# 智能计算技术调研总结

## 1．整体介绍

智能计算依托于大数据，主要涉及数据存储、计算和分布式。目前流行的技术和框架主要有：Hadoop、Spark、HBase等。

此次调研得到的主要底层架构如下：



各部分说明如下：

1. Spark

一个基于内存的计算框架，包含的子框架有：

MLLib,机器学习算法库,实现了常用的分类、回归、聚类、协同过滤等算法；

Streaming，基于数据流的处理,可用来做实时计算；

Spark SQL,能基于类SQL对数据进行查询，统计等。

2）Hadoop Yarn

一个资源调度系统，接受Spark提交的任务,对任务所需的资源：CPU、内存进行统一调度。

3）Hadoop HDFS

Hadoop的分布式存储系统,将数据以文件的形式存储在磁盘上，并能实现冗余备份。

4）Zookeeper

用于保障Hadoop Yarn和Hadoop HDFS集群的HA，避免单点故障而造成集群不可用。

## 2．底层集群部署的规划



Yarn集群和HDFS集群都是基于Master-Slaves的部署架构，为了避免单点故障，Yarn集群和HDFS集群的主节点部署两台，一台处于待命，当active节点故障时，standby节点能够接替上，使集群正常运转;这个过程通过Zookeeper集群的选举算法进行。

Spark可以只部署一台机器，用于向Yarn提交任务。

## 3．应用层设计

应用层分成两部分：离线计算和在线实时处理



1）前期，只进行数据收集工作,当数据收集得足够多时(标准可以是每个设备上传1W条数据,或者是收集三个月),对收集到的数据进行离线分析，进行建模；可以使用80%的数据进行建模，20%的数据进行模型验证。

当模型建立后，可以将模型应用到线上，进行数据的实时处理:

 2）数据的处理结果会进行决策模块(例如决策设备需要重启，或者修改某个参数)，决策的结果会在web层进行展示，并且可以由web层手动触发或者系统设定自动触发。

3）离线模块会定期执行，以便更新修正模型。

4）为了保证系统的稳健运行以及离线模块与在线模块不相互应用，分别需要有一个单独的计算集群。

## 4．工作计划

1）目前公司主要进行的是数据收集工作，但是收集到的数据格式并不满足分析的要求，所以第一步是根据现有的数据，重新规划数据结构和存储方式；

2）当明确了数据结构和存储方式之后，可以搭建集群，进行离线模块的设计,进行模型的训练(由于Spark是采用Scala进行编写的，离线模块建议采用Scala语言，此时需要对该语言进行技术的预研)；

3）模型训练并且可用之后，可以开始搭建线上模型，进行决策模块编写和系统集成阶段

4）最后是进行系统的调整和修改

## 5．所需资源

由于大数据计算基本是基于集群进行工作的，需要的资源如下

1. Yarn集群：2个主节点+3个从节点
2. Hadoop集群：2个主节点+3个从节点
3. Zookeeper集群:3个节点
4. Spark：1个节点
5. Kafka集群：2个主节点+3个从节点
6. Web服务
7. 数据收集服务
8. 决策服务
9. 缓存服务

以上可划分为基础服务1-5和应用服务6-9，基础服务可以是3-5个节点，应用服务是2个节点，基础服务需要分离线和在线两套。

## 附录

1. Zookeeper集群



1. Spark Standalone模式搭建





1. Spark On Yarn模式搭建

**Hadoop集群部署**

1.机器准备

Master 192.168.1.85

Slave1 192.168.1.116

Slave2 192.168.1.154

2.免密登录

Master Slave2

Master Slave1

Slave2 Slave1

ssh-keygen/ssh-copy-id -i /home/spark/.ssh/id\_rsa.pub spark@192.168.1.116

3.安装路径

/data/hadoop/

4.用户及组

spark/spark

5.下载安装包

wget http://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/hadoop-2.7.4/hadoop-2.7.4.tar.gz

tar -zxvf hadoop-2.7.4.tar.gz

6.设置Hadoop环境变量

vim etc/hadoop/hadoop-env.sh

export JAVA\_HOME=/usr/local/soft/jdk1.8.0\_101

7.配置core-site.xml

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://192.168.1.85:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/var/data/hadoop</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>65536</value>

</property>

</configuration>

8.配置hdfs-site.xml

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>192.168.1.116:50090</value>

</property>

<property>

<name>dfs.namenode.secondary.https-address</name>

<value>192.168.1.116:50091</value>

</property>

<property>

<name>dfs.namenode.datanode.registration.ip-hostname-check</name>

<value>false</value>

</property>

</configuration>

9.配置slaves

192.168.1.85

192.168.1.116

192.168.1.154

10.配置mapred-site.xml

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

11.配置yarn-site.xml

<?xml version="1.0"?>

<configuration>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>192.168.1.85</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.vmem-check-enabled</name>

<value>false</value>

</property>

<property>

<name>yarn.nodemanager.vmem-pmem-ratio</name>

<value>4</value>

</property>

<property>

<name>yarn.nodemanager.hostname</name>

<value>{各自机器的ip}</value>

</property>

</configuration>

12.配置环境变量

export HADOOP\_HOME=/data/hadoop/hadoop-2.7.4

export PATH=${HADOOP\_HOME}/bin:${HADOOP\_HOME}/sbin:$PATH

export HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}

export HADOOP\_COMMON\_HOME=${HADOOP\_HOME}

export HADOOP\_HDFS\_HOME=${HADOOP\_HOME}

export YARN\_HOME=${HADOOP\_HOME}

13.复制Hadoop安装包到其他机器

scp -r hadoop-2.7.4/ spark@192.168.1.154:/data/hadoop/

scp -r hadoop-2.7.4/ spark@192.168.1.116:/data/hadoop/

14.启动Hadoop集群

(1)namenode格式化

./bin/hdfs namenode -format

(2)启动HDFS

./sbin/start-dfs.sh

(3)启动Yarn

./sbin/start-yarn.sh

(4)验证,访问web

http://192.168.1.85:8088/cluster/nodes

http://192.168.1.85:8042/node

http://192.168.1.85:50070/dfshealth.html#tab-overview

**Spark On Yarn部署**

1.部署路径

/data/hadoop/

2.下载Spark

wget http://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-2.2.0/spark-2.2.0-bin-hadoop2.7.tgz

3.解压、重命名

tar -zxvf spark-2.2.0-bin-hadoop2.7.tgz

mv spark-2.2.0-bin-hadoop2.7 spark-2.2.0-hadoop2.7

4.配置spark-env.sh

export SPARK\_HOME=/data/hadoop/spark-2.2.0-hadoop2.7

export JAVA\_HOME=/usr/local/soft/jdk1.8.0\_101

export HADOOP\_HOME=/data/hadoop/hadoop-2.7.4

export PATH=$PATH:$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export SPARK\_MASTER\_IP=192.168.1.85

SPARK\_LOCAL\_DIRS=/data/hadoop/spark-2.2.0-hadoop2.7

SPARK\_DRIVER\_MEMORY=1G

SPARK\_MASTER\_HOST=192.168.1.85

export SPARK\_LIBARY\_PATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib

5.配置slaves

192.168.1.85

192.168.1.116

192.168.1.154

6.复制Spark部署包到其他机器

scp -r spark-2.2.0-hadoop2.7/ spark@192.168.1.116:/data/hadoop/

scp -r spark-2.2.0-hadoop2.7/ spark@192.168.1.154:/data/hadoop/

7.启动Spark

./sbin/start-all.sh

**备注**

spark on yarn是不需要启动spark集群的，只需要在提交任务的机器配置spark就可以了，因为任务是由hadoop来执行的，spark只负责任务的提交。

**问题：**

1.集群上运行jar程序，状态一直Accepted且不停止不报错

如果运行Spark集群时状态一直为Accepted且不停止不报错，比如像下面这样的情况：

15/06/14 11:33:33 INFO yarn.Client: Application report for application\_1434263747091\_0023 (state: ACCEPTED)

15/06/14 11:33:34 INFO yarn.Client: Application report for application\_1434263747091\_0023 (state: ACCEPTED)

15/06/14 11:33:35 INFO yarn.Client: Application report for application\_1434263747091\_0023 (state: ACCEPTED)

15/06/14 11:33:36 INFO yarn.Client: Application report for application\_1434263747091\_0023 (state: ACCEPTED)

15/06/14 11:33:37 INFO yarn.Client: Application report for application\_1434263747091\_0023 (state: ACCEPTED)

15/06/14 11:33:38 INFO yarn.Client: Application report for application\_1434263747091\_0023 (state: ACCEPTED)

15/06/14 11:33:39 INFO yarn.Client: Application report for application\_1434263747091\_0023 (state: ACCEPTED)

15/06/14 11:33:40 INFO yarn.Client: Application report for application\_1434263747091\_0023 (state: ACCEPTED)

15/06/14 11:33:41 INFO yarn.Client: Application report for application\_1434263747091\_0023 (state: ACCEPTED)

一般是由于有多个用户同时向集群提交任务或一个用户向集群同时提交了多个任务导致Yarn资源的分配错误。解决这个问题，只需要更改Hadoop的配置文件：/etc/hadoop/conf/capacity-scheduler.xml，把选项：yarn.scheduler.capacity.maximum-am-resource-percent从0.1改成0.5。顾名思义，这个选项是增加Yarn可调度的资源量，当然也可以视具体情况增加更多。也可见，默认情况下，Yarn没有将很多资源分配给任务的能力。

2.资源的机器地址显示的是http://hostname:port，需要配置host才能点击访问

配置yarn的时候，默认每个节点的访问地址是:hostname:port，修改yarn-site.xml文件的yarn.nodemanager.hostname选项，修改为每个节点的ip地址

3.Datanode denied communication with namenode because hostname cannot be resolved (ip=192.168.1.116, hostname=192.168.1.116)

DataNode配置的是ip,默认会将其作为hostName进行解析,可以将域名的解析配置到NameNode的hosts中,可以再NameNode的hdfs-site.xml添加下面配置：

<property>

<name>dfs.namenode.datanode.registration.ip-hostname-check</name>

<value>false</value>

</property>