**HDFS存储原理**

1. **冗余数据保存问题**

底层是非常廉价的机器，出故障是常态

以快为单位进行冗余保存备份，默认是3块，但是也可以个性化配置，在伪分布式中冗余因子只能为一，因为名称节点和数据节点都在一台机器上，不存在冗余

优点：

加快数据传输速度

很容易检测数据错误

保证数据可靠性

1. **数据保存策略问题**

第一个快到达🡪复制三个副本🡪第一块放在上传文件的节点（如果不是集群中的节点提交的数据，HDFS会随机挑选一台磁盘不太满，CPU又不忙的一台节点存放）🡪第二副本，将它放在与第一块不同的机架上，🡪第三个副本放在同一机架的不同节点上。（个人感觉不太合适，在Hadoop权威指南上有比较严密的解释）

**Hadoop权威指南解释数据保存策略**：

Namenode如何选择在哪个datanode存储复本，需要对**可靠性**，**写入带宽**和**读取带宽**进行权衡，**将所有复本都存储在一个节点上**，损失的写入带宽最小，但并不提供真实的冗余，同时，同一个机架上的节点之间的读取带宽是很高的；**另一个极端是将复本存储在不同的数据中心**，可以最大限度的提高冗余，但带宽的损耗非常大（到目前为止，所有的Hadoop集群都运行在同一个数据中心）。

在同一个数据中心的布局策略：默认在运行**客户端的节点上存储第一个复本**（如果客户端运行在集群之外，就随机选择一个磁盘不太满，CPU不太忙的一个节点进行存放），**第二个复本放在与第一个复本不同且随机另外选择的机架上**的节点存储，第**三个复本与第二个复本存放在同一个机架上，随机选择另一个节点**，其他复本放在集群中随机选择的节点上，系统会尽量避免在同一个机架上存储太多复本。一旦选定复本的存放位置，就根据网络拓扑创建一个**管线**。

这种存储策略不仅提供很好的**稳定性**，而且实现了很好的**负载均衡**，包括**写入带宽**，**读取性能**，集群中块的**均匀分布**。

数据读取：

就近原则。

HDFS提供了一个API，可以确定一个数据节点所属机架ID，客户端可以调用

1. 数据恢复的问题

名称节点出错

FsImage和EditLog，整个HDFS实例将失效，启动备份，Hadoop2.x有热备份

**数据节点出错**

数据节点定期向名称节点发送心跳，如果隔一个周期收不到数据节点的心跳信息，名称节点就认为该数据节点宕机，标记为不可用。将该节点上的数据重新复制（冗余备份）到其他正常可用的机器上。

**数据本身出错**

校验码，客户端读取数据之后进行校验码校验，校验码是在整个文件被创建的时候，客户端每次往里写一个文件的时候都会为这个数据块生成一个校验码，并保存在同一个目录下面去。

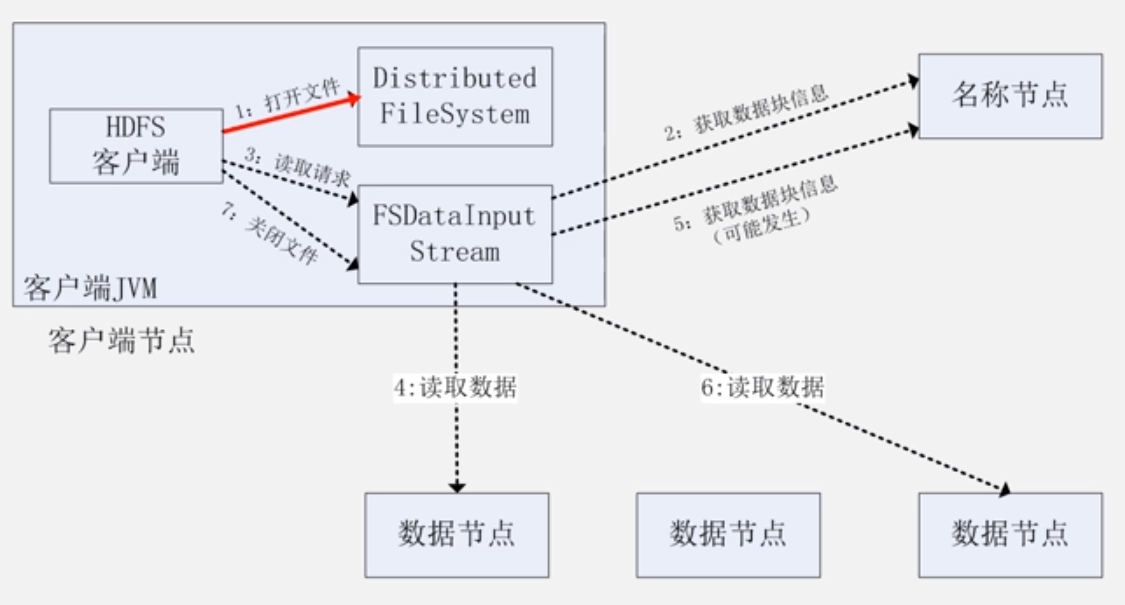
1. 数据读写过程

读取：

hdfs-site.xml

core-site.xml

🡪获取一个fs.defaultFS🡪hdfs:/ip:port



写入：

