编码：项目级-系统架构设计-Version\_1.0.0-release-20200815

文档等级：□发布级 ■项目级 □团队级

采纳标准：团队文档质量标准

网络文学大数据分析系统

**Big Data Analysis System of Network Literature**

系统架构设计

**Version 1.0.0-release**

编写人：张婷睿

西工大-华迪实训第四小组

**All Rights Reserved**

**目 录**

**[1 引言 3](#_Toc32379)**

[1.1 标识 3](#_Toc2322)

[1.2 系统概述 3](#_Toc7485)

[1.3 文档概述 3](#_Toc21337)

[1.4 基线 4](#_Toc28126)

**[2 引用文件 4](#_Toc27063)**

**[3 总体设计 4](#_Toc20002)**

[3.1 需求规定 4](#_Toc10864)

[3.2 运行环境 5](#_Toc1159)

[3.2.1 3.2.1硬件环境 5](#_Toc17963)

[3.2.2 3.2.2软件环境 5](#_Toc31414)

[3.3 设计原则与思想 5](#_Toc29986)

[1.1.1 设计原则 5](#_Toc28775)

[1.1.2 设计思想 6](#_Toc26349)

[3.4 系统设计方案 6](#_Toc4749)

[1.1.3 系统设计架构 6](#_Toc18821)

[1.1.4 处理流程 8](#_Toc199)

**[4 数据架构设计 8](#_Toc26990)**

[4.1 数据结构设计 8](#_Toc23397)

[4.2 数据存储设计 9](#_Toc15404)

**[5 运行设计 10](#_Toc1665)**

[5.1 运行模块组合 10](#_Toc28531)

[5.2 运行控制 11](#_Toc29439)

[5.3 运行时间 11](#_Toc10921)

**[6 系统出错处理设计 11](#_Toc1136)**

**[7 系统维护设计 11](#_Toc23994)**

# 引言

## 标识

适用系统：网络文学大数据分析系统

中文名称：系统架构设计（SAD）

英文名称：System Architecture Design (SAD)

文档版本：1.0.0

文档编号：项目级-系统架构设计-Version\_1.0.0-release-20200815

## 系统概述

本文档适用于“网络文学大数据分析系统”项目的开发过程。网络文学大数据分析系统项目由西北工业大学软件学院提出，由西工大-华迪实训第四小组成员负责实施，该项目编号为项目级-系统架构设计-Version\_1.0.0-release-20200815，其软件产品版本号为“1.0”。

该项目基于B/S（Brower/Server）浏览器/服务器结构，采用传统的三层结构方式进行解耦开发。对当下最热门的网络文学阅读机构——“晋江文学”提供的各项数据进行可视化展示和分析。数据分析采用Hadoop和Spark。

## 文档概述

本文档依据《GB/T 8567-2006 计算机软件文档编制规范》制定，属于技术文档，仅限于项目相关人员阅读。

本文档的编写目的是描述系统的架构设计方案，包括系统的软件总体架构设计及使用的框架说明，以及基于该架构的开发流程，并作为指导开发人员、测试人员进行系统开发及测试的依据。

本文档的保密范围如下：

①开发人员

②测试阶段人员

③对本文档进行评审的人员或机构

④项目组以及其他有权需要调用本文档的人员

## 基线

基线：GB/T 8567-2006

# 引用文件

《GB/T 8567-2006 计算机软件文档编制规范》，国家标准

《项目级-文档编写格式规范-Version\_1.0.0-release-20200812》，工作室编写

《项目级-软件需求规格说明书-Version\_2.0.0-release-20200813》，工作室编写

# 总体设计

## 需求规定

网络文学大数据分析挖掘系统主要实现三方面功能：一是对采集到的数据的清洗、存储；二是将存储在数据库中的数据进行数据分析挖掘，并存储到另一个数据库中；三是利用前端对数据进行可视化展示。系统应该具备以下功能：

数据存储：对原始数据和静态数据进行清洗和处理后，存入数据库。

数据分析：能够根据存储的数据进行排名分析和积分分析，实现月、季、年排行的可视化、了解受欢迎的题材和作者以及热门小说。

图表展示：将数据分析以后的结果以图表的形式实时显示在界面上。

## 运行环境

### 3.2.1硬件环境

(1)CPU：Intel CoreI5 1.8GHz 及以上

(2)内存：8G 及以上

(3)硬盘：60G 及以上

### 3.2.2软件环境

(1)服务器： Tomcat 8.0

(2)操作系统：Ubuntu 16.04 LTS 或 Ubuntu 18.04 LTS

(3)数据库：MySQL 6.5

(4)基本配置：Spark 2.2.0（Built for Hadoop 2.8.1）伪分布式环境、JDK 1.8、Scala 2.12.2

(5)开发工具：Intellij IDEA 2017.3.5 及以上

(6)PC 端：Chrome21.0及以上等的浏览器

## 设计原则与思想

### 设计原则

网络文学大数据分析系统遵循以下原则：

（1）快速性：在10秒内完成数据抓取、数据预处理和数据统计分析；

（2）准确性：每一次的数据分析能正确的得到作者排名、题材排行、订阅偏好分布等；

（3）易控性：系统简单易操作；

（4）简洁性：用户操作界面简单易用，信息展现准确明了；

（5）全面性：信息展示全面。

### 设计思想

由于MySQL等传统的数据库的单表在几千万行的时候有很严重的性能瓶颈，我们使用分布式的数据库HBase存储经过数据预处理的数据。分布式数据库HBase比传统的关系型数据库有更快的响应速度，这能够保证数据抓取的耗时降低20%以上，也保证了数据的准确性，使得出错率降低到了1%以下。

为了加快数据处理，我们使用spark框架在内存中运行事务，能够保证在短时间内完成数据处理。同时，我们使用spark的并发处理技术提高数据的处理速度。

本系统除了实时分析以外，还支持数据预测。通过存储每一次的分析结果，能够快速的响应数据预测和数据对比。

为了使不同分辨率和不同浏览器的客户端拥有一致的显示效果和用户体验。本系统采用了响应式布局和boostrap框架，从而可以为不同终端的用户提供更加舒适的界面和更好的用户体验。

## 系统设计方案

### 系统设计架构

网络文学大数据分析系统主要采用的技术手段有：Spark、HBase、MySQL、Spring Boot + Spring MVC框架。

选用HBase，HBase是运行在Hadoop上的NoSQL数据库，它是一个分布式的且可横向扩展的大数据仓库。HBase能够利用HDFS的分布式处理模式，在一组商业硬件上轻松的存储许多具有数十亿行和上百万列的数据表。除去Hadoop的优势，HBase本身就是十分强大的数据库，它能够融合key/value（键值对）存储模式带来快速查询的能力，以及通过Spark进行快速的数据读写。

使用Spark对HBase进行数据批处理很容易让HBase过载，比如：内存溢出导致RegionServer挂掉（一旦RegionServer挂掉，会有一段时间的读写不可用），而且经过统计分析的数据是结构化的数据，其量级也是MySQL数据库所能接收的，所以使用MySQL数据库存储Spark程序批处理的实时结果。另外一点，MySQL广泛用于前端数据库，具有良好的快速响应能力，对于前端页面的数据渲染速度有很大的帮助和提升。

最初的数据处理以Hive SQL为主，底层计算引擎为MapReduce，部分相对复杂的业务会由MapReduce程序实现。随着业务的发展，单纯的Hive SQL查询或者MapReduce程序已经越来越难以满足数据处理和分析的需求。一方面，MapReduce计算模型对多轮迭代的DAG作业支持不给力，每轮迭代都需要将数据落盘，极大地影响了作业执行效率，另外只提供Map和Reduce这两种计算因子，使得用户在实现迭代式计算（比如：机器学习算法）时成本高且效率低。另一方面，在数据仓库的按天生产中，由于某些原始日志是半结构化或非结构化数据，因此，对其进行清洗和转换操作时，需要结合SQL查询以及复杂的过程式逻辑处理，这部分工作之前是由Hive SQL结合Python脚本来完成的。这种方式存在效率问题，当数据量比较大的时候，流程的运行时间较长，会直接影响到一系列数据下游的完成时间以及各种重要数据报表的生成。

选用 Spark 主要实现和编程接口基于 Scala，Spark 的核心思路就是将数据集缓存在内存中加快读取速度，同时用 lineage 关联的 RDD 以较小的性能代价保证数据的鲁棒性。RDD 是弹性分布式数据也是 spark 的核心，完全弹性的，如果数据丢失一部分还可以重建。RDD 有自动容错、位置感知调度和可伸缩性，通过数据检查点和记录数据更新对象容错性。通过 SparkContext.textFile()加载文件变成 RDD，然后通过 transformation 构建新的 RDD，通过 action 将RDD 存储到外部系统。适用领域正如其目标作用域，Spark 适用于需要多次操作特定数据集的应用场合。需要反复操作的次数越多，所需读取的数据量越大，受益越大；同时，Spark的DataFrame和SparkSQL非常适合对数据库中的数据进行聚合、统计分析，也兼顾了Spark的并行性，极大的提高了数据处理的速度。

Spring由于其繁琐的配置，一度被人认为“配置地狱”，各种XML、Annotation配置，让人眼花缭乱，而且如果出错了也很难找出原因。Spring Boot更多的是采用Java Config的方式，对Spring进行配置。开发人员通过Spring Boot能够在几秒构建一个项目方便对外输出各种形式的服务，如 REST API、WebSocket、Web、Streaming、Tasks，也具备非常简洁的安全策略集成，同时支持关系数据库和非关系数据库，未来可以方便的升级数据库。Spring Boot支持运行期内嵌容器，如 Tomcat、Jetty。Spring Boot还支持热启动，自动管理依赖。综上所述，Spring Boot非常适合快速搭建应用程序，让我们减少配置的时间，开发更便捷，把更多的精力放在实现业务需求上。在Spring Boot的基础上，我们使用Spring MVC框架开发Java Web应用，用Mybatis管理持久层。Spring Boot也便于继承各种模块，具有良好的可拓展性。

### 处理流程

（1）数据预处理：数据清洗、数据集成、数据变化和数据规约等一系列操作；

（2）数据结构化：将数据存入数据库；

（3）数据可视化：将数据库中的数据以图表的形式呈现给用户；

# 数据架构设计

## 数据结构设计

|  |  |
| --- | --- |
| 数据结构名 | 属 性 |
| 书籍信息 | 书籍id、书籍名称、书籍题材、书籍积分、书籍起始更新时间、书籍最后更新时间、状态（完结/连载）、书籍打赏数额、一个月内更新天数、作者id |
| 作者信息 | 作者id、作者名称、作者积分、作者入站时间 |
| 读者打赏信息 | 读者名称、书籍id、读者打赏数额 |

## 数据存储设计

1. Book（MySQL数据库内的书籍信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 | 备注 |
| Book\_id | Long | 8 | 书籍id | 主键 |
| Book\_name | Varchar | 50 | 书籍名称 |  |
| Book\_type | Varchar | 20 | 书籍题材 |  |
| Book\_credit | double | 15 | 书籍积分 |  |
| Book\_begin\_update | date | 20 | 书籍起始更新时间 |  |
| Book\_final\_update | date | 20 | 书籍最后更新时间 |  |
| Book\_state | Boolean | 4 | 状态（完结/连载）  完结状态为0，连载状态为1 |  |
| Book\_total\_reward | double | 15 | 书籍打赏数额 |  |
| Book\_month\_update | Int | 4 | 一个月内更新天数 |  |
| Author\_id | Long | 8 | 作者id | 外键 |

1. Author（MySQL数据库内的作者信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 | 备注 |
| Author\_id | Long | 8 | 作者id | 主键 |
| Author\_name | Varchar | 50 | 作者名称 |  |
| Author\_credit | double | 15 | 作者积分 |  |
| Author\_join\_time | date | 20 | 作者入站时间 |  |

1. Reader（MySQL数据库内的读者打赏信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 | 备注 |
| Reader\_name | Varchar | 50 | 读者名称 | 与Book\_id一起作为复合主键 |
| Book\_id | Long | 8 | 书籍id | 外键 |
| Reader\_total\_reward | double | 15 | 读者打赏数额 |  |

# 运行设计

## 运行模块组合

客户机程序启动后，开始运行数据预处理模块，调用静态数据，对数据进行清洗和格式化，并存入HBase数据库。当数据库获得充足的数据时，启动数据处理模块，分析挖掘经过清洗的原始数据，并将统计后的数据存入MySQL数据库。 Java Web应用将会使用前端轮询，查询数据库内的数据，并将数据库中的数据抓取到前端页面，以可视化的形式展示出来。

这样就完成了数据的读取，输入，格式化，接收，传送，处理等一系列操作。

## 运行控制

运行控制将严格按照各模块间函数调用关系来实现。数据处理方面，当数据库存入了足量的数据，数据才会开始进行处理。数据可视化方面，当数据处理程序更新了数据库之后，前端页面将会访问数据库，将返回信息送回前端，并展示数据内容。

## 运行时间

为了满足客户对本系统运行时间的要求，简化sql语句，以尽量减少数据库的访问时间。其次硬件的性能对系统整体的运行速度的影响也比较大。我们将采用较好的硬件设施，确保实时数据的处理时间控制在5s以内。

# 系统出错处理设计

# 系统维护设计

(1)对于数据库维护，本系统主要由管理员对数据库基本结构进行管理维护。

(2) 对于软件维护，本系统采用的是模块化的设计方法，各模块之间相互独立性

较高，这对后期操作人员维护系统带来了很大的方便。如需修改某个单独功

能，只需修改相关页面即可;如需添加某功能，只需再添加页面选项的内容

即可。

（3）关于模块中的代码更新于修改，由于模块使用了多种语言进行编写，因此在维护时需要对不同语言熟悉的人员相互配合，共同完成。