深入 Hadoop HDFS

<1>. hdfs 架构简介

1.1 hdfs 架构挑战

hdfs 和大多数现有的分布式文件系统存在很多类似特点，但是又具有自己一些特性：具有很高的容错性 *highly fault-tolerant*，较高的数据吞吐量 *high throughput* 等。为了满足上面的特性，hdfs 将不得不解决下面的一些棘手问题：

1. 硬件错误 :在一个 hdfs 系统中可能保存大量的服务器 ,那么每个服务器均存在硬件故障的可能性，那么 hdfs 需要保证能够自动检测到某个服务器错误，同时能够自动恢复。这个目标是 hdfs 架构的首要解决的问题。

2. 流式的数据访问 *Streaming Data Access*：hdfs 需要向应用程序提供流式的数据访问。

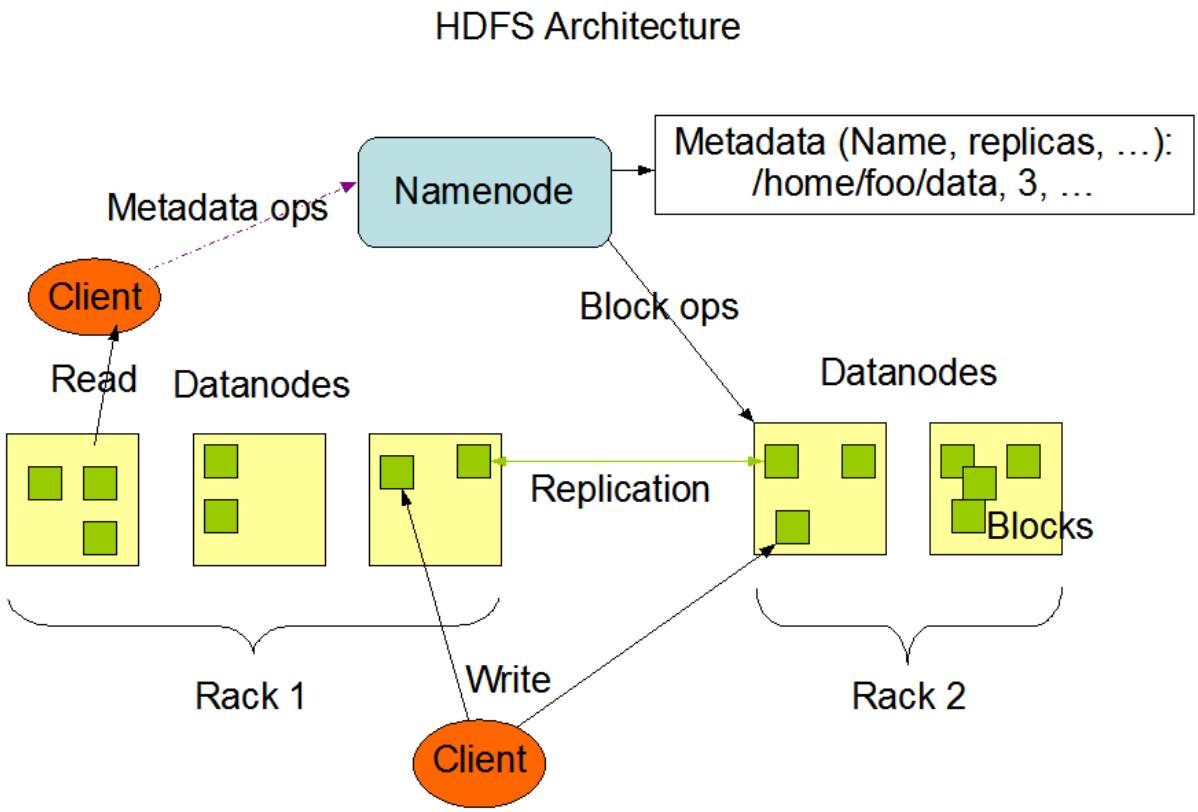
3. 大文件支持：在 hdfs 上存储的文件可能是在 G 级别或者是 T 级别的，这样 hdfs 需要能够对大文件支持。同时需要支持在一个实例中存储大量的文件(*It should support tens of millions of files in a single instance*)。

4. 数据一致性保证：hdfs 需要能够支持“write-once-read-many access” 模型。

面对上面的架构需求，我们来看看 hdfs 是如何满足上面的架构需求的。

1.2 架构简介

hdfs 采用的是 master/slave 模型，一个 hdfs cluster 包含一个 NameNode 和一些列的 DataNode，其中 NameNode 充当的是 master 的角色，主要负责管理 hdfs 文件系统，接受来自客户端的请求；DataNode 主要是用来存储数据文件，hdfs 将一个文件分割成一个多这是多个的 block，这些 block 可能存储在一个 DataNode 上或者是多个 DataNode 上。



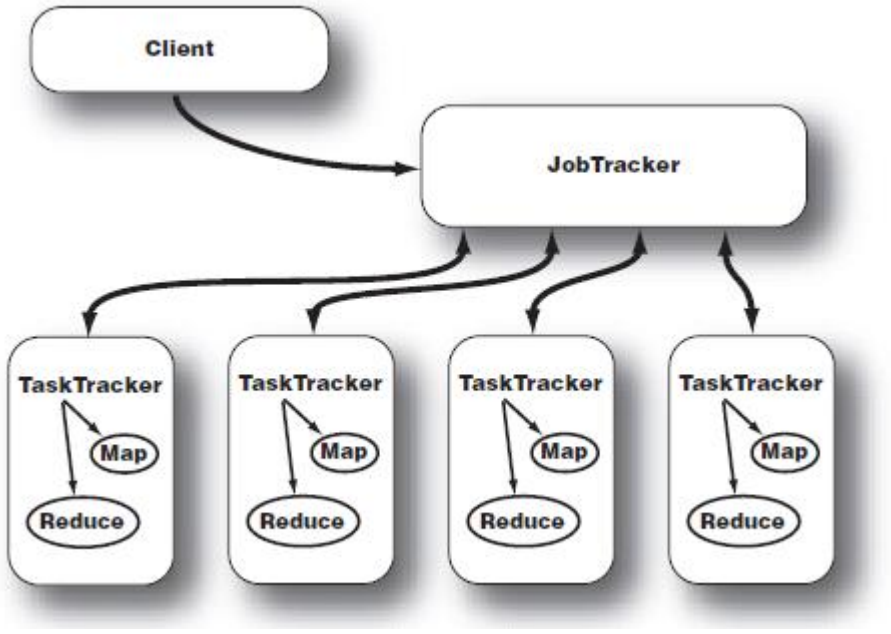
基于上面的架构需求，hadoop 采用了这种 master/slave 的架构，具体来说私有一下的几部分组成：

1. NameNode：基本上等同于 Master 的地位，复制控制底层文件的 io 操作，处理 mapreduce 任务等。

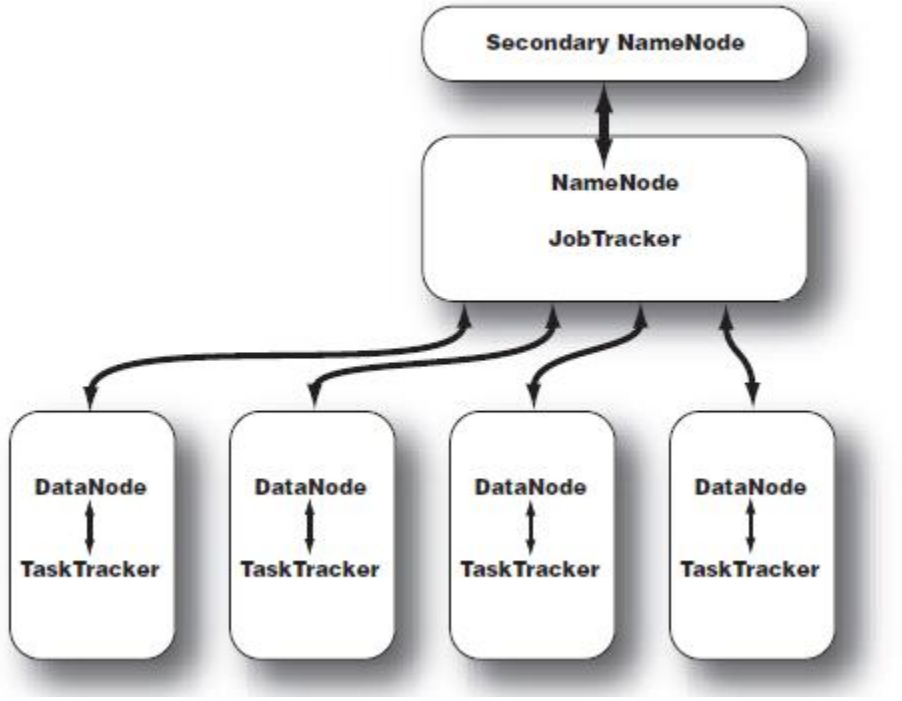
2. DataNode： 在 slave 机器上运行，负责实际的底层的文件的读写。如果客户端 client 程序发起了读 hdfs 上的文件的命令的话，那么首先将这些文件分成所谓的 block，然后 NameNode 将告知 client 这些 block 数据是存储在那些 DataNode 上的，之后，client 将直接和 DataNode 交互。

3. Secondary NameNode：该部分主要是定时对 NameNode 进行数据 snapshots 进行备份，这样尽量降低 NameNode 崩溃之后，导致数据的丢失。

4. JobTracker：该部分相当于在 client program 和 hadoop 之间的桥梁，在整个的 hadoop 系统中仅仅存在一个 JobTracker 的实例。



5. TaskTracker：TaskTracker 主要是负责的是每个具体的任务 task，如下：

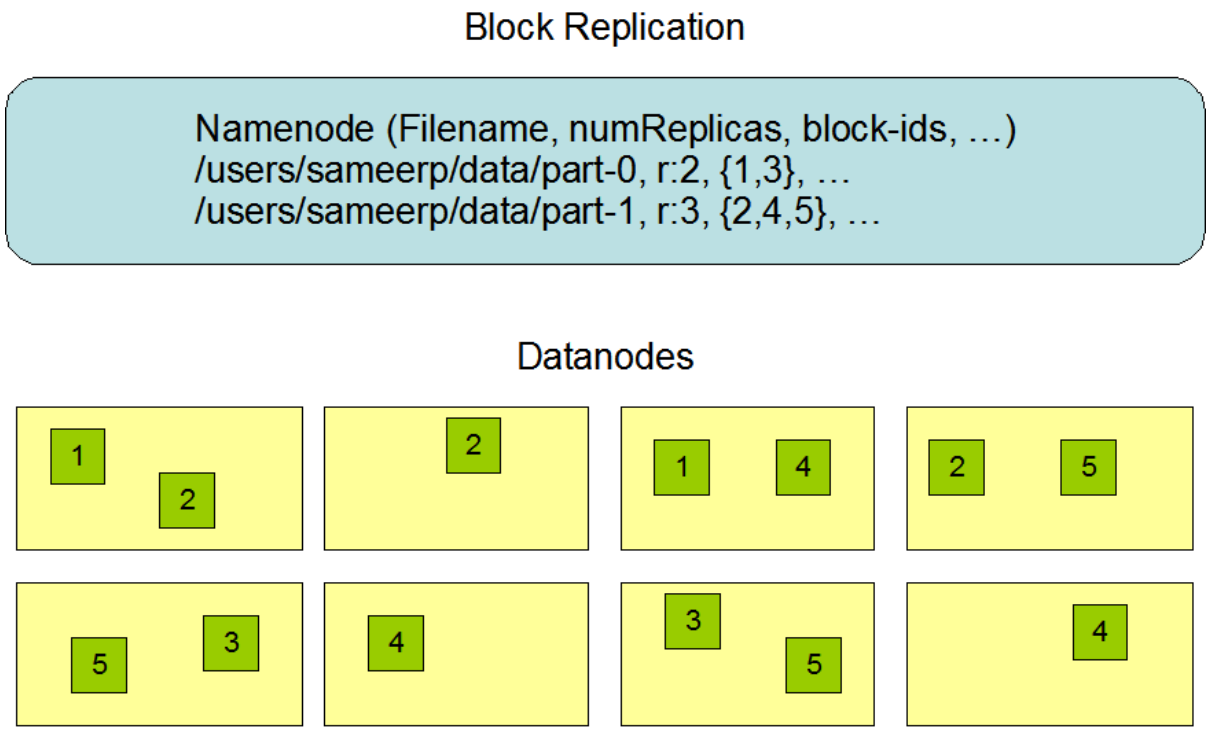


1.3 文件系统命名空间 File System Namespace

hdfs 支持传统文件系统的目录结构，应用程序能够创佳目录 directory，在这些目录中存储文件，创建文件，移动文件 remove file，重命名文件，但是不支持硬链接和软连接。

1.4 数据复制 *Data Replication*

hdfs 将一个发文件分割成 block，然后将这些 block 存储到不同的 DataNode 中，那么如何保证如果一个 DataNode 死掉，保证数据的完整性，通常的技术就是进行数据的备份，hdfs 同样使用的是这一策略。



我们现在考虑系统启动时 ,NameNode 首先进入 SafeMode ,在这种模式下是不进行数据的备份(拷贝的) 的，DataNode 向 NameNode 发送 Heartbeat 和 Blockreport，从而使得 NameNode 得到在每个 DataNode 上存储的数据文件，然后 NameNode 检查那些 block 的备份镜像数量还未达到所需备份数量，那么 NameNode 将对这些 blocks。

1.5 元数据持久化

hdfs 使用日志机制将对文件系统的操作全部存储在一个日志文件中，同时将整个文件系统信息（*the mapping of blocks to files and file system properties* ）映射成一个 FsImage 文件，该文件存储在 NameNode 主机的本地文件系统上。同时 FsImage 和 Log 支持 multiple copies，这些 hdfs 保证这些备份文件的一致性。

1.6 信息交换协议

上面讲到“DataNode 向 NameNode 发送 Heartbeat 和 Blockreport”，这其中显然涉及到协议的问题，hdfs communication 协议是构建在 tcp/ip 协议上的。客户端通过 ClientProtocol 协议和 NameNode 交换信息，NameNode 通过 DataNode Procotol 协议和 DataNode 交换信息。

<2>. 数据可访问性

2.1 web interface

hdfs 可以使用 web 页面来查看 hdfs 中文件系统中的目录 http://localhost:50075：

2.2 shell command

1. 创建目录

xuqiang@ubuntu:~/hadoop/src/hadoop-0.21.0$ ./bin/hadoop dfs -mkdir /foodir

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner
foodir	dir				2011-04-23 05:58	rwxr-xr-x	xuqiang
jobtracker	dir				2011-04-23 05:47	rwxr-xr-x	xuqiang
tmp	dir				2011-04-23 05:47	rwxr-xr-x	xuqiang

2. 删除目录

xuqiang@ubuntu:~/hadoop/src/hadoop-0.21.0$ ./bin/hadoop dfs -rmr/foodir

3. 上传文件

xuqiang@ubuntu:~/hadoop/src/hadoop-0.21.0$ ./bin/hadoop dfs -put./conf/* /foodir

4. 查看文件

xuqiang@ubuntu:~/hadoop/src/hadoop-0.21.0$ bin/hadoop dfs -cat

/foodir/capacity-scheduler.xml

5. 删除文件

xuqiang@ubuntu:~/hadoop/src/hadoop-0.21.0$ bin/hadoop dfs -rm

/foodir/capacity-scheduler.xml

更多操作：http://hadoop.apache.org/common/docs/current/file_system_shell.html

//----------

最后更新时间：2011-6-1 儿童节啊