CMU SCS

Carnegie Mellon Univ.
Dept. of Computer Science
15-415/615 - DB Applications

C. Faloutsos – A. Pavlo

Lecture#28: Modern Database Systems

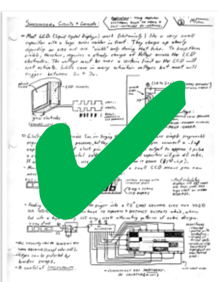
CMU SCS

Administrivia – Final Exam

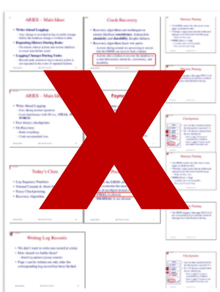
- **Who:** You
- **What:** R&G Chapters 15-22
- **When:** Tuesday May 6th 5:30pm- 8:30pm
- **Where:** WEH 7500
- **Why:** Databases will help your love life.

CMU SCS

Administrivia – Final Exam



Handwritten Notes



Printed Notes

CMU SCS

Today's Class

- Distributed OLAP
- OldSQL vs. NoSQL vs. NewSQL
- How to scale a database system

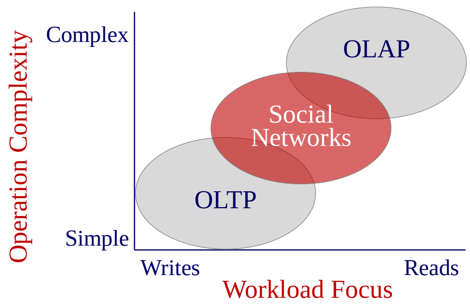
CMU SCS

OLTP vs. OLAP

- On-line Transaction Processing:
  - Short-lived txns.
  - Small footprint.
  - Repetitive operations.
- On-line Analytical Processing:
  - Long running queries.
  - Complex joins.
  - Exploratory queries.

CMU SCS

Workload Characterization



Michael Stonebraker – *"Ten Rules For Scalable Performance In Simple Operation' Datastores"*
http://cacm.acm.org/magazines/2011/6/108651

CMU SCS

Relational Database Backlash

- New Internet start-ups hit the limits of single-node DBMSs.
- Early companies used custom middleware to shard databases across multiple DBMSs.
- Google was a pioneer in developing non-relational DBMS architectures.

CMU SCS

MapReduce

- Simplified parallel computing paradigm for large-scale data analysis.
- Originally proposed by Google in 2004.
- Hadoop is the current leading open-source implementation.

CMU SCS

MapReduce Example

Calculate total order amount per day after Jan 1st.

```
REDUCE(key, values) {
   sum = 0;
   while (values.hasNext()) {
      sum += values.next();
   }
   output(key, sum);
}
```

CMU SCS

What MapReduce Does Right

- Since all intermediate results are written to HDFS, if one node crashes the entire query does not need to be restarted.
- Easy to load data and start running queries.
- Great for semi-structured data sets.

CMU SCS

What MapReduce Did Wrong

- Have to parse/cast values every time:
  - Multi-attribute values handled by user code.
  - If data format changes, code must change.
- Expensive execution:
  - Have to send data to executors.
  - A simple join requires multiple MR jobs.

CMU SCS

Join Example

- Find sourceIP that generated most adRevenue along with its average pageRank.

CMU SCS

Join Example – SQL

```
SELECT INTO Temp sourceIP,
                 AVG(pageRank) AS avgPageRank,
                 SUM(adRevenue) AS totalRevenue
  FROM Rankings AS R, UserVisits AS 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 totalRevenue DESC LIMIT 1;
```


CMU SCS

Join Example – MapReduce

Phase 1: Filter

Map:
Emit all records for Rankings.
Filter UserVisits data.

Reduce:
Compute cross product.

Phase 2: Aggregation

Map:
Emit all tuples (i.e., passthrough)

Reduce:
Compute avg pageRank for each sourceIP.

Phase 3: Search

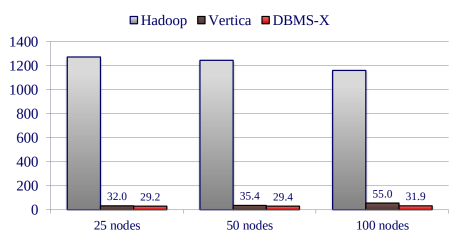
Map:
Emit all tuples (i.e., passthrough)

Reduce:
Scan entire input and emit the record with greatest adRevenue sum.

CMU SCS

Join Example – Results

- Find sourceIP that generated most adRevenue along with its average pageRank.



CMU SCS

Distributed Joins Are Hard

```
SELECT * FROM table1, table2
 WHERE table1.val = table2.val
```

- Assume tables are horizontally partitioned:
  – Table1 Partition Key → table1.key
  – Table2 Partition Key → table2.key
- **Q:** How to execute?
- Naïve solution is to send all partitions to a single node and compute join.

CMU SCS

Semi-Joins

- First distribute the join attributes between nodes and then recreate the full tuples in the final output.
  – Send just enough data from each table to compute which rows to include in output.
- Lots of choices make this problem hard:
  – What to materialize?
  – Which table to send?

CMU SCS

MapReduce in 2014

- SQL/Declarative Query Support
- Table Schemas
- Column-oriented storage.

HIVE Hortonworks STINGER cloudera IMPALA

CMU SCS

Column Stores

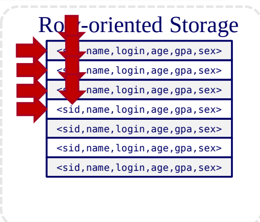
- Store tables as sections of columns of data rather than as rows of data.

CMU SCS

Column Stores

```
SELECT sex, AVG(GPA) FROM student
  GROUP BY sex
```

sid	name	login	age	gpa	sex
1001	Faloutsos	christos@cs	45	4.0	M
1002	Bieber	jbieber@cs	21	3.9	M
1003	Tupac	shakur@cs	26	3.5	M
1004	Ke$sha	kesha@cs	22	4.0	F
1005	LadyGaGa	gaga@cs	24	3.5	F
1006	Obama	obama@cs	50	3.7	M

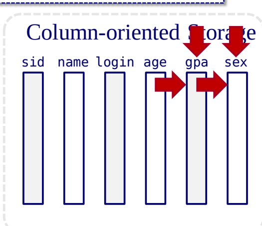


CMU SCS

Column Stores

```
SELECT sex, AVG(GPA) FROM student
  GROUP BY sex
```

sid	name	login	age	gpa	sex
1001	Faloutsos	christos@cs	45	4.0	M
1002	Bieber	jbieber@cs	21	3.9	M
1003	Tupac	shakur@cs	26	3.5	M
1004	Ke$sha	kesha@cs	22	4.0	F
1005	LadyGaGa	gaga@cs	24	3.5	F
1006	Obama	obama@cs	50	3.7	M



CMU SCS

Column Stores

- Only scan the columns that a query needs.
- Allows for amazing compression ratios:
  - Values for the same query are usually similar.
- Main goal is delay materializing a record back to its row-oriented format for as long as possible inside of the DBMS.
- Inserts/Updates/Deletes are harder…

CMU SCS

Column Store Systems

- Many column-store DBMSs
  - Examples: Vertica, Sybase IQ, MonetDB
- Hadoop storage library:
  - Example: Parquet, RCFile

CMU SCS

NoSQL

- In addition to MapReduce, Google created a distributed DBMS called BigTable.
  - It used a GET/PUT API instead of SQL.
  - No support for txns.
- Newer systems have been created that follow BigTable's anti-relational spirit.

CMU SCS

NoSQL Systems

Key/Value	Column-Family	Documents
redis	cassandra	mongoDB
riak	APACHE HBASE	COUCHBASE
FOUNDATIONDB	HYPERTABLE	CouchDB relax
membase		RethinkDB
amazon DynamoDB		

Faloutsos/Pavlo CMU SCS 15-415/615 25

CMU SCS

NoSQL Drawbacks

- Developers write code to handle eventually consistent data, lack of transactions, and joins.
- Not all applications can give up strong transactional semantics.

Faloutsos/Pavlo CMU SCS 15-415/615 26

CMU SCS

NewSQL

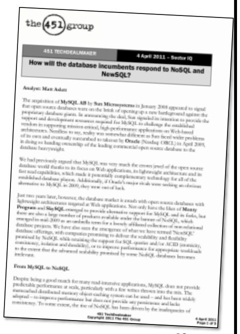
- Next generation of relational DBMSs that can scale like a NoSQL system but without giving up SQL or txns.

Faloutsos/Pavlo CMU SCS 15-415/615 27

CMU SCS

Aslett White Paper

[Systems that] deliver the scalability and flexibility promised by NoSQL while retaining the support for SQL queries and/or ACID, or to improve performance for appropriate workloads.


the 451 group

How will the database incumbents respond to NoSQL and NewSQL?

Matt Aslett – 451 Group (April 4th, 2011)
https://www.451research.com/report-short?entityId=66963

28

CMU SCS

Wikipedia Article

A class of modern relational database systems that provide the same scalable performance of NoSQL systems for OLTP workloads while still maintaining the ACID guarantees of a traditional database system.

Wikipedia (April 2014)
http://en.wikipedia.org/wiki/NewSQL

29

CMU SCS

NewSQL Systems

New Design	MySQL Engines	Middleware
NuoDB, VoltDB, SAP HANA, deepdb, memsql, Clustrix, Google Spanner	ScaleDB, Tokutek	ScaleBase, ScaleArc, dbShards

Faloutsos/Pavlo CMU SCS 15-415/615 30

CMU SCS

NewSQL Implementations

- Distributed Concurrency Control
- Main Memory Storage
- Hybrid Architectures
  - Support OLTP and OLAP in single DBMS.
- Query Code Compilation

Faloutsos/Pavlo CMU SCS 15-415/615 31