

hadoop设计思想

- - [hadoop组成](#):
 - [HDFS介绍](#)
 - [分块存储](#): 每个块是一个block
 - [备份机制](#): 默认的备份个数dfs.replication=3
 - [HDFS整体框架](#)
 - [hadoop优缺点](#):

一个分布式开源框架，支持成千上万的节点

每个节点依赖本地的计算和存储

在应用层面提供高可用

硬件错误视为一个常态

hadoop组成:

- common: 工具包
- hdfs: 分布式文件系统
 - 架构: 主从架构
 - 主——**namenode** 一个
 - 从——**datanode** 多个
 - 助理——**secondarynamenode** 分担namenode压力
- MapReduce: 分布式计算框架，有计算任务时才有相应的进程
- yarn: 集群的资源调度框架，负责集群的资源管理
 - 框架: 主从框架
 - 主——**resourcemanager** 负责统筹资源
 - 从——**nodemanager**

HDFS的目录结构和Linux操作系统的类似，以/为根节点，我们将这个目录树成为抽象目录树

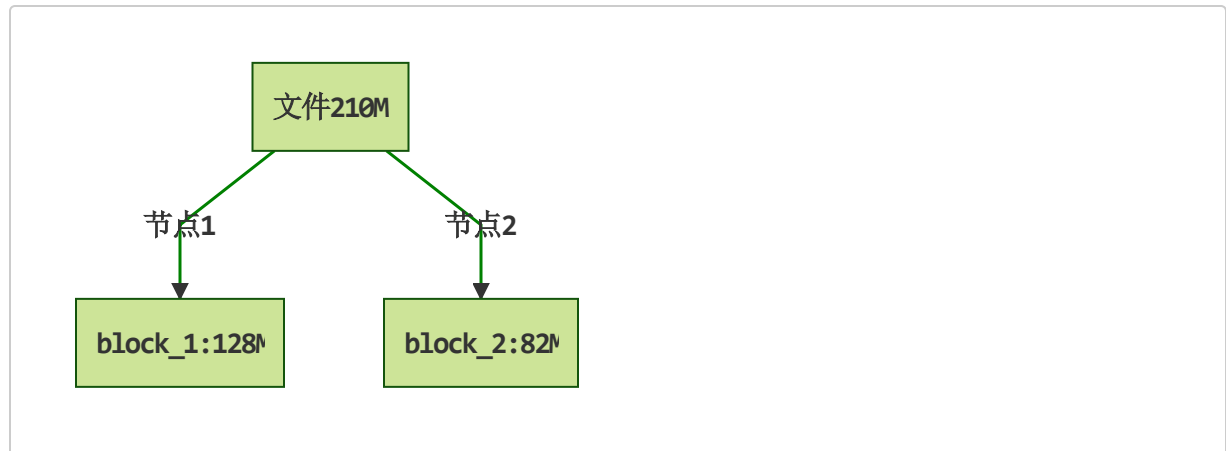
hdfs底层存储的数据块是有编号且全局唯一，存在真正的物理节点上。

HDFS介绍

设计思想：分块存储 备用机制

分块存储：每个块是一个block

hadoop2.X默认切分的块的大小是128M，块的大小就是存储数据的实际大小



- 分块设计需要考虑：**负载均衡**（各个节点上承担的数据或计算压力是否均衡）

影响负载均衡的因素：

- (1) 块的大小：128M
- (2) 所有块的副本是否均匀地分配到节点上，均匀和硬件配置有关

备份机制：默认的备份个数dfs.replication=3

假设集群中的节点4个，配置的副本3个，有一个副本的机器宕机了，副本个数小于设定个数，这时会进行复制达到3个副本。若宕机节点恢复，集群在等待一段时间后会自动删除一个副本。

- 所有备份**地位相同**没有主次之分

HDFS整体框架

namenode:

用于存储元数据，包括：

- 1) 抽象目录树
- 2) 存储数据和block的对应关系
- 3) block存储的位置

用于处理客户端的读写请求:

读——下载, 写——上传

datanode:

负责真正的数据存储, 存储数据的block

secondarynamenode:

冷备份节点, 当namenode宕机的时候, secondarynamenode不能主动切换为namenode

secondarynamenode存储的数据和namenode中相同 (备份恢复到namenode中不可避免出现数据丢失)

- (1) namenode宕机的时候帮助namenode恢复
- (2) 帮助namenode做一些事情分担namenode的压力

hadoop优缺点:

优点: 略

缺点:

1. 不支持低延迟的数据访问, 不支持实时数据访问
2. 不擅长存储大量的小文件
 - 1) 寻址时间可能会大于读取数据的时间
 - 2) 会造成元数据存储量过大, 增加namenode的压力
3. 不支持文件内容修改

