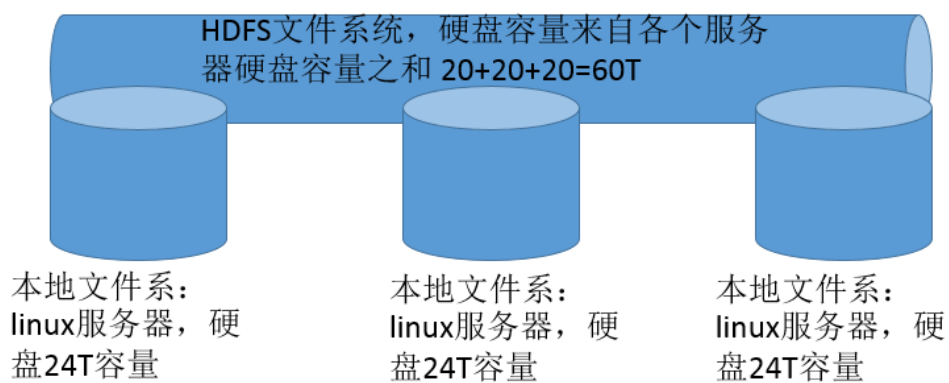


# HDFS 入门介绍

## 1、HDFS 介绍

HDFS 是 Hadoop Distribute File System 的简称，意为：Hadoop 分布式文件系统。是 Hadoop 核心组件之一，作为最底层的分布式存储服务而存在。

分布式文件系统解决的问题就是大数据存储。它们是横跨在多台计算机上的存储系统。分布式文件系统在大数据时代有着广泛的应用前景，它们为存储和处理超大规模数据提供所需的扩展能力。



## 2、HDFS 的特性

首先，它是一个文件系统，用于存储文件，通过统一的命名空间目录树来定位文件；

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。

## 2.1、 master/slave 架构

HDFS 采用 master/slave 架构。一般一个 HDFS 集群是有一个 Namenode 和一定数目的 Datanode 组成。Namenode 是 HDFS 集群主节点，Datanode 是 HDFS 集群从节点，两种角色各司其职，共同协调完成分布式的文件存储服务。

## 2.2、 分块存储

HDFS 中的文件在物理上是分块存储 (block) 的，块的大小可以通过配置参数来规定，默认大小在 hadoop2.x 版本中是 128M。

## 2.3、 名字空间 (NameSpace)

HDFS 支持传统的层次型文件组织结构。用户或者应用程序可以创建目录，然后将文件保存在这些目录里。文件系统名字空间的层次结构和大多数现有的文件系统类似：用户可以创建、删除、移动或重命名文件。

Namenode 负责维护文件系统的名字空间，任何对文件系统名字空间或属性的修改都将被 Namenode 记录下来。

HDFS 会给客户端提供一个统一的抽象目录树，客户端通过路径来访问文件，例如：hdfs://namenode:port/dir-a/dir-b/dir-c/file.data。

## 2.4、 Namenode 元数据管理

我们把目录结构及文件分块位置信息叫做元数据。Namenode 负责维护整个 hdfs 文件系统的目录树结构，以及每一个文件所对应的 block 块信息 (block 的 id, 及所在的 datanode 服务器)。

元数据信息：描述数据的数据

第一个问题：如果有一堆书，如何快速查找到我需要的时平凡的世界 书本的分类，书本的编号，书本所在的书架，书本的位置，如果记录了这些信息，就可以快速找到对应的书本

这些信息，描述了我们需要的书本在哪里，确定了这些描述信息，就可以唯一定位到这本书

如何区分关羽与王熙凤：性别，外观样貌信息（丹凤眼） 长头发，短头发，都是通过一些描述信息，来区分每一个人的

如何设计一个文件系统：

第一个：盘符

第二个：文件名

第三个：文件的类型

第四个：文件大小

第五个：创建时间修改时间

第六个：所属权限

第七个：文件的路径

这些数据都是用于描述一个文件或者文件夹，只要有了这些描述信息，那么我们就能定位到一个唯一的文件或者文件夹

这些都是一些描述数据，叫做我们的元数据信息

元数据信息：文件名称，文件路径，文件的大小，文件的权限

每一个文件或者每一个文件夹都会产生一份元数据信息

只要抓住了元数据信息，就抓住了磁盘上面存储的所有的文件或者文件夹

## 2.5、Datanode 数据存储

文件的各个 block 的具体存储管理由 datanode 节点承担。每一个 block 都可以在多个 datanode 上。Datanode 需要定时向 Namenode 汇报自己持有的 block 信息。存储多个副本（副本数量也可以通过参数设置 `dfs.replication`，默认是 3）。

## 2.6、副本机制

为了容错，文件的所有 block 都会有副本。每个文件的 block 大小和副本系数都是可配置的。应用程序可以指定某个文件的副本数目。副本系数可以在文件创建的时候指定，也可以在之后改变。

## 2.7、一次写入，多次读出

HDFS 是设计成适应一次写入，多次读出的场景，且不支持文件的修改。

正因为如此，HDFS 适合用来做大数据分析的底层存储服务，并不适合用来做网盘等应用，因为，修改不方便，延迟大，网络开销大，成本太高。