



联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

等级： 博客 3

访问量：5万+

积分：725

排名：6万+

文章搜索

文章分类

Hibernate (16)

Spring (4)

Hadoop (1)

通讯 (1)

MyBatis (3)

阅读排行

Hibernate之缓存详解 (22929)

Spring(4)-IOC-Spring的核心... (8704)

Hibernate的批量插入 (&&J... (5027)

Hibernate---QBC查询(1) (3692)

Java使用短信猫发信息(向串... (3411)

Hibernate的批量更新与删除... (3083)

Hadoop单机版环境搭建及第... (1795)

Hibernate实体类中的注解 (1609)

Spring(2)--Spring核心机制... (1460)

初识Hibernate的几种方式 (1197)

最新评论

Hibernate之缓存详解
shilykong : [reply]u014231738[/reply] 漏了一点：一级缓存不会缓存普通属性，只会缓存整...

Hibernate之缓存详解
shilykong : 我也来个总结：1.hibernate 总共有三种形式的缓存，分为二级：一级session级别的缓...

Hibernate的批量插入 (&a...

tChat 论坛 ...

写博客

发Chat

目录视图 摘要视图 RSS 订阅

16

Hibernate之缓存详解

2016年08月03日 21:57:00 23066人阅读

7) 收藏 举报

分类： Hibernate (15)

版权声明：本文为博主原创文章，未经博主允许不得转载。 //blog.csdn.net/an_2016/article/details/52088712

Hibernate中提供了两级缓存，一级缓存是Session级别的缓存，它属于事务范围的缓存，该级缓存由hibernate管理，应用程序无需干预；二级缓存是SessionFactory级别的缓存，该级缓存可以进行配置和更改，并且可以动态加载和卸载，hibernate还为查询结果提供了一个查询缓存，它依赖于二级缓存；

一，缓存的概念

缓存是位于应用程序和永久性数据存储源之间用于临时存放复制数据的内存区域，缓存可以降低应用程序之间读写永久性数据存储源的次数，从而提高应用程序的运行性能；

hibernate在查询数据时，首先会到缓存中查找，如果找到就直接使用，找不到时才从永久性数据存储源中检索，因此，把频繁使用的数据加载到缓存中，可以减少应用程序对永久性数据存储源的访问，使应用程序的运行性能得以提升；

二，缓存的范围

缓存范围决定了缓存的生命周期，缓存范围分为3类：

1>事务范围

缓存只能被当前事务访问，缓存的生命周期依赖于事务的生命周期，事务结束时，缓存的生命周期也结束了；

2>进程范围

缓存被进程内的所有事务共享，这些事务会并发访问缓存，需要对缓存采用必要的事务隔离机制，缓存的生命周期取决与进程的生命周期，进程结束，缓存的生命周期也结束了；

3>集群范围

缓存被一个或多个计算机的进程共享，缓存中的数据被复制到集群中的每个进行节点，进程间通过远程通信来保证缓存中数据的一致性；

在查询时，如果在事务范围内的缓存中没有找到，可以到进程范围或集群范围的缓存中查找，如果还没找到，则到数据库中查询；

三，Hibernate中的第一级缓存

Hibernate的一级缓存由Session提供，只存在于Session的生命周期中，当应用程序调用Session接口的save(),update(),saveOrupDate(),get(),load()或者Query和Criteria实例的list(),iterate()等方法时，如果Session缓存中没有相应的对象，hibernate就会把对象加入到一级缓存中，当session关闭时，该Session所管理的一级缓存也会立即被清除；

1，get查询测试：

1>在同一个session中发出两次get查询

[java]

1. public void Query(){

2. Session sess = HibernateSessionFactory.getSession();

http://blog.csdn.net/an_2016/article/details/52088712

1/10

v风雪山神庙v · 感谢 我用了idhc直连的方式

联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```

3. Transaction tx = sess.beginTransaction();
4. Student s1 = (Student)sess.get(Student.class, 2);
5. System.out.println(s1.getName());
6. Student s2 = (Student)sess.get(Student.class, 2);
7. System.out.println(s2.getName());
8. tx.commit();
9. HibernateSessionFactory.closeSession();
10. }</strong></span>

```

上面的两次查询，第一次执行了get方法查询了数据库，产生了一条sql语句，第二次执行get方法时，由于在一级缓存中找到了该对象，因此不会查询数据库，不再发出sql语句；

2> 开启两个session中发出两次get查询

```

[java]
1. <span style="font-size:18px;"><strong> public void Query(){
2.     Session sess1 = HibernateSessionFactory.getSession();
3.     Transaction tx1 = sess1.beginTransaction();
4.     Student s1 = (Student)sess1.get(Student.class, 2);
5.     System.out.println(s1.getName());
6.     tx1.commit();
7.     HibernateSessionFactory.closeSession();
8.     Session sess2 = HibernateSessionFactory.getSession();
9.     Transaction tx2 = sess2.beginTransaction();
10.    Student s2 = (Student)sess2.get(Student.class, 2);
11.    System.out.println(s2.getName());
12.    tx2.commit();
13.    HibernateSessionFactory.closeSession();
14. }</strong></span>

```

上面的两次查询，两次执行get方法时都查询了数据库，产生了两条sql语句，原因在于，第一次执行get方法查询出结果后，关闭了session，缓存被清除了，第二次执行get方法时，从缓存中找不到结果，只能到数据库查询；

2，iterate查询测试

插入一个iterator查询方式：

```

[java]
1. <span style="font-size:18px;"><strong> public void Query(){
2.     Session sess = HibernateSessionFactory.getSession();
3.     Transaction tx = sess.beginTransaction();
4.     Query query = sess.createQuery("from Student");
5.     Iterator iter = query.iterate();
6.     while(iter.hasNext()){
7.         System.out.println(((Student)iter.next()).getName());
8.     }
9.     tx.commit();
10.    HibernateSessionFactory.closeSession();
11. }</strong></span>

[java]
1. <span style="font-size:18px;"><strong> public void Query(){
2.     Session sess = HibernateSessionFactory.getSession();
3.     Transaction tx = sess.beginTransaction();
4.     Student s1 = (Student)sess.createQuery("from Student s where s.id = 2").iterate().next
5.     ();
6.     System.out.println(s1.getName());
7.     Student s2 = (Student)sess.createQuery("from Student s where s.id = 2").iterate().next
8.     ();
9.     System.out.println(s2.getName());
10.    tx.commit();
11.    HibernateSessionFactory.closeSession();
12. }</strong></span>

```

上面的代码执行后会生成三条sql语句，第一次执行iterate().next()时会发出查询id的sql语句(第一条sql语句)，得到s1对象，使用s1对象获得name属性值时会发出相应的查询实体对象的sql语句(第二条sql语句)，第二次执行iterate().next()时会发出查询id的sql语句(第三条sql语句)，但是不会发出查询实体对象的sql语句，因为hibernate使用缓存，不会发出sql语句

3，iterate查询属性测试：

```

[java]
1. <span style="font-size:18px;"><strong> public void Query(){
2.     Session sess = HibernateSessionFactory.getSession();

```

联系我们



请扫描二维码联系客服
 webmaster@csdn.net
 400-660-0108
 QQ客服 客服论坛

关于 招聘 广告服务 百度
 ©1999-2018 CSDN版权所有
 京ICP证09002463号

经营性网站备案信息
 网络110报警服务
 中国互联网举报中心
 北京互联网违法和不良信息举报中心

```

3. Transaction tx = sess.beginTransaction();
4. String name1 = (String) sess.createQuery("select s.name from Student s where s.id = 2").iterate().next();
5. System.out.println(name1);
6. String name2 = (String) sess.createQuery("select s.name from Student s where s.id = 2").iterate().next();
7. System.out.println(name2);
8. tx.commit();
9. HibernateSessionFactory.closeSession();
10. }</strong></span>

```



上面的代码第一次执行`iterate().next()`时发出查询属性的sql语句，第二次执行`iterate().next()`时也会发出查询属性的sql语句，这是因为`iterate`查询普通属性，一级缓存不会缓存，所以会发出sql；

4，在一个session中先save，再执行load查询

```

[java]
1. <span style="font-size:18px;"><strong> public void Query(){
2.     Session sess = HibernateSessionFactory.getSession();
3.     Transaction tx = sess.beginTransaction();
4.     Student s = new Student(8, "newAcc", 8, 8);
5.     Serializable id = sess.save(s);
6.     tx.commit();
7.     Student s1 = (Student) sess.load(Student.class, 8);
8.     System.out.println(s1.getName());
9.     HibernateSessionFactory.closeSession();
10. }</strong></span>

```

上面的代码执行save操作时，它会在缓存里放一份，执行load操作时，不会发出sql语句，因为save使用了缓存；

Session接口为应用程序提供了两个管理缓存的方法：

1> `evict()`方法：用于将某个对象从Session的一级缓存中清除；

2> `clear()`方法：用于将一级缓存中的所有对象全部清楚；

测试：

```

[java]
1. <span style="font-size:18px;"><strong> public void Query(){
2.     Session sess = HibernateSessionFactory.getSession();
3.     Transaction tx = sess.beginTransaction();
4.     Student s = (Student) sess.load(Student.class, 1);
5.     System.out.println(s.getName());
6.     sess.clear(); //清除一级缓存中的所有对象
7.     Student s1 = (Student) sess.load(Student.class, 1);
8.     System.out.println(s1.getName());
9.     tx.commit();
10.     HibernateSessionFactory.closeSession();
11. }</strong></span>

```

从上面的代码可以看出，`clear`方法可以管理一级缓存，一级缓存无法取消，但是可以管理，第一次执行load操作时会发出sql语句，接着由于一级缓存中的实体被清除了，因此第二次执行load操作时也会发出sql语句；

四，Hibernate中的第二级缓存

二级缓存是一个可插拔的缓存插件，它是由SessionFactory负责管理的；

由于SessionFactory对象的生命周期与应用程序的整个过程对应，通常一个应用程序对应一个SessionFactory，因此，二级缓存是进程范围或者集群范围的缓存；

与一级缓存一样，二级缓存也是根据对象的id来加载与缓存，当执行某个查询获得结果集为实体对象集时，hibernate就会把它们按照对象id加载到二级缓存中，在访问指定的id的对象时，首先从一级缓存中查找，找到就直接使用，找不到则转到二级缓存中查找(必须配置且启用二级缓存)，如果二级缓存中找到，则直接使用，否则会查询数据库，并将查询结果根据对象的id放到缓存中；

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

1, 常用的二级缓存插件

Hibernate的二级缓存功能是通过配置二级缓存插件来实现的, 常用的二级缓存插件包括EHCACHE, OSCACHE, SWARMCACHE和JBOSSCACHE。其中EHCACHE缓存插件是理想的进程范围的缓存实现, 此处以使用EHCACHE缓存插件为例, 来介绍如何使用hibernate的二级缓存;

2, Hibernate中使用EHCACHE的配置

1>引入EHCACHE相关的jar包;

lib\optional\ehcache下的三个jar包;

2>创建EHCACHE的配置文件ehcache.xml

```
[html]
1.  <span style="font-size:18px;"><strong><ehcache>
2.    <diskStore path="java.io.tmpdir"/>
3.    <defaultCache
4.      maxElementsInMemory="10000"
5.      eternal="false"
6.      timeToIdleSeconds="120"
7.      timeToLiveSeconds="120"
8.      overflowToDisk="true"
9.    />
10.   <cache name="sampleCache1"
11.     maxElementsInMemory="10000"
12.     eternal="false"
13.     timeToIdleSeconds="300"
14.     timeToLiveSeconds="600"
15.     overflowToDisk="true"
16.   />
17.   <cache name="sampleCache2"
18.     maxElementsInMemory="1000"
19.     eternal="true"
20.     timeToIdleSeconds="0"
21.     timeToLiveSeconds="0"
22.     overflowToDisk="false"
23.   />
24. </ehcache></strong></span>
```

在上述配置中, diskStore元素设置缓存数据文件的存储目录; defaultCache元素设置缓存的默认数据过期策略; cache元素设置具体的命名缓存的数据过期策略。每个命名缓存代表一个缓存区域, 命名缓存机制允许用户在每个类以及类的每个集合的粒度上设置数据过期策略;

在defaultCache元素中, maxElementsInMemory属性设置缓存对象的最大数目; eternal属性指定是否永不过期, true为不过期, false为过期; timeToldleSeconds属性设置对象处于空闲状态的最大秒数; timeToLiveSeconds属性设置对象处于缓存状态的最大秒数; overflowToDisk属性设置内存溢出时是否将溢出对象写入硬盘;

3>在Hibernate配置文件里面启用EHCACHE

在hibernate.cfg.xml配置文件中, 启用EHCACHE的配置如下:

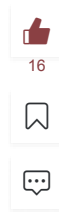
```
[html]
1.  <span style="font-size:18px;"><strong><!-- 启用二级缓存 -->
2.    <property name="hibernate.cache.use_second_level_cache">true</property>
3.    <!-- 设置二级缓存插件EHCACHE的Provider类 -->
4.    <property name="hibernate.cache.region.factory_class">
5.      org.hibernate.cache.ehcache.EhCacheRegionFactory</property></strong></span>
```

4>配置哪些实体类的对象需要二级缓存, 有两种方式:

1>>在实体类的映射文件里面配置

在需要进行缓存的持久化对象的映射文件中配置相应的二级缓存策略, 如User,hbm.xml:

```
[html]
1.  <span style="font-size:18px;"><strong><?xml version="1.0" encoding="UTF-8"?>
2.  <!DOCTYPE hibernate-mapping PUBLIC
3.  "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4.  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
```



16

联系我们



请扫描二维码联系客服
 webmaster@csdn.net
 400-660-0108
 QQ客服 客服论坛

关于 招聘 广告服务 百度
 ©1999-2018 CSDN版权所有
 京ICP证09002463号

经营性网站备案信息
 网络110报警服务
 中国互联网举报中心
 北京互联网违法和不良信息举报中心

```

5. <hibernate-mapping>
6.   <class name="com.hibtest1.entity.User" table="user" catalog="bookshop">
7.     <cache usage="read-write"/>
8.     <id name="id" type="java.lang.Integer">
9.       <column name="Id" />
10.      <generator class="native" />
11.    </id>
12.    <property name="loginName" type="java.lang.String">
13.      <column name="LoginName" length="50" />
14.    </property>
15.
16.  </class>
17. </hibernate-mapping></strong></span>
  
```



映射文件中使用<cache>元素设置持久化类User的二级缓存并发访问策略，usage属性取值为read-only时表示只读型并发访问策略；read-write表示读写型并发访问策略；nonstrict-read-write表示非严格读写型并发访问策略；EHCACHE插件不支持transactional（事务型并发访问策略）。

注意：<cache>元素只能放在<class>元素的内部，而且必须处在<id>元素的前面，<cache>元素放在哪些<class>元素下面，就说明会对哪些类进行缓存；

2>>在hibernate配置文件中统一配置，强烈推荐使用这种方式：

在hibernate.cfg.xml文件中使用<class-cache>元素来配置哪些实体类的对象需要二级缓存：

```

[java]
1. <span style="font-size:18px;"><strong><span style="font-size:18px;"><strong><span style="font-size:18px;"><strong><class-cache usage="read-only" class="com.anlw.entity.Student"/></strong></span></strong></span></strong></span>
  
```

在<class-cache>元素中，usage属性指定缓存策略，需要注意<class-cache>元素必须放在所有<mapping>元素的后面；

3，Hibernate中使用EHCACHE的测试：

```

[java]
1. <span style="font-size:18px;"><strong> public void Query(){
2.     Session sess1 = HibernateSessionFactory.getSession();
3.     Student s1 = (Student)sess1.get(Student.class, 1);
4.     System.out.println(s1.getName());
5.     HibernateSessionFactory.closeSession();
6.     Session sess2 = HibernateSessionFactory.getSession();
7.     Student s2 = (Student)sess2.get(Student.class, 1);
8.     System.out.println(s2.getName());
9.     HibernateSessionFactory.closeSession();
10. }</strong></span>
  
```

上面的代码，第一次执行get方法查询出结果后，关闭了session，一级缓存被清除了，由于配置并启用了二级缓存，查询出的结果会放入二级缓存，第二次执行get方法时，首先从一级缓存中查找，没有找到，然后转到二级缓存查找，二级缓存中找到结果，就不需要从数据库查询了。

注意：在hibernate配置二级缓存时属性的顺序如下，顺序错了会空指针异常：

```

[java]
1. <span style="font-size:18px;"><strong> <!-- 启用二级缓存 -->
2.     <property name="hibernate.cache.use_second_level_cache">true</property>
3.     <!-- 设置二级缓存插件EHCACHE的Provider类 -->
4.     <property name="hibernate.cache.region.factory_class">
5.       org.hibernate.cache.ehcache.EhCacheRegionFactory</property>
6.     <mapping class="com.anlw.entity.Student"/>
7.     <class-cache usage="read-only" class="com.anlw.entity.Student"/></strong></span>
  
```

先缓存配置，再mapping，最后class-cache；

五，Hibernate中的查询缓存

对于经常使用的查询语句，如果启用了查询缓存，当第一次执行查询语句时，hibernate会将查询结果存储在二级缓存中，以后再次执行该查询语句时，从缓存中获取查询结果，

联系我们



请扫描二维码联系客服
 webmaster@csdn.net
 400-660-0108
 QQ客服 客服论坛

关于 招聘 广告服务 百度
 ©1999-2018 CSDN版权所有
 京ICP证09002463号

经营性网站备案信息
 网络110报警服务
 中国互联网举报中心
 北京互联网违法和不良信息举报中心

从而提高查询性能；

hibernate的查询缓存主要是针对普通属性结果集的缓存，而对于实体对象的结果集只缓存id；

查询缓存的生命周期，若当前关联的表发生修改，那么查询缓存的生命周期结束；

1，查询缓存的配置

查询缓存基于二级缓存，使用查询缓存前，必须首先配置好二级缓存：

在配置了二级缓存的基础上，在hibernate的配置文件hibernate.cfg.xml中添加如下配置，可以启用查询缓存：

```
<property name="hibernate.cache.use_query_cache">false</property>
```

此外在程序中还必须手动启用查询缓存：

```
query.setCacheable(true);
```

2，测试查询缓存：

1> 开启查询缓存，关闭二级缓存，开启一个session，分别调用query.list查询属性，测试前，先在先在hibernate.cfg.xml文件中开启查询缓存，关闭二级缓存，如下所示：

```
<property name="hibernate.cache.use_query_cache">true</property>
<property name="hibernate.cache.use_second_level_cache">false</property>
```

```
[java]
1.  <span style="font-size:18px;"><strong> public static void query(){
2.      Session sess = HibernateSessionFactory.getSession();
3.      Transaction tx = sess.beginTransaction();
4.      Query query = sess.createQuery("select s.name from Student s");
5.      query.setCacheable(true);
6.      List names = query.list();
7.      for(Iterator iter = names.iterator();iter.hasNext();){
8.          String name = (String)iter.next();
9.          System.out.println(name);
10.     }
11.     System.out.println("-----");
12.     query = sess.createQuery("select s.name from Student s");
13.     query.setCacheable(true);
14.     names = query.list();
15.     for(Iterator iter = names.iterator();iter.hasNext();){
16.         String name = (String)iter.next();
17.         System.out.println(name);
18.     }
19.     tx.commit();
20.     HibernateSessionFactory.closeSession();
21. }
22. </strong></span>
```

第二次没有去查数据库，因为启用了查询缓存；

2> 开启查询缓存，关闭二级缓存，开启两个session，分别调用query.list查询属性，测试前，先在先在hibernate.cfg.xml文件中开启查询缓存，关闭二级缓存，如下所示：

```
<property name="hibernate.cache.use_query_cache">true</property>
<property name="hibernate.cache.use_second_level_cache">false</property>
```

```
[java]
1.  <span style="font-size:18px;"><strong> public static void query(){
2.      Session sess1 = HibernateSessionFactory.getSession();
3.      Transaction tx1 = sess1.beginTransaction();
4.      Query query = sess1.createQuery("select s.name from Student s");
5.      query.setCacheable(true);
6.      List names = query.list();
7.      for(Iterator iter = names.iterator();iter.hasNext();){
8.          String name = (String)iter.next();
9.          System.out.println(name);
10.     }
11.     tx1.commit();
12.     HibernateSessionFactory.closeSession();
13.     System.out.println("-----");
14.     Session sess2 = HibernateSessionFactory.getSession();
15.     Transaction tx2 = sess2.beginTransaction();
16.     query = sess2.createQuery("select s.name from Student s");
```


联系我们



请扫描二维码联系客服
 webmaster@csdn.net
 400-660-0108
 QQ客服 客服论坛

关于 招聘 广告服务 百度
 ©1999-2018 CSDN版权所有
 京ICP证09002463号

经营性网站备案信息
 网络110报警服务
 中国互联网举报中心
 北京互联网违法和不良信息举报中心

```

17. query.setCacheable(true);
18. names = query.list();
19. for(Iterator iter = names.iterator();iter.hasNext();){
20.     String name = (String)iter.next();
21.     System.out.println(name);
22. }
23. tx2.commit();
24. HibernateSessionFactory.closeSession();
25. }</strong></span>

```



第二次没有去查数据库，因为查询缓存生命周期与session生命周期无关；

3>开启查询缓存，关闭二级缓存，开启两个session，分别调用query.list查询实体对象，测试前，先在hibernate.cfg.xml文件中开启查询缓存，关闭二级缓存，如下所示：

```

<property name="hibernate.cache.use_query_cache">true</property>
<property name="hibernate.cache.use_second_level_cache">false</property>

```

```

[java]
1. <span style="font-size:18px;"><strong> public static void query(){
2.     Session sess1 = HibernateSessionFactory.getSession();
3.     Transaction tx1 = sess1.beginTransaction();
4.     Query query = sess1.createQuery("from Student");
5.     query.setCacheable(true);
6.     List student = query.list();
7.     for(Iterator iter = student.iterator();iter.hasNext();){
8.         Student s = (Student)iter.next();
9.         System.out.println(s.getName()+"-"+s.getAge());
10.    }
11.    tx1.commit();
12.    HibernateSessionFactory.closeSession();
13.    System.out.println("-----");
14.    Session sess2 = HibernateSessionFactory.getSession();
15.    Transaction tx2 = sess2.beginTransaction();
16.    query = sess2.createQuery("from Student");
17.    query.setCacheable(true);
18.    student = query.list();
19.    for(Iterator iter = student.iterator();iter.hasNext();){
20.        Student s = (Student)iter.next();
21.        System.out.println(s.getName()+"-"+s.getAge());
22.    }
23.    tx2.commit();
24.    HibernateSessionFactory.closeSession();
25. }</strong></span>

```

查询结果如下：

```

[java]
1. <span style="font-size:18px;"><strong>Hibernate:
2.     select
3.         student0_.id as id1_0_,
4.         student0_.age as age2_0_,
5.         student0_.name as name3_0_
6.     from
7.         student student0_
8. anliwaaaa--1
9. test--2
10. -----
11. Hibernate:
12.     select
13.         student0_.id as id1_0_,
14.         student0_.age as age2_0_,
15.         student0_.name as name3_0_
16.     from
17.         student student0_
18.     where
19.         student0_.id=?
20. Hibernate:
21.     select
22.         student0_.id as id1_0_,
23.         student0_.age as age2_0_,
24.         student0_.name as name3_0_
25.     from
26.         student student0_
27.     where
28.         student0_.id=?
29. anliwaaaa--1
30. test--2</strong></span>

```

第二次查询数据库时，会发出n条sql语句，因为开启了查询缓存，关闭了二级缓存，那么查询缓存会缓存实体对象的id，所以hibernate会根据实体对象的id去查询相应的实体，如

联系我们



请扫描二维码联系客服

 webmaster@csdn.net

 400-660-0108

 QQ客服  客服论坛

关于 招聘 广告服务  百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

果缓存中不存在相应的实体，那么将发出根据实体id查询的sql语句，否则不会发出sql，使用缓存中的数据；

4> 开启查询缓存，开启二级缓存，开启两个session，分别调用query.list查询实体对象，测试前，先在hibernate.cfg.xml文件中开启查询缓存，开启二级缓存，如下所示：


```
<property name="hibernate.cache.use_query_cache">true</property>
<property name="hibernate.cache.use_second_level_cache">true</property>
```

代码和3>一样，但是结果不同，第二次不会发出sql，因为开启了二级缓存和缓存查询，查询缓存缓存了实体对象的id，hibernate会根据实体对象的id到数据库中取得相应的数据；

- 上一篇 Hadoop单机版环境搭建及第一个示例
- 下一篇 Spring(1)--初识Spring之HelloWorld

0/

查看评论



shilykong

5楼 2017-12-26 21:12发表

我也来个总结：


1.hibernate 总共有三种形式的缓存，分为二级：一级session级别的缓存，二级sessionfactory级别的缓存。二级缓存有两种形式，一是外部插件(EHCache)，二是查询缓存，查询缓存基于外部插件存在，要使用查询缓存，必须先配置好二级缓存。

2.一级缓存和二级缓存都是根据Id来加载于缓存的

3.查询缓存大多数是用来缓存实体属性的结果集的，如果查询结果为实体对象的集合时，查询缓存里就只有实体对象的Id


查看更多评论

关于hibernate的缓存使用

 woshichenxu 2006年01月22日 20:06 53671

1. 关于hibernate缓存的问题：1.1.1. 基本的缓存原理Hibernate缓存分为二级，第一级存放于session中称为一级缓存，默认带有且不能卸载。第二级是...

Hibernate学习笔记（六）——缓存

 u011024652 2016年10月09日 21:10 2311


一、Hibernate缓存 1、什么是缓存？Hibernate中用到了缓存的概念，那么什么是缓存呢？这里介绍的缓存并不是指计算机的内存或者CPU的一二级缓存，这里的缓存是指为了降低程序对物理数据...

Hibernate中性能优化之缓存

 u013037201 2016年07月31日 22:05 777


在Hibernate中，提到性能优化，很自然地我们就想到了缓存。缓存是什么，都有哪些呢？接下来就来分享...

hibernate 缓存

 u012743772 2016年04月11日 11:13 3228

简介缓存 缓存是广泛使用的用于优化数据库应用程序。缓存的目的是减少你的应用程序，并通过保存从数据库已加载数据的数据库之间的流量。检索数据当前未在高速缓存仅当数据库访问是必要的。应用程序可能需要从时间空...

Hibernate中的缓存机制与可能遇到的问题

 ls5718 2016年08月11日 15:52 4094

一、一级缓存(session级别) 概念 我们来看看hibernate提供的一级缓存： /** * 此时会发出一条sql，将所有学生全部查询出来，并放到s...

联系我们



请扫描二维码联系客服

 webmaster@csdn.net

 400-660-0108

 QQ客服  客服论坛

关于 招聘 广告服务  百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

Hibernate二级缓存，使用Ehcache缓存框架

1、首先下载第三方ehcache.jar ehcache.jar包的话，有两种，一种是org.ehcache，一种是net.sf.ehcache。Hibernate集成的是net.sf.eh...

 yh_zeng2 2017年07月06日 19:43  210

 16



hibernate 缓存

 weixin_39893693 2018年01月27日 15:41  20

定义： 高速缓存是数据的一个本地副本，处在应用程序和数据库之间，用于提升查询效率 区别优点： 使用ORM持久层远胜于直接用JDBC的另一个重要原因是：高速缓存。 使用缓存涉及到三个...

Hibernate ehcache缓存配置详解+实战

依赖管理我这里使用的maven来管理的依赖，pom如下： org.hibernate hibernate-core 5.0.2.Final org....

 u011404265 2017年01月10日 16:37  2393

Hibernate缓存一级缓存

 king624498030 2015年08月05日 18:20  650

Hibernate缓存一级缓存 Hibernate缓存二级缓存 Hibernate缓存查询缓存 Hibernate缓存总结 hibernate一级缓存只对当前session有用， ...

新手上路之Hibernate：第一个Hibernate例子

一、Hibernate概述 （一）什么是Hibernate？ Hibernate核心内容是ORM（关系对象模型）。可以将对象自动的生成数据库中的信息，使得开发更加的面向对象。这样作为程序员就可以使用面...

 aboy123 2013年08月21日 19:33  96589

梳理hibernate篇-原理及缓存机制

 u014586894 2016年06月11日 00:52  5629

Hibernate工作原理 现在我们知道了一个概念Hibernate Session，只有处于Session管理下的POJO才具有持久化操作能力。当应用程序对于处于Session管理下的...



Hibernate之缓存详解

 a724888 2017年12月06日 23:11  55

Hibernate之缓存详解 分类： Hibernate（15） 版权声明：本文为博主原创文章，未经博主允许不得转载。 Hibernate中提供了两...

hibernate缓存机制详细分析(一级、二级、查询缓存，非常清晰明白)

在本篇随笔里将会分析一下hibernate的缓存机制，包括一级缓存(session级别)、二级缓存(sessionFactory级别)以及查询缓存，当然还要讨论下我们的N+1的问题。 随笔虽长，但我...

 w1033162186 2016年02月05日 16:33  872

Hibernate 性能优化及缓存的使用

 liangwenmail 2015年08月08日 21:55  1022

hibernate性能优化，及二级缓存的配置使用。

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

浅谈Hibernate3中的缓存

当你的系统出现性能瓶颈时，改善业务逻辑



VincenLlin 2014年05月25日 19:34 526



16



性能优化（一）Hibernate 利用缓存（一级、二级、查询）提高系统性能

查询缓存 一级缓存 二级缓存



lilongsheng1125 2014年11月17日 10:39 17700

Hibernate 之Hibernate缓存



lrcoop 2014年02月12日 11:35 3110

1、缓存：缓存是什么，解决什么问题？位于速度相差较大的两种硬件/软件之间的，用于协调两者数据传输速度差异的结构，均可称之为 Cache。目的：让数据更接近于应用程序，协调速度不匹配，使访问速度更快...

Hibernate 缓存机制



qq_35845339 2017年10月10日 16:06 44

===Hibernate 缓存机制 一、why（为什么要用Hibernate缓存？）Hibernate是一个持久层框架，经常访问物理数据库。为了降低应用程序对物理数据源访问的频...

hibernate缓存机制详细分析(一级/二级/查询缓存)

在本篇随笔里将会分析一下hibernate的缓存机制，包括一级缓存(session级别)、二级缓存(sessionFactory级别)以及查询缓存，当然还要讨论下我们的N+1的问题。随笔虽长，但我...



zmx729618 2016年04月29日 14:19 1503

Hibernate缓存机制和MyBatis缓存机制

原文地址：<https://my.oschina.net/u/1445731/blog/416200?p=%7B%7BtotalPage%7D%7D> 摘要: mybatis缓存分为1级缓存和2级缓存 ...



tanga842428 2016年09月29日 09:42 10552