北京交通大学
BEIJING JIAOTONG UNIVERSITY




《大数据概论》
大数据存储与管理

鲍鹏
软件学院

本章内容

- 分布式文件系统
- 分布式数据库
- 非关系型数据库

文件系统概述

- 操作系统中负责管理和存储文件信息的软件机构称为文件管理系统，简称文件系统。
- 文件系统对文件存储设备的空间进行组织和分配，并对存入的文件进行保护和检索。
- 文件系统负责为用户建立文件，存入、读出、修改、转储文件，控制文件的存取，当用户不再使用时撤销文件等。

文件系统分类

- 本地文件系统 (Local File System, LFS)
  - 本地主机中实际可以访问到的文件系统。
- 分布式文件系统 (Distributed File System, DFS)
  - 分布式文件系统把文件分布存储到多个计算机节点上，成千上万的计算机节点构成计算机集群。
  - 分布式文件系统管理的物理存储资源不仅存储在本地节点上，还可以通过网络连接存储在非本地节点上。

本地 vs. 分布式文件系统

特性	LFS	DFS
存储方式	直接存储	分块存储
存储结构	树结构	Master-Slaver 主从架构
数据检索	慢	快
数据备份	否	是
数据安全	低	高
数据对象	不适用于大规模文件	适用于大规模文件
存储空间	不需要额外的存储空间	需要额外的存储空间
访问方式	直接访问	加载后访问
复杂性	低	高

- 相较于本地文件系统，分布式文件系统改变了数据的存储和管理方式，具有优异的数据备份、数据安全、规模可扩展等优点。

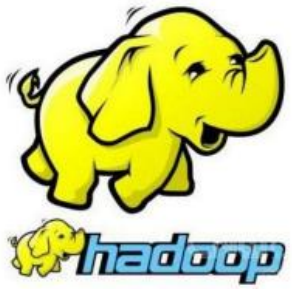
分布式文件系统

- 常见的分布式文件系统
  - HDFS
  - Ceph
  - GlusterFS
  - GFS
  - Lustre
  - mogileFS
  - FastDFS
  - TFS
  - GridFS
  - … …

分布式文件系统分类

- 从用途来看，目前主流的分布式文件系统主要有两类：
  - 第一类分布式文件系统主要面向以大文件、块数据顺序读写为特点的数据分析业务，其典型代表是Apache旗下的HDFS。
  - 另一类主要服务于通用文件系统需求并支持标准的可移植操作系统接口（Portable Operating System Interface of UNIX，POSIX），其代表包括Ceph和GlusterFS。
- 这种分类仅表示各种分布式文件系统的专注点有所不同，并非指一种分布式文件系统只能用于某种用途。

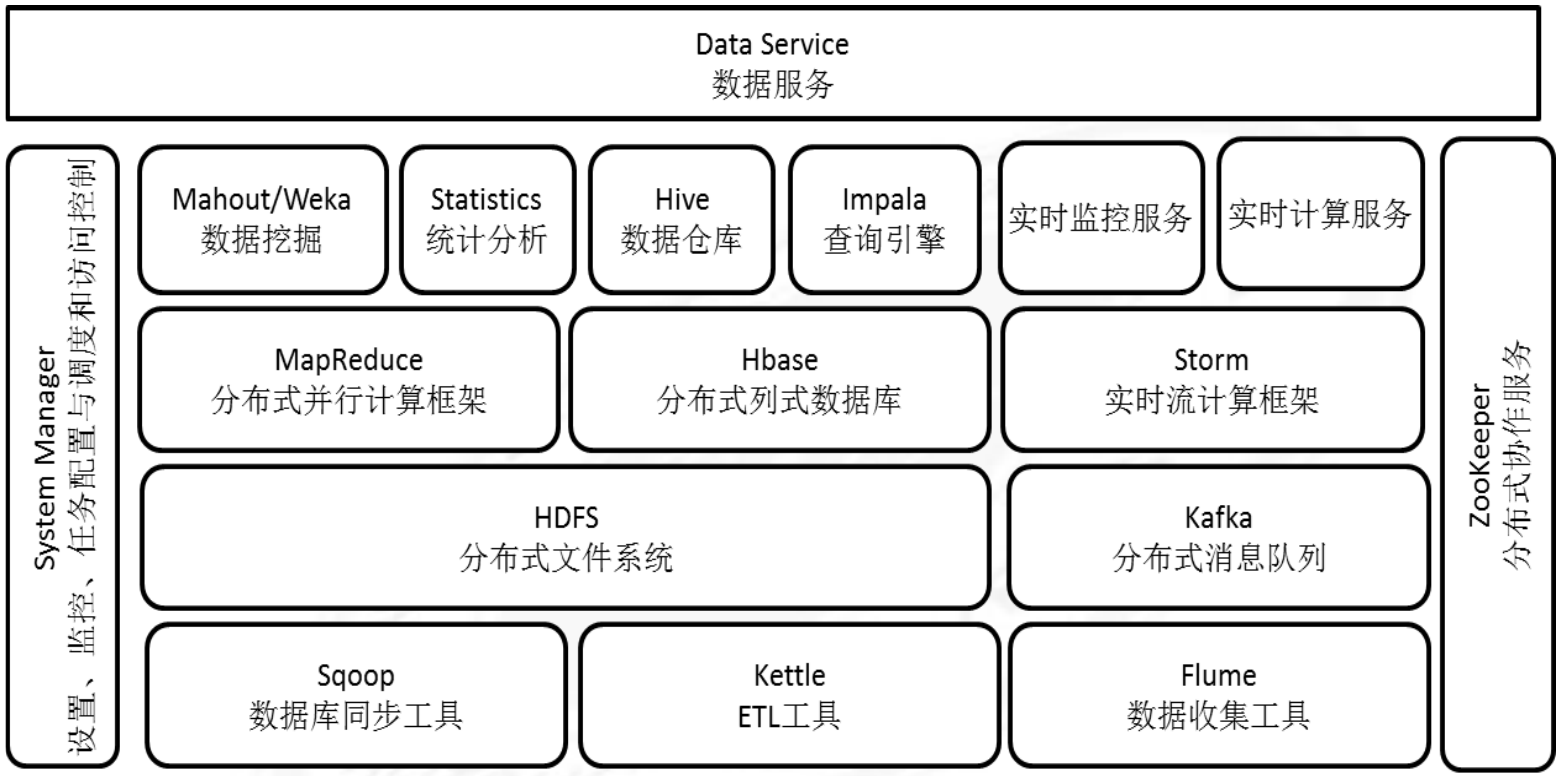
Hadoop



- Hadoop是由Apache基金会开发的分布式系统基础架构。

- Hadoop的特点
  - 存储和高速运算。
  - 核心设计：HDFS和MapReduce，HDFS (Hadoop Distributed File System) 为海量数据提供存储，MapReduce为海量数据提供计算。

Hadoop的组成



Hadoop组成示意图

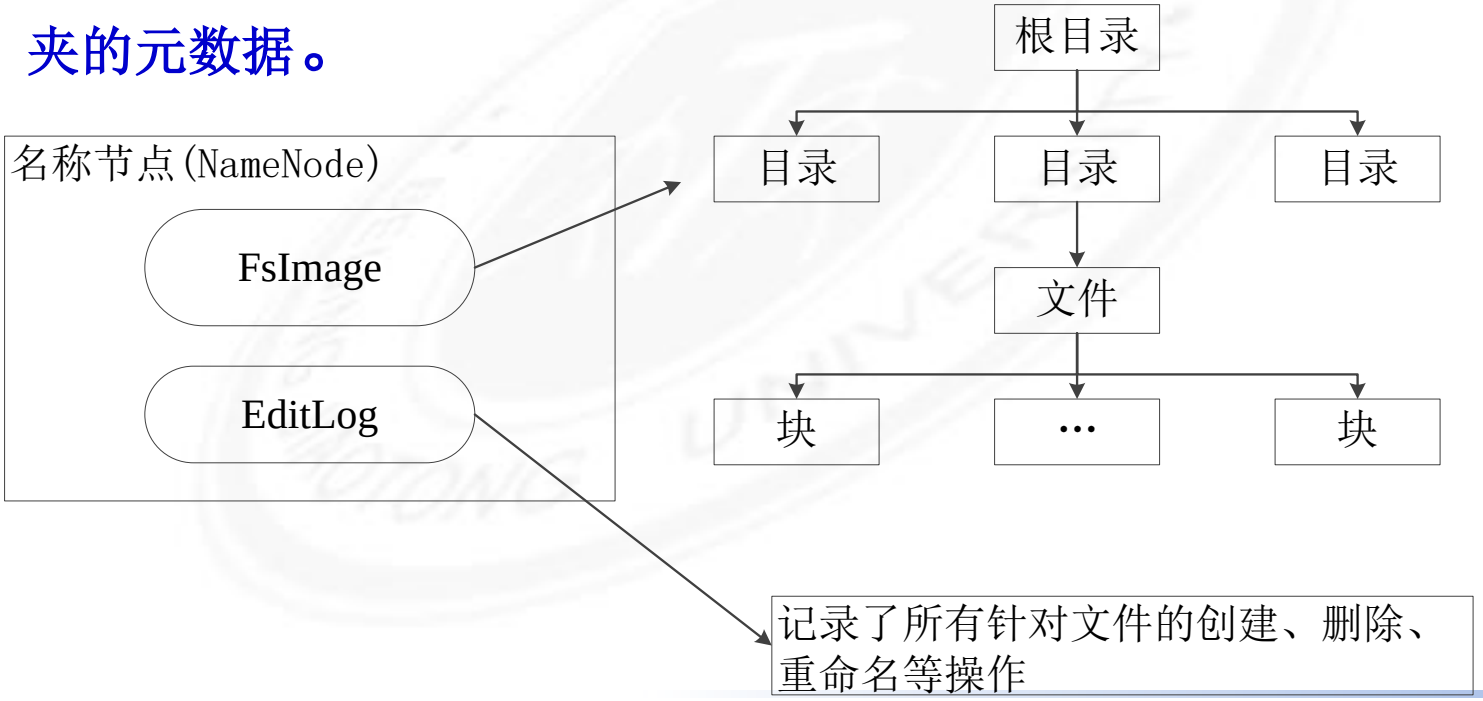
Hadoop分布式文件系统 (HDFS)

- HDFS作为Hadoop的分布式文件系统，其功能为数据的存储、管理和出错处理。设计的目的是用于可靠地存储大规模的数据集，并提高用户访问数据的效率。

A	B	C	D
适合大文件存储和处理	集群规模可动态扩展	能有效保证数据一致性	数据的吞吐量大跨平台移植性好
可处理的文件规模可达到百MB乃至数百TB，目前应用已到PB级。	存储节点可在运行状态下加入到集群中，集群仍然可以正常地工作。	基于“一次写入，多次读取”设计，简化处理文件访问方式，当一个文件创建、写入并关闭后就不能再修改。	采用数据流式读写的方式，用以增加数据的吞吐量。具有很好的跨平台移植性，源代码开放。

HDFS相关概念

- 名节点 (NameNode)
  - 也称为名称节点，用于维护和记录文件系统的命名空间，保存了两个核心的数据结构，即FsImage和EditLog。
  - FsImage用于维护文件系统树以及文件树中所有的文件和文件夹的元数据。



名节点数据结构图

HDFS相关概念

- 数据节点 (DataNode)
  - 数据节点是分布式文件系统HDFS的工作节点，负责数据的存储和读取，会根据客户端或者是名节点的调度来进行数据的存储和检索，并且向名节点定期发送自己所存储的块的列表。
  - 响应文件系统客户端发出的读写请求，同时在名节点的指导下执行数据库的创建、删除及复制。
  - 每个数据节点中的数据会被保存在各自节点的本地Linux文件系统中。

HDFS相关概念

- HDFS客户端
  - 客户端是用户操作HDFS最常用的方式，HDFS在部署时都提供了客户端。
  - HDFS客户端是一个库，暴露了HDFS文件系统接口。
  - 客户端通过一个可配置的端口向名节点主动发起TCP连接，并使用客户端协议与名节点进行交互。
  - 此外，HDFS也提供了Java API，作为应用程序访问文件系统的客户端编程接口。

HDFS相关概念

- HDFS命名空间管理
  - HDFS的命名空间包含目录、文件和块。
  - 在HDFS1.0体系结构中，在整个HDFS集群中只有一个命名空间，并且只有唯一一个名节点，该节点负责对这个命名空间进行管理。
  - HDFS使用的是传统的分级文件体系，因此，用户可以像使用普通文件系统一样，创建、删除目录和文件，在目录间转移文件，重命名文件等。

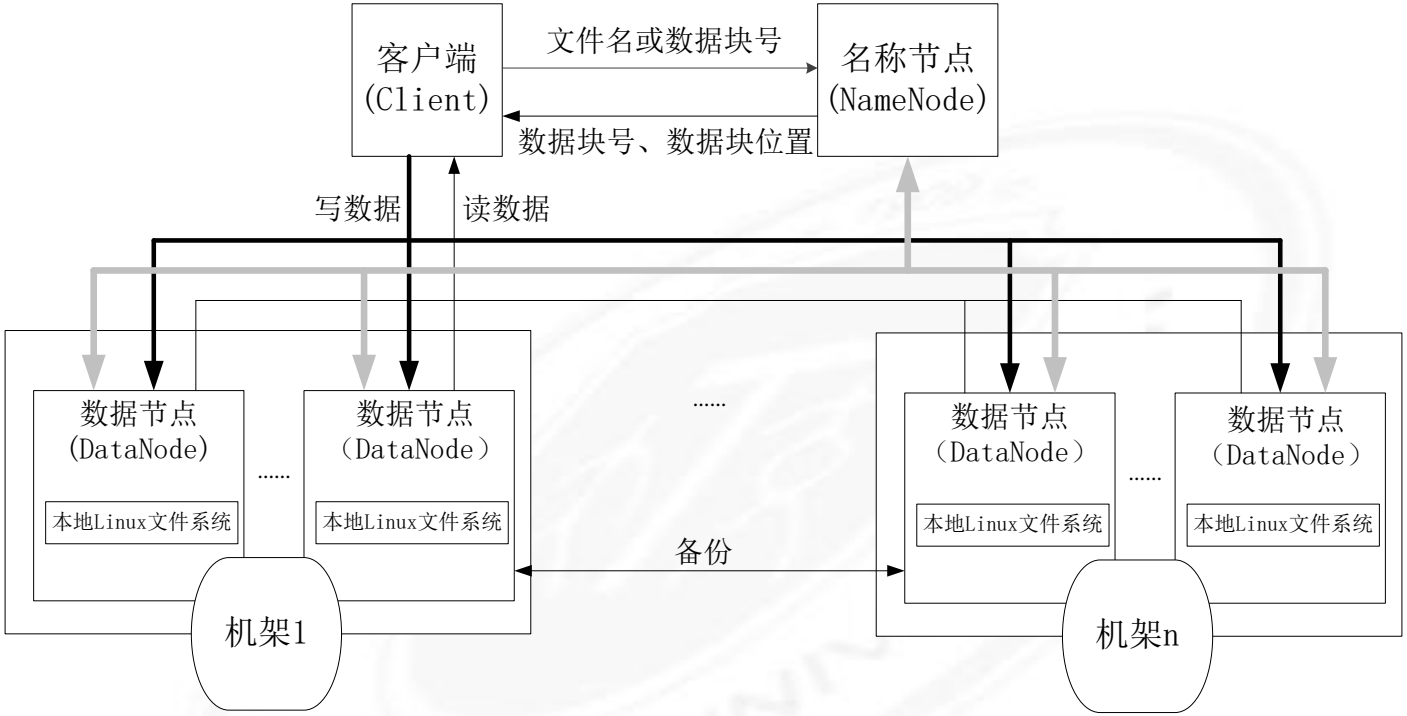
HDFS相关概念

- HDFS通信协议
  - HDFS是一个部署在集群上的分布式文件系统，因此，很多数据需要通过网络进行传输。
  - 所有的HDFS通信协议都是构建在TCP/IP协议基础之上。
  - 客户端通过一个可配置的端口向名节点主动发起TCP连接，并使用客户端协议与名节点进行交互。
  - 名节点和数据节点之间则使用数据节点协议进行交互。
  - 客户端与数据节点的交互则是通过RPC（Remote Procedure Call）来实现。

HDFS体系结构概述

- HDFS采用了主从（Master/Slave）架构，一个HDFS集群包括一个名节点和若干个数据节点。
  - 名节点作为中心服务器，负责管理文件系统的命名空间及客户端对文件的访问。
  - 集群中的数据节点负责处理文件系统客户端的读/写请求，在名节点的统一调度下进行数据块的创建、删除和复制等操作。所有的HDFS通信协议都是构建在TCP/IP协议基础之上。
  - 每个数据节点的数据实际上是保存在本地Linux文件系统中的。

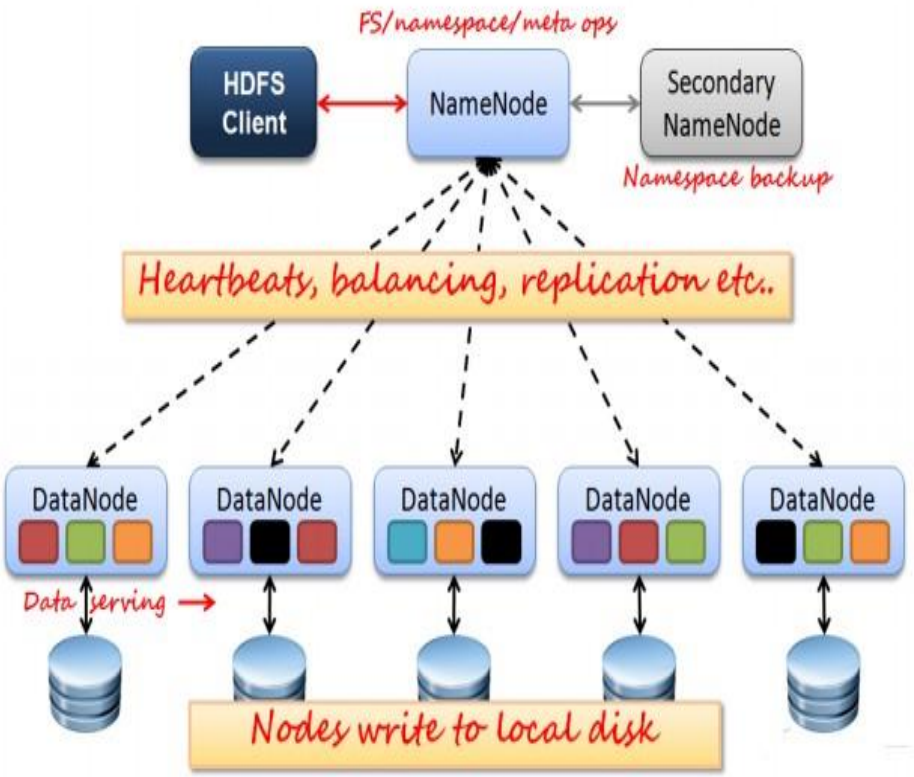
HDFS体系结构概述



HDFS体系结构示意图

HDFS体系结构

- 一个HDFS集群通常由一台主服务器和若干台数据服务器构成，有一台后备主服务器用于定期对主服务器存储的元数据进行备份，保障命名空间、元数据等系统信息的完整性。这台后备主服务器只与主服务器进行交互，对系统中的其他节点不可见。



Active Namenode

1.主Master(只有一个)
2.管理HDFS的名称空间
3.管理数据块映射信息
4.配置副本策略
5.处理客户端读写请求

Standy Namenode

1.NameNode的热备
2.定期合并fsimage和fsedit，推送给NameNode
3.当Active NameNode出现故障时，快速切换为新的Active NameNode

Datanode

1.Slave（有多个）
2.存储实际的数据块
3.执行数据块的读写

Client

1.文件切分
2.与NamdeNode交互，获取文件位置信息
3.与DataNode交互，读取或者写入数据;
4.管理HDFS 和访问HDFS

HDFS体系结构设计原理

- 元数据与数据分离
  - 名节点与数据节点分离。
- 主从结构
  - 主控件控制从控件。
- 一次写入多次读取
  - HDFS中的文件只能写入一次，确保了数据的一致性。
- 移动计算比移动数据更划算
  - 将运算的执行移到离它要处理的数据更近的地方。

HDFS数据存储

- 冗余数据保存
- 数据存取策略
- 数据错误与恢复

HDFS数据存储

- 冗余数据保存
  - 为了保证系统的容错性和可用性，HDFS采用了多副本方式对数据进行冗余存储，通常一个数据块的多个副本会被分布到不同的数据节点上，如图所示，数据块1被分别存放到数据节点A和C上，数据块2被存放在数据节点A和B上。这种多副本方式具有以下优点。

（1）加快数据传输速度

（2）容易检查数据错误

（3）保证数据可靠性

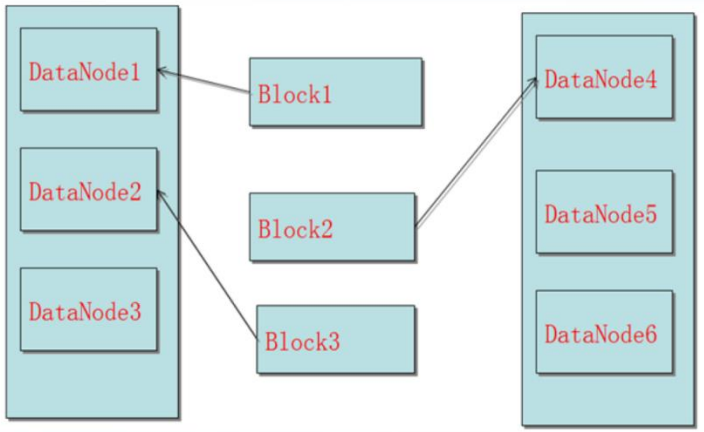
HDFS数据块多副本存储

HDFS数据存储

- 数据存取策略

（1）数据存放

- 第一个副本：放置在上传文件的数据节点；如果是集群外提交，则随机挑选一台磁盘不太满、CPU不太忙的节点。
- 第二个副本：放置在与第一个副本不同的机架的节点上。
- 第三个副本：与第一个副本相同机架的其他节点上。
- 更多副本：随机节点。



Block的副本放置策略

HDFS数据存储

- 数据存取策略

（2）数据读取

- HDFS提供了API接口以确定数据节点所属的机架ID，客户端也可以调用API获取自己所属的机架ID。
- 当客户端读取数据时，从名节点获得数据块不同副本的存放位置列表，列表中包含了副本所在的数据节点，可以调用API来确定客户端和这些数据节点所属的机架ID，当发现某个数据块副本对应的机架ID和客户端对应的机架ID相同时，就优先选择该副本读取数据，如果没有发现，就随机选择一个副本读取数据。

HDFS数据存储

- 数据错误与恢复

（1）名节点出错

– 名节点保存了所有的元数据信息，其中，最核心的两大数据结构是FsImage和Editlog，若这两个文件发生损坏，那么整个HDFS实例将失效。

– 将核心文件同步复制到备份服务器SecondaryNameNode上。当名节点出错时，可根据备份服务器SecondaryNameNode中的FsImage和Editlog数据进行恢复。

HDFS数据存储

- 数据错误与恢复

（2）数据节点出错

- 每个数据节点会定期向名节点发送“心跳”信息，向名节点报告自己的状态。
- 当数据节点发生故障，或者网络发生断网时，名节点就无法收到来自数据节点的心跳信息，这时，这些数据节点就会被标记为“宕机”，节点上面的所有数据都会被标记为“不可读”，名节点不会再给它们发送任何I/O请求。
- 名节点会定期检查，一旦发现某个数据块的副本数量小于冗余因子，就会启动数据冗余复制，为它生成新的副本。

HDFS数据存储

- 数据错误与恢复

（3）数据出错

- 网络传输和磁盘错误。
- 在文件被创建时，客户端就会对每一个文件块进行信息摘录，并把这些信息写入到同一个路径的隐藏文件里面。
- 客户端读取文件时，会先读取信息文件，然后，利用该信息文件对每个读取的数据块进行校验，如果校验出错，客户端就会请求到另外一个数据节点读取该文件块，并且向名节点报告这个文件块有错误，名节点会定期检查并且重新复制这个块。

HDFS存储原理总结

- 名节点与数据节点
  - 从HDFS系统的内部架构来看，一个文件被分成多个文件块储存在数据节点集上。
- 文件系统命名空间
  - HDFS支持传统的层级文件组织结构，任何有关文件系统的改变都会被记录。
- 数据复制
  - HDFS将文件分成大小相同的若干份存储于各个数据节点，同时在其它若干数据节点上也保存各个文件块的副本。

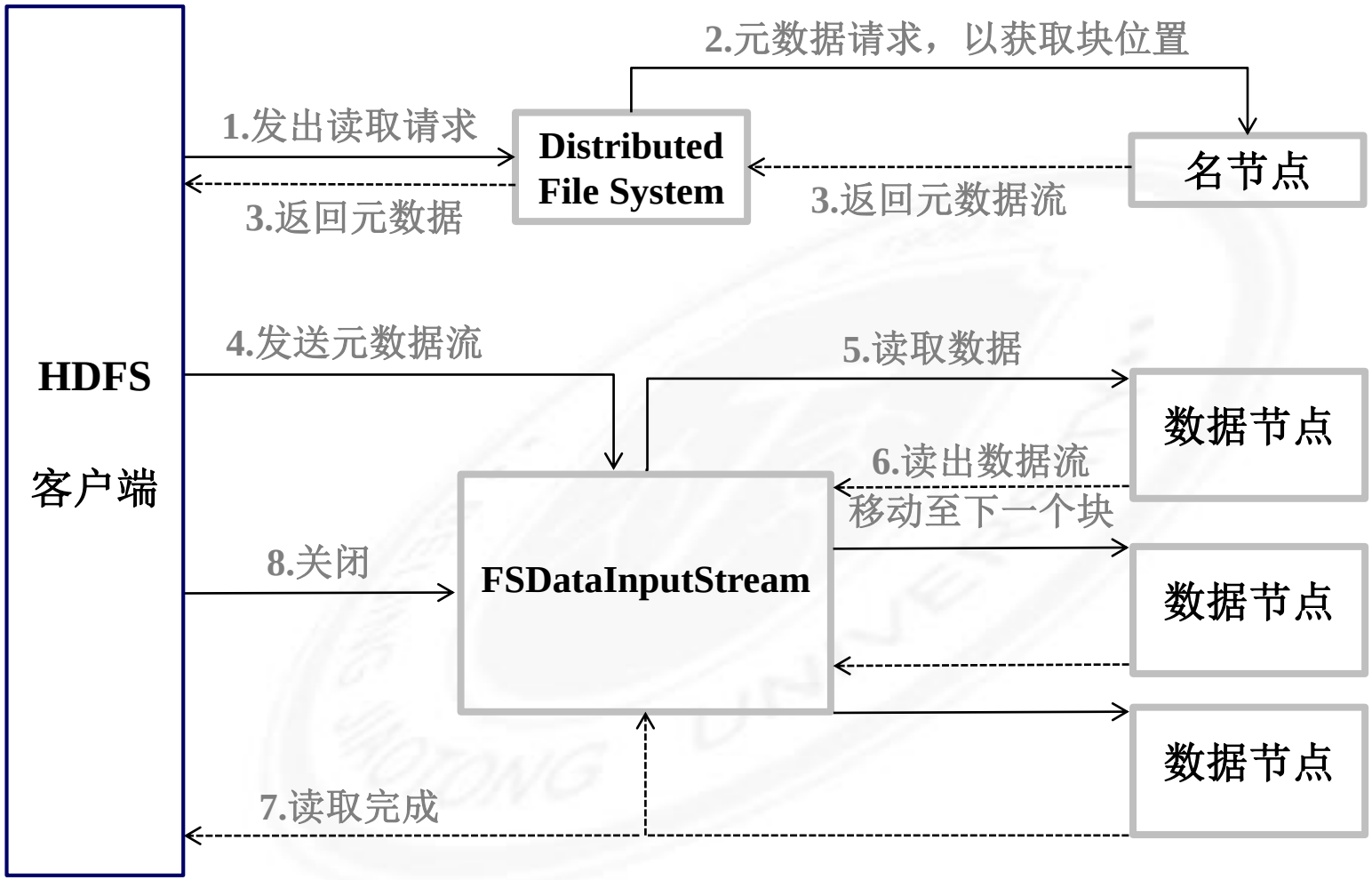
HDFS存储原理总结

- 文件系统元数据持久存储
  - HDFS文件系统的元数据信息存储在名节点上，名节点将持久记录文件系统上的变化。
- 多副本的流式复制
  - 将名节点上存储副本的节点信息循环写入数据节点。
- 心跳检测和重新复制
  - 数据节点定期向名节点报告状态。

HDFS数据读写

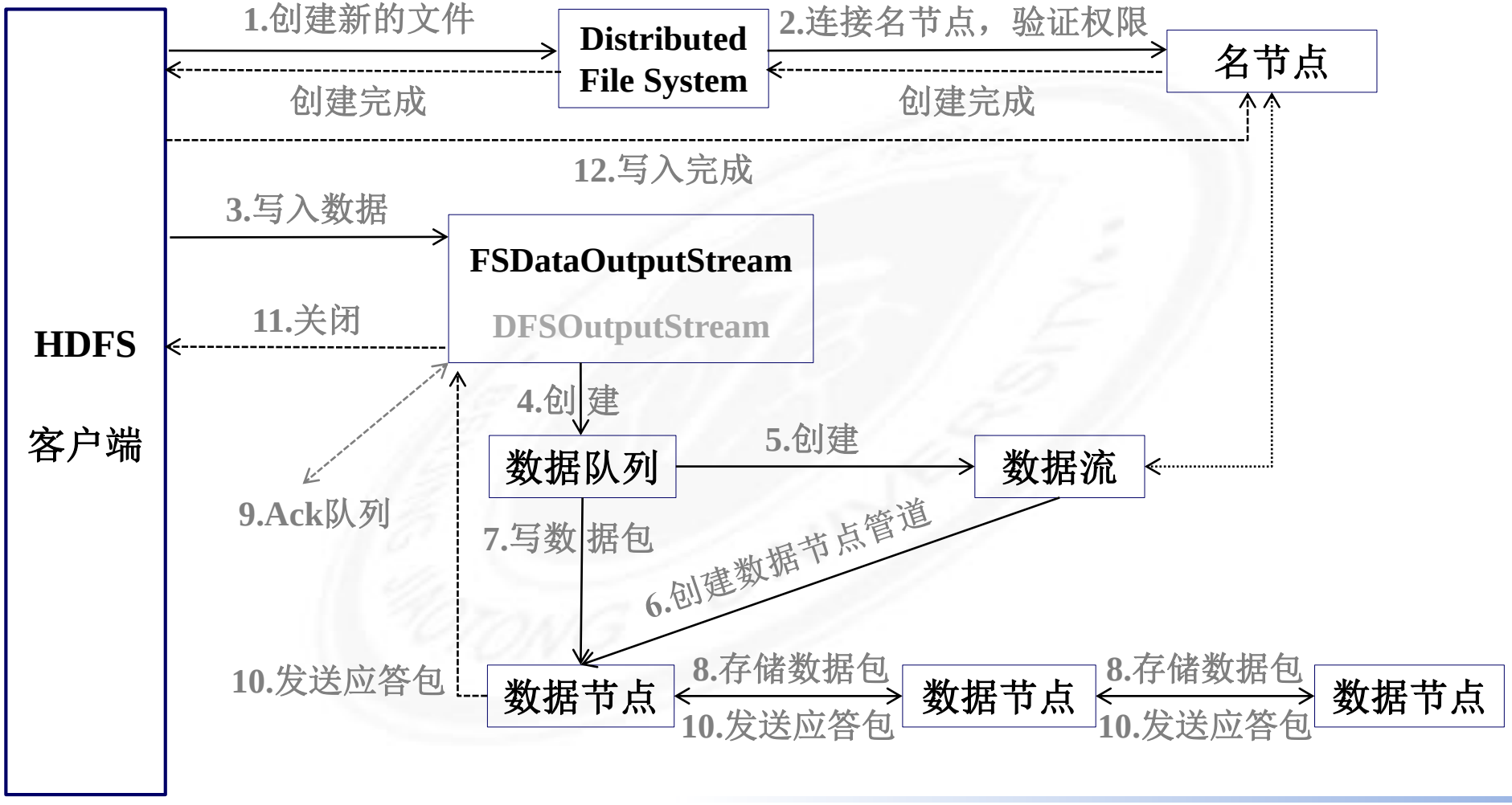
- HDFS数据文件被分成固定的块，默认块大小为64MB，读/写操作运行在块级。其优势在于：
  - 支持大规模文件存储：文件以块为单位进行存储。
  - 简化系统设计：简化了存储管理。
  - 适合数据备份：每个文件块都可以冗余存储到多个节点上，大大提高了系统的容错性和可用性。

HDFS数据读写——读操作



Hadoop读操作流程图

HDFS数据读写——写操作



Hadoop写操作流程图

HDFS总结

- HDFS的优势在于：
  - 高容错性

(1) 数据自动保存多个副本。
(2) 副本丢失以后，可以自动恢复。

  - 适合处理大规模数据

(1) 数据规模：能够处理数据规模达到GB、TB、甚至PB级别。
(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。

  - 可构建在廉价机器上，通过多副本机制，提高可靠性

HDFS总结

- HDFS存在的问题：
  - 不适合低延时数据访问，如毫秒级的存储数据
  - 无法高效的对大量小文件进行储存

(1) 储存大量小文件，会占用名节点大量的内存来存储文件目录和块信息。这样是不可取的，因为名节点的内存是有限制的。

(2) 小文件存储的寻址时间会超过读取时间，违反了HDFS的设计目标。

  - 不支持并发写入、文件随机修改

(1) 一个文件只能由一个写入，不允许多个线程同时写。

(2) 仅支持数据追加，不支持文件的随机修改。

其它常见的分布式文件系统

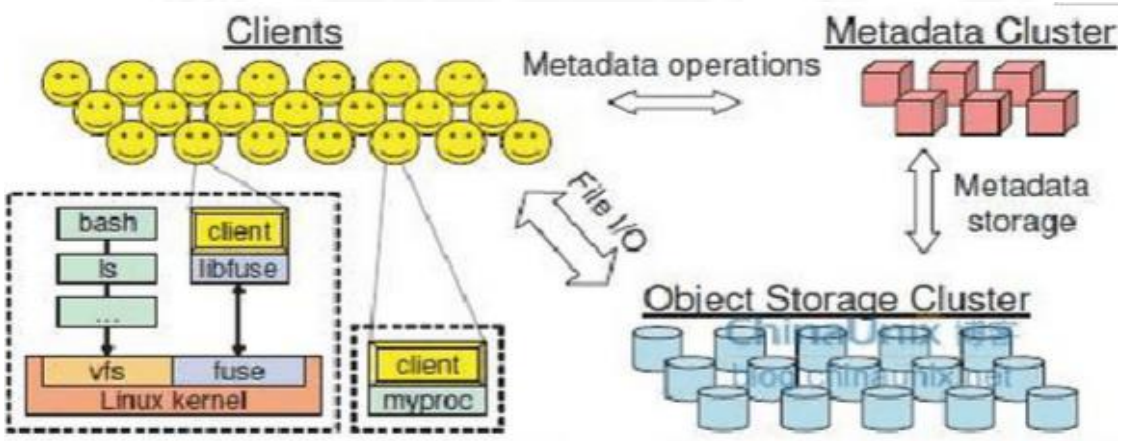
- Ceph
- GlusterFS (GUN Cluster File System)

分布式文件系统——Ceph

- Ceph是一个高可用、易于管理、开源的分布式存储系统，可以同时提供对象存储、块存储以及文件存储服务，其优势包括统一存储能力、可扩展性、可靠性、性能、自动化的维护等等。
- Ceph优势均来源于其先进的核心设计思想，可其概括为八个字——“无需查表，算算就好”。基于这种设计思想，Ceph充分发挥存储设备自身的计算能力，同时消除了对系统单一中心节点的依赖，从而实现了真正的无中心结构。
- Ceph项目起源于其创始人Sage Weil在加州大学圣克鲁兹分校攻读博士期间的研究课题。

分布式文件系统——Ceph

- 客户端通过与OSD（Object Storage Device）的直接通讯实现文件操作。在打开一个文件时，客户端会向MDS（Metadata storage ）发送一个请求。MDS把请求的文件名翻译成文件节点（inode），并获得节点号、访问模式、大小以及文件的其他元数据。如果文件存在并且客户端可以获得操作权，则MDS向客户端返回上述文件信息并且赋予客户端操作权。



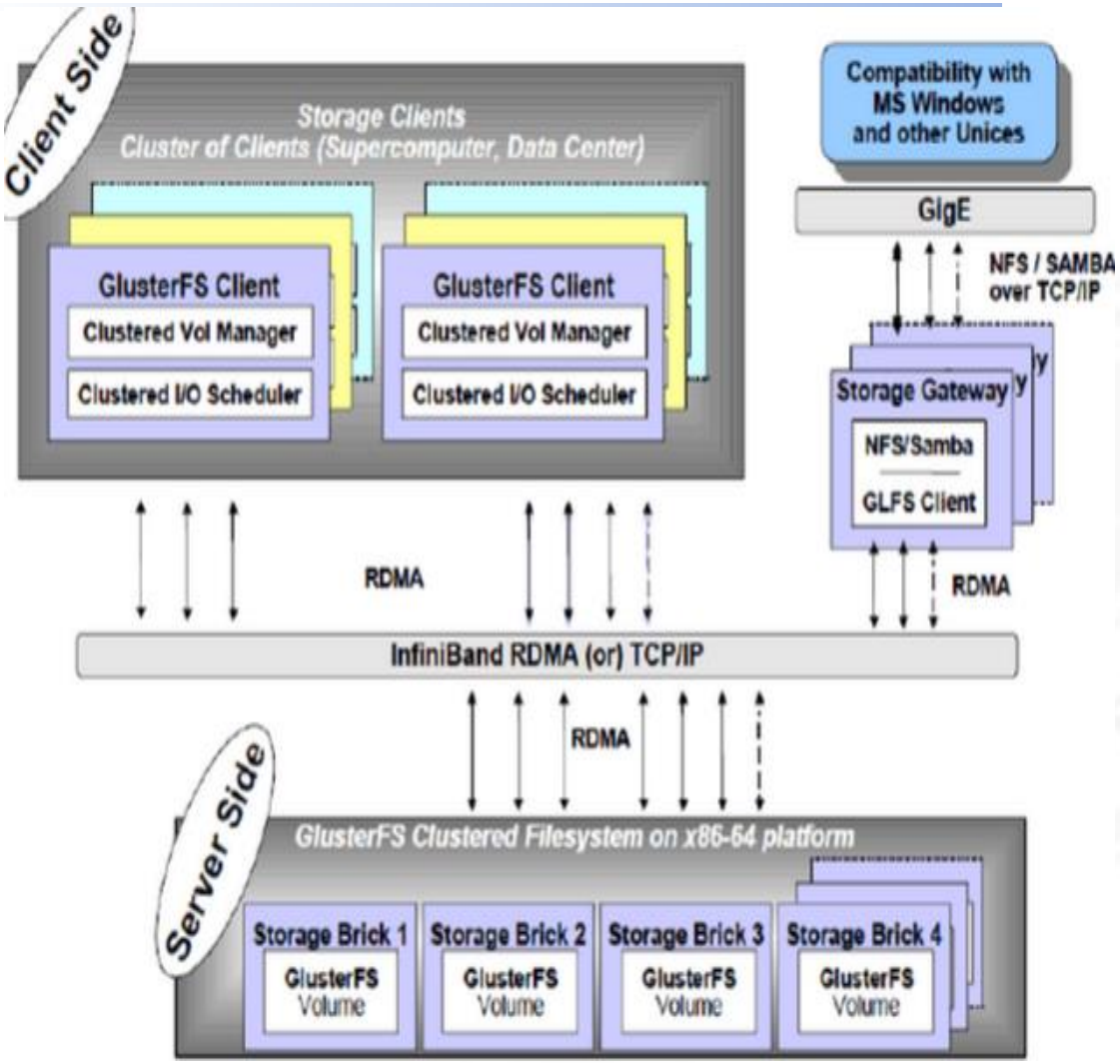
分布式文件系统——Ceph

- 相对于面向离线批处理的HDFS来说，Ceph更偏向于成为一种高性能、高可靠、高扩展性的实时分布式存储系统，其对于写入操作特别是随机写入的支持要更好。

分布式文件系统——GlusterFS

- GlusterFS是Scale-Out存储解决方案Gluster的核心，它是一个开源的分布式文件系统，具有强大的横向扩展能力，通过扩展能够支持数PB存储容量和处理数千客户端。
- GlusterFS借助TCP/IP或InfiniBand RDMA网络将物理分布的存储资源聚集在一起，使用单一全局命名空间来管理数据。GlusterFS基于可堆叠的用户空间设计，可为各种不同的数据负载提供优异的性能。

分布式文件系统——GlusterFS



- Storage Brick: GlusterFS中的存储单元，可以通过主机名和目录名来标识。
- Storage Client: 挂载了GlusterFS卷的设备
- RDMA:远程直接内存访问，支持不通过双方的OS进行直接内存访问。
- RRDNS: round robin DNS，是一种通过DNS轮转返回不同的设备以进行负载均衡的方法。
- Self-heal: 用于后台运行检测副本中文件和目录的不一致性并解决这些不一致。

分布式文件系统——GlusterFS

- GlusterFS支持运行在任何标准IP网络上标准应用程序的标准客户端，用户可以在全局统一的命名空间中使用NFS/CIFS等标准协议来访问应用数据。
- GlusterFS使得用户可摆脱原有的独立、高成本的封闭存储系统，能够利用普通廉价的存储设备来部署可集中管理、横向扩展、虚拟化的存储池，存储容量可扩展至TB/PB级。
- GlusterFS由于缺乏一些关键特性，可靠性也未经过长时间考验，还不适合应用于需要提供 24 小时不间断服务的产品环境。目前适合应用于大数据量的离线应用。

分布式文件系统对比

特性	HDFS	Ceph	GlusterFS
元数据服务器	单个 存在单点故障风险	多个 不存在单点故障风险	无 不存在单点故障风险
POSIX兼容	不完全	兼容	兼容
配额限制	支持	支持	不详
文件分割	默认分成64MB块	采用RAID0	不支持
网络支持	仅TCP/IP	多种网络，包括 TCP/IP、Infiniband	多种网络，包括TCP/IP、 Infiniband
元数据	元数据服务器管理全量元数据	元数据服务器管理少量元数据	客户端管理全量元数据
商业应用	大量，国内包括中国移动、百度、网易、淘宝、腾讯、华为等	非常不成熟，尚不适合生产环境	测试和使用案例多为欧美，国内用户很少