软件系统分析与设计

PaperIsAllYouNeed学术成果分享平台

数据设计报告

|  |  |  |  |
| --- | --- | --- | --- |
| 项目组成员信息 | | | |
| 小组编号 | 17 | | |
| 小组名称 | 请填写团队名称 | | |
| 小组联系人 | 18351015 张津赫 | | |
| 学号 | 姓名 | 联系方式 | 本次实践中主要承担的工作内容 |
| 19373239 | 王子涵 | 18519208373 | 数据库设计 |
| 18351040 | 高野淇 | 15141810175 | 数据库设计 |
| 18351015 | 张津赫 | 17812032698 | Elasticsearch介绍 |
| 19231179 | 申玉啸 | 18810955230 | 主从库实现 |
| 19373802 | 刘志一 | 13986777478 | 数据库设计 |
| 19231251 | 奚望 | 13817322661 | 主从库实现 |
| 18351021 | 冯宇扬 |  | 数据库灾难 |
| 18351008 | 高嘉兴 | 15210224408 | Elasticsearch介绍 |
| 19373308 | 邹文祥 | 18811797692 | 数据库灾难 |

2021

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 初稿 | 11.25 | 高野淇 | 张津赫 | 初步建模，简单总结 |
| 终稿 | 11.26 | 张津赫 | 张津赫 | 规范格式 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[概述 4](#_Toc57382449)

[一、Elasticsearch 4](#_Toc57382450)

[1.1 简介 4](#_Toc57382451)

[1.2 应用 4](#_Toc57382452)

[二、主库从库具体实现 5](#_Toc57382453)

[2.1选用理由 5](#_Toc57382454)

[2.2运行原理 5](#_Toc57382455)

[2.3.具体操作 5](#_Toc57382456)

[三. 数据库内部设计 6](#_Toc57382457)

[3.1 数据库表设计 6](#_Toc57382458)

[3.2 数据库规模估计 9](#_Toc57382459)

[3.3 数据库备份容灾 9](#_Toc57382460)

## 

# 概述

在设计数据库时，考虑到本数据库的大数据量对检索的高要求，我们使用Elasticsearch以快速和方便的检索大量数据；考虑到高并发和读写频率高的问题，我们使用Mysql主从库分库以实现读写库分离；最后，对数据库各表进行相应设计。

# 一、Elasticsearch

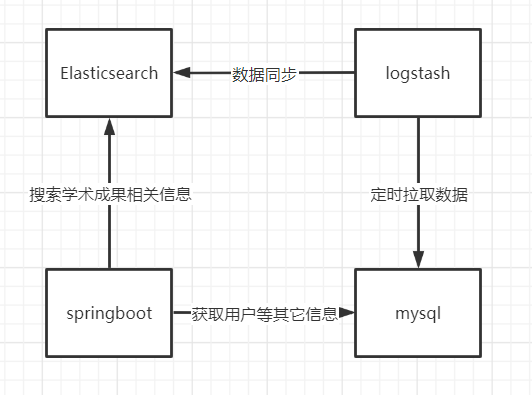
## 1.1 简介

Elasticsearch 是一个分布式、高扩展、高实时的搜索与数据分析引擎。它能很方便的使大量数据具有搜索、分析和探索的能力。充分利用Elasticsearch的水平伸缩性，能使数据在生产环境变得更有价值。

## 1.2 应用

Elasticsearch具有优秀的搜索性能，因此该系统采取mysql存储数据，ES搜索数据的部署方式，ES通过中间组件logstash定时同步mysql数据，对获取到的数据建立索引，springboot链接ES进行快速搜索。

同时可利用elasticsearch-head和kibana配合ES



**图2-1 ES示意图**

二、主库从库具体实现

2.1选用理由

由于单台MySQL数据库服务器不能满足数据备份和读写频率高的要求，我们可以使用将数据库分为主库和从库来解决这一问题。MYSQL数据库本身所带的主从复制技术比较成熟，配置也不复杂，我们决定在项目中采用它。之后在相应的进行读写分离即可。

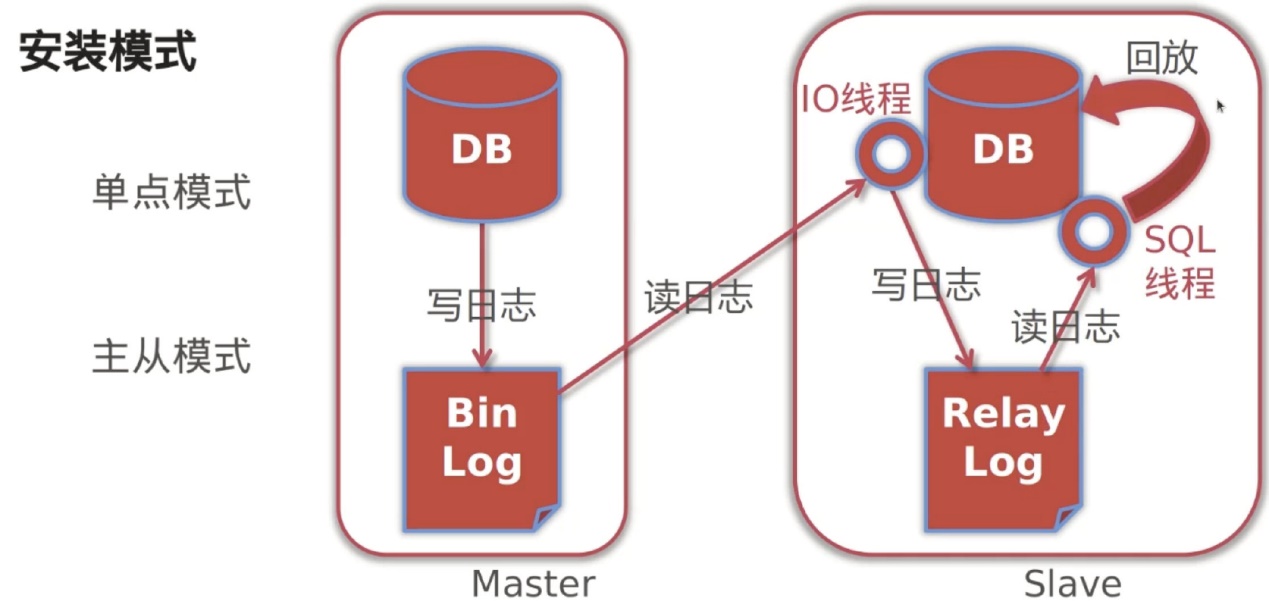
2.2运行原理

主从数据库的大致运行流程如下：

1. master会将变动记录到二进制文件(BinLog日志)中；

2. master有一个I/O线程将BinLog日志发送到slave；

3. slave有一个I/O线程把接受到的BinLog日志写到relay日志里面；

4. slave有一个sql线程，按照relay日志处理slave的数据。

**图2-1运行原理**

2.3.具体操作

2.3.1 主从同步注意点

我们在设置主从同步时必须要注意一下四点，这也是我们主从数据库设置的关键

（1）准备好两个版本一致(最好系统环境也一样)的mysql服务器。

（2）主库服务器和从库服务器数据一致，即保证在数据初始化时是一样的

（3）主库开启二进制日志（设置log-bin参数）

（4）主从server-id不同

2.3.2 主库从库设置

首先将数据库分配在两个不同的服务器上

1.主库（master）设置

打开MySQL的配置文件，新增配置，例如开启二进制日志功能(log-bin=mysql-bin)等，设置server\_id并重启mysql服务；

创建用于同步的用户账号；

最后使用show master status命令，查看主数据库二进制文件名及位置，用于从库设置。

2. 从库（slave）设置

验证同步账号能否登录；

打开MySQL的配置文件，新增配置，注意server\_id不能和主库一样，重启mysql服务；

设置对应的主数据库配置，包括主数据库地址，主数据库同步用户及相应密码，主数据库端口，主数据库二进制文件及其位置；

输入 START SLAVE 开始同步；

最后输入SHOW SLAVE STATUS查看同步是否成功

2.3.3 读库写库操作

考虑到后端框架为Spring boot + Spring Data JPA，我们需要自定义注入一个AbstractRoutingDataSource，将JPA的操作与对应的库相连，并使用ThreadLoacl使数据源与线程绑定即可实现相应的读写库绑定。

三. 数据库内部设计

## 3.1 数据库表设计

用于记录用户的User表

User表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | userID | 用户ID | varchar(100) | 否 | 主键 |
| 2 | userName | 用户名 | varchar(100) | 否 |  |
| 3 | passWord | 用户密码 | varchar(100) | 否 |  |
| 4 | email | 邮箱 | varchar(100) | 是 |  |
| 5 | realName | 真实姓名 | varchar(100) | 否 |  |
| 6 | email | 所属单位(研究机构或者学校) | varchar(100) | 是 |  |
| 7 | resume | 个人简介 | varchar(100) | 否 |  |

用于记录文献的Paper表

Document表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | paperId | 文献ID | varchar(255) | 否 | 主键 |
| 2 | title | 文献名 | varchar(100) | 否 |  |
| 3 | experts | 作者列表 | varchar(100) | 否 |  |
| 4 | keywords | 关键词 | varchar(100) | 否 |  |
| 5 | publishedYear | 发布时间 | Int | 是 |  |
| 6 | pdfUrl | 文章链接 | varchar(255) | 否 |  |
| 7 | abstraction | 摘要 | varchar(255) | 是 |  |
| 8 | views | 浏览次数 | Int | 否 |  |
| 9 | type | 文章所属分区 | varchar(255) | 否 |  |

用于记录成果分享的Share表

Share表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | shareId | ID | Varchar(100) | 否 | 主键 |
| 2 | sharerUserId | 分享用户ID | Varchar(100) | 否 | 外键 |
| 3 | shareContent | 分享内容 | Varchar(255) | 否 |  |
| 4 | shareCreateTime | 分享时间 | varchar(255) | 否 |  |
| 5 | supportCnt | 点赞次数 | Int | 否 |  |
| 6 | unSupportCnt | 点踩次数 | Int | 否 |  |
| 7 | commentCnt | 评论数 | Int | 否 |  |

用于记录成果分享评论的commentShare表

commentShare表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | commentShareId | ID | Varchar(100) | 否 | 主键 |
| 2 | shareId | 分享ID | Varchar(100) | 否 | 外键 |
| 3 | commentShareCreateTime | 评论时间 | Date | 否 |  |
| 4 | commentContent | 评论内容 | varchar(255) | 否 |  |

用于记录关注关系的Follow表

Follow表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | followId | ID | Varchar(100) | 否 | 主键 |
| 2 | userId | 用户ID | Varchar(100) | 否 | 外键 |
| 3 | followerUserid | 关注者ID | Varchar(255) | 否 | 外键 |

用于记录申请的Application表

Application表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | formID | 帖子ID | Int | 否 | 主键 |
| 2 | userID | 作者ID | Int | 否 | 外键 |
| 3 | flag | 认领类型 | Int | 否 |  |
| 4 | objectID | 认领的门户或作者ID | Varchar(255) | 否 | 外键 |
| 5 | time | 时间 | Date | 否 |  |
| 6 | result | 结果 | Varchar(255) | 是 |  |

用于记录评论消息的msg表

msg表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | msgsendid | 消息记录id | varchar(100) | 否 | 主键 |
| 2 | msgsendfrom | 发消息者id | varchar(100) | 否 | 外键(userid) |
| 3 | msgsendto | 接收消息者id | varchar(100) | 否 | 外键(userid) |
| 4 | msgsendcreatetime | 消息创建时间 | Varchar(100) | 否 |  |
| 5 | msgcontent | 消息内容 | text | 否 |  |
| 6 | Msgboxid | 对应消息盒子id | Varchar(100) | 否 | 外键(msgboxid) |

msgbox表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | msgboxid | 消息盒子id | varchar(100) | 否 | 主键 |
| 2 | msgboxfrom | 发消息者id | varchar(100) | 否 | 外键(userid) |
| 3 | msgboxto | 接收消息者id | varchar(100) | 否 | 外键(userid) |
| 4 | msgboxread | 消息是否已读 | boolean | 否 |  |

用于记录用户发表文献的表

userpaper表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | userpaperid | 记录id | varchar(100) | 否 | 主键 |
| 2 | userid | 用户id | varchar(100) | 否 | 外键(userid) |
| 3 | papaerid | 文献id | varchar(100) | 否 | 外键(paperid) |
| 4 | createtime | 发表时间 | varchar(100) | 否 |  |

记录对于论文评论的表

Commentpeper

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | commentpaperid | 论文评论id | varchar(100) | 否 | 主键 |
| 2 | commentpaperpaperid | 论文id | varchar(100) | 否 | 外键(paperid) |
| 3 | commentpaperfrom | 评论用户id | varchar(100) | 否 | 外键(userid) |
| 4 | commentpaperto | 发表时间 | varchar(100) | 否 |  |

记录评论更新的表

Commentnoti

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 是否允许为空 | 主键/外键 |
| 1 | sharenotiid | 消息通知id | varchar(100) | 否 | 主键 |
| 2 | sharenotito | 用户id | varchar(100) | 否 | 外键(userid) |
| 3 | sharenoticontent | 消息评论内容 | varchar(100) | 否 |  |
| 4 | sharenoticreatetime | 消息发布时间 | varchar(100) | 否 |  |

## 3.2 数据库规模估计

以百度学术宣传为准，百度学术显示，百度学术有120万国内外学术站点，400万个国内学者主页，有6亿篇中外文学术文献，12亿文献全文链接，300个学科研究方向，1万中文学术期刊，300万科研主题词。（数据来源：百度学术首页简介）

本项目预计承载专家400万人，以人均10个科研成果计算，共有4000万成果，则科研成果表，关联成果表将有4000万条表项，对表搜索能力造成极大的影响。

据此假设，论文表也将有4000万条表项，大小高达90GB。预计数据库总大小最大为120GB。

数据库中，单条最大的论文，单条仅为2.5k，且所有表均不涉及联合查询，所以在表条目较多时仍可以保证比较高的访问速度，依此情况，限制单表500万条表项，大小最大12GB

## 3.3 数据库备份容灾

本系统数据资源庞大且十分重要，为保证数据安全，对数据库进行备份容灾，采用腾讯云备份DBS服务器，数据库预计总大小200GB。

采用基础版MySQL8，1核1GB，物理备份，时长2月。