实验六 spark 的配置和测试

这一周的实验需要完成 spark 的配置及测试。Spark 是专为大规模数据处理而设计的快速通用的计算引擎，有与 hadoop 相似的开源集群计算环境，但是能在某些工作负载方面表现得更加优越，换句话说，spark 启用了内存分布数据集，除了能够提供交互式查询外，它还可以优化迭代工作负载。

spark 有三种模式，分别是 standalone、spark on yarn，spark on mesos，这里我们选择 spark on yarn 的模式，即在我们之前部署的 hadoop 集群上部署 spark。

【操作前请先阅读】:

1、 具体的内存配置请依据自己的电脑合适修改，不要直接套 pdf 数据。我的电脑是 1G 运行内存，所以设置的是 1000M，executor cores 是 2（经过测试 driver memory 和 worker memory 如果小于 512M，executor cores 大于 2 的话用 spark 提交任务会很容易就崩死，提示的是 SparkContext 相关错误）。

2、 关于 spark 的概念、运行机制、spark on yarn 的内容，请阅读下面两个链接内容

http://blog.csdn.net/wwwxxdddx/article/details/51087188 spark 基本概念

http://www.aboutyun.com/thread-12294-1-1.html spark on yarn 的两种运行模式

3、操作命令请大家尽量自己手敲，不要直接复制粘贴，便于发现问题。

一、以下步骤是在 master 主机上完成的，完成后可以直接将整个 spark 文件夹传给 slave。

1、将 spark-1.6.0-bin-hadoop2.6.tgz 放到/home/hadoop/目录下

2、解压文件到/usr/local 下，重命名文件夹并修改属主

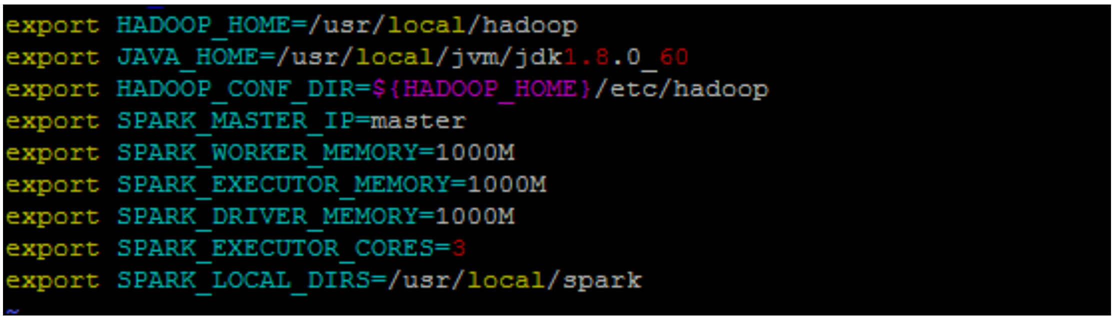
```
sudo tar -xzvf spark-1.6.0-bin-hadoop2.6.tgz -C /usr/local/
sudo mv /usr/local/spark-1.6.0-bin-hadoop2.6 /usr/local/spark
sudo chown -R hadoop:hadoop /usr/local/spark/
```

3、利用 spark 的 template 文件生成配置文件

```
cp /usr/local/spark/conf/spark-env.sh.template /usr/local/spark/conf/spark-env.sh
cp /usr/local/spark/conf/slaves.template /usr/local/spark/conf/slaves
cp /usr/local/spark/conf/spark-defaults.conf.template /usr/local/spark/conf/spark-defaults.conf
```

4、修改 spark-env.sh，在文件末尾添加如下内容：

```
export HADOOP_HOME=/usr/local/hadoop
export JAVA_HOME=/usr/local/jvm/jdk1.8.0_60
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop
export SPARK_MASTER_IP=master
export SPARK_LOCAL_DIRS=/usr/local/spark
export SPARK_WORKER_MEMORY=1000M
export SPARK_EXECUTOR_MEMORY=1000M
export SPARK_DRIVER_MEMORY=1000M
export SPARK_EXECUTOR_CORES=2
```

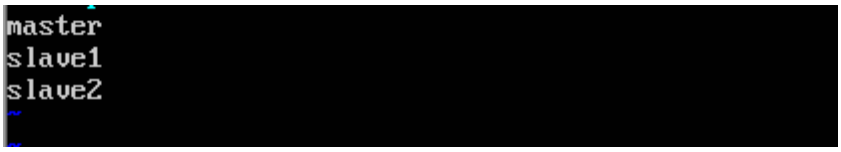


```
export HADOOP_HOME=/usr/local/hadoop
export JAVA_HOME=/usr/local/jvm/jdk1.8.0_60
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop
export SPARK_MASTER_IP=master
export SPARK_WORKER_MEMORY=1000M
export SPARK_EXECUTOR_MEMORY=1000M
export SPARK_DRIVER_MEMORY=1000M
export SPARK_EXECUTOR_CORES=3
export SPARK_LOCAL_DIRS=/usr/local/spark
~
```

这一步是为了配置 spark 的运行参数，hadoop_conf_dir 的设置是为了让 spark 运行在 yarn 上。几个 memory 命令分别用于设置 driver 和 executor 进程的内存，executor_cores 设置的是每个 executor 进程的 CPU cores 的数量，这些设置请依据自己的电脑实际可负载情况设置。

5、修改 slaves 文件，在文件末尾添加其他节点 IP。

```
vi /usr/local/spark/conf/slaves
```

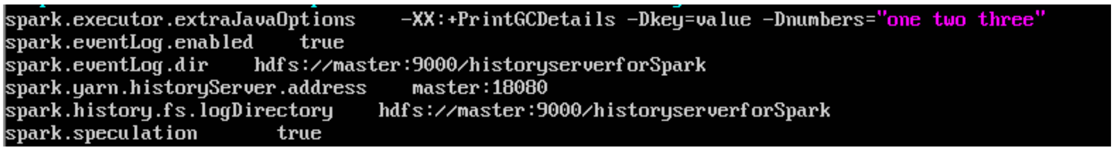


```
master
slave1
slave2
~
~
```

6、修改 spark-defaults.conf，在文件末尾添加如下内容：

```
vi /usr/local/spark/conf/spark-defaults.conf
```

```
spark.executor.extraJavaOptions -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"
spark.eventLog.enabled        true
spark.eventLog.dir        hdfs://master:9000/historyserverforSpark
spark.yarn.historyServer.address        master:18080
spark.history.fs.logDirectory        hdfs://master:9000/historyserverforSpark
spark.speculation              true
```



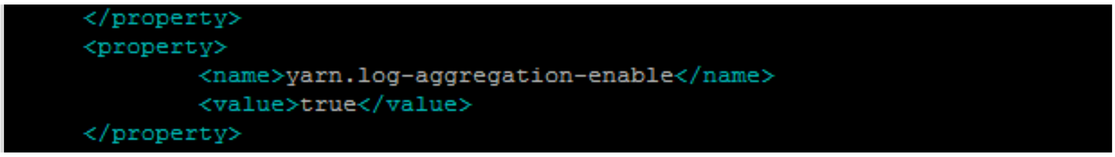
```
spark.executor.extraJavaOptions    -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"
spark.eventLog.enabled     true
spark.eventLog.dir     hdfs://master:9000/historyserverforSpark
spark.yarn.historyServer.address     master:18080
spark.history.fs.logDirectory     hdfs://master:9000/historyserverforSpark
spark.speculation        true
```

这一步是为保存 spark 的运行日志，并且是保存到 hdfs 上的文件夹里面，方便运维。

7、将配置好的 spark 文件夹传到 slave1、slave2。

二、配置 hadoop

修改 yarn-site.xml 文件，添加新的属性。



```
    </property>
    <property>
            <name>yarn.log-aggregation-enable</name>
            <value>true</value>
    </property>
```

这一步是为了开启日志整合功能，spark 在遇到问题时，利用命令

/usr/local/hadoop/bin/yarn logs -applicationId XXX

即可查看 applicationId 为 XXX 的实例的运行日志

三、运行 spark

1、运行 hadoop

```
/usr/local/hadoop/sbin/start-all.sh
```

2、在 spark 中创建 historyserverforSpark 文件夹

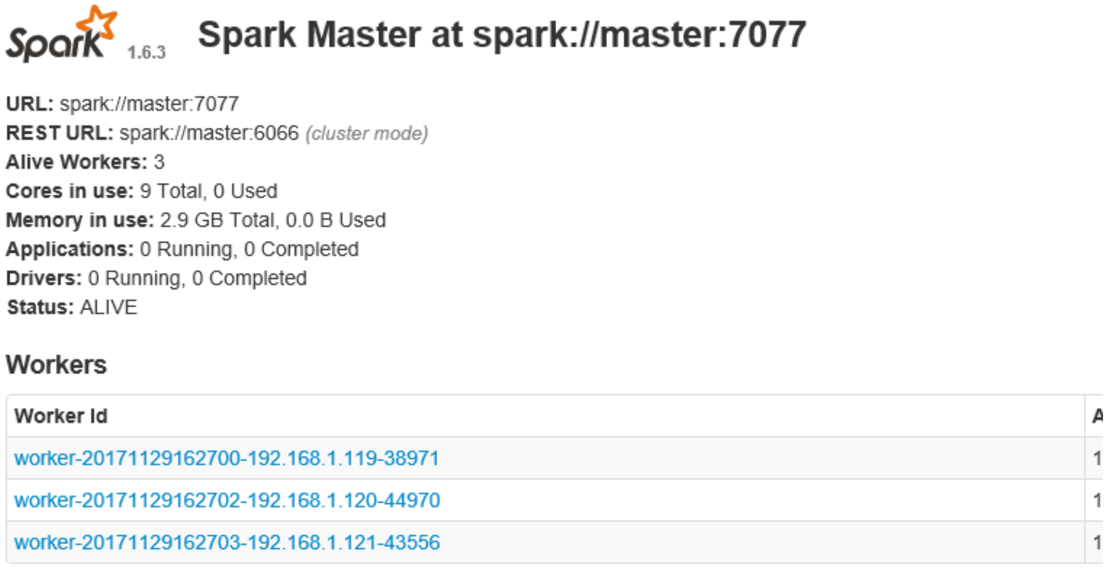
```
/usr/local/hadoop/bin/hdfs dfs -mkdir historyserverforSpark
```

3、运行 spark

```
/usr/local/spark/sbin/start-all.sh
```

可以进入 spark 的 webui 查看是否成功启动：

masterIP:8080/



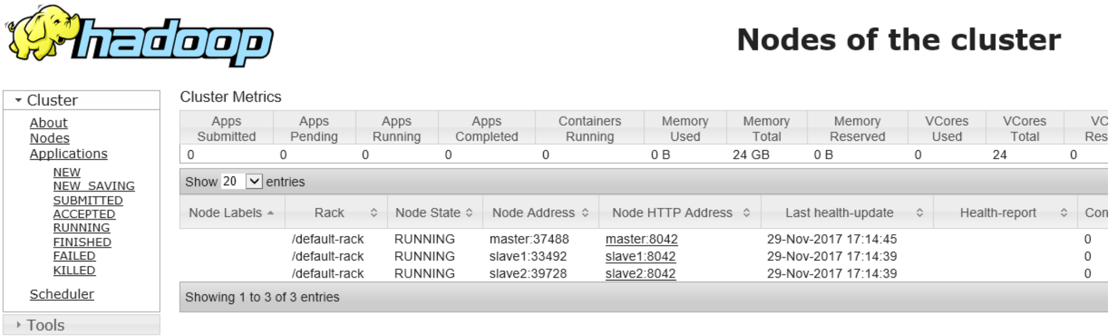
Spark Master at spark://master:7077

URL: spark://master:7077
REST URL: spark://master:6066 *(cluster mode)*
Alive Workers: 3
Cores in use: 9 Total, 0 Used
Memory in use: 2.9 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers

Worker Id	A
worker-20171129162700-192.168.1.119-38971	1
worker-20171129162702-192.168.1.120-44970	1
worker-20171129162703-192.168.1.121-43556	1

masterIP:8088/cluster



Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VC Res
0	0	0	0	0	0 B	24 GB	0 B	0	24	0

Show 20 entries

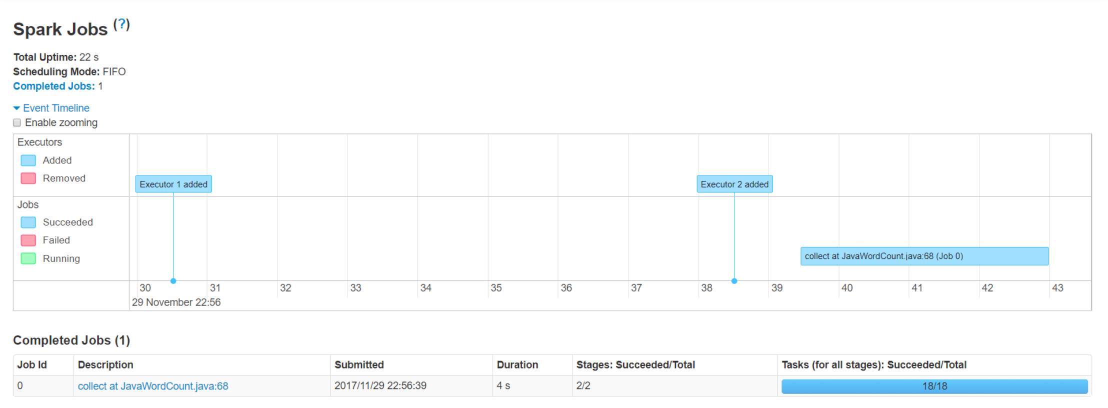
Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Con
	/default-rack	RUNNING	master:37488	master:8042	29-Nov-2017 17:14:45		0
	/default-rack	RUNNING	slave1:33492	slave1:8042	29-Nov-2017 17:14:39		0
	/default-rack	RUNNING	slave2:39728	slave2:8042	29-Nov-2017 17:14:39		0

Showing 1 to 3 of 3 entries

4、运行 history-server，这样应用运行完的结果可以通过 webui 看到。

```
usr/local/spark/sbin/start-history-server.sh
```

进入 masterIP：18080 页面，就能看到已经跑过的应用信息了



Spark Jobs (?)

Total Uptime: 22 s
Scheduling Mode: FIFO
Completed Jobs: 1

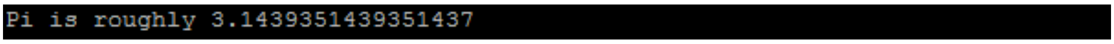
Completed Jobs (1)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	collect at JavaWordCount.java:68	2017/11/29 22:56:39	4 s	2/2	18/18

四、运行实例

1、运行本地模式实例

```
/usr/local/spark/bin/run-example SparkPi > SparkPiRes.txt
vi SparkPiRes.txt
```



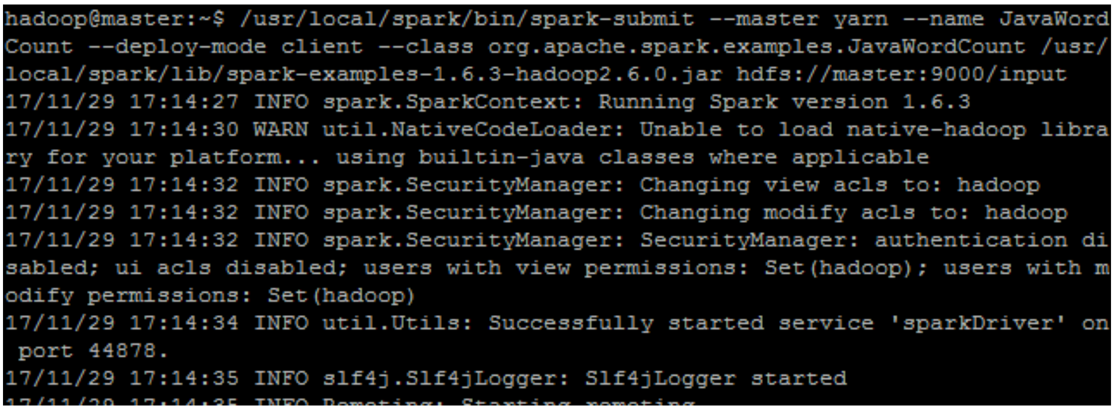
```
Pi is roughly 3.1439351439351437
```

2、在 yarn 上运行实例

```
/usr/local/spark/bin/spark-submit --master yarn --name JavaWordCount --deploy-mode client
--class org.apache.spark.examples.JavaWordCount
/usr/local/spark/lib/spark-examples-1.6.3-hadoop2.6.0.jar hdfs://master:9000/input/
```

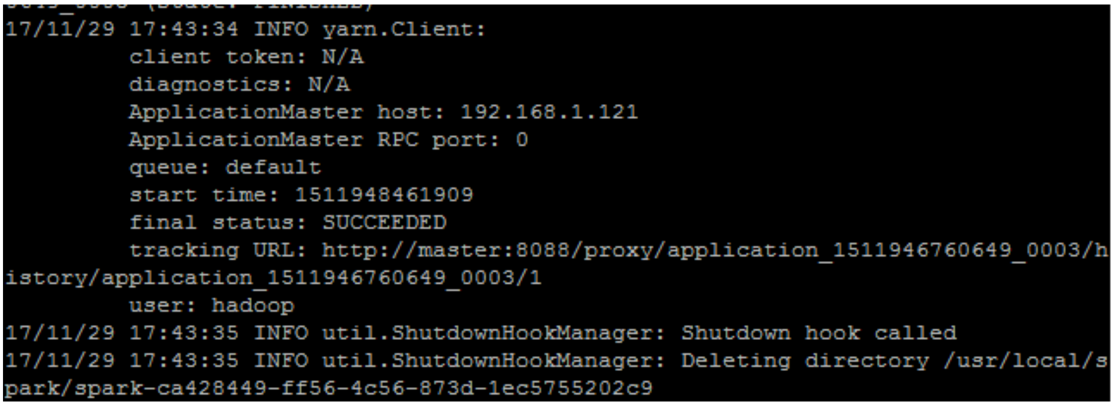
（这是一条命令）

这里用是 yarn client 的模式，spark 在 yarn 上运行还有 yarn cluster 模式。



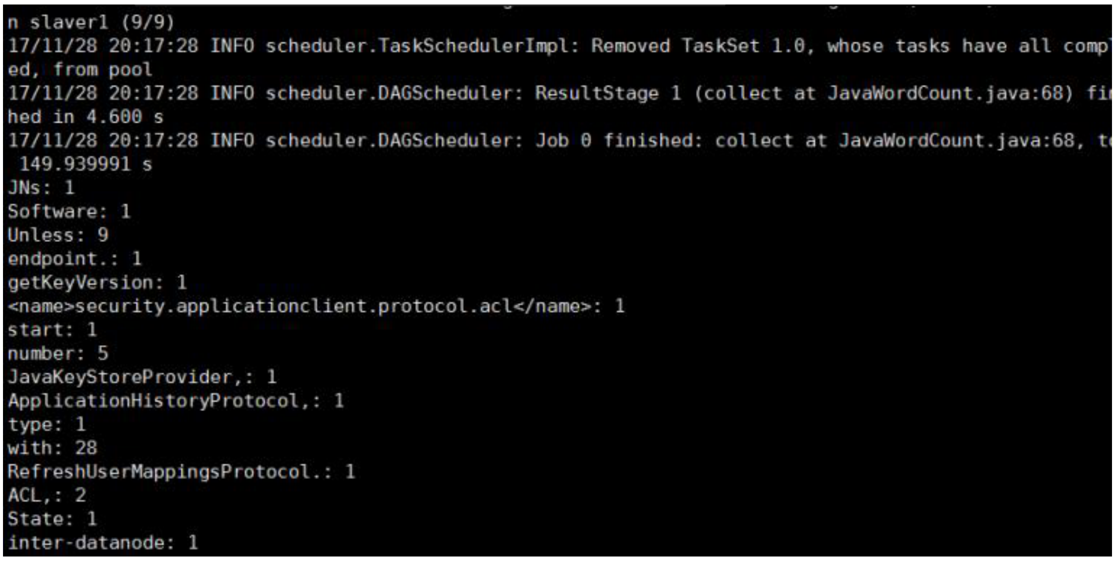
```
hadoop@master:~$ /usr/local/spark/bin/spark-submit --master yarn --name JavaWord
Count --deploy-mode client --class org.apache.spark.examples.JavaWordCount /usr/
local/spark/lib/spark-examples-1.6.3-hadoop2.6.0.jar hdfs://master:9000/input
17/11/29 17:14:27 INFO spark.SparkContext: Running Spark version 1.6.3
17/11/29 17:14:30 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
17/11/29 17:14:32 INFO spark.SecurityManager: Changing view acls to: hadoop
17/11/29 17:14:32 INFO spark.SecurityManager: Changing modify acls to: hadoop
17/11/29 17:14:32 INFO spark.SecurityManager: SecurityManager: authentication di
sabled; ui acls disabled; users with view permissions: Set(hadoop); users with m
odify permissions: Set(hadoop)
17/11/29 17:14:34 INFO util.Utils: Successfully started service 'sparkDriver' on
 port 44878.
17/11/29 17:14:35 INFO slf4j.Slf4jLogger: Slf4jLogger started
17/11/29 17:14:35 INFO Remoting: Starting remoting
```

最后运行结束结果



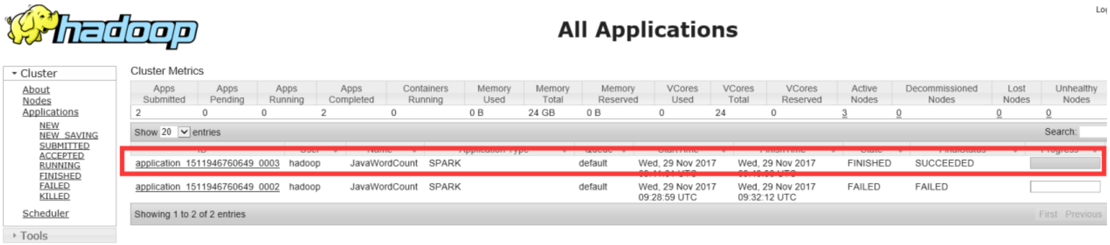
```
17/11/29 17:43:34 INFO yarn.Client:
         client token: N/A
         diagnostics: N/A
         ApplicationMaster host: 192.168.1.121
         ApplicationMaster RPC port: 0
         queue: default
         start time: 1511948461909
         final status: SUCCEEDED
         tracking URL: http://master:8088/proxy/application_1511946760649_0003/h
istory/application_1511946760649_0003/1
         user: hadoop
17/11/29 17:43:35 INFO util.ShutdownHookManager: Shutdown hook called
17/11/29 17:43:35 INFO util.ShutdownHookManager: Deleting directory /usr/local/s
park/spark-ca428449-ff56-4c56-873d-1ec5755202c9
```

wordcount 的结果在运行中可以看得到：



```
n slaver1 (9/9)
17/11/28 20:17:28 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all comp
ed, from pool
17/11/28 20:17:28 INFO scheduler.DAGScheduler: ResultStage 1 (collect at JavaWordCount.java:68) fi
hed in 4.600 s
17/11/28 20:17:28 INFO scheduler.DAGScheduler: Job 0 finished: collect at JavaWordCount.java:68, t
 149.939991 s
JNs: 1
Software: 1
Unless: 9
endpoint.: 1
getKeyVersion: 1
<name>security.applicationclient.protocol.acl</name>: 1
start: 1
number: 5
JavaKeyStoreProvider,: 1
ApplicationHistoryProtocol,: 1
type: 1
with: 28
RefreshUserMappingsProtocol.: 1
ACL,: 2
State: 1
inter-datanode: 1
```

在 web 页面上也可以看到



All Applications

- Cluster
  - About
  - Nodes
  - 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	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
2	0	0	2	0	0 B	24 GB	0 B	0	24	0	3	0	0	0

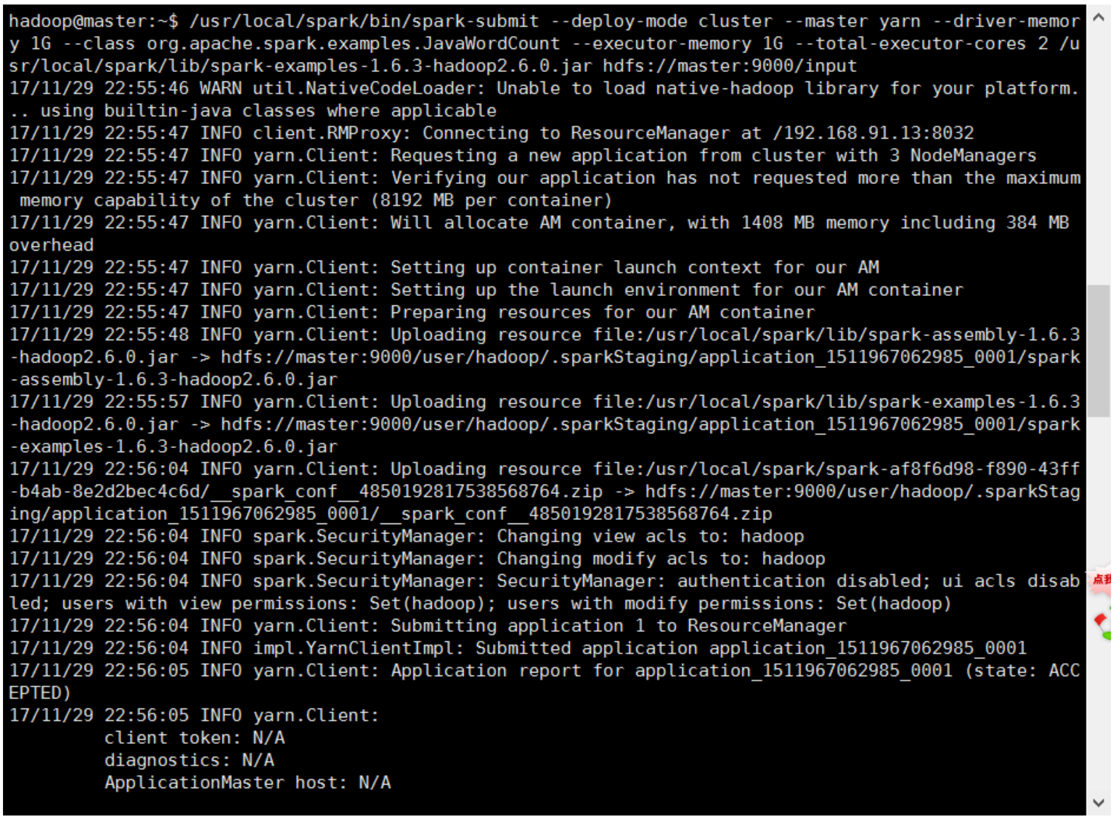
Show 20 entries Search:

application_1511946760649_0003	hadoop	JavaWordCount	SPARK	default	Wed, 29 Nov 2017	Wed, 29 Nov 2017	FINISHED	SUCCEEDED	
application_1511946760649_0002	hadoop	JavaWordCount	SPARK	default	Wed, 29 Nov 2017 09:28:59 UTC	Wed, 29 Nov 2017 09:32:12 UTC	FAILED	FAILED	

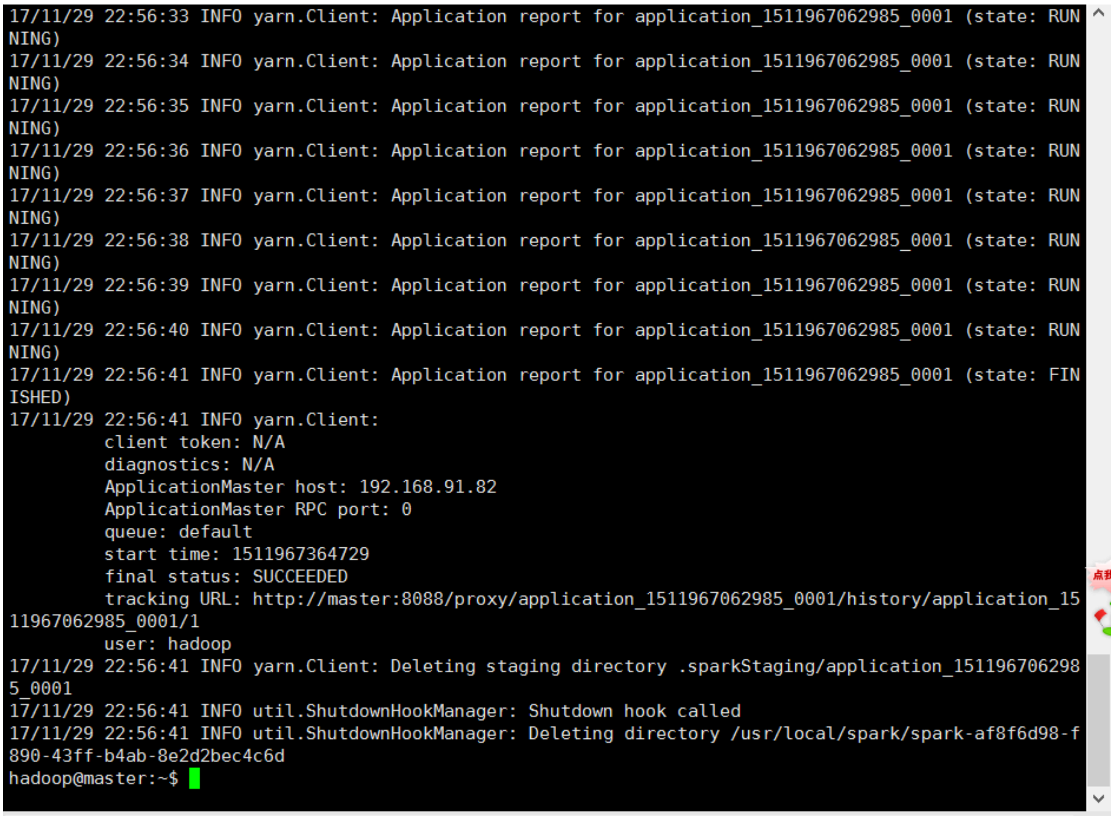
Showing 1 to 2 of 2 entries

First Previous

如果是使用 yarn-cluster 模式，在运行过程中不会看到统计结果：

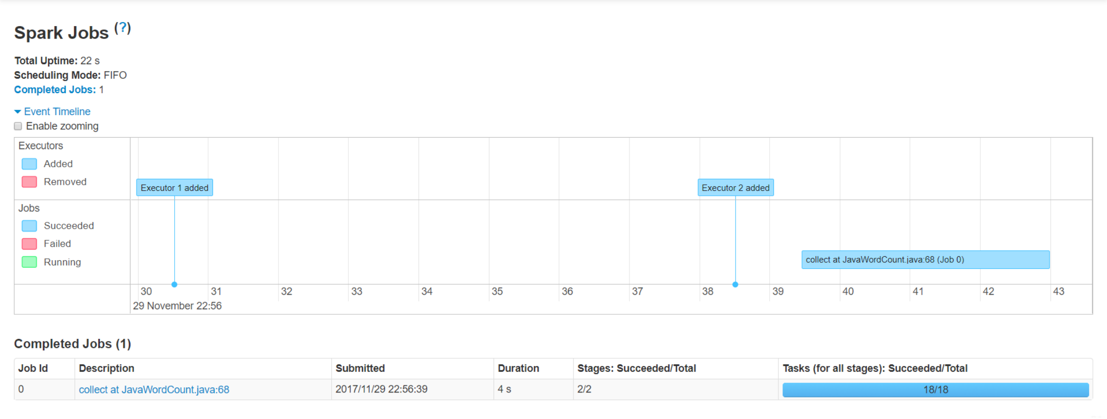


```
hadoop@master:~$ /usr/local/spark/bin/spark-submit --deploy-mode cluster --master yarn --driver-memor
y 1G --class org.apache.spark.examples.JavaWordCount --executor-memory 1G --total-executor-cores 2 /u
sr/local/spark/lib/spark-examples-1.6.3-hadoop2.6.0.jar hdfs://master:9000/input
17/11/29 22:55:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform.
.. using builtin-java classes where applicable
17/11/29 22:55:47 INFO client.RMProxy: Connecting to ResourceManager at /192.168.91.13:8032
17/11/29 22:55:47 INFO yarn.Client: Requesting a new application from cluster with 3 NodeManagers
17/11/29 22:55:47 INFO yarn.Client: Verifying our application has not requested more than the maximum
 memory capability of the cluster (8192 MB per container)
17/11/29 22:55:47 INFO yarn.Client: Will allocate AM container, with 1408 MB memory including 384 MB
overhead
17/11/29 22:55:47 INFO yarn.Client: Setting up container launch context for our AM
17/11/29 22:55:47 INFO yarn.Client: Setting up the launch environment for our AM container
17/11/29 22:55:47 INFO yarn.Client: Preparing resources for our AM container
17/11/29 22:55:48 INFO yarn.Client: Uploading resource file:/usr/local/spark/lib/spark-assembly-1.6.3
-hadoop2.6.0.jar -> hdfs://master:9000/user/hadoop/.sparkStaging/application_1511967062985_0001/spark
-assembly-1.6.3-hadoop2.6.0.jar
17/11/29 22:55:57 INFO yarn.Client: Uploading resource file:/usr/local/spark/lib/spark-examples-1.6.3
-hadoop2.6.0.jar -> hdfs://master:9000/user/hadoop/.sparkStaging/application_1511967062985_0001/spark
-examples-1.6.3-hadoop2.6.0.jar
17/11/29 22:56:04 INFO yarn.Client: Uploading resource file:/usr/local/spark/spark-af8f6d98-f890-43ff
-b4ab-8e2d2bec4c6d/__spark_conf__4850192817538568764.zip -> hdfs://master:9000/user/hadoop/.sparkStag
ing/application_1511967062985_0001/__spark_conf__4850192817538568764.zip
17/11/29 22:56:04 INFO spark.SecurityManager: Changing view acls to: hadoop
17/11/29 22:56:04 INFO spark.SecurityManager: Changing modify acls to: hadoop
17/11/29 22:56:04 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disab
led; users with view permissions: Set(hadoop); users with modify permissions: Set(hadoop)
17/11/29 22:56:04 INFO yarn.Client: Submitting application 1 to ResourceManager
17/11/29 22:56:04 INFO impl.YarnClientImpl: Submitted application application_1511967062985_0001
17/11/29 22:56:05 INFO yarn.Client: Application report for application_1511967062985_0001 (state: ACC
EPTED)
17/11/29 22:56:05 INFO yarn.Client:
         client token: N/A
         diagnostics: N/A
         ApplicationMaster host: N/A
```



```
17/11/29 22:56:33 INFO yarn.Client: Application report for application_1511967062985_0001 (state: RUN
NING)
17/11/29 22:56:34 INFO yarn.Client: Application report for application_1511967062985_0001 (state: RUN
NING)
17/11/29 22:56:35 INFO yarn.Client: Application report for application_1511967062985_0001 (state: RUN
NING)
17/11/29 22:56:36 INFO yarn.Client: Application report for application_1511967062985_0001 (state: RUN
NING)
17/11/29 22:56:37 INFO yarn.Client: Application report for application_1511967062985_0001 (state: RUN
NING)
17/11/29 22:56:38 INFO yarn.Client: Application report for application_1511967062985_0001 (state: RUN
NING)
17/11/29 22:56:39 INFO yarn.Client: Application report for application_1511967062985_0001 (state: RUN
NING)
17/11/29 22:56:40 INFO yarn.Client: Application report for application_1511967062985_0001 (state: RUN
NING)
17/11/29 22:56:41 INFO yarn.Client: Application report for application_1511967062985_0001 (state: FIN
ISHED)
17/11/29 22:56:41 INFO yarn.Client:
         client token: N/A
         diagnostics: N/A
         ApplicationMaster host: 192.168.91.82
         ApplicationMaster RPC port: 0
         queue: default
         start time: 1511967364729
         final status: SUCCEEDED
         tracking URL: http://master:8088/proxy/application_1511967062985_0001/history/application_15
11967062985_0001/1
         user: hadoop
17/11/29 22:56:41 INFO yarn.Client: Deleting staging directory .sparkStaging/application_151196706298
5_0001
17/11/29 22:56:41 INFO util.ShutdownHookManager: Shutdown hook called
17/11/29 22:56:41 INFO util.ShutdownHookManager: Deleting directory /usr/local/spark/spark-af8f6d98-f
890-43ff-b4ab-8e2d2bec4c6d
hadoop@master:~$ 
```

但是在 masterIP:18080 webui 中可以看到运行的结果：



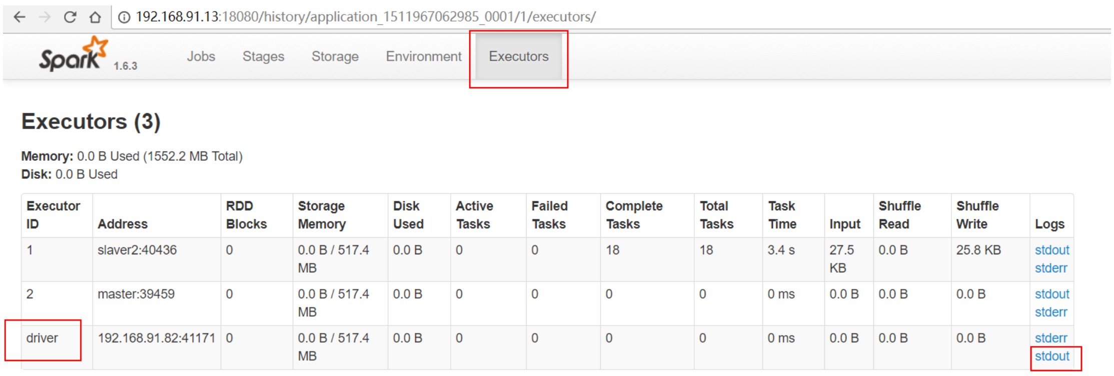
Spark Jobs (?)

Total Uptime: 22 s
Scheduling Mode: FIFO
Completed Jobs: 1

▾ Event Timeline
□ Enable zooming

Completed Jobs (1)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	collect at JavaWordCount.java:68	2017/11/29 22:56:39	4 s	2/2	18/18

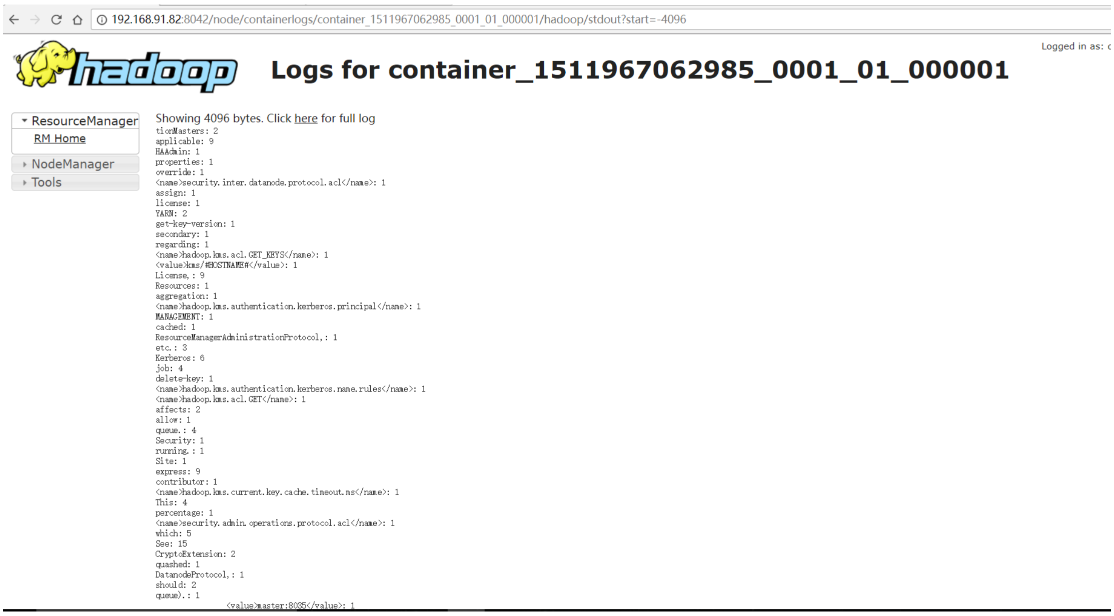


192.168.91.13:18080/history/application_1511967062985_0001/1/executors/

Executors (3)

Memory: 0.0 B Used (1552.2 MB Total)
Disk: 0.0 B Used

Executor ID	Address	RDD Blocks	Storage Memory	Disk Used	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time	Input	Shuffle Read	Shuffle Write	Logs
1	slaver2:40436	0	0.0 B / 517.4 MB	0.0 B	0	0	18	18	3.4 s	27.5 KB	0.0 B	25.8 KB	stdout stderr
2	master:39459	0	0.0 B / 517.4 MB	0.0 B	0	0	0	0	0 ms	0.0 B	0.0 B	0.0 B	stdout stderr
driver	192.168.91.82:41171	0	0.0 B / 517.4 MB	0.0 B	0	0	0	0	0 ms	0.0 B	0.0 B	0.0 B	stderr stdout


192.168.91.82:8042/node/containerlogs/container_1511967062985_0001_01_000001/hadoop/stdout?start=-4096
Logged in as: d
hadoop
ResourceManager
RM Home
NodeManager
Tools

Logs for container_1511967062985_0001_01_000001

Showing 4096 bytes. Click here for full log

```
tionMasters: 2
applicable: 9
HAAdmin: 1
properties: 1
override: 1
<name>security.inter.datanode.protocol.acl</name>: 1
assign: 1
license: 1
YARN: 2
get-key-version: 1
secondary: 1
regarding: 1
<name>hadoop.kms.acl.GET_KEYS</name>: 1
<value>kms/#HOSTNAME#</value>: 1
License,: 9
Resources: 1
aggregation: 1
<name>hadoop.kms.authentication.kerberos.principal</name>: 1
MANAGEMENT: 1
cached: 1
ResourceManagerAdministrationProtocol,: 1
etc.: 3
Kerberos: 6
job: 4
delete-key: 1
<name>hadoop.kms.authentication.kerberos.name.rules</name>: 1
<name>hadoop.kms.acl.GET</name>: 1
affects: 2
allow: 1
queue.: 4
Security: 1
running.: 1
Site: 1
express: 9
contributor: 1
<name>hadoop.kms.current.key.cache.timeout.ms</name>: 1
This: 4
percentage: 1
<name>security.admin.operations.protocol.acl</name>: 1
which: 5
See: 15
CryptoExtension: 2
quashed: 1
DatanodeProtocol,: 1
should: 2
queue).: 1
                <value>master:8035</value>: 1
```