课程介绍

第一部分：大数据生态和hadoop的入门

hadoop

hdfs，mapreduce，hbase。

第二部分：大数据分布式存储系统hdfs

hdfs架构， block，副本

hdfs的读流程和写流程。

namenode， metadata，fsimage，edits

datanode：心跳机制，负载均衡

hdfs shell客户端操作

java客户端操作。

第三部分：大数据分布式数据库系统Hbase

1、初识hbase

整体架构， hbase表的数据模型。

安装和shell命令行的操作演示。

2、hbase原理深入

hbase的数据存储的原理。

LSM tree

hbase读写流程。

region，拆分和并发。

3、hbase应用和优化

java api操作

rowkey的设计

热点，

hbase的二级索引

布隆过滤器。

第1部分 大数据概论

1.1 大数据概念

大数据（big data），IT行业术语，是指无法在一定时间范围内用常规软件工具进行捕捉、管理和处理的数据集合，是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产。

大数据的5V特点（IBM提出）：Volume（大量）、Velocity（高速）、Variety（多样）、Value（低价值密度）、Veracity（真实性）。

1.2 大数据特点

一、Volume：

采集，计算，存储量

二、Variety：

多种类，多样化。结构化，半，非结构化

三、Value：

价值密度相对较低。

四、Velocity：

增长，处理

五、Veracity：

数据的质量。

1.3 大数据应用场景

海量数据 同计算能力相结合的结果。

用户行为记录。

电商行业：

金融场景：

交通：

农牧业：

1.4 大数据部门业务流程分析

大数据部门业务流程

产品人员提需求------>数据部门收集需求，搭建数据平台，分析数据指标 ----->数据可视化

1.5 大数据发展前景

习近平总书记在报告中指出，“建设现代化经济体系，必须把发展经济的着力点放在实体经济上，把提高供给体系质量作为主攻方向，显著增强我国经济质量优势。加快建设制造强国，加快发展先进制造业，推动互联网、大数据、人工智能和实体经济深度融合，在中高端消费、创新引领、绿色低碳、共享经济、现代供应链、人力资本服务等领域培育新增长点、形成新动能。

据IDC预测，到2020年，企业基于大数据计算分析平台的支出将突破5000亿美元。目前，我国大数据人才只有46万，未来3到5年人才缺口达150万之多。数据分析师、数据挖掘工程师已成为一些大型互联网企业和知名IT公司必备的高薪岗位之一。因此大数据人才的储备培养已然变成了技术型企业竞争的新战场。

大数据人才对企业未来的发展有很大的战略价值，而目前市场上的凤毛麟角的大数据人才大多数集中在阿里、百度、华为、中国联通等这样的名企中。

学习大数据知识刻不容缓 时代在发展、行业在变迁，我们必须快人一步，少走弯路，让企业在未来的竞争中脱颖而出，让人才在职业发展中不被淘汰。。

领英报告表明，数据分析人才的供给指数最低，仅为0.05，属于高度稀缺。数据分析人才跳槽速度也最快，平均跳槽速度为19.8个月。

根据中国商业联合会数据分析专业委员会统计，未来中国基础性数据分析人才缺口将达到1400万，而在BAT企业招聘的职位里，60%以上都在招大数据人才。

大数据主要的三大就业方向：大数据系统研发类人才、大数据应用开发类人才和大数据分析/挖掘AI类人才。

1.6 大数据部门组织结构

平台组： 构建基础的数据处理设施。系统搭建维护，运维，还包括数据处理系统流程的开发。

数据收集，流转，处理，存储，展现，应用。

数仓组： 偏向于应用。数据是一个重要的资产。sql，spark ， mapreduce的开发

数据挖掘组/算法组：数学要求高一些。AI独立成一个学科。

前端，服务组：

第2部分 大数据生态

2.1 Hadoop概念

1）Hadoop 是一个能够对`大量数据进行分布式处`理的`基础框架`。具有可靠、高效、可伸缩的特点。作为大数据，首先你要能存的下大数据，其次存的下数据之后，你就开始考虑怎么处理数据。hadoop就是用来解决这两个问题的。

hdfs, mapreduce.

2）广义上来说，HADOOP通常是指一个更广泛的概念——HADOOP生态圈

3）适合应用的场景

Hadoop在设计的最初被设计为针对超大文件及流式数据访问

日志处理，数据分析，etl，广告推荐，离线计算。

4）不适合应用的场景

oltp

低延迟的数据访问，多用户写入和读写

大量小文件处理

2.2 Hadoop发展历程

1）Lucene--Doug Cutting开创的开源软件，用Java书写代码，实现与Google类似的全文搜索功能，它提供了全文检索引擎的架构，包括完整的查询引擎和索引引擎

2）2001年年底成为apache基金会的一个子项目

3）对于大数量的场景，Lucene面对与Google同样的困难

4）学习和模仿Google解决这些问题的办法： 微型版Nutch

5）可以说Google是hadoop的思想之源(Google在大数据方面的三篇论文)谷歌的三驾马车

GFS --->HDFS

Map-Reduce --->MR

BigTable --->Hbase

6）2003-2004年，Google公开了部分GFS和Mapreduce思想的细节，以此为基础Doug Cutting等人用了2年业余时间实现了DFS和Mapreduce机制

7）2005 年Hadoop 作为 Lucene的子项目 Nutch的一部分正式引入Apache基金会。2006 年 3 月份，Map-Reduce和Nutch Distributed File System (NDFS) 分别被纳入称为 Hadoop 的项目中

8）Hadoop就此诞生并迅速发展，标志这云计算时代来临

2.3 Hadoop版本

Hadoop 三大发行版本: Apache、Cloudera、Hortonworks。

1）Apache Hadoop

官网地址：http://hadoop.apache.org/releases.html

下载地址：https://archive.apache.org/dist/hadoop/common/

2）Cloudera Hadoop

（1）2008年成立的Cloudera是最早将Hadoop商用的公司，为合作伙伴提供Hadoop的商用解决方案，主要是包括支持、咨询服务、培训。

（2）2009年Hadoop的创始人Doug Cutting也加盟Cloudera公司。Cloudera产品主要为CDH，Cloudera Manager，Cloudera Support

（3）CDH是Cloudera的Hadoop发行版，完全开源，比Apache Hadoop在兼容性，安全性，稳定性上有所增强。

（4）Cloudera Manager是集群的软件分发及管理监控平台，可以在几个小时内部署好一个Hadoop集群，并对集群的节点及服务进行实时监控。Cloudera Support即是对Hadoop的技术支持。

3）Hortonworks Hadoop

官网地址：https://hortonworks.com/products/data-center/hdp/

下载地址：https://hortonworks.com/downloads/#data-platform

（1）2011年成立的Hortonworks是雅虎与硅谷风投公司Benchmark Capital合资组建。

（2）公司成立之初就吸纳了大约25名至30名专门研究Hadoop的雅虎工程师，上述工程师均在2005年开始协助雅虎开发Hadoop，贡献了Hadoop80%的代码。

（3）雅虎工程副总裁、雅虎Hadoop开发团队负责人Eric Baldeschwieler出任Hortonworks的首席执行官。

（4）Hortonworks的主打产品是Hortonworks Data Platform（HDP），也同样是100%开源的产品，HDP除常见的项目外还包括了Ambari，一款开源的安装和管理系统。

（5）HCatalog，一个元数据管理系统，HCatalog现已集成到Facebook开源的Hive中。Hortonworks的Stinger开创性的极大的优化了Hive项目。Hortonworks为入门提供了一个非常好的，易于使用的沙盒。

2.4 Hadoop特点

1）高可靠性：因为Hadoop假设计算元素和存储会出现故障，因为它维护多个工作数据副本，在出现故障时可以对失败的节点重新分布处理。

IBM,oracle,大型商业机，多个副本，3个副本。

2）高扩展性：在集群间分配任务数据，可方便的扩展数以千计的节点。

3）高效性：在MapReduce的思想下，Hadoop是并行工作的，以加快任务处理速度。

4）高容错性：自动保存多份副本数据，并且能够自动将失败的任务重新分配。

2.5 Hadoop组成

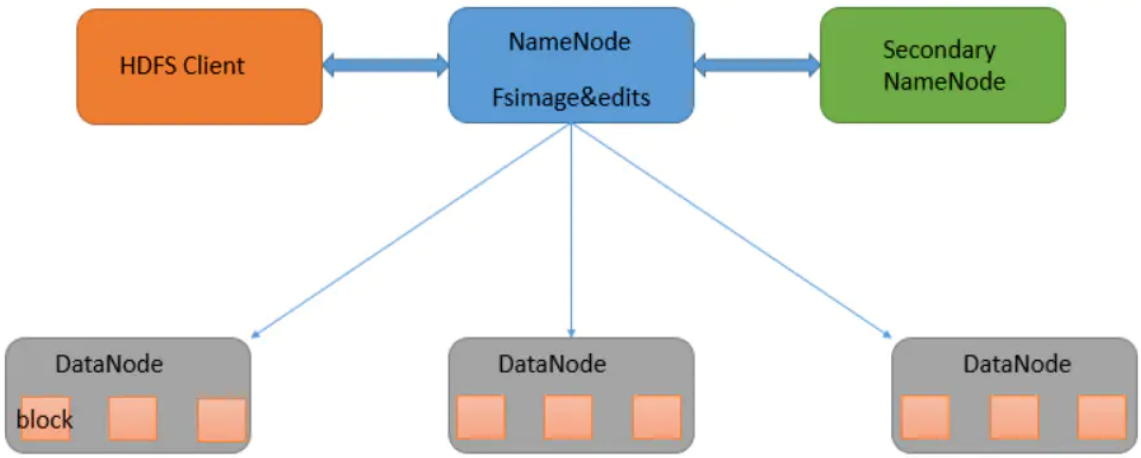
1）Hadoop HDFS：（hadoop distribute file system ）一个高可靠、高吞吐量的分布式文件系统。

2）Hadoop MapReduce：一个分布式的离线并行计算框架。

3）Hadoop YARN：作业调度与集群资源管理的框架。

4）Hadoop Common：支持其他模块的工具模块（Configuration、RPC、序列化机制、日志操作）。

2.5.1 HDFS架构概述



namenode

存储文件的元数据，文件名，目录结构，属性（time，副本数，文件权限）

datanode

在本地文件系统存储文件数据，块数据的校验

second namenode

用来监控hdfs状态的辅助后台程序，每隔一段时间获取hdfs元数据的快照。

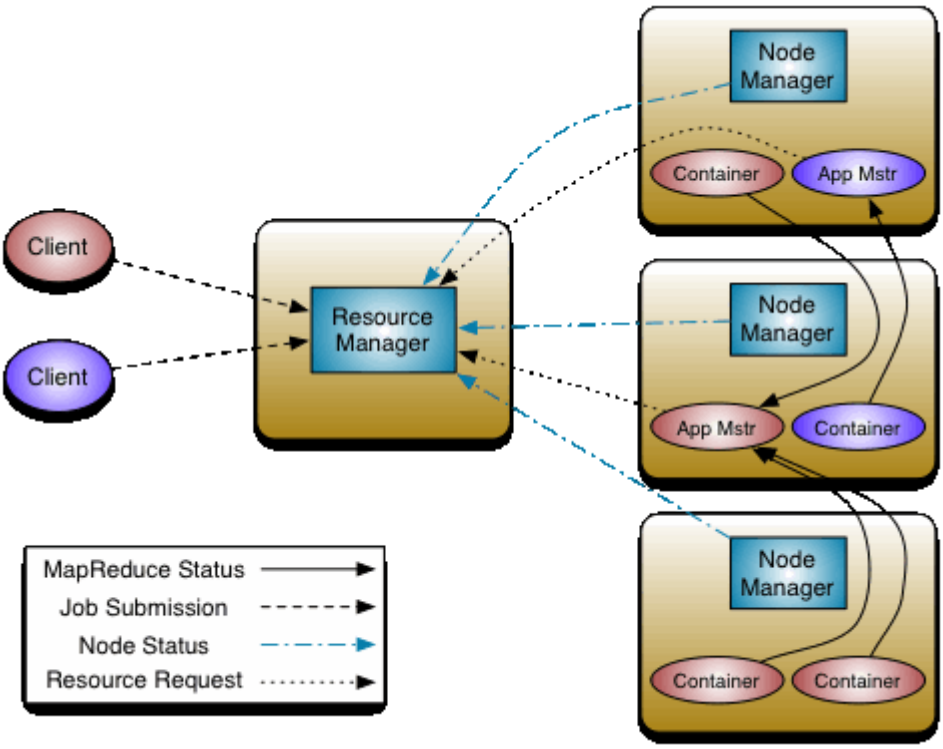
2.5.2 YARN架构概述

YARN是Hadoop2.0版本新引入的资源管理系统，直接从MR1演化而来。

核心思想：将MP1中JobTracker的资源管理和作业调度两个功能分开，分别由ResourceManager（RM）和ApplicationMaster（AM）进程来实现。

1）ResourceManager：负责整个集群的资源管理和调度。

2）ApplicationMaster：负责应用程序相关的事务，比如任务调度、任务监控和容错等。



img

yarn的基本架构

1） ResourceManager(rm)：处理客户端请求、启动/监控ApplicationMaster、监控NodeManager、资源分配与调度；

1、处理客户端请求；

2、启动或监控ApplicationMaster；

3、监控NodeManager；

4、资源的分配与调度。

2）NodeManager(nm)：单个节点上的资源管理、处理来自ResourceManager的命令、处理来自ApplicationMaster的命令；

1、单个节点上的资源管理；

2、处理来自ResourceManager上的命令；

3、处理来自ApplicationMaster上的命令。

3）ApplicationMaster：数据切分、为应用程序申请资源，并分配给内部任务、任务监控与容错。

管理一个在YARN内运行的应用程序的每个实例。ApplicationMaster 负责协调来自 ResourceManager 的资源，并通过 NodeManager 监视容器的执行和资源使用（CPU、内存等的资源分配）

1）负责数据的切分；

2）为应用程序申请资源并分配给内部的任务；

3）任务的监控与容错。

4）Container：对任务运行环境的抽象，封装了CPU、内存等多维资源以及环境变量、启动命令等任务运行相关的信息。

主从结构：master/slave

2.5.3 MapReduce架构概述

Mapreduce是一个分布式运算程序的编程框架，是用户开发“基于hadoop的数据分析应用”的核心框架；
Mapreduce核心功能是将用户编写的业务逻辑代码和自带默认组件整合成一个完整的分布式运算程序，并发运行在一个hadoop集群上。

MapReduce将计算过程分为两个阶段：Map（映射）和Reduce（归约）

1）Map阶段并行处理输入数据

2）Reduce阶段对Map结果进行汇总

分而治之

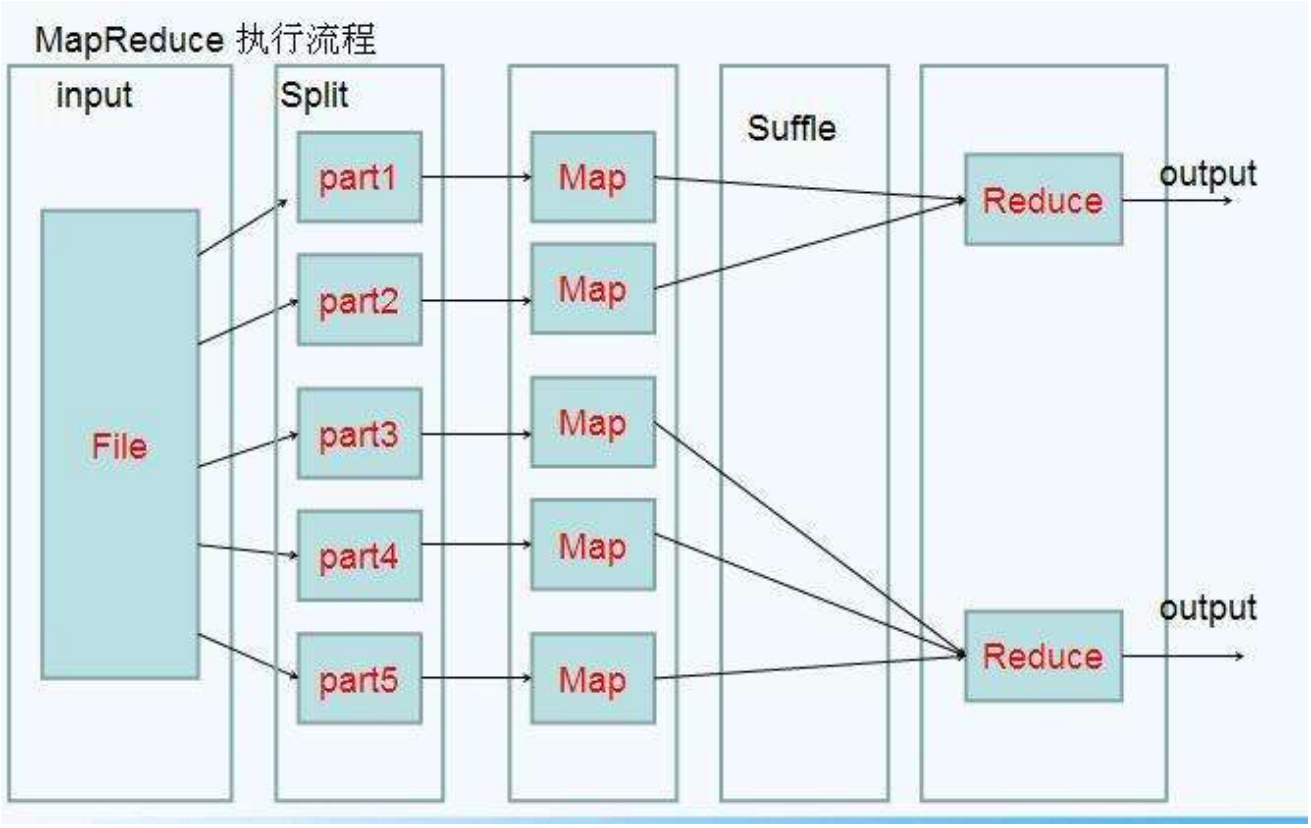
map 负责分：

1、小

2、就近计算原则

3、并行计算

reduce：



Mapper任务的执行过程详解

2.6 大数据生态组件

1）Sqoop：sqoop是一款开源的工具，主要用于在Hadoop(Hive)与传统的数据库(mysql)间进行数据的传递，可以将一个关系型数据库（例如： MySQL ,Oracle 等）中的数据导进到Hadoop的HDFS中，也可以将HDFS的数据导进到关系型数据库中。

2）Flume：Flume是Cloudera提供的一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统，Flume支持在日志系统中定制各类数据发送方，用于收集数据；同时，Flume提供对数据进行简单处理，并写到各种数据接受方（可定制）的能力。

3）Kafka：Kafka是一种高吞吐量的分布式发布订阅消息系统，有如下特性：

（1）通过O(1)的磁盘数据结构提供消息的持久化，这种结构对于即使数以TB的消息存储也能够保持长时间的稳定性能。

（2）高吞吐量：即使是非常普通的硬件Kafka也可以支持每秒数百万的消息

（3）支持通过Kafka服务器和消费机集群来分区消息。

（4）支持Hadoop并行数据加载。

4）Oozie：Oozie是一个管理Hadoop作业（job）的工作流程调度管理系统。Oozie协调作业就是通过时间（频率）和有效数据触发当前的Oozie工作流程。

5）Hbase：HBase是一个分布式的、面向列的开源数据库。HBase不同于一般的关系数据库，它是一个适合于非结构化数据存储的数据库。

6）Hive：hive是基于Hadoop的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供简单的sql查询功能，可以将sql语句转换为MapReduce任务进行运行。 其优点是学习成本低，可以通过类SQL语句快速实现简单的MapReduce统计，不必开发专门的MapReduce应用，十分适合数据仓库的统计分析。

7）Storm：Storm为分布式实时计算提供了一组通用原语，可被用于“流处理”之中，实时处理消息并更新数据库。这是管理队列及工作者集群的另一种方式。 Storm也可被用于“连续计算”（continuous computation），对数据流做连续查询，在计算时就将结果以流的形式输出给用户。

8）Spark/sparkstreaming：Spark是当前最流行的开源大数据内存计算框架。可以基于Hadoop上存储的大数据进行计算。

9）flink 流式计算引擎

10）Mahout，sparkMlib

Apache Mahout是个可扩展的机器学习和数据挖掘库，当前Mahout支持主要的4个用例：

推荐挖掘，聚集，分类。

频繁项集挖掘：将一组项分组，并识别哪些个别项会经常一起出现。

11）ZooKeeper：Zookeeper是Google的Chubby一个开源的实现。它是一个针对大型分布式系统的可靠协调系统，提供的功能包括：配置维护、名字服务、 分布式同步、组服务等。ZooKeeper的目标就是封装好复杂易出错的关键服务，将简单易用的接口和性能高效、功能稳定的系统提供给用户。

第3部分 Hadoop集群搭建

3.1虚拟机环境准备

1、如果没有服务器环境，我们自己windows电脑可以按照vmware虚拟机，在虚拟机上按照linux服务器。

2、克隆三台虚拟机

3、关闭防火墙 service iptables stop；chkconfig iptables off

3.2 安装jdk

```
wget https://download.oracle.com/otn/java/jdk/8u231-b12/478a62b7d4e34b78b671c754eaaf38ab/jdk-8u231-linux-x64.tar.gz


tar -zxvf jdk-8u231-linux-x64.tar.gz
```

我自己的测试服务器的目录：

/opt/module/jdk1.8.0_231

```
export JAVA_HOME=/opt/module/jdk1.8.0_231
export PATH=$PATH:$JAVA_HOME/bin
```

环境变量生效：

source /etc/profile

3.3 安装Hadoop

Hadoop下载地址：https://archive.apache.org/dist/hadoop/common/hadoop-2.7.2/

1. 用文件传输工具工具将hadoop-2.7.2.tar.gz导入到opt目录下面的software文件夹下面
2. 进入到Hadoop安装包路径下

   ```
   cd ~/software/
   ```

3. 解压安装文件到/opt/module下面

```
tar -zxvf hadoop-2.7.2.tar.gz  -C /home/teacher/opt/module/
```

4. 将hadoop添加到环境变量

   打开/etc/profile:

   在profie文件末尾添加jdk路径，

```
export HADOOP_HOME=/home/teacher/opt/module/hadoop-2.7.2
export PATH=$PATH:$HADOOP_HOME/bin
```

source /etc/profile

5. 测试是否安装成功

```
hadoop version
```

3.4 集群配置

1. **集群部署规划**

	teacher1	**teacher2**	**teacher3**
HDFS	NameNode,DataNode	DataNode	datanode，secondarynamenode
YARN	NodeManager	ResourceManager，NodeManager	NodeManager

2. 配置集群

序号	配置文件名	配置对象	主要内容
1	hadoop-env.sh	**hadoop运行环境**	**用来定义hadoop运行环境相关的配置信息**
2	core-site.xml	集群全局参数	用于定义系统级别的参数，如HDFS URL 、Hadoop的临时目录等
3	hdfs-site.xml	HDFS	如名称节点和数据节点的存放位置、文件副本的个数、文件的读取权限等
4	mapred-site.xml	Mapreduce参数	包括JobHistory Server 和应用程序参数两部分，如reduce任务的默认个数、任务所能够使用内存的默认上下限等
5	yarn-site.xml	集群资源管理系统参数	配置ResourceManager ， nodeManager的通信端口，web监控端口等

（1）配置hadoop-env.sh

```
//文件末尾
export JAVA_HOME=/opt/module/jdk1.8.0_231
```

核心配置文件：core-site.xml（hdfs的核心配置文件）

core-site.xml

```
<!-- 指定HDFS中NameNode的地址 -->
        <property>
                <name>fs.defaultFS</name>
        <value>hdfs://teacher1:9000</value>
        </property>

        <!-- 指定hadoop运行时产生文件的存储目录 -->
        <property>
                <name>hadoop.tmp.dir</name>
                <value>/home/teacher/opt/module/hadoop-2.7.2/data/tmp</value>
        </property>
```

hdfs配置文件 hdfs-site.xml

```
<property>
        <name>dfs.replication</name>
        <value>3</value>
 </property>
        <!--secondarynamenode的地址--> 辅助namenode工作
 <property>
   <name>dfs.namenode.secondary.http-address</name>
     <value>teacher3:50090</value>
    </property>

<property>
    <name>dfs.name.dir</name>
    <value>/home/teacher/data/hadoop/name/</value>
</property>

<property>
    <name>dfs.data.dir</name>
    <value>/home/teacher/data/hadoop/data/</value>
</property>
```

（3）yarn配置文件

yarn-env.sh

重点看下yarn-site.xml

内存参数解释：

yarn.nodemanager.resource.memory-mb

表示该节点上YARN可使用的物理内存总量，默认是8192（MB），注意，如果你的节点内存资源不够8GB，则需要调减小这个值，而YARN不会智能的探测节点的物理内存总量。

yarn.scheduler.minimum-allocation-mb

单个任务可申请的最少物理内存量，默认是1024（MB），如果一个任务申请的物理内存量少于该值，则该对应的值改为这个数。

yarn.scheduler.maximum-allocation-mb

单个任务可申请的最多物理内存量，默认是8192（MB）。

CPU相关配置参数如下：

YARN中目前的CPU被划分成虚拟CPU（CPU virtual Core），这里的虚拟CPU是YARN自己引入的概念，初衷是，考虑到不同节点的CPU性能可能不同，每个CPU具有的计算能力也是不一样的，比如某个物理CPU的计算能力可能是另外一个物理CPU的2倍，这时候，你可以通过为第一个物理CPU多配置几个虚拟CPU弥补这种差异。用户提交作业时，可以指定每个任务需要的虚拟CPU个数。

yarn.nodemanager.resource.cpu-vcores：

表示该节点上YARN可使用的虚拟CPU个数，默认是8，注意，目前推荐将该值设值为与物理CPU核数数目相同。如果你的节点CPU核数不够8个，则需要调减小这个值，而YARN不会智能的探测节点的物理CPU总数。
yarn.scheduler.minimum-allocation-vcores：

单个任务可申请的最小虚拟CPU个数，默认是1，如果一个任务申请的CPU个数少于该数，则该对应的值改为这个数。 yarn.scheduler.maximum-allocation-vcores：

单个任务可申请的最多虚拟CPU个数，默认是32。 对于一个CPU核数较多的集群来说，上面的默认配置显然是不合适的.

```
<!-- Site specific YARN configuration properties -->
<!--NodeManager上运行的附属服务。需配置成mapreduce_shuffle，才可运行MapReduce程序>
        <property>
                 <name>yarn.nodemanager.aux-services</name>
                 <value>mapreduce_shuffle</value>
        </property>

        <!-- 指定YARN的ResourceManager的地址 -->
        <property>
                <name>yarn.
                resourcemanager.hostname</name>
                <value>teacher2</value>
        </property>

        <property>
         <name>yarn.nodemanager.resource.memory-mb</name>
          <value>3072</value>
          </property>
```

（4）mapreduce配置文件

mapred-env.sh

```
 cp mapred-site.xml.template mapred-site.xml
```

mapred-site.xml增加

```
<!-- 指定mr运行在yarn上 -->
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
```

3.配置集群中从节点信息

vim slaves

```
teacher1
teacher2
teacher3
```

4.分发文件

将teacher1中hadoop目录下的软件拷贝到其他机器

```
scp  -r hadoop-2.7.2 teacher3:/home/teacher/opt/module/
```

3.5 集群单点启动

（1）如果集群是第一次启动，需要格式化**NameNode（格式化只进行一次！）**

hadoop namenode -format

（2）在teacher1上启动NameNode

sbin/hadoop-daemon.sh start namenode

（3）在teacher1,teacher2,teacher3上分别启动DataNode

sbin/hadoop-daemon.sh start datanode

总结：单点启动需要到每个服务上去启动，非常麻烦

3.6 SSH无密登录配置

teacher1机器执行：

1、生成密钥

```
ssh-keygen
```

1. id_rsa （私钥）
2. id_rsa.pub (公钥)

2、复制公钥到其他节点

```
ssh-copy-id -i .ssh/id_rsa.pub teacher2
```

vim ~/.ssh/authorized_keys可以 查看复制过来的公钥

3、复制完成即可实现免密登录，测试一下：

ssh teacher2

4、其他机器teacher2，teacher3同样操作。

3.7集群启动/停止方式

1. 各个服务组件逐一启动/停止（集群某个进程挂掉使用这种方式重启 ）
   （1）分别启动/停止hdfs组件

```
hadoop-daemon.sh  start|stop  namenode|datanode|secondarynamenode
```

（2）启动/停止yarn.

```
yarn-daemon.sh  start|stop  resourcemanager|nodemanager
```

2、**分模块启动，集群启动方式**

各个模块分开启动/停止（配置ssh是前提）常用

```
（1）整体启动/停止hdfs（在namenode节点启动）

start-dfs.sh

stop-dfs.sh

（2）整体启动/停止yarn （在resourcemanager节点启动）

start-yarn.sh

stop-yarn.sh
```

3.8 集群测试

1. 启动集群

（1）如果集群是第一次启动，需要格式化NameNode

hadoop namenode -format

（2）启动HDFS：

sbin/start-dfs.sh

（3）启动yarn

sbin/start-yarn.sh

Yarn的web页面查看地址：http://teacher2:8088/

（4）web端查看SecondaryNameNode http://teacher3:50090/status.html

2. 集群基本测试

（1）上传文件到集群

上传小文件 file system

```
hadoop fs -mkdir -p /user/root/input
hadoop fs -put wc.input /user/root/input
```

（2）上传文件后查看文件存放在什么位置

查看HDFS文件存储路径

pwd

查看HDFS在磁盘存储文件内容

（3）下载

```
hadoop fs -get  xxx   ./
```

3.9 集群时间同步（配置即可）

时间同步的方式：在集群中找一台机器，作为时间服务器，集群中其他机器与这台机器定时的同步时间，比如，每隔十分钟，同步一次时间。

配置时间同步实操：

1. **时间服务器配置**（必须root用户）

（1）检查ntp是否安装,若没有安装则使用yum install -y ntp进行安装

[root@teacher1]# rpm -qa|grep ntp

（2）修改ntp配置文件

[root@teacher1]# vim /etc/ntp.conf

① 配置方法一：只允许192.168.100.0网段的客户机进行时间同步

在restrict default kod nomodify notrap nopeer noquery（表示默认拒绝所有IP的时间同步）之后增加一行：

restrict 192.168.100.0 mask 255.255.255.0 nomodify notrap

② 配置方法二：允许任何ip的客户机都可以进行时间同步

将restrict default kod nomodify notrap nopeer noquery修改为如下行：

Restrict default nomodify

（3）修改/etc/sysconfig/ntpd 文件

[root@teacher1]# vim /etc/sysconfig/ntpd

增加内容如下（让硬件时间与系统时间一起同步）

SYNC_HWCLOCK=yes

（4）重新启动ntpd

[root@teacher1]# service ntpd status

ntpd 已停

[root@teacher1]# service ntpd start

（5）使NTP服务可以在系统引导的时候自动启动：

[root@teacher1]# chkconfig ntpd on

2. **其他机器配置**（必须root用户）

（1）在其他机器配置10分钟与时间服务器同步一次

[root@teacher1 hadoop-2.7.2]# crontab -e

编写脚本

*/10 * * * * /usr/sbin/ntpdate teacher1

（2）修改任意机器时间

[root@teacher2]# date -s "2020-5-19 11:11:11"

（3）十分钟后查看机器是否与时间服务器同步

[root@teacher2 root]# date