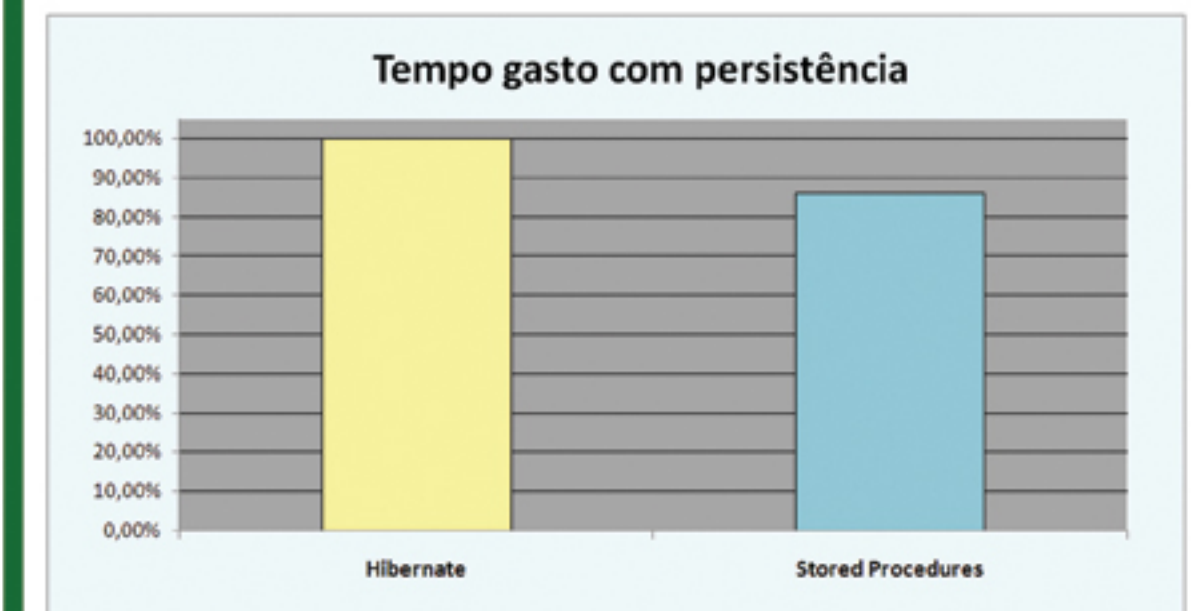
Apenas 4 métodos eram responsáveis por quase 85% de todo o tempo de execução. Tais métodos foram substituídos por acesso direto a stored procedures.



Além disso foram analisados quais métodos da aplicação eram mais custosos, com o intuito de criar procedures mais 'densas'. Desta forma seria possível tirar maior proveito das stored procedures pré-compiladas no gerenciador de banco de dados.

4. Resultados e Conclusões

A substituição dos quatro métodos de **Hibernate** mostrou um ganho de desempenho de aproximadamente 14%.



Além disso, a criação de stored procedures 'densas' trouxe o maior ganho de desempenho à aplicação. Houve métodos com ganho de até 50%.

O **Hibernate** é uma ferramenta extremamente útil que diminui muito o tempo gasto com desenvolvimento de persistência. No entanto, dependendo da demanda por desempenho, é justificada a criação de uma solução híbrida, unindo a praticidade do **Hibernate** e a performance de acesso direto ao gerenciador de banco de dados.

5. Mais informações

Um texto com mais detalhes da aplicação, dos testes e dos resultados logo estará disponível em:

<http://www.linux.ime.usp.br/~rogeriom/mac499/>