Design of Online Examination System Based on Java EE

Xiaoyun Liu[1, a], Jian Shen[2,b]

[1] School of foreign languages, Jiangxi Science & Technology Normal University, Fenglin Avenue 605#, Nanchang, Jiangxi P.R.China 330013

[2] School of Civil Engineering and Architecture, Jiangxi Science & Technology Normal University, Fenglin Avenue 605#, Nanchang, Jiangxi P.R.China 330013

[a]505132338@qq.com, [b]894282069@qq.com

Key words: the online examination system, hibernate, load balancing

Abstract: The online examination system is based on Java EE. The lightweight Java EE framework technology, such as Struts + Hibernate, is applied to implement this subsystem. In order to improve the system ability of concurrent access, the Ehcache is added in data persistence layer as Hibemate second level cache. The DataGuard scheme for the database and the WEB application server load balancing are carried out to improve the system availability.

Introduction

The main problem of online examination system establishment and stable operation is the application of the storage of data and cache. The three layers structure of Java EE is referred to the presentation layer, business logic layer and the infrastructure layer[1]. The most commonly expansion is used by increasing the domain layer and data persistence layer between the infrastructure layer to facilitate saving the data[2,3]. To improve the performance of the application, the cache is used to reduce the frequency of the direct access to the database[4]. The database is applied to directly access only when the retrieving data is not in cache.

Data Persistence

The data persistence layer of this system is implemented by Hibernate. All the entities in the system and the relationship between entities and entity is adopted by the class structure definition. The mapping relationship between the PO(persistent objects) and the table is implemented by the Hibernate maintenance in the data persistence layer. By using the Hibernate persistence layer, the direct manipulation database using JDBC can be avoided.

In hibernate, each data table is corresponded with an entity class actually and each entity class has a corresponding configuration file (hbm.xml) to match. Not only the tables in database can be generated automatically by Hibernate according to the POJO class and the corresponding hbm.xml mapping files, but also the data table corresponding POJO class and the corresponding hbm.xml mapping files can be obtained by using hibernate Reverse Engineering function in MyEclipse after the corresponding data table in the database management system was designed and created

After the PO(persistent objects) being designed, it is needed to map hibernate persistence classes. So the associated entities table can be accessed persistently. Mapping file to complete is that the mapping relationship between the data table and the persistent classes. The persistent classes corresponding data table means the persistent instance corresponds to one row of data table and the persistent class attribute fields for the data table. The code snippet of the PO class EntranceExam mapping file EntranceExam.hbm.xml is shown as below.

```
  <hibernate-mapping>
       <class name="com.gdou.bean.EntranceExam" table="ENTRANCE_EXAM"
schema="GDOU">
        <id name="id" type="java.lang.String">
           <column name="ID" length="50" />
           <generator class="uuid" />
           </id>
        <many-to-one name="peRecruitplan" class= "com. gdou. bean. PeRecruitplan "
fetch="select">
            <column name="FK_RECRUITPLAN_ID" length="50" />
           </many-to-one>
             ......
            <property name="password" type="java.lang.String">
              <column name="PASSWORD" length="50" />
           </property>
           <property name="maxNum" type="java.lang.Integer">
               <column name="MAX_NUM"    />
           </property>
       </class>
  </hibernate-mapping>
```

Through this mapping, the entity EntranceExam is associated with the EntranceExam table in the database. The data between Java classes and data table can be processed by Hibernate. Through the above methods, the system database independence is implemented and the underlying database system change and migration is more convenient.

PO class attribute

The PO attributes and the relevant database table field is corresponding and every attribute has a corresponding get set method. It should be avoided to declare the PO as a final class. A parameterless constructor must be explicitly defined if a constructor with parameters of PO class is defined.

```
public final class HibernateUtil {
    private HibernateUtil(){}
    private static SessionFactory sessionFactory;
    static{
     Configuration cfg=new Configuration();
     cfg.configure();
     sessionFactory=cfg.buildSessionFactory();
    }
    public static SessionFactory getSessionFactory()
    {
     return sessionFactory;
    }
    public static Session getSesstion(){
     return sessionFactory.openSession();
    }
}
```

The Hibernate session initialization process is a operation occupying resources seriously. The getSession method is used to implement initialization process of Hibernate session in order to avoid each action to access the database with a initializing online.

Hibernate cache application implementation

Hibernate second level cache covers the field of distributed cache and application level cache. It is Shared by the subordinate all Session of this SessionFactory instance. Querying data in Session, the first level cache is precedence one and the second level cache is queried only when the first level one is missed. The Hibernate itself does not provide a two-level cache implementation and only provides a simple cache based on HashTable. The Hibenlate provides the most mainstream third-party cache component interface. The second level cache is supported by Hibernate including: Ehcache, OSCache, HashTable, SwarmCache (distributed cache), JBossCache (step by step the cache). The Ehcache is used in this article.

```
          <cache name="AppCache"
             maxElementsInMemory="0"
             maxElementsOnDisk="0"
             eternal="true"
           overflowToDisk="false"
           diskSpoolBufferSizeMB="30"
           diskPersistent="false"
           diskExpiryThreadIntervalSeconds="120"
             memoryStoreEvictionPolicy="LFU">
         <terracotta concurrency="8"/>          </cache>
         <terracottaConfig url="127.0.0.1:9510"/>
    </ehcache>
    <property name="show_sql">true</property>
    <propertyname="hibernate.cache.provider_class">org.hibernate.cache.EhCache             Provider
</property>
    <property name="hibernate.cache.use_query_cache">true</property>
    AppEhCacheService service =    AppEhCacheServiceImpl.getEhCacheService();
    service.putToCache("questionMapForExam", mapList);
    Map<String,List<PaperQuestion>> questionMap = null;
    AppEhCacheService service = AppEhCacheServiceImpl.getEhCacheService();
    Object object = service.getFromCache("questionMapForExam");
    if(object != null){
    Map<String, Map<String,List<PaperQuestion>>> questionMapForExam = (Map<String,
Map<String,List<PaperQuestion>>>) object;
    if(questionMapForExam!=null&&questionMapForExam.size()>0){
    questionMap = questionMapForExam.get(user.getId()+ "+" + paperId);
    }
    }
```

Conclusion:

1. The system of data persistence layer is realized by using Hibernate.
2. the demand for online examination system is refined and the mapping relationship between the persistent classes and data sheet is set up.
3. the second level cache and the system load balance is carried out to increase the concurrency performance of the system.

Acknowledgments

This research was supported by 2014 field scientific research subject "learning attitude research ESL students under the environment of culture" Funds of Jiangxi Science & Technology Normal University (300098010626), which is gratefully acknowledged.

References：

[1] Zhang Yonghen. Design and Implementation of Paperless Examination System [M]. Xidian University. 2010

[2] Chen Shan. Design and Realization of the Web Based on J2EE Lightweight Architecture [M]. Shandong University. 2009

[3] Li Huoqin． Based on B／S structure of online examination system analysis and design[J]． Science &Technology Information，2008(30):149-149

[4] Wang Juzhu, Hou Tongpu, Shun Mingzhu． Permission Management System Design and Implementation Based on Struts-Hibernate Architecture[J]． Computer and Digital Engineering. 2011(4):101-105

[5] Jing Yong． The Human Resource Management System Based on Struts/Hibermate[M]． Jilin Univrsity，2014

[6] Chen Xinghuan． The Online Testing System Based on Struts Framework[J]. Software Development and Design. 2011(22)39-41