The Google File System

&

Big Data

Kenneth M. Brooks III

12/12/2014

Main Idea

- Design system from inexpensive commodity components
  - Fail often
  - Self-monitoring
- Handles Large streaming reads, also smaller random reads
- Scalable file system for large distributed data-intensive applications
- One master, multiple chunkservers to track long-term writes
  - Minimize master's involvement

Implementation

- 3 basic sections of GFS: Single Master, Multiple Chunkservers, Multiple Clients
- Master holds and maintains all metadata
- Clients interact with the master for metadata operations

Analysis

- Incredibly efficient & reliable
  - self-diagnose problems & solve
  - data stored on many chunk servers
- One Master
  - Reduced chance of bottleneck
- Unmatched fault tolerance
- Cheap components
  - possibility of failing is high
-

Comparison: MapReduce & Parallel DBMS

- Similar goals, as well as ability to process data on a large scale
- Parallel DBMSs utilize the relation model made up of rows and columns. MapReduce does not use this model
  - DB Admin can structure data how they please
- Cluster computer is used in each case
- MR doesnt require a schema like a DBMS

Advantages & Disadvantages

Advantages:

- MR ease of setup
  - More freedom for admins & programmers
- Elasticity
- Good Fault Tolerance

Disadvantages:

- MR takes more time to start up
  - Chunkserver & Node “boot/warmup” time
- Wasteful energy use