# 大数据计算基础
# 实验指导书
# （必修）

**哈尔滨工业大学**

**2025 年 9 月**

# 目 录

# 实验要求

1. 实验分为实验一至实验四，四个部分。请按照本指导书先完成实验一，然后从实验二至实验四所列出的所有题目当中任选一题完成，即只需完成实验一以及实验二至四中的任意一个题目（10 选 1）。
2. 题目分为基础题目以及创新题目两类。基础题目的考核会更侧重完成程度，创新题目的考核会更侧重创新程度
3. 请在大作业完成后，撰写 **1 份整体报告**作为评分依据提交。实验报告应包括但不限于以下内容：
   a) 实验过程及步骤；
   b) 各项实验的结果；
   c) 实验遇到的问题及解决方案。
4. 实验中所使用的编程语言不限，软件平台版本不限（实验一中提及的软件版本仅作推荐）。
5. 请在完成实验及报告后，将各项实验的代码按照实验顺序组织好，并与实验报告一同压缩并以附件形式发送至邮箱 hit_tcs@126.com。（实验包命名格式为“大数据必修-学号-姓名”）
6. 提交实验截止日期，在群里通知。
7. 每位同学需独立完成，不得抄袭，一经发现核实，实验成绩记为 0 分。
8. 其他未尽事宜，可等待群内通知并咨询助教和任课教师。

# 实验1 大数据平台的搭建和调试

## 1.1 实验目的

本实验旨在通过亲手搭建与调试 Hadoop 与 Spark 两大核心平台，深入理解大数据生态系统的底层架构。核心目的是掌握从零开始部署、配置及验证一个完整分布式系统的工程方法，并融会贯通相关理论知识。

在 Hadoop 方面，实验重点在于搭建 HDFS 并调试其核心组件（NameNode、DataNode 等），通过 HDFS Shell 命令实操以验证分布式存储的可靠性。同时，通过运行 MapReduce 范例程序，调试并理解其分布式计算流程，验证集群计算能力。

在 Spark 方面，实验重点在于将其集成到 Hadoop YARN 资源管理器上，调试 Spark on YARN 的部署模式。通过对比 MapReduce 与 Spark 执行相同任务的性能差异，直观体验 Spark 内存计算和 DAG 执行引擎的优势，并掌握通过日志调试任务和优化参数的基本技能。

## 1.2 实验内容

（1） 搭建 Ubuntu 伪分布式计算环境：选择合适的虚拟机服务框架，下载指定版本的 Ubuntu 或者 Linux 发行版安装包；设置虚拟机占用资源，安装用于部署分布式计算平台的主从虚拟机；设置虚拟机之间的 SSH 链接，安装必要的编译环境。

（2） 搭建 Hadoop 以及 HDFS 分布式文件系统：下载并安装 Hadoop，配置其环境变量；配置 HDFS 和 YARN，启动并通过端口网页查看；使用上传文件验证 HDFS，使用样例程序验证 YARN。

（3） 搭建 Spark 分布式计算框架：下载并安装 Spark，配置其环境变量；启动 Spark 并通过端口网页查看；使用样例程序验证 Spark；利用 spark-submit 命令读写 HDFS 中的文件。

## 1.3 实验步骤

### 1.3.1 搭建伪分布式计算环境

#### 1.3.1.1下载安装虚拟机运行环境

下载并安装 Oracle VirtualBox，Vmware WorkStation 或者 Docker 作为虚拟机环境运行的基础。

#### 1.3.1.2安装 Linux 虚拟机

（1） 下载 Ubuntu 或其他 Linux 发行版本。

https://releases.ubuntu.com/22.04/ubuntu-22.04.5-desktop-amd64.iso.torrent

推荐 22.04 版本，否则系统资源占用太高

（2） 安装虚拟机。

虚拟机的配置应至少按如下表格进行。

| | CPU 核心 | 内存 | 硬盘 | 用户名 | 机器名 |
|---|---|---|---|---|---|
| Master | 2 | 4GB | 20 | dblab | master |
| Worker01 | 1 | 2GB | | | worker01 |
| Worker02 | 1 | 2GB | | | worker02 |

须确保用户名一致，机器的 CPU，内存等计算资源配置视个人电脑配置决定，如果计算资源过于紧张，可以跟助教商议申请减少节点。

（3） 确保两台物理机可以互相免密 SSH 登录。

提示：一些可能用到的命令如下

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install -y build-essential vim openssh-server net-tools

ssh-keygen -t rsa

ssh-copy-id master/worker01/worker02

修改/etc/hosts 文件，将三台机器的 IP 与机器名对应填入，并删除 127.0.1.1 的条目配置成功后应如下

```
master@master:~$ ssh worker01@worker01
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-84-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

worker01@worker01:~$ ssh worker02@worker02
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-84-generic x86_64)
```

（4） 在虚拟机上下载 JDK21，Scala2.13

https://www.oracle.com/cn/java/technologies/downloads/#java21

https://www.scala-lang.org/download/2.13.16.html

下载至在主文件夹下新建的 work 文件夹下，并解压，然后复制到/usr/lib 目录下

sudo mv jdk /usr/lib/jvm

sudo mv scala /usr/lib/scala

（5） 在虚拟机中的环境变量文件（/etc/profile）添加如下内容并重新加载（source /etc/profile）：

```
export JAVA_HOME=/usr/lib/jvm
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
export SCALA_HOME=/usr/lib/scala
export PATH=$PATH:$SCALA_HOME/bin
```

安装成功后应可按如下图所示的步骤验证配置成功。

```
worker01@worker01:~/work$ java -version
java version "1.8.0_461"
Java(TM) SE Runtime Environment (build 1.8.0_461-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.461-b11, mixed mode)
worker01@worker01:~/work$ scala -version
Scala code runner version 2.13.16 -- Copyright 2002-2025, LAMP/EPFL and Lightbend, Inc. dba Akka
```

## 1.3.2 搭建 Hadoop 以及 HDFS 分布式文件系统

### 1.3.2.1 下载 Hadoop

（1） 网址: https://hadoop.apache.org 版本 3.4.2

（2） 下载至在虚拟机的 work 目录下后解压缩，并重命名为 hadoop。

（3） 修改环境变量文件，添加如下内容并重新加载

export HADOOP_HOME=/home/dblab/work/hadoop

export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

### 1.3.2.2配置 HDFS 与 YARN

（1） 设置 JDK 安装目录（hadoop-env.sh）

（2） 指定 HDFS 主节点（core-site.xml,hdfs-site.xml）

（3） 配置 workers 文件(workers)

（4） 修改 yarn 配置文件(mapred-site.xml,yarn-site.xml)

（5） 复制配置到其他虚拟节点上

### 1.3.2.3 启动 HDFS 与 Yarn

（1） 在 master 服务器上格式化主节点

hadoop namenode -format

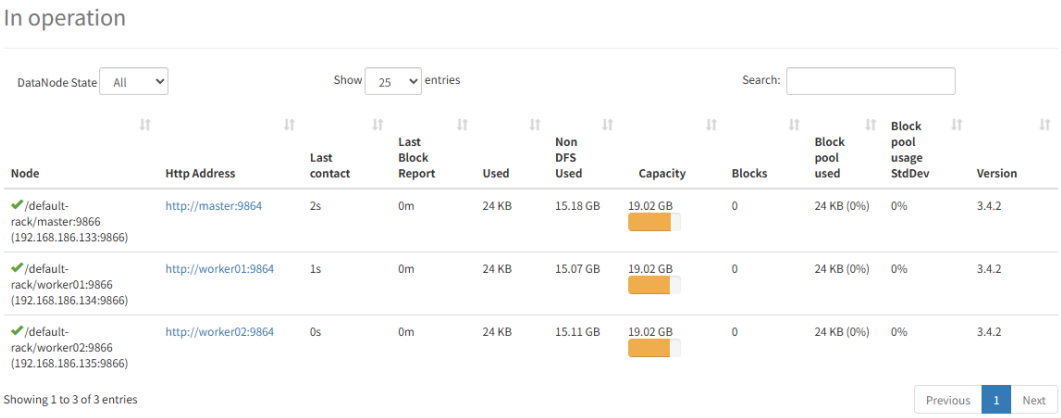
（2） 启动 HDFS 与 yarn

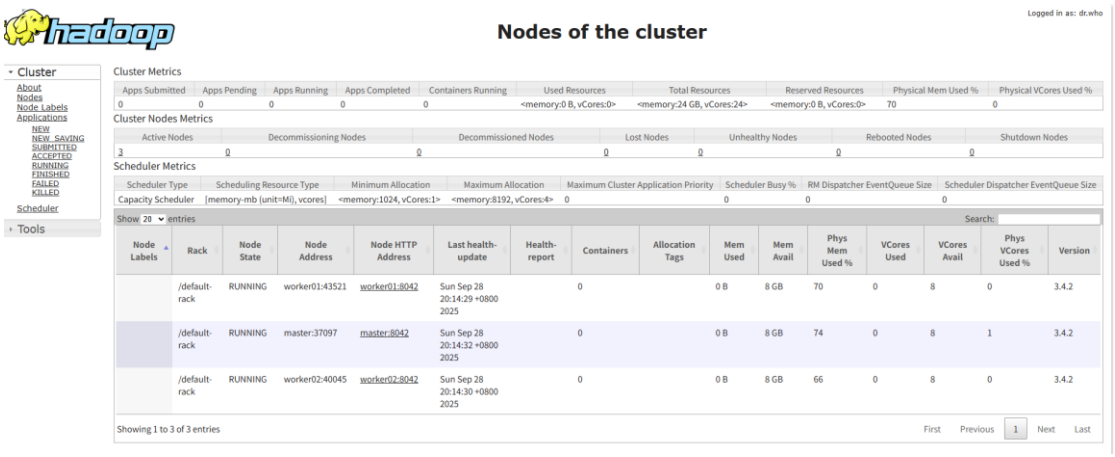
start-dfs.sh

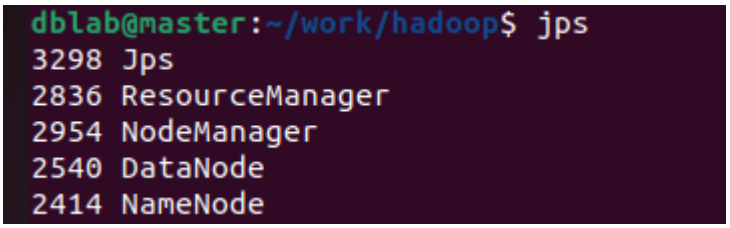
start-yarn.sh

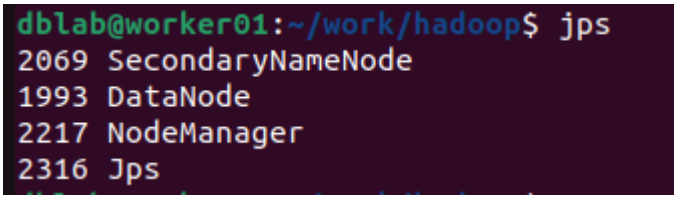
（3） 通过查看 WebUI 与运行进程验证启动成功

提示：HDFS 的 WebUI 端口号为 9870，Yarn 的 WebUI 端口号为 8088





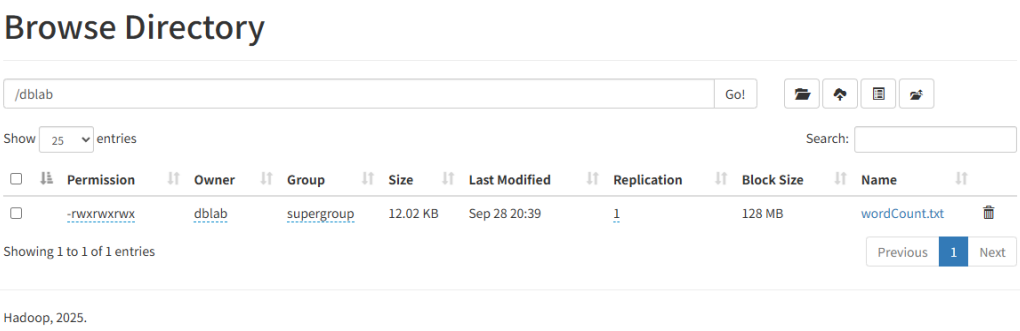

（4） 向 HDFS 上传文件验证配置成功

通过下述指令，将实验文件包中的 wordCount.txt 文件上传至 HDFS 中

hadoop fs -mkdir /dblab

hadoop fs -put /home/dblab/work/wordCount.txt /dblab/

WebUI 中应可以查看到如下




（5） 运行示例程序验证 Yarn 可正常运行

A．运行 distributed shell

hadoop jar $HADOOP_HOME/share/hadoop/yarn/hadoop-yarn-applications-distributedshell-3.4.2.jar \

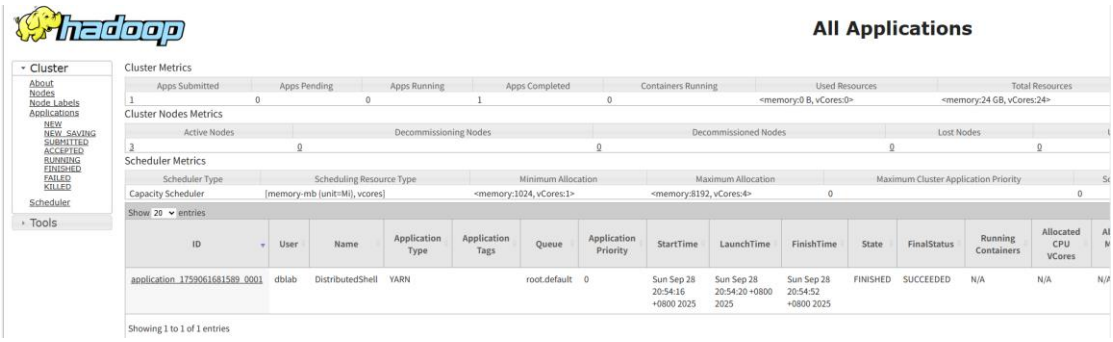
org.apache.hadoop.yarn.applications.distributedshell.Client \

--jar $HADOOP_HOME/share/hadoop/yarn/hadoop-yarn-applications-distributedshell-3.4.2.jar\

--shell_command "touch /home/dblab/work/hello-yarn" \

--num_containers 3 \

yarn 的 WebUI 及终端中运行结果如下图所示



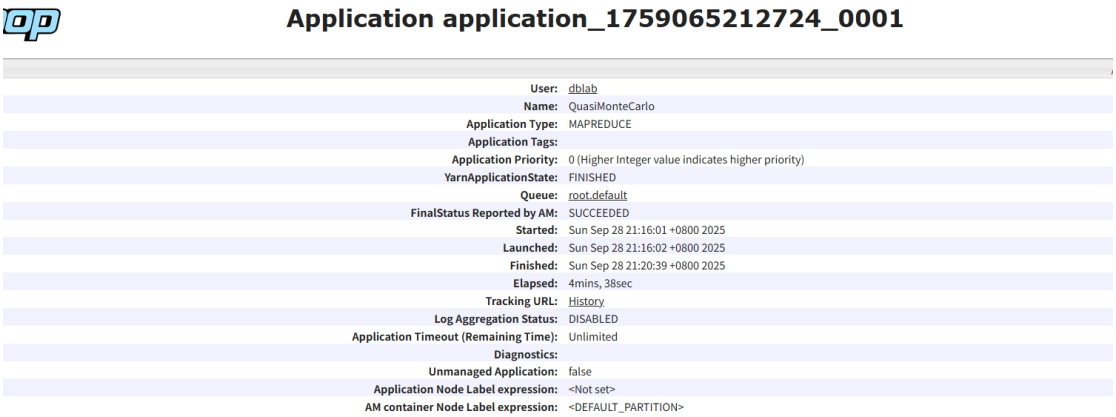

```
dblab@master:~/work$ ll hello-yarn
-rw-rw-r-- 1 dblab dblab 0  9月 28 20:54 hello-yarn
dblab@master:~/work$
```

B.运行

hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.2.jar pi 10 10

终端及 yarn 的 WebUI 中运行结果应如下显示：

```
        Map-Reduce Framework
                Map input records=10
                Map output records=20
                Map output bytes=180
                Map output materialized bytes=280
                Input split bytes=1440
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=280
                Reduce input records=20
                Reduce output records=0
                Spilled Records=40
                Shuffled Maps =10
                Failed Shuffles=0
                Merged Map outputs=10
                GC time elapsed (ms)=54007
                CPU time spent (ms)=71120
                Physical memory (bytes) snapshot=3163201536
                Virtual memory (bytes) snapshot=27907198976
                Total committed heap usage (bytes)=2508922880
                Peak Map Physical memory (bytes)=331710464
                Peak Map Virtual memory (bytes)=2553372672
                Peak Reduce Physical memory (bytes)=129310720
                Peak Reduce Virtual memory (bytes)=2508165120
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1180
        File Output Format Counters
                Bytes Written=97
Job Finished in 287.07 seconds
Estimated value of Pi is 3.20000000000000000000
dblab@master:~/work$
```



**Application application_1759065212724_0001**

| | |
|---|---|
| User: | dblab |
| Name: | QuasiMonteCarlo |
| Application Type: | MAPREDUCE |
| Application Tags: | |
| Application Priority: | 0 (Higher Integer value indicates higher priority) |
| YarnApplicationState: | FINISHED |
| Queue: | root.default |
| FinalStatus Reported by AM: | SUCCEEDED |
| Started: | Sun Sep 28 21:16:01 +0800 2025 |
| Launched: | Sun Sep 28 21:16:02 +0800 2025 |
| Finished: | Sun Sep 28 21:20:39 +0800 2025 |
| Elapsed: | 4mins, 38sec |
| Tracking URL: | History |
| Log Aggregation Status: | DISABLED |
| Application Timeout (Remaining Time): | Unlimited |
| Diagnostics: | |
| Unmanaged Application: | false |
| Application Node Label expression: | <Not set> |
| AM container Node Label expression: | <DEFAULT_PARTITION> |

## 1.3.3搭建 Spark 分布式计算框架

### 1.3.3.1下载 Spark

（1） 下载 Spark4.0.1

https://dlcdn.apache.org/spark/spark-4.0.1/spark-4.0.1-bin-without-hadoop.tgz

下载至 work 文件夹下，解压后重命名为 spark

（2） 修改环境变量文件，添加如下内容并重新加载

export SPARK_HOME=/home/dblab/work/spark

export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin

1.3.3.2 配置 Spark

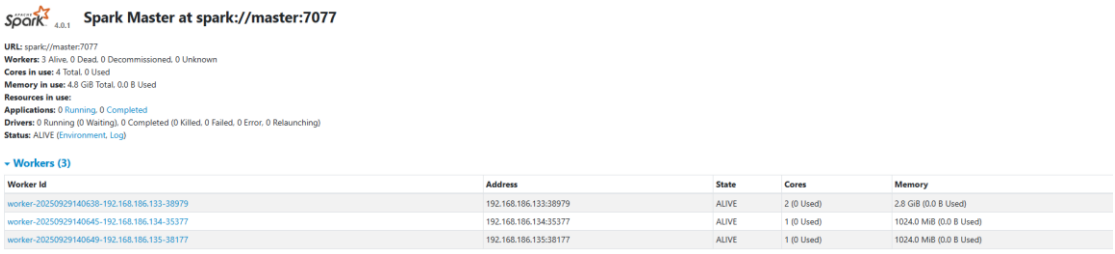
（1） 设置 Spark 运行的环境变量（spark-env.sh）

（2） 配置 Spark 默认运行配置（spark-default.conf）

（3） 设置 Spark 运行的 worker 服务器（workers）

1.3.3.3验证 Spark

（1） 启动 Spark 集群，进入 spark 目录下的 sbin 目录下，输入命令“./start-all.sh”启动 Spark 集群。结果如下。（Spark 的 WebUI 端口号是 8080）



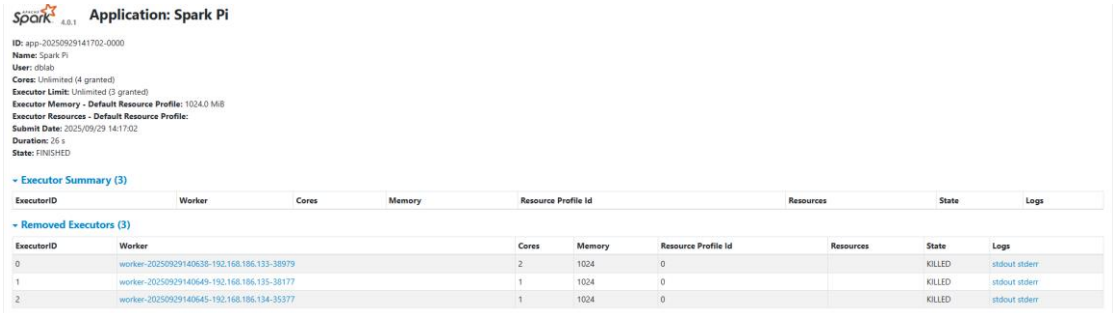


（2） 运行估计圆周率的实例程序

命令如下：

~/work/spark/bin/run-example org.apache.spark.examples.SparkPi

WebUI 及控制台输出的结果如下图所示。



```
dblab@master:~/work/spark/sbin$ ~/work/spark/bin/run-example org.apache.spark.examples.SparkPi
WARNING: Using incubator modules: jdk.incubator.vector
Pi is roughly 3.1443357216786083
dblab@master:~/work/spark/sbin$
```

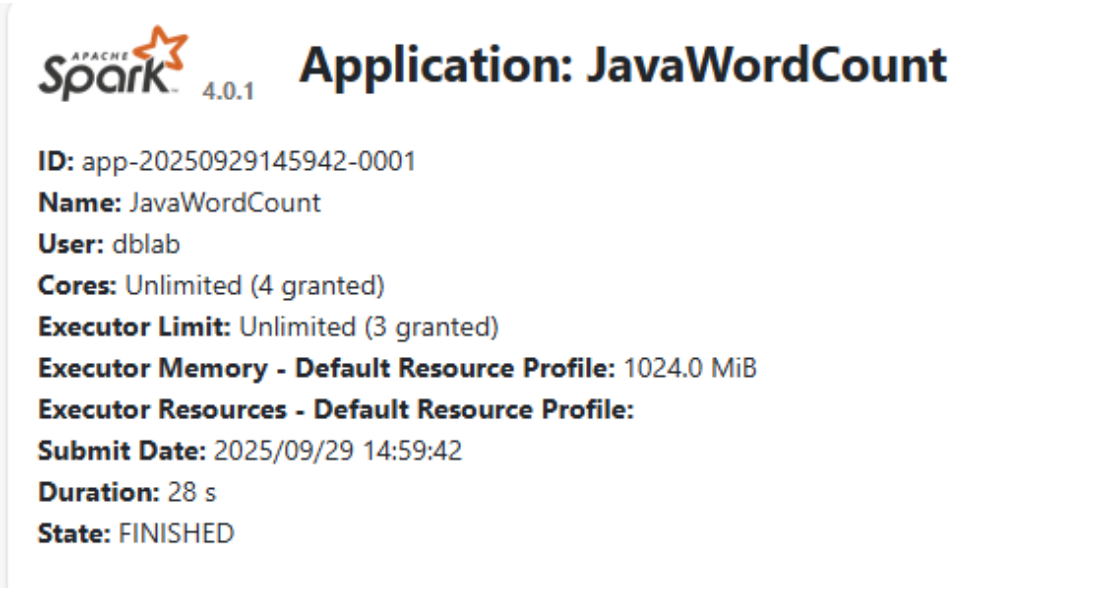
（3） 利用 spark-submit 命令读写 HDFS 中的文件

命令如下：

spark-submit --master spark://master:7077 --class org.apache.spark.examples.JavaWordCount ~/work/spark/examples/jars/spark-examples_2.13-4.0.1.jar hdfs://master:9000/dblab/wordCount.txt

终端及 WebUI 的输出如图所示。

```
dblab@master:~/work/spark/examples/src/main/scala/org/apache/spark/examples$ source /etc/profile
dblab@master:~/work/spark/examples/src/main/scala/org/apache/spark/examples$ spark-submit --master spark://master:7077 --class org.apache.spark.examples.JavaWordCount ~/work/spark/examples/jars/spark-exa
mples_2.13-4.0.1.jar hdfs://master:9000/dblab/wordCount.txt
WARNING: Using incubator modules: jdk.incubator.vector
reporting,: 1
cosmopolitan: 1
tears.: 1
anonymity,: 1
previous: 1
coach: 2
reality: 1
Uan: 1
order: 1
champion: 1
lockstep.: 1
behind: 1
nationali: 1
*And: 1
```



# 实验2 基于大数据平台的数据存储与访问优化

## 2.1 实验目的

本次实验聚焦于大数据平台的数据存储与访问优化，通过三角形计数、基于机器学习的图数据划分及大数据框架下的图数据划分三个核心问题，深入研究不同计算框架下的数据存储与访问优化策略。总体目标是掌握 MapReduce 和 Spark GraphX 两种框架的存储访问机制，理解图数据在分布式环境中的分区策略与存储布局，探究数据本地性对计算性能的影响机理。

学习通过使用 MapReduce 自定义分区器优化 Shuffle 过程的数据分布，减少网络传输开销；在 Spark GraphX 方面，理解如何利用内存计算、血缘关系和惰性求值优化执行计划，降低数据移动成本。同时，探索基于机器学习的智能图划分方法，研究如何通过预测数据访问模式优化存储布局。

## 2.2 实验内容

（1） 三角形计数与性能优化：研究 MapReduce 和 Spark GraphX 框架下的数据存储与访问优化。探究图数据划分策略对计算性能的影响，分析数据本地性、网络传输开销等关键因素，掌握分布式图计算性能优化方法。

（2） 基于机器学习的图划分：研究基于机器学习的智能图数据划分技术，利用图神经网络等先进方法优化数据分布策略。探索数据依赖关系对划分效果的影响，设计改进方案以提升划分质量，降低通信开销。

（3） 大数据框架下图划分评估：实现图数据划分算法，并通过 PageRank 和最短路径等典型图算法验证划分效果。重点分析不同上层应用对数据划分的需求特点，掌握图划分质量评估的实践方法。

## 2.3 实验题目

### 2.3.1基础题目

1. 三角形计数问题

在大图数据分析中，由于图模式的大小对相关计算问题的复杂性影响非常大，而一般的图匹配问题是 NP-难的，因此，利用尺寸较小的图模式分析结果构造一

般模式图的匹配结果是一个重要的手段。三角形是一类最为重要的小尺寸图模式，在图计算中，尤其是大图计算中，三角形分析是重要的组件。其中，三角形计数问题是一个经典问题，在数据挖掘，查询优化等研究中有着重要的应用。

(a) 请选择合适的大数据分布式计算平台，并利用虚拟机工具搭建大数据计算平台，支持后续的大图数据计算；

(b) 设计并实现至少 2 个三角形计数算法，并分析计数算法的理论性能；

(c) 利用所搭建的平台实现三角形计数算法，尝试通过调整计算平台的可调节参数优化算法的执行时间，利用实际的图数据（自行构建或者使用题目给定的数据）评测所实现的算法；

(d) 分布式平台中，图数据存储对应用层的算法是透明的，尝试探索算法运行中图数据的实际存储位置对算法性能的影响，并尝试利用所学的知识优化算法性能。

(e) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报 ppt。

(f) 提交：代码、数据（如自行构建）、实验报告、汇报 ppt

数据来源：http://snap.stanford.edu/data/com-DBLP.html

### 2.3.2创新题目

1. 基于 ML 的图数据划分问题

利用文献资料 ML 帮助图数据划分。图神经网络（GNN）是一类重要的神经网络模型，由于 GNN 在训练过程中要处理复杂的数据依赖，因此，在训练过程中需要数据管理的相关技术来解决这些数据依赖问题，尤其是图数据划分问题。请你参考下面这篇文献，复现其中的图数据划分技术。

(a) 搭建实验平台，支持后续的算法运行和测试；

(b) 复现参考文献的算法；

(c) 选择两个不同于原论文的数据集，并至少完成 5.3.1 节的实验；

(d) 尝试发现划分结果的不足，并设计、实现改进方法，用实验验证；

(e) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报 ppt。

(f) 提交：代码、数据、实验报告、汇报 ppt

参考文献：

[vldb24] Comprehensive Evaluation of GNN Training Systems: A Data Management Perspective

2. 大数据框架下的图数据划分问题

图数据划分在图数据预处理过程中的重要一环，一个好的划分会显著降低集群处理指定计算任务时内部的通讯代价。请你结合以下参考文献，在大数据计算框架（Hadoop 或者 Spark）下实现一个图数据划分算法，并使用 PageRank 算法以及最短路径算法来检验你实现的划分算法的有效性。

(a) 请选择合适的大数据分布式计算平台，并利用虚拟机工具搭建大数据计算平台，支持后续的算法运行和测试；

(b) 复现并有效运行参考文献的算法；

(c) 选择一个不同于原论文的数据集，并使用 PageRank 算法以及最短路径算法来检验你实现的划分算法的有效性；

(d) 对比分析不同上层应用算法对划分的需求，尝试改进划分方法，并用实验验证你的方法；

(e) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报 ppt。

(f) 提交：代码、数据、实验报告、汇报 ppt

参考文献：

[vldb24] FSM: A Fine-grained Spliting and Merging Framework for Dual-balanced Graph Partition

# 实验3 基于大数据平台的数据分析方法设计与实现

## 3.1 实验目的

本实验旨在通过 Spark SQL 和 Spark Streaming 两大核心组件，培养学生构建完整数据分析解决方案的能力。实验将从数据统计分析、不完整数据填充和连接操作三个典型场景出发，系统掌握大数据分析流程的设计与实现方法。

深入理解 Spark SQL 处理结构化数据的原理，掌握 DataFrame API 和 Spark SQL 的使用方法；同时理解 Spark Streaming 的微批次处理模型，掌握流式数据处理的基本概念。

使用 Spark SQL 完成复杂的数据统计分析，包括多维度聚合、分组计算和排序等操作；处理数据质量问题，实现不完整数据的识别与智能填充；优化连接操作性能，研究不同连接策略对执行效率的影响。在实时分析方面，设计并实现流式数据处理管道，完成实时统计、窗口计算等任务。

## 3.2 实验内容

（1） 人口大数据统计分析与存储优化：研究数据的分布式存储策略，使用工具对数据的多维度统计分析。探索数据聚类算法在大规模数据集上的应用，研究数据存储格式和平台参数对分析性能的影响。

（2） 不完整数据填充方法：研究基于深度学习的不完整数据填充算法。复现并测试先进的数据填充方法，探索不同填充策略对后续数据挖掘任务的影响。评估填充算法在提升数据质量方面的效果。

（3） 大数据连接操作性能优化：实现不少于 3 种不同的 Join 算法。通过测试对比分析各类连接算法在时间复杂度、资源消耗等方面的性能差异。探索新型分区连接算法在大规模数据场景下的优势。

## 3.3 实验题目

### 3.3.1基础题目

1. 人口大数据统计分析问题

利用统计数据分析整体情况是大数据分析范式的重要应用。美国人口普查局从 1994 年到 1995 年进行了一次人口普查，获取了人口大数据，现在需要设计相关方法参与管理、统计和分析这些数据。这份数据包含 41 个人口统计和就业相关变量。详细信息请参考：数据源：https://archive.ics.uci.edu/dataset/117/census+income+kdd

(a) 请选择合适的大数据分布式计算平台，并利用虚拟机工具搭建大数据计算平台，支持后续的大数据计算；

(b) 设计人口统计数据的存储策略，并利用所搭建的平台存储数据；

(c) 搭建 SQL 工具（如 HIVE 等）支持统计如下信息：TOP 10 职业编码（occupation code）、调查人群的教育程度分布（education AHGA）、统计调查人群的孩子情况(family members under 18 PARENT )，并用图表工具可视化上述分析结果；

(d) 调研并设计人口大数据的聚簇算法，并利用所搭建的平台实现算法，可视化并尝试解释聚簇结果；

(e) 如果数据规模特别大，达到 10GB、100GB 甚至 1TB，尝试探索数据的存储形式是否可以改进？对统计、分析算法的影响如何？尝试通过调整计算平台的可调节参数优化所设计的算法执行时间，尝试利用所学的大数据计算平台存储相关的知识优化算法性能。

(f) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报 ppt。

(g) 提交：代码、数据（如自行构建）、实验报告、汇报 ppt

### 3.3.2创新题目

1. 不完整数据填充问题

填充不完整数据是数据质量问题中一个重要的研究方向。不完整数据可以定义为数据集中不完整或缺失的数据部分。数据集中缺失值的存在会降低数据质量，削弱数据分析能力，并在数据科学应用中引发偏差。因此，处理不完整信息对于

大多数数据挖掘和机器学习技术至关重要。从下面两个参考文献及其提及的基于深度学习的不完整数据填充算法中选择一个实现，并在至少两个不同于算法原论文使用的公开数据集上测试算法。

(a) 搭建虚拟机实验平台，支持后续的算法运行和测试；
(b) 下面两个参考文献及其提及的基于深度学习的不完整数据填充算法中选择一个复现；
(c) 选择两个不同于原论文的数据集，并至少完成算法准确性对比实验；
(d) 尝试探索填充方法对后续可能的预测或分类等任务的影响，设计、实现并验证你的想法；
(e) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报 ppt。
(f) 提交：代码、数据、实验报告、汇报 ppt

参考文献：

Missing value imputation designs and methods of nature-inspired metaheuristic techniques: A systematic review

Deep imputation of missing values in time series health data: A review with benchmarking

What's a good imputation to predict with missing values?

Long-term missing value imputation for time series data using deep neural networks

Missing value imputation techniques: A survey

2. 连接操作问题

连接（Join）操作是数据库中的一项重要基本操作。其平方时间的复杂度，在大数据计算任务中，通常是很难接受的。下面这篇文献提出了一种新型的 Join 算法。请你在熟读文献后，了解 Join 算法的技术难点，并自主查找其他相关文献，在大数据计算平台下实现不少于 3 种不同的 Join 算法，并测试比较它们各自的优缺点。**注意：你最终的得分将会与所实现算法的难易程度以及数量相关**

(a) 请选择并部署合适的大数据计算平台，支持后续的算法运行和测试；
(b) 实现不少于 3 种不同的 Join 算法
(c) 选择至少两个公开数据集，并测试比较它们各自的优缺点
(d) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报 ppt。
(e) 提交：代码、数据（如自行构建）、实验报告、汇报 ppt

参考文献：

[sigmod23] NOCAP: Near-Optimal Correlation-Aware Partitioning Joins

# 实验4 基于大数据平台的数据分析方法优化

## 4.1 实验目的

本实验旨在通过 GraphX 图计算框架与 MLlib 机器学习库，深入探索大数据分析方法的优化策略。实验将围绕数据预测、聚类分析、模型数据遗忘和时间序列变点检测四个核心任务，系统研究分布式计算环境下数据分析算法的性能优化方法。

深入理解图计算模型与机器学习算法在大数据平台上的运行机制，掌握 GraphX 的图数据存储结构与计算模式，以及 MLlib 的分布式机器学习算法实现原理。实验将重点研究：基于 GraphX 的图神经网络模型优化与预测任务实现；MLlib 聚类算法在大规模数据集上的参数调优与性能优化；针对模型数据遗忘问题的增量学习与模型更新策略；时间序列数据的分布式变点检测算法实现与优化。

## 4.2 实验内容

（1） 预测性分析模型构建与优化：基于 Spark 平台实现森林火灾面积预测任务。研究多种机器学习算法的实现与比较，通过参数调优和分布式计算优化提升模型性能。

（2） 聚类分析与降维技术应用：运用 PCA 等降维技术对高维特征进行处理，采用合适的距离度量函数实现数据聚类。探索数据内在结构，为后续建模提供有效特征表示。研究数据规模扩展对聚类效果的影响。

（3） 数据遗忘机制与模型更新策略：研究机器学习中的数据遗忘问题，实现特定数据的模型遗忘机制。通过复现先进的数据遗忘算法，探究数据移除对模型在下游任务中表现的影响，优化模型更新策略。

（4） 时间序列分析与变点检测：通过复现深度学习等先进检测方法，完成对非平稳时间序列的分析，结合可视化技术展示状态划分结果，掌握时序数据中关键变化点的检测与分析方法。

## 4.3 实验题目

### 4.3.1基础题目

1. 森林火灾预测任务

数据驱动的分析是大数据的重要应用。考虑如下的数据应用实例：利用气象和其他数据来预测森林火灾的烧毁面积。该数据集包含变量：空间、时间、FWI成分和天气属性。使用预测模型根据上述变量预测火灾面积。

(a) 请选择合适的大数据分布式计算平台，并利用虚拟机工具搭建分布式计算平台，支持后续的预测分析计算；

(b) 利用大数据平台的统计分析工具或者实现统计分析算法，统计在数据集中各个变量的值域以及范围分布，作出分布直方图；利用机器学习算法，设计并实现火灾面积的预测算法；

(c) 利用所搭建的平台实现统计分析和预测算法，利用给定的数据资源评测所实现的算法，并调整计算平台的参数来优化算法的训练或者推理时间性能；

(d) 在分布式平台上实现更多的基于机器学习的预测算法，调成平台参数来观察对不同算法的影响，比较不同算法的时空间效率；

(e) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报ppt。

(f) 提交：代码、数据（如自行构建）、实验报告、汇报 ppt

数据来源：https://archive.ics.uci.edu/dataset/162/forest+fires

2.评估肥胖程度

利用大数据进行相关的生命健康分析研究是大数据的一个非常重要的应用场景。根据个人的生活习惯、饮食习惯和身体状况，用于评估其肥胖程度。数据来源：https://archive.ics.uci.edu/dataset/544/estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition

(a) 并利用虚拟机工具搭建 spark 分布式计算平台，支持后续的分析计算；

(b) 利用 Spark 使用 PCA(主成分分析)对特征进行降维；

(c) 选择适当的距离度量函数，使用 Spark 进行聚类；

(d) 使用 Spark 的 MLlib 中的机器学习方法，建立预测肥胖程度的模型，并进行模型检验并给出模型预测的正确率；

(e) 分布式平台中，复制数据以增大数据规模。利用多计算节点优化预测算法的时间效率，尝试探索所搭建平台下预测算法的加速方法，并尝试实现并给出评测；

(f) Spark 平台分析数据通常是存储在内存中的，尝试探索 Spark 加载数据的方式，并测试和对比数据存储位置对上述算法的影响；

(g) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报 ppt。

(h) 提交：代码、数据（如自行构建）、实验报告、汇报 ppt

### 4.3.2创新题目

1. 数据遗忘问题

随着 AI 技术的不断发展，机器学习过程中的过拟合现象受到越来越多的重视。一方面，过拟合会影响 AI 的准确性，另一方面，可以通过获得每批次的梯度信息来逆向还原原始数据信息，因此，我们需要引入机器学习模型训练过程中的数据遗忘（Machine Unlearning）的概念，具体指针对于特定的数据而言的，让模型遗忘掉某个数据之后模型的表现能力，应该等同于这个数据没有参与模型训练时模型的表现能力。现有很多研究将机器学习方法引入数据库处理复杂的计算任务。由于数据库自身高动态变化的特性，数据遗忘的方法也被引入处理数据库的任务。请你阅读以下参考文献，复现这篇论文中，进行的数据遗忘方法对数据库任务的影响的实验测试。（任选其中一个下游数据库任务即可，如完成多个，可适当加分，满分封顶）

(a) 搭建实验平台，支持后续的算法运行和测试；

(b) 复现并评测文献中的算法；

(c) 选择一个不同于原论文的数据集，实现原论文 7.2.1 节中任意一个下游数据库任务的实验；

(d) 尝试优化所实现的算法，并用实际实验验证你的想法；

(e) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报 ppt。

(f) 提交：代码、数据、实验报告、汇报 ppt

参考文献：

[sigmod24] Machine Unlearning in Learned Databases: An Experimental Analysis

2. 时间序列变点检测问题

时间序列变点检测是数据管理领域的一个重要研究方向。这一任务的目标是从大量数据中提取主要指标，来表示整个系统的状态机器变化。请你阅读以下文献后，选择并复现一个典型的变点检测算法。

(a) 搭建实验环境（不一定是分布式平台），支持后续的项目开发；

(b) 实现时间序列变点检测的算法；

(c) 选择若干合适的数据集，测试变点检测算法；

(d) 构造可视化结果，分析数据状态划分和变点检测结果；

(e) 结合论文和实验结果，讲述该算法的主要原理；

(f) 针对上述结果撰写实验报告（字数不限），并制作 15 页左右的结题汇报 ppt。

(g) 提交：代码、数据、实验报告、汇报 ppt

参考文献：

1、Detecting change points in time series of inSAR persistent scatterers using deep learning models

2、Quantifying input data drift in medical machine learning models by detecting change-points in time-series data

3、Change points detection for nonstationary multivariate time series

4、Deep learning for change detection in remote sensing: a review