

# 《并行与分布式计算》实验和课程报告

（2021年3月版）

**郑重声明：**

**1、本报告中的所有内容均由两人合作独立编码、调试和测试。本报告中给出的实验数据和结果均由本小组所完成的程序得出。**

**2、论文综述环节，每人单独完成，且内容互不相同。**

**3、本人了解：不按照前述要求所完成的实验报告已经构成了抄袭或造假行为，本人将承担相应的不良后果。**

**成员A：姓名 （签名） 学号**

**成员B：姓名 （签名） 学号**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 题目 | 单题成绩 | 成员A比例 | 成员B比例 | 成员A成绩 | 成员B成绩 |
| 实验1（30分） |  |  |  |  |  |
| 实验2（30分） |  |  |  |  |  |
| 课程论文(40分) | —— | —— | —— |  |  |
| 合计 | —— | —— | —— |  |  |

注：成员A/B比例由学生按照对代码的实际贡献比例填写，两者和为100%

### 实验1 基于MapReduce的网络搜索引擎（30分）

网络搜索引擎需要首先根据关键词构建反向索引。例如对于下述文字：

This benchmark suite contains 9 typical Hadoop workloads (including micro benchmarks, HDFS benchmarks, web search benchmarks, machine learning benchmarks, and data analytics benchmarks).

可以得到反向索引：

benchmark: URL 6 //URL为网页网址，6是该词出现在文档中的次数

This: URL 1

suite: URL 1

…

本实验将基于实际网页使用MapReduce计算反向索引文件，并构造一个初步的搜索引擎。

#### 1、使用网络爬虫收集网页（3分）

可以从此开源网站建立网络爬虫程序<http://www.oschina.net/project/tag/64>

使用的网络爬虫为：

起始搜索点设置为：<http://web.mit.edu/research/>（也可以是其他起始搜索点）

收集数据的规则是 ；

总计收集了 个HTML页面，总计 GB（应大于20GB）

#### 2、搭建Hadoop/MapReduce集群（2分）

搭建的主要过程：

你的集群包括了 个节点，每个节点的配置是 。

**硬件设置要求：至少包含两个节点。**

#### 3、基于MapReduce的反向索引构造（10分）

反向索引构造程序的设计方法是： 。在不同节点数量的构造时间如表1-1所示。

表1-1 构造反向索引的时间（*N*为系统总结点数）

|  |  |
| --- | --- |
| 节点数 | 构造时间 |
| 1 |  |
| *N*/2 |  |
| *N* |  |

在构造MapReduce程序时使用了哪些技巧？性能有多少提升？

得到的反向索引规模有多大？

#### 4、数据存储与检索（5分）

存储和检索反向索引的方法：

#### 5、搜索规则（5分）

如何根据用户关键词推荐最有效的网页？

#### 6、搜索结果页面的产生（2分）

根据搜索规则产生的结果HTLM页面，并展示给用户。请展示相应的用户界面。

#### 7、系统测试案例（3分）

设计一个搜索测试案例，并和Google等测试引擎进行对比。

### 实验2 MPI编程（30分）

MPI是高性能计算的基本编程方法。请使用MPI在多机系统上实现奇偶并行排序和PSRS(Parallel Sorting by Regular Sampling)算法，并进行性能比较。

#### MPI计算环境的搭建（5分）

你的集群包括了 个节点，每个节点的配置是 。

你使用的MPI平台是 。

MPI搭建的过程包括： 。

下文中设MPI的进程数为*P*，参与排序的总元素数量为*N*。

**硬件设置要求：至少包含两个节点。**

#### 2、数据的产生和串行快速排序的性能（5分）

请使用程序2-1产生*N*个随机浮点数，并写入数据文件，其中*seed*为外部输入的随机数种子，需要设置为(0,1.0)区间中的浮点数。使用C语言中的qsort函数按照递增方式对*N*个浮点数排序。测量排序所需要的时间（不包括文件读入时间，仅仅包含排序时间）。

表2-1 串行排序的时间

|  |  |
| --- | --- |
| *N* | 排序时间（s） |
| 256M |  |
| 1G |  |
| 4G |  |

float rand\_float(float s){

return 4\*s\*(1-s);

}

void matrix\_gen(float \*a,uint64\_t N,float seed){

float s=seed;

for(uint64\_t i=0;i<N;i++){

s=rand\_float(s);

a[i]=s;

}

}

程序2-1 随机数产生

#### 3、数据划分（3分）

设计并完成一个程序将包含*N*个浮点数的文件，平均划分为*P*个文件。划分后的文件是如何分布存储在不同的节点上？

#### 3、并行奇偶排序和PSRS（12分）

测试不同进程数和不同数据规模下的排序时间，结果如表2-2所示。

表2-2 MPI并行排序的时间

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *N* | *P* | 奇偶并行排序时间（s） | PSRS排序时间(s) | 加速比 |
| 256M | 2 |  |  |  |
| 4 |  |  |  |
| 8 |  |  |  |
| 1G | 2 |  |  |  |
| 4 |  |  |  |
| 8 |  |  |  |
| 4G | 2 |  |  |  |
| 4 |  |  |  |
| 8 |  |  |  |

#### 4、并行排序的性能分析（5分）

主要包括：两种并行排序方法的性能特点，与单个节点的性能比较，影响并行排序性能的主要因素，后续的改进方法等。

## 内容3 课程论文（40分）

从国际学术期刊或会议中选择并阅读并行与分布式计算领域的特定专题3~5篇论文，撰写相关论文综述。相关专题可以包括但不限于：特定问题的并行与分布式算法、并行与分布式计算系统的设计与实现技术、

【论文综述提纲与评分标准】

1、问题背景（5分）

准确描述所综述论文研究的问题，以及该问题的难点。

评分标准：

5分：问题描述准确和清晰，所讨论问题为并行与分布式计算应用的中的热点问题

3~4分：问题描述较为准确，可读性稍差，所讨论问题适用面较窄

1~2分：问题描述混乱不清，难以理解。

2、研究综述（20分）

介绍和分析每一篇论文对此问题的解决方法。

评分标准：

16~20分：所引用论文来自于国际权威会议或期刊，反应了该领域的研究脉络和最新研究成果；对论文的解读详尽而且清晰；

12~15分：所引用论文的数量和质量一般，基本能反应该领域的研究现状；对论文的解读较为准确；

8~11分：所引用论文的质量较差，未能深入了解相关领域的研究现状；论文解读可读性较差；

3、对比（10分）

对比这些解决方法的优缺点，并提出自己的想法。

评分标准：

8~10分：较为深刻地剖析了不同论文之间的差异和优劣，并能提出自己的看法，并加以论证。

6~7分：分析了不同论文之间的差异，提出了自己的见解，但是并没有进行更深入的论证。

4~5分：简单对比了不同论文的差异，未提出自己的见解。

4、参考文献（5分）

按照IEEE的参考文献标准。

评分标准：

5分：参考文献列举规范，引用正确；

3~4分：参考文献列举较为规范，但是没有正确引用；

1~2分：参考文献列举不够规范。

可供参考的会议：

PLDI——ACM SIGPLAN Conference on Programming Language Design and Implementation

PPoPP——Principles and Practice of Parallel Programming

SIGMOD——International Conference on Management of Data

OSDI——USENIX Symposium on Operating Systems Design and Implementation

ATC——Annual Technical Conference