**大数据系统开发实践项目---技术文档**



**作 者： 黄昱欣（1120200807） 郑子帆（1120200822）**

**王天泽（1120200718） 张 驰（1120200837）**

**安禹堃（1120200860）**

**组 长： 郑子帆（1120200822）**

**所在学院： 计算机学院 计算机科学与技术专业 07112002**

**教 师： 郭贵锁**

2022年9月21日

目录

[1 搜索引擎实现的技术方案 1](#_Toc115453074)

[1.1功能描述 1](#_Toc115453075)

[1.1.1信息采集模块 1](#_Toc115453076)

[1.1.2信息处理模块 3](#_Toc115453077)

[1.1.2.1网页内容分析 3](#_Toc115453078)

[1.1.2.2构建数据索引 3](#_Toc115453079)

[1.1.3查询模块 3](#_Toc115453080)

[1.1.3.1响应用户检索 4](#_Toc115453081)

[1.1.3.2检索结果处理 4](#_Toc115453082)

[1.2技术选型 5](#_Toc115453083)

[1.2.1优势 6](#_Toc115453084)

[1.2.2不足 7](#_Toc115453085)

[1.3功能实现 9](#_Toc115453086)

[1.3.1分布式爬行子系统 9](#_Toc115453087)

[1.3.2分布式索引子系统 10](#_Toc115453088)

[1.3.3分布式查询子系统 11](#_Toc115453089)

[1.4工作计划 12](#_Toc115453090)

[1.5组织结构 13](#_Toc115453091)

[1.5.1人员构成 13](#_Toc115453092)

[1.5.2分工 13](#_Toc115453093)

[1.6软件质量保证、非功能性保证 14](#_Toc115453094)

[1.6.1产品特性 14](#_Toc115453095)

[1.6.1.1功能性要素 14](#_Toc115453096)

[1.6.1.2正确性 14](#_Toc115453097)

[1.6.1.3可维护性 15](#_Toc115453098)

[1.6.1.4可重用性 15](#_Toc115453099)

[1.6.2过程特性 16](#_Toc115453100)

[1.6.2.1理解过程化程度 16](#_Toc115453101)

[1.6.2.2实施过程化程度 16](#_Toc115453102)

[1.6.3用户满意度特性 16](#_Toc115453103)

[参考文献 18](#_Toc115453104)

# 搜索引擎实现的技术方案

近年来，随着信息技术科技的进步，人们的生活方式发生了巨大的改变，强大的数据信息化的世界正在逐渐包裹着每一个生活在当下的人们。全世界每年产生1EB到2EB信息，相当于地球上每个人每年大概产生250MB信息。在人们的日常生活和工作中，信息的获取途径逐渐被网络所取代，而使用者想要快速地对海量信息进行搜索就需要应用一定的媒介，即搜索引擎。用户仅需提交检索关键词等信息即可进行检索，它能很好地解决用户对需要的数据资源进行快速且准确的检索，减小开发成本，提升信息服务的效益，提高信息资源的利用效率。[1]

## 1.1功能描述

网络搜索引擎根据初始配置，自动从网络上抓取信息并处理，将提取到的信息结构化，然后建立索引，通过服务器接受用户请求，在索引库中找到相关结果并进行排序，最终通过页面的形式呈现给用户。在本文中重点关注全文搜索引擎，即通过从互联网上提取的各个网站的信息而建立的数据库中，检索与用户查询条件匹配的相关记录，然后按一定的排列顺序将结果返回给用户。

我们将搜索引擎的功能展示为下图，其功能主要分为三部分：信息采集模块、信息处理模块和查询模块。

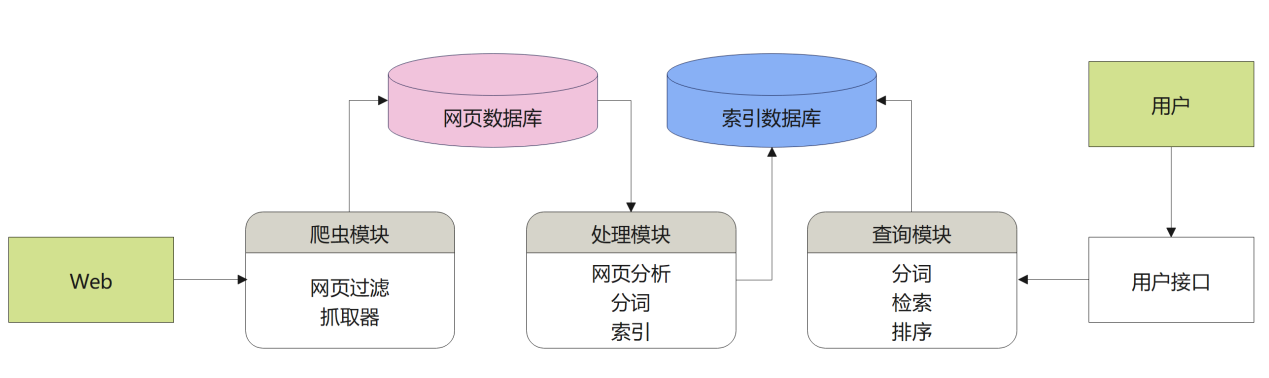


图1搜索引擎结构图

### 1.1.1信息采集模块

由于 Web 信息的大容量、分布性和动态性，目前搜索引擎的对象都是数以千计的网页，因此搜索引擎面临的第一个问题就是如何将这些网页存储下来，将信息保存，便于后续构建索引，也就是信息采集。搜索器是一个机器人程序Robot ，也称为 Spider crawler，自动地在互联网中搜集信息下载到本地文档库。[2]

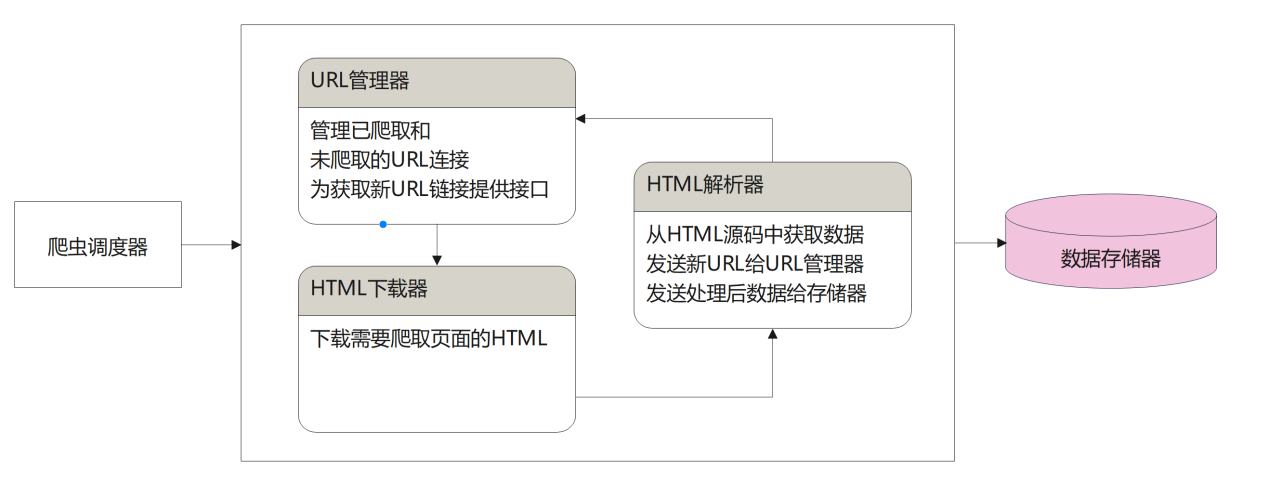
****

图2基础爬虫架构图

爬虫系统的整体流程：爬虫系统首先从互联网页面中精心选择一部分网页，以这些网页的链接地址作为种子URL，将这些种子放入待抓取URL队列中，爬虫从待抓取URL队列依次读取，并将URL通过DNS解析，把链接地址转换为网站服务器对应的IP地址。然后将其和网页相对路径名称交给网页下载器，网页下载器负责页面的下载。对于下载到本地的网页，一方面将其存储到页面库中，等待建立索引等后续处理；另一方面将下载网页的URL放入已抓取队列中，这个队列记录了爬虫系统已经下载过的网页URL，以避免系统的重复抓取。对于刚下载的网页，从中抽取出包含的所有链接信息，并在已下载的URL队列中进行检查，如果发现链接还没有被抓取过，则放入待抓取URL队列，爬虫会采用不同的抓取策略，如广度优先遍历、深度优先遍历、非完全PageRank、反向链接数、OPIC、大站优先策略等，来确定待抓取URL队列中URL的优先顺序，在之后的抓取调度中会下载这个URL对应的网页。如此这般，形成循环，直到待抓取URL队列为空，这代表着爬虫系统将能够抓取的网页已经悉数抓完，此时完成了一轮完整的抓取过程。

搜索器的工作重点有两部分，一是搜索策略，包括优先搜索重要网页的策略、及时更新页面信息、搜索隐藏网页等技术，目的在于提高抓取网页的覆盖率、时效性和重要性，提高搜索引擎的用户体验感。二是系统结构能在下载页面速度高时确保其性能稳定，即效率的提高。

### 1.1.2信息处理模块

信息处理模块可以对爬取到的信息进行进一步处理、构建索引，存储到索引数据库便于查询操作。

#### 1.1.2.1网页内容分析

对于爬取的内容，搜索引擎需要对其进行各种预处理，才能构建索引存入索引数据库。处理技术包括网页净化和去重、关键词提取。[3]

网页净化和去重。在网页中，除了网页的主题内容之外，还包含一些噪声信息。例如：广告、导航条、版权信息等。大量的噪声会引起主题漂移，使得同一网页包含多个主题，而且它们给互联网上的各种应用的效率、效果、存储等带来不利影响。网页净化技术可以去除网页中与应用不相符合的内容，提高索引质量。网页去重技术则是删除内容大量重复的网站，针对同源网页，可以排除相同URL，针对网页内容，可以去除聚类结果距离过近网站。[4]

关键词提取。一般将内容先进行切词处理，再通过词频统计的方式提取出网页的重要关键词列表，提取后放入索引数据库。

#### 1.1.2.2构建数据索引

索引器的功能是理解搜索器所搜索的信息，从中抽取出索引项，将文档表示为一种便于检索的方式并存储在索引数据库中，生成文档库的索引表。

倒排表是当前搜索引擎广泛应用的索引技术，也是最有效的实现方式，其存储<关键词,网页>的映射关系，在搜索关键词后，只需要扫描关键词列表就可找到相关网页。

在建立索引时，可以采用两遍文档遍历法、归并法。由于网页是动态变化的，因此在网页变化后，需要利用倒排索引、临时索引和已删除文档列表。当搜索引擎系统发现有新的文档被爬虫下载，会先将其加入临时索引中；如果有文档被删除，则将其加入删除文档队列；文档被更改的话，将原来的文档加入删除文档队列，更改后的文档加入临时文档队列。

### 1.1.3查询模块

检索器的功能是根据用户的查询在索引库中找出相关文档，进行文档与查询的相关度评价，返回相关度符合某一阈值的文档集合。

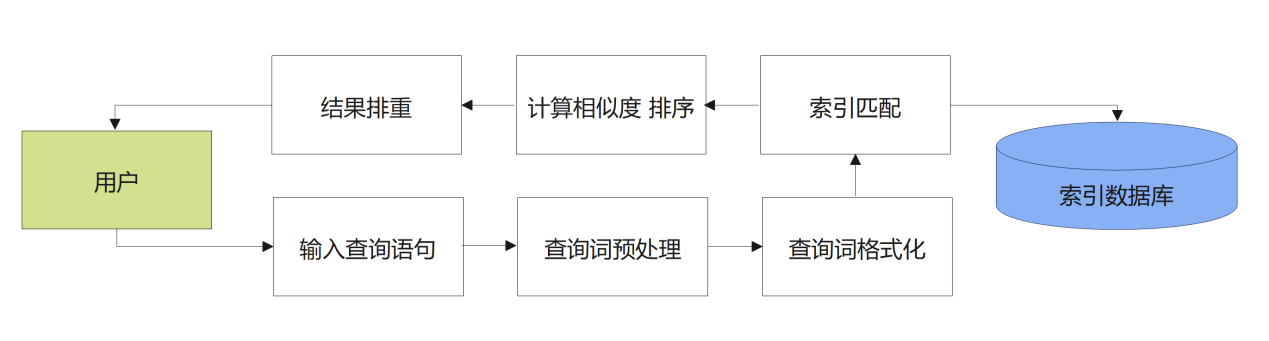
****

图3 查询流程图

#### 1.1.3.1响应用户检索

在用户输入后，搜索引擎需要从索引数据库中根据用户输入的查询语句进行查找。搜索引擎需要首先处理用户输入的查询语句，进行过滤助词和表单符号、分词、去重，然后根据查询词从倒排索引库中提取匹配的检索结果。

#### 1.1.3.2检索结果处理

搜索引擎通常会对一个查询返回大量的结果列表，混合着相关文档和不相关文档，为提高查询效率和用户体验感，可以进一步对结果列表进行整理。

提供摘要。向用户返回结果时给出文档摘要，帮助用户决定是否进一步查看此文档。

检索结果排序。可以根据用户所查询的词条在文档中的出现情况，包括词频、位置等因素，得出网页与用户查询的相关度，也可根据该页面被其他页面引用的次数，得出该页面的重要性。综合上面两种因素对检索结果进行排序呈现。

检索结果聚类。由于用户所查询的页面通常应该内容相似，因此在聚类后这些页面的距离也会比较近，分在同一类别中，因此给用户呈现检索结果时，可以集中考虑聚类距离近的一批文档。

推测用户目的。在大数据时代，除了用户的查询语句，我们也可以了解用户的历史查询信息、同类用户的查询记录，分析这些信息可以推测用户在查询语句之外的查询目的，针对这些目的修正查询结果，提高搜索结果的精确度。[2]

## 1.2技术选型

搜索引擎的运行过程中通常伴随着超高并发、超大数据量，由于率先对其进行研究和开发，当前国外主流搜索引擎技术已经比较成熟，包括结果、性能和稳定性等，而国内搜索技术相对落后，多数采用集中式的体系结构，将爬取的页面经分析处理后采用集中式存储服务器存放，用户通过访问这个站点实现查询。这种集中式的体系结构对每台服务器性能要求很高，系统的安全性和可扩展性相对较差。另外由于其每台服务器是独立执行搜索和分析处理，带来很大的宽带浪费和重复性工作，容易造成网络阻塞，难以适应当前日益扩大的数字化信息规模。而如果仅仅改用大型服务器来改善性能，则费用昂贵。因此开发具有分布式处理能力的搜索引擎，实现高性能、高效、快速和低成本的信息搜索十分有必要。

Hadoop 是一个由 Apache 基金会开发的分布式系统基础架构，其主要模块有：

HDFS:分布式存储系统，采用Master/Slave结构，由一个NameNode和多个DataNode组成，NameNode协调用户对文件的访问，DataNode负责数据存储，将一个完整的文件拆分成多个块，每个块分别存储到数据节点的磁盘中。

YARN: 任务调度和集群资源管理。

MapReduce:一种基于Hadoop YARN的大型数据集并行计算处理系统，也采用Master/Slave结构。由一个JobTracker和多个TaskTracker组成，JobTracker负责作业调度，TaskTracker执行计算任务，所有TaskTraker都需要运行在DataNode上。

Hbase：分布式的、面向列的开源数据库。

Hive：基于Hadoop的一个数据仓库工具。

Spark：Hadoop数据快速通用的计算引擎。

​ ZooKeeper: 一个分布式的，开放源码的分布式应用程序协调服务，是Hadoop和Hbase的重要组件。

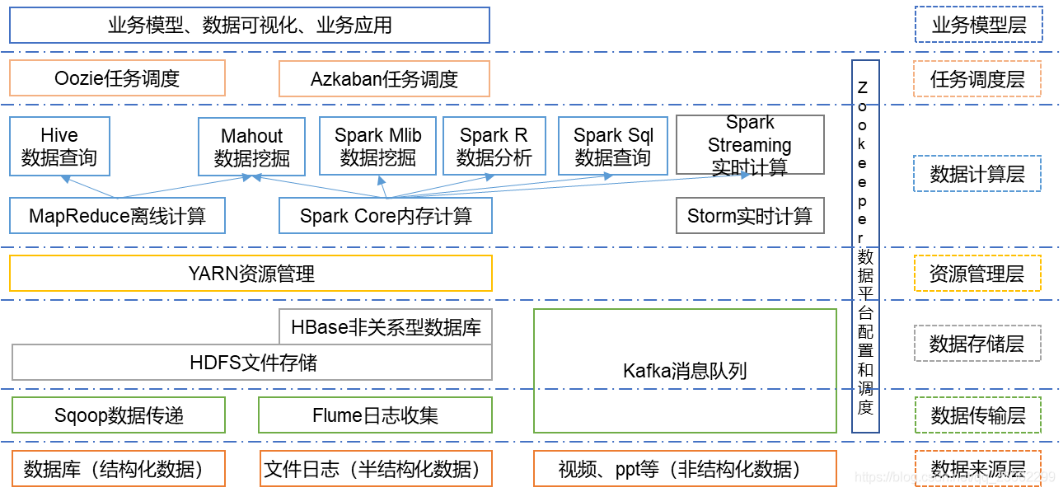


图4 Hadoop生态圈

其框架的核心是HDFS和MapReduce，HDFS为海量的数据提供了存储，而MapReduce为海量的数据提供了计算。

在本文中，我们选择Hadoop作为分布式搜索引擎的基础架构进行开发，在本节中将对Hadoop的优势和劣势分别进行分析。

### 1.2.1优势

(1)Hadoop平台数据处理高效。MapReduce的功能是对海量的数据进行并行处理，在模型中首先对输入的键值对数据<key, value>进行分割，然后分配给各个节点的Map函数，Map函数把分配到的数据映射为另一组键值对<key2, value2>，由各个节点的Reduce函数对中间数据进行规约，最终在汇总服务器上进行汇总。因此数据在不同节点上可以进行并行处理，对于18GB的数据量，用集群比用单机节省4.8倍的时间。而且，随着待处理数据量的不断增大，Hadoop集群为机构数据处理节省的时间比单机越来越大。Hadoop能够在节点之间动态地移动数据，并保持各个节点的动态平衡，因此处理速度非常快，这满足搜索引擎对数据处理的速度和质量要求。

(2)Hadoop平台来源开放，使用成本低。因此我们可以将资金放在其他开发应用，同时带来Hadoop版本快速更新与技术创新。

(3)Hadoop具备高扩展性。Hadoop是在可用的计算机集群间分配数据并完成计算任务，这些集群便于扩展，可以适当扩展集群数来满足日益增加的数据量，而这并不会毁坏原集群的特性。

(4)安全可靠性高、容错性高。HDFS可以实现将文件切分成固定大小的数据块，存储在相应的节点上，并且还会对每一块进行备份，不同的副本存储在不同的节点上，集群的数据冗余机制使得Hadoop能从单点失效中恢复，即Hadoop能自动进行数据的多次备份，以确保数据不丢失，即使当某个服务器发生故障时，它也能重新部署计算任务。

（5）高容错性。Hadoop不仅能够自动保存数据的多个副本，并且能够自动失败的任务重新分配，比如一个计算机出现故障了，那有其他副本的计算机会重新进行工作。

Hadoop使中小型开发团队更容易进行数据分析，减少计算和存储所需的硬件空间，减少数据计算的时间，提升搜索效果，快速高效地获取有价值的信息。除了实现数据分析，Hadoop也帮助中小型团队更快速高效地从网络和其自身部署的网络应用中获取有价值的数据。充分利用Hadoop集群良好的扩充能力、较低的运作成本、较高的效率和较好的稳定性，团队可以在缺乏雄厚的资金和技术人才的情况下提升搜索服务质量。

### 1.2.2不足

由于不同代的Hadoop集群功能上略有区别，本文主要讨论Hadoop3.x集群的缺点。

(1)Hadoop不适用于低延迟数据访问。

(2)Hadoop不能高效存储大量小文件。Hadoop适用于处理相对较大的文件，但是涉及到处理大量小文件的时(小文件比Hadoop的块大小小得多的文件，默认情况下，该块大小可以为128MB或256MB)，Hadoop效率不高。这些大量的小文件使Namenode过载，因为Namenode存储了系统的名称空间，并使Hadoop难以运行。

(3)Hadoop不支持多用户写入并修改任意文件。

(4)安全问题。Hadoop用Java编写，Java是一种广泛使用的编程语言，因此它容易被网络犯罪分子利用，这使得Hadoop容易受到安全漏洞的攻击。

(5)处理开销大。在Hadoop中，数据是从磁盘读取并写入磁盘的，这在我们处理兆兆字节和PB级数据时使读/写操作非常昂贵。Hadoop无法执行内存中计算，因此会增加处理开销。

(6)仅支持批处理。Hadoop 的 核心是一个批处理引擎，该引擎在流处理方面效率不高。它不能以低延迟实时生成输出。它仅适用于我们在处理之前预先收集并存储在文件中的数据。

(7)迭代处理。Hadoop本身无法进行迭代处理。机器学习 或迭代处理具有周期性的数据流，而Hadoop的数据是在多个阶段链中流动的，其中一个阶段的输出成为另一阶段的输入。

## 1.3功能实现

在选择Hadoop作为实现搜索引擎的模块后，我们对搜索引擎的三大部分分别进行功能实现说明。[5]它们相互独立，可以单独运行，所以都可采用MapReduce并行实现，同时在子系统间通过接口实现最终的分布式搜索。

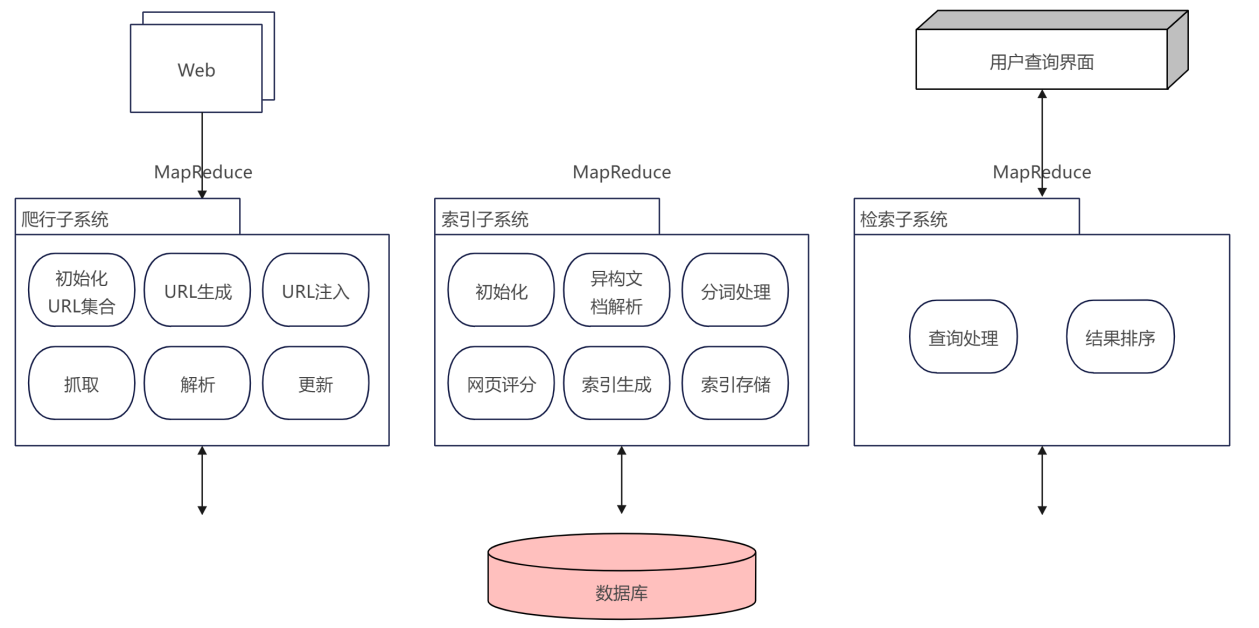
****

图5 系统架构图

### 1.3.1分布式爬行子系统

爬行子系统的主要功能是抓取网页数据，提取链接并生成下一次爬行的链接列表，数据库主要由网页链接库，存放爬取的网页url信息，和网页数据库，存放爬取的网页数据。分布式爬行子系统是由部署在Slave节点上的爬行器和调度器组成的。调度器爬行子系统设计的核心就是任务调度，其所有的爬行器统一JobTracker负责调度。[6]

执行流程如下：

(1)JobTracker首先获得所有TaskTracker的信息，形成空闲Slave节点列表。访问网页链接库获得爬行链接列表的规模，此后JobTracker根据空闲节点的数量和实际网页链接存储的物理位置，将链接列表分割为多个爬行任务，分发给空闲节点。

(2) TaskTracker接收任务后，启动Map进行数据下载，下载完成后Map将数据映射为<网页链接，网页数据>这种类型的<key,value>键值对，作为中间数据。

(3)TaskTracker监测到Map的输出后会启动Reduce，Reduce汇总中间数据后，会根据网页链接从中间数据中提取出网页数据，对网页数据进行分析，提取出对其他网页的引用连接，同时记录这些链接，将网页数据保存到分布式文件系统，将提取出的链接添加到网页链接列表，供下一轮爬行使用。

### 1.3.2分布式索引子系统

分布式索引子系统主要负责计算网页的PageRank值，构建倒排文档，以及索引文件的分布式存储。此系统为整个搜索引擎的核心。

Map/Reduce计算模型首先进行网页信息处理：网页去重和生成倒排索引。

网页去重我们采用简单策略，目标是将网页集合内所有内容相同的网页找出来，采取对网页内容取哈希值的方法，比如如果两个网页的MD5值相同，则可以认为两页内容完全相同。在Map/Reduce框架下，输入数据是网页本身，可以用网页的URL作为输入数据的Key,网页内容是输入数据的value，Map操作则对每个网页的内容利用MD5计算哈希值，以这个哈希值作为中间数据的Key，网页的URL作为中间数据的value，Reduce操作则将相同Key的中间数据对应的URL建立成一个链表结构，这个链表代表了具有相同网页内容哈希值的都有哪些网页。这样就完成了识别内容相同网页的任务。

对于建立倒排索引这个任务，在爬行子系统时得到的是正排文档，数据形式为<DocID,Doc>，倒排文档希望得到<word，List(DocID)>的形式。输入数据是网页，以网页的DocID作为输入数据的Key, 网页中出现的单词集合是输入数据的Value。Map操作将输入数据转化为(word，DocID)的形式，即某个单词作为key， DocID作为中间数据的value，其含义是单词word在DocID这个网页出现过；Reduce操作将中间数据中相同Key的记录融合，得到某个单词对应的网页列表<word，List(DocID)>。这就是单词word对应的倒排列表。通过这种方式就可以建立简单的倒排索引，在Reduce阶段也可以做些复杂操作，获得形式更为复杂的倒排索引。执行流程如下：

(1)JobTracker将网页数据库分为m块，形成m个子任务，通过与TaskTracker的通信，将这些任务分配给m个空闲的TaskTracker。

(2)每个TaskTracker启动2个Map，Map对输入的每一个<网页链接，网页数据>进行处理，执行分词，网页去噪（去除没有实际意义的词语，如“是”、“的”），记录关键词所属的文档号和所在文档的位置等信息，生成<关键词，文档号和位置信息>的中间数据并输出给Reduce。

(3)每个TaskTracke启动1个Reduce，它汇总中间数据，根据关键词进行排序，合并相同关键词的Value，生成<Key,List(value)>数据构建倒排文档（索引），每个Slave完成的只是最终索引的一部分，都暂存在本地，经过合并后再保存到分布式文件系统中。文件系统使用副本存储策略来实现可靠性。系统的复制因子为3，意味着每个块都会有3个副本，分别位于3个DataNode上，其中一个位于不同机架的DataNode上。所有块的元数据都被注册在NameNode。当一个DataNode出现故障后，其保存的块仍然可以通过NameNode上注册的冗余块进行读取。NameNode周期性的收到来自集群内DataNode的心跳报告，能收到心跳证明DataNode工作是正常的，如果NameNode没有收到一个DataNode的心跳报告，则说明此DataNode已经出现故障，这时NameNode把应由此DataNode保存

的块的副本复制后存储到其他DataNode上，时刻保持系统中每个块都会有3个副本，以此来保证系统的高可靠性。当用户读取文件时，首先必须向NameNode提交读取请求，NameNode查询元数据表后将文件的元数据(此文件分为几个块，每个块属于哪个DataNode等)返回给用户。接下来用户直接访问相关DataNode获得所需块，得到完整的文件。而当用户保存文件时，同样首先向NameNode提交保存请求，NameNode在文件命名空间中写入文件名，根据文件大小将文件分割为许多64M的片段，并查询元数据表为文件分配空闲块，最后将相关元数据返回给用户，然后用户直接与相关DataNode建立连接，将文件写入到块中。

### 1.3.3分布式查询子系统

分布式查询子系统主要负责响应用户查询请求，并向用户返回查询结果，它是搜索引擎系统与用户交互的平台。查询子系统采用的也是MapReduce模型。

用户在网页中输入关键词，Web服务器获得数据后提交给查询模块，提取出关键词，生成查询任务，通过JobTracker把查询任务发送到TaskTracker，Map根据关键词检索分布式文件系统得到索引数据，Reduce汇总索引数据并根据PageRank得分进行排序之后输出到Web服务器，Web服务器对数据缓冲，分页，最后把查询结果以网页形式呈现给用户。

## 1.4工作计划

**第一阶段 设计hadoop分布式计算引擎**

搜索引擎的信息绝大部分来源于别的网站，获取信息的过程就是利用至主程序对这些网站目录进行分析，并对内容进行抓取。这一步算法的优劣性决定着搜索引擎的信息丰富程度及结果的准确性。

**第二阶段 嵌入倒排文档全文检索技术**

倒排文档与人们日常所认识的正排文档有很大区别，但是在全文检索领域却扮演者极为重要的角色。

**第三阶段 中文分词技术**

分词处理是信息采集的后续工作，同时也是搜索引擎技术中相对比较核心的技术。其算法的好坏对搜索引擎的效率、信息量和结果的准确性都有着直接或间接的影响。

**第四阶段 分布式数据库设计**

搜索引擎所有的操作都是围绕数据库来实现的，在项目开发中数据库的设计至关重要，从短期来看，数据库的设计直接决定着搜索引擎的效率和整个系统的架构；从长期来看，它还决定着以后运行过程中的维护、升级等相关操作。所以，我们必须根据需求设计出良好的数据库。

**第五阶段 开发web客户端**

为了提高用户体验以及帮助用户进行高效率的检索，前端web界面需要提供一些简单的提示功能。

**第六阶段 测试功能，完善功能**

## 1.5组织结构

### 1.5.1人员构成

表 1 人员构成表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 性别 | 所属院校 | 年级 | 专业 |
| 郑子帆 | 男 | 北京理工大学 | 20级 | 计算机科学与技术 |
| 王天泽 | 男 | 北京理工大学 | 20级 | 计算机科学与技术 |
| 安禹堃 | 男 | 北京理工大学 | 20级 | 计算机科学与技术 |
| 张驰 | 男 | 北京理工大学 | 20级 | 计算机科学与技术 |
| 黄昱欣 | 女 | 北京理工大学 | 20级 | 计算机科学与技术 |

### 1.5.2分工

**郑子帆（组长）：**

基于Ubuntu的Hadoop环境搭建，伪分布式环境实现，实验报告撰写

**王天泽：**

基于Ubuntu的Hadoop环境搭建，倒排索引的代码编写

**安禹堃：**

基于CentOS的Hadoop环境搭建，分布式环境实现，实验报告撰写

**张驰：**

基于CentOS、Ubuntu的Hadoop环境搭建，分布式环境实现，技术文档撰写

**黄昱欣：**

基于CentOS的Hadoop环境搭建，分布式环境实现，技术文档撰写

## 1.6软件质量保证、非功能性保证

### 1.6.1产品特性

#### 1.6.1.1功能性要素

指系统实际具有的功能以及系统提供满足明确的和隐含的需求的能力。功能性在对软件评价中占有重要地位，如果提交软件的功能性得不到满足，其它属性就没有了存在的价值。

(1)需求覆盖性

为证明被测软件能否满足技术指标的要求，必须详细列出软件功能、性能测试表，也就是说根据软件需求规格说明中规定的功能、性能要求，对所验收的软件逐项进行测试或检查已有的测试结果。

(2)效率：反应时间、处理时间、资源消耗测试

反映了规定的条件下软件性能水平与所使用资源量之间的关系。性能与效率体现软件实际运行时的处理能力：由于软件类型的差异性，不同软件架构、不同运行环境、不同用户类型，软件的性能与效率均会表现出很大的差异性。

(3)复杂性：需求复杂性及源码复杂性度量

复杂性是功能性的一个重要属性，一般复杂度越高，完成的功能越强。复杂性是度量技术研究较多的质量属性，可以应用需求分析中的用例数量及复杂度、系统设计方案和源代码中的类数、程序结构复杂性度量等来度量复杂性。

(4)易用性：用户友好性

现代软件产品，人机关系越来越密切，在软件设计时应充分考虑用户的操作，运行的界面标准应操作便捷、显示清晰、尽可能多的提供帮助信息。

(5)适应性：配置分支评估

#### 1.6.1.2正确性

要求软件本身提供强有力的管理工具或功能。当软件发生不可避免或不可预见的故障时，可以迅速恢复数据，从而可以避免对业务的影响。这对于软件产品的长期稳定运行、节省系统的运行成本有着重要意义。

(1)精确性：输出结果与期望结果的匹配程度

当用户输入合理时，系统的输出结果与用户期望的精确度的匹配程度，可采用匹配算法度量。

(2)容错性：异常数据测试评价

指我们在承认软件系统存在不正确与不精确因素的前提下，为了防止潜在的不正确与不精确因素引发灾难，系统为此设计了安全措施，对它的度量可采用异常数据测试评价。

(3)可靠性：故障数和失效率度量

指在既定的时间内，系统正确运行，不发生故障的能力，可以通过出错率来进行度量。

#### 1.6.1.3可维护性

软件维护占据了软件生命周期成本的大部分花费。改正错误、提高性能、维持软件的正常工作都需要软件的维护。这要求软件产品具备清晰、良好的内部结构，结构复杂性较低，易于阅读和理解，从而方便对软件产品的修改、维护以及复用。

(1)可更正性：错误数及发现和修改错误需要的资源、工作量度量

(2)可测试性：测试程度和资源消耗、工作量度量

①测试计划的完整性

②测试用例的数量

③分支覆盖率

④语句覆盖率

⑤耗费资源及工作量预测

(3)可扩充性：变更数（变更规模、变更率）和所需资源、工作量的评估

在保持软件不变的情况下计算机系统性能可随系统规模扩充而提高的特性。

#### 1.6.1.4可重用性

对于每个软件开发项目而言，其开发过程和经验都是一笔财富，尤其是成功的软件开发项目。对于一个成熟的软件企业来说，企业五领域一般集中在几个方面，所以在以后的项目中，可以充分利用以往成功的经验。软件开发企业可以通过不同级别的复用，提高生产效率，保证软件质量。

(1)构件化程度

指软件构件的独立性和构建功能的一般性程度，可以通过设计方案中的子节点数、继承深度、封装度、依赖度度量。

(2)标准化程度

指软件是否遵从某个较通用的标准，软件文档是否遵从某个较通用的标准。

(3)跨平台程度

指软件在其它平台上能直接运行或经过一定的修改而运行的能力，可以通过分析评估和测试度量。

### 1.6.2过程特性

指从软件的生产过程来评价软件的质量。软件质量保证不仅仅要求最终生成的软件产品具有高可用、高可维护性及高可重用性，同时要求产生这个软件产品的过程也是高质量的，利于以后类似软件产品的生产重复。

#### 1.6.2.1理解过程化程度

过程特性将软件生成的过程和结果融合到一起，从延续的观点看待软件质量，只有高质量的软件开发经验才是值得推广和重用的。

(1)定义过程化程度

在软件项目初始阶段对项目开发所要经历的过程进行定义的程度，可以通过分析和评估确定。

(2)传播过程化程度

指对所定义的过程的传播力度，它是过程有效执行的前提，可以通过项目组成员对定义过程的了解程度来度量。

#### 1.6.2.2实施过程化程度

过程实施数据收集和度量。

(1)计划时间、实际时间

(2)计划成本、实际成本

(3)后续发现的错误数、最终用户报告的错误数

### 1.6.3用户满意度特性

用户满意度是软件开发的重要目的之一，是对产品质量的一个最终确认。加强用户满意度是商业成功的基础。

(1)用户忠诚度

(2)使用依赖性

(3)价格满意度

(4)用户口碑

(5)心理满足感

# 参考文献

[1] 张伟哲,等.分布式搜索引擎系统效能建模与评价[J].软件学报,2012(2):253—265.

[2] 印鉴,陈忆群,张钢. 搜索引擎技术研究与发展[J]. 计算机工程,2005,31(14):54-56,104. DOI:10.3969/j.issn.1000-3428.2005.14.020.

[3] 搜索引擎原理：数据预处理，[https://blog.csdn.net/gaosini0001/article/details/46391407?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522166381616516782428671794%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=166381616516782428671794&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~baidu\_landing\_v2~default-1-46391407-null-null.142^v49^new\_blog\_pos\_by\_title,201^v3^control\_2&utm\_term=%E6%90%9C%E7%B4%A2%E5%BC%95%E6%93%8E%E6%96%87%E6%9C%AC%E9%A2%84%E5%A4%84%E7%90%86&spm=1018.2226.3001.4187](https://blog.csdn.net/gaosini0001/article/details/46391407?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522166381616516782428671794%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=166381616516782428671794&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~baidu_landing_v2~default-1-46391407-null-null.142%5ev49%5enew_blog_pos_by_title,201%5ev3%5econtrol_2&utm_term=%E6%90%9C%E7%B4%A2%E5%BC%95%E6%93%8E%E6%96%87%E6%9C%AC%E9%A2%84%E5%A4%84%E7%90%86&spm=1018.2226.3001.4187)

[4] 罗元. 网页净化及去重研究综述[J]. 现代计算机（专业版）,2013(10):3-7,12. DOI:10.3969/j.issn.1007-1423.2013.15.001.

[5] 郑睿颖,王芷婷. 基于Hadoop的分布式搜索引擎的研究[J]. 求知导刊,2017(35):9-10. DOI:10.3969/j.issn.2095-624X.2017.35.004.

[6] 胡彧,封俊. Hadoop下的分布式搜索引擎[J]. 计算机系统应用,2010,19(7):224-228. DOI:10.3969/j.issn.1003-3254.2010.07.054.