2) 修改 conf/spark-env.sh，添加 JAVA_HOME 和 YARN_CONF_DIR 配置

```
mv spark-env.sh.template spark-env.sh
。。。
export JAVA_HOME=/opt/module/jdk1.8.0_144
YARN_CONF_DIR=/opt/module/hadoop/etc/hadoop
```

3.3.3 启动 HDFS 以及 YARN 集群

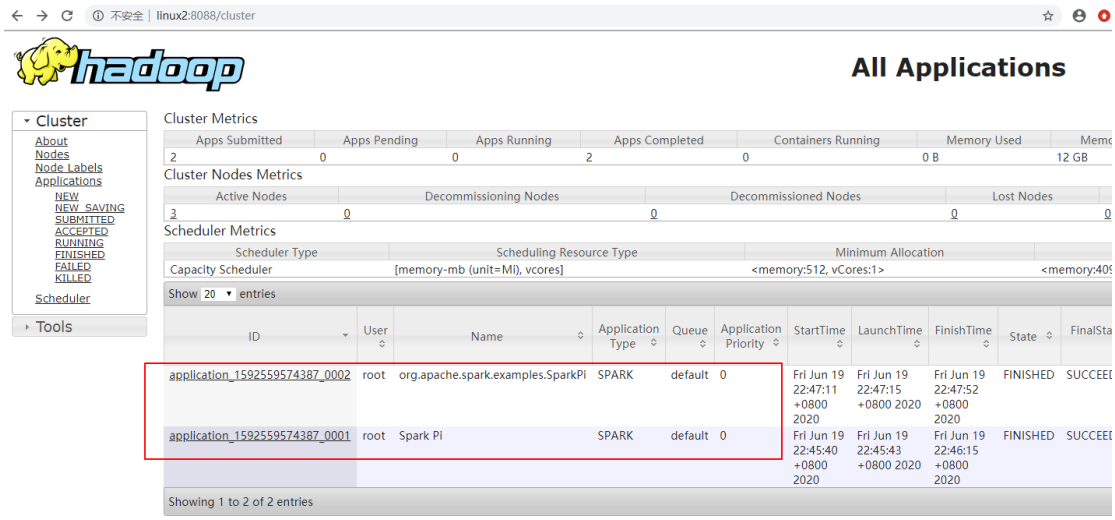
瞅啥呢，自己启动去！

3.3.4 提交应用

```
bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master yarn \
--deploy-mode cluster \
./examples/jars/spark-examples_2.12-3.0.0.jar \
10
```

```
2020-06-19 22:47:42,632 INFO yarn.Client: Application report for application_1592559574387_0002 (state: RUNNING)
2020-06-19 22:47:43,875 INFO yarn.Client: Application report for application_1592559574387_0002 (state: RUNNING)
2020-06-19 22:47:44,879 INFO yarn.Client: Application report for application_1592559574387_0002 (state: RUNNING)
2020-06-19 22:47:46,358 INFO yarn.Client: Application report for application_1592559574387_0002 (state: RUNNING)
2020-06-19 22:47:47,601 INFO yarn.Client: Application report for application_1592559574387_0002 (state: RUNNING)
2020-06-19 22:47:48,621 INFO yarn.Client: Application report for application_1592559574387_0002 (state: RUNNING)
2020-06-19 22:47:49,629 INFO yarn.Client: Application report for application_1592559574387_0002 (state: RUNNING)
2020-06-19 22:47:50,643 INFO yarn.Client: Application report for application_1592559574387_0002 (state: RUNNING)
2020-06-19 22:47:51,673 INFO yarn.Client: Application report for application_1592559574387_0002 (state: RUNNING)
2020-06-19 22:47:52,675 INFO yarn.Client: Application report for application_1592559574387_0002 (state: FINISHED)
2020-06-19 22:47:52,676 INFO yarn.Client:
        client token: N/A
        diagnostics: N/A
        ApplicationMaster host: linux1
        ApplicationMaster RPC port: 33892
        queue: default
        start time: 1592578031667
        final status: SUCCEEDED
        tracking URL: http://linux2:8088/proxy/application_1592559574387_0002/
        user: root
2020-06-19 22:47:52,726 INFO util.ShutdownHookManager: Shutdown hook called
2020-06-19 22:47:52,730 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-aedaddf7-b276-49ee-a1b6-d98e31ab52da
2020-06-19 22:47:52,736 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-4850a8e3-e949-490d-95cb-649f423d678d
```

查看 http://linux2:8088 页面，点击 History，查看历史页面



```
command:
  {{JAVA_HOME}}/bin/java \
    -server \
    -Xmx1024m \
    -Djava.io.tmpdir={{PWD}}/tmp \
    '-Dspark.driver.port=48441' \
    -Dspark.yarn.app.container.log.dir=<LOG_DIR> \
    -XX:OnOutOfMemoryError='kill %p' \
    org.apache.spark.executor.CoarseGrainedExecutorBackend \
    --driver-url \
    spark://CoarseGrainedScheduler@linux1:48441 \
    --executor-id \
    <executorId> \
    --hostname \
    <hostname> \
    --cores \
    1 \
    --app-id \
    application_1587439373824_0001 \
    --user-class-path \
    file:$PWD/__app__.jar \
    1><LOG_DIR>/stdout \
    2><LOG_DIR>/stderr

resources:
  __spark_libs__ -> resource { scheme: "hdfs" host: "linux1" port: 9000 file: "/user/root/.sparkStaging/application_1587439373824_0001/__spark_libs__2267073519899007339.zip" }
  __spark_conf__ -> resource { scheme: "hdfs" host: "linux1" port: 9000 file: "/user/root/.sparkStaging/application_1587439373824_0001/__spark_conf__.zip" } size: 193430 timest
```

3.3.5 配置历史服务器

1) 修改 spark-defaults.conf.template 文件名为 spark-defaults.conf

```
mv spark-defaults.conf.template spark-defaults.conf
```

2) 修改 spark-default.conf 文件，配置日志存储路径

```
spark.eventLog.enabled          true
spark.eventLog.dir              hdfs://linux1:8020/directory
```

注意：需要启动 hadoop 集群，HDFS 上的目录需要提前存在。

```
[root@linux1 hadoop]# sbin/start-dfs.sh
[root@linux1 hadoop]# hadoop fs -mkdir /directory
```

3) 修改 spark-env.sh 文件，添加日志配置

```
export SPARK_HISTORY_OPTS="
-Dspark.history.ui.port=18080
-Dspark.history.fs.logDirectory=hdfs://linux1:8020/directory
-Dspark.history.retainedApplications=30"
```

- 参数 1 含义：WEB UI 访问的端口号为 18080
- 参数 2 含义：指定历史服务器日志存储路径
- 参数 3 含义：指定保存 Application 历史记录的个数，如果超过这个值，旧的应用程序信息将被删除，这个是内存中的应用数，而不是页面上显示的应用数。

4) 修改 spark-defaults.conf

```
spark.yarn.historyServer.address=linux1:18080
spark.history.ui.port=18080
```

5) 启动历史服务

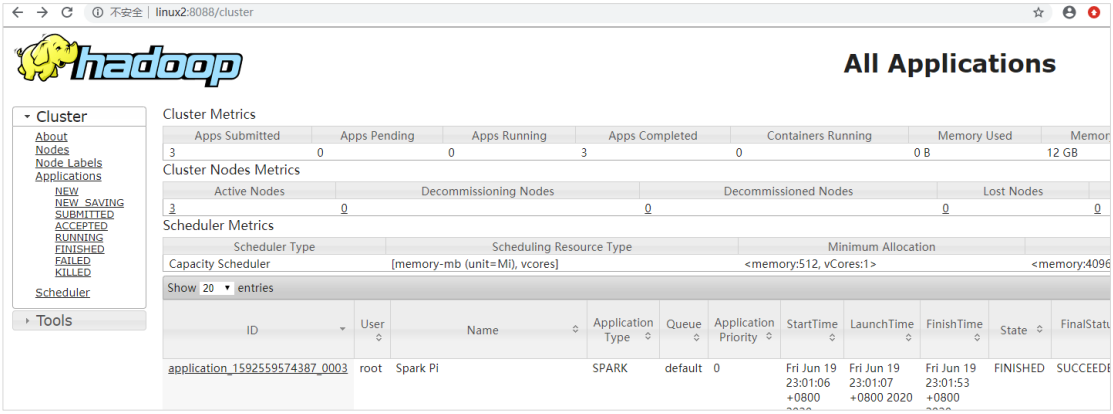
```
sbin/start-history-server.sh
```

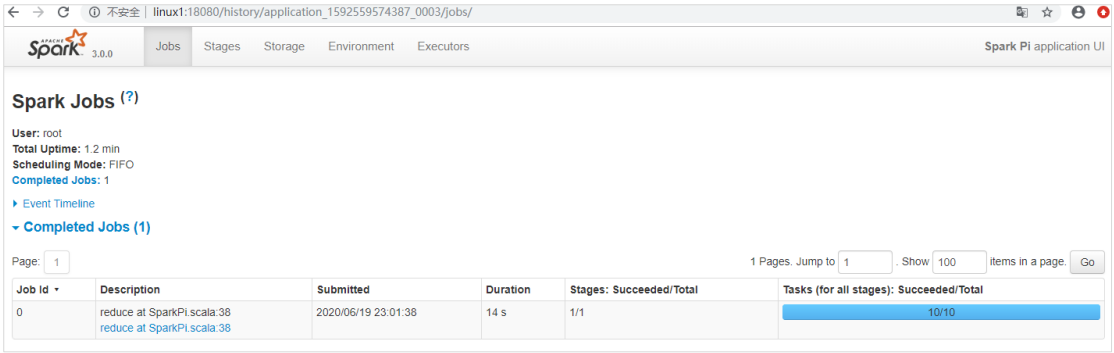
6) 重新提交应用

```
bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master yarn \
--deploy-mode client \
./examples/jars/spark-examples_2.12-3.0.0.jar \
10
```

```
2020-06-19 23:01:53,036 INFO cluster.YarnScheduler: Removed TaskSet 0.0, whose tasks have all completed, from pool
2020-06-19 23:01:53,038 INFO scheduler.DAGScheduler: ResultStage 0 (reduce at SparkPi.scala:38) finished in 14.375 s
2020-06-19 23:01:53,048 INFO scheduler.DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
2020-06-19 23:01:53,051 INFO cluster.YarnScheduler: Killing all running tasks in stage 0: Stage finished
2020-06-19 23:01:53,056 INFO scheduler.DAGScheduler: Job 0 finished: reduce at SparkPi.scala:38, took 14.488243 s
Pi is roughly 3.138131138131138
2020-06-19 23:01:53,088 INFO server.AbstractConnector: Stopped Spark@3e587920{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
2020-06-19 23:01:53,092 INFO ui.SparkUI: Stopped Spark web UI at http://linux1:4040
2020-06-19 23:01:53,103 INFO cluster.YarnClientSchedulerBackend: Interrupting monitor thread
2020-06-19 23:01:53,159 INFO cluster.YarnClientSchedulerBackend: Shutting down all executors
2020-06-19 23:01:53,160 INFO cluster.YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
2020-06-19 23:01:53,173 INFO cluster.YarnClientSchedulerBackend: YARN client scheduler backend Stopped
2020-06-19 23:01:53,308 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
2020-06-19 23:01:53,338 INFO memory.MemoryStore: MemoryStore cleared
2020-06-19 23:01:53,339 INFO storage.BlockManager: BlockManager stopped
2020-06-19 23:01:53,357 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
2020-06-19 23:01:53,362 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
```

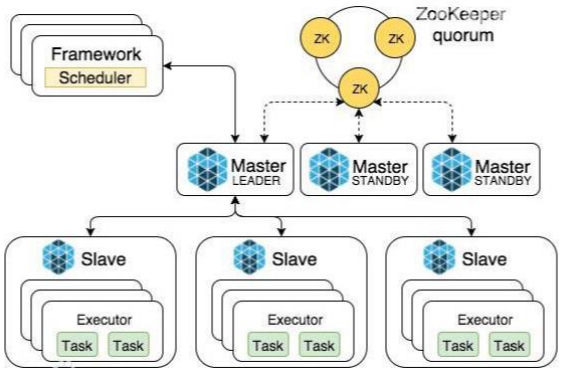
7) Web 页面查看日志：http://linux2:8088



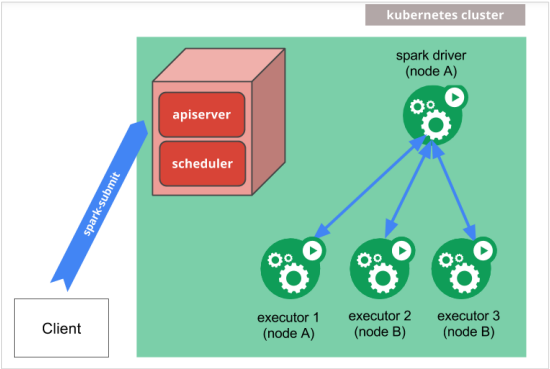


3.4 K8S & Mesos 模式

Mesos 是 Apache 下的开源分布式资源管理框架，它被称为是分布式系统的内核,在 Twitter 得到广泛使用,管理着 Twitter 超过 30,0000 台服务器上的应用部署，但是在国内，依然使用着传统的 Hadoop 大数据框架，所以国内使用 Mesos 框架的并不多，但是原理其实都差不多，这里我们就不做过多讲解了。



容器化部署是目前业界很流行的一项技术，基于 Docker 镜像运行能够让用户更加方便地对应用进行管理和运维。容器管理工具中最为流行的就是 Kubernetes（k8s），而 Spark 也在最近的版本中支持了 k8s 部署模式。这里我们也不做过多的讲解。给个链接大家自己感受一下：https://spark.apache.org/docs/latest/running-on-kubernetes.html



3.5 Windows 模式

在同学们自己学习时，每次都需要启动虚拟机，启动集群，这是一个比较繁琐的过程，并且会占大量的系统资源，导致系统执行变慢，不仅仅影响学习效果，也影响学习进度，Spark 非常暖心地提供了可以在 windows 系统下启动本地集群的方式，这样，在不使用虚拟机的情况下，也能学习 Spark 的基本使用，摸摸哒！

在后续的教学中，为了能够给同学们更加流畅的教学效果和教学体验，我们一般情况下都会采用 windows 系统的集群来学习 Spark。

3.5.1 解压缩文件

将文件 spark-3.0.0-bin-hadoop3.2.tgz 解压缩到无中文无空格的路径中

3.5.2 启动本地环境

1) 执行解压缩文件路径下 bin 目录中的 spark-shell.cmd 文件，启动 Spark 本地环境

```
20/06/19 23:38:06 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://windows10.microdone.cn:4040
Spark context available as 'sc' (master = local[*], app id = local-1592581095879).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.0.0
      /_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_111)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

2) 在 bin 目录中创建 input 目录，并添加 word.txt 文件，在命令行中输入脚本代码

```
20/06/19 23:43:33 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://windows10.microdone.cn:4040
Spark context available as 'sc' (master = local[*], app id = local-1592581423102).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.0.0
      /_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_111)
Type in expressions to have them evaluated.
Type :help for more information.

scala> sc.textFile("input/word.txt").flatMap(_.split(",")).map((_,1)).reduceByKey(_+_).collect20/06/19 23:43:55 WARN Pro
cfsMetricsGetter: Exception when trying to compute pagesize, as a result reporting of ProcessTree metrics is stopped

res0: Array[(String, Int)] = Array((world,1), (hello,1))
```

3.5.3 命令行提交应用

在 DOS 命令行窗口中执行提交指令

```
spark-submit --class org.apache.spark.examples.SparkPi --master local[2] ../examples/jars/spark-examples_2.12-3.0.0.jar 10
```

```
20/06/19 23:45:44 INFO Executor: Running task 9.0 in stage 0.0 (TID 9)
20/06/19 23:45:44 INFO TaskSetManager: Finished task 7.0 in stage 0.0 (TID 7) in 58 ms on windows10.microdone.cn (execut
or driver) (8/10)
20/06/19 23:45:44 INFO Executor: Finished task 8.0 in stage 0.0 (TID 8). 914 bytes result sent to driver
20/06/19 23:45:44 INFO TaskSetManager: Finished task 8.0 in stage 0.0 (TID 8) in 38 ms on windows10.microdone.cn (execut
or driver) (9/10)
20/06/19 23:45:44 INFO Executor: Finished task 9.0 in stage 0.0 (TID 9). 914 bytes result sent to driver
20/06/19 23:45:44 INFO TaskSetManager: Finished task 9.0 in stage 0.0 (TID 9) in 37 ms on windows10.microdone.cn (execut
or driver) (10/10)
20/06/19 23:45:44 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
20/06/19 23:45:44 INFO DAGScheduler: ResultStage 0 (reduce at SparkPi.scala:38) finished in 1.780 s
20/06/19 23:45:44 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
20/06/19 23:45:44 INFO TaskSchedulerImpl: Killing all running tasks in stage 0: Stage finished
20/06/19 23:45:44 INFO DAGScheduler: Job 0 finished: reduce at SparkPi.scala:38, took 1.853504 s
Pi is roughly 3.1391991391991394
20/06/19 23:45:44 INFO SparkUI: Stopped Spark web UI at http://windows10.microdone.cn:4040
20/06/19 23:45:44 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
20/06/19 23:45:44 INFO MemoryStore: MemoryStore cleared
20/06/19 23:45:44 INFO BlockManager: BlockManager stopped
20/06/19 23:45:44 INFO BlockManagerMaster: BlockManagerMaster stopped
20/06/19 23:45:44 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
20/06/19 23:45:44 INFO SparkContext: Successfully stopped SparkContext
20/06/19 23:45:44 INFO ShutdownHookManager: Shutdown hook called
20/06/19 23:45:44 INFO ShutdownHookManager: Deleting directory C:\Users\18801\AppData\Local\Temp\spark-8bd95ad0-30ae-45d
7-9257-94304c799329
20/06/19 23:45:44 INFO ShutdownHookManager: Deleting directory C:\Users\18801\AppData\Local\Temp\spark-c471b8d1-6da1-482
f-b2bd-d2f5be278e48
```