

# NoSQL 与关系数据库相结合的设计与实践

王艳<sup>1,2</sup>,董丽丽<sup>1</sup>

(1.西安建筑科技大学 信控学院,陕西 西安 710055;2.西安航空职业技术学院,陕西 西安 710089)

**摘要:** 为了实现对数据库的字段属性动态扩展方面的需求,提出一种将NoSQL和关系数据库相结合的架构模式设计方案。该设计方案主要以NoSQL良好的扩展性弥补关系数据库在属性横向扩展方面的缺陷,其架构方法主要采用在原有的关系数据库架构模式上增加一层辅助的NoSQL存储,能够完成对字段属性的动态扩展。实际应用表明,该设计模式具有易理解、易实现的特点,达到了设计要求。

**关键词:** 关系数据库;NoSQL;属性扩展

**中图分类号:** TP313 **文献标识码:** A **文章编号:** 1009-3044(2014)09-1857-03

## Design and Practice of the Combination of Nosql and Relational Database

WANG Yan<sup>1,2</sup>, DONG Li-li<sup>1</sup>

(1.Xi'an University of Architecture and Technology, Xi'an 710055,China;2.Xi'an Aerotechnical Polytechnic College School of Information And Control, Xi'an 710089,China)

**Abstract:** In order to realize the dynamic database of extended field properties of demand, a combination of NoSQL and relational database schema design. The design mainly extended NoSQL good for relational database properties in the horizontal expansion of its architecture, main method of using an additional layer of assisted NoSQL stored in relational database schema on the original, can complete the dynamic extension of the field properties. Practical application shows that, this design model has the features of easy to understand, easy to implement, has achieved the design requirements.

**Key words:** extended relational database; NoSQL; attribute

在现代软件工程中,随着社会信息化程度的不断提高及企业信息化进程的不断深入,人们对软件的要求越来越高,需求不断变化,数据存储也随之复杂多变。现在大多数数据存储都采用关系数据库来实现,但是由于关系数据库表结构需要提前定义,表结构一旦确定,数据表中的字段属性就固定不变,无法在程序运行过程中动态扩展,这就造成了关系数据库表字段属性无法横向扩展的问题。然而 NoSQL 数据库天生就具有良好的扩展性,如果将 NoSQL 数据库与关系数据库相结合,就可解决由于关系数据库本身在属性动态扩展方面的缺陷,实现字段属性的动态扩展问题。

## 1 关系数据库属性动态扩展设计

在软件开发实践过程中总结出来可以使用间接扩展的办法来实现,以电子商务软件系统的供求模块为例来说明。供求模块是一个公共表,表中存储的数据是所有的供求信息,其中供求信息又按照供求产品进行分类,例如:煤矿类、机电类、服装类等等,而不同产品类型的产品属性各不相同,例如:煤矿类产品的属性分为:低位发热量、高位发热量、全水分、干基水分、挥发份等等,而机电类产品的属性分为:设备名称、设备型号、设备品牌、设备产地等等,服装类产品的属性分为:尺码、颜色、品牌、材质等等。在后续程序升级过程中,还会有更多的产品分类加入供求,而且产品属性也相差各异。

为了能够动态添加供求商品的分类,并且能够对商品属性进行自定义。设计了如下的数据表结构:



图1 数据表结构

收稿日期:2014-01-11

作者简介:王艳(1981-),女,陕西大荔人,汉族,讲师,研究方向为信息处理、软件开发。

本栏目责任编辑:代 影

数据库与信息管

1857

当系统需要发布新的商品供求信息时,步骤如下:首先在供求类型表中加入新的供求分类,其次在商品属性表中加入此商品分类的属性,然后通过表单页面录入工期信息表,选择供求类型,填写供求商品属性相关的数据,最后保存数据到供求信息表和供求信息扩展表。

当需要查询某种类型的供求信息时,对以上四张数据表进行关联查询,得到某种供求类型的商品的所有数据集,此数据集为纵表形式,得到纵表数据集后,再通过SQL转换为横表,就得到了供求信息的数据记录。通过以上步骤,即可完成了供求信息表根据商品类型动态扩展属性的功能。通过这种纵表转横表的办法,在程序设计过程中,成功将煤炭供求信息的所有记录查询出来。而且实现了供求信息根据商品属性动态扩展。

## 2 关系数据库属性动态扩展的问题

在实际软件开发过程中,虽然关系数据库字段的属性无法在程序运行过程中通过物理的方式动态扩展,但是通过对表结构的拆分和组合,我们找到了对属性间接动态扩展的方法,并成功应用与软件项目实践。但通过软件项目投产运行后,经过跟踪观察以及对软件系统的监控,发现此方法虽然解决了属性动态扩展的问题,但又带来了数据查询的性能问题。纵表数据集的中供求信息记录数量与商品属性个数成正比关系,商品属性个数越多,纵表数据集中工期记录数量也会越多,例如某种商品有20个属性,而供求信息中此类商品有10万条记录,那么纵表数据集中对应的记录数就是:10万 $\times$ 20=200万条。也就是说纵表记录集数据量会随着商品属性的增加而成倍的增长。然而,供求信息查询的结果又直接来源于纵表记录集的转换,这样随着供求信息和商品种类的增多,系统供求模块的查询性能会急剧下降,即使通过索引、分表等方法进行优化,由于纵表记录集本身的数据基数庞大,系统性能仍然无法令客户方满意。

## 3 NoSQL与关系数据库结合设计实践

NoSQL数据库天生就具有良好的扩展性,由于关系数据库本身在属性动态扩展方面的缺陷,考虑将两者结合起来使用来弥补这方面的问题,以电子商务综合缴费平台项目为原型,架构思想以关系数据库与NoSQL数据库结合,总结如下:

以下关系数据库采用MySQL为例,关系数据库MySQL与NoSQL数据库在软件项目中结合使用,通常有如下两种方案:

### 3.1 NoSQL作为镜像

第一种(如图2所示):架构模式以原有的MySQL关系数据库作为存储方式不变,以NoSQL为辅助镜像,采用NoSQL的扩展优势提升关系数据库性能。

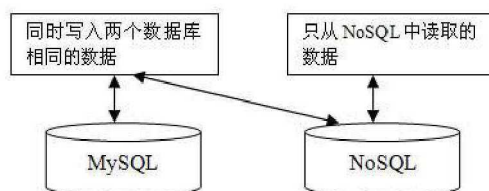


图2 NoSQL辅助镜像存储

此方案的程序如下:

```
//data 为要存储的数据对象
data.title="title";
data.name="name";
data.time="2013-12-11 12:09:23";
data.from="1";
id=DB.Insert(data);//数据写入MySQL中
NoSQL.Add(id,data);//以id为主键数据写入NoSQL数据库中
如果有数据一致性要求,实现方式如下:
bool status=false;
DB.startTransaction();//事务开始
id=DB.Insert(data);//数据写入MySQL中
if(id>0){
    status=NoSQL.Add(id,data);//以id为主键数据写入NoSQL数据库中
    if(id>0 && status==true){
        DB.commit();//事务提交
    }
    else
    {
        DB.rollback();//回滚
    }
}
```

这种架构模式将一层辅助的NoSQL存储增加到原有的基于MySQL数据库的架构上。其实现过程为将数据同时写入关系数据库MySQL和NoSQL数据库中去,使其两个数据库具有同样的镜像数据。当根据id主键查询时,采用NoSQL数据库完成查询,这样就免去了关系数据库的查询。这种架构模式不仅代码量不大,而且技术难度小,易于实现。

第二种(如图3所示):该设计方案是第一种方案的变体。第一种方案是采用同步写入数据,让两个数据库具有相同的镜像数据,在需要查询时,运用NoSQL数据库完成操作,实现过程以代码为主。而这种方案主要采用关系数据库MySQL将数据同步到NoSQL中去,是一种写入数据透明,但是具有极高的技术难度的方案。

此方案对以前比较老、复杂的系统较为适应。如果老系统采用第一种方案来修改代码程序实现扩展,那么会带来无法预测的新问题。并且该设计方案也适用于需要把数据同步到各种类型的存储中。

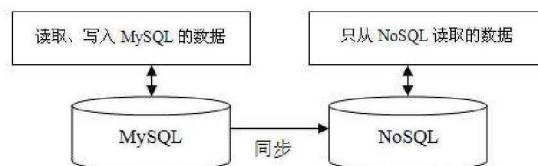


图3 NoSQL为镜像(同步模式)

### 3.2 MySQL和NoSQL相结合

该设计方案主要是将需要查询的小字段存储到MySQL中,将所有数据存储到NoSQL中。

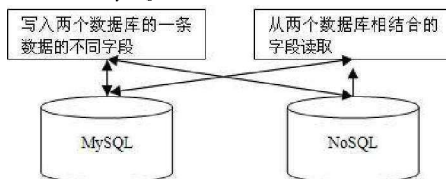


图4 两个数据库相结合

此设计方案程序如下:

//数据写入的示例伪代码,要存储的数据对象为data

```
data.title="title";
```

```
data.name="name";
```

```
data.time="2013-12-11 12:09:23";
```

```
data.from="1";
```

```
bool status=false;
```

```
DB.startTransaction();//事务开始
```

```
id=DB.Insert("INSERT INTO table (from) VALUES(data.from)");//数据写入MySQL中,只写from需要where查询的字段
```

```
if(id>0){
```

```
status=NoSQL.Add(id,data);} //以id为主键数据写入NoSQL数据库中
```

```
if(id>0 && status==true){
```

```
DB.commit();//事务提交
```

```
}else{
```

```
DB.rollback();//回滚
```

在MySQL中主要存储需要查询的字段(数字、时间等类型),根据查询建立对应的索引,而NoSQL中则存储其他不需要查询的字段(大文本属性)。因此在查询时,先把主键数据从关系数据库中查询出来,然后直接从NoSQL中取出相应的数据。

## 4 结论

综上所述,根据关系数据库属性动态扩展中的弱点,提出将NoSQL与关系数据库相结合的架构模式,并结合实际项目中的应用,设计和分析了NoSQL与关系数据库相结合的两架构模式的优缺点,从而实现属性的动态扩展问题。

## 参考文献:

[1] 韩君易.基于电子商务平台的No SQL解决方案应用研究[J].科技信息.2011(7).

[2] 沈姝.NoSQL数据库技术及其应用研究[D].南京信息工程大学.2012.

[3] 李冯筱,罗高松.NoSQL理论体系及应用[J].电信科学.2012(12).