大数据大作业实验报告

汇报小组：12组 汇报时间：2024年6月

目 录

COMPANY

01 问题描述

02 完成时长

03 方案介绍

04 流程详述

ONE

问题描述

Describing problem

实验内容：健康码红码生成模拟

Experimental content : simulating the
generation of red health code

问题背景

新冠抗疫期间，在发现感染病人后可以通过其手机的漫游信息发现与其行程有交集的人，从而将其健康码标注为红色。

实验文件

cdinfo.txt：提供了基站下载汇总的人员漫游信息4列信息分别是基站编号，时间，注册状态（1表示注册入基站，2表示离开基站），手机号码。

Infected.txt：被感染人员的手机号码。

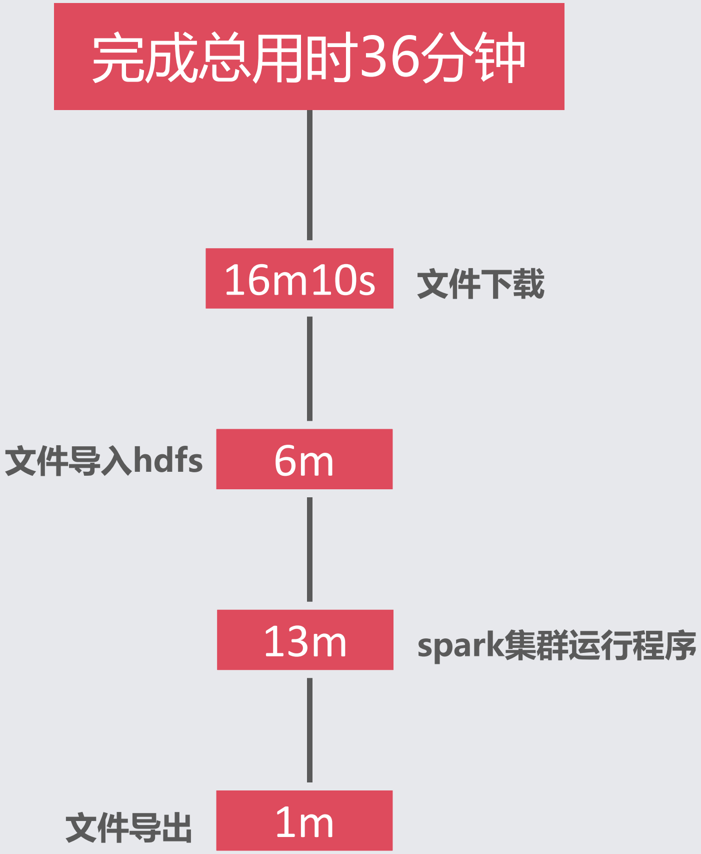
实验要求

依据infected.txt在cdinfo.txt中找到与感染人员同时间在同一基站的的手机列表

TWO

完成时长

Completion time



完成总用时36分钟

16m10s 文件下载

文件导入hdfs 6m

13m spark集群运行程序

文件导出 1m

THREE

方案介绍

solution introduction

实验工具

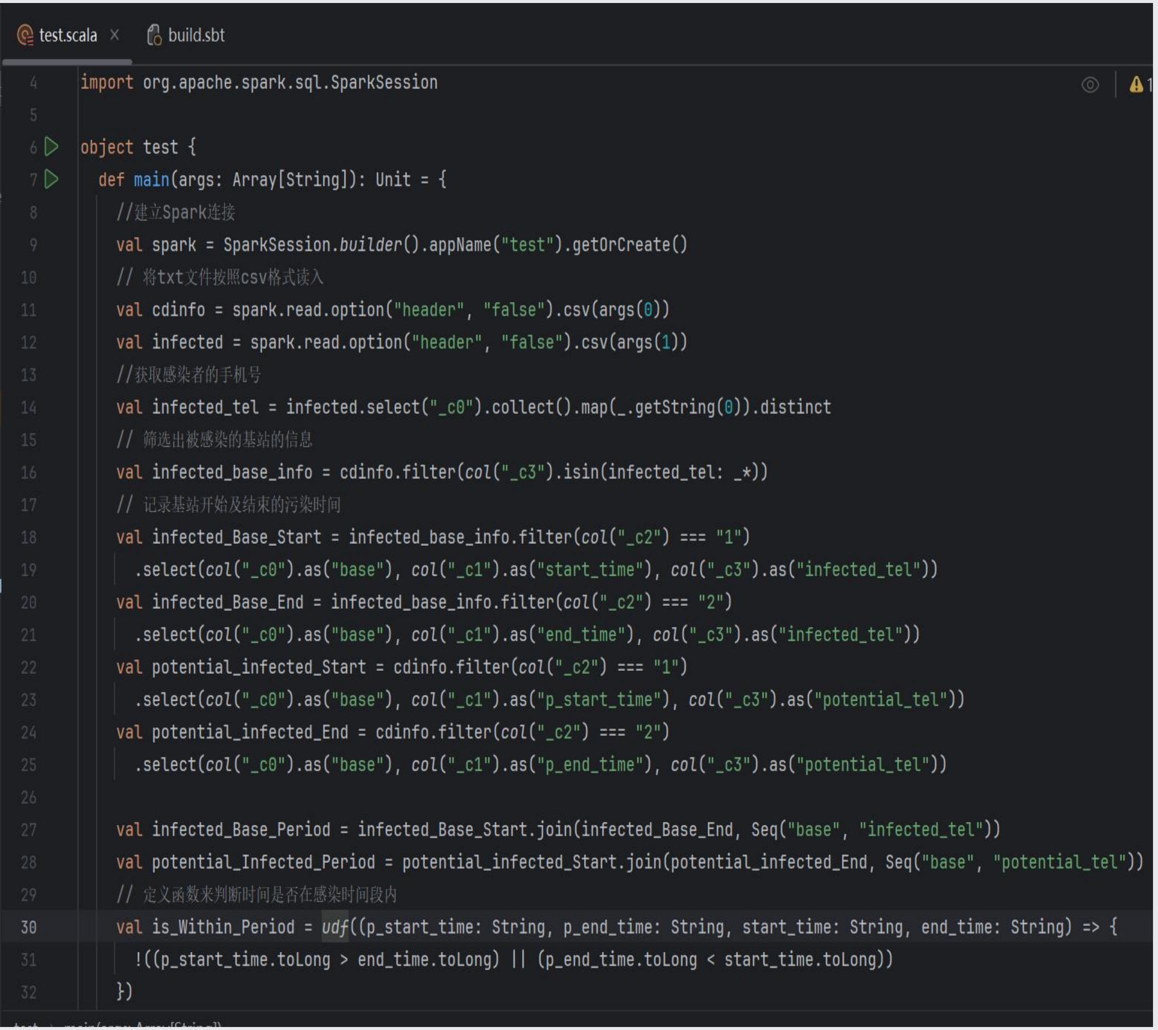
Experimental tools

实验平台 ❯❯❯ 快速实训中的大数据综合实训平台

编程语言 ❯❯❯ 在IDEA上采用scala编程并打包生成jar

集群搭建 ❯❯❯ 在Hadoop分布式集群的基础上搭建spark分布式集群

筛选感染者算法



```scala
import org.apache.spark.sql.SparkSession

object test {
  def main(args: Array[String]): Unit = {
    //建立Spark连接
    val spark = SparkSession.builder().appName("test").getOrCreate()
    // 将txt文件按照csv格式读入
    val cdinfo = spark.read.option("header", "false").csv(args(0))
    val infected = spark.read.option("header", "false").csv(args(1))
    //获取感染者的手机号
    val infected_tel = infected.select("_c0").collect().map(_.getString(0)).distinct
    // 筛选出被感染的基站的信息
    val infected_base_info = cdinfo.filter(col("_c3").isin(infected_tel: _*))
    // 记录基站开始及结束的污染时间
    val infected_Base_Start = infected_base_info.filter(col("_c2") === "1")
      .select(col("_c0").as("base"), col("_c1").as("start_time"), col("_c3").as("infected_tel"))
    val infected_Base_End = infected_base_info.filter(col("_c2") === "2")
      .select(col("_c0").as("base"), col("_c1").as("end_time"), col("_c3").as("infected_tel"))
    val potential_infected_Start = cdinfo.filter(col("_c2") === "1")
      .select(col("_c0").as("base"), col("_c1").as("p_start_time"), col("_c3").as("potential_tel"))
    val potential_infected_End = cdinfo.filter(col("_c2") === "2")
      .select(col("_c0").as("base"), col("_c1").as("p_end_time"), col("_c3").as("potential_tel"))

    val infected_Base_Period = infected_Base_Start.join(infected_Base_End, Seq("base", "infected_tel"))
    val potential_Infected_Period = potential_infected_Start.join(potential_infected_End, Seq("base", "potential_tel"))
    // 定义函数来判断时间是否在感染时间段内
    val is_Within_Period = udf((p_start_time: String, p_end_time: String, start_time: String, end_time: String) => {
      !((p_start_time.toLong > end_time.toLong) || (p_end_time.toLong < start_time.toLong))
    })
```

第一步：读入文件，利用cdinfo与infected文件创建被感染者的基站信息

基站id	时间	状态（1或2）	电话
…			

第二步：记录基站开始及结束被感染的时间

索引	基站id	开始时间	被感染者电话

索引	基站id	离开时间	被感染者电话

第三步：利用cdinfo创建所有人员的进站时间和离站时间

索引	基站id	开始时间	潜在感染者电话

索引	基站id	开始时间	潜在感染者电话

第四步：将感染者和潜在感染者的进站时间表和离站时间表连接，并用函数判断潜在感染者是否应被标记。**判断条件**为：如果潜在感染者的进站时间小于等于感染结束时间并且潜在感染者离站时间大于等于感染开始时间则标记为感染

索引	基站id	感染开始时间	感染结束时间	潜在感染者进站时间	潜在感染者离站时间	潜在感染者电话

算法细节

考虑到有同一感染者多次进出同一基站的情况，在创建新表时第一列为**索引**，这样连接两个表时将按照索引连接，而不会出现一个感染区间的开始时间与另一个感染区间结束时间关联到一起的情况。

Scala语言生成的表不能使用行索引，需要使用RDD的方法。但RDD的collect（）方法太耗时，因此采用不断创建和连接新表的方法

平台提供的sbt无法使用，因此在windows系统下下载sbt并将代码打包形成jar包再导入到平台

Spark on yarn集群搭建

添加节点

添加节点：每个人的账号自带**两个slave节点**，将所有人的节点连接到master节点。

修改/etc/hosts： 增加slave节点与master节点的映射。

修改hadoop配置：yarn-site.xml 、hdfs-site.xml capacity-scheduler.xml

workers

修改spark配置：spark-defaults.conf 、spark-env.sh 、workers

将上述配置通过scp分发到集群

Spark on yarn 集群搭建

集群参数

集群节点配置：

节点数量 15

节点内存： 35

节点虚拟内核数： 3

spark on yarn参数：

--num-executors： 20

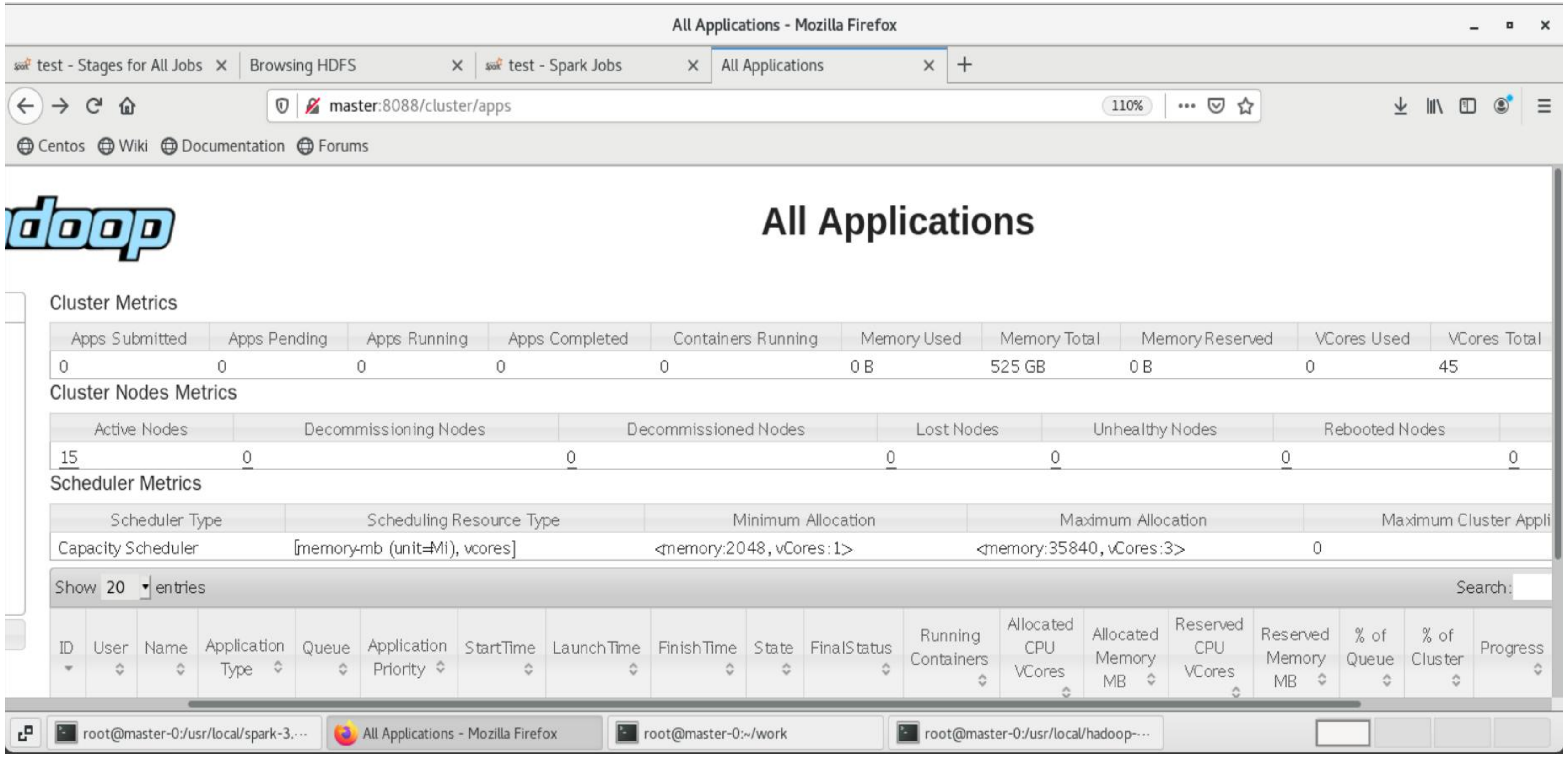
--executor-memory: 16G

--executor-cores： 2

--driver-cores 2

--driver-memory 16G

--使用G1 CG垃圾回收技术


All Applications - Mozilla Firefox
test - Stages for All Jobs
Browsing HDFS
test - Spark Jobs
All Applications
master:8088/cluster/apps
110%
Centos
Wiki
Documentation
Forums
doop
root@master-0:/usr/local/spark-3.···
All Applications - Mozilla Firefox
root@master-0:~/work
root@master-0:/usr/local/hadoop-···

All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total
0	0	0	0	0	0 B	525 GB	0 B	0	45

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	
15	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Appli
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:2048, vCores:1>	<memory:35840, vCores:3>	0

Show 20 entries Search:

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

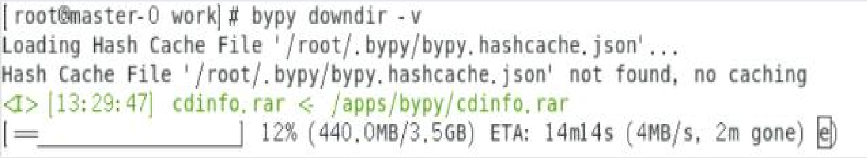
FOUR

流程详述

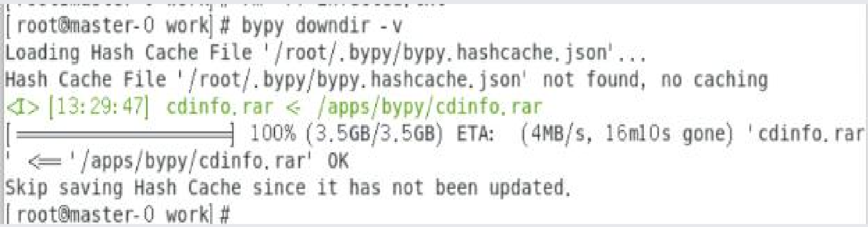
Process description

数据导入

利用python的bypy库将文件导入到平台



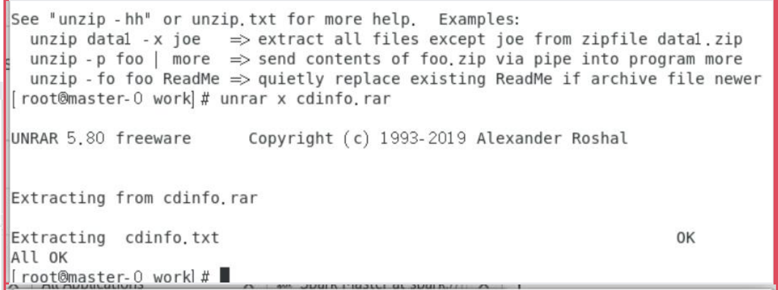
```
[root@master-0 work]# bypy downdir -v
Loading Hash Cache File '/root/.bypy/bypy.hashcache.json'...
Hash Cache File '/root/.bypy/bypy.hashcache.json' not found, no caching
<I> [13:29:47] cdinfo.rar < /apps/bypy/cdinfo.rar
[=                ] 12% (440.0MB/3.5GB) ETA: 14m14s (4MB/s, 2m gone) e
```



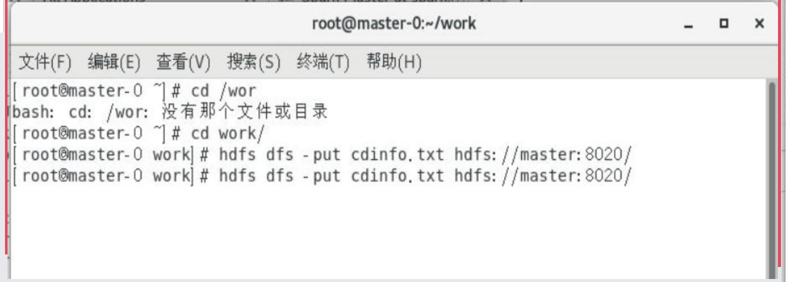
```
[root@master-0 work]# bypy downdir -v
Loading Hash Cache File '/root/.bypy/bypy.hashcache.json'...
Hash Cache File '/root/.bypy/bypy.hashcache.json' not found, no caching
<I> [13:29:47] cdinfo.rar < /apps/bypy/cdinfo.rar
[=================] 100% (3.5GB/3.5GB) ETA:  (4MB/s, 16m10s gone) 'cdinfo.rar
'  <== '/apps/bypy/cdinfo.rar' OK
Skip saving Hash Cache since it has not been updated.
[root@master-0 work]#
```

将文件传入hdfs

将文件解压，传入hdfs

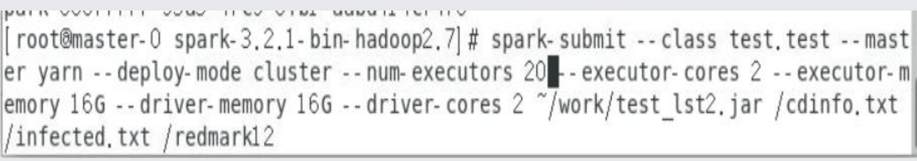


```
See "unzip -hh" or unzip.txt for more help.  Examples:
  unzip data1 -x joe   => extract all files except joe from zipfile data1.zip
  unzip -p foo | more  => send contents of foo.zip via pipe into program more
  unzip -fo foo ReadMe => quietly replace existing ReadMe if archive file newer
[root@master-0 work]# unrar x cdinfo.rar

UNRAR 5.80 freeware      Copyright (c) 1993-2019 Alexander Roshal


Extracting from cdinfo.rar

Extracting  cdinfo.txt                                              OK
All OK
[root@master-0 work]# 
```

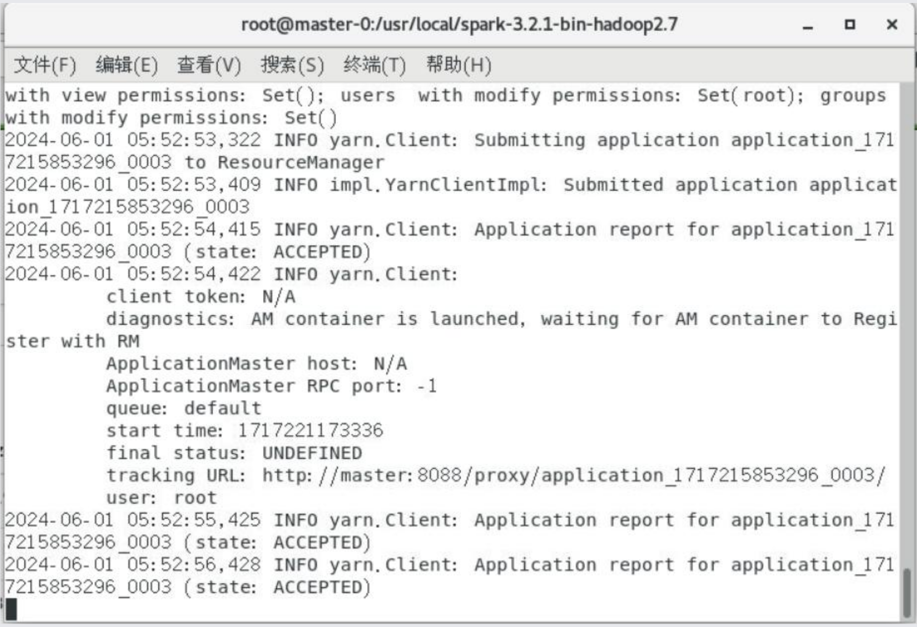


```
[root@master-0 ~]# cd /wor
bash: cd: /wor: 没有那个文件或目录
[root@master-0 ~]# cd work/
[root@master-0 work]# hdfs dfs -put cdinfo.txt hdfs://master:8020/
[root@master-0 work]# hdfs dfs -put cdinfo.txt hdfs://master:8020/
```

启动spark on yarn集群



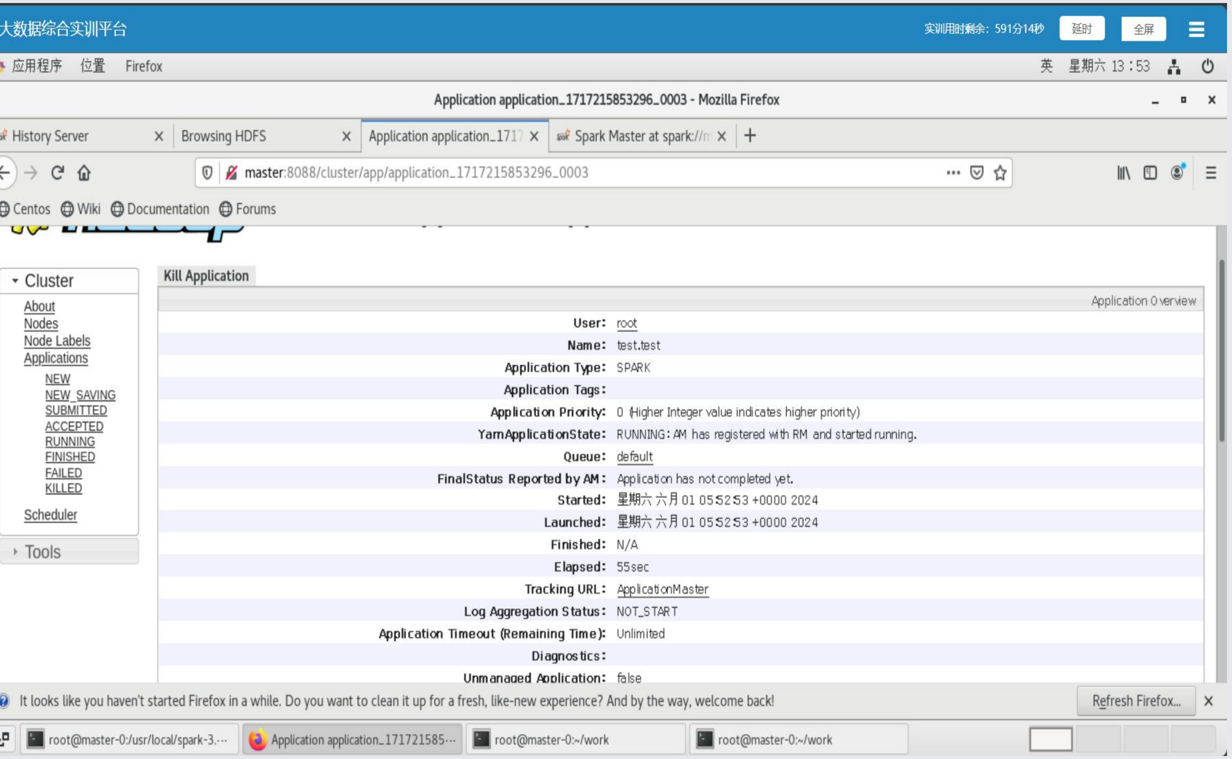
```
[root@master-0 spark-3.2.1-bin-hadoop2.7]# spark-submit --class test.test --mast
er yarn --deploy-mode cluster --num-executors 20 --executor-cores 2 --executor-m
emory 16G --driver-memory 16G --driver-cores 2 ~/work/test_lst2.jar /cdinfo.txt
/infected.txt /redmark12
```



root@master-0:/usr/local/spark-3.2.1-bin-hadoop2.7
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
with view permissions: Set(); users  with modify permissions: Set(root); groups
with modify permissions: Set()
2024-06-01 05:52:53,322 INFO yarn.Client: Submitting application application_171
7215853296_0003 to ResourceManager
2024-06-01 05:52:53,409 INFO impl.YarnClientImpl: Submitted application applicat
ion_1717215853296_0003
2024-06-01 05:52:54,415 INFO yarn.Client: Application report for application_171
7215853296_0003 (state: ACCEPTED)
2024-06-01 05:52:54,422 INFO yarn.Client:
         client token: N/A
         diagnostics: AM container is launched, waiting for AM container to Regi
ster with RM
         ApplicationMaster host: N/A
         ApplicationMaster RPC port: -1
         queue: default
         start time: 1717221173336
         final status: UNDEFINED
         tracking URL: http://master:8088/proxy/application_1717215853296_0003/
         user: root
2024-06-01 05:52:55,425 INFO yarn.Client: Application report for application_171
7215853296_0003 (state: ACCEPTED)
2024-06-01 05:52:56,428 INFO yarn.Client: Application report for application_171
7215853296_0003 (state: ACCEPTED)
```

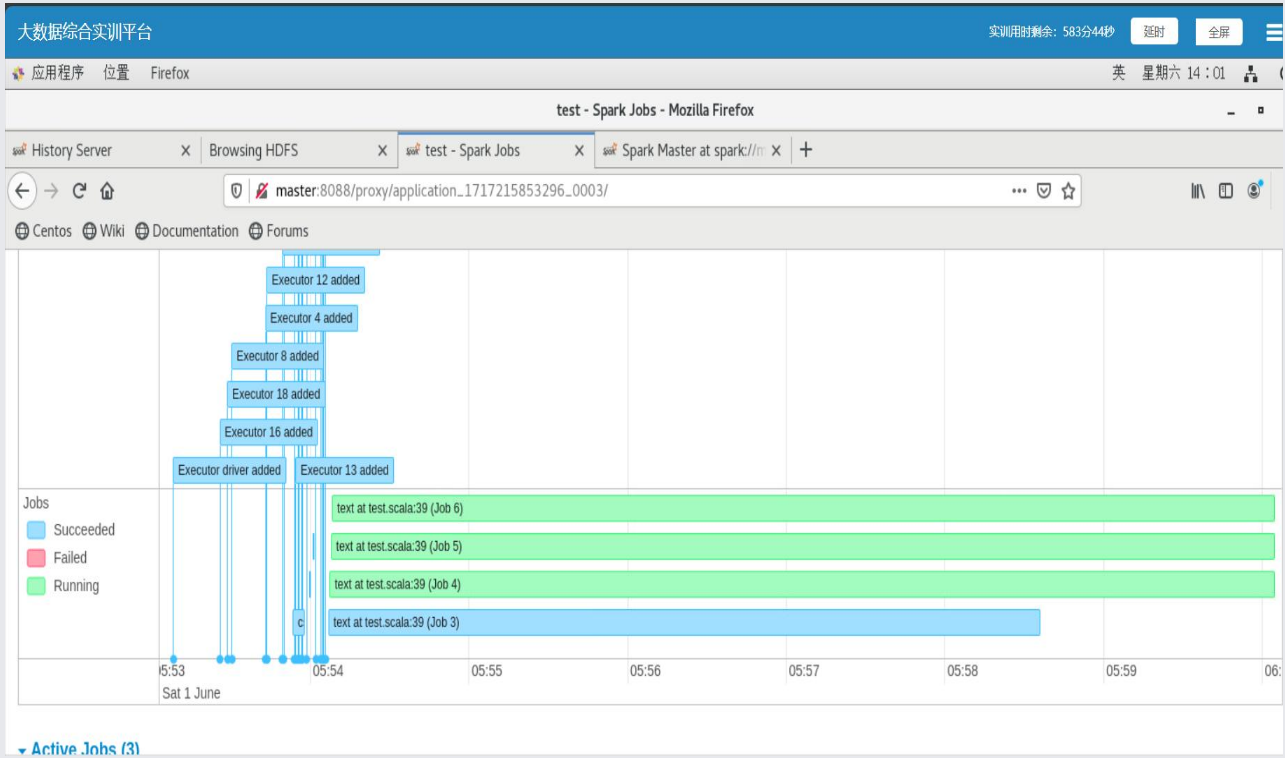
集群开始工作



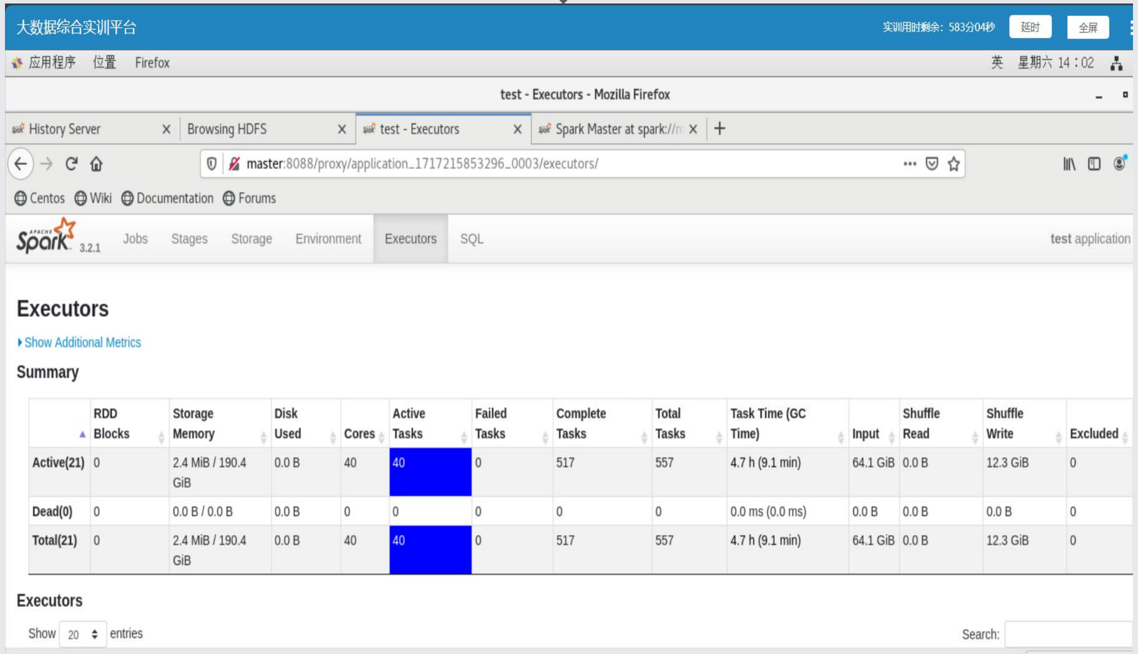
User:	root
Name:	test.test
Application Type:	SPARK
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	RUNNING: AM has registered with RM and started running.
Queue:	default
FinalStatus Reported by AM:	Application has not completed yet.
Started:	星期六 六月 01 05:52:53 +0000 2024
Launched:	星期六 六月 01 05:52:53 +0000 2024
Finished:	N/A
Elapsed:	55sec
Tracking URL:	ApplicationMaster
Log Aggregation Status:	NOT_START
Application Timeout (Remaining Time):	Unlimited
Diagnostics:	
Unmanaged Application:	false

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Job情况



集群运行过程



Executors

▸ Show Additional Metrics

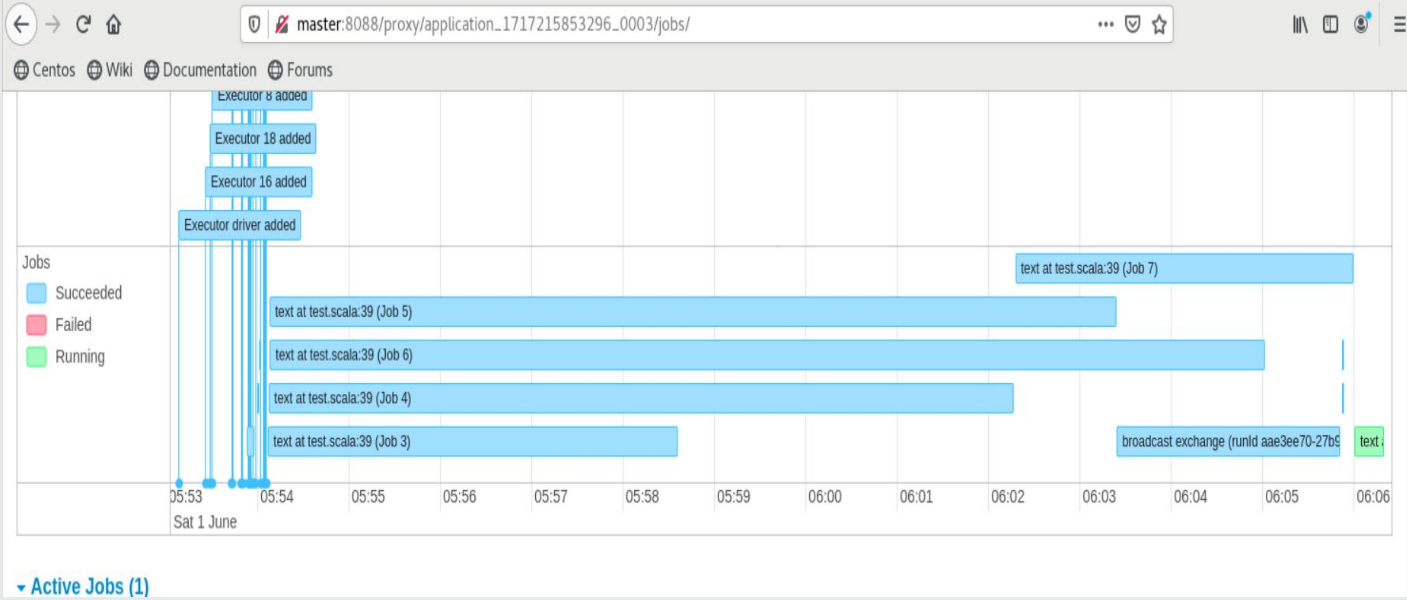
Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(21)	0	2.4 MiB / 190.4 GiB	0.0 B	40	40	0	517	557	4.7 h (9.1 min)	64.1 GiB	0.0 B	12.3 GiB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(21)	0	2.4 MiB / 190.4 GiB	0.0 B	40	40	0	517	557	4.7 h (9.1 min)	64.1 GiB	0.0 B	12.3 GiB	0

Executors

Show 20 entries

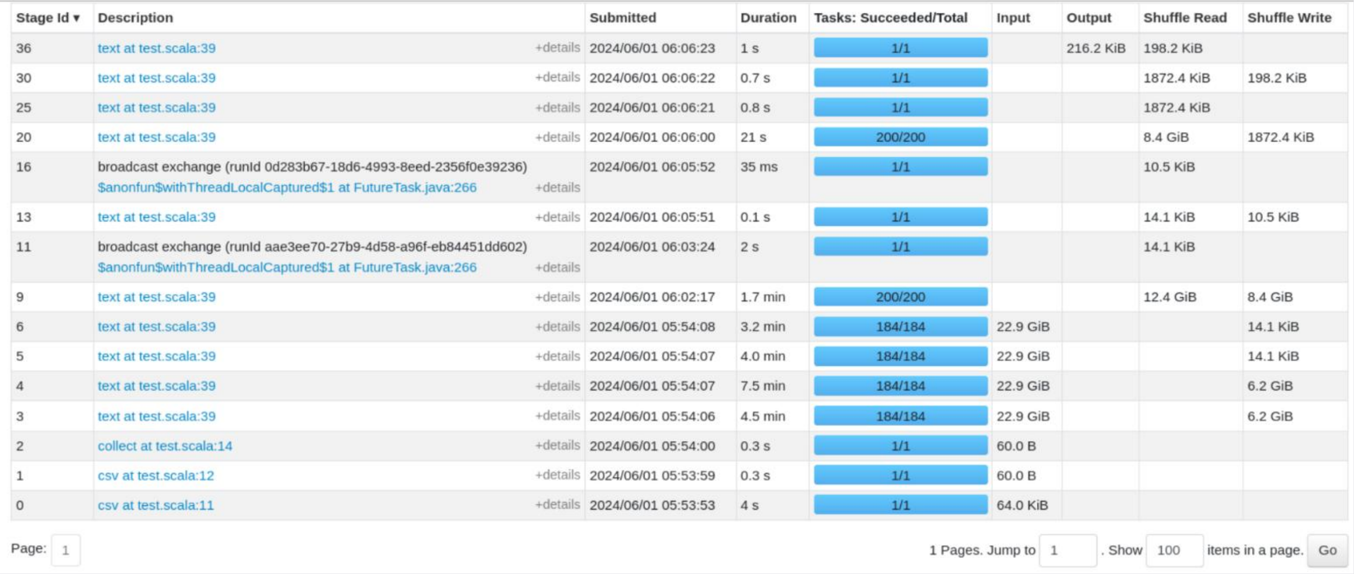
Search:



master:8088/proxy/application_1717215853296_0003/jobs/

▾ Active Jobs (1)

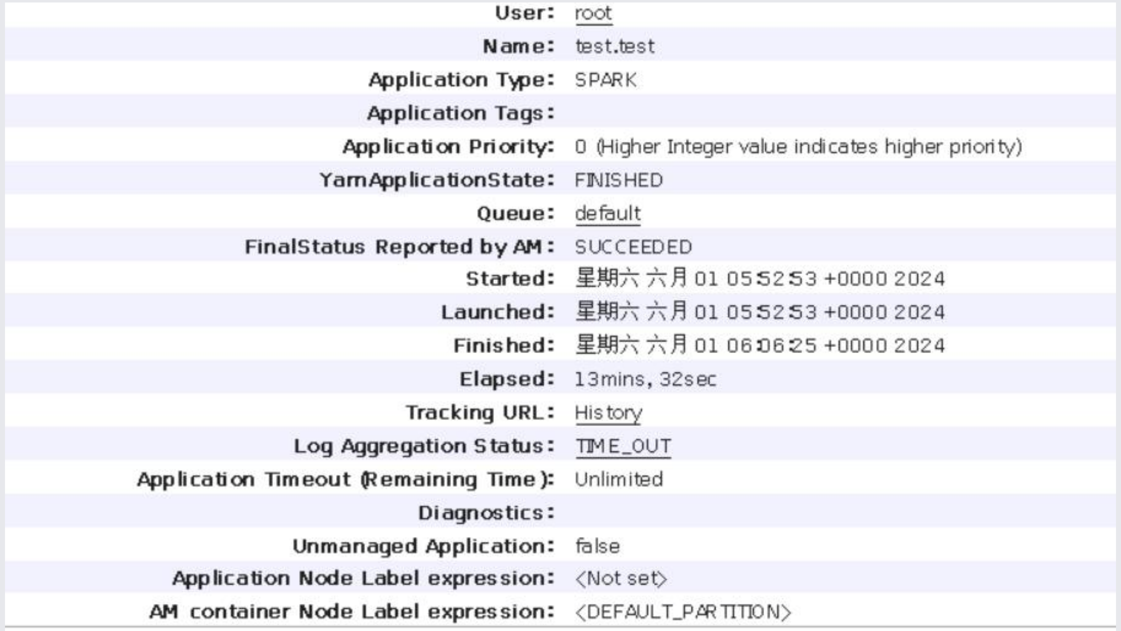
集群运行完成



Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
36	text at test.scala:39 +details	2024/06/01 06:06:23	1 s	1/1		216.2 KiB	198.2 KiB	
30	text at test.scala:39 +details	2024/06/01 06:06:22	0.7 s	1/1			1872.4 KiB	198.2 KiB
25	text at test.scala:39 +details	2024/06/01 06:06:21	0.8 s	1/1			1872.4 KiB	
20	text at test.scala:39 +details	2024/06/01 06:06:00	21 s	200/200			8.4 GiB	1872.4 KiB
16	broadcast exchange (runId 0d283b67-18d6-4993-8eed-2356f0e39236) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266 +details	2024/06/01 06:05:52	35 ms	1/1			10.5 KiB	
13	text at test.scala:39 +details	2024/06/01 06:05:51	0.1 s	1/1			14.1 KiB	10.5 KiB
11	broadcast exchange (runId aae3ee70-27b9-4d58-a96f-eb84451dd602) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266 +details	2024/06/01 06:03:24	2 s	1/1			14.1 KiB	
9	text at test.scala:39 +details	2024/06/01 06:02:17	1.7 min	200/200			12.4 GiB	8.4 GiB
6	text at test.scala:39 +details	2024/06/01 05:54:08	3.2 min	184/184	22.9 GiB			14.1 KiB
5	text at test.scala:39 +details	2024/06/01 05:54:07	4.0 min	184/184	22.9 GiB			14.1 KiB
4	text at test.scala:39 +details	2024/06/01 05:54:07	7.5 min	184/184	22.9 GiB			6.2 GiB
3	text at test.scala:39 +details	2024/06/01 05:54:06	4.5 min	184/184	22.9 GiB			6.2 GiB
2	collect at test.scala:14 +details	2024/06/01 05:54:00	0.3 s	1/1	60.0 B			
1	csv at test.scala:12 +details	2024/06/01 05:53:59	0.3 s	1/1	60.0 B			
0	csv at test.scala:11 +details	2024/06/01 05:53:53	4 s	1/1	64.0 KiB			

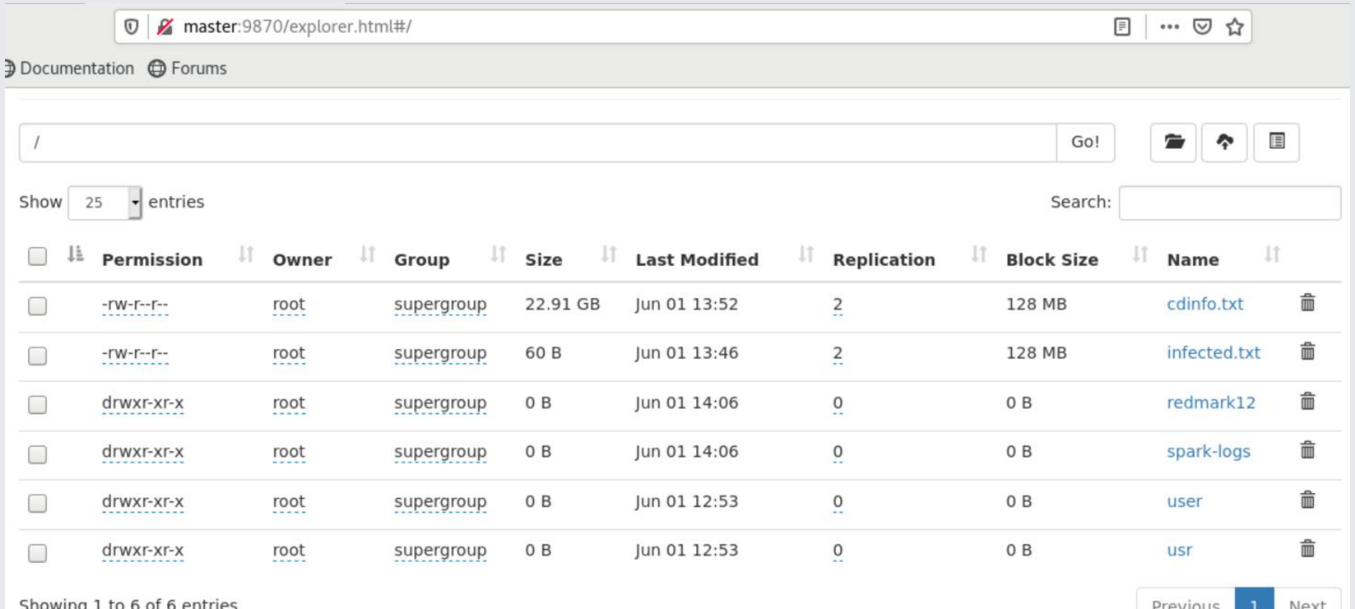
Page: 1

1 Pages. Jump to 1 . Show 100 items in a page. Go



User:	root
Name:	test.test
Application Type:	SPARK
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	default
FinalStatus Reported by AM:	SUCCEEDED
Started:	星期六 六月 01 05:52:53 +0000 2024
Launched:	星期六 六月 01 05:52:53 +0000 2024
Finished:	星期六 六月 01 06:06:25 +0000 2024
Elapsed:	13mins, 32sec
Tracking URL:	History
Log Aggregation Status:	TIME_OUT
Application Timeout (Remaining Time):	Unlimited
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>

Hdfs生成文件并导出



/

Show 25 entries

Search:

☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
☐	-rw-r--r--	root	supergroup	22.91 GB	Jun 01 13:52	2	128 MB	cdinfo.txt
☐	-rw-r--r--	root	supergroup	60 B	Jun 01 13:46	2	128 MB	infected.txt
☐	drwxr-xr-x	root	supergroup	0 B	Jun 01 14:06	0	0 B	redmark12
☐	drwxr-xr-x	root	supergroup	0 B	Jun 01 14:06	0	0 B	spark-logs
☐	drwxr-xr-x	root	supergroup	0 B	Jun 01 12:53	0	0 B	user
☐	drwxr-xr-x	root	supergroup	0 B	Jun 01 12:53	0	0 B	usr

Showing 1 to 6 of 6 entries

Previous 1 Next

谢谢观看