大数据处理综合实验3 Hive表的join操作

组号：15　　**组长**：191220029 傅小龙　　**组员**：191220005 陈诚 191220014 陈元旻

一、提交及运行说明

提交的文件目录结构如下:

```
MPLab3-15-傅小龙.zip
├── MPLab3-15-傅小龙.pdf                实验报告 本文档
├── lab3.jar                            JAR包
├── readme.txt                          JAR包执行方式说明
└── src                                 源码目录
    ├── UniversityCountryJoiner.java    任务1源码
    └── CourseUniversityJoiner.java     任务2源码
```

本次实验的两个任务我们打包成了一个JAR包提交，通过指定主类的方式来完成对应任务。JAR包执行方式如下:

```
任务1: hadoop jar lab3.jar UniversityCountryJoiner input_path1 output_dir1
任务2: hadoop jar lab3.jar CourseUniversityJoiner input_path2 output_dir2
```

其中input_path是输入文件所在路径，output_path是输出目录，该目录要求在运行前不存在，由程序自动创建。

二、设计思路与功能实现

任务1

设计思路：在Map阶段中将文件中每一行数据打上标记，标注其来自University表还是Country表，在Reduce阶段进行join操作。考虑设计新的数据类型UniversityCountry进行数据封装，在Map阶段，来自University的行n_name项设为空串，来自Country表的行除n_name外均为空串，从而在传输数据的同时完成标记任务，类成员均为输出阶段要求的列。

```
private static class UniversityCountry implements
WritableComparable<UniversityCountry> {
    private String u_key = "";
    private String u_name = "";
    private String u_webpage = "";
    private String n_name = "";
}
```

首先实现抽象类WritableComparable的write和readFields接口，由于数据均为UTF8字符，选择使用writeUTF和readUTF实现。

```
@Override
public void write(DataOutput dataOutput) throws IOException {
    dataOutput.writeUTF(u_key);
    dataOutput.writeUTF(u_name);
    dataOutput.writeUTF(u_webpage);
```

```
    dataOutput.writeUTF(n_name);
} // in class UniversityCountry

@Override
public void readFields(DataInput dataInput) throws IOException {
    u_key = dataInput.readUTF();
    u_name = dataInput.readUTF();
    u_webpage = dataInput.readUTF();
    n_name = dataInput.readUTF();
}
```

同时，UniversityCountry类还向外提供一些接口以供使用，例如setCountry，setUniversity等处理文件输入的行的函数。

```
public String setCountry(String country) { // Map阶段处理country.tbl
    String s[] = country.split("\\|"); // 按字符'|'分割字符串
    u_key = "";       // 清空u_country
    u_name = "";      // 清空u_name
    u_webpage = "";   // 清空u_webpage
    n_name = s[1];    // 设置n_name
    return s[0];      // 返回n_alpha-2-code
} // in class UniversityCountry

public String setUniversity(String university) { // Map阶段处理university.tbl
    String s[] = university.split("\\|"); // 按字符'|'分割字符串
    u_key = s[0];         // 设置u_country
    u_name = s[1];        // 设置u_name
    u_webpage = s[4];     // 设置u_webpage
    n_name = "";          // 清空n_name
    return s[2];          // 返回u_alpha-2-code
} // in class UniversityCountry

public void setN_name(String name) { // 设置n_name, 不清空其他数据(Reduce阶段调用)
    n_name = name;
} // in class UniversityCountry

public String getN_name() { // 获取n_name(Reduce阶段调用)
    return n_name;
} // in class UniversityCountry

@Override
public String toString() { // UniversityCountry类变量转字符串(Reduce阶段调用)
    return u_key + '|' + u_name + '|' + u_webpage + '|' + n_name;
}
```

在完成了新数据类型UniversityCountry后，进一步实现Mapper和Reducer，Map阶段的输入为<Object, Text>类型，输出为<Text, UniversityCountry>类型，其中输出的key中存放国家2字母编码，value中存放需要连接的其他数据，从而Reduce阶段的输入为<Text, UniversityCountry>类型，由于输出只需要一行字符串，因此选择<Text, NullWritable>类型，即将全部输出存放在key中。Mappper和Reducer的代码及注释如下：

```
private static class UniversityCountryJoinMapper extends Mapper<Object, Text,
Text, UniversityCountry> {
    private Text k = new Text();
    private UniversityCountry v = new UniversityCountry();
    private Boolean countryFlag = true;
```

```
    @Override
    protected void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
        FileSplit fileSplit = (FileSplit)context.getInputSplit();
        String fileName = fileSplit.getPath().getName();
        if (fileName.equals("university.tbl"))
            countryFlag = false; // 标记为来自university.tbl的数据
        else if (fileName.equals("country.tbl"))
            countryFlag = true; // 标记为来自country.tbl的数据
        else
            return ; // 来自其他文件的数据直接舍弃
        StringTokenizer itr = new StringTokenizer(value.toString(), "\n"); // 按
行分割数据
        while (itr.hasMoreTokens()) { // 对于每行数据均输出一个<k, v>对
            String line = itr.nextToken(); // 获取下一行数据
            if (countryFlag) // 数据来自university.tbl
                k.set(v.setCountry(line)); // setCountry的返回值为n_alpha-2-code
            else // 数据来自country.tbl
                k.set(v.setUniversity(line)); // setUniversity的返回值为u_alpha-2-
code
            context.write(k, v); // 输出<k, v>对
        } // end while
    } // end void map()
} // end class UniversityCountryJoinMapper

private static class UniversityCountryJoinReducer extends Reducer<Text,
UniversityCountry, Text, NullWritable> {
    String n_name;
    Text k = new Text();
    NullWritable v = NullWritable.get();

    @Override
    protected void reduce(Text key, Iterable<UniversityCountry> values, Context
context) throws java.io.IOException, java.lang.InterruptedException {
        Vector<UniversityCountry> UCList = new List<UniversityCountry>();
        for (UniversityCountry value : values) {
            UniversityCountry uc = new UniversityCountry(value);
            UCList.add(uc); // 将Iterator中的数据存储到List中 以供后续遍历修改
        }
        for (UniversityCountry uc : UCList) {
            if (!uc.getN_name().equals("")) { // 数据来自country.tbl(n_name不为空
值)
                n_name = uc.getN_name(); // 存储n_name, 以供join操作使用
                break;
            }
        }
        for (UniversityCountry uc : UCList) {
            if (uc.getN_name().equals("")) { // 数据来自university.tbl
                uc.setN_name(n_name); // 设置n_name
                k.set(uc.toString());
                context.write(k, v); // 输出<k, v>对
            } // end if
        } // end for
    } // end void reduce()
} // end class UniversityCountryJoinReducer
```

任务2

设计思路：在map阶段中，将university表复制100份，分别打上1到100的标签作为key，而course表随机一个1到100之间的数作为标签作为key发给reduce。则在reduce阶段，同一个key下包含着若干个标签为key的course表项，以及完整的university表，将其作笛卡尔积输出，最后得到的总表即为整个university表和course表的笛卡尔积。考虑设计新的数据类型CourseUniversity进行数据封装。num为每个表项的标签。

```
private static class CourseUniversity implements
WritableComparable<CourseUniversity> {
        private String c_key = "";
        private String c_name = "";
        private String c_subject = "";
        private String c_hours = "";
        private String u_key = "";
        private String u_name = "";
        private String u_webpage = "";
        private int num = 0;
```

由于在reduce阶段需要将iterable中的CourseUniversity分别作为Course和University的表项取出，还设计了CourseUniversity的构造函数和拷贝构造函数。

```
public CourseUniversity(){
    c_key = "";
    c_name = "";
    c_subject = "";
    c_hours = "";
    u_key = "";
    u_name = "";
    u_webpage = "";
    num = 0;
}//构造函数

public CourseUniversity(CourseUniversity uc){
    c_key = uc.c_key;
    c_name = uc.c_name;
    c_subject = uc.c_subject;
    c_hours = uc.c_hours;
    u_key = uc.u_key;
    u_name = uc.u_name;
    u_webpage = uc.u_webpage;
    num = uc.num;
}//拷贝构造函数
```

实现抽象类WritableComparable的write和readFields接口，由于数据均为UTF8字符，选择使用writeUTF和readUTF实现。

```
@Override
public void write(DataOutput dataOutput) throws IOException {
    dataOutput.writeUTF(c_key);
    dataOutput.writeUTF(c_name);
    dataOutput.writeUTF(c_subject);
```

```
    dataOutput.writeUTF(c_hours);
    dataOutput.writeUTF(u_key);
    dataOutput.writeUTF(u_name);
    dataOutput.writeUTF(u_webpage);
}

@Override
public void readFields(DataInput dataInput) throws IOException {
    c_key = dataInput.readUTF();
    c_name = dataInput.readUTF();
    c_subject = dataInput.readUTF();
    c_hours = dataInput.readUTF();
    u_key = dataInput.readUTF();
    u_name = dataInput.readUTF();
    u_webpage = dataInput.readUTF();
}
```

同时，CourseUniversity类还向外提供一些接口以供使用，例如setCourse，setUniversity等处理文件输入的行的函数。

```
public void setnum(int number){
    num = number;//设置当前表项的标签
}

public void setCourse(String course) {  // Map阶段处理course.tbl
    String s[] = course.split("\\|");    // 按字符'|'分割字符串
    c_key = s[0];                        // 设置c_key
    c_name = s[1];                       // 设置c_name
    c_subject = s[2];                    // 设置c_subject
    c_hours = s[3];                      // 设置c_hours
    u_key = "";                          // 清空u_key
    u_name = "";                         // 清空u_name
    u_webpage = "";                      // 清空u_webpage
}

public void setUniversity(String university) {  // Map阶段处理university.tbl
    String s[] = university.split("\\|");        // 按字符'|'分割字符串
    c_key = "";                                  // 清空c_key
    c_name = "";                                 // 清空c_name
    c_subject = "";                              // 清空c_subject
    c_hours = "";                                // 清空c_hours
    u_key =s[0];                                 // 设置u_key
    u_name = s[1];                               // 设置u_name
    u_webpage = s[4];                            // 设置u_Webpage
}

public String getckey(){// 获取c_key,Reduce阶段用来判断是university还是course表项
    return c_key;
}

@Override
public String toString(){// CourseUniversity类变量转字符串(Reduce阶段调用，用于输出)
    if (!c_key.equals(""))
        return c_key + "|" + c_name + "|" + c_subject + "|" +c_hours;
    else return u_key + "|" + u_name +"|" + u_webpage;
}
```

在完成了新数据类型CourseUniversity后，进一步实现Mapper和Reducer，Map阶段的输入为<Object, Text>类型，输出为<Text, CourseUniversity>类型，其中输出的key中每条表项的标签，value中存放需要连接的数据表项，从而Reduce阶段的输入为<Text, CourseUniversity>类型，由于输出只需要一行字符串，因此选择<Text, NullWritable>类型，即将全部输出存放在key中。Mappper和Reducer的代码及注释如下：

```
private static class CourseUniversityJoinMapper extends Mapper<Object, Text,
Text, CourseUniversity> {
        private Text k = new Text();
        private CourseUniversity v = new CourseUniversity();
        private Boolean courseFlag = true;
        @Override
        protected void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
            Random rd=new Random();
            FileSplit fileSplit = (FileSplit)context.getInputSplit();
            String fileName = fileSplit.getPath().getName();
            if(fileName.equals("university.tbl"))
                courseFlag = false;  // 标记为来自university.tbl的数据
            else if(fileName.equals("course.tbl"))
                courseFlag = true;   // 标记为来自course.tbl的数据
            else
                return ;             // 来自其他文件的数据直接舍弃
            StringTokenizer itr = new StringTokenizer(value.toString(), "\n");
            //  按行分割数据
            while (itr.hasMoreTokens()) {           // 每行数据均输出一个<k, v>对
                String line = itr.nextToken();      // 获取下一行数据
                int randint=rd.nextInt(100);        // 随机一个新的course表项的标签
                if(courseFlag){                     // 数据来自course.tbl
                    k.set(String.valueOf(randint)); // 将随机的标签作为key
                    v.setCourse(line);              // 数据设为value
                    v.setnum(randint);              // 给当前数据加上标签
                    context.write(k, v);            // 输出<k, v>对
                }
                else{
                    v.setUniversity(line);          // 数据设为value
                    for (int i=0;i<100;i++){
                        k.set(String.valueOf(i));   // 复制100份，并赋上1到100的key
                        v.setnum(i);                // 给数据加上标签
                        context.write(k, v);        // 输出<k, v>对
                    } // end for
                } // end else
            }  // end while
        } // end void map
    } // end class CourseUniversityJoinMapper
```

```
private static class CourseUniversityJoinReducer extends Reducer<Text,
CourseUniversity, Text, NullWritable> {
        Text k = new Text();
        Text a = new Text();
        NullWritable v = NullWritable.get();
        @Override
        protected void reduce(Text key, Iterable<CourseUniversity> values,
Context context) throws IOException, InterruptedException {
            Vector<CourseUniversity> c = new Vector<CourseUniversity>();
```

```
            // 新建vector保存course表项
            Vector<CourseUniversity> u = new Vector<CourseUniversity>();
            // 新建vector保存university表项
            for (CourseUniversity value:values) { //遍历当前标签下所有的数据表项

                CourseUniversity cu=new CourseUniversity(value);
                if(!cu.getckey().equals("")) { //ckey不为空说明为course表项
                    c.add(cu);  //放入course表
                }
                else {  //为university表项
                    u.add(cu);  //放入unviersity表
                }

            }
            //遍历course和university表
            for (CourseUniversity course: c){
                for (CourseUniversity university: u){
                    if (course.num==university.num) { //保证标签一致
                        k.set(course.toString() + "|" + university.toString());
                        //连接两个表的输出
                        context.write(k, v); // 输出<k, v>对
                    } // end if
                } //end for u
            } //end for c
        } //end void reduce
    } //end class CourseUniversityJoinReducer
```

三.实验结果和输出路径

输出路径

任务1: /user/2021sg15/lab3/out1
任务2: /user/2021sg15/lab3/out2

UniversityCountry表

Universitycountry.u Key	Universitycountry.u Name	Universitycountry.u Webpage	Universitycountry.n Name
8380	Khalifa University of Science, Technology and Research	http://www.ku.ac.ae/	United Arab Emirates
8379	Al Khawarizmi International College	http://www.khawarizmi.com/	United Arab Emirates
8378	Jumeira University	http://www.ju.ac.ae/	United Arab Emirates
8377	Ittihad University	http://www.ittihad.ac.ae/	United Arab Emirates
8376	Higher Colleges of Technology	http://www.hct.ac.ae/	United Arab Emirates
8375	Hamdan Bin Mohammed e-University	http://www.hbmeu.ac.ae/	United Arab Emirates
8374	Gulf Medical University	http://www.gmu.ac.ae/	United Arab Emirates
8373	The Emirates Academy of Hotel Managment	http://www.emiratesacademy.edu/	United Arab Emirates
8372	Etisalat University College	http://www.ece.ac.ae/	United Arab Emirates

Universitycountry.u Key	Universitycountry.u Name	Universitycountry.u Webpage	Universitycountry.n Name
8368	British University in Dubai	http://www.buid.ac.ae/	United Arab Emirates
8367	American University of Sharjah	http://www.aus.edu/	United Arab Emirates
8366	American University in the Emirates	http://www.aue.ae/	United Arab Emirates
8365	American University in Dubai	http://www.aud.edu/	United Arab Emirates
8364	Alhosn University	http://www.alhosnu.ae/	United Arab Emirates
8363	Alain University of Science and Technology	http://www.alainuniversity.ac.ae/	United Arab Emirates
8362	Ajman University of Science & Technology	http://www.ajman.ac.ae/	United Arab Emirates
8361	Al Ghurair University	http://www.agu.ae/	United Arab Emirates
8360	Abu Dhabi University	http://www.adu.ac.ae/	United Arab Emirates

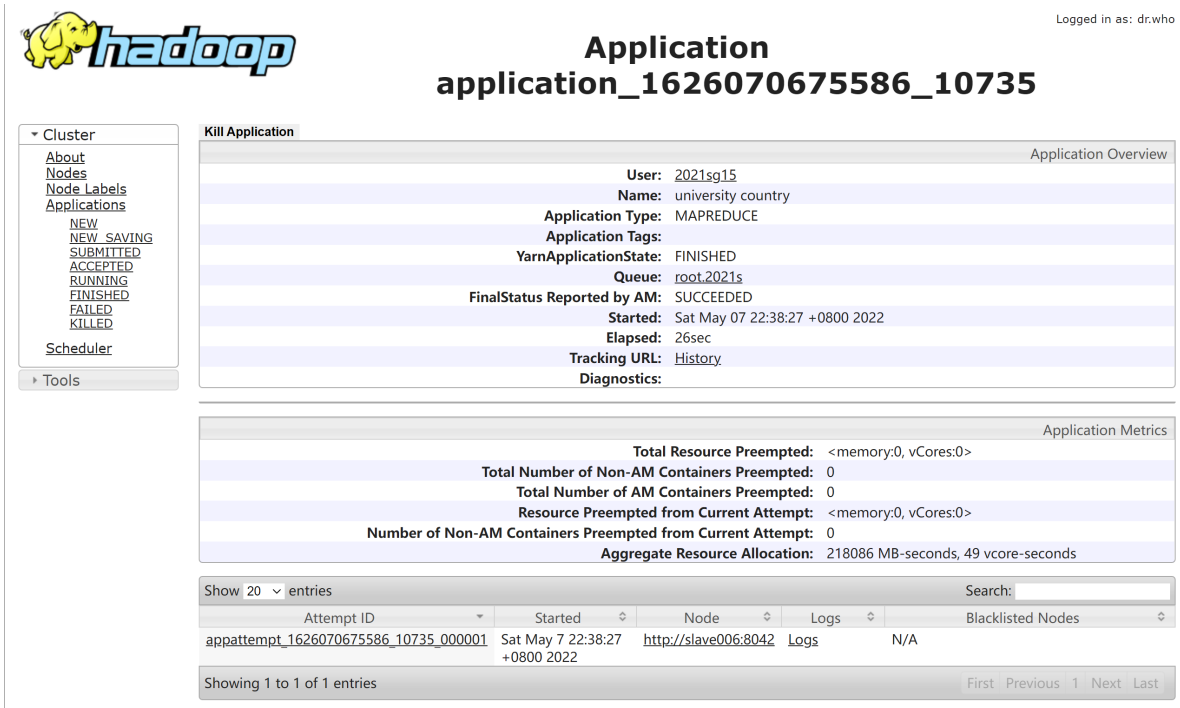
Universitycountry.u Key	Universitycountry.u Name	Universitycountry.u Webpage	Universitycountry.n Name
8391	University of Jazeera	http://www.uojazeera.com/	United Arab Emirates
8390	University Of Dubai	http://www.ud.ac.ae/	United Arab Emirates
8389	United Arab Emirates University	http://www.uaeu.ac.ae/	United Arab Emirates
8388	Paris-Sorbonne University Abu Dhabi	http://www.sorbonne.ae/	United Arab Emirates
8387	Skyline University College, Sharjah	http://www.skylineuniversity.com/	United Arab Emirates
8386	University of Sharjah	http://www.sharjah.ac.ae/	United Arab Emirates
8385	Rak Medical & Health Sciences University	http://www.rakmhsu.com/	United Arab Emirates
8384	The Petroleum Institute	http://www.pi.ac.ae/	United Arab Emirates
8383	New York University, Abu Dhabi	http://nyuad.nyu.edu/	United Arab Emirates

CourseUniversity表

Courseuniversity.c Key	Courseuniversity.c Name	Courseuniversity.c Subject	Courseuniversity.c Hours	Courseuniversity.u Key	Courseuniversity.u Name	Courseuniversity.u Webpage
131	History of Africa from 1800	AFST	3 hours.	9067	Harper College	http://www.harpercollege.edu
131	History of Africa from 1800	AFST	3 hours.	8104	King Mongkut's Institute of Technology Ladkrabang	http://www.kmitl.ac.th/
131	History of Africa from 1800	AFST	3 hours.	6853	Southwestern University	http://www.swu.edu.ph/
131	History of Africa from 1800	AFST	3 hours.	3965	Center for Entrepreneurship and Small Business Management	http://www.cesbm.ac.in/
131	History of Africa from 1800	AFST	3 hours.	1813	Centro Universitário Claretiano	http://www.claretiano.edu.br/
131	History of Africa from 1800	AFST	3 hours.	6344	Stenden University	http://www.stenden.com/
131	History of Africa from 1800	AFST	3 hours.	4474	Universitas Palangka Raya	http://www.upr.ac.id/
131	History of Africa from 1800	AFST	3 hours.	33	Western New England University	http://www1.wne.edu/
131	History of Africa from 1800	AFST	3 hours.	8251	Selcuk University	http://www.selcuk.edu.tr/
131	History of Africa from 1800	AFST	3 hours.	2466	Jiangxi Normal University	http://www.jxnu.edu.cn/

131	History of Africa from 1800	AFST	3 hours.	4473	Universitas Pembangunan Nasional "Veteran" Yogyakarta	http://www.upnyk.ac.id/
131	History of Africa from 1800	AFST	3 hours.	8776	Istanbul Kemerburgaz University	http://www.kemerburgaz.edu.tr/
131	History of Africa from 1800	AFST	3 hours.	1099	University of South Alabama	http://www.southalabama.edu/
131	History of Africa from 1800	AFST	3 hours.	6852	Silliman University	http://www.su.edu.ph/
131	History of Africa from 1800	AFST	3 hours.	5039	Kanazawa Gakuin University	http://www.kanazawa-gu.ac.jp/
131	History of Africa from 1800	AFST	3 hours.	2798	Université Kongo	http://www.universitekongo.org/
131	History of Africa from 1800	AFST	3 hours.	801	Saint Joseph's University	http://www.sju.edu/
131	History of Africa from 1800	AFST	3 hours.	6345	Tilburg University	http://www.tilburguniversity.nl/
131	History of Africa from 1800	AFST	3 hours.	2111	University of Moncton, Shippagan	http://www.cus.ca/
131	History of Africa from 1800	AFST	3 hours.	9228	Washtenaw Community College	http://www.wccnet.edu
131	History of Africa from 1800	AFST	3 hours.	9146	Gateway Community and Technical College	http://www.gateway.kctcs.edu
131	History of Africa from 1800	AFST	3 hours.	1515	Fachhochschule Wiener Neustadt	http://www.fhwn.ac.at/
131	History of Africa from 1800	AFST	3 hours.	446	Governors State University	http://www.govst.edu/
131	History of Africa from 1800	AFST	3 hours.	2713	Pontificia Universidad Javeriana	http://www.javeriana.edu.co/
131	History of Africa from 1800	AFST	3 hours.	6851	Samar State University	http://www.ssu.edu.ph/
131	History of Africa from 1800	AFST	3 hours.	7835	Sudan Academy of Sciences	http://www.sas-sd.net/
131	History of Africa from 1800	AFST	3 hours.	7930	FFHS - Fernfachhochschule Schweiz	http://www.ffhs.ch/
131	History of Africa from 1800	AFST	3 hours.	3967	Central Institute of English and Foreign Languages	http://www.ciefl.org/

UniversityCountry连接的执行报告



Logged in as: dr.who

Application application_1626070675586_10735

Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools

Kill Application

Application Overview

User:	2021sg15
Name:	university country
Application Type:	MAPREDUCE
Application Tags:	
YarnApplicationState:	FINISHED
Queue:	root.2021s
FinalStatus Reported by AM:	SUCCEEDED
Started:	Sat May 07 22:38:27 +0800 2022
Elapsed:	26sec
Tracking URL:	History
Diagnostics:	

Application Metrics

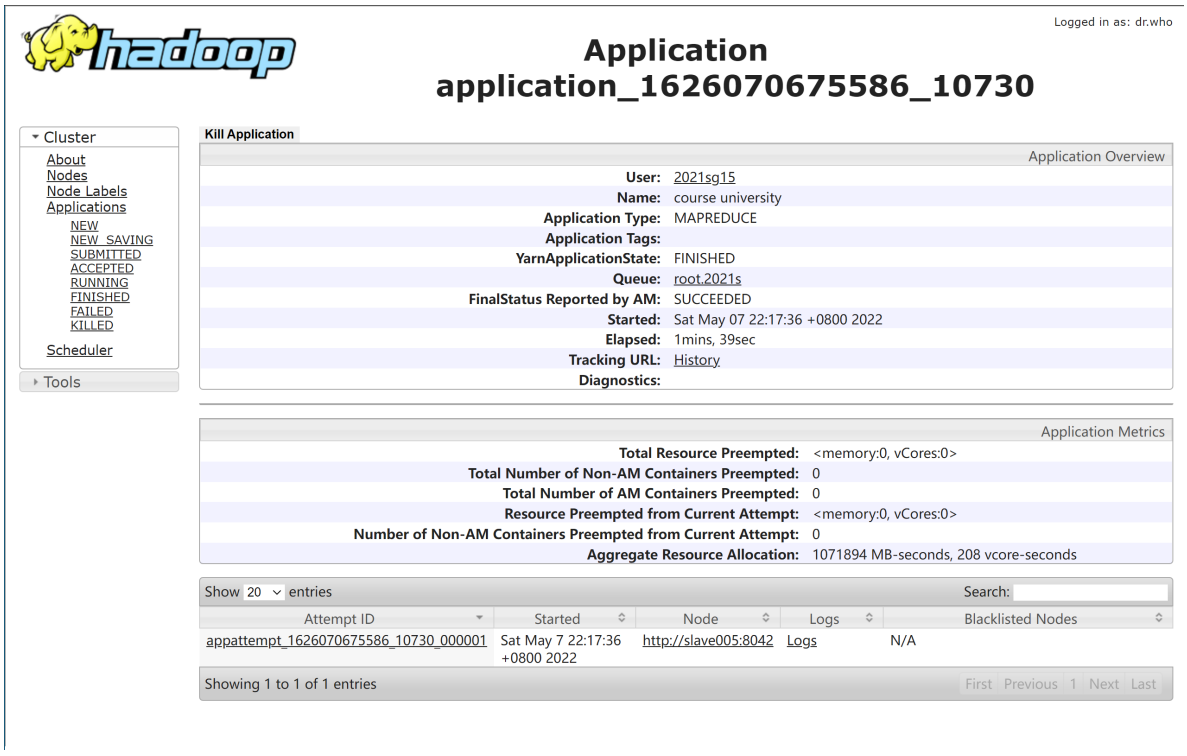
Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	218086 MB-seconds, 49 vcore-seconds

Show 20 entries Search:

Attempt ID	Started	Node	Logs	Blacklisted Nodes
appattempt_1626070675586_10735_000001	Sat May 7 22:38:27 +0800 2022	http://slave006:8042	Logs	N/A

Showing 1 to 1 of 1 entries First Previous 1 Next Last

CourseUniversity连接的执行报告



Logged in as: dr.who

Application application_1626070675586_10730

Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools

Kill Application

Application Overview

User:	2021sg15
Name:	course university
Application Type:	MAPREDUCE
Application Tags:	
YarnApplicationState:	FINISHED
Queue:	root.2021s
FinalStatus Reported by AM:	SUCCEEDED
Started:	Sat May 07 22:17:36 +0800 2022
Elapsed:	1mins, 39sec
Tracking URL:	History
Diagnostics:	

Application Metrics

Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	1071894 MB-seconds, 208 vcore-seconds

Show 20 entries Search:

Attempt ID	Started	Node	Logs	Blacklisted Nodes
appattempt_1626070675586_10730_000001	Sat May 7 22:17:36 +0800 2022	http://slave005:8042	Logs	N/A

Showing 1 to 1 of 1 entries First Previous 1 Next Last