大数据处理

Hadoop生态

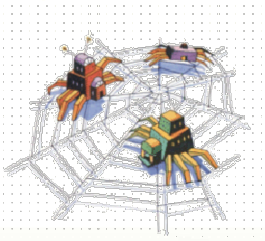
毛波
厦门大学

What is Hadoop?

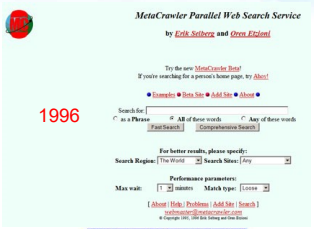
- Apache top level project, open-source implementation of frameworks for reliable, scalable, distributed computing and data storage.
- It is a flexible and highly-available architecture for large scale computation and data processing on a network of commodity hardware.

Brief History of Hadoop

- Designed to answer the question: **"How to process big data with reasonable cost and time?"**



Search engines in 1990s



1996

1996

1997

Google search engines

1998

2013

Hadoop's Developers

Doug Cutting

2005: Doug Cutting and Michael J. Cafarella developed Hadoop to support distribution for the Nutch search engine project.

The project was funded by Yahoo.

2006: Yahoo gave the project to Apache Software Foundation.

Google Origins

2003 — The Google File System
Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Google

→ hadoop HDFS

2004 — MapReduce: Simplified Data Processing on Large Clusters
Jeffrey Dean and Sanjay Ghemawat
jeff@google.com, sanjay@google.com
Google, Inc.

→ hadoop MapReduce

2006 — Bigtable: A Distributed Storage System for Structured Data

→ APACHE HBASE

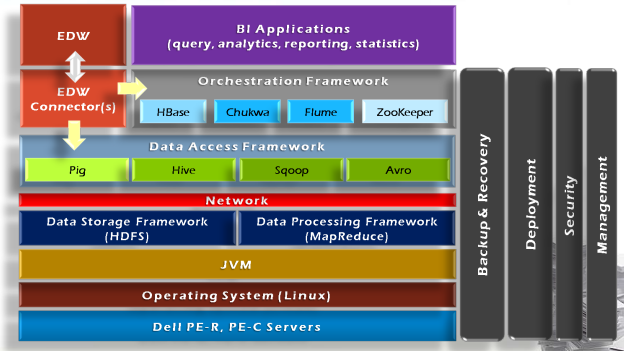
Some Hadoop Milestones

- **2008 - Hadoop Wins Terabyte Sort Benchmark (**sorted 1 terabyte of data in 209 seconds, compared to previous record of 297 seconds)
- 2009 - Avro and Chukwa became new members of Hadoop Framework family
- 2010 - Hadoop's Hbase, Hive and Pig subprojects completed, adding more computational power to Hadoop framework
- **2011 - ZooKeeper Completed**
- **2013 - Hadoop 1.1.2 and Hadoop 2.0.3 alpha.**
  - Ambari, Cassandra, Mahout have been added

What is Hadoop?

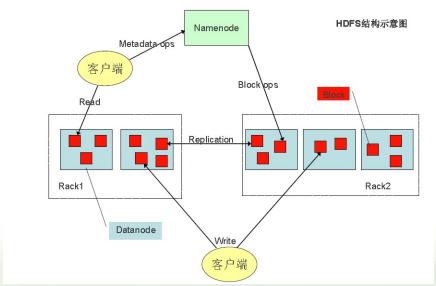
- **Hadoop:**
  - an open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license.
- **Goals / Requirements:**
  - Abstract and facilitate the storage and processing of large and/or rapidly growing data sets
    - Structured and non-structured data
    - Simple programming models
  - High scalability and availability
  - Use commodity (cheap!) hardware with little redundancy
  - Fault-tolerance
  - Move computation rather than data

Hadoop Framework



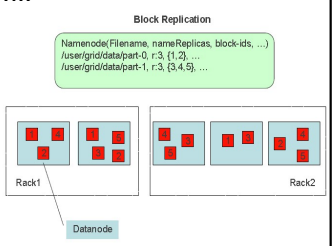
HDFS体系结构

- NameNode→Master
- DataNode→Chunksever



HDFS关键运行机制
--保障可靠性的措施

- 一个名字节点和多个数据节点
- 数据复制（冗余机制）
  --存放的位置（机架感知策略）
- 故障检测
  --数据节点
    心跳包（检测是否宕机）
    块报告（安全模式下检测）
    数据完整性检测（校验和比较）
  --名字节点（日志文件，镜像文件）
- 空间回收机制



HDFS关键运行机制
--写文件流程

- 客户端缓存
- 流水线复制
- 并发写控制
- 流程：
  1. 客户端把数据缓存到本地临时文件夹
  2. 临时文件夹数据超过64M，客户端联系NameNode，NameNode分配DataNode，DataNode依照客户端的位置被排列成一个有着最近物理距离和最小的序列
  3. 与序列的第一个数据服务器建立Socket连接，发送请求头，然后等待回应，依次下传，客户端得到回包，流水线建立成功，
  4. 正式发送数据

HDFS关键运行机制
--读文件流程

- 客户端联系NameNode，得到所有数据块信息，以及数据块对应的所有数据服务器的位置信息
- 尝试从某个数据块对应的一组数据服务器中选出一个，进行连接（选取算法未加入相对位置的考虑）
- 数据被一个包一个包发送回客户端，等到整个数据块的数据都被读取完了，就会断开此链接，尝试连接下一个数据块对应的数据服务器，整个流程，依次如此反复，直到所有想读的都读取完了为止

Hadoop VS. Google

- 技术架构的比较
  - 数据结构化管理组件：Hbase → BigTable
  - 并行计算模型：MapReduce → MapReduce
  - 分布式文件系统：HDFS → GFS
  - 锁管理：ZooKeeper → Chubby

Hadoop VS. Google

- HDFS与GFS比较
  - 子服务器管理模式差异
    - GFS：Chunk Server在Chubby中获取独占锁表示其生存状态，Master通过轮询这些独占锁获知Chunk Server的生存状态
    - HDFS：DataNode通过心跳的方式告知NameNode其生存状态
    - GFS中，Master损坏时，替补服务器可以快速获知Chunk Server的状态
    - HDFS中，NameNode损坏后，NameNode恢复时需要花费一段时间获知DataNode的状态

Hadoop VS. Google

- HDFS与GFS比较
  - HDFS具备空间回收机制
    - 文件删除时，仅删除目录结构
    - 实际数据的删除在等待一段时间后实施
    - 优点：便于恢复文件

Hadoop Related Subprojects

- Pig
  - High-level language for data analysis
- HBase
  - Table storage for semi-structured data
- Hive
  - SQL-like Query language and Metastore
- Mahout
  - Machine learning
- ……