

# Métricas para Avaliação de Persistência de Dados num Ambiente Objeto-Relacional

## O Problema e a Solução

### Qualidade da Persistência

Problemas comuns nas implementações de ORM:

- Baixa qualidade da representação semântica;
- Recuperação em larga escala ineficiente.

Como avaliar a qualidade da persistência?

### Métricas e Testes

Missão do nosso trabalho:

- Definir métricas para avaliar a qualidade da persistência de dados feita pela ORM;
- Testar diferentes implementações de ORM utilizando essas métricas;
- Elaborar uma análise comparativa das implementações a partir dos testes.

## Etapas de Desenvolvimento

### Fase 1

Construção de aplicações modelo com um esquema comum de BD cada qual utilizando uma ferramenta específica para realizar o ORM. Serão utilizadas dentro de uma plataforma comum de testes.

### Fase 2

Poleposition: framework de testes (GPL) adaptado para os testes de performance e produção de relatórios. O Poleposition aplica o conceito de circuito. Cada aplicação modelo percorre uma volta no circuito e o framework coordena a corrida. [8].

### Fase 3

Definir maneiras seguras e eficazes de avaliar a qualidade da persistência de dados produzida num ambiente objeto-relacional pelas ferramentas.

Serão divididas em qualitativas e quantitativas.

### Fase 4

À partir das métricas e do ambiente prontos, as aplicações irão desempenhar uma mesma sequência de operações (voltas) dentro do circuito.

### Fase 5

Utilizar as métricas criadas para avaliar os resultados obtidos com os testes.

Gerar gráficos, relatórios, análises e conclusões.

## Implementações ORM

### Hibernate

É um framework para persistência de objetos, baseado no mapeamento objeto-relacional. Isto inclui:

- Metadados para a definição do mapeamento objeto-relacional;
- Dialeto SQL próprio chamado de HQL (Hibernate Query Language). Este dialeto é totalmente orientado a objetos, portanto temos polimorfismo, herança;
- Além de outras funcionalidades como Linguagem de Definição de Dados (DDL), Controle de Transação, SQL Embutida e Dinâmica, Integridade. [1]. [5]. [7].

### TopLink Essentials(JPA)

É uma implementação de referência da especificação Java Persistence API (JPA). Esta especificação provê um padrão POJO de persistência para o mapeamento objeto-relacional. Então:

- Pelo fato do Toplink Essentials ser uma implementação da especificação JPA e esta basear-se no Hibernate, Toplink e JDO, este framework contém muitas funcionalidades do próprio Hibernate, descrito anteriormente;
- Porém ele introduziu e padronizou as anotações Java para persistência, em contrapartida aos arquivos XML's. [2]. [6].

### Active Record(RoR)

É uma implementação do padrão de projeto, de mesmo nome, descrito por Martin Fowler, para mapeamento objeto-relacional. A implementação de estudo é a desenvolvida para Ruby, e que faz parte da camada de persistência do framework Ruby on Rails (RoR). O diferencial deste framework com relação aos outros reside na sua configuração:

- Não há metadados, quer dizer, não há arquivos XML de configuração, ou anotações;
- Não temos a chamada "build phase", ActiveRecord é configurado "on-the-fly". [3]. [4].

## Resultados

### Métricas

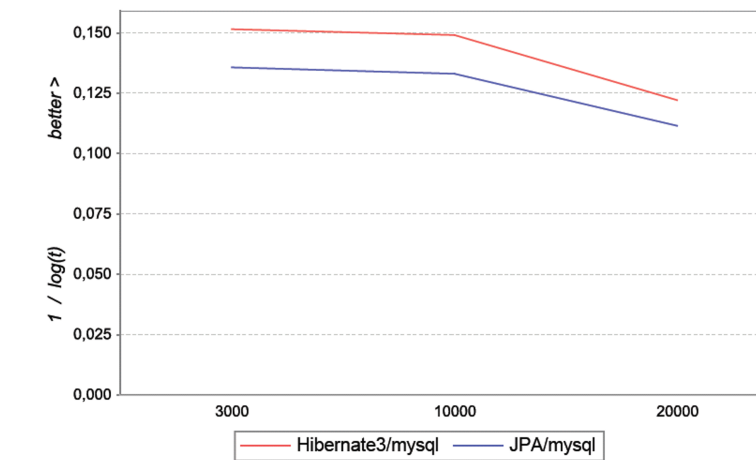
Os principais fatores a serem analisados pelas métricas são:

- Qualidade da representação semântica;
- Performance;
- Escalabilidade;
- Evolução;

Alunos: Diogo Vernier dos Santos  
Marcelo de Rezende Martins  
Orientador: João Eduardo Ferreira

Circuit: Matao  
writes, reads and deletes unstructured flat objects of one kind in bulk mode  
Lap: insercaoArvoreModelo

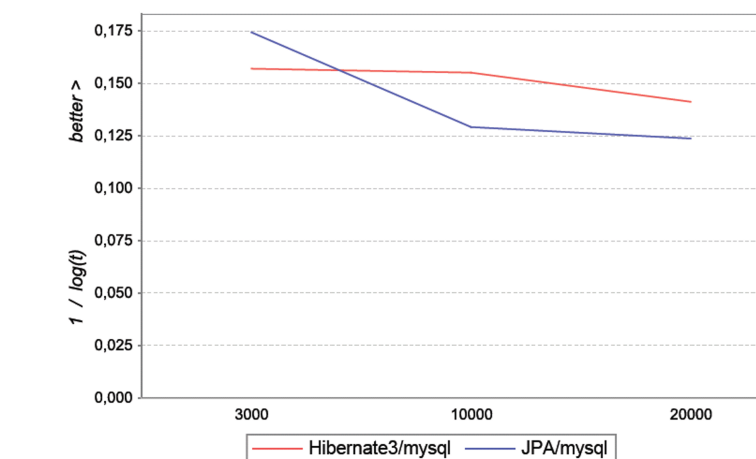
t [time in ms]	selectdepth:3 objects:3000 depth:7	selectdepth:6 objects:10000 depth:9	selectdepth:6 objects:20000 depth:11
Hibernate3/mysql	735	818	3595
JPA/mysql	1587	1836	7837



Inserção de entidades num relacionamento  
recursivo de grau  $n$

Circuit: Matao  
writes, reads and deletes unstructured flat objects of one kind in bulk mode  
Lap: consulta\_relacionamento\_recursivo

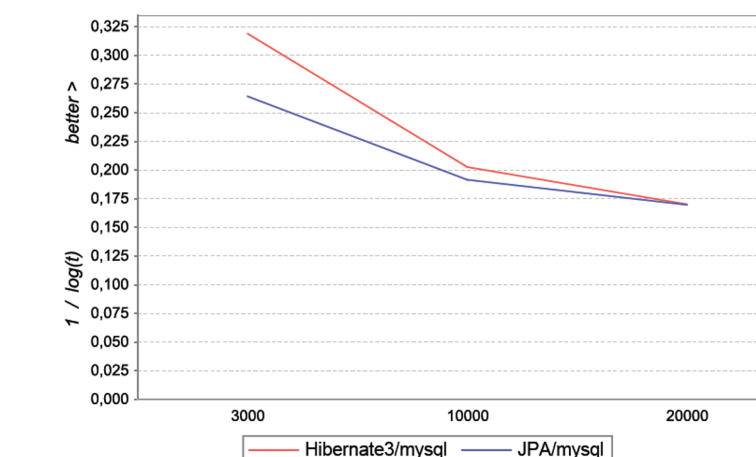
t [time in ms]	selectdepth:3 objects:3000 depth:7	selectdepth:6 objects:10000 depth:9	selectdepth:6 objects:20000 depth:11
Hibernate3/mysql	584	631	1189
JPA/mysql	310	2302	3241



Recuperação de entidades de grau  $n$  num  
relacionamento recursivo

Circuit: Matao  
writes, reads and deletes unstructured flat objects of one kind in bulk mode  
Lap: consulta\_relacionamento\_recursivo\_hot

t [time in ms]	selectdepth:3 objects:3000 depth:7	selectdepth:6 objects:10000 depth:9	selectdepth:6 objects:20000 depth:11
Hibernate3/mysql	23	139	356
JPA/mysql	44	185	362



Recuperação de entidades in cache num  
relacionamento recursivo

ORM Comparative Table			
	Hibernate	TopLink Essentials	ActiveRecord(RoR)
One-to-One	✓	✓	✓
One-to-Many	✓	✓	✓
Many-to-Many	✓	✓	✓
Ternary	✓	✗	✗
Inheritance			
Table per class hierarchy	✓	✓	✓
Table per subclass	✓	✓	✗
Table per concrete class	✓	✗	✗
Database Integrity(Constraints)	✓	✓	✗
Batch Operations			
Update and Delete	✓	✗	✗
Batch Insert	✓	✗	✗
Composite Keys	✓	✓	✗

Quadro Comparativo de Implementações ORM

## Referências

- [1] BAUER, Christina. KING, Gauvin. Java Persistence With Hibernate. Manning, 2007.
- [2] KEITH, Mike. SCHINCARIOL, Merrick. PRO EJB3:Java Persistence API. Apress, 2006.
- [3] THOMAS, Dave. Hansson, David H.. Agile Web Development with Rails: Second Edition. The Pragmatic Bookshelf, 2006
- [4] *Ruby on Rails: ActiveRecord*. Hansson, David Heinemeier. 2007. URL:<http://wiki.rubyonrails.org/rails/pages/ActiveRecord>.
- [5] *Hibernate*. Red Hat, 2006. URL: [http://www.hibernate.org/hib\\_docs/v3/reference/en/html/queryhql.html](http://www.hibernate.org/hib_docs/v3/reference/en/html/queryhql.html)
- [6] *Java Persistence API FAQ*. Sun Microsystems, 2007. URL: <http://java.sun.com/javaee/overview/faq/persistence.jsp>
- [7] SILBERSCHATZ, Abraham. KORTH, Henry F.. SUDARSHAN, S.. Database System Concepts, 5th Edition. McGraw-Hill, 2006.
- [8] *PolePosition: the open source database benchmark*. URL: <http://www.polepos.org/>