**Java企业级应用课程报告**

2020-2021学年第1学期（CST31205）

1. **项目要求**

**1.1 内容**

利用Java EE技术设计并实现企业应用数据上传模块，具体内容包括：

（1）设计并实现数据上传页面；

（2）设计并实现上传数据的查看页面；

（3）实现excel表格数据的上传和存储。

**1.2 具体要求**

（1）在项目中使用Maven；

（2）在项目中使用Spring框架；

（3）上传数据的数据模式不固定，请设计一种支持可变数据格式的数据存储方案；

（4）上传数据量较大，可以达到几十MB，请设计一种对用户友好的上传方案，以避免用户操作时间过长；

（5）请在课程报告中详细描述数据的存储方案、上传数据的处理算法以及为了加快数据插入速度所使用的算法，分析算法的时间复杂度，给出执行时间与数据量的关系。

1. **项目背景介绍**

**2.1 国内外研究现状：**

**2.2 项目拟解决的问题：**

该项目解决的问题主要有以下几个方面：

1.首先解决数据存储格式，实现可变数据格式的数据存储

2.解决在web端的大数据上传费时长的问题

3.解决在文件上传时用户等待的页面响应事件

4.选择良好的数据库存储模式解决数据存储问题

5.解决数据存储数据库耗时过多的问题

6.加快查询数据的速度

7.设计系统上传以及查看界面

**2.3 现有的解决方案：**

针对以上问题，本项目设计了以下几个方案解决上述问题：

1. 依靠XML实现可变实体的数据库设计方案
2. 依靠非结构化数据库实现可变实体的数据库设计方案
3. 通过异步文件通道上传机制加快文件上传速度
4. 利用分片等算法对文件流进行优化加快存储速度
5. 依靠hbase建立分布式数据库增快数据访问的速度
6. 使用数据库连接池技术加快数据访问速度
7. 设计实现进度显示优化用户等待数据上传界面
8. 设计良好的系统上传以及查看界面，具有良好的交互性
9. **需求分析**

**3.1 数据处理需求：**

本项目需要实现数据的上传和查看功能

**3.2 数据存储：**

用户通过前端向服务器发起请求，请求上传数据，然后我们将用户提交的数据存储到数据库中，进行基于数据库的数据存储

**3.3 系统功能：**

本项目需要设计以下功能：数据上传，数据查看，数据搜索，数据删除，数据存储等功能

；

1. **系统设计**

**4.1 系统功能设计**

**4.1.1 首先是设计可变数据格式的数据库存储方案，本项目设计了两种可行的方案，分别是依靠XML实现的可变数据结构和依靠非结构化数据库实现方案**

1.依靠XML实现的可变数据结构：

基于XML结构化语言实现可变数据结构的存储主要是针对传统型数据库，在这里主要是依靠了XML结构语言本身的易变性，很好的处理了关系型数据库不方便存储可变实体的问题

具体实现方式，我们改变传统关系型数据库的存储模式，将传统型数据的存储的多重属性转变成XML表示，也就是说在这里将多种属性融合成XML，记录进入数据库，这样就可以实现可变实体

2.基于非结构化数据库实现方案：

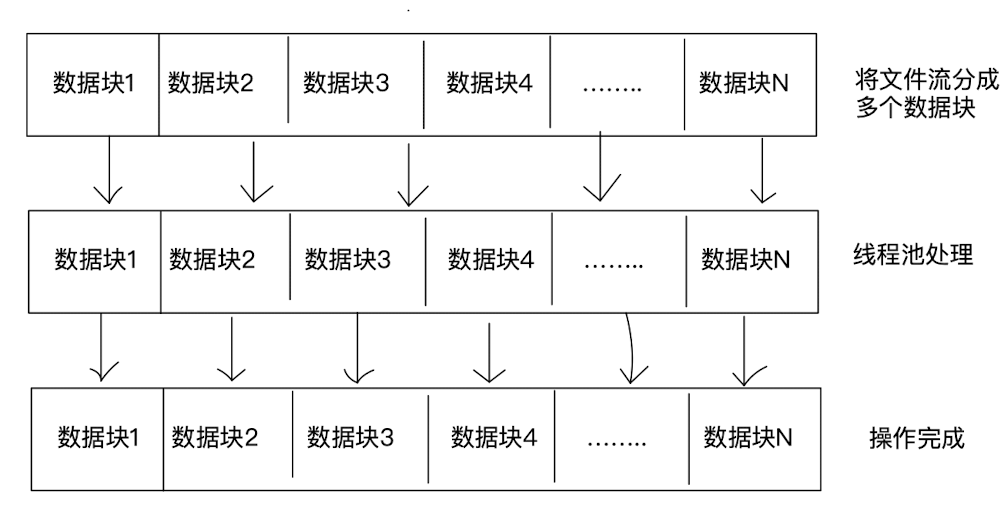
依靠XML实现方案可以在传统型数据库实现，在第二种方案中直接采取更换数据库类型实现可变数据格式的数据库存储，在这里我们依靠hbase进行设计，hbase是一种非结构化的数据库，可以巧妙的实现可变数据格式的数据存储，由于hbase还具有存储速度与访问速度较传统型数据库更快的特性，本项目将选择这种方案进行接下来的设计

**4.1.2 解决在web端的大数据上传费时长的问题：**

1.通过异步文件通道上传机制加快文件上传速度：

由于普通文件上传是采用的同步方式，所以速度较慢，本项目中采用的是利用IO的异步方式实现异步文件上传，利用线程池对异步文件上传进行处理，具体实现步骤：

首先进行文件数据流解析，然后将数据流进行分块，获得多个数据块，然后利用线程池技术对多个数据块进行处理，由于我们这里的IO是异步式的，所以每个数据块的操作是独立完成，且同时进行，这样做的话就会降低所需要的传输时间，提高文件上传速度，大概实现的过程图如下：



1. 利用切片算法优化上传速度

根据上述的异步文件上传算法，我们会进行切片方式将数据流进行切片形成数据块，这里我们主要讨论切片算法的选择和应用

在综合多种算法之后，本项目决定采取的是**基于位串内容感知的数据分块算法BCCA**

基于位串内容感知的数据分块算法BCCA：传统的分块方式是采取的字符比较，在这里我们所采用的方法就是将提取每个字符中的一位二进制进行比较，通过减少字符串匹配计算量的方式来使得性能更加优化，这样做的好处就是可以有效的减少CPU资源消耗，这样也可以减少数据指纹的消耗和减少数据的冗余程度，是一种较好的分块算法

实现：

1. 首先将数据子串中的数据提取出来，然后在基本单位中取最低位进行合并存储，得到一个身份识别信息
2. 在数据中进行身份识别信息与数据的匹配，利用子位串进行数据匹配，采取位串长度增加的方式进行匹配，最后依靠该方法实现数据的分块，减少数据的冗余

**4.1.3 解决在文件上传时用户界面的响应事件：**

在这里本项目利用**文档对象模型 DOM(DocumentObjectMode1）**去实现该功能，他是一种与浏览器平台以及语言无关的接口，利用它可以获得组件对象，在这里我们会从服务端获得上传的信息，经过计算，即可在进度条里进行显示，展现给用户，为用户提供良好的体验

**4.2 系统数据存储**

**4.2.1**依靠hbase建立分布式数据库增快数据访问的速度

Hbase是大数据常用的数据库之一，本项目在这里利用Hbase作为数据的存储方式，这样做的好处就是可以建立伪分布式的数据库存储系统，在单个服务器上上运行，如果有多个服务器的情况，我们可以直接利用其搭建一个分布式的架构，这样也可以提高数据上传以及响应的效率，这样做也可以和之前我们项目中采取的将数据块分片进行传输联系起来，在数据进行切片后，利用线程池将数据存入到不同服务器中的数据库中，这样的话就可以进一步加快速度

**4.2.2**使用数据库连接池技术加快数据访问速度

数据库连接池主要采用的是以空间换时间的思想，在用户与数据库建立连接的时候，不用重复建立连接，而是直接连接上后，放入连接池，在下次需要连接的时候直接从连接池中拿出来使用，这样可以缩减连接所花费的时间，也可以避免开销的浪费，且Hbase支持连接池的使用，故此方案也是可行的

实现方式：

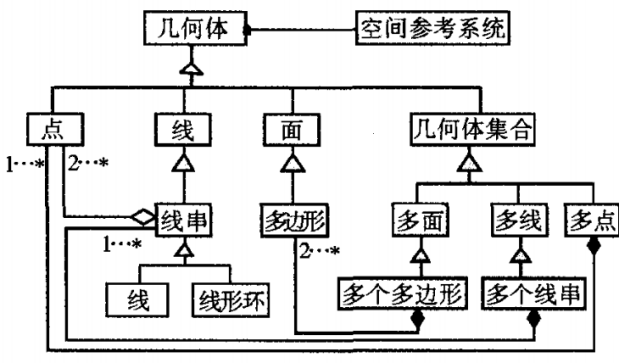
利用官方提供的hbasePool类来实现这部分的操作

**4.3 系统处理大数据方案**

**4.3.1** 加快查询数据的速度

本项目方案中使用的hbase数据库本身具有比传统型数据库更快的特点，且本方案利用了**基于HBase的矢量空间数据分布式存储**，这样也可以加快Hbase的索引速度，在用户进行访问数据的时候，可以更快的提供数据访问支持，同时我们还可以利用分布式数据库将索引扩展到多个计算机节点进行，也可以缩短访问时间

在这里我们构建索引的空间模型，利用OGC的简单要素模型（见下图），将存储和实体进行有效的连接，我们构建起矢量数据，然后存储矢量数据，在hbase中，我们在存储的时候将其属性设置为字符串，在解析使用的时候将其转化为我们所需要的类型，利用hbase本身的结构特性，利用矢量数据的不同层次建立起不同的表，最后构成矢量型数据库



OGC简单要素模型

在数据存储完成后，我们便可以利用hbase的特性，使用mapreduce算法对其索引进行优化，以索引ID为Key、以空间要素的ID为 Value写人本地HDFS，然后利用reduce函数将其进行合并，最后就得到了Hbase索引表，由于Hbase本身就是支持mapreduce算法的，在这里可以很好的应用，因为其本身支持分布式架构，我们将其运行在分布系统中，这样可以实现并行计算，大大缩短索引时间提高效率

具体的实现步骤：

1. 设计行键：hbase是依靠行键查询的，我们在这里将矢量空间数据的坐标作为行键，可以利用经纬度思想进行实现
2. 设计列族：根据其数据的几何属性的不同，我们会创建不同数量的Attribute，进行数据的存储，在这里我们使用WKT格式存储数据
3. 索引的并行构建：我们在这里采取的是网格构建数据方法，在索引表中用网格编码作为行键：

1）首先读取Hbase中的数据表，会返回数据的行键和自主编号

2）然后使用数据中的行键并截取字符串计算出经纬度，利用经 纬度我们就可以获得数据在其中的网格编码

3）以网格编码为key，数据行键为value，最后构成key-value

4）以网格索引为Key以“，”拼接原 Value构成新的Value，形成 reduceRDD

5）利用reduceRDD使用Hbase的convert变化，最后生成reduceResult，这样就可以得到索引表

**4.4 系统用户界面设计**

**4.4.1** 用户上传界面

界面：本项目将采用Jsp和Css共同构成前端界面，提供良好的用户体验，在该界面我们设计了如下功能：

1. 用户数据上传
2. 用户登录
3. 用户数据信息查询

页面预计成果图如下：

**4.4.1** 用户访问数据界面

在本界面我们为用户提供了如下功能：

1. 数据搜索
2. 数据分类查询
3. 数据下载

页面预计成果图如下：

1. **系统可行性分析：**

针对本方案的实现进行可行性的分析，分别从以下几个方面进行阐述

**5.1 技术支持**

1. Spring是否支持使用hbase数据库:SpringBoot本身支持hbase数据库，可以直接使用，并且可以构建分布式结构的数据库
2. Spring是否支持异步文件上传机制：异步文件上传方式可以通过线程操作进行，是可以在Spring框架中使用的
3. Hbase搭建：可以在单个服务器上建立伪分布式集群，可以建立数据库的分布式
4. Hbase是否支持线程池操作：根据Hbase官方文档是可行的，也支持数据库连接池

**5.2 技术之间可连接性**

本项目依靠Spring作为项目框架，在这里主要分析各个组件和算法是否能够一起使用，分析是否会存在冲突，首先是Spring本身能够很好的建立起hbase数据库还有使用线程进行多线程数据分块上传，在进行分块数据上传之后，直接使用分布式数据库进行分布式存储，再次出发进行数据的转化，转为成矢量数据，存储到hbase中，整个过程经过分析是可行的

**5.3 整体系统评价**

系统：本系统采用的是以Spring为框架，以hbase为数据库进行数据存储，以分块思想进行数据上传整体设计，以Jsp结合Css为项目的前端展示界面，其中分块思想体现在建立分布式数据库，使用多线程，使用异步通道文件上传，使用数据库连接池，在数据查找的时候使用已经建立好的分布式数据库创造索引表，进行分布式查找，利用以上的分布式搭建，从逻辑上可以大大减少数据上传，数据查找的时间

用户体验：本项目采用的利用DOM文档模型可以有效的实现文件上传进度条显示，也为用户提供了查询数据的功能，增加用户的体验

1. **总结**

项目总结：

1. **参考文献**

[1]范建永,龙明,熊伟.基于HBase的矢量空间数据分布式存储研究[J].地理与地理信息科学,2012,28(5):39-42.

[2]周斌,朱容波,张莹.基于位串内容感知的数据分块算法[J].计算机工程与科学,2016,38(10):1967-1973.

[3]张文辉.基于OGC标准的空间数据共享技术研究[J].数字通信世界,2018(12):66-66.

[4]白鹤,吕红亮,王劲林.进度显示的大文件上传组件的设计与实现[J].计算机工程与应用,2009,45(5):91-94.

[5]何标.文档对象模型(DOM)及其应用[J].现代商贸工业,2009,21(11):291-293.

[6]朱庆生,葛亮.一种可变数据结构存储处理模型[J].计算机应用,2004,24(8):69-71.

[7]陈洁,褚龙现,夏栋梁.一种支持并行处理的矢量数据存储与查询方法[J].电子设计工程,2017,25(10):31-33.

[8]刘翠霞.Java数据库连接池的原理与应用[J].无线互联科技,2020,17(4):167-168.

[9]陈冈,夏火松.基于异步文件通道的Java Web多任务分块文件上传[J].河南理工大学学报：自然科学版,2015,34(3):400-405.

[10]陈涛,黄艳峰.Java Web开发中文件上传方法研究与实现[J].电脑知识与技术,2016,12(11):48-49.