HadoopDB: An architectural hybrid of MapReduce and DBMS technologies

Azza Abouzeid[a], Kamil Bajda-Pawlikowski[a],
Daniel J. Abadi[a], Avi Silberschatz[a], Alex Rasin[b]

[a]Yale University, [b]Brown University

August 27, 2009

Major Trends

1. Data explosion:
   - Automation of business processes, proliferation of digital devices.
   - eBay has a 6.5 petabyte warehouse.

Major Trends

1. Data explosion:
   - Automation of business processes, proliferation of digital devices.
   - eBay has a 6.5 petabyte warehouse.
2. Deep analysis over raw data:
   - Inefficient to push data from database into specialized analysis engines $\longrightarrow$ process data in the database.
   - Best price/performance $\longrightarrow$ data partitioned across 100-1000s of cheap, commodity, shared-nothing machines.
   - Clouds of processing nodes on demand, pay for what you use

Major Trends

1. Data explosion:
   - Automation of business processes, proliferation of digital devices.
   - eBay has a 6.5 petabyte warehouse.
2. Deep analysis over raw data:
   - Inefficient to push data from database into specialized analysis engines $\longrightarrow$ process data in the database.
   - Best price/performance $\longrightarrow$ data partitioned across 100-1000s of cheap, commodity, shared-nothing machines.
   - Clouds of processing nodes on demand, pay for what you use

Bottom line

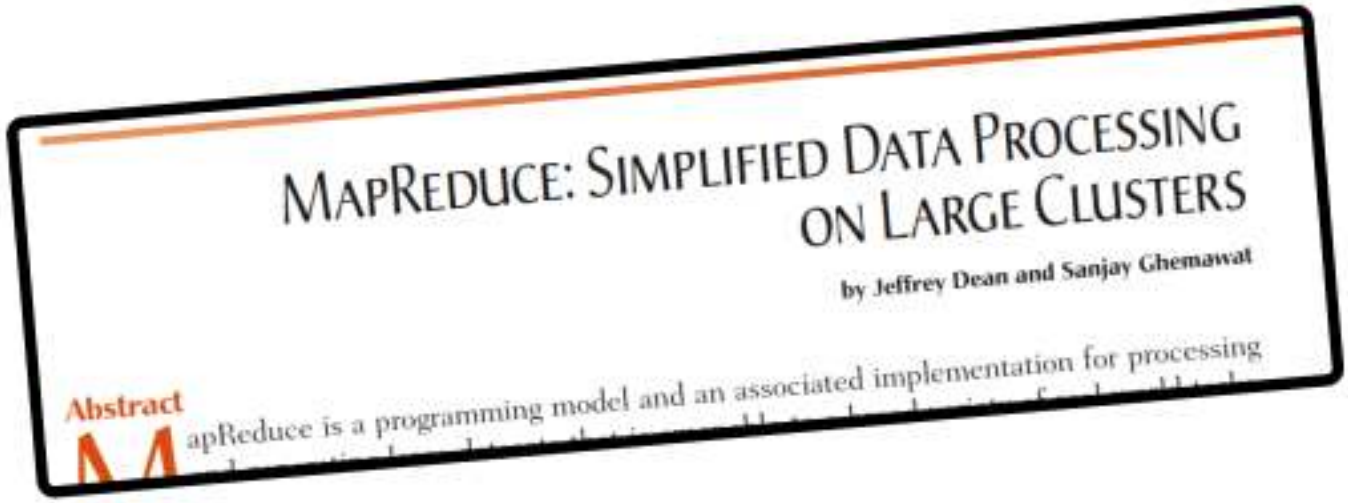
Processing massive structured data on 1000s of shared-nothing nodes.

The Candidates

	Scalability*	High Performance**
MapReduce	✓	✗
Parallel Databases	✗	✓
What we need	✓	✓

** 1000s of nodes*
*** Queries on structured data*

A bit of history ...



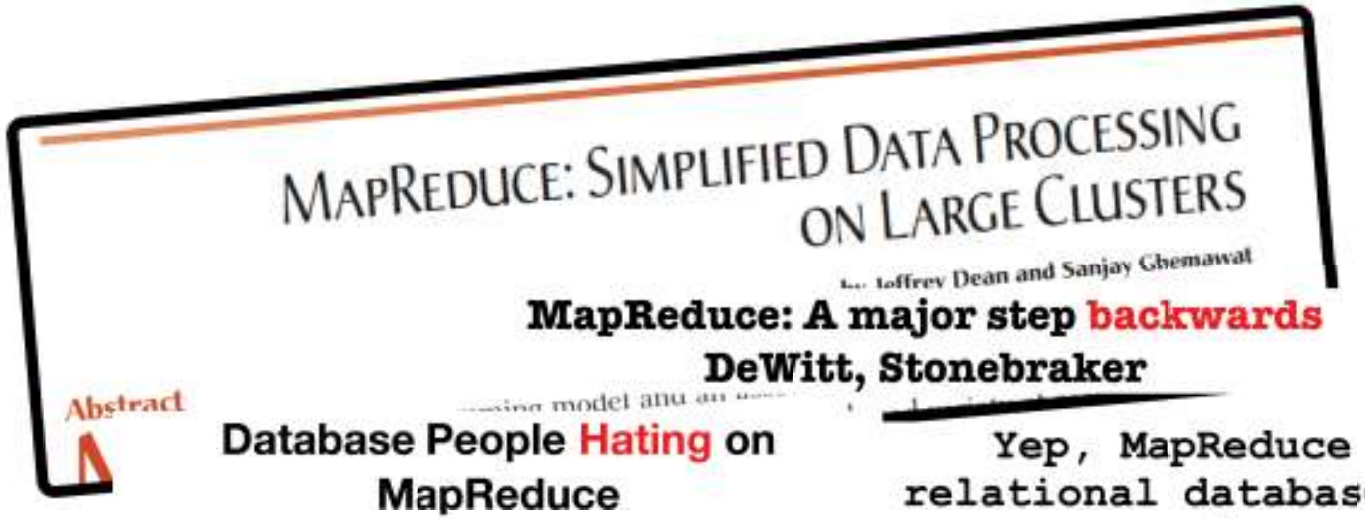
MapReduce: Simplified Data Processing on Large Clusters

by Jeffrey Dean and Sanjay Ghemawat

Abstract

MapReduce is a programming model and an associated implementation for processing

A bit of history ...



MapReduce: A major step backwards
DeWitt, Stonebraker

Database People Hating on MapReduce
Hoff

Yep, MapReduce isn't a relational database, So what?
Homan

DATABASES ARE HAMMERS; MAPREDUCE IS A SCREWDRIVER
CHU-CAROLL

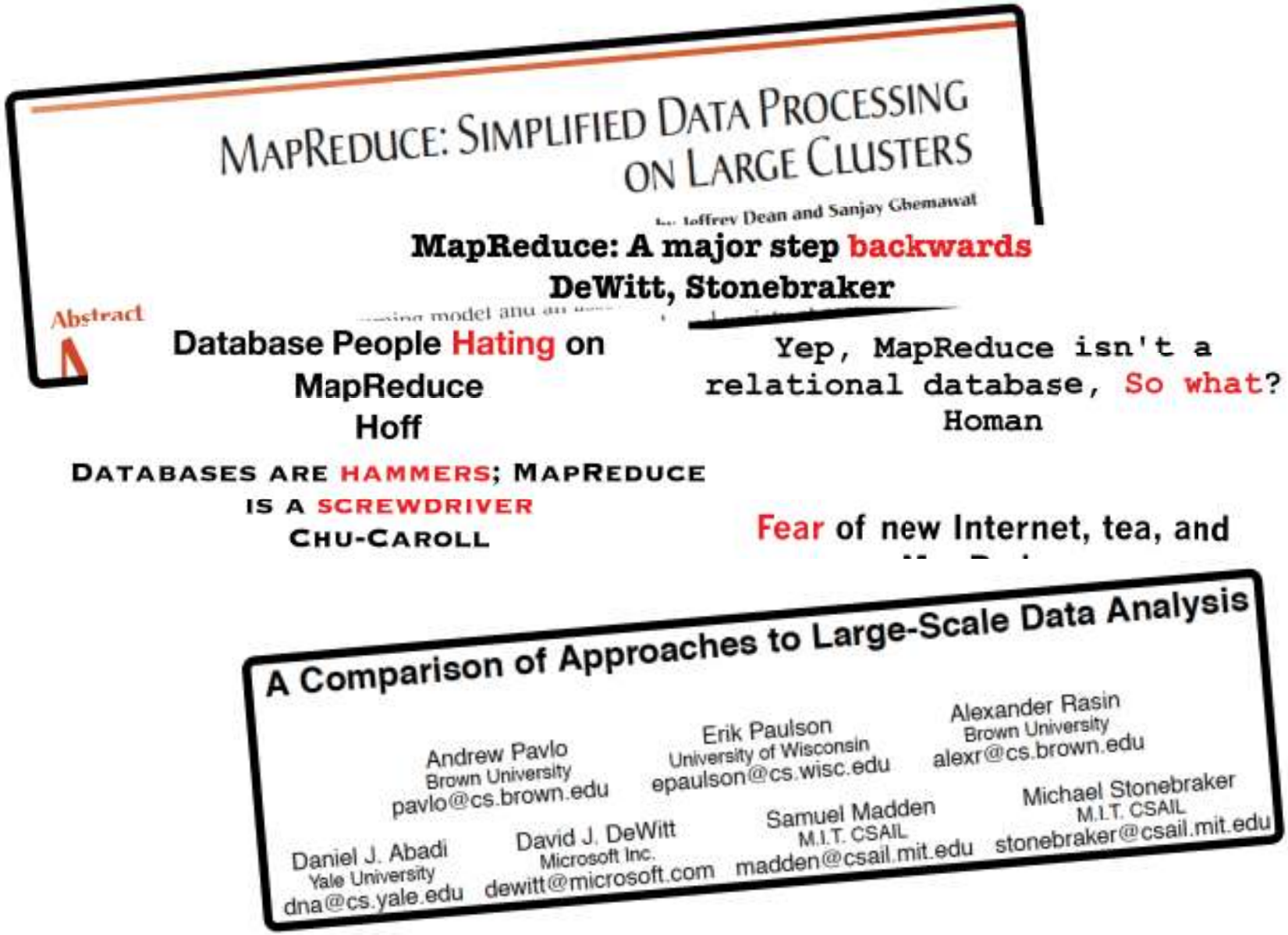
Fear of new Internet, tea, and MapReduce
Marks

Relational Database Experts Jump The MapReduce Shark
Jorgensen

Winning in the Cloud: We lose!
Hellerestein

Endless Blogging!

A bit of history ...



At Yale, we looked deeper ...

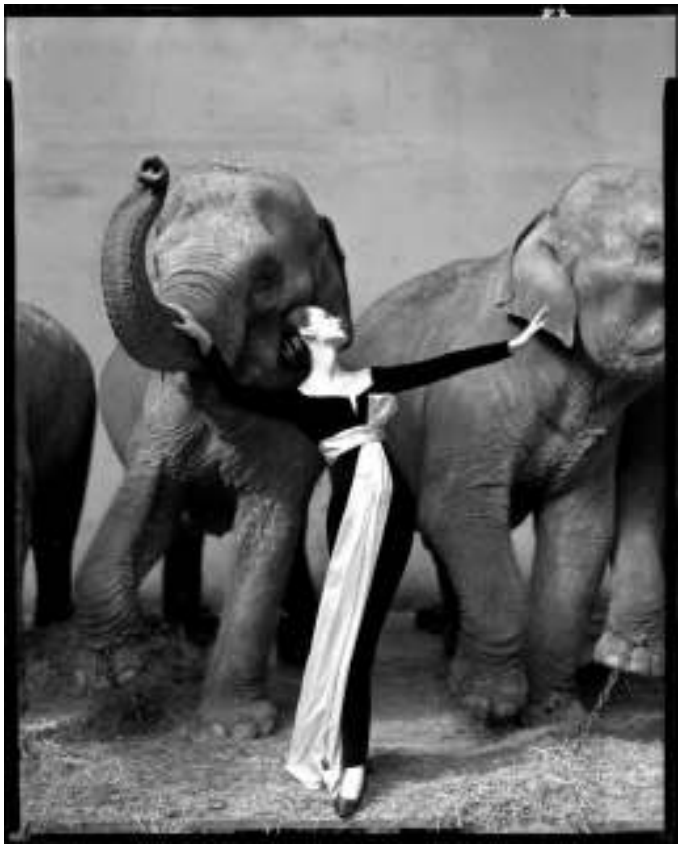
At Yale, we looked deeper ...

Parallel Databases

1. Great performance with queries on structured data

But ...

1. "It's okay to lose work!"
   - fault: restarts the query
     - Google reports 1.2 failures/job
   - performance fluctuations: wait for slowest node
2. No open-source parallel database!! Commercial ones are expensive $$$



"Postgres is a high-maintenance, perfectionist, fussy, city girl"

http://briansmithphoto.files.wordpress.com/2009/05/avedon-dovima-with-elephants1.jpg

MapReduce

1. Great scalability
   - Jobs broken into more granular independent tasks
   - Run-time scheduling
   - Yahoo! runs 4000+ node clusters with Hadoop
2. Free and open-source

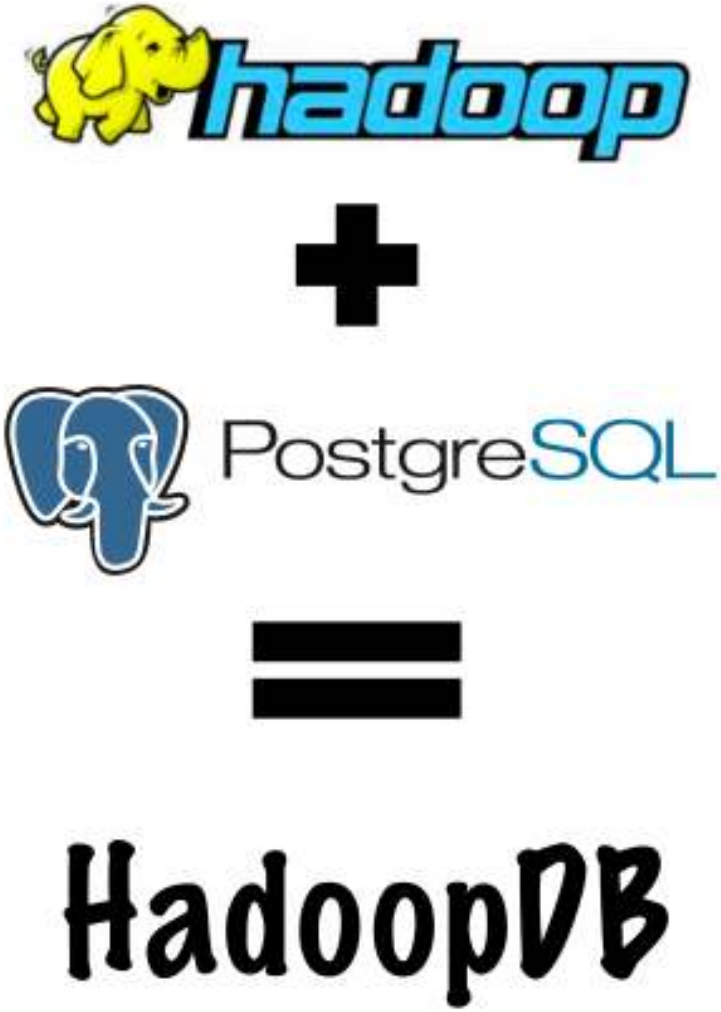
But ...

1. Poor performance with queries on structured data
   - Ignores schema
   - Brute-force model

"Hadoop is a slow, lazy, brute, farm boy"

http://www.breedbay.co.uk/gallery//data/500/elephant-chmai-basketball.jpg

Until we discovered ...



... that they complete each other

http://i214.photobucket.com/albums/cc19/brittanybutton/elephants.jpg

HadoopDB's Design

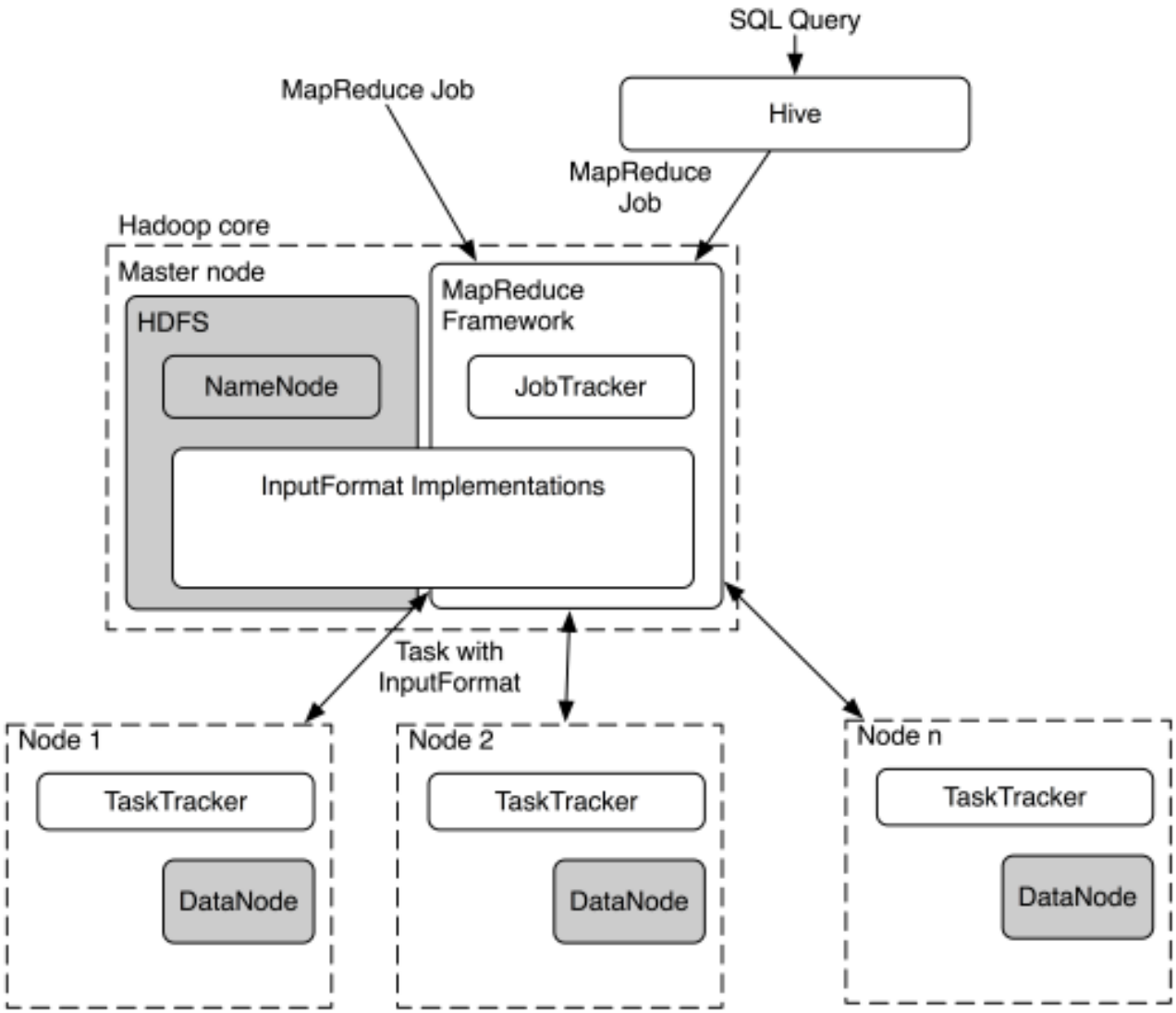
Goals:

- Performance
- Flexible query interface
- Fault-tolerance
- Tolerance for fluctuations from expected performance
- Scalability

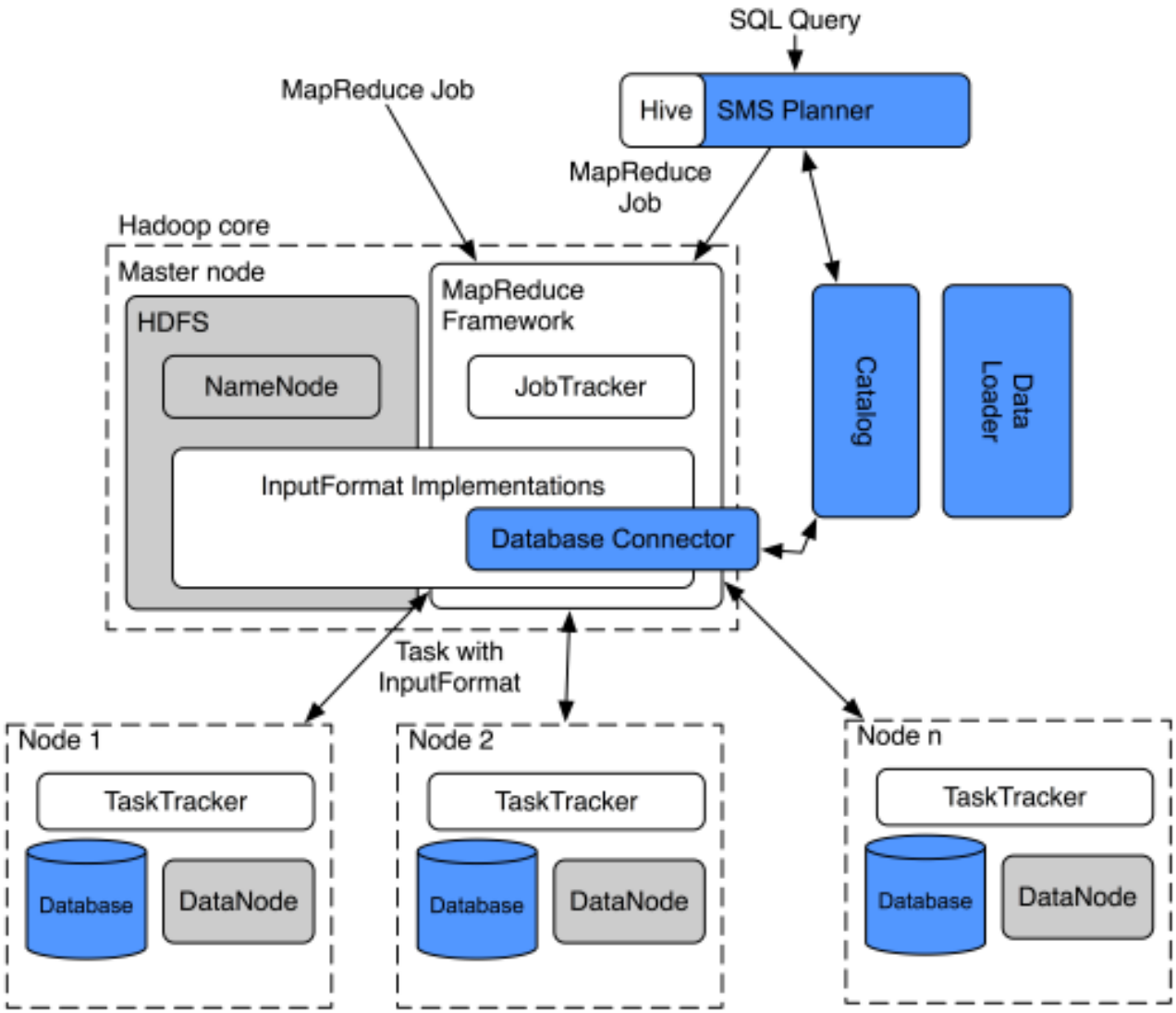
Basic design idea

Multiple, independent, single-node databases coordinated by Hadoop.

Hadoop Basics



Architecture

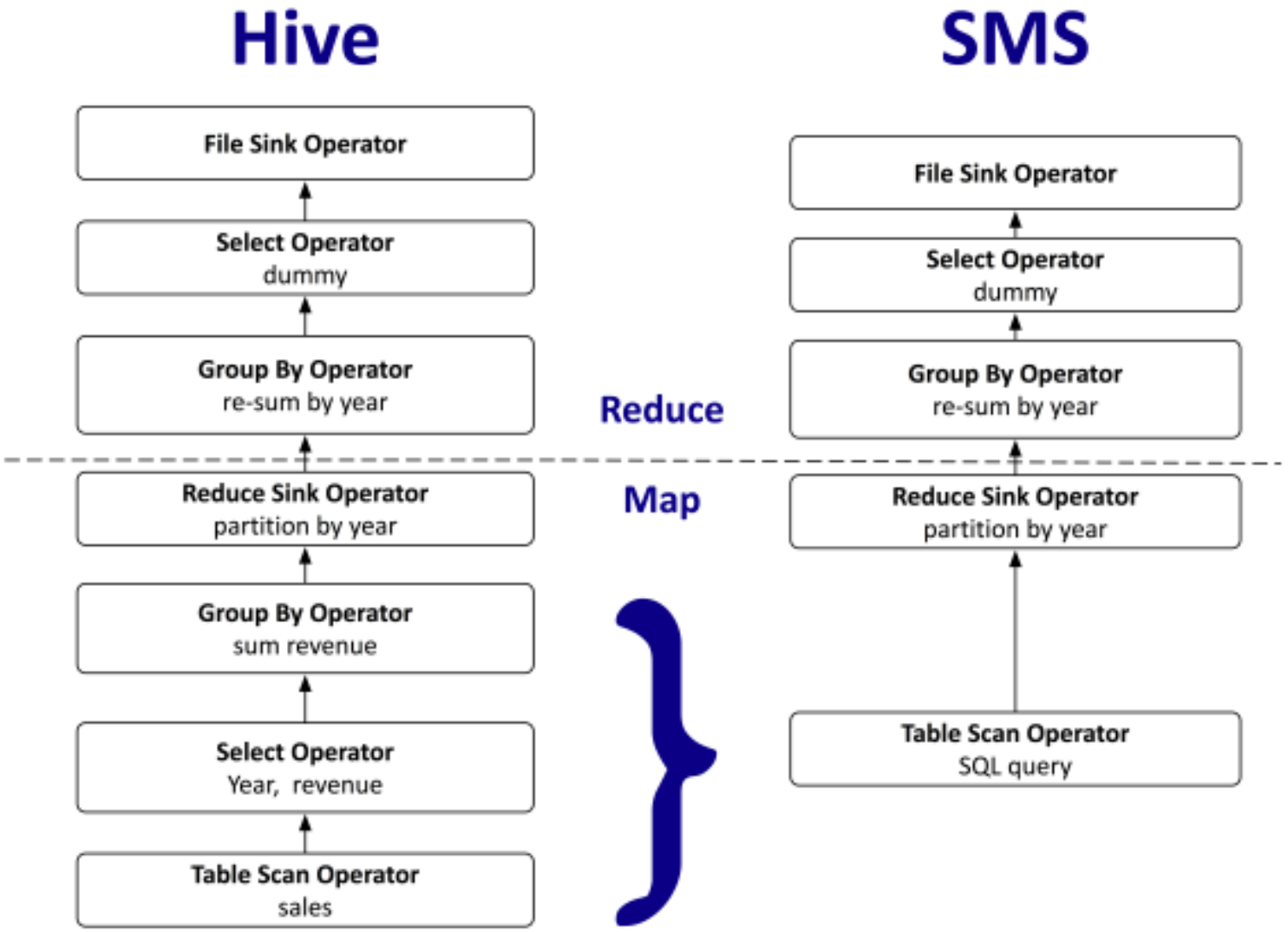


SQL-MR-SQL (SMS): Hive Basics

Hive Converts SQL queries into MapReduce jobs over HDFS files

1. Derives schema of files from an internal catalog
2. Parses, plans, optimizes the SQL query into a *relational* operator DAG
3. Breaks down plan into series of Map / Reduce task with interleaving re-partition operators

SQL-MR-SQL



SELECT YEAR(saleDate), SUM(revenue) FROM sales GROUP BY YEAR(saleDate);

Evaluating HadoopDB

Compare HadoopDB to Hadoop and Parallel databases:

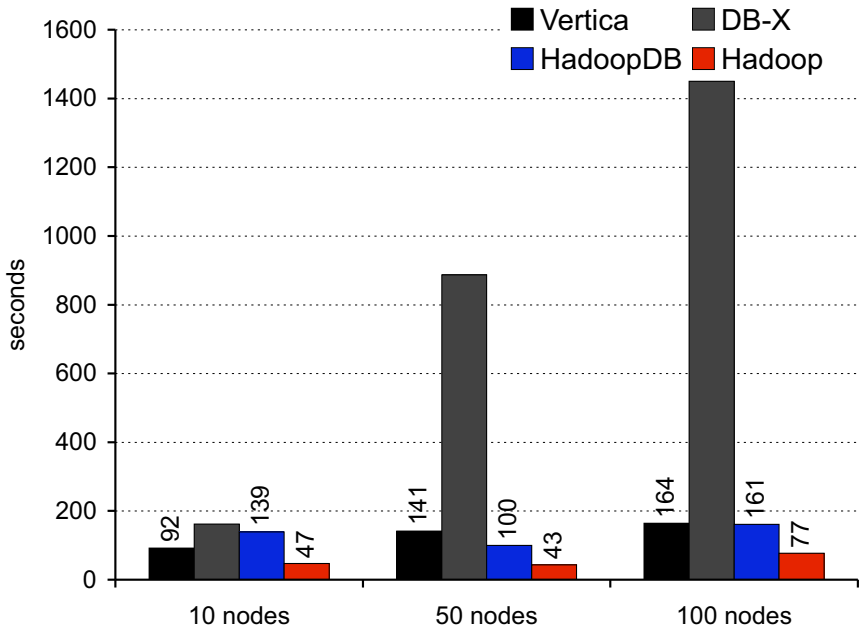
1. Performance:
   - *We expected HadoopDB to approach the performance of parallel databases*
   - Load times vs. performance trade-offs
2. Scalability:
   - *We expected HadoopDB to scale as well as Hadoop*
   - Fault- and fluctuation- tolerance

Experimental Setup

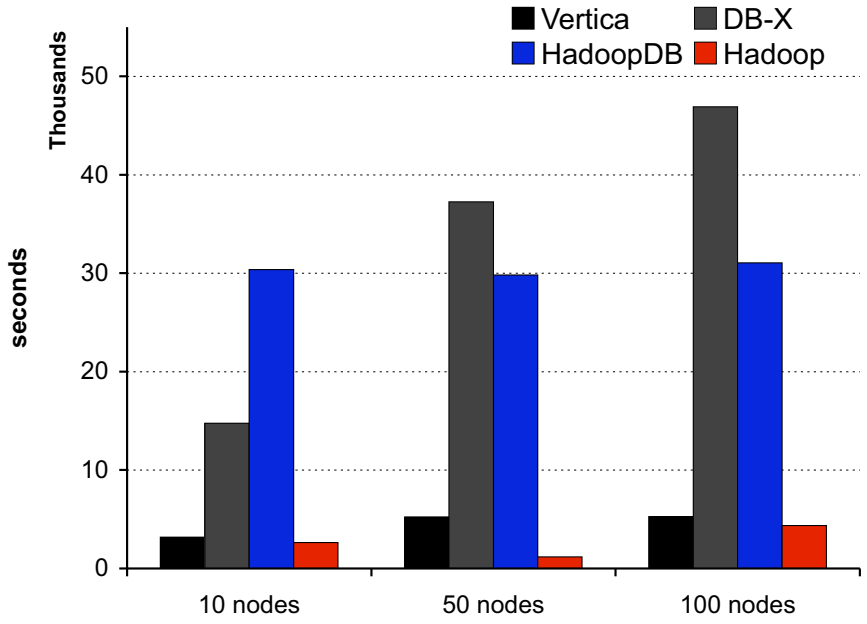
1. Stage
   - Amazon EC2 cloud, clusters of 10, 50, 100 machines
2. Characters
   - Hadoop
   - HadoopDB
   - Vertica
   - DB-X*
3. Plot
   - Pavlo et al. SIGMOD benchmark of large-scale analytical queries derived from processing web-data
   - 20+ GB/node

*DB-X results reproduced from Pavlo et al. 2009

Load

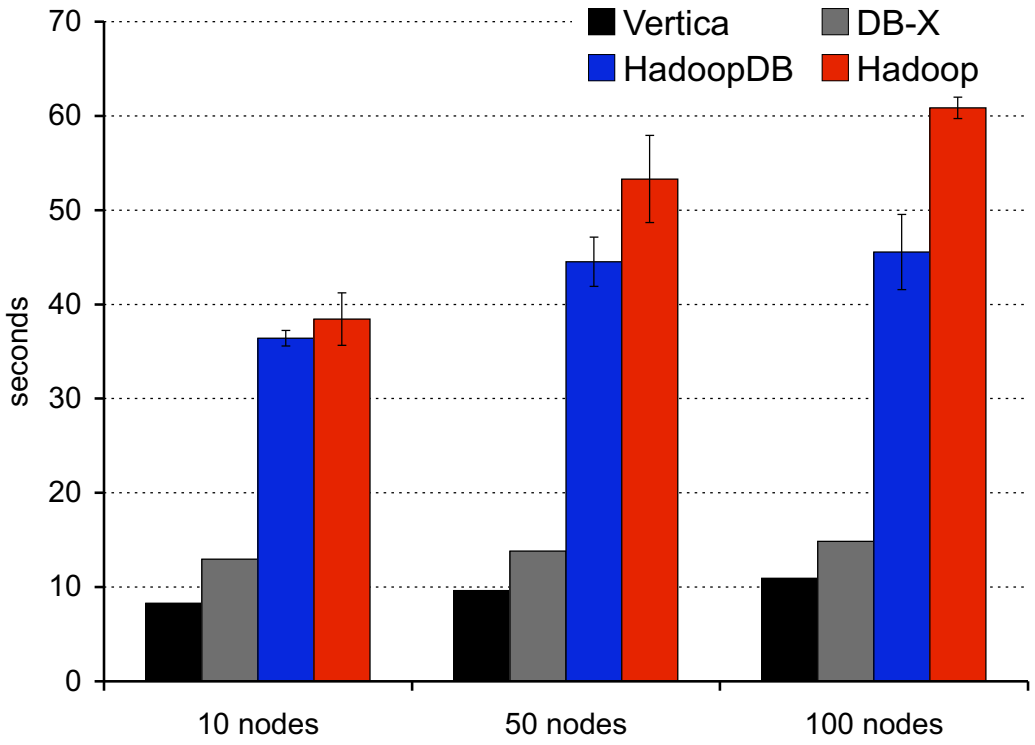


Grep Data (535MB/node):
No pre-processing, data randomly generated



User Visits Log (20GB/node):
Partitioning, chunking (1GB chunks), sorting and indexing

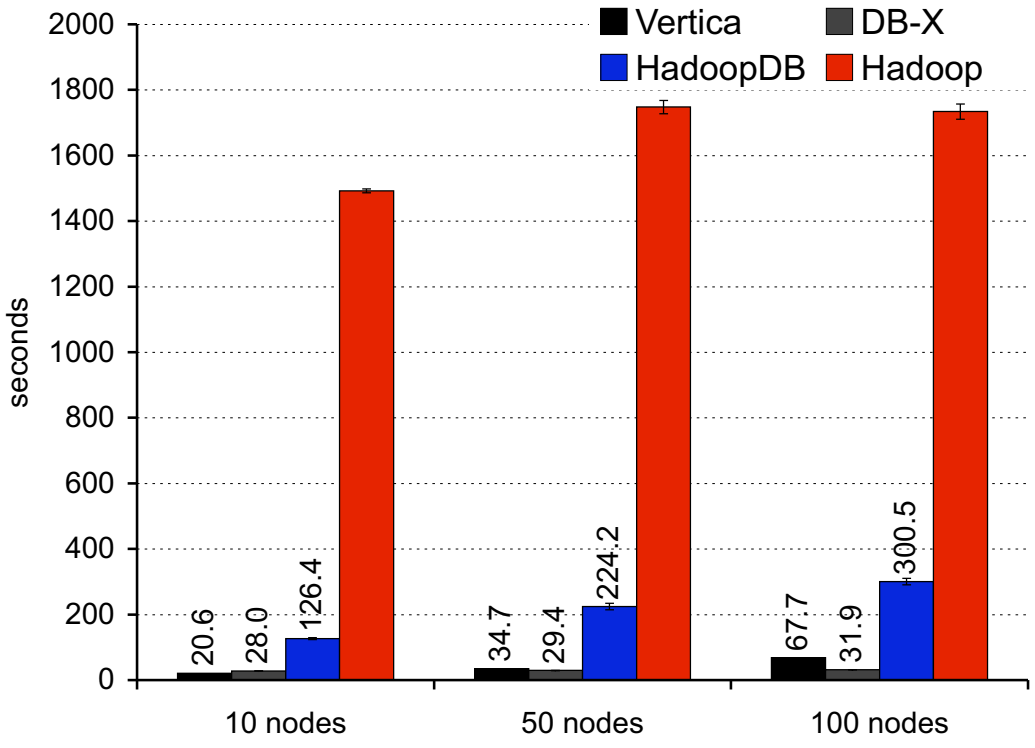
Performance: Grep Task



SELECT * FROM grep WHERE field LIKE '%xyz%';

1. Full table scan, highly selective filter
2. Random data, no room for indexing
3. Hadoop overhead outweighs query processing time in single-node databases

Performance: Join Task




```
SELECT sourceIP, AVG(pageRank), SUM(adRevenue)
FROM rankings, uservisits
WHERE pageURL=destURL
AND visitDate BETWEEN 2000-1-15 AND 2000-1-22
GROUP BY sourceIP
ORDER BY SUM(adRevenue) DESC LIMIT 1;
```

1. No full table scan due to clustered indexing
2. Hash partitioning and efficient join algorithm
3. Partial aggregation pushed into DB layer

Performance: Bottom Line

1. Unstructured data
   - HadoopDB's performance matches Hadoop
2. Structured data
   - HadoopDB's performance is close to parallel databases

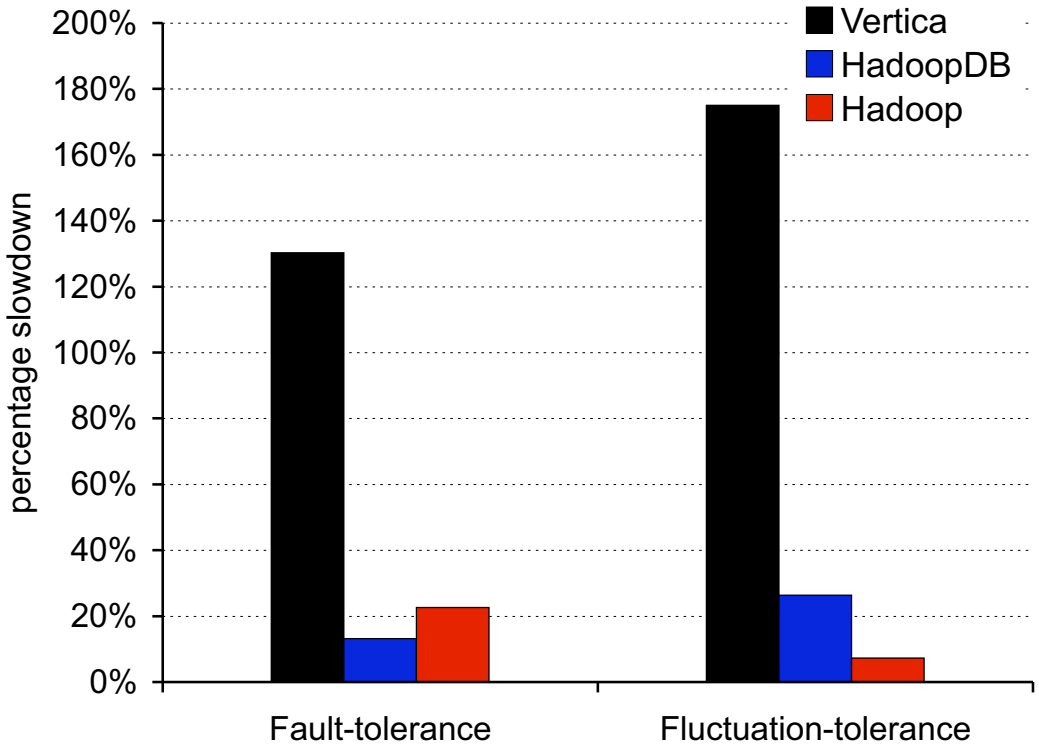
Scalability: Setup

1. Simple aggregation task - full table scan
2. Data replicated across 10 nodes
3. Fault-tolerance: Kill a node halfway
4. Fluctuation-tolerance: Slow down a node for the entire experiment

Key differences

- HadoopDB and Hadoop take advantage of runtime scheduling by splitting data into chunks or blocks
- Parallel databases restart wait for the slowest node

Scalability: Results



1. Run-time scheduling
   - Block-level vs. Query-level restart
2. Frequent checkpointing vs. pipelining results

To summarize

HadoopDB ...

1. is a hybrid of DBMS and MapReduce
2. scales better than commercial parallel databases
3. is as fault-tolerant as Hadoop
4. approaches the performance of parallel databases
5. is free and open-source

`http://hadoopdb.sourceforge.net`

Future work

Engineering work:

1. Full SQL support in SMS
2. Data compression
3. Integration with other open source databases
4. Full automation of the loading and replication process
5. Out-of-the box deployment
6. We're hiring!

Research Work:

- Incremental loading and on-the-fly repartitioning
- Dynamically adjusting fault-tolerance levels based on failure rate

Thank You ...

We welcome all thoughts on how to raise HadoopDB ...

http://www.jpbutler.com/thailand/images/elephant-8-days-old.jpg

Backup: Teradata Fault-tolerance

What happens if a processing node fails?

"Teradata's parallel architecture ensures that part of every request is executing on every node. When a processing node fails, all work is affected. This is an area where Teradata is fault resilient rather than fault tolerant. Our nodes are achieving such a large mean time between failures (MTBF) today that this is a rare occurrence.

If a node fails, the rest of the system is immediately notified and cycles through a recovery process. All requests are halted and rolled back. ... moves units of parallelism from the failed node to an operational one. ..."

http://www.teradata.com/td/go.aspx/?id=115417&logout_127166=1