The Cloud: What’s it all About?

- Cloudera
- Hive
- Impala
- Hadoop
- Spark
Beyond RDBMS

The Relational Model is too limiting!

• Simple data model – doesn’t capture semantics
  – Object-Oriented DBMS (‘80s)
• Fixed schema – not flexible enough
  – XML databases (‘90s)
• Too heavyweight/slow
  – NoSQL databases (‘00s)

The Latest:  Cloud Databases

• PERFORMANCE!
  – More speed, bigger data
• But this doesn’t come for free
  – Eventual consistency (eventually all the updates will occur)
  – No isolation guarantees
  – Limited reliability guarantees
Cloud Databases: Why?

• Scaling
  – 1000’s of nodes working simultaneously to analyze data

• Answer challenging queries on big data
  – If you can express the query in a limited query language

• Several examples
  – Hadoop, Spark, …

Are we Post-Relational?

• Object-oriented database → object-relational database
  – Today: Commercial RDBMS includes type extensibility and OO features

• XML database
  – XML storage tools for RDBMS

• Cloud Database
  – See Hive – will we see Map-Reduce engines as part of traditional RDBMS?
Cloud Data Processing Basic Idea: Divide and Conquer

- Divide data into units
- Compute on those units
- Combine results
- *Need algorithms where this works!*

Distributed Indexing

- Distributed processing driven by need to index and analyze huge amounts of data (i.e., the Web)
- Large numbers of inexpensive servers used rather than larger, more expensive machines
- *MapReduce* is a distributed programming tool designed for indexing and analysis tasks
Map/Reduce

- Map/Reduce is a programming model for efficient distributed computing
- Works like a Unix pipeline:
  - cat input | grep | sort | uniq -c | cat > output
  - Input | Map | Shuffle & Sort | Reduce | Output
- Efficiency from
  - Streaming through data, reducing seeks
  - Pipelining
- A good fit for a lot of applications
  - Log processing
  - Web index building

MapReduce

- Distributed programming framework that focuses on data placement and distribution
- **Mapper**
  - Generally, transforms a list of items into another list of items of the same length
- **Reducer**
  - Transforms a list of items into a single item
  - Definitions not so strict in terms of number of outputs
- Many mapper and reducer tasks on a cluster of machines
MapReduce Dataflow

- Basic process
  - Map stage which transforms data records into pairs, each with a key and a value
  - Shuffle uses a hash function so that all pairs with the same key end up next to each other and on the same machine
  - Reduce stage processes records in batches, where all pairs with the same key are processed at the same time
- Idempotence of Mapper and Reducer provides fault tolerance
  - multiple operations on same input gives same output
Hash Join Revisited
(Please watch the lecture!)

Aardvark  Caiman  Eagle  Deer  Alpaca  Alligator  Butterfly  Ferret  Bison  Bobcat  Bear  Bird  Bat

Aardvark  Caiman  Eagle  Deer  Albatross  Avocet  Butterfly  Ferret  Badger  Bobcat  Bear  Bird  Bat