

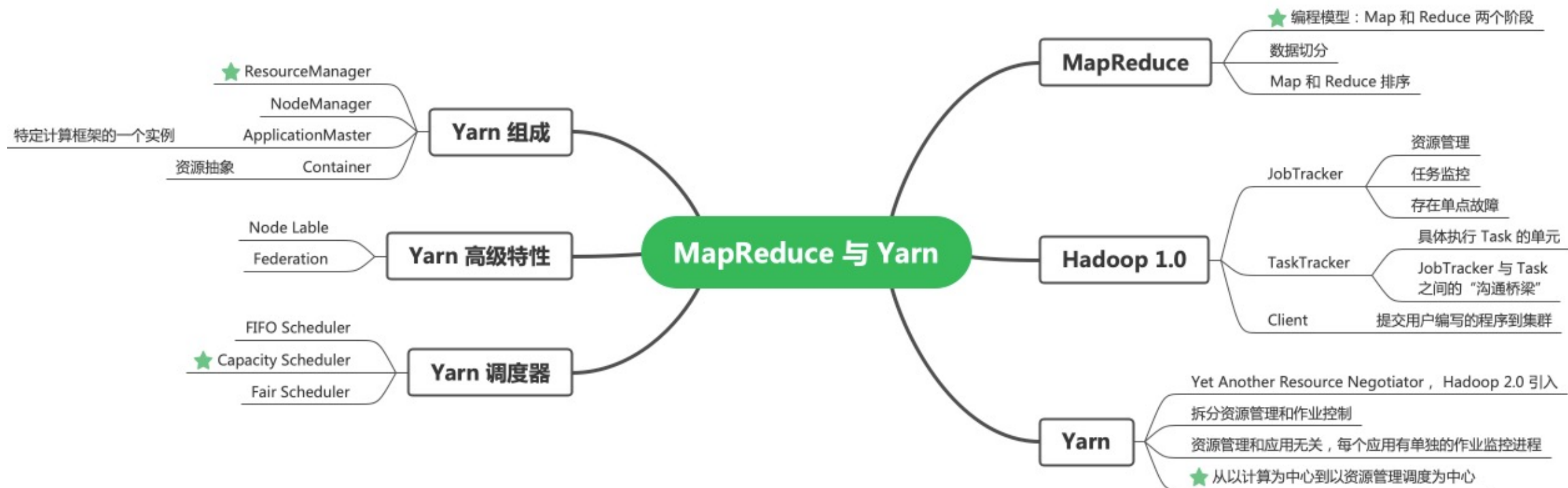
# MapReduce与Yarn

张语

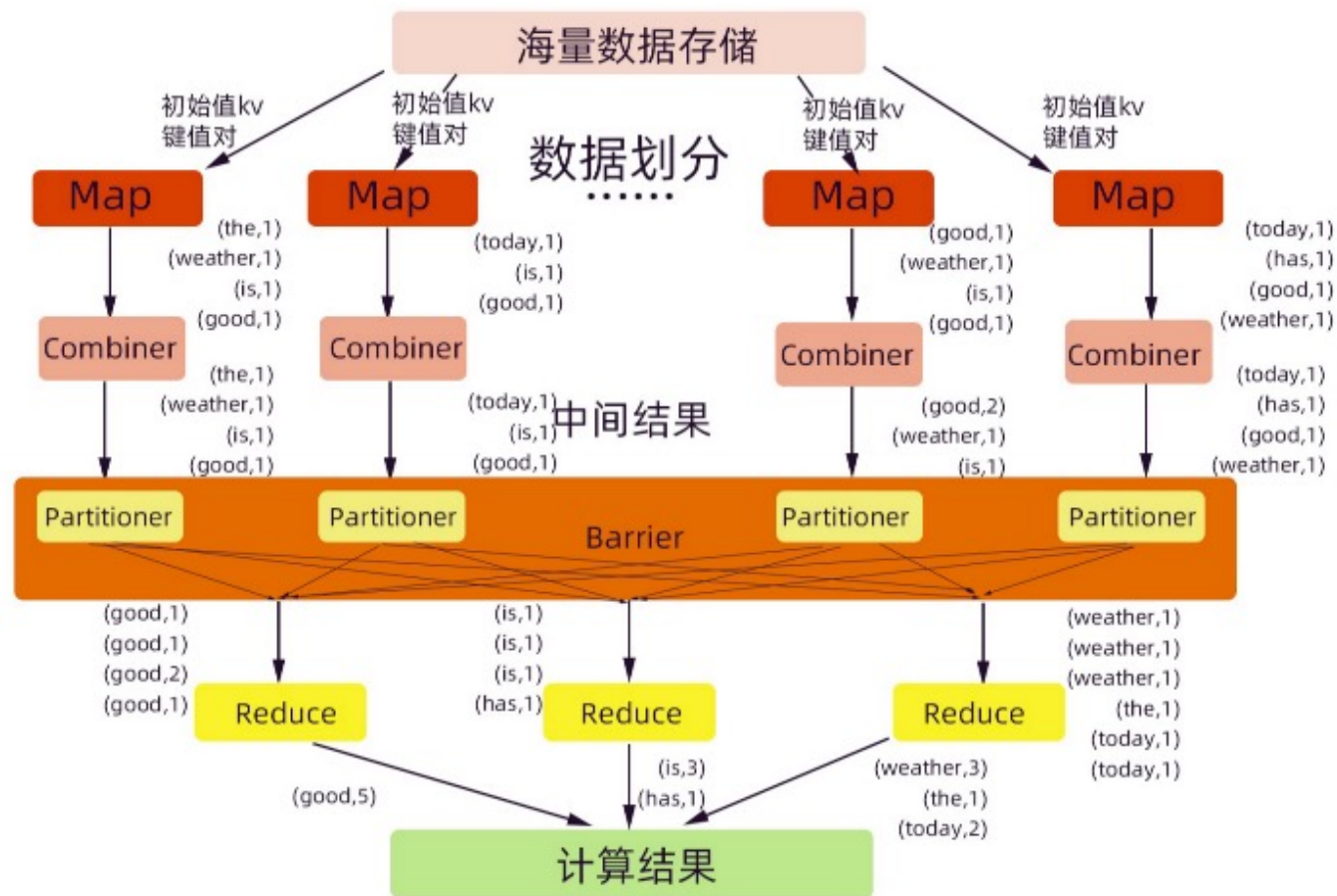
# 目录

- 重点内容回顾
- 作业讲解
- Hadoop Streaming
- Yarn UI
- QA

# 重点内容回顾



# MapReduce编程模型



# MapReduce编程模型

- 分治法

1. 分解原问题。将原问题分解为若干个规模较小，相互独立，且与原问题形式相同的子问题
2. 求解子问题。若子问题规模较小且容易被解决则直接求解，否则递归地求解各个子问题
3. 合并解，就是将各个子问题的解合并为原问题的解

- MapReduce

- Map：分解，把复杂的任务分解为若干个简单的任务执行
- Reduce：合并，对 Map 阶段的结果进行汇总

# MapReduce编程模型

- 使用场景：任务可被分解成相互独立的子问题
  1. 迭代。遍历输入数据，并解析成 key/value 对
  2. 将输入 key/value 对映射（map）成另外一些 key/value 对
  3. 根据 key 对中间数据进行分组（group）
  4. 以组为单位对数据进行归约（reduce）
  5. 迭代。将 reduce 阶段产生的key/value对保存到输出文件中

# MapReduce 缺点

- **高昂的维护成本**

- MapReduce 模型过于抽象，当遇到复杂度较高的需求时需要手动编排多个Map和多个Reduce任务
- Spark、Flink等使用有向无环图（DAG）表达数据处理流程
  - 每个节点都表达一种通用的数据集
  - 每一条边表达一种通用的数据变换

- **时间性能较差**

- Map 和 Reduce 的输出都需要落盘（本地磁盘、HDFS）

# 作业讲解

- 统计每一个手机号耗费的总上行流量、下行流量、总流量
- 作业详见：<https://u.geekbang.org/lesson/180?article=491889>
- 基本思路：
  - Map 阶段：
    - 读取一行数据，切分字段。
    - 抽取手机号、上行流量、下行流量。
    - 以手机号为 key，bean 对象为 value 输出，即 `context.write(手机号, bean)`。
  - Reduce 阶段：
    - 累加上行流量和下行流量得到总流量。
    - 实现自定义的 bean 来封装流量信息，并将 bean 作为 map 输出的 key 来传输



# 作业讲解

hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/Mapper.java

```
public class Mapper<KEYIN, VALUEIN, KEYOUT, VALUEOUT> {  
  
    /** The <code>Context</code> passed on to the {@link Mapper} implementations. */  
    public abstract class Context  
        implements MapContext<KEYIN,VALUEIN,KEYOUT,VALUEOUT> {  
    }  
  
    /** Called once at the beginning of the task. */  
    protected void setup(Context context  
        ) throws IOException, InterruptedException {}  
  
    /**  
     * Called once for each key/value pair in the input split. Most applications  
     * should override this, but the default is the identity function.  
     */  
    /unchecked/  
    protected void map(KEYIN key, VALUEIN value,  
        Context context) throws IOException, InterruptedException {  
        context.write((KEYOUT) key, (VALUEOUT) value);  
    }  
}
```

# 作业讲解

hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java/org/apache/hadoop/mapreduce/Reducer.java

```
public class Reducer<KEYIN, VALUEIN, KEYOUT, VALUEOUT> {  
  
    /** The <code>Context</code> passed on to the {@link Reducer} implementations. */  
    public abstract class Context  
        implements ReduceContext<KEYIN, VALUEIN, KEYOUT, VALUEOUT> {  
    }  
  
    /** Called once at the start of the task. */  
    protected void setup(Context context  
        ) throws IOException, InterruptedException {}  
  
    /**  
     * This method is called once for each key. Most applications will define  
     * their reduce class by overriding this method. The default implementation  
     * is an identity function.  
     */  
    /unchecked/  
    protected void reduce(KEYIN key, Iterable<VALUEIN> values, Context context  
        ) throws IOException, InterruptedException {  
        for(VALUEIN value: values) {  
            context.write((KEYOUT) key, (VALUEOUT) value);  
        }  
    }  
}
```

# 作业讲解

```
public class PhoneTraffic implements Writable {
    // 上行流量
    private long up;
    // 下行流量
    private long down;
    // 总流量
    private long sum;

    public PhoneTraffic() {}

    public PhoneTraffic(long up, long down, long sum) {...}

    @Override
    public void write(DataOutput dataOutput) throws IOException {
        dataOutput.writeLong(up);
        dataOutput.writeLong(down);
        dataOutput.writeLong(sum);
    }

    @Override
    public void readFields(DataInput dataInput) throws IOException {
        this.up = dataInput.readLong();
        this.down = dataInput.readLong();
        this.sum = dataInput.readLong();
    }
}
```

# 作业讲解

```
public static class TrafficMapper extends Mapper<Object, Text, Text, PhoneTraffic> {

    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
        String[] lines = value.toString().split(regex: "\t");
        if (lines.length < 10) {
            return;
        }

        String phone = lines[1];
        try {
            long up = Long.parseLong(lines[8]);
            long down = Long.parseLong(lines[9]);
            context.write(new Text(phone), new PhoneTraffic(up, down, sum: up + down));
        } catch (NumberFormatException e) {
            System.err.println("parseLong failed" + e.getMessage());
        }
    }
}
```



# 作业讲解

```
public static class TrafficReducer extends Reducer<Text, PhoneTraffic, Text, PhoneTraffic> {  
    public void reduce(Text key, Iterable<PhoneTraffic> values, Context context  
    ) throws IOException, InterruptedException {  
        int totalUp = 0;  
        int totalDown = 0;  
        int sumTraffic = 0;  
        for (PhoneTraffic val : values) {  
            totalUp += val.getUp();  
            totalDown += val.getDown();  
            sumTraffic += val.getSum();  
        }  
  
        context.write(key, new PhoneTraffic(totalUp, totalDown, sumTraffic));  
    }  
}
```

# 作业讲解

```
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
    if (otherArgs.length < 2) {
        System.err.println("Usage: TrafficStat <in> <out>");
        System.exit(status: 2);
    }
    System.out.println("otherArgs: " + Arrays.toString(otherArgs));

    Job job = Job.getInstance(conf, jobName: "TrafficStat");
    job.setJarByClass(TrafficStat.class);
    job.setMapperClass(TrafficMapper.class);
    job.setCombinerClass(TrafficReducer.class);
    job.setReducerClass(TrafficReducer.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(PhoneTraffic.class);

    job.setNumReduceTasks(1);

    FileInputFormat.addInputPath(job, new Path(otherArgs[otherArgs.length - 2]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));
    System.exit(job.waitForCompletion(verbose: true) ? 0 : 1);
}
```

# 作业讲解



## 运行演示

```
hadoop jar bigdata-tutorial-1.0-SNAPSHOT.jar TrafficStat /user/public/week2/ /user/public/week2_out/
```

# Hadoop Streaming

- 允许用户使用**可执行文件或者脚本**作为 Mapper/Reducer 创建和运行 MapReduce 任务，降低开发门槛
- 要求编写的 Mapper/Reducer 从标准输入中读取数据，将结果写到标准输出中
- 官方文档：<https://hadoop.apache.org/docs/stable/hadoop-streaming/HadoopStreaming.html>

```
mapred streaming \  
    -input myInputDirs \  
    -output myOutputDir \  
    -mapper /bin/cat \  
    -reducer /usr/bin/wc
```



# Hadoop Streaming – 配置



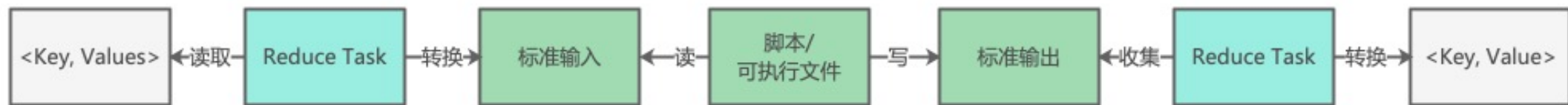
Parameter	Optional/Required	Description
-input directoryname or filename	Required	Input location for mapper
-output directoryname	Required	Output location for reducer
-mapper executable or JavaClassName	Optional	Mapper executable. If not specified, IdentityMapper is used as the default
-reducer executable or JavaClassName	Optional	Reducer executable. If not specified, IdentityReducer is used as the default
-file filename	Optional	Make the mapper, reducer, or combiner executable available locally on the compute nodes
-combiner streamingCommand or JavaClassName	Optional	Combiner executable for map output
-cmdenv name=value	Optional	Pass environment variable to streaming commands
-numReduceTasks	Optional	Specify the number of reducers

# Hadoop Streaming – 原理

- 使用可执行文件或者脚本文件充当Mapper或者Reducer时，Java端的Mapper或者Reducer充当了wrapper角色，将输入文件中的key和value直接传递给可执行文件或者脚本文件进行处理，并将处理结果写入HDFS



Map 阶段



Reduce 阶段

# Hadoop Streaming – 原理

难点：如何使用标准输入输出实现Java与可执行文件、脚本之间的通信？

```
205 // Start the process
206 ProcessBuilder builder = new ProcessBuilder(argvSplit);
207 builder.environment().putAll(childEnv.toMap());
208 sim = builder.start();
209
210 clientOut_ = new DataOutputStream(new BufferedOutputStream(
211                                     sim.getOutputStream(),
212                                     BUFFER_SIZE));
213 clientIn_ = new DataInputStream(new BufferedInputStream(
214                                   sim.getInputStream(),
215                                   BUFFER_SIZE));
216 clientErr_ = new DataInputStream(new BufferedInputStream(sim.getErrorStream()));
```

hadoop-tools/hadoop-streaming/src/main/java/org/apache/hadoop/streaming/PipeMapRed.java

# Hadoop Streaming – 使用

```
def traffic_mapper():  
    for line in sys.stdin:  
        arr = line.strip().split("\t")  
        if len(arr) < 10:  
            continue  
        # 电话号码, 上行流量, 下行流量  
        print("%s\t%s\t%s" % (arr[1], arr[8], arr[9]))
```

# Hadoop Streaming – 使用

```
def traffic_reducer():
    last_phone = ""
    up, down = 0, 0
    for line in sys.stdin:
        arr = line.strip().split("\t")

        if last_phone == "":
            last_phone = arr[0]
        elif last_phone != arr[0]:
            # 输出上一个电话的流量统计
            print("%s\t%s\t%s\t%s" % (last_phone, up, down, up + down))
            last_phone = arr[0]
            up, down = 0, 0

        up += int(arr[1])
        down += int(arr[2])

    if last_phone != "":
        print("%s\t%s\t%s\t%s" % (last_phone, up, down, up + down))
```

# Hadoop Streaming – 使用

```
def main():  
    if sys.argv[1] == "map":  
        traffic_mapper()  
    elif sys.argv[1] == "reduce":  
        traffic_reducer()  
  
if __name__ == "__main__":  
    main()
```

## 运行演示

mapred streaming \  
-input /user/public/week2/ \  
-output /user/public/week2\_streaming/ \  
-mapper '/usr/bin/python3 traffic\_stat.py map' \  
-reducer '/usr/bin/python3 traffic\_stat.py reduce' \  
-file traffic\_stat.py \  
-numReduceTasks 1

# Yarn UI



<https://knox.c-01b94588f59c7655.cn-hangzhou.emr.aliyuncs.com:8443/gateway/cluster-topo/yarn/cluster/apps>



Logged in as: student

## About the Cluster

### Cluster

#### About

#### Nodes

#### Node Labels

#### Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

#### Scheduler

### Tools

### Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
9	0	0	9	0	0 B	74.50 GB	0 B	0	64	0

### Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
8	0	0	0	0	0	0

### Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mb), vcores]	<memory:32, vCores:1>	<memory:9536, vCores:8>	0

Cluster overview

Cluster ID: 1645699879292

ResourceManager state: STARTED

ResourceManager HA state: active

ResourceManager HA zookeeper connection state: Could not find leader elector. Verify both HA and automatic failover are enabled.

ResourceManager RMStateStore: org.apache.hadoop.yarn.server.resourcemanager.recovery.NullRMStateStore

ResourceManager started on: Thu Feb 24 18:51:19 +0800 2022

ResourceManager version: 3.2.1 from fdbf79bb25ebd52e198bcc564baf822d0a6b7024 by jenkins source checksum 3120421249ad6ad216e5915e1442e18 on 2021-07-15T08:00Z

Hadoop version: 3.2.1 from fdbf79bb25ebd52e198bcc564baf822d0a6b7024 by jenkins source checksum a727b26fa21579ad1b194bc17821d8 on 2021-07-15T07:57Z



# Yarn UI-Applications



## All Applications

Logged in as: student

Cluster

- About Nodes
- Node Labels
- Applications
- NEW
- NEW SAVING
- SUBMITTED
- ACCEPTED
- RUNNING
- FINISHED
- FAILED
- KILLED
- Scheduler

Tools

### Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
9	0	0	9	0	0 B	74.50 GB	0 B	0	64	0

### Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
8	0	0	0	0	0	0

### Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:32, vCores:1>	<memory:9536, vCores:8>	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCoers	Allocated Memory MB	Reserved CPU VCoers	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1645699879292_0013	student1	mobileflow-1.0-SNAPSHOT.jar	MAPREDUCE	default	0	Mon Mar 7 18:02:01 +0800 2022	Mon Mar 7 18:02:02 +0800 2022	Mon Mar 7 18:02:18 +0800 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History	0
application_1645699879292_0012	student1	mobileflow-1.0-SNAPSHOT.jar	MAPREDUCE	default	0	Mon Mar 7 17:40:06 +0800 2022	Mon Mar 7 17:40:07 +0800 2022	Mon Mar 7 17:40:21 +0800 2022	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History	0
application_1645699879292_0011	student1	mobileflow-1.0-SNAPSHOT.jar	MAPREDUCE	default	0	Mon Mar 7 16:12:50	Mon Mar 7 16:12:50	Mon Mar 7 16:13:05	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History	0



# Yarn UI-Applications



## Application application\_1645699879292\_0013

Logged in as: student

### Cluster

About  
Nodes  
Node Labels  
Applications  
NEW  
NEW SAVING  
SUBMITTED  
ACCEPTED  
RUNNING  
FINISHED  
FAILED  
KILLED

Scheduler

### Tools

### Application Overview

User: [student1](#)  
Name: mobileflow-1.0-SNAPSHOT.jar  
Application Type: MAPREDUCE  
Application Tags:  
Application Priority: 0 (Higher Integer value indicates higher priority)  
YarnApplicationState: FINISHED  
Queue: [default](#)  
FinalStatus Reported by AM: SUCCEEDED  
Started: Mon Mar 07 18:02:01 +0800 2022  
Launched: Mon Mar 07 18:02:02 +0800 2022  
Finished: Mon Mar 07 18:02:18 +0800 2022  
Elapsed: 16sec  
Tracking URL: [History](#)  
Log Aggregation Status: SUCCEEDED  
Application Timeout (Remaining Time): Unlimited  
Diagnostics:  
Unmanaged Application: false  
Application Node Label expression: <Not set>  
AM container Node Label expression: <DEFAULT\_PARTITION>

### 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: 122908 MB-seconds, 47 vcore-seconds  
Aggregate Preempted Resource Allocation: 0 MB-seconds, 0 vcore-seconds

Show 20 entries

Search:

Attempt ID	Started	Node	Logs	Nodes blacklisted by the app	Nodes blacklisted by the system
appattempt_1645699879292_0013_000001	Mon Mar 7 18:02:01 +0800 2022	<a href="http://emr-worker-3.cluster-285604.8042">http://emr-worker-3.cluster-285604.8042</a>	<a href="#">Logs</a> 0	0	0

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

# Yarn UI-Applications



Application History

About Applications

FINISHED  
FAILED  
KILLED

Tools

Log Type: directory.info  
Log Upload Time: Mon Mar 07 18:02:26 +0800 2022  
Log Length: 2244

```
ls -l:
total 36
-rw-r----- 1 hadoop hadoop 100 Mar 7 18:02 container_tokens
-rwx----- 1 hadoop hadoop 616 Mar 7 18:02 default_container_executor_session.sh
-rwx----- 1 hadoop hadoop 671 Mar 7 18:02 default_container_executor.sh
lrwxrwxrwx 1 hadoop hadoop 95 Mar 7 18:02 job.jar -> /mnt/disk1/yarn/usercache/student1/appcache/application_1645699879292_0013/filecache/11/job.jar
drwxrwxr-x 2 hadoop hadoop 4096 Mar 7 18:02 jobSubmitDir
lrwxrwxrwx 1 hadoop hadoop 95 Mar 7 18:02 job.xml -> /mnt/disk3/yarn/usercache/student1/appcache/application_1645699879292_0013/filecache/13/job.xml
-rwx----- 1 hadoop hadoop 6403 Mar 7 18:02 launch_container.sh
drwx--x--- 2 hadoop hadoop 4096 Mar 7 18:02 tmp
find -L . -maxdepth 5 -ls:
2097165 4 drwx--x--- 4 hadoop hadoop 4096 Mar 7 18:02 .
2097171 4 -rwx----- 1 hadoop hadoop 616 Mar 7 18:02 ./default_container_executor_session.sh
2097168 4 -rw-r----- 1 hadoop hadoop 12 Mar 7 18:02 ./container_tokens.crc
2097175 4 drwxrwxr-x 2 hadoop hadoop 4096 Mar 7 18:02 ./jobSubmitDir
2228243 4 -r-x----- 1 hadoop hadoop 161 Mar 7 18:02 ./jobSubmitDir/job.split
2228240 4 -r-x----- 1 hadoop hadoop 71 Mar 7 18:02 ./jobSubmitDir/job.splitmetainfo
2097172 4 -rw-r----- 1 hadoop hadoop 16 Mar 7 18:02 ./default_container_executor_session.sh.crc
1835022 4 drwx----- 2 hadoop hadoop 4096 Mar 7 18:02 ./job.jar
1835023 8 -r-x----- 1 hadoop hadoop 6573 Mar 7 18:02 ./job.jar/job.jar
2097166 4 drwx--x--- 2 hadoop hadoop 4096 Mar 7 18:02 ./tmp
2097169 8 -rwx----- 1 hadoop hadoop 6403 Mar 7 18:02 ./launch_container.sh
2097174 4 -rw-r----- 1 hadoop hadoop 16 Mar 7 18:02 ./default_container_executor.sh.crc
2097173 4 -rwx----- 1 hadoop hadoop 671 Mar 7 18:02 ./default_container_executor.sh
2097170 4 -rw-r----- 1 hadoop hadoop 60 Mar 7 18:02 ./launch_container.sh.crc
2097167 4 -rw-r----- 1 hadoop hadoop 100 Mar 7 18:02 ./container_tokens
1835021 204 -r-x----- 1 hadoop hadoop 206913 Mar 7 18:02 ./job.xml
broken symlinks(find -L . -maxdepth 5 -type l -ls):
```

Log Type: launch\_container.sh  
Log Upload Time: Mon Mar 07 18:02:26 +0800 2022  
Log Length: 6403

Showing 4096 bytes of 6403 total [Click here for the full log.](#)

点击跳转到日志详情

```
disk2/log/hadoop-yarn/containers/application_1645699879292_0013/container_1645699879292_0013_01_000001,/mnt/disk3/log/hadoop-yarn/containers/application_1645699879292_0013_01_000001
export USER="student1"
export LOGNAME="student1"
export HOME="/home/"
export PWD="/mnt/disk2/yarn/usercache/student1/appcache/application_1645699879292_0013/container_1645699879292_0013_01_000001"
```

# Yarn UI-Applications



## MapReduce Job job\_1645699879292\_0013

▸ Application
▾ Job
Overview
Counters
Configuration
Map tasks
Reduce tasks
▸ Tools

Job Name:	mobileflow-1.0-SNAPSHOT.jar
User Name:	student1
Queue:	default
State:	SUCCEEDED
Uberized:	false
Submitted:	Mon Mar 07 18:02:01 CST 2022
Started:	Mon Mar 07 18:02:07 CST 2022
Finished:	Mon Mar 07 18:02:18 CST 2022
Elapsed:	10sec
Diagnostics:	
Average Map Time	2sec
Average Shuffle Time	2sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster				
Attempt Number	Start Time	Node	Logs	
1	Mon Mar 07 18:02:03 CST 2022	emr-worker-3.cluster-285604:8042	https://knox.c-01b94588f59c7655.cn-hangzhou.emr.aliyuncs.com:8443/gateway/jobhistoryui/jobhistory/logs	
Task Type		Total	Complete	
Map		1	1	
Reduce		7	7	
Attempt Type		Failed	Killed	Successful
Maps		0	0	1
Reduces		0	0	7

# Yarn UI-Scheduler



## NEW,NEW\_SAVING,SUBMITTED,ACCEPTED,RUNNING Applications

Logged in as: student

Cluster

- About
- Nodes
- Node Labels
- Applications
  - NEW
  - NEW\_SAVING
  - SUBMITTED
  - ACCEPTED
  - RUNNING
  - FINISHED
  - FAILED
  - KILLED
  - Scheduler**

Tools

### Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
9	0	0	9	0	0 B	74.50 GB	0 B	0	64	0

### Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
8	0	0	0	0	0	0

### Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:32, vCores:1>	<memory:9536, vCores:8>	0

Dump scheduler logs 1 min

### Application Queues

Legend: Capacity Used Used (over capacity) Max Capacity Users Requesting Resources Auto Created Queues

Queue: root	0.0% used
Queue: default	0.0% used

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No data available in table																				

Showing 0 to 0 of 0 entries

First Previous Next Last

QA