HadoopDB: An open source hybrid of MapReduce and DBMS technologies

Azza Abouzeid, Kamil Bajda-Pawlikowski
Daniel J. Abadi, Avi Silberschatz

Yale University
http://hadoopdb.sourceforge.net

October 2, 2009

HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. Azza Abouzeid, Kamil Bajda-Pawlikowski, Daniel J. Abadi, Avi Silberschatz, Alex Rasin. In Proceedings of VLDB, 2009.

Major Trends

1. Data explosion:
   - Automation of business processes, proliferation of digital devices.
   - eBay has a 6.5 PB warehouse, Yahoo! Everest has 10 PB.
2. Analysis over raw data

Major Trends

1. Data explosion:
   - Automation of business processes, proliferation of digital devices.
   - eBay has a 6.5 PB warehouse, Yahoo! Everest has 10 PB.
2. Analysis over raw data

Bottom line

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

Sales Record Example

Consider a large data set of sales log records, each consisting of sales information including:

1. a date of sale
2. a price

We would like to take the log records and generate a report showing the total sales for each year.

Question:

How do we generate this report efficiently and cheaply over massive data contained in a shared-nothing cluster of 1000s of machines?

MapReduce (Hadoop)

MapReduce is a programming model which specifies:

- A map function that processes a key/value pair to generate a set of intermediate key/value pairs,
- A reduce function that merges all intermediate values associated with the same intermediate key.

Hadoop

- is a MapReduce implementation for processing large data sets over 1000s of nodes.
- Maps (and Reduces) run independently of each other over blocks of data distributed across a cluster.

Sales Record Example using Hadoop

Query: Calculate total sales for each year.

We write a MapReduce program:

- Map: Takes log records and extracts a key-value pair of year and sale price in dollars. Outputs the key-value pairs.
- *Shuffle: Hadoop automatically partitions the key-value pairs by year to the nodes executing the Reduce function*
- Reduce: Simply sums up all the dollar values for a year.

Relational Databases

Suppose that the data is stored in a relational database system, the sales record example could be expressed in SQL as:

```
SELECT YEAR(date) AS year, SUM(price)
FROM sales
GROUP BY year
```

The execution plan is:

$\text{projection}_{(\text{year},\text{price})} \longrightarrow \text{hash aggregation}_{(\text{year},\text{price})}$.

Question:

How do we process this efficiently if the data is very large?

Parallel Databases

Parallel Databases are like single-node databases except:

- Data is partitioned across nodes
- Individual relational operations can be executed in parallel

SELECT YEAR(date) AS year, SUM(price)
FROM sales GROUP BY year

Execution plan for the query:
$\text{projection}_{(\text{year,price})} \rightarrow$ partial hash aggregation$_{(\text{year,price})} \rightarrow$ partitioning$_{(\text{year})} \rightarrow$ final aggregation$_{(\text{year,price})}$.

Note that the execution plan resembles the map and reduce phases of Hadoop.

Differences between Parallel Databases and Hadoop

	Parallel Database	MapReduce
Data	Designed for structured, relational data	Designed for unstructured data
Query Interface	SQL	MapReduce programs written in a variety of languages *(some SQL support)*
Query Execution	Pipelines results between operators	Materializes results between Map and Reduce phases
Job Granularity	Entire query	Determined by data storage block size *(Runtime scheduler)*

Differences between Parallel Databases and Hadoop

	Parallel Database	MapReduce
Data	Designed for structured, relational schema	Designed for unstructured data
Query Interface	SQL	MapReduce programs written in a variety of languages. *(some SQL support)*
Query Execution	Pipelines results between operators	Materializes results between Map and Reduce phases
Job Granularity	Entire query	Determined by data storage block size *(Runtime scheduler)*

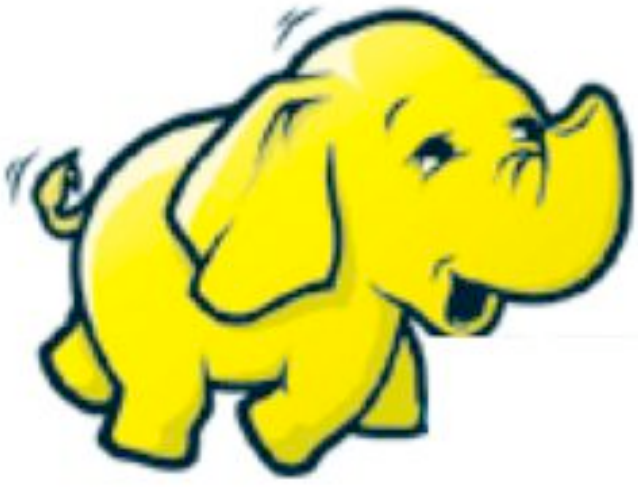
To summarize

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

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

At Yale, we looked beyond the differences ...

At Yale, we looked beyond the differences ...



and we discovered ...

... that they complete each other

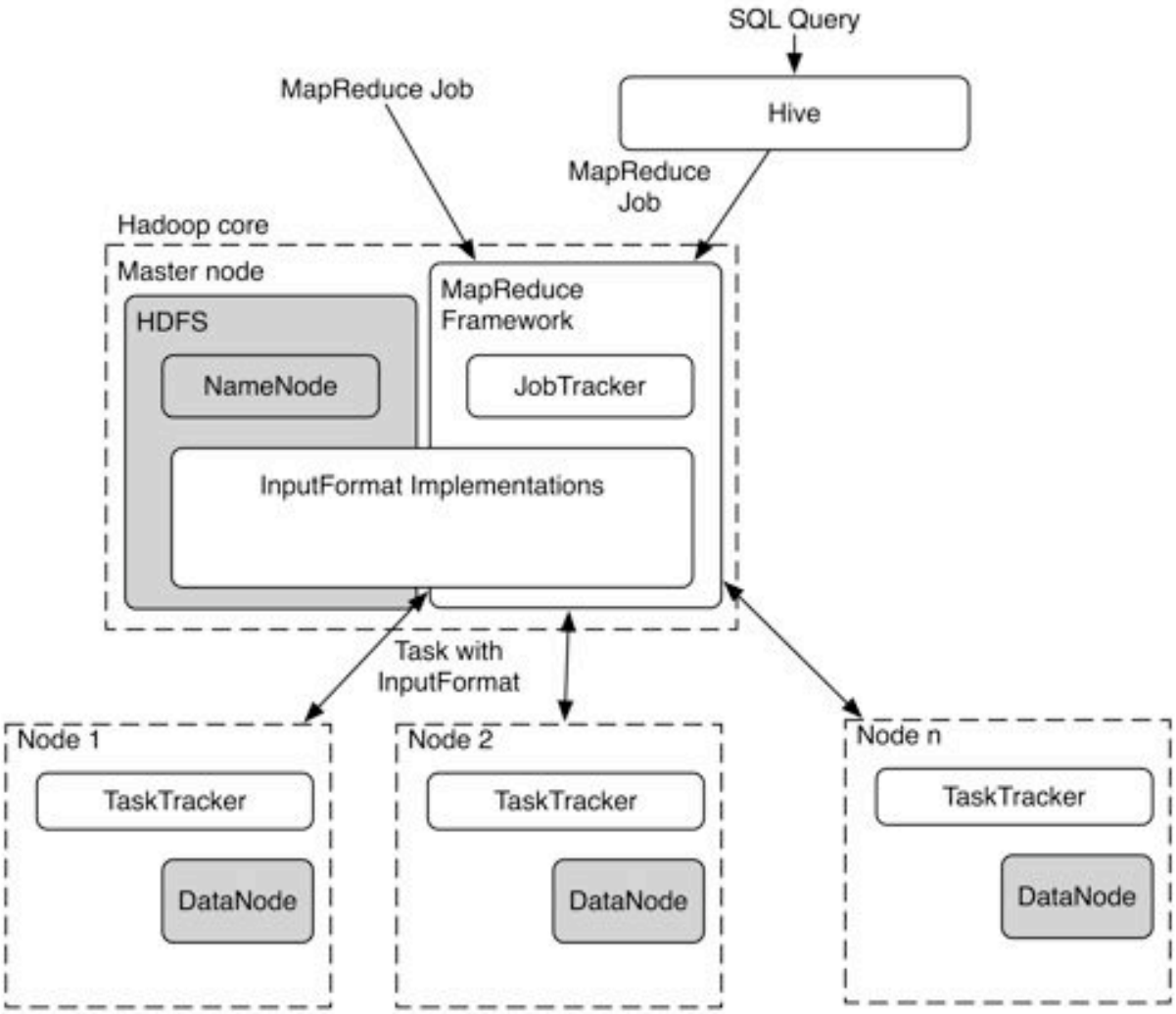
hadoop + PostgreSQL = HadoopDB

Basic design idea

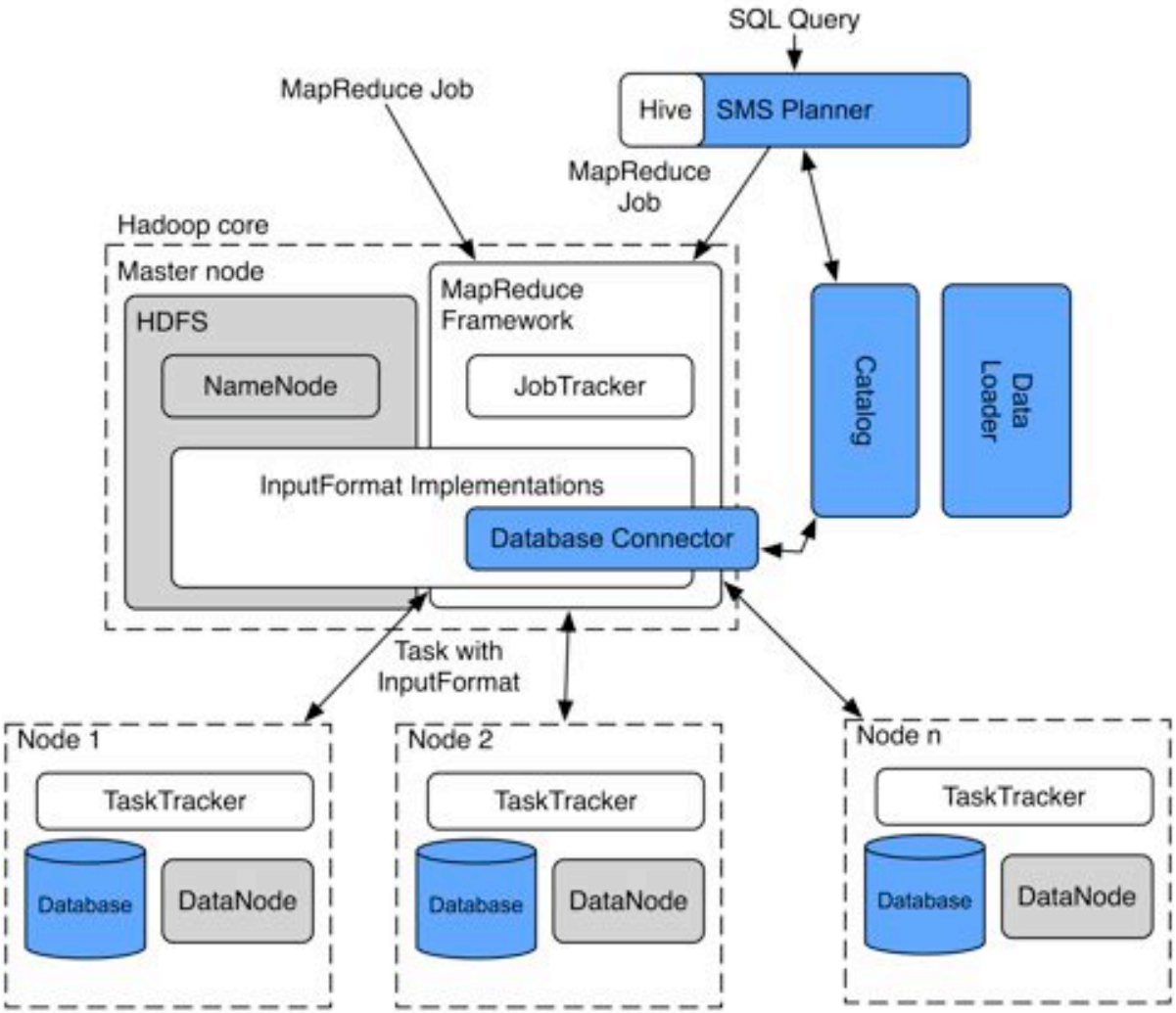
Multiple, independent, single node databases coordinated by Hadoop.

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

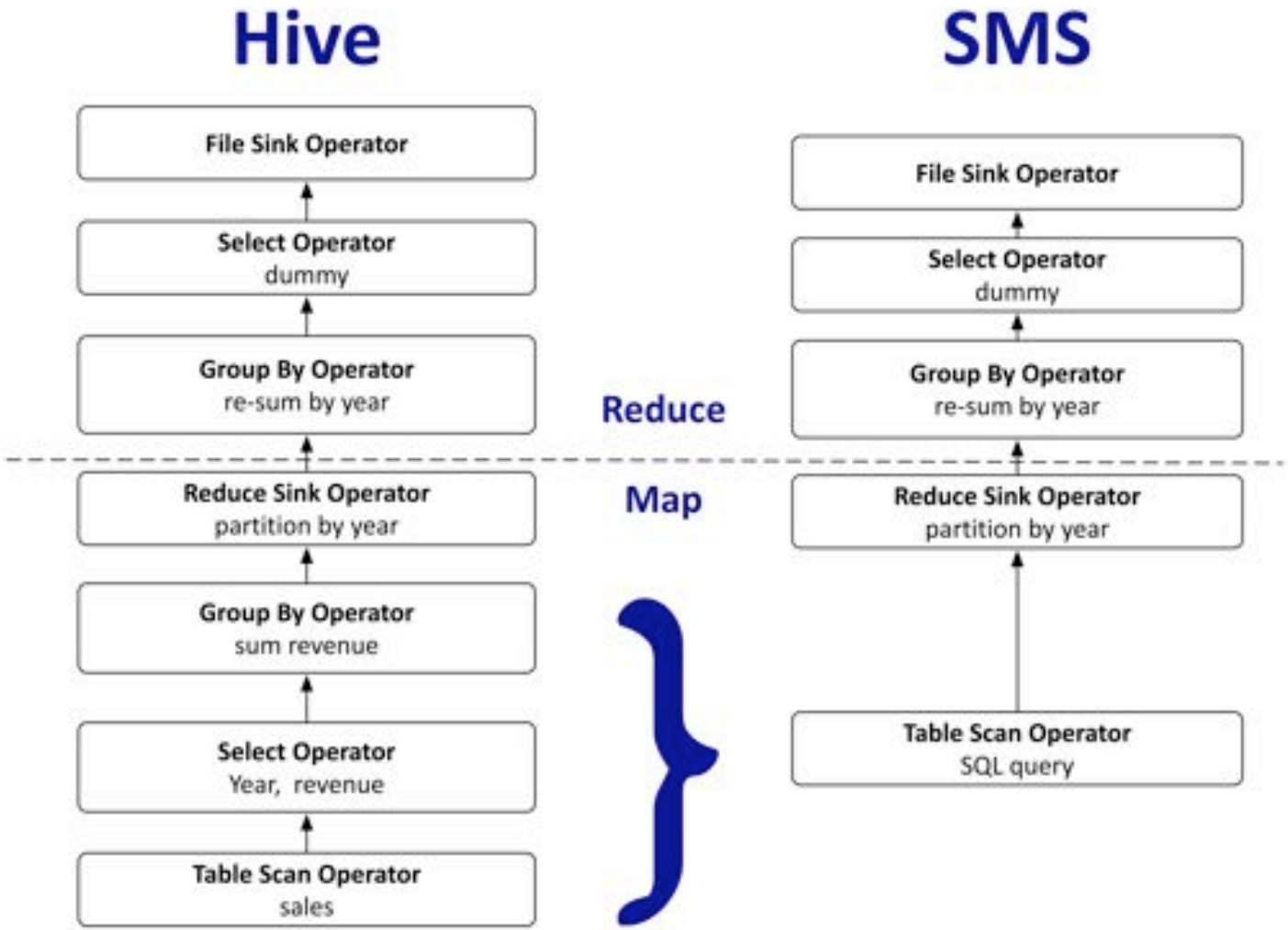
Hadoop Basics



Architecture



SQL-MR-SQL



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

Evaluating HadoopDB

Compare HadoopDB to

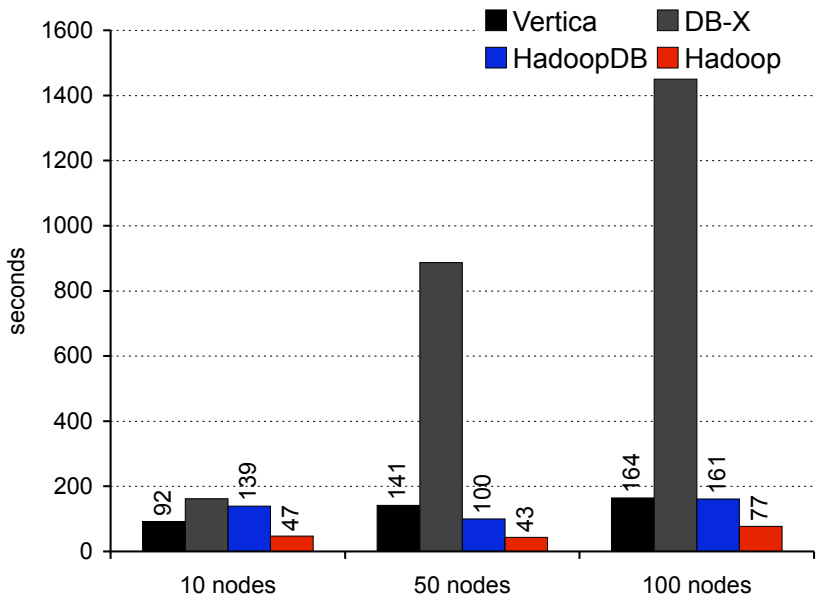
1. Hadoop
2. Parallel databases (Vertica, DBMS-X)

Features:

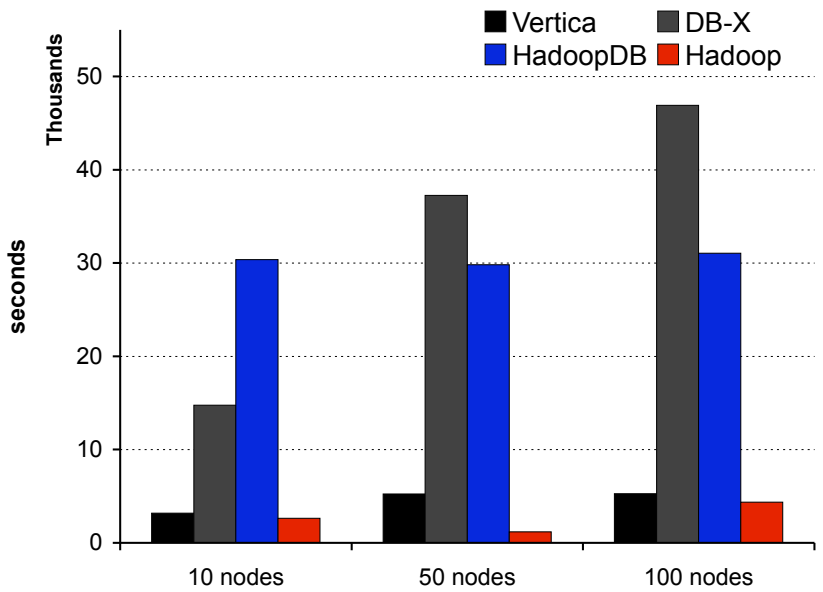
1. Performance:
   - *We expected HadoopDB to approach the performance of parallel databases*
2. Scalability:
   - *We expected HadoopDB to scale as well as Hadoop*

We ran the Pavlo et al. SIGMOD'09 benchmark on Amazon EC2 clusters of 10, 50, 100 nodes.

Load

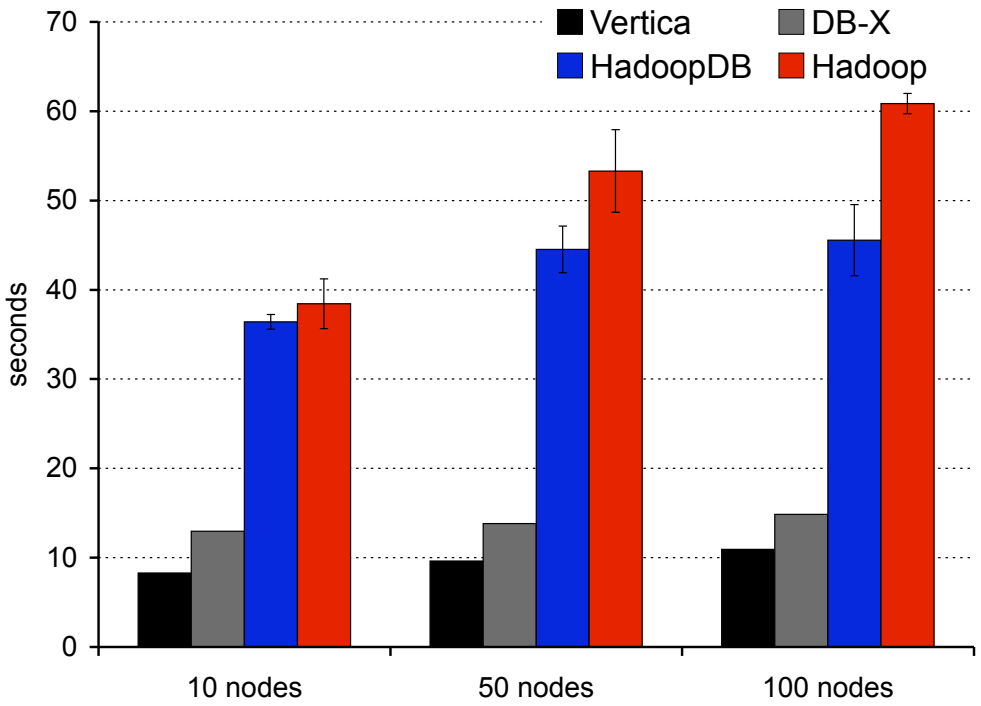


Random Unstructured Data (535MB/node)



Structured data (20GB/node)

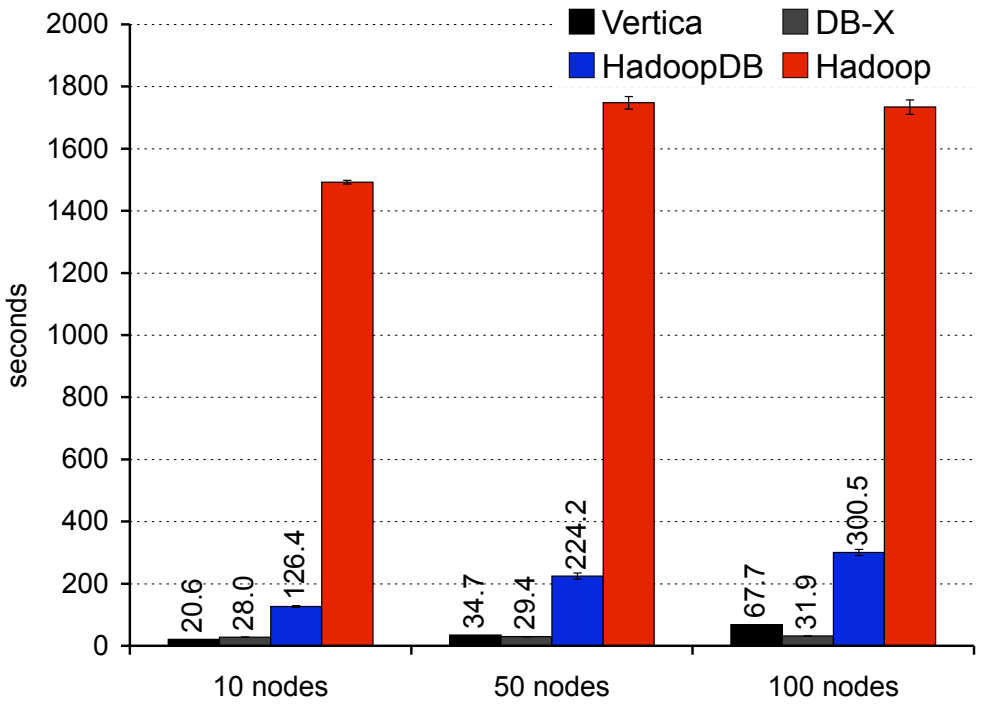
Performance: Grep Task



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

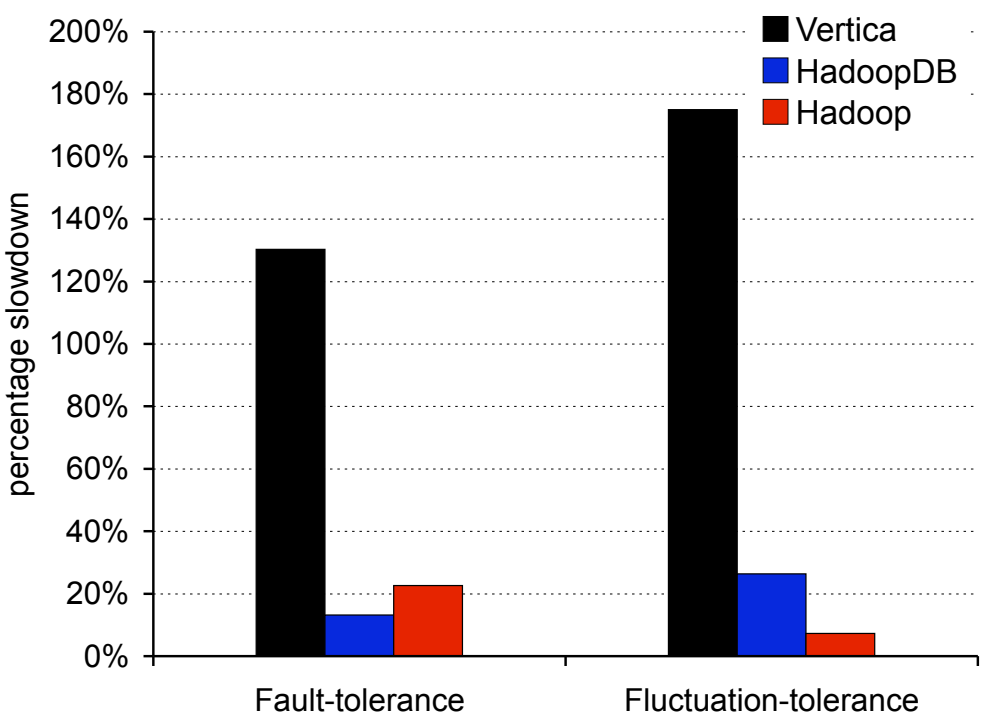
Performance: Bottom Line

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

Scalability: Results



1. HadoopDB and Hadoop take advantage of runtime scheduling by splitting data into chunks or blocks
2. Parallel databases restart entire query on node failure or wait for the slowest node

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