HDFS和P2P文件分发系统的对比分析

HDFS

自底向上理解HDFS的历史

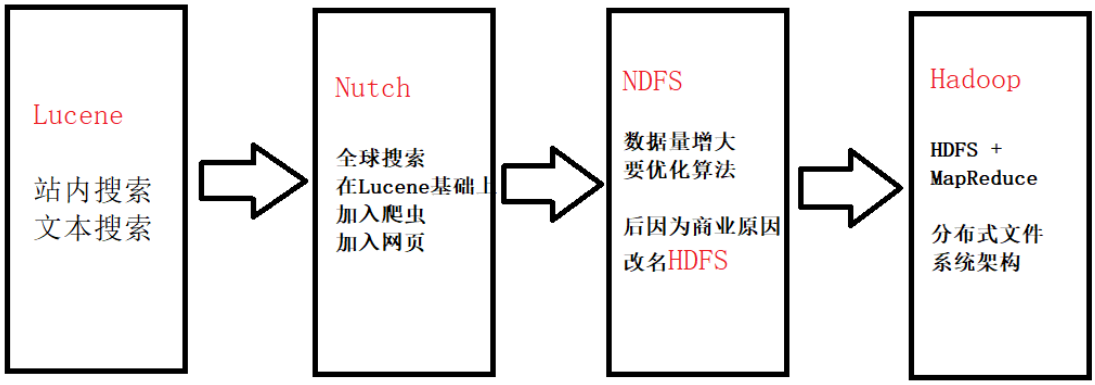
HDFS的全称是Hadoop Distributed File System，中文名叫Hadoop分布式文件系统。HDFS是Hadoop的一个组件。

1998年，Google公司成立，开始做搜索引擎技术。同时，一位名叫Cutting的美国工程师，也开始做搜索引擎技术。Cutting首先做了Lucene，Lucene是用于文本搜索的数据库，目标是为各种中小型应用软件添加全文检索的功能。

2004年，在Lucene的基础上，开发了一款可以替代当时主流的搜索引擎的开源搜索引擎，叫Nutch。Nutch是一个建立在Lucene核心之上的网页搜索应用程序，可以下载下来直接使用。它在Lucene的基础上加了网络爬虫和一些网页相关的功能，目的就是从一个简单的站内检索推广到全球网络的搜索上，就像Google一样。

随着搜索对象的数据量不断增大的问题的出现，作为搜索引擎，面临着要优化自己的搜索算法，提升搜索效率的问题。2004年，Cutting实现了NDFS，全称叫Nutch Distributed File System，中文名叫Nutch分布式文件存储系统。

后来，Cutting将NDFS和另一个技术MapReduce一起进行了升级改造，并重新命名为Hadoop。NDFS也改名为HDFS，Hadoop Distributed File System。

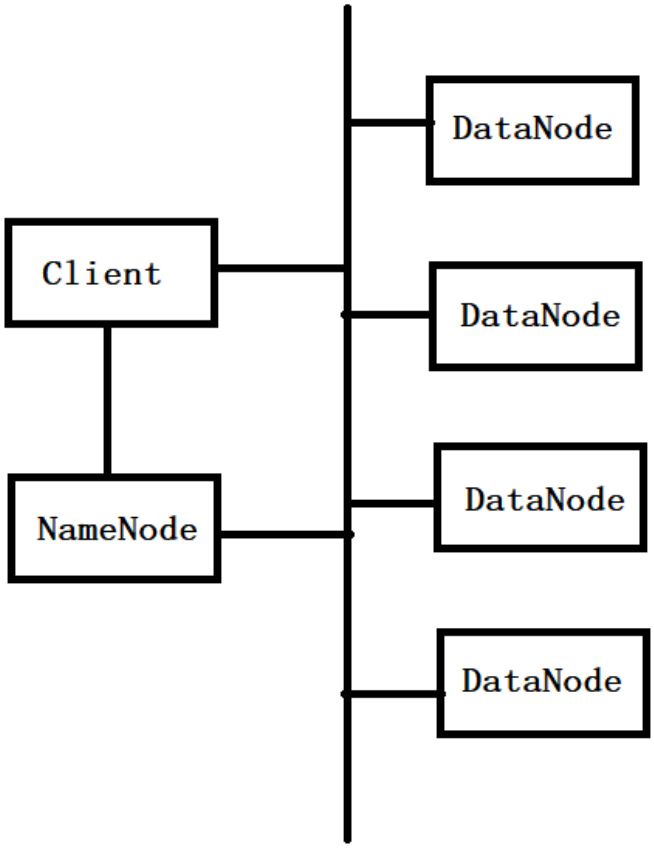


HDFS的组成部分

Hadoop是一个分布式文件系统框架，能够进行高速运算和存储。负责运算的组件是MapReduce，负责存储的组件是HDFS。

HDFS有三个组成部分，分别是NameNode、DataNode和Client。

- NameNode：是主节点，系统中的管理者，主要负责管理文件系统的命名空间、集群配置信息和存储块的复制等。NameNode保存了文件信息、文件对应的文件块信息和文件所在DataNode的信息等。
- DataNode：是从节点，储存文件的基本单元，它将基本读写单元——块存储在本地文件系统中，保存了块的元数据，同时周期性地将所有块地信息发送给NameNode。
- Client：Client是和用户交互的部分，它可以和NameNode交互，获取文件位置信息。也可以和DataNode交互，读取和写入数据。

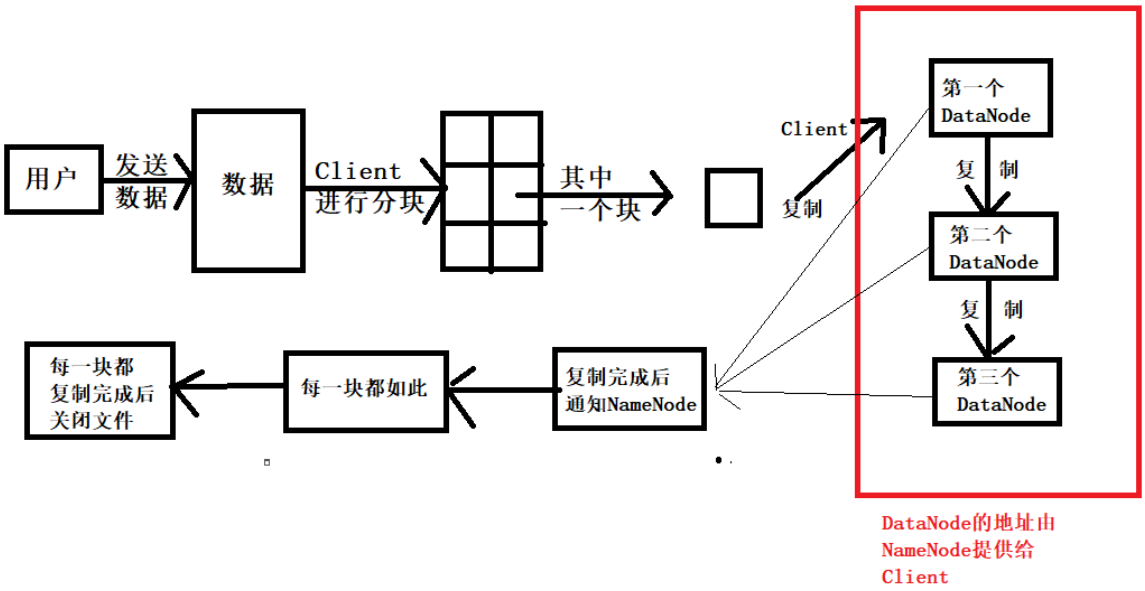


Block：中文名叫块，是HDFS的基本读写单元，HDFS的文件都是被切割成块进行存储的。块的大小通常为64MB，也可以由Client定义。数据分成多个块，每个块被复制到3个DataNode中，也可以由Client定义复制到多少个DataNode中。

HDFS的工作原理

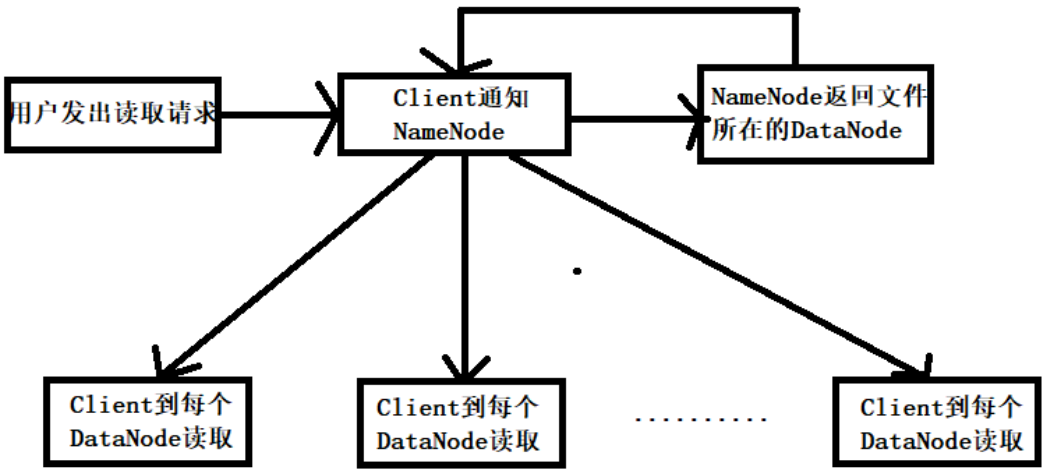
写入流程

1. 用户向Client提出文件存储请求。
2. Client制定计划：将数据按照64MB为块，进行切割；所有的块都保存三份。
3. Client将大文件切分成块。
4. Client通知NameNode，将每一个块复制三份。
5. NameNode返回三个DataNode的地址给Client，并且根据DataNode到Client的距离，进行了排序。
6. Client把块发给第一个DataNode。
7. 第一个DataNode将块复制给第二个DataNode，第二个DataNode将块复制给第三个DataNode。
8. 如果DataNode完成了写入数据的任务，就向NameNode反馈。
9. 每一个块都写入后，关闭文件。



读取流程

1. 用户向Client提出读取请求。
2. Client向NameNode请求这个文件的所有信息。
3. NameNode将给Client这个文件的块列表，以及存储各个块的数据节点清单。
4. Client从距离最近的数据节点下载所需的块。



HDFS的特点

优点：

- 高容错性。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。某一个副本丢失以后，它可以自动恢复，这是由 HDFS 内部机制实现的。
- 适合批处理。它是通过移动计算而不是移动数据，它会把数据位置提供给计算框架。

- 适合大数据处理。能够处理百万规模以上的文件数量，数量相当之大，而且能够处理10K节点的规模。
- 流式文件访问。一次写入，多次读取。文件一旦写入不能修改，只能追加，它能保证数据的一致性。
- 可构建在廉价机器上。它通过多副本机制，提高可靠性。它提供了容错和恢复机制。比如某一个副本丢失，可以通过其它副本来恢复。

缺点：

- 低延时数据访问不可行。比如毫秒级的来存储数据，这是不行的，它做不到。它适合高吞吐率的场景，就是在某一时间内写入大量的数据。但是它在低延时的情况下是不行的，比如毫秒级以内读取数据，这样它是很难做到的。
- 小文件存储不可行。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。小文件存储的寻道时间会超过读取时间，它违反了HDFS的设计目标。
- 不能并发写入、文件随机修改。一个文件只能有一个写，不允许多个线程同时写。仅支持数据追加，不支持文件的随机修改。

P2P文件分发

分布式系统的概念由来已久，现在越发火爆。分布式系统一定是由多个节点组成的系统，其中节点指的是计算机服务器，而且这些节点一般不是孤立的，而是互通的。这些连通的节点上部署了我们的节点，并且相互的操作会有协同。分布式系统对于用户而言，他们的感觉就像是面对一个服务器，提供用户需要的服务而已，而实际上这些服务是通过背后的众多服务器组成的一个分布式系统，因此分布式系统看起来想是超级计算机一样。分布式系统的核心理念是让多台服务器协同工作，完成单台服务器无法处理的任务，尤其是高并发或者大数据量的任务。但通常情况下，分布式系统的每个节点一般会不采用高性能的服务器，而是性能相对一般的普通 PC 服务器。也就是通过横向扩展，增加更多的服务器，以此来提升分布式系统的整体性能。所以廉价高效，是分布式系统最显著的特点。

Dragonfly

Dragonfly 是一款基于 P2P 的智能镜像和文件分发工具。它旨在提高文件传输的效率和速率，最大限度地利用网络带宽，尤其是在分发大量数据时，例如应用分发、缓存分发、日志分发和镜像分发。

在阿里巴巴，Dragonfly 每个月会被调用 20 亿次，分发的数据量高达 3.4PB。Dragonfly 已成为阿里巴巴基础设施中的重要一环。

尽管容器技术大部分时候简化了运维工作，但是它也带来了一些挑战：例如镜像分发的效率问题，尤其是必须在多个主机上复制镜像分发时。

Dragonfly 在这种场景下能够完美支持 Docker 。它也兼容其他格式的容器。相比原生方式，它能将容器分发速度提高 57 倍，并让 Registry 网络出口流量降低 99.5%。

Dragonfly 能让所有类型的文件、镜像或数据分发变得简单而经济。

P2P文件分发的特点

- 非中心化：网络中的资源和服务分散在所有结点上，信息的传输和服务的实现都直接在结点之间进行，可以无需中间环节和服务器的介入，避免了可能的瓶颈。P2P的非中心化基本特点，带来了其在可扩展性、健壮性等方面的优势。
- 可扩展性：在P2P网络中，随着用户的加入，不仅服务的需求增加了，系统整体的资源和服务能力也在同步地扩充，始终能较容易地满足用户的需要。整个体系是全分布的，不存在瓶颈。理论上其可扩展性几乎可以认为是无限的。
- 健壮性：P2P架构天生具有耐攻击、高容错的优点。由于服务是分散在各个结点之间进行的，部分结点或网络遭到破坏对其它部分的影响很小。P2P网络一般在部分结点失效时能够自动调整整体拓扑，保持其它结点的连通性。P2P网络通常都是以自组织的方式建立起来的，并允许结点自由地加入和离开。P2P网络还能够根据网络带宽、结点数、负载等变化不断地做自适应式的调整。
- 高性能/价格比：性能优势是P2P被广泛关注的一个重要原因。随着硬件技术的发展，个人计算机的计算和存储能力以及网络带宽等性能依照摩尔定理高速增长。采用P2P架构可以有效地利用互联网中散布的大量普通结点，将计算任务或存储资料分布到所有结点上。利用其中闲置的计算能力或存储空间，达到高性能计算和海量存储的目的。通过利用网络中的大量空闲资源，可以用更低的成本提供更高的计算和存储能力。
- 隐私保护: 在P2P网络中，由于信息的传输分散在各节点之间进行而无需经过某个集中环节，用户的隐私信息被窃听和泄漏的可能性大大缩小。此外，目前解决Internet隐私问题主要采用中继转发的技术方法，从而将通信的参与者隐藏在众多的网络实体之中。在传统的一些匿名通信系统中，实现这一机制依赖于某些中继服务器节点。而在P2P中，所有参与者都可以提供中继转发的功能，因而大大提高了匿名通讯的灵活性和可靠性，能够为用户提供更好的隐私保护。
- 负载均衡: P2P 网络环境下由于每个节点既是服务器又是客户机，减少了对传统C/S结构服务器计算能力、存储能力的要求，同时因为资源分布在多个节点，更好的实现了整个网络的负载均衡。