

Why MapReduce?

JIANNAN WANG



“Organize the world’s information and make it universally accessible and useful.”

1 PB = 100GB *10,000 Machines

How to store 1PB using 10,000 machines?

GFS (HDFS)

How to process 1PB using 10,000 machines?

MapReduce

Why MapReduce?

1. Fault Tolerant

(Your program will be OK when failures happen)

# of machines	Failure Probability
1	0.1%
10	0.9%
100	9.5%
1000	63.2%
10,000	??? 99.9%

Cost for 5 nodes	Failure Probability
\$100/day	0.1%
\$1/day	10%

Reserved Instance **Spot Instance**

The diagram illustrates the trade-off between system reliability and cost. On the left, a table shows that as the number of machines increases, the failure probability drops significantly, reaching 99.9% at 10,000 machines. On the right, another table shows the cost for 5 nodes and their corresponding failure probabilities. A red arrow points from the 'Failure Probability' column of the first table to the 'Cost for 5 nodes' column of the second table, indicating that lower failure probability corresponds to higher cost. A blue arrow points from the 'Failure Probability' column of the first table to the 'Failure Probability' column of the second table, indicating that the failure probability for 5 nodes is 10 times greater than for individual machines. Labels 'Reserved Instance' and 'Spot Instance' are placed under the respective cost and failure probability entries.

Why MapReduce?

2. Complex Analytics

SQL Machine Learning Graph Processing

SQL

Machine
Learning

Graph
Processing



MapReduce



3. Heterogeneous Storage Systems



HDFS



MySQL®



JSON

APACHE
HBASE

The Great MapReduce Debate

(2008-2010)

<http://tiny.cc/mapreduce-debate>

Mapreduce vs. SQL

Map (k, v)

==

SELECT Map(v)
FROM Table

Reduce (k, v_list)

==

SELECT Reduce(v)
FROM Table
GROUP BY k

MapReduce: A Major Step Backwards



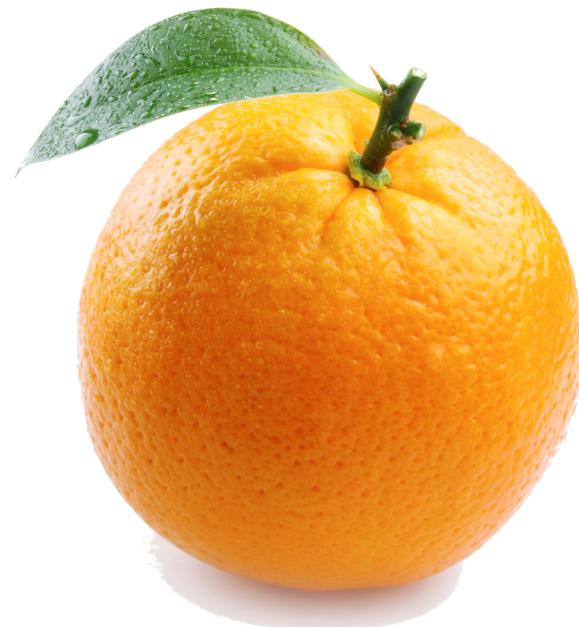
1. MapReduce is a step backwards in database access
2. MapReduce is a poor implementation
3. MapReduce is not novel
4. MapReduce is missing features
5. MapReduce is incompatible with the DBMS tools

Dewitt, D. and Stonebraker, M. *MapReduce: A Major Step Backwards* blogpost; January 17, 2008

Comments From The Other Side



VS.



MapReduce is a program model
rather than a **database** system

From Stonebraker et al.

MapReduce complements DBMSs since databases are not designed for extract-transform-load tasks, a MapReduce specialty.

BY MICHAEL STONEBRAKER, DANIEL ABADI,
DAVID J. DEWITT, SAM MADDEN, ERIK PAULSON,
ANDREW PAVLO, AND ALEXANDER RASIN

MapReduce and Parallel DBMSs: Friends or Foes?



From Dean and Ghemawat

MapReduce advantages over parallel databases include storage-system independence and fine-grain fault tolerance for large jobs.

BY JEFFREY DEAN AND SANJAY GHEMAWAT

MapReduce: A Flexible Data Processing Tool

What They Agree On?

Advantages of MapReduce:

1. Fault Tolerant
2. Complex Analytics
3. Heterogeneous Storage Systems
4. No Data Loading Requirement

Both should Learn from Each Other

Who won the debate?



Nobody is writing
MapReduce code right now.



Many new systems (e.g., Spark,
HIVE) were built on MapReduce