Hadoop Intro

Session 1

Agenda

- Introduction to Hadoop
- What is Big data and Why Hadoop?
- Big Data Characteristics and Challenges
- Comparison between Hadoop and RDBMS
- Hadoop History and Origin
- Hadoop Ecosystem overview
- Anatomy of Hadoop Cluster

Big Data

- Think at Scale Data is in TB even in PB
  - Facebook has 400 terabytes of stored data and ingest 20 terabytes of new data per day. Hosts approx. 10 billion photos, 5PB(2011) and is growing 4TB per day
  - NYSE generates 1TB data/day
  - The Internet Archive stores around 2PB of data and is growing at a rate of 20PB per month
- Flood of data is coming from many resources
  - Social network profile, activity , logging and tracking
  - Public web information
  - Data ware house appliances
  - Internet Archive store etc

Big Data-how it is? What it means ?

- Volume
  - Big Data comes in on large scale. Its on TB and even PB Records, Transaction, Tables , Files
- Velocity
  - Data flown continues, time sensitive, streaming flow Batch, Real time, Streams, Historic
- Variety
  - Big Data extends structured, including semi- structured and unstructured data of all variety: text, log, xml, audio, video, stream, flat files etc. Structured, Semi structured, Unstructured
- Veracity
  - Quality, consistency, reliability and provenance of data Good, bad, undefined, inconsistency, incomplete.

80 - 20 Unstructured vs structured

Use cases

- Social media and websites
- IT – services, Software and Hardware services and support.
- Finance: Better and deeper understanding of risk to avoid credit crisis
- Media: More content that is lined up with your personal preferences
- Life science: Better targeted medicine with fewer complications and side effects
- Retail: A personal experience with product and offer that are just what and you need
- Google, yahoo and others need to index the entire internet and return searched results in milliseconds Business Drivers and sceneries for large data

Challenges in Big Data Storage and Analysis

- Slow to process, can't scale
  - Disk seek for every access
  - Buffered reads, locality -> still seeking every disk page
  - It not Storage Capacity but access speeds which is the bottleneck.
  - Challenges to both store and analyze datasets
  - Scaling is expensive
- Hard Drive capacity to process
  - IDE drive – 75 MB/sec, 10ms seek
  - SATA drive – 300MB/s, 8.5ms seek
  - SSD – 800MB/s, 2 ms "seek"
  - Apart from this analyze, compute, aggregation, processing delay etc.
- Unreliable machines: Risk
  - 1 Machine 1 time in 3 years

Challenges in Big Data Storage and Analysis

- Reliability
  - Partial failure, graceful decline rather than full halt
  - Data recoverability, if a node fails, another picks up its workload
  - Node recoverability, a fixed node can rejoin the group without a full group restart
  - Scalability, adding resources adds load capacity
  - Backup
  - Not affordable, expensive(faster, more reliability more cost)
  - Easy to use and Secure
  - Process data in parallel

An Idea: Parallelism-but not simple

- Parallelism
  - Transfer speed improves at a greater rate than seek speed.
  - Process read/write parallel rather then sequential.
    - 1 drive – 75 MB/sec 16 days for 100TB
    - 1000 drives – 75 GB/sec 22 minutes for 100TB
- A problem: Parallelism is Hard
  - Synchronization
  - Deadlock
  - Limited bandwidth
  - Timing issues and co-ordination
  - Spilt & Aggregation
- Computer are complicate
  - Driver failure
  - Data availability
  - Co-ordination

Distributed Computing

- Yes, We have distributed computing and it also come up with some challenges
  - Resource sharing.
    - Access any data and utilize CPU resource across the system.
  - Portability
  - Reliable
  - Concurrency: Allow concurrent access, update of shared resource, availability with high throughput
  - Scalability: With data, with load
  - Fault tolerance : By having provisions for redundancy and recovery
  - Heterogeneity: Different operating system, different hardware
  - Transparency: Should appear as a whole instead of collection of computers
  - Hide details and complexity by accomplishing above challenges from the user and need a common unified interface to interact with it.

Hadoop

- Apache Hadoop is a framework that allows for the distributed processing of large data sets across clusters of commodity computers using a simple programming model. It is designed to scale up from single servers to thousands of machines, each providing computation and storage.
- Hadoop is an open-source implementation of Google MapReduce, GFS(distributed file system).
- Hadoop was created by Doug Cutting, the creator of Apache Lucene, the widely used text search library.
- Hadoop fulfill need of common infrastructure
  - Efficient, reliable, easy to use
  - Open Source, Apache License
- The name ‘Hadoop’

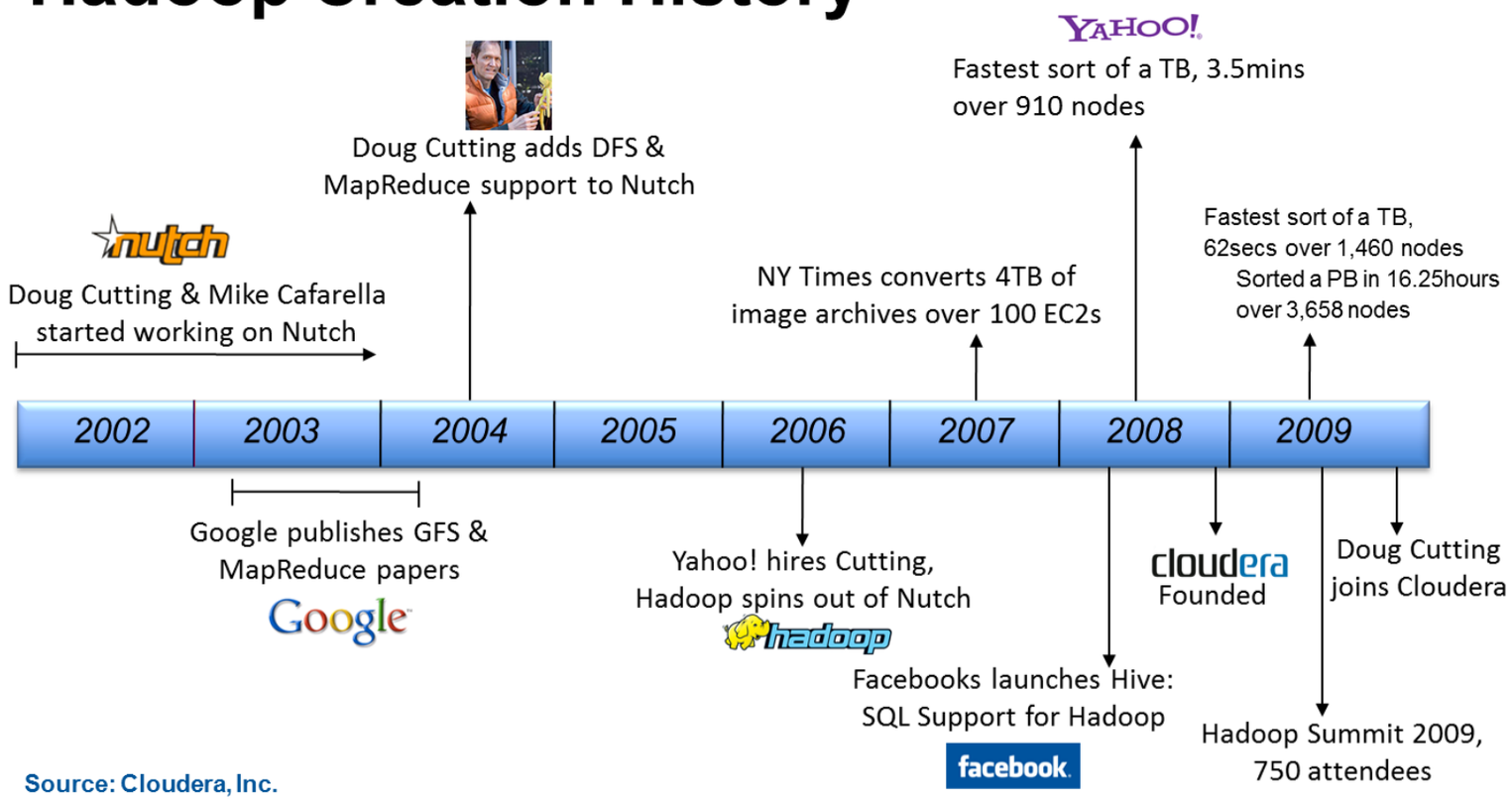
Hadoop Design Axioms

- Store and process large amounts of data (PetaBytes)
- Performance, storage, processing scale linearly
- Compute should move to data
- Simple core , modular and extensible
- Failure is normal, expected
- Manageable and Heal self
- Design run on commodity hardware-cost effective

Solve : Hadoop achieves complete parallelism

- For Storage and Distributed computing (MapReduce)
- Spilt up the data
- Process Data in parallel
- Sort and combine to get the answer
- Schedule, Process and aggregate independently
- Failures are independent, Handle failures.
- Handle fault tolerance

Hadoop History



Source: Cloudera, Inc.

Hadoop Architecture

- Hadoop designed and built on two independent frame works.
- Hadoop = HDFS + Map reduce HDFS (storage and File system) :
  - HDFS is a reliable distributed file system that provides high-throughput access to data
  - MapReduce (processing) : MapReduce is a framework for performing high performance distributed data processing using the divide and aggregate programming paradigm
  - Hadoop has a master/slave architecture for both storage and processing.