> 真正的学习是去探索，思考和重建

HDFS产生背景

> 随着数据量越来越大，在一个操作系统存不下所有的数据，那么就分配到更多的操作系统管理的磁盘中，但是不方便管理和维护，迫切需要一种系统来管理多台机器上的文件，这就是分布式文件管理系统。HDFS只是分布式文件管理系统中的一种。

HDFS定义

> HDFS（Hadoop Distributed File System），它是一个文件系统，用于存储文件，通过目录树来定位文件；其次，它是分布式的，由很多服务器联合起来实现其功能，集群中的服务器有各自的角色

关键词：文件系统，分布式

使用场景

> 适合一次写入，多次读出的场景，且不支持文件的修改。适合用来做数据分析，并不适合用来做网盘应用

优点

- **高容错性**

```
（1）数据自动保存多个副本。它通过增加副本的形式，提高容错性。
（2）某一个副本丢失以后，它可以自动恢复
```

- **适合处理大数据**

```
（1）数据规模：能够处理数据规模达到GB、TB、甚至PB级别的数据：
（2）文件规模：能够处理百万规模以上的文件数量，数量相当之大
（3）可构建在廉价机器上，通过多副本机制，提高可靠性
```

缺点

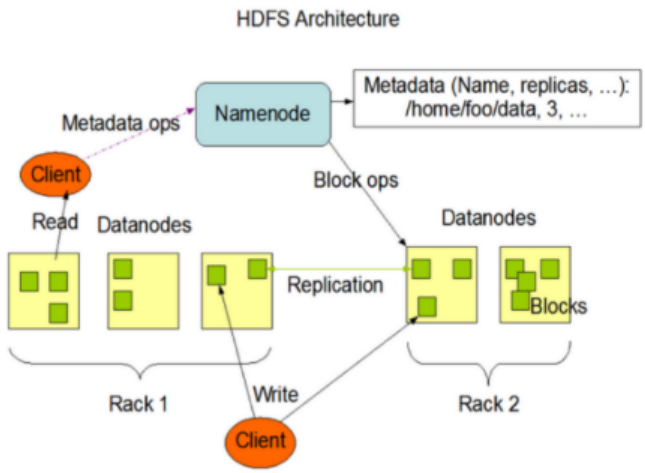
- **不适合低延时数据访问，比如毫秒级的存储数据，是做不到的**
- **无法高效的对大量小文件进行存储**

```
（1）存储大量小文件的话，它会占用 Namenode大量的内存来存储文件目录和块信息。这样是不可取的，因为 Namenode的内存总是有限的：
（2）小文件存储的寻址时间会超过读取时间，它违反了HDFS的设计目标。
```

- **不支持并发写入、文件随机修改HDFS**

```
（1）一个文件只能有一个写，不允许多个线程同时写：
（2）仅支持数据 append（追加），不支持文件的随机修改
```

HDFS组成架构图



1）NameNode（nn）：就是Master，它是一个主管、管理者。

（1）管理HDFS的名称空间；

（2）配置副本策略；

（3）管理数据块（Block）映射信息；

（4）处理客户端读写请求。

2）DataNode：就是Slave。NameNode下达命令，DataNode执行实际的操作。

（1）存储实际的数据块；

（2）执行数据块的读/写操作。

3）Client：就是客户端。

（1）文件切分。文件上传HDFS的时候，Client将文件切分成一个一个的Block，然后进行上传；

（2）与NameNode交互，获取文件的位置信息；

（3）与DataNode交互，读取或者写入数据；

（4）Client提供一些命令来管理HDFS，比如NameNode格式化；

（5）Client可以通过一些命令来访问HDFS，比如对HDFS增删查改操作；

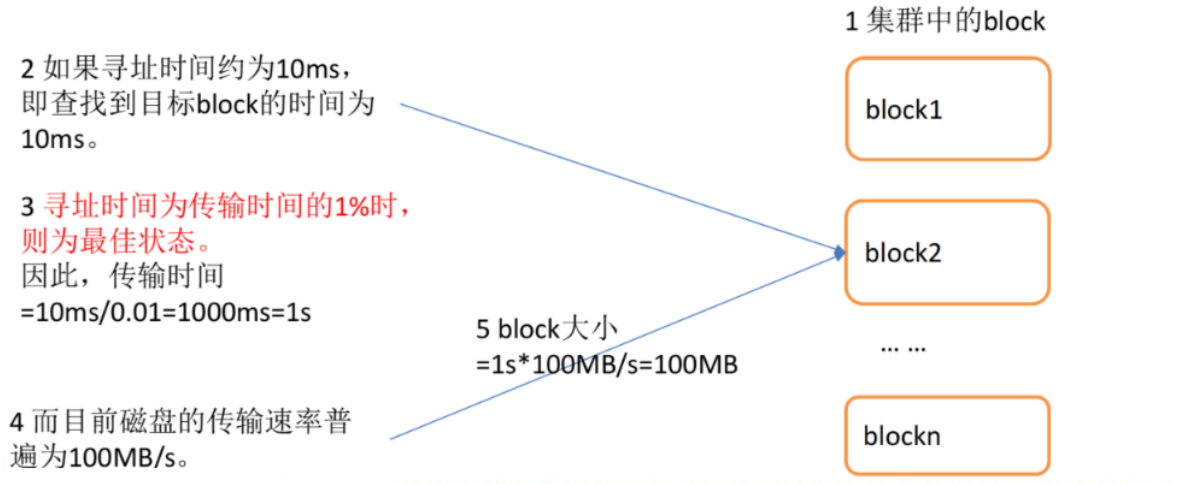
4）Secondary NameNode：并非NameNode的热备。当NameNode挂掉的时候，它并不能马上替换NameNode并提供服务。

（1）辅助NameNode，分担其工作量，比如定期合并Fsimage和Edits，并推送给NameNode ；

（2）在紧急情况下，可辅助恢复NameNode。

HDFS文件块大小

HDFS中的文件在物理上是分块存储（Block），块的大小可以通过配置参数(dfs.blocksize)来规定，默认大小在Hadoop2.x版本中是128M，老版本中是64M。



思考：快为什么不能设置太小，也不能设置太大呢？

> （1）HDFS的块设置太小，会增加寻址时间，程序一直在找块的开始位置；
> （2）HDFS的块比磁盘的块大，其目的是为了最小化寻址开销；
> （3）如果块设置的太大，从磁盘传输数据的时间会明显大于定位这个块开始位置所需的时间。导致程序在处理这块数据时，会非常慢。

> 总结：HDFS块的大小设置主要取决于磁盘传输速率。