三个月大数据工程师学习计划

申明：

本文旨在为普通程序员（Java程序员最佳）提供一个入门级别的大数据技术学习路径，不适用于大数据工程师的进阶学习，也不适用于零编程基础的同学。

前言：

正文：

前言

一、背景介绍

2. 数据采集之后，该如何存储？，对应出现了GFS，HDFS，TFS等分布式文件存储系统。

3. 由于数据增长速度快，数据存储就必须可以水平扩展。

4. 数据存储之后，该如何通过运算快速转化成一致的格式，该如何快速运算出自己想要的结果？

   对应的MapReduce这样的分布式运算框架解决了这个问题；但是写MapReduce需要Java代码量很大，所以出现了Hive，Pig等将SQL转化成MapReduce的解析引擎；

   普通的MapReduce处理数据只能一批一批地处理，时间延迟太长，为了实现每输入一条数据就能得到结果，于是出现了Storm/JStorm这样的低时延的流式计算框架；

   但是如果同时需要**批处理**和**流处理**，按照如上就得搭两个集群，Hadoop集群（包括HDFS+MapReduce+Yarn）和Storm集群，不易于管理，所以出现了Spark这样的一站式的计算框架，既可以进行批处理，又可以进行流处理（实质上是微批处理）。

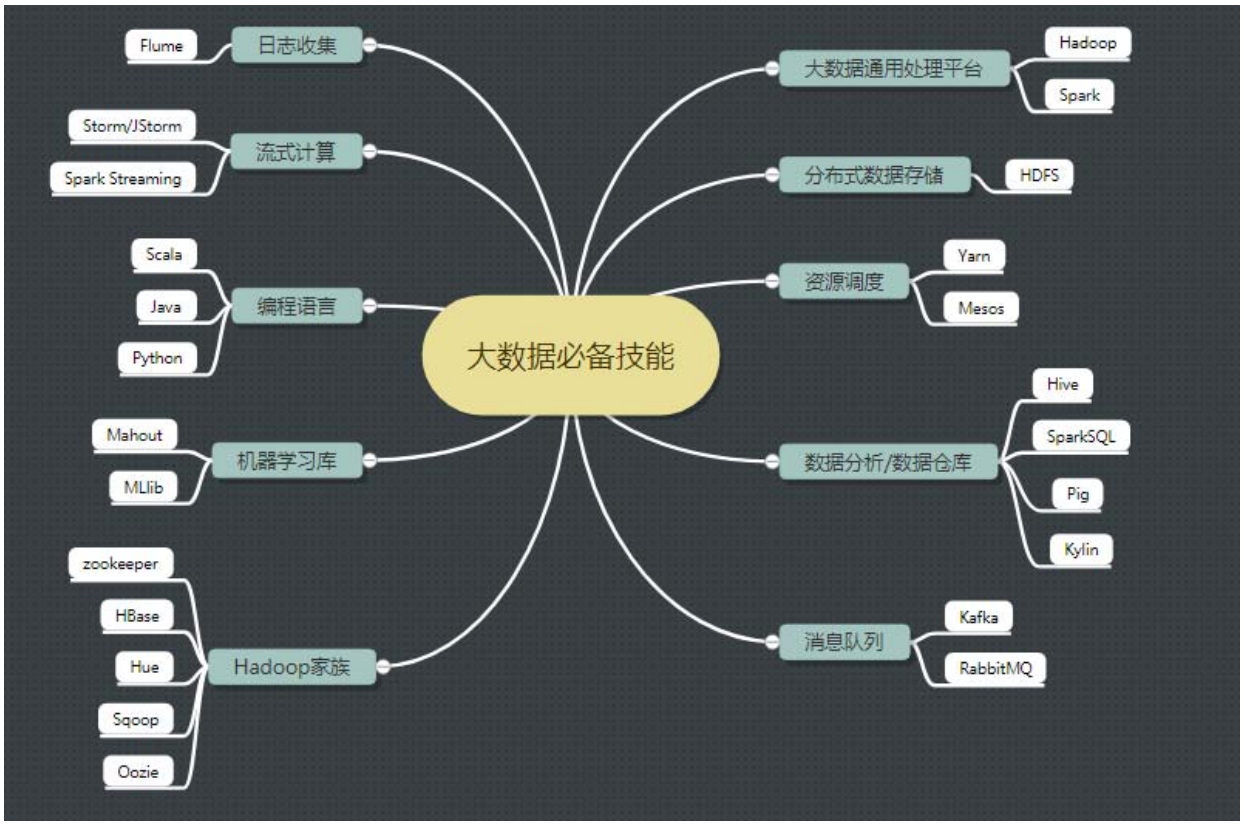
5. 而后Lambda架构，Kappa架构的出现，又提供了一种业务处理的通用架构。

6. 为了提高工作效率，加快运速度，出现了一些辅助工具：

   - Ozzie，azkaban：定时任务调度的工具。
   - Hue，Zepplin：图形化任务执行管理，结果查看工具。
   - Scala语言：编写Spark程序的最佳语言，当然也可以选择用Python。
   - Python语言：编写一些脚本时会用到。
   - Allluxio，Kylin等：通过对存储的数据进行预处理，加快运算速度的工具。

以上大致就把整个大数据生态里面用到的工具所解决的问题列举了一遍，知道了他们为什么而出现或者说出现是为了解决什么问题，进行学习的时候就有的放矢了。

正文



必须掌握的技能11条

1. Java高级(虚拟机、并发)
2. Linux 基本操作
3. Hadoop（ HDFS+MapReduce+Yarn ）
4. HBase（ JavaAPI操作+Phoenix ）
5. Hive(Hql基本操作和原理理解）
6. Kafka
7. Storm/JStorm
8. Scala
9. Python
10. Spark (Core+sparksql+Spark streaming ）
11. 辅助小工具(Sqoop/Flume/Oozie/Hue等)

高阶技能6条

3个月会有（21*3+4*2*10）*3=**423**小时的学习时间。

第一阶段（基础阶段）

1）Linux学习（跟鸟哥学就ok了）——20小时

1. Linux操作系统介绍与安装。
2. Linux常用命令。
3. Linux常用软件安装。
4. Linux网络。
5. 防火墙。
6. Shell编程等。

官网：https://www.centos.org/download/
中文社区：http://www.linuxidc.com/Linux/2017-09/146919.htm

2）Java 高级学习（《深入理解Java虚拟机》、《Java高并发实战》）—30小时

1. 掌握多线程。
2. 掌握并发包下的队列。
3. 了解JMS。
4. 掌握JVM技术。
5. 掌握反射和动态代理。

官网：https://www.java.com/zh_CN/
中文社区：http://www.java-cn.com/index.html

3）Zookeeper学习（可以参照这篇博客进行学习：http://www.cnblogs.com/wuxl360/p/5817471.html）

1. Zookeeper分布式协调服务介绍。
2. Zookeeper集群的安装部署。
3. Zookeeper数据结构、命令。
4. Zookeeper的原理以及选举机制。

2. MapReduce

    - 运行WordCount示例程序。
    - 了解MapReduce内部的运行机制。
        - MapReduce程序运行流程解析。
        - MapTask并发数的决定机制。
        - MapReduce中的combiner组件应用。
        - MapReduce中的序列化框架及应用。
        - MapReduce中的排序。
        - MapReduce中的自定义分区实现。
        - MapReduce的shuffle机制。
        - MapReduce利用数据压缩进行优化。
        - MapReduce程序与YARN之间的关系。
        - MapReduce参数优化。

3. MapReduce的Java应用开发

官网：http://hadoop.apache.org/
中文文档：http://hadoop.apache.org/docs/r1.0.4/cn/
中文社区：http://www.aboutyun.com/forum-143-1.html

5）Hive（《Hive开发指南》）–20小时

1. Hive 基本概念

    - Hive 应用场景。
    - Hive 与hadoop的关系。
    - Hive 与传统数据库对比。
    - Hive 的数据存储机制。

2. Hive 基本操作

    - Hive 中的DDL操作。
    - 在Hive 中如何实现高效的JOIN查询。
    - Hive 的内置函数应用

1. hbase简介。
2. habse安装。
3. hbase数据模型。
4. hbase命令。
5. hbase开发。
6. hbase原理。

官网：http://hbase.apache.org/
中文文档：http://abloz.com/hbase/book.html
中文社区：http://www.aboutyun.com/forum-142-1.html

7）Scala（《快学Scala》）–20小时

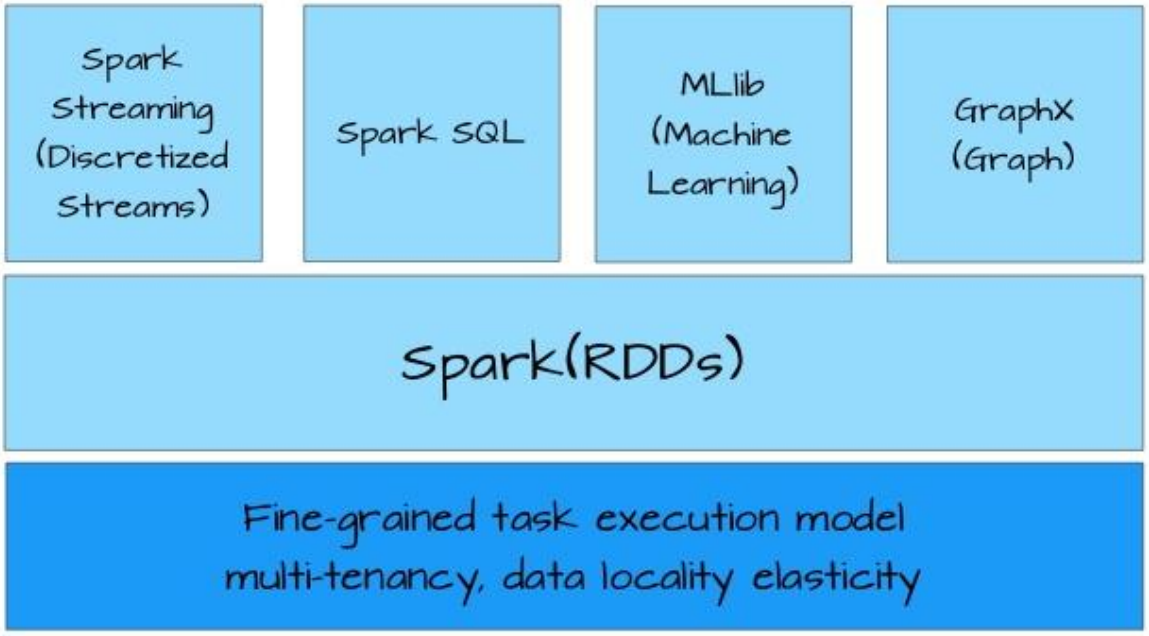
1. Scala概述。
2. Scala编译器安装。
3. Scala基础。
4. 数组、映射、元组、集合。
5. 类、对象、继承、特质。
6. 模式匹配和样例类。
7. 了解Scala Actor并发编程。
8. 理解Akka。
9. 理解Scala高阶函数。
10. 理解Scala隐式转换。

官网：http://www.scala-lang.org/
初级中文教程：http://www.runoob.com/scala/scala-tutorial.html

8）Spark（《Spark 权威指南》）—60小时

- 执行第一个Spark案例程序（求PI）。

2. RDD



- RDD概述。
- 创建RDD。
- RDD编程API（Transformation 和 Action Operations）。
- RDD的依赖关系
- RDD的缓存
- DAG（有向无环图）

3. Spark SQL and DataFrame/DataSet

- park Streaming概述。
- 理解DStream。
- DStream相关操作（Transformations 和 Output Operations）。

5. Structured Streaming

6. 其他（MLlib and GraphX）

这个部分一般工作中如果不是数据挖掘，机器学习一般用不到，可以等到需要用到的时候再深入学习。

官网：http://spark.apache.org
中文文档（但是版本有点老）：https://www.gitbook.com/book/aiyanbo/spark-programming-guide-zh-cn/details
中文社区：http://www.aboutyun.com/forum-146-1.html

9）Python (推荐廖雪峰的博客—30小时

虚机/安装软件	YS01(10.11.145)	YS02 (10.11.146)	YS03(10.11.148)	YS04(10.11.149)
JDK-7	JDK	JDK	JDK	JDK
Hadoop-2.6.4	NameNode(active)	NameNode(standby)	DataNode	DataNode
	ResourceManager	NodeManager	NodeManager	NodeManager
		DataNode		
Zookeeper-3.4.5	QuorumPeerMain	QuorumPeerMain		QuorumPeerMain
Qjournal	DFSZKFailoverController(zkfc)	DFSZKFailoverController(zkfc)		
		JournalNode	JournalNode	JournalNode
MySQL-5.6	MySQL-server/MySQL-client			
Hive-1.2.1	Hive			
Kafka-	Broker	Broker		Broker
Spark-1.6.1	Master (active)	slave	slave	slave
		Master (standby)		
Azkaban-2.5.0	azkaban			
Zeppelin-0.7.1	zepelin			
KafkaOffsetMonit	KafkaOffsetMonitor			

所有需要用到的软件：

链接：http://pan.baidu.com/s/1jIlAz2Y
密码：kyxl

2. 前期准备

2.0 系统安装

HOSTNAME=ys04

2.2 host文件修改

2.2.0 vi /etc/hosts

10.1.1.149 ys01

10.1.1.148 ys02

10.1.1.146 ys03

10.1.1.145 ys04

2.3 关闭防火墙(centos 7默认使用的是firewall，centos 6 默认是iptables)

2.3.0 systemctl stop firewalld.service （停止firewall）

2.3.1 systemctl disable firewalld.service （禁止firewall开机启动）

2.3.2 firewall-cmd --state （查看默认防火墙状态（关闭后显示notrunning，开启后显示running）

2.4 免密登录(ys01 ->ys02,03,04)

ssh-keygen -t rsa

ssh-copy-id ys02(随后输入密码)

ssh-copy-id ys03（随后输入密码）

ssh-copy-id ys04（随后输入密码）

ssh ys02(测试是否成功)

ssh ys03(测试是否成功)

ssh ys04(测试是否成功)

2.5 系统时区与时间同步

tzselect（生成日期文件）

cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime（将日期文件copy到本地时间中）

3. 软件安装

3.0 安装目录规划（软件为所有用户公用）

3.0.0所有软件的安装放到/usr/local/ys/soft目录下(mkdir

3.1.3将java添加到环境变量中

vim /etc/profile

#在文件最后添加

export JAVA_HOME= /usr/local/ys/app/ jdk-7u80

export PATH=$PATH:$JAVA_HOME/bin

3.1.4 刷新配置

source /etc/profile

3.2 Zookeeper安装

3.2.0解压

tar -zxvf zookeeper-3.4.5.tar.gz -C /usr/local/ys/app（解压）

3.2.1 重命名

mv zookeeper-3.4.5 zookeeper（重命名文件夹zookeeper-3.4.5为zookeeper）

3.2.2修改环境变量

vi /etc/profile(修改文件)

添加内容：

export ZOOKEEPER_HOME=/usr/local/ys/app/zookeeper

export PATH=$PATH:$ZOOKEEPER_HOME/bin

3.2.3 重新编译文件：

source /etc/profile

注意：3台zookeeper都需要修改

3.2.4修改配置文件

cd zookeeper/conf

cp zoo_sample.cfg zoo.cfg

vi zoo.cfg

添加内容：

dataDir=/usr/local/ys/app/zookeeper/data

dataLogDir=/usr/local/ys/app/zookeeper/log

server.1=ys01:2888:3888 (主机名, 心跳端口、数据端口)

server.2=ys02:2888:3888

scp -r /usr/local/ys/app/zookeeper ys04:/usr/local/ys/app/

3.2.7修改其他机器的配置文件
到ys02上：修改myid为：2
到ys02上：修改myid为：3

3.2.8启动（每台机器）
zkServer.sh start
查看集群状态
jps（查看进程）
zkServer.sh status（查看集群状态，主从信息）

3.3 Hadoop（HDFS+Yarn）

3.3.0 alt+p 后出现sftp窗口，使用sftp上传tar包到虚机ys01的/usr/local/ys/soft目录下

3.3.1 解压jdk
```
cd /usr/local/ys/soft
  #解压
  tar -zxvf cenos-7-hadoop-2.6.4.tar.gz -C /usr/local/ys/app
```

3.3.2 修改配置文件

core-site.xml

```
<configuration>
        <!-- 指定hdfs的nameservice为ns1 -->
        <property>
          <name>fs.defaultFS</name>
          <value>hdfs://ns1/</value>
        </property>
        <!-- 指定hadoop临时目录 -->
        <property>
          <name>hadoop.tmp.dir</name>
          <value>/usr/local/ys/app/ hadoop-2.6.4/tmp</value>
        </property>
        <!-- 指定zookeeper地址 -->
        <property>
          <name>ha.zookeeper.quorum</name>
          <value>ys01:2181,ys02:2181,ys04:2181</value>
        </property>
</configuration>
```


hdfs-site.xml

```
<configuration>
<!--指定hdfs的nameservice为ns1，需要和core-site.xml中的保持一致 -->
<property>
   <name>dfs.nameservices</name>
   <value>ns1</value>
</property>
<!-- ns1下面有两个NameNode，分别是nn1，nn2 -->
<property>
   <name>dfs.ha.namenodes.ns1</name>
   <value>nn1,nn2</value>
</property>
<!-- nn1的RPC通信地址 -->
<property>
   <name>dfs.namenode.rpc-address.ns1.nn1</name>
   <value>ys01:9000</value>
</property>
```

```
<!-- nn1的http通信地址 -->
<property>
   <name>dfs.namenode.http-address.ns1.nn1</name>
   <value>ys01:50070</value>
</property>
<!-- nn2的RPC通信地址 -->
<property>
   <name>dfs.namenode.rpc-address.ns1.nn2</name>
   <value>ys02:9000</value>
</property>
<!-- nn2的http通信地址 -->
<property>
   <name>dfs.namenode.http-address.ns1.nn2</name>
   <value>ys02:50070</value>
</property>
<!-- 指定NameNode的edits元数据在JournalNode上的存放位置 -->
<property>
   <name>dfs.namenode.shared.edits.dir</name>
   <value>qjournal://ys02:8485;ys03:8485;ys04:8485/ns1</value>
</property>
```

```
<!-- 指定JournalNode在本地磁盘存放数据的位置 -->
<property>
   <name>dfs.journalnode.edits.dir</name>
   <value>/usr/local/ys/app/hadoop-2.6.4/journaldata</value>
</property>
<!-- 开启NameNode失败自动切换 -->
<property>
   <name>dfs.ha.automatic-failover.enabled</name>
   <value>true</value>
</property>
<!-- 配置失败自动切换实现方式 -->
<property>
   <name>dfs.client.failover.proxy.provider.ns1</name>
   <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverP
roxyProvider</value>
</property>
```



```
<!-- 配置隔离机制方法，多个机制用换行分割，即每个机制暂用一行 -->
<property>
   <name>dfs.ha.fencing.methods</name>
   <value>
      sshfence
      shell(/bin/true)
   </value>
</property>
<!-- 使用sshfence隔离机制时需要ssh免登陆 -->
```

```
<property>
  <name>dfs.ha.fencing.ssh.connect-timeout</name>
  <value>30000</value>
</property>
</configuration>
```

mapred-site.xml

```
<configuration>
<!-- 指定mr框架为yarn方式 -->
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

yarn-sifite.xml

```
<configuration>
<!-- 指定YARN的老大（ResourceManager）的地址 -->
    <property>
      <name>yarn.resourcemanager.hostname</name>
      <value>ys01</value>
    </property>
    <!-- reducer获取数据的方式 -->
    <property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
```

```
./zkServer.sh start
#查看状态：一个leader，两个follower
./zkServer.sh status
```

3.3.3.2启动journalnode（分别在在mini5、mini6、mini7上执行）

```
cd /usr/local/ys/app/hadoop-2.6.4
sbin/hadoop-daemon.sh start journalnode
#运行jps命令检验，ys02、ys03、ys04上多了JournalNode进程
```

3.3.3.3格式化HDFS

```
#在ys01上执行命令:
hdfs namenode -format
#格式化后会在根据core-site.xml中的hadoop.tmp.dir配置生成个文件，这里我配置的是/usr/local/ys/app/hadoop-2.6.4/tmp，然后将/usr/local/ys/app/hadoop-2.6.4/tmp拷贝到ys02的/usr/local/ys/app/hadoop-2.6.4/下。
scp -r tmp/ ys02:/usr/local/ys /app/hadoop-2.6.4/
##也可以这样，建议hdfs namenode -bootstrapStandby
```

3.3.3.4格式化ZKFC(在ys01上执行一次即可)

```
hdfs zkfc -formatZK
```

3.3.3.5启动HDFS(在ys01上执行)

```
sbin/start-dfs.sh
```


3.3.3.6启动YARN

```
sbin/start-yarn.sh
```

3.3MySQL-5.6安装

略过

3.4 Hive

3.4.1 alt+p 后出现sftp窗口，cd /usr/local/ys/soft，使用sftp上传tar包到虚机ys01的/usr/local/ys/soft目录下

3.4.2解压

```
cd /usr/local/ys/soft
```

```
<configuration>
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:mysql://localhost:3306/hive?
createDatabaseIfNotExist=true</value>
<description>JDBC connect string for a JDBC metastore</description>
</property>
<property>
<name>javax.jdo.option.ConnectionDriverName</name>
<value>com.mysql.jdbc.Driver</value>
<description>Driver class name for a JDBC metastore</description>
</property>
<property>
```

```
<name>javax.jdo.option.ConnectionUserName</name>
<value>root</value>
<description>username to use against metastore
database</description>
</property>
<property>
<name>javax.jdo.option.ConnectionPassword</name>
<value>Root123456</value>
<description>password to use against metastore database</description>
</property>
</configuration>
```

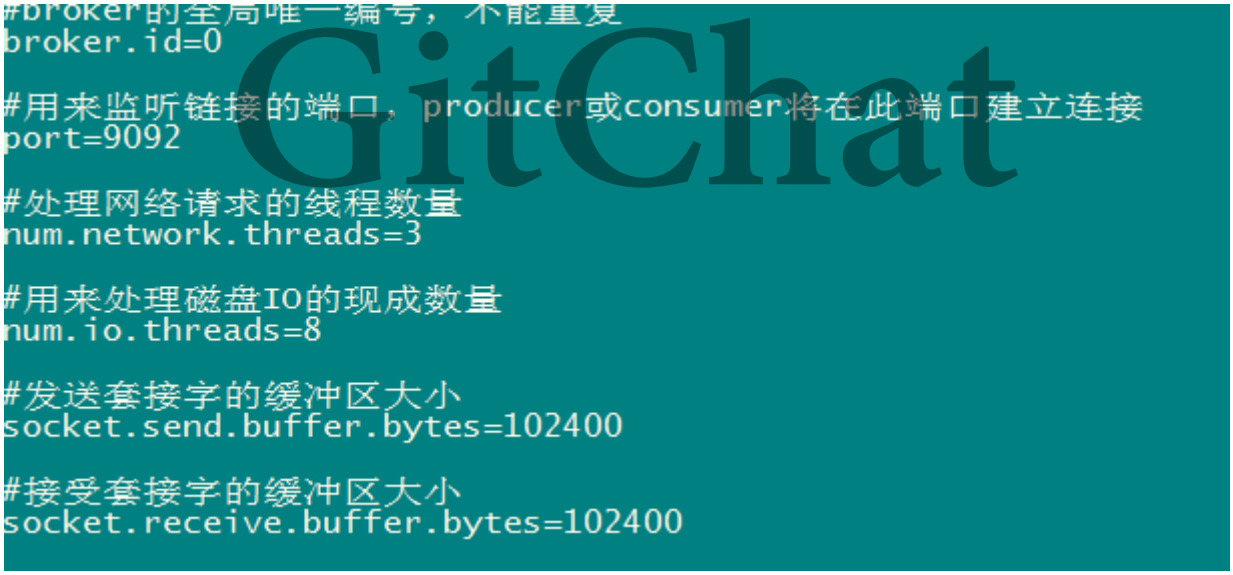

3.4.4 安装hive和mysql完成后，将mysql的连接jar包拷贝到$HIVE_HOME/lib目

```
3.5 Kafka
    3.5.1 下载安装包
    http://kafka.apache.org/downloads.html
    在linux中使用wget命令下载安装包
    wget
http://mirrors.hust.edu.cn/apache/kafka/0.8.2.2/kafka_2.11-
0.8.2.2.tgz

  3.5.2 解压安装包
    tar -zxvf /usr/local/ys/soft/kafka_2.11-0.8.2.2.tgz -C
/usr/local/ys/app/
cd /usr/local/ys/app/
ln -s kafka_2.11-0.8.2.2 kafka

  3.5.3 修改配置文件
    cp
    /usr/local/ys/app/kafka/config/server.properties
   /usr/local/ys/app/kafka/config/server.properties.bak
    vi  /usr/local/ys/kafka/config/server.properties
```

输入以下内容：



3.5.4 分发安装包

scp -r /usr/local/ys/app/kafka_2.11-0.8.2.2 ys02:/usr/local/ys/app/

scp -r /usr/local/ys/app/kafka_2.11-0.8.2.2 ys03:/usr/local/ys/app/

scp -r /usr/local/ys/app/kafka_2.11-0.8.2.2 ys04:/usr/local/ys/app/

然后分别在各机器上创建软连

cd /usr/local/ys/app/

ln -s kafka_2.11-0.8.2.2 kafka

3.5.5 再次修改配置文件（重要）

依次修改各服务器上配置文件的的broker.id，分别是0,1,2不得重复。

3.5.6 启动集群

依次在各节点上启动kafka

bin/kafka-server-start.sh config/server.properties

3.6 Spark

3.6.1 alt+p 后出现sftp窗口，cd /usr/local/ys/soft，使用sftp上传tar包到虚机ys01的/usr/local/ys/soft目录下

3.6.2 解压安装包

tar -zxvf /usr/local/ys/soft/ spark-1.6.1-bin-hadoop2.6.tgz -C /usr/local/ys/app/

3.6.3 修改Spark配置文件(两个配置文件spark-env.sh和slaves)

cd /usr/local/ys/soft/spark-1.6.1-bin-hadoop2.6

进入conf目录并重命名并修改spark-env.sh.template文件

cd conf/

mv spark-env.sh.template spark-env.sh

vi spark-env.sh

在该配置文件中添加如下配置

export JAVA_HOME=/usr/java/jdk1.7.0_45

export SPARK_MASTER_PORT=7077

export SPARK_DAEMON_JAVA_OPTS="-

```
scp -r spark-1.6.1-in-hadoop2.6/ ys02:/usr/local/ys/app
scp -r spark-1.6.1-bin-hadoop2.6/ ys03:/usr/local/ys/app
scp -r spark-1.6.1-bin-hadoop2.6/ ys04:/usr/local/ys/app
```

3.6.5 集群启动

在ys01上执行sbin/start-all.sh脚本

然后在ys02上执行sbin/start-master.sh启动第二个Master

3.7 Azkaban

3.7.1 azkaban web服务器安装

解压azkaban-web-server-2.5.0.tar.gz

命令: tar -zxvf /usr/local/ys/soft/azkaban-web-server-2.5.0.tar.gz -C /usr/local/ys/app/azkaban

将解压后的azkaban-web-server-2.5.0 移动到 azkaban目录中,并重新命名 webserver

命令: mv azkaban-web-server-2.5.0 ../azkaban

```
cd ../azkaban
 mv azkaban-web-server-2.5.0   webserver
```

3.7.2 azkaban 执行服器安装

解压azkaban-executor-server-2.5.0.tar.gz

命令:tar -zxvf /usr/local/ys/soft/azkaban-executor-server-2.5.0.tar.gz -C /usr/local/ys/app/azkaban

将解压后的azkaban-executor-server-2.5.0 移动到 azkaban目录中,并重新命名 executor

```
命令:mv azkaban-executor-server-2.5.0  ../azkaban
cd ../azkaban
mv azkaban-executor-server-2.5.0  executor
```

3.7.3 azkaban脚本导入

解压: azkaban-sql-script-2.5.0.tar.gz

命令:tar -zxvf azkaban-sql-script-2.5.0.tar.gz

将解压后的mysql 脚本,导入到mysql中:

进入mysql

```
mysql> create database azkaban;
mysql> use azkaban;
Database changed
```

您的组织单位名称是什么？
[Unknown]：
您的组织名称是什么？
[Unknown]：
您所在的城市或区域名称是什么？
[Unknown]：
您所在的州或省份名称是什么？
[Unknown]：
该单位的两字母国家代码是什么
[Unknown]： CN
CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=CN
正确吗？
[否]： y
输入<jetty>的主密码（如果和 keystore 密码相同，按回车）：
再次输入新密码
完成上述工作后,将在当前目录生成 keystore 证书文件,将keystore 考贝到
azkaban web服务器根目录中.如:cp keystore azkaban/webserver

3.7.5 配置文件
注：先配置好服务器节点上的时区
先生成时区配置文件Asia/Shanghai，用交互式命令 tzselect 即可
拷贝该时区文件，覆盖系统本地时区配置
cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

3.7.6 azkaban web服务器配置
进入azkaban web服务器安装目录 conf目录
修改azkaban.properties文件
命令vi azkaban.properties

内容说明如下：

```
*Azkaban Personalization Settings
azkaban.name=Test                              #服务器UI名称,用于服务器
上方显示的名字
azkaban.label=My Local Azkaban                                  #描
述
```

```
文件所在位置
azkaban.project.dir=projects
#

database.type=mysql
#数据库类型
mysql.port=3306
#端口号
mysql.host=localhost
#数据库连接IP
mysql.database=azkaban
#数据库实例名
mysql.user=root
#数据库用户名
mysql.password=Root123456
#数据库密码
mysql.numconnections=100
#最大连接数

* Velocity dev mode
velocity.dev.mode=false
* Jetty服务器属性.
jetty.maxThreads=25
#最大线程数
jetty.ssl.port=8443
#Jetty SSL端口
jetty.port=8081
#Jetty端口
jetty.keystore=keystore
#SSL文件名
jetty.password=123456
#SSL文件密码
jetty.keypassword=123456
#Jetty主密码 与 keystore文件相同
jetty.truststore=keystore
#SSL文件名
jetty.trustpassword=123456
# SSL文件密码
```

```
job.failure.email=xxxxxxxx@163.com                                    #
任务失败时发送邮件的地址
job.success.email=xxxxxxxx@163.com                                  #任
务成功时发送邮件的地址
lockdown.create.projects=false
#
cache.directory=cache
#缓存目录
```

3.7.7azkaban 执行服务器executor配置

进入执行服务器安装目录conf,修改azkaban.properties

vi azkaban.properties

```
*Azkaban
default.timezone.id=Asia/Shanghai
#时区

* Azkaban JobTypes 插件配置
azkaban.jobtype.plugin.dir=plugins/jobtypes
#jobtype 插件所在位置

*Loader for projects
executor.global.properties=conf/global.properties
azkaban.project.dir=projects

*数据库设置
database.type=mysql
#数据库类型(目前只支持mysql)
mysql.port=3306
#数据库端口号
mysql.host=192.168.20.200
#数据库IP地址
mysql.database=azkaban
#数据库实例名
mysql.user=root
#数据库用户名
mysql.password=Root23456                                    #数据库密
码
mysql.numconnections=100
```

```
<azkaban-users>
    <user username="azkaban" password="azkaban"
roles="admin" groups="azkaban" />
    <user username="metrics" password="metrics"
roles="metrics"/>
    <user username="admin" password="admin"
roles="admin,metrics" />
    <role name="admin" permissions="ADMIN" />
    <role name="metrics" permissions="METRICS"/>
</azkaban-users>
```

3.7.9 web服务器启动

在azkaban web服务器目录下执行启动命令

bin/azkaban-web-start.sh

注:在web服务器根目录运行

或者启动到后台

```
nohup  bin/azkaban-web-start.sh  1>/tmp/azstd.out
2>/tmp/azerr.out &
```


3.7.10执行服务器启动

在执行服务器目录下执行启动命令

bin/azkaban-executor-start.sh

注:只能要执行服务器根目录运行

启动完成后,在浏览器(建议使用谷歌浏览器)中输入https://服务器IP地址:8443,即可访问azkaban服务了.在登录中输入刚才新的户用名及密码,点击 login

3.8 Zeppelin

参照如下文件:

http://blog.csdn.net/chengxuyuanyonghu/article/details/54915817

```
    export HBASE_CLASSPATH=/usr/local/ys/app/hadoop-
2.6.4/etc/hadoop  //hadoop配置文件的位置
    export HBASE_MANAGES_ZK=false #如果使用独立安装的zookeeper这个地方
就是false（此处使用自己的zookeeper）
```

hbase-site.xml

```
<configuration>
 <property>
 <name>hbase.master</name>    #hbasemaster的主机和端口
 <value>ys01:60000</value>
 </property>
 <property>
 <name>hbase.master.maxclockskew</name>    #时间同步允许的时间差
 <value>180000</value>
 </property>
 <property>
 <name>hbase.rootdir</name>
 <value>hdfs://ns1/hbase</value>#hbase共享目录，持久化hbase数据
 </property>
 <property>
 <name>hbase.cluster.distributed</name> #是否分布式运行，false即为单
机
```

```
<value>true</value>
</property>
<property>
<name>hbase.zookeeper.quorum</name>#zookeeper地址
<value>ys01,ys02,ys03</value>
</property>
<property>
<name>hbase.zookeeper.property.dataDir</name>#zookeeper配置信息快照的位置
<value>/usr/local/ys/app/hbase/tmp/zookeeper</value>
</property>
</property>
<!-- 新增的配置,方便页面查看 -->
<property>
<name>hbase.master.info.port</name>
<value>60010</value>
</property>
</configuration>
```



```
Regionservers     //是从机器的域名
Ys02
ys03
ys04
```

注：此处HBase配置是针对HA模式的hdfs

3.9.4将Hadoop的配置文件hdfs-site.xml和core-site.xml拷贝到HBase配置文件中

```
cp /usr/local/ys/app/Hadoop-2.6.4/etc/hadoop/hdfs-site.xml
```

进程：jps
进入hbase的shell：hbase shell
退出hbase的shell：quit
页面：http://master:60010/

3.10KAfkaOffsetMonitor(Kafka集群的监控程序，本质就是一个jar包)

3.10.1上传jar包
略

3.10.2 运行jar包
```
nohup java -cp KafkaOffsetMonitor-assembly-0.2.1.jar
com.quantifind.kafka.offsetapp.OffsetGetterWeb --zk ys01,ys02,ys04
--refresh 5.minutes --retain 1.day --port 8089 $
```

4. 集群调优

4.1 辅助工具尽量不安装到数据或者运算节点，避免占用过多计算或内存资源。

4.2 dataNode和spark的slave节点尽量在一起；这样运算的时候就可以避免通过网络拉取数据，加快运算速度。

4.3 Hadoop集群机架感知配置，配置之后可以使得数据在同机架的不同机器2份，然后其他机架机器1份，可是两台机器四台虚机没有必要配感知个人感觉。

4.4 配置参数调优
可以参考http://blog.csdn.net/chndata/article/details/46003399

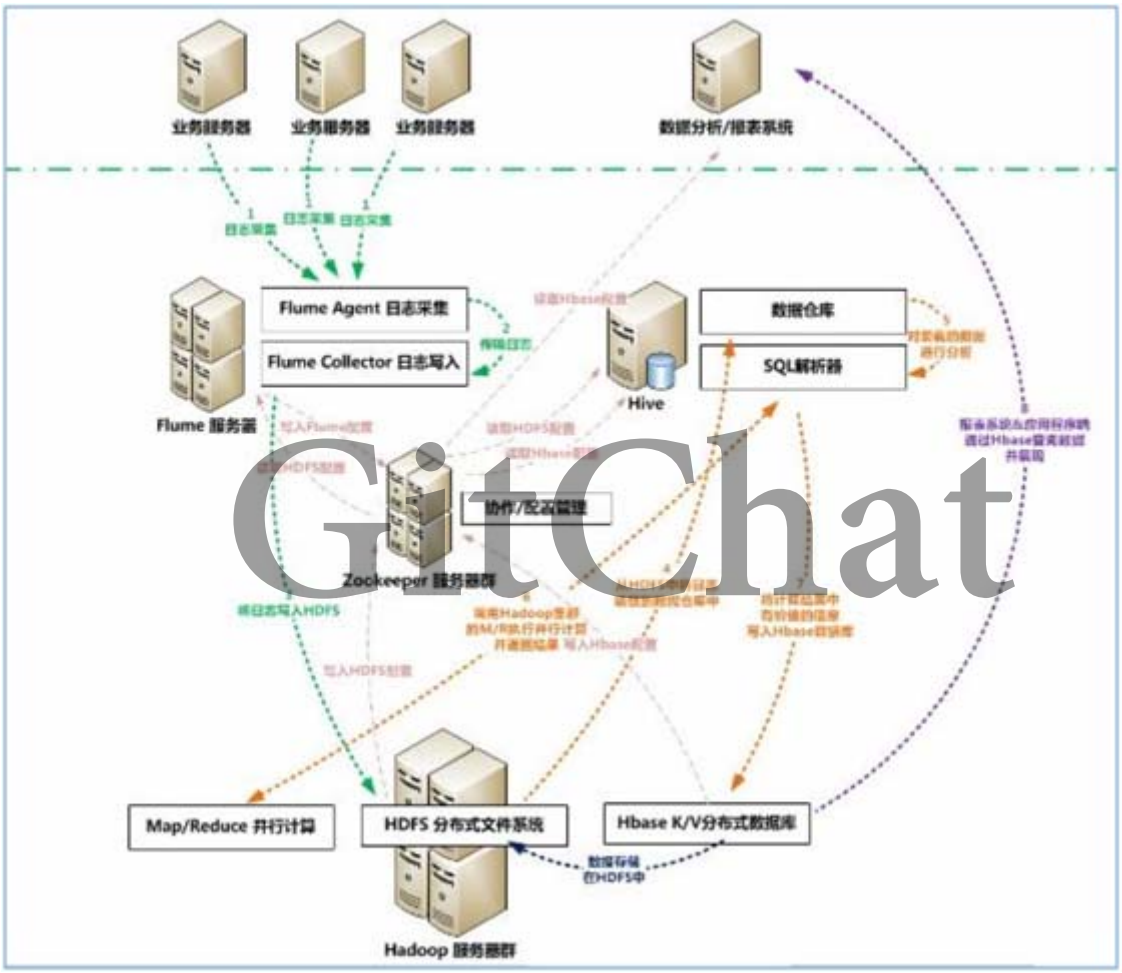
第三阶段（辅助工具工学习阶段）

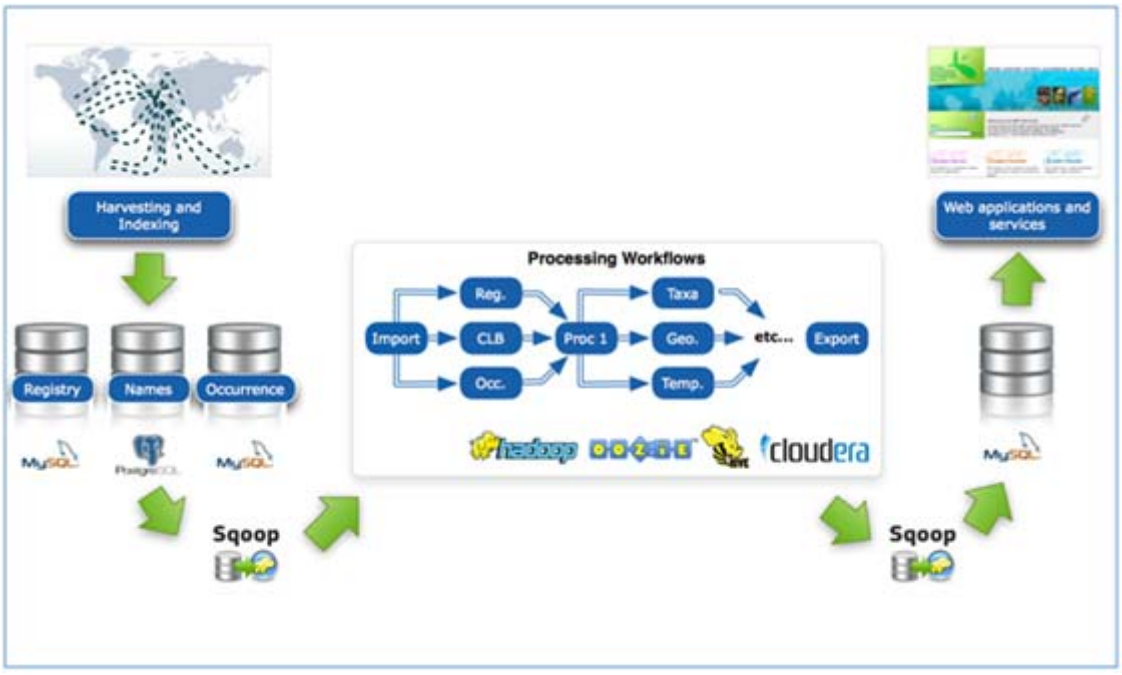
11）Sqoop（CSDN，51CTO ，以及官网）—20小时

1. 数据导出概念介绍
2. Sqoop基础知识
3. Sqoop原理及配置说明
4. Sqoop数据导入实战
5. Sqoop数据导出实战、
6. Sqoop批量作业操作

推荐学习博客：http://student-lp.iteye.com/blog/2157983
官网：http://sqoop.apache.org/

12）Flume（CSDN，51CTO ，以及官网）—20小时





1. 任务调度系统概念介绍。
2. 常用任务调度工具比较。
3. Oozie介绍。
4. Oozie核心概念。
5. Oozie的配置说明。
6. Oozie实现mapreduce/hive等任务调度实战案例。

推荐学习博客：http://www.infoq.com/cn/articles/introductionOozie
官网：http://oozie.apache.org/

14）Hue（CSDN，51CTO ，以及官网）–20小时

推荐学习博客：http://ju.outofmemory.cn/entry/105162
官网：http://gethue.com/

第四阶段（不断学习阶段）

每天都会有新的东西出现，需要关注最新技术动态，不断学习。任何一般技术都是先学习理论，然后在实践中不断完善理论的过程

题）。

5）视频课程推荐：

可以去万能的淘宝购买一些视频课程，你输入“大数据视频课程”，会出现很多，多购买几份（100块以内可以搞定），然后选择一个适合自己的。个人认为小象学院的董西成和陈超的课程含金量会比较高。

四、持续学习资源推荐

1. Apache 官网（http://apache.org/）
2. Stackoverflow（https://stackoverflow.com/）
3. Github(https://github.com/)
4. Cloudra官网(https://www.cloudera.com/)
5. Databrick官网(https://databricks.com/)
6. About 云 ：http://www.aboutyun.com/
7. CSDN，51CTO（http://www.csdn.net/，http://www.51cto.com/）
8. 至于书籍当当一搜会有很多，其实内容都差不多。

五、项目案例分析

1）点击流日志项目分析（此处借鉴CSDN博主的文章，由于没有授权，所以就没有贴过来，下面附上链接）—批处理

http://blog.csdn.net/u014033218/article/details/76847263

2）Spark Streaming在京东的项目实战（京东的实战案例值得好好研究一下，由于没有授权，所以就没有贴过来，下面附上链接）—实时处理
http://download.csdn.net/download/csdndataid_123/8079233

最后但却很重要一点:每天都会有新的技术出现，要多关注技术动向，持续学习。

以上内容不保证一年以后仍适用。