UNIVERSITY of WASHINGTON

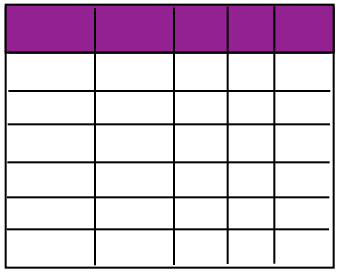
Mapreduce vs Databases

Bill Howe

INFX 575: Data Science III

Scaling, Applications, and Ethics

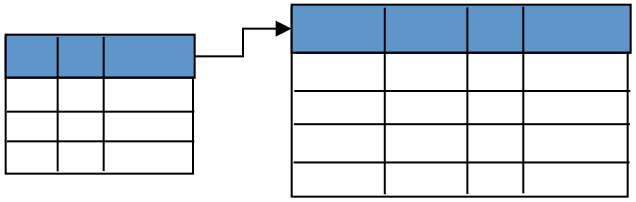
Key Idea: “Logical Data Independence”



views

```
SELECT *
  FROM my_sequences
```

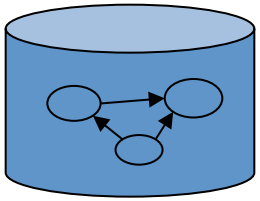
logical data independence



relations

```
SELECT seq
  FROM ncbi_sequences
 WHERE seq = 'GATTACGATATTA';
```

physical data independence



files and pointers

```
f = fopen('table_file');
fseek(10030440);
while (True) {
  fread(&buf, 1, 8192, f);
  if (buf == GATTACGATATTA) {
    . . .
```

What are Views?

- A view is just a query with a name
- We can use the view just like a real table

Why can we do this?

Because we know that every query returns a relation:
We say that the language is "algebraically closed"

View example

A view is a relation defined by a query

Purchase(customer, product, store)
Product(pname, price)

StorePrice(store, price)

```
CREATE VIEW  StorePrice AS
SELECT x.store, y.price
FROM  Purchase x, Product y
WHERE x.pid = y.pid
```

This is like a new table
StorePrice(store,price)

Customer(cid, name, city)
Purchase(customer, product, store)
Product(pname, price)

StorePrice(store, price)

How to Use a View?

- A "high end" store is a store that sold some product over 1000. For each customer, find all the high end stores that they visit. Return a set of (customer-name, high-end-store) pairs.

```
SELECT DISTINCT z.name, u.store
FROM Customer z, Purchase u, StorePrice v
WHERE z.cid = u.customer
AND u.store = v.store
AND v.price > 1000
```

Key Idea: Indexes

- Databases are especially, but not exclusively, effective at "Needle in Haystack" problems:
  - **Extracting small results from big datasets**
  - **Your query will always* finish, regardless of dataset size.**
  - **Indexes are easily built and automatically used when appropriate**

```
CREATE INDEX seq_idx ON sequence(seq);

SELECT seq
  FROM sequence
 WHERE seq = 'GATTACGATATTA';
```

*almost

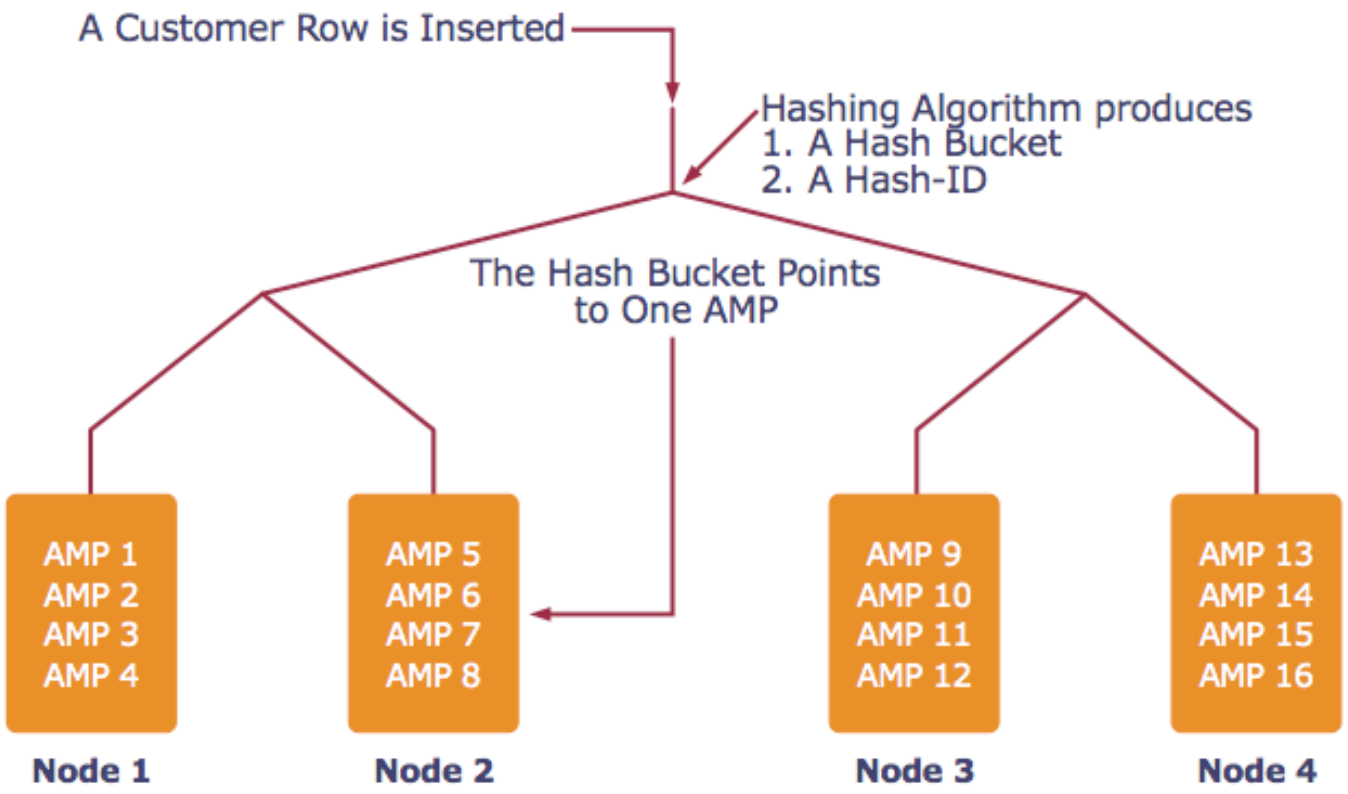
Distributed Query Example

```
CREATE VIEW Sales AS

SELECT * FROM JanSales
  UNION ALL
SELECT * FROM FebSales
  UNION ALL
SELECT * FROM MarSales
```

```
CREATE TABLE MarSales(
   OrderID       INT,
   CustomerID    INT       NOT NULL,
   OrderDate     DATETIME       NULL
      CHECK (DATEPART(mm, OrderDate) = 3),
  CONSTRAINT OrderIDMonth PRIMARY KEY(OrderID)
)
```

Parallel Query Example: Teradata

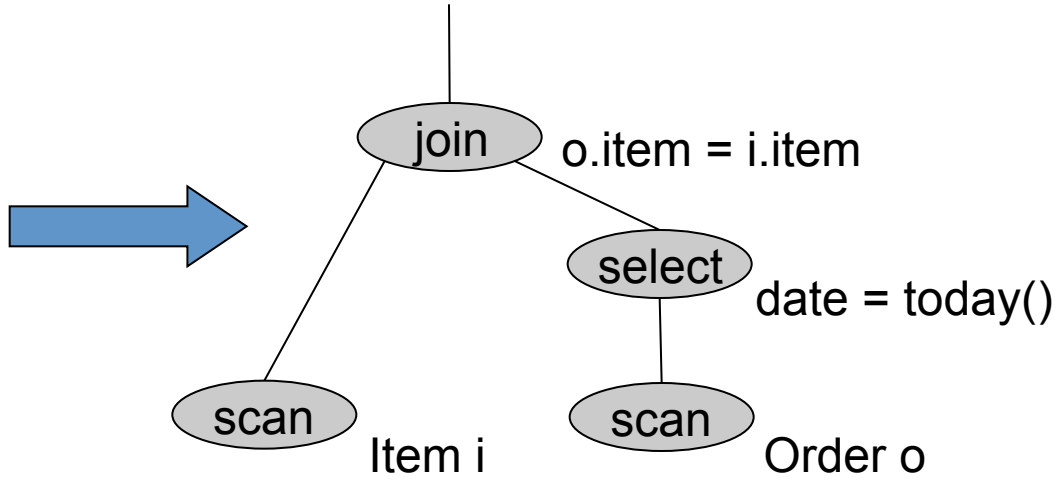


AMP = unit of parallelism

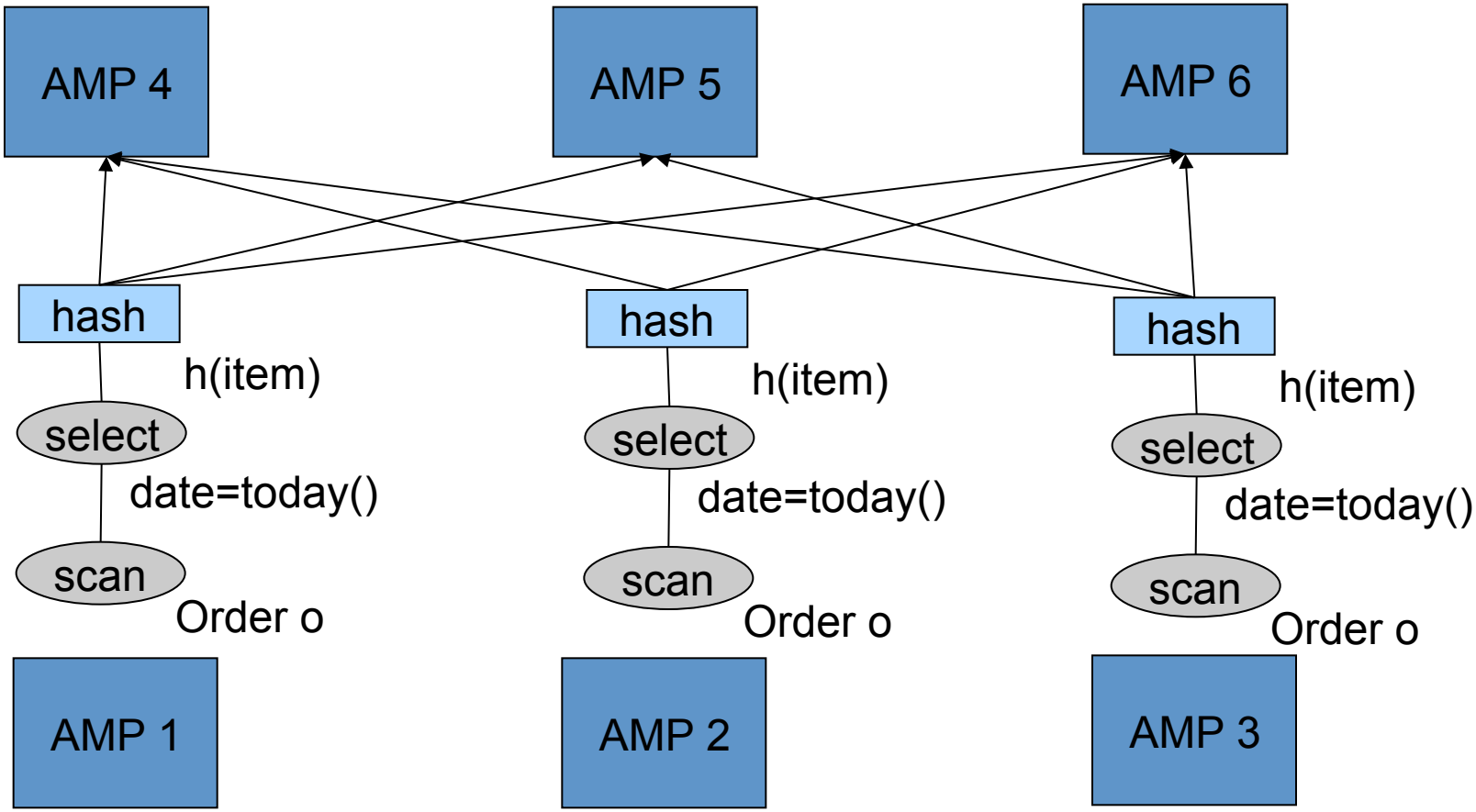
Example System: Teradata

Find all orders from today, along with the items ordered

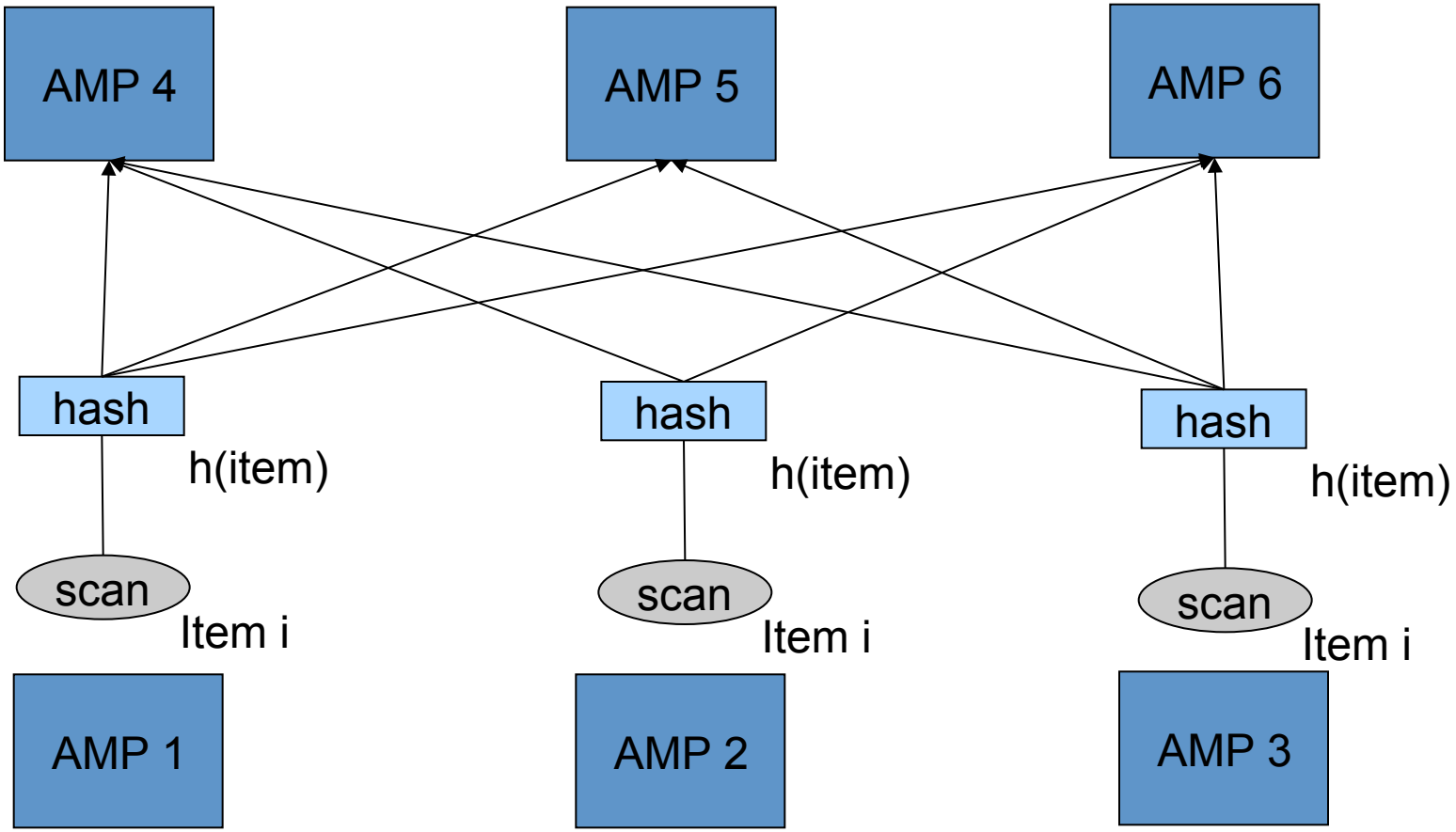
```
SELECT *
  FROM Orders o, Lines i
 WHERE o.item = i.item
   AND o.date = today()
```



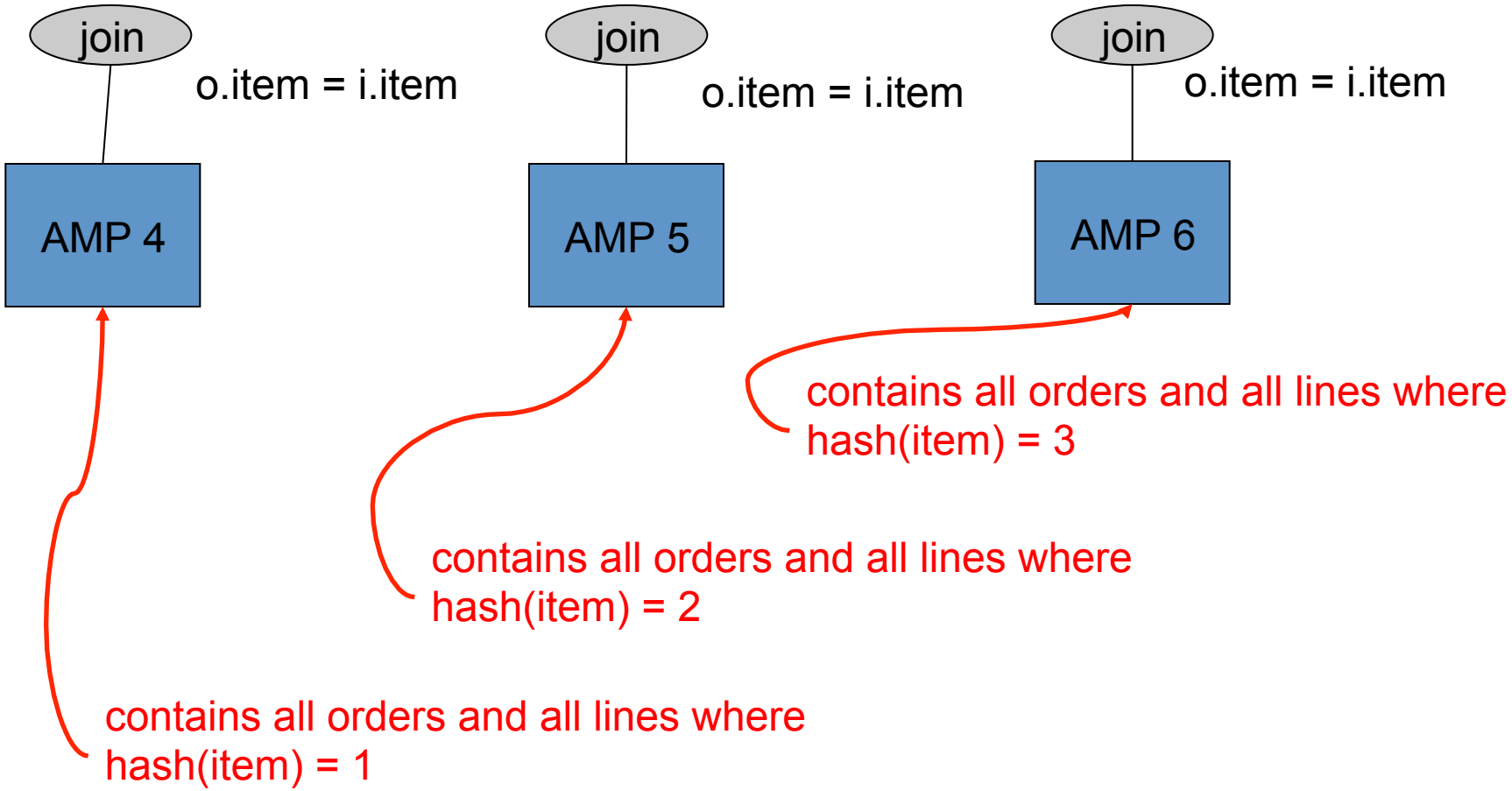
Example System: Teradata



Example System: Teradata



Example System: Teradata



- Reading: Pavlo 2009

MR VS. Databases

MapReduce vs RDBMS

- RDBMS
  - Declarative query languages — HIVE, Pig, Spark, many more
  - Schemas — HIVE, Pig, Spark, many more
  - Logical Data Independence
  - Indexing — Hbase, Accumulo
  - Algebraic Optimization — HIVE, Pig, Spark
  - Caching/Materialized Views
  - *ACID/Transactions*
- MapReduce
  - Very High Scalability
  - Direct programmability
  - Fault-tolerance

Pavlo et al. 2009

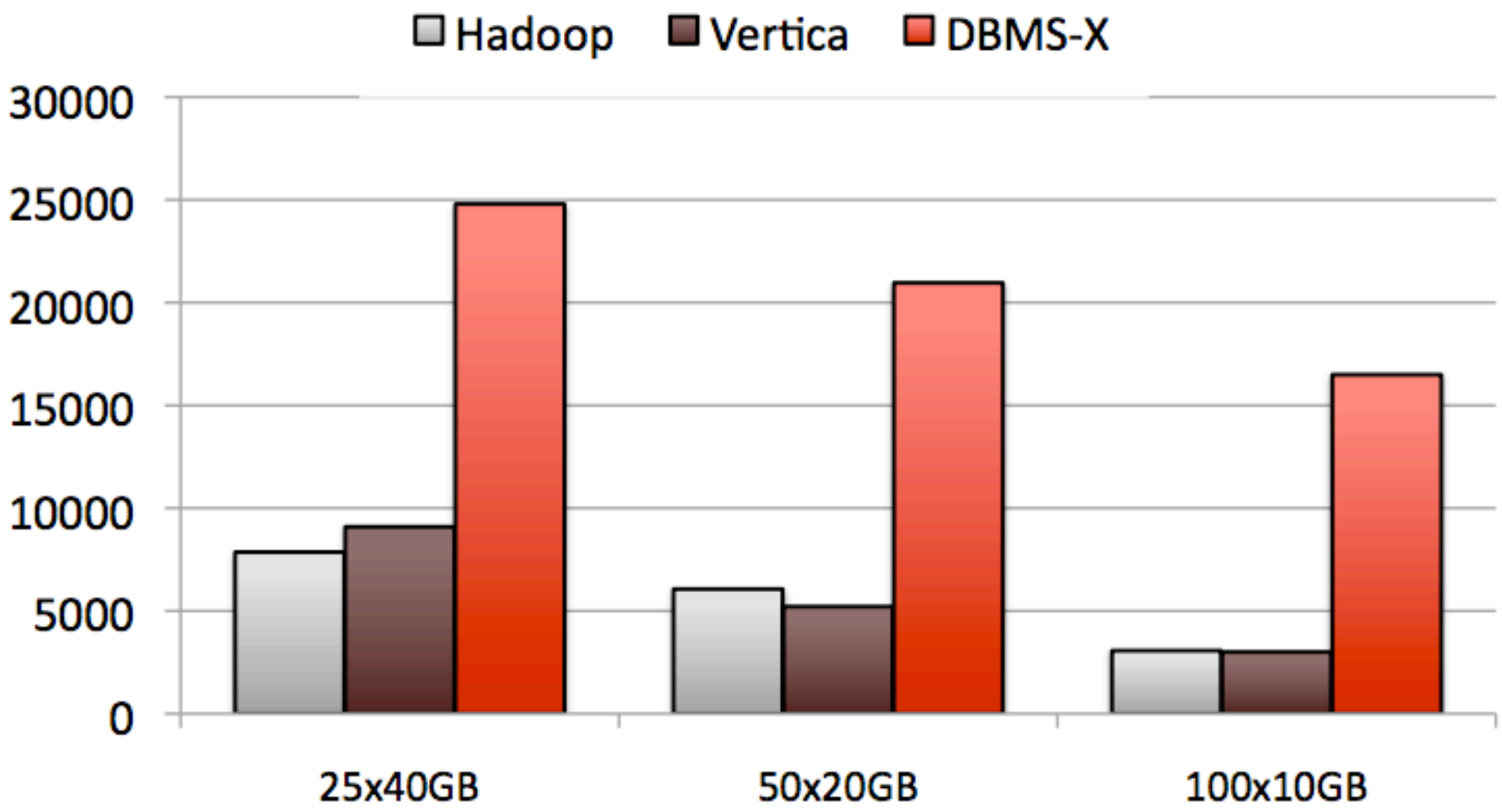
Hadoop vs. RDBMS

- Comparison of 3 systems
  - Hadoop
  - Vertica (a column-oriented database)
  - DBMS-X (a row-oriented database)
    - rhymes with “schmoracle”
- Qualitative
  - Programming model, ease of setup, features, etc.
- Quantitative
  - Data loading, different types of queries

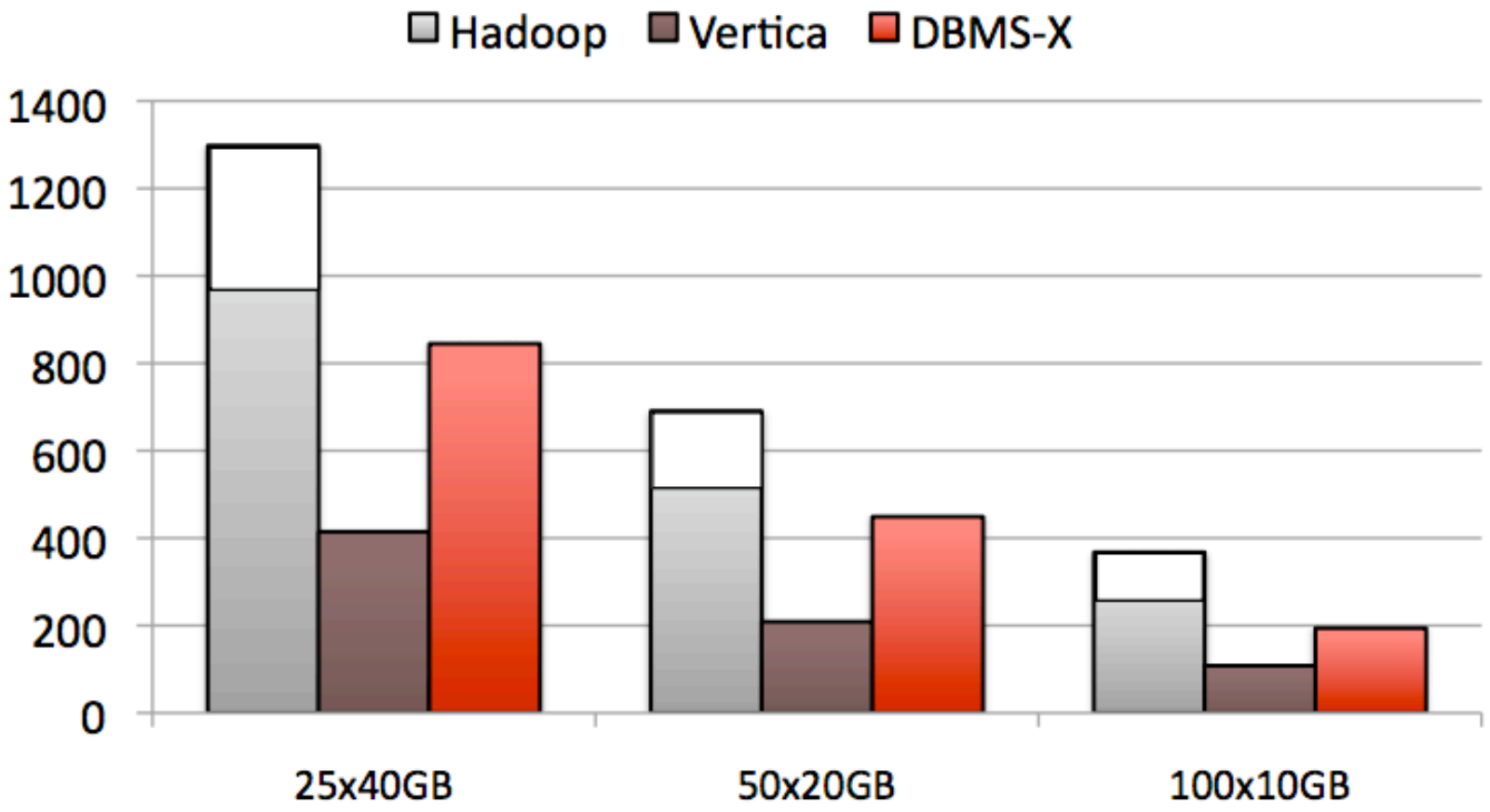
Grep Task

- Find 3-byte pattern in 100-byte record
  - *1 match per 10,000 records*
- Data set:
  - *10-byte unique key, 90-byte value*
  - *1TB spread across 25, 50, or 100 nodes*
  - *10 billion records*
- Original MR Paper (Dean et al. 2004)

Grep Task Loading Results

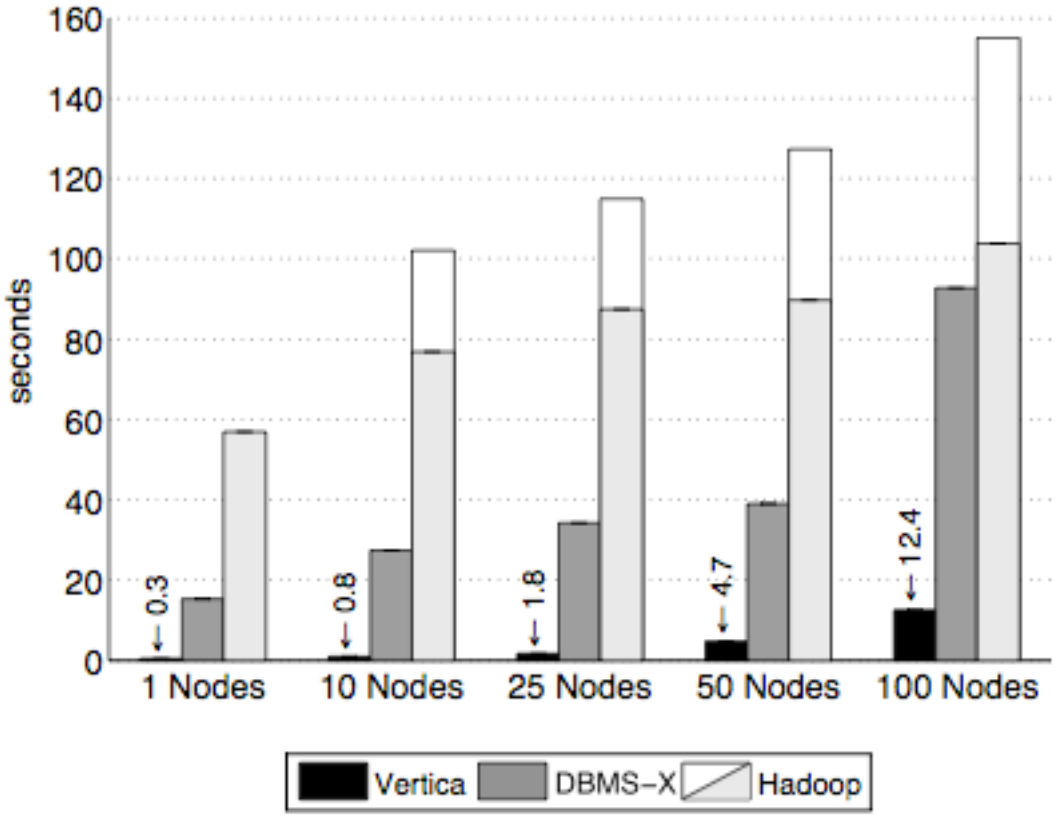


Grep Task Execution Results



Selection Task

```
SELECT pageURL, pageRank
FROM Rankings WHERE pageRank > X
```



1 GB / node

Analytical Tasks

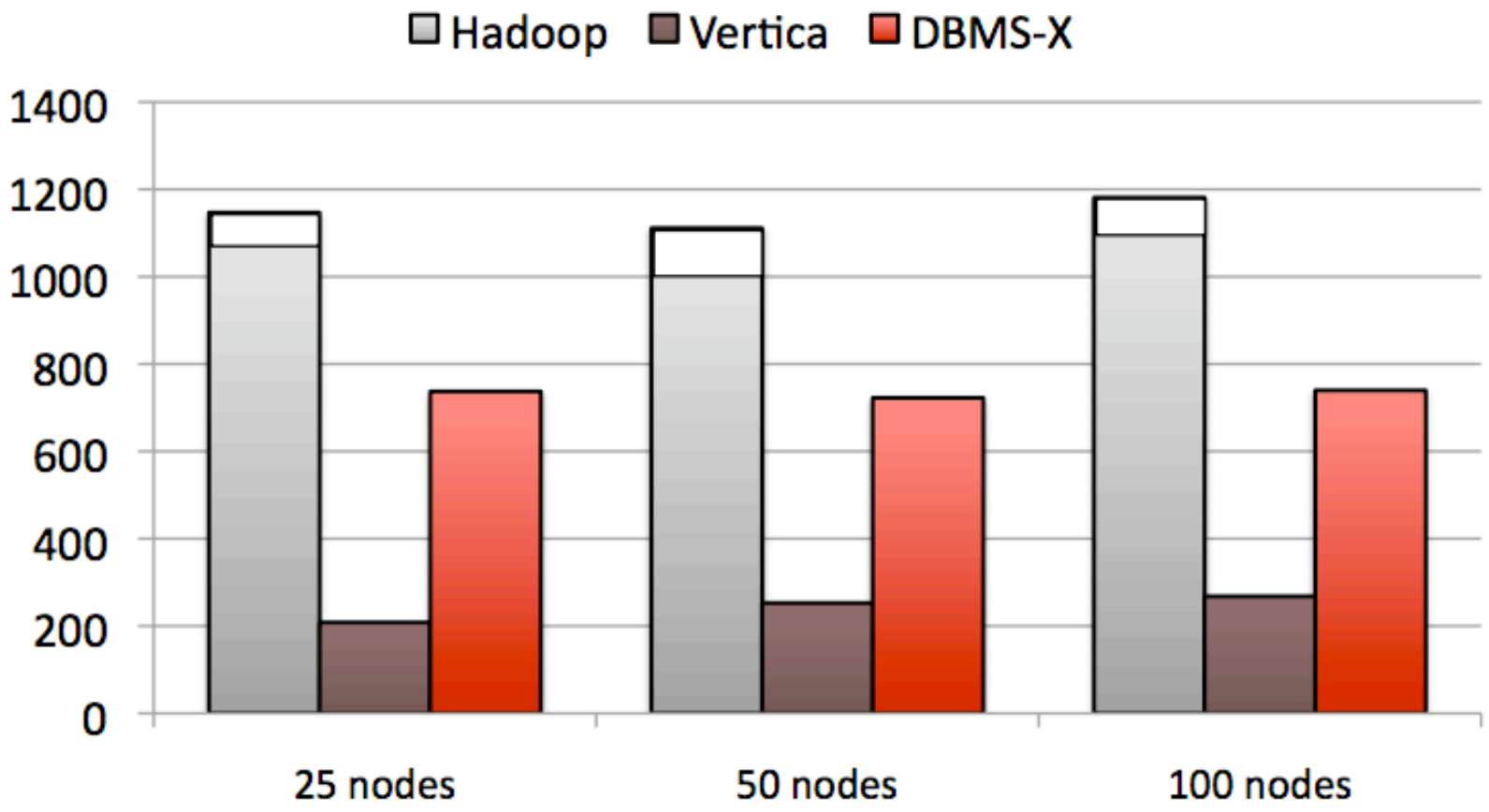
- **Simple web processing schema**
- **Data set:**
  - ***600k HTML Documents (6GB/node)***
  - ***155 million UserVisit records (20GB/node)***
  - ***18 million Rankings records (1GB/node)***

Aggregate Task

- Simple query to find adRevenue by IP prefix

```
SELECT SUBSTR(sourceIP, 1, 7),
       SUM(adRevenue)
  FROM userVistits
 GROUP BY SUBSTR(sourceIP, 1, 7)
```

Aggregate Task Results



Join Task

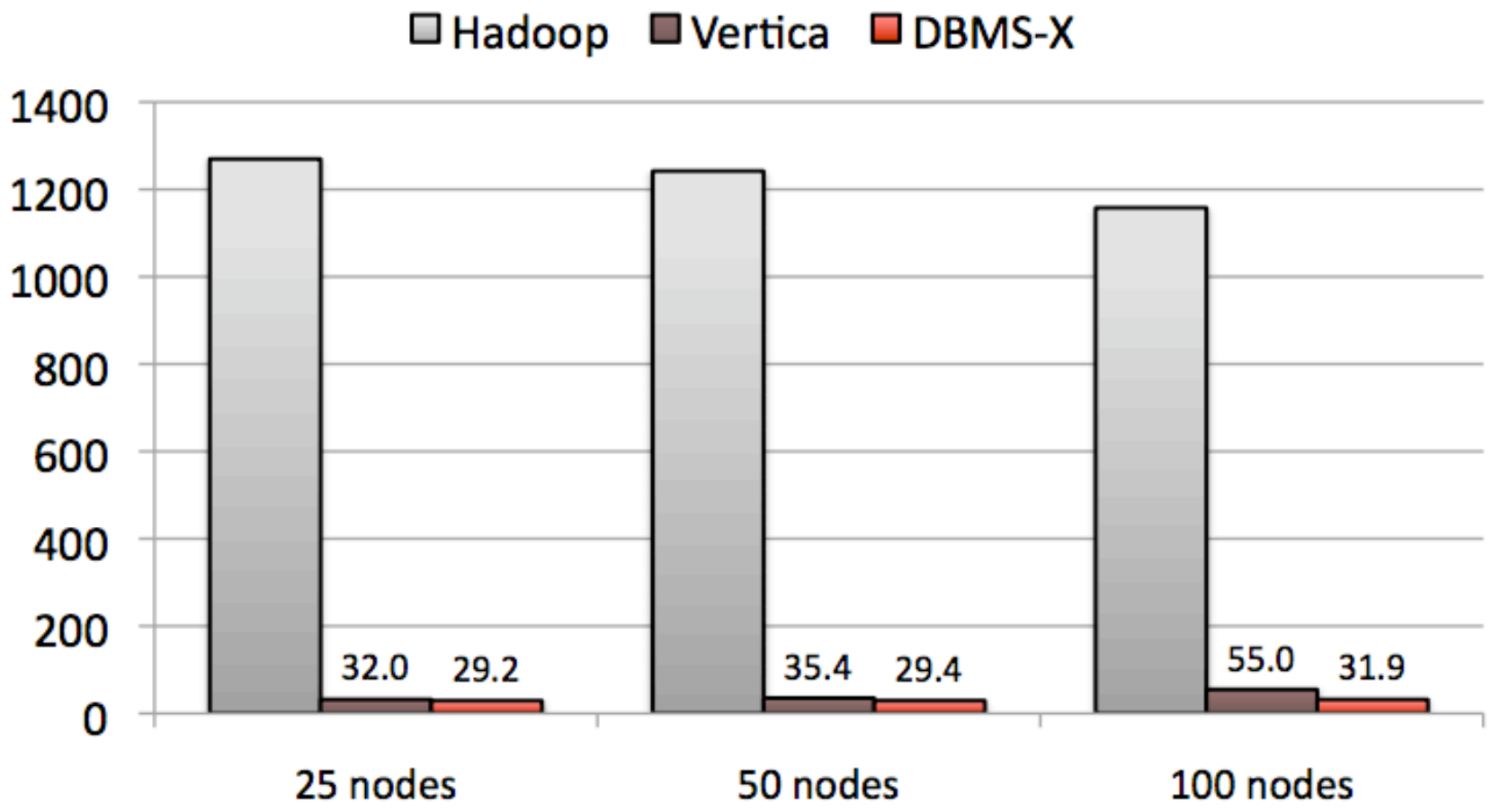
- **Find the sourceIP that generated the most adRevenue along with its average pageRank.**
- **Implementations:**
  - ***DBMSs – Complex SQL using temporary table.***
  - ***MapReduce – Three separate MR programs.***

Join Task

```
SELECT INTO TempsourceIP,
            AVG(pageRank)as avgPageRank,
            SUM(adRevenue)as totalRevenue
FROM RankingsAS R
   , UserVisitsAS UV
WHERE R.pageURL = UV.destURL
AND UV.visitDate
  BETWEEN '2000-01-15'
  AND '2000-01-22'
GROUP BY UV.sourceIP;

SELECT sourceIP,
       totalRevenue,
       avgPageRank
FROM Temp
ORDER BY totalRevenueDESC
LIMIT 1;
```

Join Task Results



Problems with this analysis?

- Other ways to avoid sequential scans?
- Fault-tolerance in large clusters?
- Tasks that cannot be expressed as queries?

Warning: VERY OLD DATA

Google's Response: Cluster Size

- **Largest known database installations:**
  - ***Greenplum – 96 nodes – 4.5 PB (eBay) [1]***
  - ***Teradata – 72 nodes – 2+ PB (eBay) [1]***
- **Largest known MR installations:**
  - ***Hadoop – 3658 nodes – 1 PB (Yahoo) [2]***
  - ***Hive – 600+ nodes – 2.5 PB (Facebook) [3]***

[1] eBay's two enormous data warehouses – April 30th, 2009
http://www.dbms2.com/2009/04/30/ebays-two-enormous-data-warehouses/

[2] Hadoop Sorts a Petabyte in 16.25 Hours and a Terabyte in 62 Seconds – May 11th, 2009
http://developer.yahoo.net/blogs/hadoop/2009/05/hadoop_sorts_a_petabyte_in_162.html

[3] Hive - A Petabyte Scale Data Warehouse using Hadoop – June 10th, 2009
http://www.facebook.com/note.php?note_id=89508453919

Concluding Remarks

- What can *MapReduce* learn from *Databases*?
  - *Declarative languages are a good thing.*
  - *Schemas are important.*
- What can *Databases* learn from *MapReduce*?
  - *Query fault-tolerance.*
  - *Support for in situ data.*
  - *Embrace open-source.*

Other Benchmarked Systems

- **HadoopDB (Abadi '09 - Yale)**
  - ***Replaced Hadoop filesystem with Postgres.***
  - ***Makes JDBC calls inside of MR functions.***
- **Hive (Thusoo '09 - Facebook)**
  - ***Data warehouse interface on top of Hadoop.***
  - ***Converts SQL-like language to MR programs.***