MAPREDUCE: SIMPLIFIED DATA PROCESSING ON LARGE CLUSTERS

Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters." *OSDI* (2004): n. pag. Web. 17 Oct. 2016.

A COMPARISON OF APPROACHES TO LARGE-SCALE DATA ANALYSIS

Pavlo, Andrew, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, and Michael Stonebreaker. "A Comparison of Approaches to Large-Scale Data Analysis." *SIGMOD* (2009): n. pag. Web. 17 Oct. 2016.

MEGAN MAKINI, OCTOBER 20, 2016

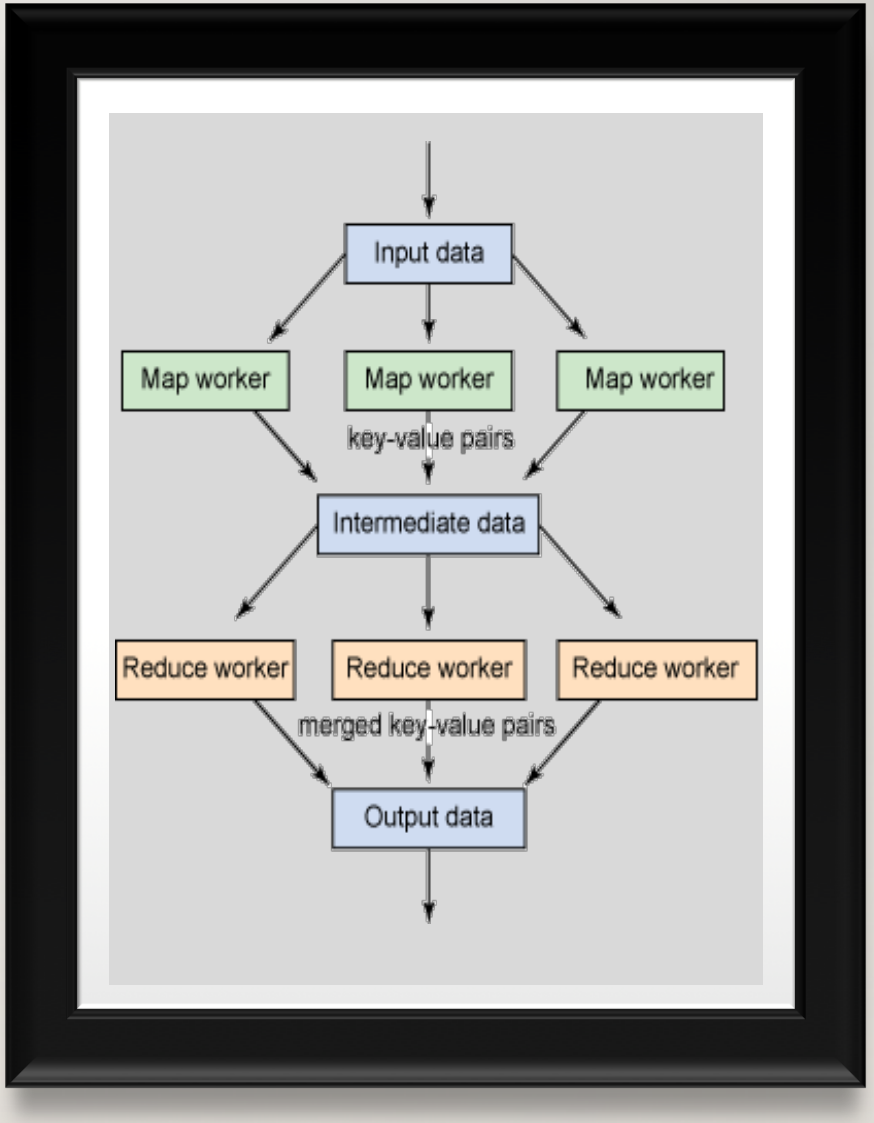
MAPREDUCE: SIMPLIFIED DATA PROCESSING ON LARGE CLUSTERS

- Issue of how to…
  - Parallelize the computation
  - Distribute the data
  - Handle failures
  - The original simple compuation with large amounts of complex code to deal with these issues
- Outcome – MapReduce
  - User-specified map and reduce operations
  - Parallelize large computations
  - Use re-execution for fault tolerance
  - Large-scale indexing

MAPREDUCE: SIMPLIFIED DATA PROCESSING ON LARGE CLUSTERS

Implementation:

1. Map Function: parses key/ value pairs out
2. Intermediate key/value pairs
3. Reduce Function: sorts intermediate keys (groups), passes on unique intermediate keys



ANALYSIS OF MAPREDUCE

- The model is easy to use but requires implementations of restrictions
- A large variety of problems are easily expressible as MapReduce computations
- Can scale to large clusters of machines but network bandwidth is a scarce resource
- Needs redundant executions to reduce impact of slow machines and handle machine failures/ data loss
- Conclusion: A somewhat efficient model, but has limitations.

A COMPARISON OF APPROACHES TO LARGE-SCALE DATA ANALYSIS

MAPREDUCE VS. PARALLEL DBMS

- Data in arbitrary format
- Faster to tune and load the data
- Input data set exists as a collection of one or more partitions in the distributed file system
- MR scheduler & MR central controller

- Data conform to a well-defined schema
- Significantly faster and require less code
- Most tables are partitioned over the nodes in a cluster
- Uses an optimizer that translates SWL commands into a query whose execution is divided amongst nodes

- **Achieve parallelism by dividing any data set to be utilized into partitions**

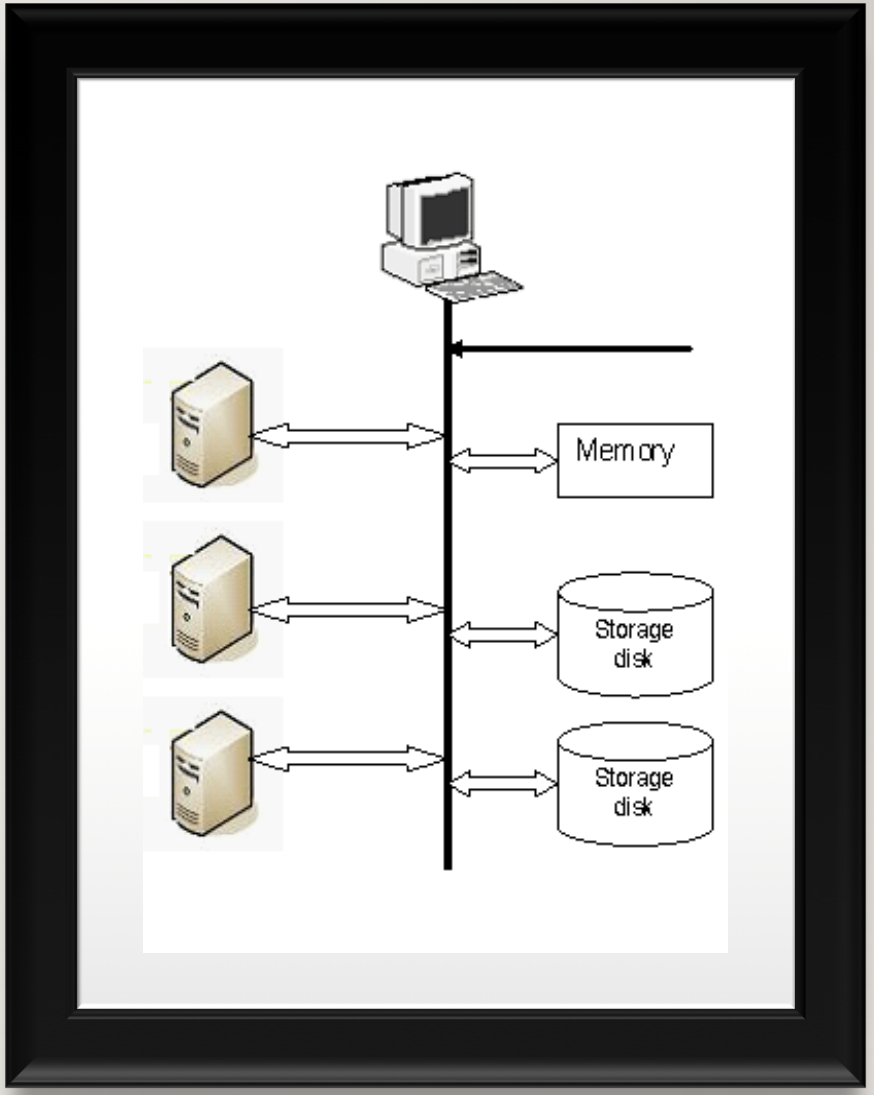
A COMPARISON OF APPROACHES TO LARGE-SCALE DATA ANALYSIS

MapReduce

- Development environments
- Small number of programmers
- Limited application domain

Parallel DBMs

- Integrity of data enforced by default
- B-tree index to accelerate access to data
- Parallel query optimizer



Parallel DBMs

ANALYSIS OF A COMPARISON OF APPROACHES TO LARGE-SCALE DATA ANALYSIS

- MapReduce requires a lot of manual input by the programmer
  - Not recommended for longer-term of larger-sized projects
  - Simplicity of use
  - Minimizes amount of work lost when hardware fails
- Parallel DBMSs advantages
  - B-tree indices to speed the execution of selection operations
  - Novel storage mechanisms
  - Aggressive compression techniques with ability to operate directly on compressed data
  - Sophisticated parallel algorithms for querying large amounts of relational data