## 实验一：Hadoop环境搭建和熟悉常用的Linux操作

### 一、实验目的

本实验旨在学习在虚拟环境下搭建Hadoop集群系统，熟悉常用的Linux操作和Hadoop操作，为顺利开展后续其他实验奠定基础。

### 二、实验平台

* 操作系统：Linux（建议Ubuntu16.04）；
* Hadoop版本：2.7.1。

### 三、实验步骤

**（一）熟悉常用的Linux操作**

* cd命令：切换目录

1. 切换到目录“/usr/local”
2. 切换到当前目录的上一级目录
3. 切换到当前登录Linux系统的用户的自己的主文件夹

* ls命令：查看文件与目录

（4）查看目录“/usr”下的所有文件和目录

* mkdir命令：新建目录

（5）进入“/tmp”目录，创建一个名为“a”的目录，并查看“/tmp”目录下已经存在哪些目录

（6）进入“/tmp”目录，创建目录“a1/a2/a3/a4”

* rmdir命令：删除空的目录

（7）将上面创建的目录a（在“/tmp”目录下面）删除

（8）删除上面创建的目录“a1/a2/a3/a4” （在“/tmp”目录下面），然后查看“/tmp”目录下面存在哪些目录

* cp命令：复制文件或目录

（9）将当前用户的主文件夹下的文件.bashrc复制到目录“/usr”下，并重命名为bashrc1

（10）在目录“/tmp”下新建目录test，再把这个目录复制到“/usr”目录下

* mv命令：移动文件与目录，或更名

（11）将“/usr”目录下的文件bashrc1移动到“/usr/test”目录下

（12）将“/usr”目录下的test目录重命名为test2

* rm命令：移除文件或目录

（13）将“/usr/test2”目录下的bashrc1文件删除

（14）将“/usr”目录下的test2目录删除

* cat命令：查看文件内容

（15）查看当前用户主文件夹下的.bashrc文件内容

* tac命令：反向查看文件内容

（16）反向查看当前用户主文件夹下的.bashrc文件的内容

* more命令：一页一页翻动查看

（17）翻页查看当前用户主文件夹下的.bashrc文件的内容

* head命令：取出前面几行

（18）查看当前用户主文件夹下.bashrc文件内容前20行

（19）查看当前用户主文件夹下.bashrc文件内容，后面50行不显示，只显示前面几行

* tail命令：取出后面几行

（20）查看当前用户主文件夹下.bashrc文件内容最后20行

（21）查看当前用户主文件夹下.bashrc文件内容，并且只列出50行以后的数据

* touch命令：修改文件时间或创建新文件

（22）在“/tmp”目录下创建一个空文件hello，并查看文件时间

（23）修改hello文件，将文件时间整为5天前

* chown命令：修改文件所有者权限

（24）将hello文件所有者改为root帐号，并查看属性

* find命令：文件查找

（25）找出主文件夹下文件名为.bashrc的文件

* tar命令：压缩命令

（26）在根目录“/”下新建文件夹test，然后在根目录“/”下打包成test.tar.gz

（27）把上面的test.tar.gz压缩包，解压缩到“/tmp”目录

* grep命令：查找字符串

（28）从“～/.bashrc”文件中查找字符串'examples'

* 配置环境变量

（29）请在“～/.bashrc”中设置，配置Java环境变量

（30）查看JAVA\_HOME变量的值

**（二）熟悉常用的Hadoop操作**

（31）使用hadoop用户登录Linux系统，启动Hadoop（Hadoop的安装目录为“/usr/local/hadoop”），为hadoop用户在HDFS中创建用户目录“/user/hadoop”

（32）接着在HDFS的目录“/user/hadoop”下，创建test文件夹，并查看文件列表

（33）将Linux系统本地的“～/.bashrc”文件上传到HDFS的test文件夹中，并查看test

（34）将HDFS文件夹test复制到Linux系统本地文件系统的“/usr/local/hadoop”目录下

### 四、实验报告

详见实验报告模板

## 实验二：MapReduce初级编程实践

### 一、实验目的

* 通过实验掌握基本的MapReduce编程方法；
* 掌握用MapReduce解决一些常见的数据处理问题，包括数据去重、数据排序和数据挖掘等。

### 二、实验平台

* 操作系统：Linux（建议Ubuntu16.04）
* Hadoop版本：2.7.1

### 三、实验步骤

**（一）编程实现文件合并和去重操作**

对于两个输入文件，即文件A和文件B，请编写MapReduce程序，对两个文件进行合并，并剔除其中重复的内容，得到一个新的输出文件C。下面是输入文件和输出文件的一个样例供参考。

输入文件A的样例如下：

|  |
| --- |
| 20170101 x  20170102 y  20170103 x  20170104 y  20170105 z  20170106 x |

输入文件B的样例如下：

|  |
| --- |
| 20170101 y  20170102 y  20170103 x  20170104 z  20170105 y |

根据输入文件A和B合并得到的输出文件C的样例如下：

|  |
| --- |
| 20170101 x  20170101 y  20170102 y  20170103 x  20170104 y  20170104 z  20170105 y  20170105 z  20170106 x |

**（二）编写程序实现对输入文件的排序**

现在有多个输入文件，每个文件中的每行内容均为一个整数。要求读取所有文件中的整数，进行升序排序后，输出到一个新的文件中，输出的数据格式为每行两个整数，第一个数字为第二个整数的排序位次，第二个整数为原待排列的整数。下面是输入文件和输出文件的一个样例供参考。

输入文件1的样例如下：

|  |
| --- |
| 33  37  12  40 |

输入文件2的样例如下：

|  |
| --- |
| 4  16  39  5 |

输入文件3的样例如下：

|  |
| --- |
| 1  45  25 |

根据输入文件1、2和3得到的输出文件如下：

|  |
| --- |
| 1 1  2 4  3 5  4 12  5 16  6 25  7 33  8 37  9 39  10 40  11 45 |

**（三）对给定的表格进行信息挖掘**

下面给出一个child-parent的表格，要求挖掘其中的父子辈关系，给出祖孙辈关系的表格。

输入文件内容如下：

|  |
| --- |
| child parent  Steven Lucy  Steven Jack  Jone Lucy  Jone Jack  Lucy Mary  Lucy Frank  Jack Alice  Jack Jesse  David Alice  David Jesse  Philip David  Philip Alma  Mark David  Mark Alma |

输出文件内容如下：

|  |
| --- |
| grandchild grandparent  Steven Alice  Steven Jesse  Jone Alice  Jone Jesse  Steven Mary  Steven Frank  Jone Mary  Jone Frank  Philip Alice  Philip Jesse  Mark Alice  Mark Jesse |

### 四、实验报告

## 实验三：熟悉常用的HBase操作

### 一、实验目的

* + - * 理解HBase在Hadoop体系结构中的角色；
      * 熟练使用HBase操作常用的Shell命令；
      * 熟悉HBase操作常用的Java API。

### 二、实验平台

* + - * 操作系统：Linux（建议Ubuntu16.04）；
      * Hadoop版本：2.7.1；
      * HBase版本：1.1.5；
      * JDK版本：1.7或以上版本；
      * Java IDE：Eclipse。

### 三、实验步骤

**（一）编程实现以下指定功能，并用Hadoop提供的HBase Shell命令完成相同任务：**

1. 列出HBase所有的表的相关信息，例如表名；
2. 在终端打印出指定的表的所有记录数据；
3. 向已经创建好的表添加和删除指定的列族或列；
4. 清空指定的表的所有记录数据；
5. 统计表的行数。

**（二）HBase数据库操作**

1. 现有以下关系型数据库中的表和数据，要求将其转换为适合于HBase存储的表并插入数据：

学生表（Student）

|  |  |  |  |
| --- | --- | --- | --- |
| 学号（S\_No） | 姓名（S\_Name） | 性别（S\_Sex） | 年龄（S\_Age） |
| 2015001 | Zhangsan | male | 23 |
| 2015003 | Mary | female | 22 |
| 2015003 | Lisi | male | 24 |

课程表（Course）

|  |  |  |
| --- | --- | --- |
| 课程号（C\_No） | 课程名（C\_Name） | 学分（C\_Credit） |
| 123001 | Math | 2.0 |
| 123002 | Computer Science | 5.0 |
| 123003 | English | 3.0 |

选课表（SC）

|  |  |  |
| --- | --- | --- |
| 学号（SC\_Sno） | 课程号（SC\_Cno） | 成绩（SC\_Score） |
| 2015001 | 123001 | 86 |
| 2015001 | 123003 | 69 |
| 2015002 | 123002 | 77 |
| 2015002 | 123003 | 99 |
| 2015003 | 123001 | 98 |
| 2015003 | 123002 | 95 |

2. 请编程实现以下功能：

（1）createTable(String tableName, String[] fields)

创建表，参数tableName为表的名称，字符串数组fields为存储记录各个字段名称的数组。要求当HBase已经存在名为tableName的表的时候，先删除原有的表，然后再创建新的表。

（2）addRecord(String tableName, String row, String[] fields, String[] values)

向表tableName、行row（用S\_Name表示）和字符串数组fields指定的单元格中添加对应的数据values。其中，fields中每个元素如果对应的列族下还有相应的列限定符的话，用“columnFamily:column”表示。例如，同时向“Math”、“Computer Science”、“English”三列添加成绩时，字符串数组fields为{“Score:Math”, ”Score:Computer Science”, ”Score:English”}，数组values存储这三门课的成绩。

（3）scanColumn(String tableName, String column)

浏览表tableName某一列的数据，如果某一行记录中该列数据不存在，则返回null。要求当参数column为某一列族名称时，如果底下有若干个列限定符，则要列出每个列限定符代表的列的数据；当参数column为某一列具体名称（例如“Score:Math”）时，只需要列出该列的数据。

（4）modifyData(String tableName, String row, String column)

修改表tableName，行row（可以用学生姓名S\_Name表示），列column指定的单元格的数据。

（5）deleteRow(String tableName, String row)

删除表tableName中row指定的行的记录。

### 四、实验报告

## 实验四：RDD编程初级实践

### 一、实验目的

* + - * 熟悉Spark的RDD基本操作及键值对操作；
      * 熟悉使用RDD编程解决实际具体问题的方法。

### 二、实验平台

* + - * 操作系统：Ubuntu16.04
      * Spark版本：2.1.0

### 三、实验内容和要求

**1．spark-shell交互式编程**

请在班级群里下载Data01.txt，该数据集包含了某大学计算机系的成绩，数据格式如下所示：



请根据给定的实验数据，在spark-shell中通过编程来计算以下内容：

（1）该系总共有多少学生；

（2）该系共开设来多少门课程；

（3）Tom同学的总成绩平均分是多少；

（4）求每名同学的选修的课程门数；

（5）该系DataBase课程共有多少人选修；

（6）各门课程的平均分是多少；

（7）使用累加器计算共有多少人选了DataBase这门课。

**2. 编写独立应用程序实现数据去重**

对于两个输入文件A和B，编写Spark独立应用程序，对两个文件进行合并，并剔除其中重复的内容，得到一个新文件C。下面是输入文件和输出文件的一个样例，供参考。

输入文件A的样例如下：

20170101 x

20170102 y

20170103 x

20170104 y

20170105 z

20170106 z

输入文件B的样例如下：

20170101 y

20170102 y

20170103 x

20170104 z

20170105 y

根据输入的文件A和B合并得到的输出文件C的样例如下：

20170101 x

20170101 y

20170102 y

20170103 x

20170104 y

20170104 z

20170105 y

20170105 z

20170106 z

**3. 编写独立应用程序实现求平均值问题**

每个输入文件表示班级学生某个学科的成绩，每行内容由两个字段组成，第一个是学生名字，第二个是学生的成绩；编写Spark独立应用程序求出所有学生的平均成绩，并输出到一个新文件中。下面是输入文件和输出文件的一个样例，供参考。

Algorithm成绩：

小明 92

小红 87

小新 82

小丽 90

Database成绩：

小明 95

小红 81

小新 89

小丽 85

Python成绩：

小明 82

小红 83

小新 94

小丽 91

平均成绩如下：

(小红,83.67)

(小新,88.33)

(小明,89.67)

(小丽,88.67)

### 四、实验报告