Spark部署文档

中国科学院软件研究所

部署Hadoop、HBase集群

假设集群中有1个master和9个slaver，角色分配如下

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 主机名 | IP地址 | HDFS角色 | YARN角色 | HBase角色 | Zookeeper角色 |
| master | 133.133.10.1 | NameNode | ResourceManager  MR JobHistoryServer  Spark HistoryServer | HMaster | HQuorumPeer |
| slave1 | 133.133.10.11 | DataNode | NodeManager | HRegionServer | HQuorumPeer |
| slave2 | 133.133.10.12 | DataNode | NodeManager | HRegionServer | HQuorumPeer |
| slave3 | 133.133.10.13 | DataNode | NodeManager | HRegionServer |  |
| slave4 | 133.133.10.14 | DataNode | NodeManager | HRegionServer |  |
| slave5 | 133.133.10.15 | DataNode | NodeManager | HRegionServer |  |
| slave6 | 133.133.10.16 | DataNode | NodeManager | HRegionServer |  |
| slave7 | 133.133.10.17 | DataNode | NodeManager | HRegionServer |  |
| slave8 | 133.133.10.18 | DataNode | NodeManager | HRegionServer |  |
| slave9 | 133.133.10.19 | SecondaryNameNode |  |  |  |

Hadoop部署参考文档：<http://blog.csdn.net/zhu_xun/article/details/42077311>

注意：部署HBase需要对集群进行时间同步<http://jerrylead.iteye.com/blog/1179492>

安装Scala

1. 每个节点都需要安装scala-2.10.5
2. 只需要解压scala-2.10.5.tgz，并设置环境变量

|  |
| --- |
| JAVA\_HOME=~/apps/jdk1.7.0\_79  CLASSPATH=.:$JAVA\_HOME/lib/tools.jar  PATH=$JAVA\_HOME/bin:$PATH  export JAVA\_HOME CLASSPATH PATH  export HADOOP\_HOME=~/hadoop-2.7.1  export PATH=$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin:$PATH  export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop  export SCALA\_HOME=~/apps/scala-2.10.5  export PATH=$SCALA\_HOME/bin:$PATH  export HIVE\_HOME=~/apache-hive-1.2.1-bin |

部署Spark

1. 解压spark-1.6.2-bin-hadoop2.7.1-dynamic.tgz后放到master根目录下
2. 修改spark-1.6.2-bin-hadoop2.7.1-dynamic/conf/spark-defaults.conf

|  |
| --- |
| spark.master spark://master:7077  spark.eventLog.enabled true  spark.eventLog.dir hdfs://master:9000/sparklogs  spark.serializer org.apache.spark.serializer.KryoSerializer  spark.driver.memory 2g  spark.yarn.historyServer.address master:18080  spark.executor.extraClassPath ~/spark-1.6.2-bin-hadoop2.7.1-dynamic/lib/hbase/\*  spark.driver.extraClassPath ~/hbase-1.1.2/conf  # spark.executor.extraJavaOptions -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"  spark.yarn.jar hdfs://master:9000/spark\_lib/spark-assembly-1.6.3-SNAPSHOT-hadoop2.7.1.jar  spark.dynamicAllocation.enabled true  spark.dynamicAllocation.executorIdleTimeout 60s  spark.dynamicAllocation.initialExecutors 1  spark.dynamicAllocation.minExecutors 0  spark.dynamicAllocation.maxExecutors 集群CPU总数 / spark.executor.cores  spark.dynamicAllocation.schedulerBacklogTimeout 1s  spark.shuffle.service.enabled true  spark.executor.cores 2  spark.executor.memory 2g |

1. 修改spark-1.6.2-bin-hadoop2.7.1-dynamic/conf/spark-env.sh

|  |
| --- |
| HADOOP\_CONF\_DIR=~/hadoop-2.7.1/etc/hadoop  SPARK\_YARN\_QUEUE=default  SPARK\_HISTORY\_OPTS="-Dspark.history.fs.logDirectory=hdfs://master:9000/sparklogs"  JAVA\_HOME=~/apps/jdk1.7.0\_79  SCALA\_HOME=~/apps/scala-2.10.5 |

1. 需要修改hadoop-2.7.1/etc/hadoop/yarn-site.xml，添加

|  |
| --- |
| <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle,spark\_shuffle</value>  </property>  <property>  <name>yarn.nodemanager.aux-services.spark\_shuffle.class</name>  <value>org.apache.spark.network.yarn.YarnShuffleService</value>  </property> |

一个完整的yarn-site.xml如下

|  |
| --- |
| <configuration>  <property>  <name>yarn.resourcemanager.hostname</name>  <value>master</value>  <description>ResourceManager host</description>  </property>  <property>  <name>yarn.log-aggregation-enable</name>  <value>true</value>  </property>  <property>  <name>yarn.log.server.url</name>  <value>http://master:19888/jobhistory/logs</value>  </property>  <property>  <name>yarn.resourcemanager.scheduler.class</name>  <value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler</value>  </property>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle,spark\_shuffle</value>  </property>  <property>  <name>yarn.nodemanager.aux-services.spark\_shuffle.class</name>  <value>org.apache.spark.network.yarn.YarnShuffleService</value>  </property>  </configuration> |

1. 保证安装的apache-hive-1.2.1-bin可运行hive命令，将apache-hive-1.2.1-bin/conf/hive-site.xml 拷贝到spark-1.6.2-bin-hadoop2.7.1-dynamic/conf。如果apache-hive-1.2.1-bin/conf/里面没有hive-site.xml，将hive-default.xml.template拷贝为hive-site.xml。
2. 将spark-1.6.2-bin-hadoop2.7.1-dynamic/lib/spark-1.6.2-yarn-shuffle.jar拷贝到所有节点的hadoop-2.7.1/share/hadoop/yarn/lib/
3. 复制hbase-1.1.2/lib下的某些jar包到Spark/lib/hbase

|  |
| --- |
| 将hbase-1.1.2/lib里面的以下jars拷贝到 spark-1.6.2-bin-hadoop2.7.1-dynamic/lib/hbase目录  (1) guava-12.0.1.jar  (2) htrace-core-3.1.0-incubating.jar  (3) protobuf-java-2.5.0.jar  (4) 以hbase开头所有jar |

1. 在HDFS上新建一个文件夹hdfs://master:9000/sparklogs用于保存Spark job logs。
2. 在HDFS上新建一个文件夹hdfs://master:9000/spark\_lib用于存放spark jar。

|  |
| --- |
| hdfs dfs -mkdir /sparklogs  hdfs dfs -mkdir /spark\_lib  hdfs dfs -put spark-assembly-1.6.3-SNAPSHOT-hadoop2.7.1.jar /spark\_lib |

测试MapReduce应用

1. 上传测试数据

|  |
| --- |
| lijie@master:~$ hdfs dfs -mkdir /mrtest  lijie@master:~$ hdfs dfs -mkdir /mrtest/input  lijie@master:~$ hdfs dfs -put hadoop-2.7.1/etc/hadoop/\*.xml /mrtest/input |

1. 测试MapReduce程序

|  |
| --- |
| hdfs dfs -rm -r /mrtest/output  hadoop jar ~/hadoop-2.7.1/share/hadoop/mapreduce/hadoop-mapreduce-examples-\*.jar \  grep -Dmapreduce.job.queuename=default \  /mrtest/input \  /mrtest/output \  'dfs[a-z.]+' |

测试Spark Core 应用

1. 在master节点上运行testSparkCore.sh，该脚本会测试13个Spark examples包中的应用（其中第13个应用SparkTC运行时间较长），具体内容如下：

|  |
| --- |
| #/bin/bash  SPARK\_HOME=~/spark-1.6.2-bin-hadoop2.7.1-dynamic  SPARK\_SUBMIT=$SPARK\_HOME/bin/spark-submit  ENV="--master yarn --deploy-mode client --executor-memory 4g --executor-cores 2 --queue default"  JAR="$SPARK\_HOME/lib/spark-examples\*.jar"  START="======================================================================"  MID="-------------------------------------------------------------------------------"  END="========================================================================"  # 0. Submit data onto HDFS  hdfs dfs -rm -r /sparkbench  hdfs dfs -mkdir /sparkbench  hdfs dfs -mkdir /sparkbench/mllib  hdfs dfs -put $SPARK\_HOME/data/mllib/lr\_data.txt /sparkbench/mllib/  hdfs dfs -put $SPARK\_HOME/data/mllib/kmeans\_data.txt /sparkbench/mllib/  hdfs dfs -put $SPARK\_HOME/data/mllib/pagerank\_data.txt /sparkbench/mllib/  # 1. BroadcastTest [slices=2] [numElem=1000000] [broadcastAlgo=HTTP] [blockSize=4096]  APP="1. BroadcastTest [slices=2] [numElem=1000000] [broadcastAlgo=HTTP] [blockSize=4096]"  CLASS=org.apache.spark.examples.BroadcastTest  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 5 1000000 Http 4096"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 5 1000000 Torrent 4096"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 2. DFSReadWriteTest localfile dfsDirPath  APP="2. DFSReadWriteTest localfile dfsDirPath"  CLASS=org.apache.spark.examples.DFSReadWriteTest  hdfs dfs -rm -r /spark\_dfs\_read\_write\_test  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR $SPARK\_HOME/conf/spark-env.sh /spark\_dfs\_read\_write\_test"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 3. GroupByTest [numMappers] [numKVPairs] [KeySize] [numReducers]  APP="3. GroupByTest [numMappers] [numKVPairs] [KeySize] [numReducers]"  CLASS=org.apache.spark.examples.GroupByTest  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 20 10000 1000 1000"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 4. MultipleBroadcastTest  APP="4. MultipleBroadcastTest"  CLASS=org.apache.spark.examples.MultiBroadcastTest  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 5 1000000"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 5. SimpleSkewedGroupByTest [numMappers] [numKVPairs] [valSize] [numReducers] [ratio]  APP="5. SimpleSkewedGroupByTest [numMappers] [numKVPairs] [valSize] [numReducers] [ratio]"  CLASS=org.apache.spark.examples.SimpleSkewedGroupByTest  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 20 10000 1000 8 2"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 6. SkewedGroupByTest [numMappers] [numKVPairs] [KeySize] [numReducers]  APP="6. SkewedGroupByTest [numMappers] [numKVPairs] [KeySize] [numReducers]"  CLASS=org.apache.spark.examples.SkewedGroupByTest  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 20 10000 1000 8"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 7. SparkALS [M] [U] [F] [iters] [slices]  APP="7. SparkALS [M] [U] [F] [iters] [slices]"  CLASS=org.apache.spark.examples.SparkALS  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 100 500 10 10 10"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 8. SparkHdfsLR <file> <iters>  APP="8. SparkHdfsLR <file> <iters>"  CLASS=org.apache.spark.examples.SparkHdfsLR  FILE="/sparkbench/mllib/lr\_data.txt"  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR $FILE 10"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 9. SparkKMeans <file> <k> <convergeDist>  APP="9. SparkKMeans <file> <k> <convergeDist>"  CLASS=org.apache.spark.examples.SparkKMeans  FILE="/sparkbench/mllib/kmeans\_data.txt"  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR $FILE 2 0.01"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 10. SparkLR [slices]  APP="10. SparkLR [slices]"  CLASS=org.apache.spark.examples.SparkLR  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 4"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 11. SparkPageRank <file> <iter>  APP="11. SparkPageRank <file> <iter>"  CLASS=org.apache.spark.examples.SparkPageRank  FILE="/sparkbench/mllib/pagerank\_data.txt"  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR $FILE 10"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 12. SparkPi  APP="12. SparkPi"  CLASS=org.apache.spark.examples.SparkPi  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 5"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 13. SparkTC  APP="13. SparkTC"  CLASS=org.apache.spark.examples.SparkTC  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR 8"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 14. SparkTachyonHdfsLR  #APP="14. SparkTachyonHdfsLR"  #CLASS=org.apache.spark.examples.SparkTachyonHdfsLR  #FILE="/sparkbench/mllib/lr\_data.txt"  #CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR $FILE 10"  #echo -e "$START \n $APP \n $CMD \n $MID"  #$CMD  #read -p "Press any key to continue." var  # 15. SparkTachyonPi  #APP="15. SparkTachyonPi"  #CLASS=org.apache.spark.examples.SparkTachyonPi  #CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR"  #echo -e "$START \n $APP \n $CMD \n $MID"  #$CMD  #read -p "Press any key to continue." var  # Delete the test data  hdfs dfs -rm -r /sparkbench  hdfs dfs -rm -r /spark\_dfs\_read\_write\_test |

测试Spark SQL应用

1. 在master节点上运行testSparkSQL.sh，该脚本会测试两个Spark examples包中的应用。第一个应用测试SQL基本语句及操作，第二个测试Spark读取Hive上的表。

|  |
| --- |
| #/bin/bash  SPARK\_HOME=~/spark-1.6.2-bin-hadoop2.7.1-dynamic  SPARK\_SUBMIT=$SPARK\_HOME/bin/spark-submit  ENV="--master yarn --deploy-mode client --executor-memory 4g --executor-cores 2 --queue default"  JAR="$SPARK\_HOME/lib/spark-examples\*.jar"  START="======================================================================"  MID="-------------------------------------------------------------------------------"  END="========================================================================"  # 1. RDDRelationTest  APP="1. RDDRelationTest"  CLASS=org.apache.spark.examples.sql.RDDRelation  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var  # 2. HiveFromSpark  APP="2. HiveFromSpark"  CLASS=org.apache.spark.examples.sql.hive.HiveFromSpark  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var |

测试Spark HBase应用

1. 启动HBase，运行hbase-1.1.2/bin/start-hbase.sh
2. 建立一个表scores，有两个列族grad和course

|  |
| --- |
| hbase-1.1.2/bin/hbase shell  *hbase(main):001:0> create 'scores', 'grade', 'course'* |

1. 按设计的表结构插入值

|  |
| --- |
| *put 'scores','Tom','grade:','5' put 'scores','Tom','course:math','97' put 'scores','Tom','course:art','87' put 'scores','Jim','grade','4' put 'scores','Jim','course:','89' put 'scores','Jim','course:','80'* |

可以使用list命令查看当前HBase里有哪些表。使用describe命令来查看表结构。（所有的表名、列名都需要加上引号）

1. 运行testHBase.sh测试HBase的表格读写，具体脚本为

|  |
| --- |
| #/bin/bash  SPARK\_HOME=~/spark-1.6.2-bin-hadoop2.7.1-dynamic  SPARK\_SUBMIT=$SPARK\_HOME/bin/spark-submit  ENV="--master yarn --deploy-mode client --executor-memory 4g --executor-cores 2 --queue default --driver-class-path /home/lijie/hbase-1.1.2/conf"  JAR="$SPARK\_HOME/lib/spark-examples\*.jar"  START="====================================================="  MID="-------------------------------------------------------------------------------"  END="======================================================="  # 1. HBaseTest  APP="1. HBaseTest"  CLASS=org.apache.spark.examples.HBaseTest  CMD="$SPARK\_SUBMIT --class $CLASS $ENV $JAR scores"  echo -e "$START \n $APP \n $CMD \n $MID"  $CMD  read -p "Press any key to continue." var |