**小组体系结构方案设计**

目录

[一：组员情况 2](#_Toc327214162)

[二：关注点 2](#_Toc327214163)

[三：体系结构需求定义 4](#_Toc327214164)

[体系结构需求描述和设计约束 4](#_Toc327214165)

[用例视图 6](#_Toc327214166)

[非功能用例场景 7](#_Toc327214167)

[四：设计决策 9](#_Toc327214168)

[五：最终高层体系结构 11](#_Toc327214169)

[系统介绍 11](#_Toc327214170)

[Hadoop 12](#_Toc327214171)

[Nutch 13](#_Toc327214172)

[体系结构 14](#_Toc327214173)

[逻辑视图 14](#_Toc327214174)

[开发视图 15](#_Toc327214175)

[进程视图 17](#_Toc327214176)

[部署视图 20](#_Toc327214177)

[六：小组分工 20](#_Toc327214178)

# 一：组员情况

**全组共9人，具体人员情况如下：**

**PM：**钟晓诚（091250232）

**组员：**靳峥（091250069），鞠元（091250070），李东煦（091250072），娄鹏呈（091250094），陆昊君（091250097），陆君之（091250099 ），陆星恒（091250102），陆怡平（091250103），周率（091250236）

**大组内又分为4小组，具体的小组及人员分配情况如下：**

**小组一：**

**PM：**李东煦（091250072）

**组员：**靳峥（091250069），娄鹏呈（091250094）

**小组二：**

**PM：**陆昊君（091250097）

**组员：**陆君之（091250099 ）

**小组三：**

**PM**：陆星恒（091250102）

**组员：**陆怡平（091250103）

**小组四：**

**PM：**钟晓诚（091250232）

**组员：**鞠元（091250070），周率（091250236）

# 二：关注点

以列表方式展现本搜索引擎体系结构的关注点，包括功能需求、质量以及项目环境等因素，见下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **关注点** | **类型** | **描述** | **灵活性** |
| 网页爬取 | 功能需求 | 实现对网页的采集工作，用于站点资源的监视和资料库的更新 | 采集过程中，可以构造适当的启发策略，来指导机器人的路径选择和采集范围，减少文档采集的盲目性 |
| 内容处理 | 功能需求 | 对收集到的内容进行处理，提取特征元素 | 灵活性不大，要提取的特征元素可能发生变化 |
| 全文索引 | 功能需求 | 为收集到的内容建立索引以便于检索 | 灵活性不大，基本稳定 |
| 快速检索 | 功能需求 | 根据用户提供的检索条件实现快速的匹配 | 匹配算法有可能发生变化 |
| 排序 | 功能需求 | 将搜索结果按相关度进行排序, 把最相关的结果放在最前面 | 排序算法有可能发生变化 |
| 用户接口 | 功能需求 | 为用户提供适当的交互界面，对用户输入词汇进行解析 | 解析算法可能发生变化 |
| 定时爬取 | 质量属性 | 网页爬取能够定期执行，定期更新储存库 | 时间可能发生变化 |
| 并发爬取 | 质量属性 | 爬取应该能够多机器（>=3）同时并发进行 | 并发机器数目有可能发生变化 |
| 可扩展性 | 质量属性 | 系统能够存储大容量数据，能够分布式使用多台机器的存储设备 | 能够在2小时内添加新的数据存储设备以扩充存储容量 |
| 安全性 | 质量属性 | 系统中储存的内容应该加密 | 加密算法可能发生变化 |
| 及时性 | 质量属性 | 系统应反应及时 | 能够在10秒内给出查询结果 |
| 可靠性 | 质量属性 | 系统应及时发现系统中的故障 | 能够在1分钟内发现各服务器及进程的故障 |
| 易用性 | 质量属性 | 系统要具有高易用性 | 在查询时，能够返回“非字符匹配”的相关结果 |
| 容错性 | 质量属性 | 系统可能发生故障，但必须拥有尽快修复故障的能力 | 系统应能够在4小时内能够恢复工作 |
| 可修改性 | 质量属性 | 系统的要求可能会发生变更 | 可能的变更点包括：爬取算法；对爬取网页的解析规则；加密算法；检索匹配算法；排序算法等等 |
| 法律规则 | 质量属性 | 系统应能够进行敏感词过滤 | 敏感词随时可以调整 |
| 商业规则 | 质量属性 | 系统能够实现竞价策略，可按照加权的方式对某些搜索结果的先后顺序进行调整 | 加权算法可能发生变化 |
| 人员技能 | 开发环境 | 团队成员对搜索引擎开发技术了解欠缺 | 灵活性不大，只有加强团队的学习能力 |
| 团队组织 | 开发环境 | 项目计划有时间限制，在学期结束前必须有系统原型交付 | 灵活性不大，项目交付时间基本不会变化 |
| 无 | 商业环境 | 无 | 无 |
| 软件环境 | 技术环境 | 不要求多平台、多浏览器的系统实现 | 灵活性变化不大 |
| 硬件环境 | 技术环境 | 系统应运行在至少八台机器上 | 拥有随时增加计算或存储设备的能力 |
| 支撑技术 | 技术环境 | 系统在开源的搜索引擎框架上修改完成 | 灵活性变化不大 |

# 三：体系结构需求定义

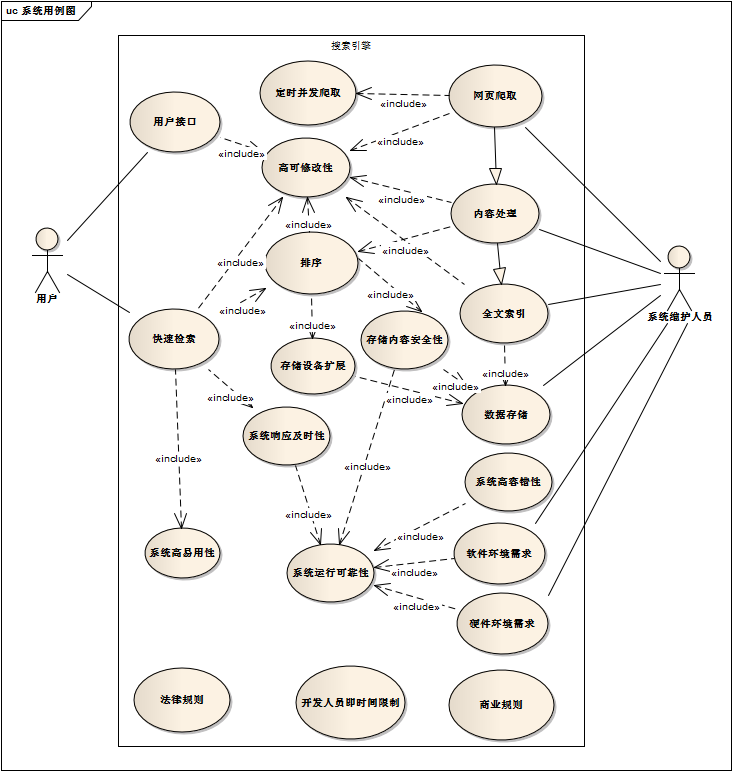
## 体系结构需求描述和设计约束

系统的体系结构需求描述和设计约束如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **体系结构需求ID** | **描述** | **设计约束** | **相关约束** | **优先级（小为高）** |
| R1 | 网页爬取功能：网页爬取实现对网页的采集工作，即要对海量的网页进行数据的采集 | C1 适当的启发策略，减少盲目性 |  | 1 |
| R2 | 内容处理功能：对收集到的内容进行处理，提取特征元素。 | C2 系统应能够处理项目过程中特征元素发生变化的情况 |  | 2 |
| R3 | 全文索引功能：为收集到的内容建立索引以便于检索 | 无 |  | 2 |
| R4 | 快速检索功能：根据用户提供的检索条件实现快速的匹配 | C3 系统应能够根据用户提供的检索条件实现快速的匹配 |  | 1 |
| R5 | 排序功能：系统需要对结果进行排序，将用户可能觉得重要的放在前面 | 无 |  | 2 |
| R6 | 提供用户接口：为用户提供适当的交互界面 | C3系统应能够根据用户提供的检索条件实现快速的匹配 |  | 2 |
| R7 | 定时爬取：要求网页爬取每天定时进行，爬取完成后更新一次储存库 | C4 爬取应定时进行，定期更新存储库 | C1 | 3 |
| R8 | 并发爬取：爬取在3台独立机器上同时进行并相互协调，避免爬取重复数据 | C5 系统应能够支持多处理器并发爬取功能 | C1，C4 | 3 |
| R9 | 存储设备的可扩展性：系统需要有处理大数据量的能力 | C6 系统应能够在2小时内添加新的数据存储设备 |  | 3 |
| R10 | 存储内容的安全性：系统数据存储需要有一定的安全措施 | C7 系统应对存储数据提供加密算法 |  | 3 |
| R11 | 系统响应的及时性：系统能够同时允许大量用户访问，要求系统具备负载均衡能力 | C8 系统应能够在10秒内给出查询结果 | C3 | 3 |
| R12 | 系统运行的可靠性：系统采用冗余机制实现高可靠性 | C9 系统应能够在1分钟内发现各服务器及进程的故障 |  | 3 |
| R13 | 系统的高易用性 | C10 系统在查询时应能够返回“非字符匹配”的相关结果 | C3，C8 | 3 |
| R14 | 系统的高容错性：系统要有容灾能力 | C11 发生故障时系统能够在4小时内能够恢复工作 | C9 | 3 |
| R15 | 系统具有高可修改性 | C12 系统的要求随时会发生变更 | C1，C3，C4 | 3 |
| R16 | 系统应遵守法律规则 | C13 敏感词随时会变更 | C12 | 4 |
| R17 | 系统遵守商业规则 | C14 加权算法变更 | C12 | 4 |
| R18 | 开发人员要求 | C15 8-10人小组 |  | 5 |
| R19 | 开发时间要求 | C16 学期结束前 |  | 5 |
| R20 | 软件环境要求 | 系统分布式部署在运行linux操作系统的机器上 |  |  |
| R21 | 硬件环境需求 | C17 系统应运行在至少八台机器上 | C5，C6 | 3 |

## 用例视图

下面是系统的用例视图：



## 非功能用例场景

下面是对系统的非功能用例定义的可验证的场景描述（表格）：

|  |  |  |
| --- | --- | --- |
| **项目** | **内容** |  |
| **场景ID** | S1 | |
| **商业目标** | 扩大并发爬取机器数量 | |
| **相关需求和设计约束** | R8 | C5 |
| **场景内容** | **刺激** | 新的爬取机器 |
| **刺激源** | 系统维护人员 |
| **环境** | 爬取机器数量不足 |
| **制品** | 搜索引擎爬取子系统 |
| **响应** | 增加并发爬取网页的机器数目 |
| **响应的度量** | 爬取机器数量>=3 |

|  |  |  |
| --- | --- | --- |
| **项目** | **内容** |  |
| **场景ID** | S2 | |
| **商业目标** | 扩展存储设备数量 | |
| **相关需求和设计约束** | R9 | C6 |
| **场景内容** | **刺激** | 新的存储机器 |
| **刺激源** | 系统维护人员 |
| **环境** | 数据存储机器数量不足 |
| **制品** | 搜索引擎数据存储子系统 |
| **响应** | 增加新的存储机器数目 |
| **响应的度量** | 2小时内添加新的数据存储设备 |

|  |  |  |
| --- | --- | --- |
| **项目** | **内容** |  |
| **场景ID** | S3 | |
| **商业目标** | 响应的及时性 | |
| **相关需求和设计约束** | R11 | C8 |
| **场景内容** | **刺激** | 新的用户查询请求 |
| **刺激源** | 用户 |
| **环境** | 查询环境 |
| **制品** | 搜索引擎查询系统 |
| **响应** | 快速响应用户的查询 |
| **响应的度量** | 在10秒内给出查询结果 |

|  |  |  |
| --- | --- | --- |
| **项目** | **内容** |  |
| **场景ID** | S4 | |
| **商业目标** | 运行的可靠性 | |
| **相关需求和设计约束** | R12 | C9 |
| **场景内容** | **刺激** | 系统发生故障 |
| **刺激源** | 系统 |
| **环境** | 系统运行错误，发生故障 |
| **制品** | 系统 |
| **响应** | 系统及时检测出错误原因 |
| **响应的度量** | 在1分钟内发现各服务器及进程的故障 |

|  |  |  |
| --- | --- | --- |
| **项目** | **内容** |  |
| **场景ID** | S5 | |
| **商业目标** | 系统的高容错性 | |
| **相关需求和设计约束** | R14 | C11 |
| **场景内容** | **刺激** | 系统发生故障 |
| **刺激源** | 系统 |
| **环境** | 系统运行错误，发生故障 |
| **制品** | 系统 |
| **响应** | 系统恢复正常运行 |
| **响应的度量** | 发生故障时系统能够在4小时内能够恢复工作 |

|  |  |  |
| --- | --- | --- |
| **项目** | **内容** |  |
| **场景ID** | S6 | |
| **商业目标** | 系统的可修改性 | |
| **相关需求和设计约束** | R15 | C12 |
| **场景内容** | **刺激** | 系统需求变更 |
| **刺激源** | 客户 |
| **环境** | 开发过程中系统的需求发生变化，如新的排序，解析算法，新的功能性需求等等 |
| **制品** | 系统 |
| **响应** | 系统能很好的维护变更需求 |
| **响应的度量** | 需求的变更不会影响项目的进度 |

# 四：设计决策

下面是针对初始体系结构进行的一系列设计决策：

|  |  |
| --- | --- |
| **编号** | D1 |
| **需求** | R7 定时爬取 |
| **约束** | C4 爬取应定时进行，定期更新存储库 |
| **对策** | 把爬取进程设置为定时任务 |
| **影响** | 进程视图，部署视图 |
| **详细设计约束** | 对爬取进程进行修改，变为定时执行任务 |

|  |  |
| --- | --- |
| **编号** | D2 |
| **需求** | R8 并发爬取 |
| **约束** | C6 系统应能够支持多处理器并发爬取功能，要求至少3个以上处理器并发 |
| **对策** | 把爬取过程作为一个独立的进程，增加并发爬取网页的机器数目 |
| **影响** | 进程视图，部署视图 |
| **详细设计约束** | 一致性更新与Cluster访问 |

|  |  |
| --- | --- |
| **编号** | D3 |
| **需求** | R9 存储设备的可扩展性 |
| **约束** | C7 系统应能够在2小时内添加新的数据存储设备 |
| **对策** | 封装数据存储过程，提高存储设备的可扩展性 |
| **影响** | 逻辑视图；开发视图；进程视图；部署视图 |
| **详细设计约束** | 一致性更新与Cluster访问 |

|  |  |
| --- | --- |
| **编号** | D4 |
| **需求** | R11 系统响应的及时性 |
| **约束** | C8 系统应能够在10秒内给出查询结果 |
| **对策** | 设计算法进行查询优化，设计用户接口和存储数据端的通信3间隔 |
| **影响** | 开发视图 |
| **详细设计约束** | 通信规则 |

|  |  |
| --- | --- |
| **编号** | D5 |
| **需求** | R12 系统运行的可靠性 |
| **约束** | C9 系统应能够在1分钟内发现各服务器及进程的故障 |
| **对策** | 使用Ping/Echo方法检测服务器故障 |
| **影响** | 所有4个视图 |
| **详细设计约束** | Ping/Echo规则 |

|  |  |
| --- | --- |
| **编号** | D6 |
| **需求** | R14 系统的高容错性 |
| **约束** | C11 发生故障时系统能够在4小时内能够恢复工作 |
| **对策** | 使用冗余服务器提高可靠性，发生故障时使用冗余服务器 |
| **影响** | 所有4个视图 |
| **详细设计约束** | 一致性更新与Cluster访问 |

|  |  |
| --- | --- |
| **编号** | D7 |
| **需求** | R15 系统的可修改性 |
| **约束** | C12 系统的要求随时会发生变更 |
| **对策** | 通过划分模块，封装算法来降低模块间耦合度 |
| **影响** | 逻辑视图，开发视图 |
| **详细设计约束** | 提供算法接口，隐藏算法详细信息 |

|  |  |
| --- | --- |
| **编号** | D8 |
| **需求** | R19 开发时间要求 |
| **约束** | C16 学期结束前至少能提交项目的beta版本 |
| **对策** | 使用分层式结构，方便并行开发 |
| **影响** | 开发视图 |
| **详细设计约束** | 无 |

# 五：最终高层体系结构

## 系统介绍

根据实验要求，本小组通过查阅一些资料后决定使用开源的分布式系统基础架构**hadoop**搭建起搜索引擎的运行环境，通过在开源的搜索引擎**nutch**基础之上对其进行修改以满足此次实验的项目需求，最终部署在**hadoop**上并运行在**tomcat**容器中。最终的部署会如下图所示：

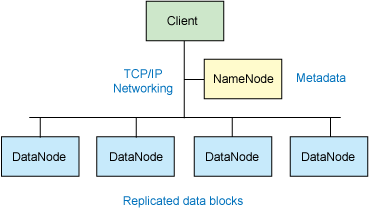


下面分别对**hadoop**和**nutch**进行简要的介绍。

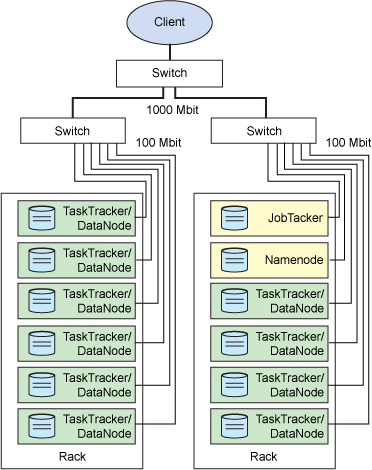
### Hadoop

hadoop是一个分布式系统基础架构，由Apache基金会开发。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力高速运算和存储。Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有着高容错性的特点，并且设计用来部署在低廉的（low-cost）硬件上。而且它提供高传输率（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序。

Hadoop 有许多元素构成。其最底部是 Hadoop Distributed File Syst Hadoop 集群的简化视图em（HDFS），它存储 Hadoop 集群中所有存储节点上的文件。HDFS（对于本文）的上一层是 MapReduce 引擎，该引擎由 JobTrackers 和 TaskTrackers 组成。如下图所示：



下面是一个显示处理和存储的物理分布的 Hadoop 集群的简单示例：



### Nutch

Nutch是一个由Java实现的开源web搜索引擎，包括crawl,distributed computing,search三个部分。Crawler主要用于从网络上抓取网页并为这些网页建立索引。distributed computing 用于进行分布式计算。Searcher主要利用Crawler生成的索引检索用户的查找关键词来产生查找结果。下图是nutch的常见工作流程：

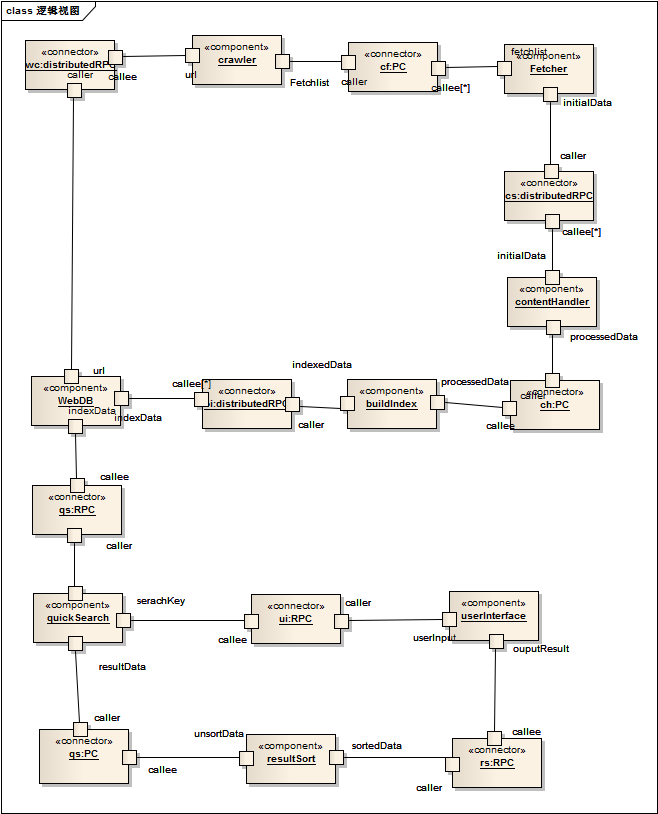


## 体系结构

使用UML表示法和4+1模型描述最终的高层结构，下面分别给出系统最终高层体系结构的逻辑视图、开发视图、进程视图和部署视图。

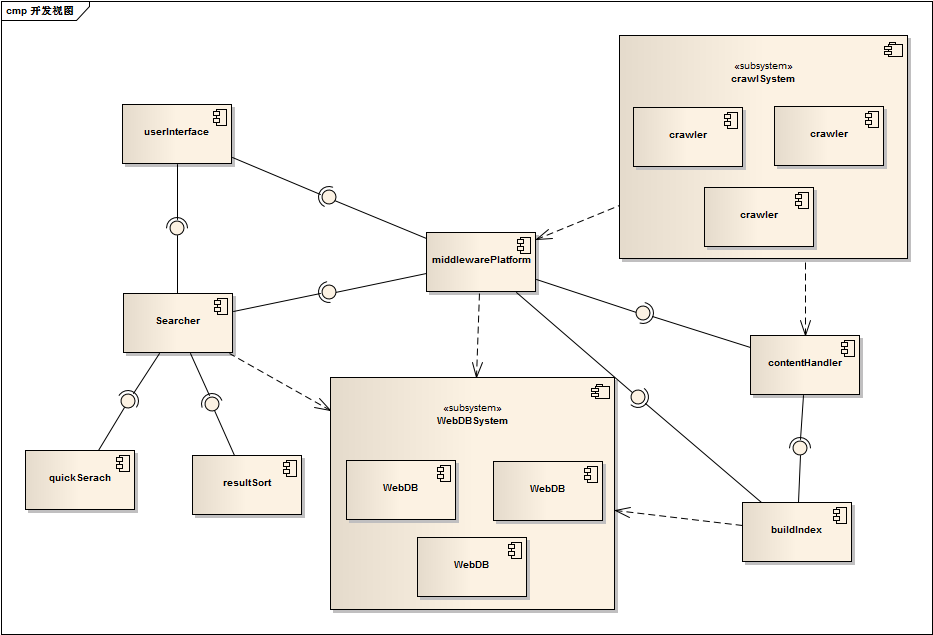
### 逻辑视图

下面是系统最终高层体系结构的逻辑视图：



### 开发视图

下面是系统最终高层体系结构的开发视图：



模块间接口定义：

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I01 | **接口名** | url获取接口 |
| **方法** | | getUrl（） | |
| **前置条件** | | WebDB中存储有未抓取的或者新发现的URLs | |
| **后置条件** | | 分布式远程RPC调用成功 | |
| **需求接口** | | 无 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I02 | **接口名** | 数据爬取接口 |
| **方法** | | fetchList（） | |
| **前置条件** | | 有url输入 | |
| **后置条件** | | 生成一个或多个urllist | |
| **需求接口** | | I01 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I03 | **接口名** | 建立索引接口 |
| **方法** | | buildIndex（） | |
| **前置条件** | | 有 urllist输入 | |
| **后置条件** | | 生成全文索引数据 | |
| **需求接口** | | I02 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I04 | **接口名** | 索引存储接口 |
| **方法** | | indexStore（） | |
| **前置条件** | | 有 索引数据输入 | |
| **后置条件** | | 全文索引数据保存如WebDB中 | |
| **需求接口** | | I03 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I05 | **接口名** | 用户搜索接口 |
| **方法** | | userSearch（） | |
| **前置条件** | | 用户输入的关键字有效 | |
| **后置条件** | | 中文分词 | |
| **需求接口** | | 无 | |

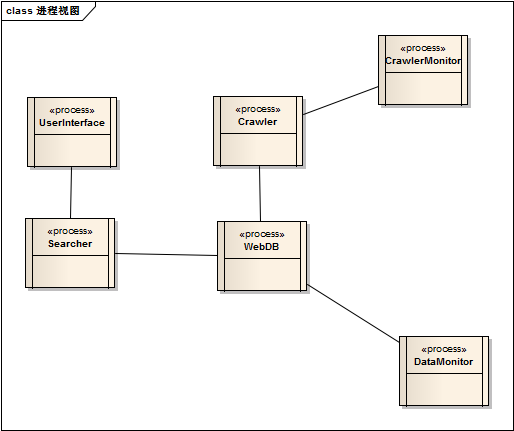
|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I06 | **接口名** | 查询接口 |
| **方法** | | indexSearch（） | |
| **前置条件** | | 经过解析后的搜索关键字 | |
| **后置条件** | | 成功返回结果数据 | |
| **需求接口** | | I05 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I07 | **接口名** | 结果排序接口 |
| **方法** | | resultSort（） | |
| **前置条件** | | 输入结果数据 | |
| **后置条件** | | 成功返回排序后的结果数据 | |
| **需求接口** | | I06 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I08 | **接口名** | 结果展示接口 |
| **方法** | | outputResult（） | |
| **前置条件** | | 输入结果数据 | |
| **后置条件** | | 向用户展现搜索结果 | |
| **需求接口** | | I07 | |

### 进程视图

下面是系统最终高层体系结构的进程视图：



进程间接口定义：

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I01 | **接口名** | url获取接口 |
| **方法** | | 远程RPC调用 | |
| **前置条件** | | 网络连通，分布式系统运行良好 | |
| **后置条件** | | 爬取进程Crawler从数据存储进程WebDB中获得未抓取的或者新发现的URLs | |
| **需求接口** | | 无 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I02 | **接口名** | 索引数据存储接口 |
| **方法** | | 远程RPC调用 | |
| **前置条件** | | 网络连通，分布式系统运行良好 | |
| **后置条件** | | 数据存储进程WebDB从爬取进程Crawler中获得爬取后的索引数据并分析存储 | |
| **需求接口** | | 无 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I03 | **接口名** | 爬取管理接口 |
| **方法** | | 远程RPC调用 | |
| **前置条件** | | 网络连通，分布式系统运行良好 | |
| **后置条件** | | CrawlerMonitor进程对运行在多个机器上的爬取进程Crawler进行协调管理 | |
| **需求接口** | | 无 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I04 | **接口名** | 数据管理接口 |
| **方法** | | 远程RPC调用 | |
| **前置条件** | | 网络连通，分布式系统运行良好 | |
| **后置条件** | | DataMonitor进程对运行在多个机器上的WebDB进程进行协调管理 | |
| **需求接口** | | 无 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I05 | **接口名** | 用户搜索接口 |
| **方法** | | 远程RPC调用 | |
| **前置条件** | | 网络连通，分布式系统运行良好 | |
| **后置条件** | | UserInterface进程对运行在其他机器上的Searcher进程发出搜索请求并传送用户输入数据 | |
| **需求接口** | | 无 | |

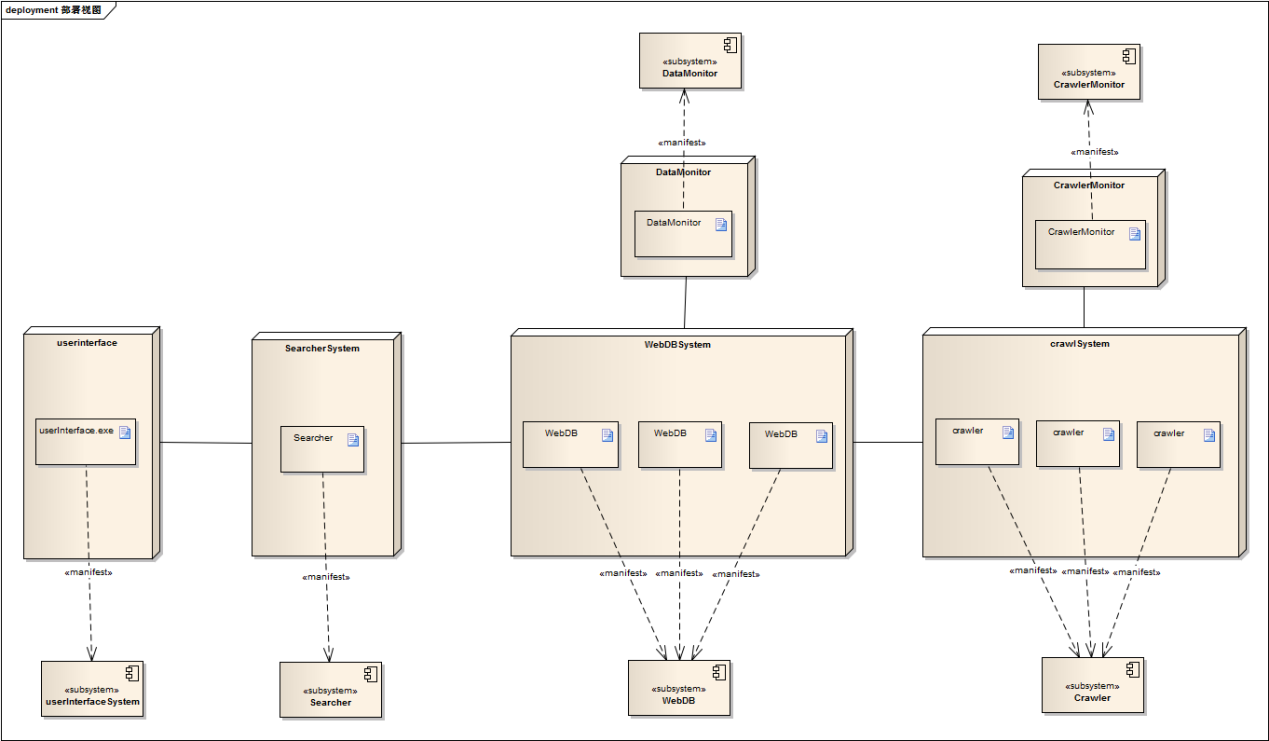
|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I06 | **接口名** | 搜索查询接口 |
| **方法** | | 远程RPC调用 | |
| **前置条件** | | 网络连通，分布式系统运行良好 | |
| **后置条件** | | Searcher进程对运行在其他多个机器上的WebDB进程发出查询请求并传送经过分词解析后的用户输入的搜索关键字 | |
| **需求接口** | | 无 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I07 | **接口名** | 查询结果返回接口 |
| **方法** | | 远程RPC调用 | |
| **前置条件** | | 网络连通，分布式系统运行良好 | |
| **后置条件** | | WebDB进程对运行在其他机器上的Searcher进程发送查询结果数据 | |
| **需求接口** | | 无 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **接口ID** | I08 | **接口名** | 结果输出接口 |
| **方法** | | 远程RPC调用 | |
| **前置条件** | | 网络连通，分布式系统运行良好 | |
| **后置条件** | | Searcher进程对运行在其他机器上的UserInterface进程发送搜索结果数据 | |
| **需求接口** | | 无 | |

### 部署视图

下面是系统最终高层体系结构的部署视图：



# 六：小组分工

|  |  |  |
| --- | --- | --- |
| **小组** | **小组成员** | **分配模块** |
| 小组一 | 李东煦（091250072） | Hadoop分布式系统平台的分析和搭建 |
| 靳峥（091250069） |
| 娄鹏呈（091250094） |
| 小组二 | 陆昊君（091250097） | 用户接口设计 |
| 陆君之（091250099 ） |
| 小组三 | 陆星恒（091250102） | 开源搜索引擎Nutch的crawler模块的分析、修改与部署 |
| 陆怡平（091250103） |
| 小组四 | 钟晓诚（091250232） | 开源搜索引擎Nutch的searcher模块的分析、修改与部署 |
| 鞠元（091250070） |
| 周率（091250236） |