

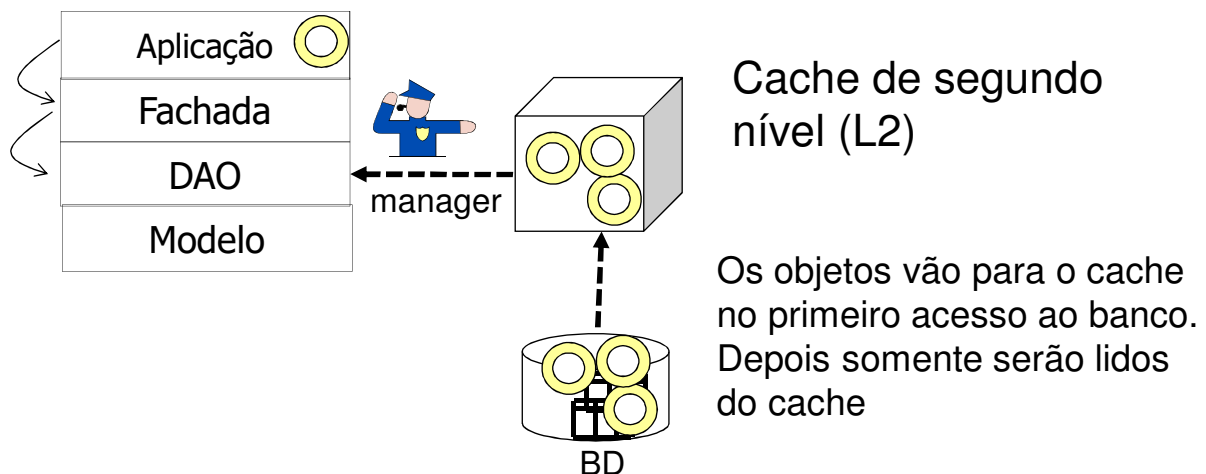
Persistência de Objetos

Fausto Maranhão Ayres

13 JPA – Cache de Objetos

Cache de objetos

- O cache de objetos é utilizado para guardar os objetos na memória durante a execução.



Cache de objetos

■ Vantagem

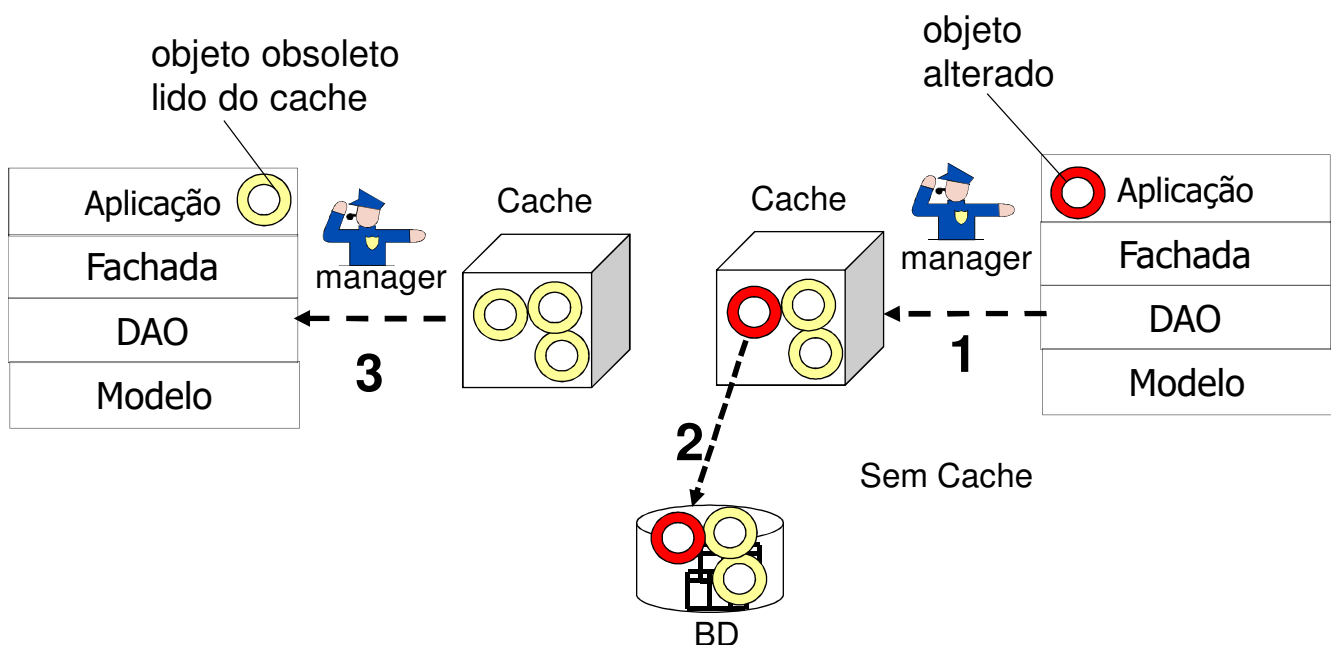
- Aumento da performance da aplicação quando os objetos são muito consultados e pouco alterados
 - Ex: Classe Estado/Cidade são raramente alteradas

■ Desvantagem:

- Aumento do consumo de memória, dependendo da classe
- Objetos podem ficar **obsoletos** no cache de outra **aplicação concorrente**

Objetos obsoletos

- Objetos podem ficar **obsoletos** no cache de outra **aplicação concorrente**



Habilitar/desabilitar uso do cache

- No JPA, pode-se habilitar o uso do cache de duas maneiras:
 - ESTÁTICA:
 - antes de instanciar o manager, configura-se o cache no **persistence.xml** e nas classes
 - DINÂMICA:
 - depois de instanciar o manager, configura-se operações **createQuery()** e **find()**

Habilitar/Desabilitar de forma ESTÁTICA

- Persistence.xml

```
<persistence-unit name="hibernate-postgres" ...  
<shared-cache-mode>NONE</shared-cache-mode>  
<properties> ... </properties>  
</persistence-unit>
```

Opções

- **ALL** – habilita o cache para todas as classes
- **NONE** – desabilita o cache para todas as classes
- **ENABLE_SELECTIVE** – desabilita o cache para todas as classes, exceto aquelas com **@Cacheable(true)**
- **DISABLE_SELECTIVE** – habilita o cache para todas as classes, exceto aquelas com **@Cacheable(false)**

Opções seletivas

- Opção ENABLE_SELECTIVE

@Entity

@Cacheable(true) //ativa o cache da classe

public class Vaga {...}

- Opção DISABLE_SELECTIVE

@Entity

@Cacheable(false) //desativa o cache da classe

public class Vaga {...}

Habilitar/Desabilitar de forma DINÂMICA

manager.createQuery()

Habilitar cache

TypedQuery<Vaga> query;

query = manager.createQuery("select v from Vaga v order by v.id", Vaga.class)

.setHint("javax.persistence.cache.retrieveMode", CacheRetrieveMode.USE)

.setHint("javax.persistence.cache.storeMode", CacheStoreMode.USE);

Desabilitar cache

TypedQuery<Vaga> query;

query = manager.createQuery("select v from Vaga v order by v.id", Vaga.class)

.setHint("javax.persistence.cache.retrieveMode", CacheRetrieveMode.BYPASS)

.setHint("javax.persistence.cache.storeMode", CacheStoreMode.REFRESH);

Habilitar/Desabilitar cache de forma DINÂMICA

manager.find()

Habilitar cache

```
Map<String, Object> propriedades = new HashMap<String, Object>();  
propriedades.put("javax.persistence.cache.retrieveMode", "USE");  
propriedades.put("javax.persistence.cache.storeMode", "USE");  
Vaga v = manager.find(Vaga.class, id, propriedades );
```

Desabilitar cache

```
Map<String, Object> propriedades = new HashMap<String, Object>();  
propriedades.put("javax.persistence.cache.retrieveMode", "BYPASS");  
propriedades.put("javax.persistence.cache.storeMode", "BYPASS");  
Vaga v = manager.find(Vaga.class, id, propriedades );
```

OBS

- Deve-se utilizar transação em consultas concorrentes.

Ex:

Fachada:

```
public static List<Vaga> listarVagas(){  
    DAO.begin();  
  
    List<Vaga> result = daovaga.readAll();  
  
    DAO.commit();  
    return result;  
}
```