

PURDUE
UNIVERSITY

CS54100: Database Systems

Cloud Databases: The Next Post-Relational World

18 April 2012

Prof. Chris Clifton



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)

CS54100



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

CS54100



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
- **Example: Hadoop**
 - Slides courtesy Yahoo!



CS54100

Introduction to Hadoop

Owen O'Malley
Yahoo!, Grid Team
owen@yahoo-inc.com

YAHOO!



Problem

- How do you scale up applications?
 - Run jobs processing 100's of terabytes of data
 - Takes 11 days to read on 1 computer
- Need lots of cheap computers
 - Fixes speed problem (15 minutes on 1000 computers), but...
 - Reliability problems
 - In large clusters, computers fail every day
 - Cluster size is not fixed
- Need common infrastructure
 - Must be efficient and reliable

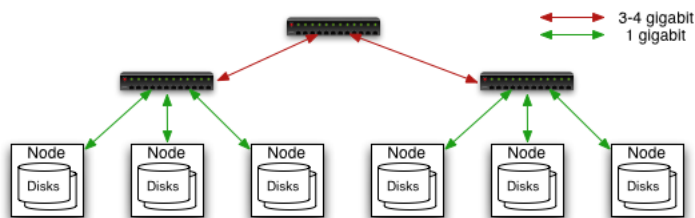
Solution

- Open Source Apache Project
- Hadoop Core includes:
 - Distributed File System - distributes data
 - Map/Reduce - distributes application
- Written in Java
- Runs on
 - Linux, Mac OS/X, Windows, and Solaris
 - Commodity hardware

CCA – Oct 2008

YAHOO!

Commodity Hardware Cluster



- Typically in 2 level architecture
 - Nodes are commodity PCs
 - 40 nodes/rack
 - Uplink from rack is 8 gigabit
 - Rack-internal is 1 gigabit

CCA – Oct 2008

YAHOO!

Distributed File System

- Single namespace for entire cluster
 - Managed by a single *namenode*.
 - Files are single-writer and append-only.
 - Optimized for streaming reads of large files.
- Files are broken in to large blocks.
 - Typically 128 MB
 - Replicated to several *datanodes*, for reliability
- Client talks to both namenode and datanodes
 - Data is not sent through the namenode.
 - Throughput of file system scales nearly linearly with the number of nodes.
- Access from Java, C, or command line.

CCA – Oct 2008

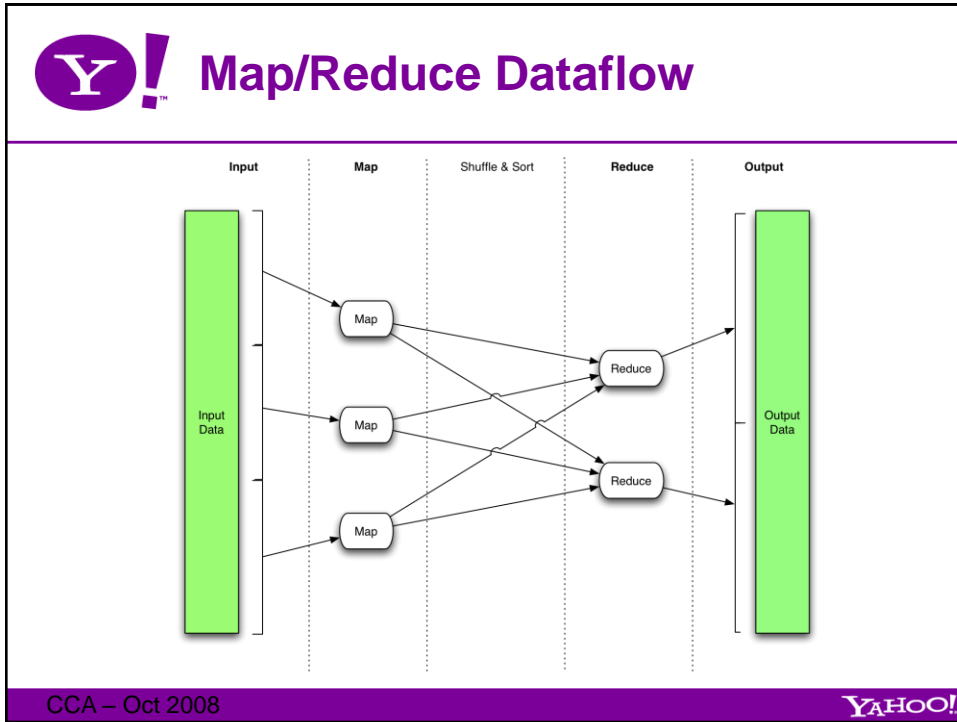
YAHOO!

Map/Reduce

- Map/Reduce is a programming model for efficient distributed computing
- It 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

CCA – Oct 2008

YAHOO!



Map/Reduce features

- **Java and C++ APIs**
 - In Java use Objects, while in C++ bytes
- **Each task can process data sets larger than RAM**
- **Automatic re-execution on failure**
 - In a large cluster, some nodes are always slow or flaky
 - Framework re-executes failed tasks
- **Locality optimizations**
 - Map-Reduce queries HDFS for locations of input data
 - Map tasks are scheduled close to the inputs when possible

At the bottom of the slide, the text "CCA – Oct 2008" is on the left and the "YAHOO!" logo is on the right.



Select word, count(*) from doc group by word;

```

public class WordCount {
public static class Map extends MapReduceBase implements
    Mapper<LongWritable, Text, Text, IntWritable> {
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();
public void map(LongWritable key, Text value,
    OutputCollector<Text, IntWritable> output, Reporter
    reporter) throws IOException {
    String line = value.toString();
    StringTokenizer tokenizer = new StringTokenizer(line);
    while (tokenizer.hasMoreTokens()) {
        word.set(tokenizer.nextToken());
        output.collect(word, one);
    }
}
}
public static class Reduce extends MapReduceBase
implements Reducer<Text, IntWritable, Text,
    IntWritable> {
public void reduce(Text key, Iterator<IntWritable> values,
    OutputCollector<Text, IntWritable> output, Reporter
    reporter) throws IOException {
    int sum = 0;
    while (values.hasNext()) {
        sum += values.next().get();
    }
    output.collect(key, new IntWritable(sum));
}
}
}

public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(WordCount.class);
    conf.setJobName("wordcount");
    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);
    conf.setMapperClass(Map.class);
    conf.setCombinerClass(Reduce.class);
    conf.setReducerClass(Reduce.class);
    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class);
    FileInputFormat.setInputPaths(conf, new
        Path(args[0]));
    FileOutputFormat.setOutputPath(conf, new
        Path(args[1]));
    JobClient.runJob(conf);
}

```



How is Yahoo using Hadoop?

- We started with building better applications
 - Scale up web scale batch applications (search, ads, ...)
 - Factor out common code from existing systems, so new applications will be easier to write
 - Manage the many clusters we have more easily
- The mission now includes research support
 - Build a **huge** data warehouse with many Yahoo! data sets
 - Couple it with a huge compute cluster and programming models to make using the data easy
 - Provide this as a service to our researchers
 - We are seeing great results!
 - Experiments can be run much more quickly in this environment



Running Production WebMap

- Search needs a graph of the “known” web
 - Invert edges, compute link text, whole graph heuristics
- Periodic batch job using Map/Reduce
 - Uses a chain of ~100 map/reduce jobs
- Scale
 - 1 trillion edges in graph
 - Largest shuffle is 450 TB
 - Final output is 300 TB compressed
 - Runs on 10,000 cores
 - Raw disk used 5 PB
- Written mostly using Hadoop’s C++ interface

CCA – Oct 2008

YAHOO!



Research Clusters

- The grid team runs the research clusters as a service to Yahoo researchers
- Mostly data mining/machine learning jobs
- Most research jobs are *not* Java:
 - 42% Streaming
 - Uses Unix text processing to define map and reduce
 - 28% Pig
 - Higher level dataflow scripting language
 - 28% Java
 - 2% C++

CCA – Oct 2008

YAHOO!



- Needed offline conversion of public domain articles from 1851-1922.
- Used Hadoop to convert scanned images to PDF
- Ran 100 Amazon EC2 instances for around 24 hours
- 4 TB of input
- 1.5 TB of output

A COMPUTER WANTED.

WASHINGTON, May 1.—A civil service examination will be held May 18 in Washington, and, if necessary, in other cities, to secure eligibles for the position of computer in the Nautical Almanac Office, where two vacancies exist—one at \$1,000, the other at \$1,400. The examination will include the subjects of algebra, geometry, trigonometry, and astronomy. Application blanks may be obtained of the United States Civil Service Commission.

Published 1892, copyright New York Times

CCA – Oct 2008

YAHOO!



- Started by Jim Gray at Microsoft in 1998
- Sorting 10 billion 100 byte records
- Hadoop won the general category in 209 seconds
 - 910 nodes
 - 2 quad-core Xeons @ 2.0Ghz / node
 - 4 SATA disks / node
 - 8 GB ram / node
 - 1 gb ethernet / node
 - 40 nodes / rack
 - 8 gb ethernet uplink / rack
- Previous records was 297 seconds
- Only hard parts were:
 - Getting a total order
 - Converting the data generator to map/reduce

CCA – Oct 2008

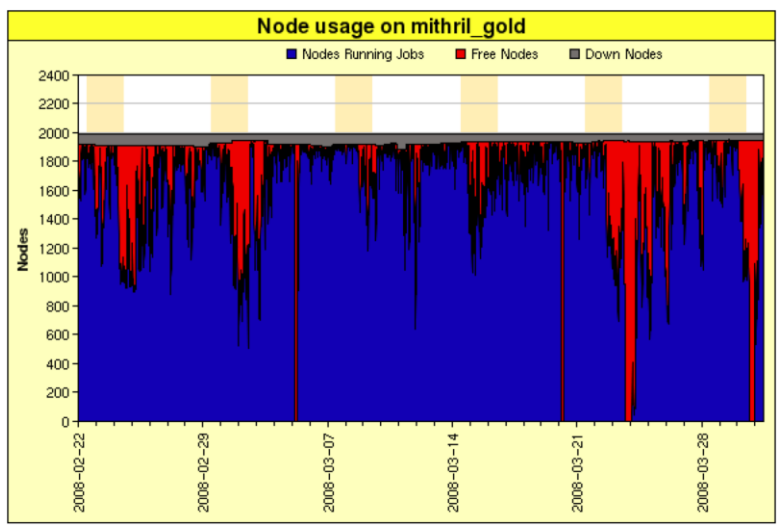
YAHOO!

Y! Hadoop clusters

- We have ~20,000 machines running Hadoop
- Our largest clusters are currently 2000 nodes
- Several petabytes of user data (compressed, unreplicated)
- We run hundreds of thousands of jobs every month



Y! Research Cluster Usage



CCA – Oct 2008





Hadoop Community

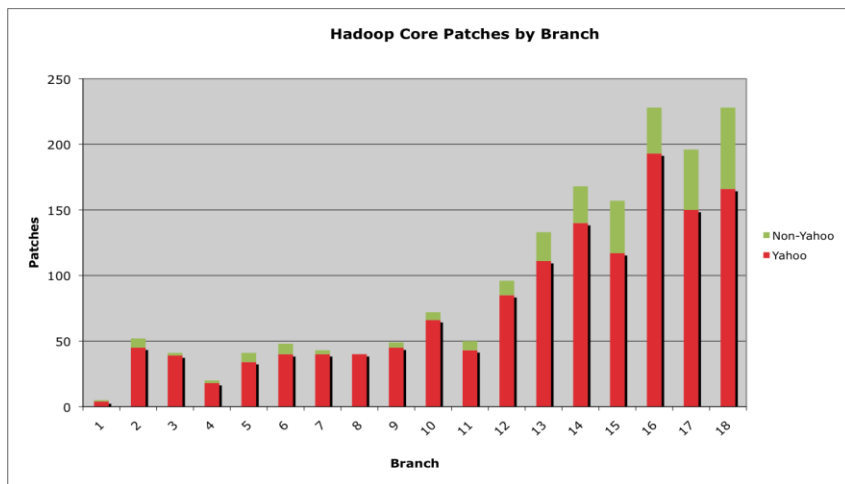
- Apache is focused on project communities
 - Users
 - Contributors
 - write patches
 - Committers
 - can commit patches **too**
 - Project Management Committee
 - vote on new committers and releases **too**
- Apache is a meritocracy
- Use, contribution, and diversity is growing
 - But we need and want more!

CCA – Oct 2008

YAHOO!



Size of Releases



CCA – Oct 2008

YAHOO!



Who Uses Hadoop?

- Amazon/A9
- AOL
- Facebook
- Fox interactive media
- Google / IBM
- New York Times
- PowerSet (now Microsoft)
- Quantcast
- Rackspace/Mailtrust
- Veoh
- Yahoo!
- More at <http://wiki.apache.org/hadoop/PoweredBy>

CCA – Oct 2008

YAHOO!



What's Next?

- Better scheduling
 - Pluggable scheduler
 - Queues for controlling resource allocation between groups
- Splitting Core into sub-projects
 - HDFS, Map/Reduce, Hive
- Total Order Sampler and Partitioner
- Table store library
- HDFS and Map/Reduce security
- High Availability via Zookeeper
- Get ready for Hadoop 1.0

CCA – Oct 2008

YAHOO!



But what about...

- Schema
 - Need to know what the data is about
- Queries
 - Do you really want to write map-reduce programs?
 - Optimization?

CS54100



HIVE: RDBMS on Hadoop



- Limited schema
 - Tables
 - Primitive types
- Subset of SQL
 - Select-Project
 - (equi)join
 - Group by
- Operations implemented using Map-Reduce

CS54100



What is Hive?

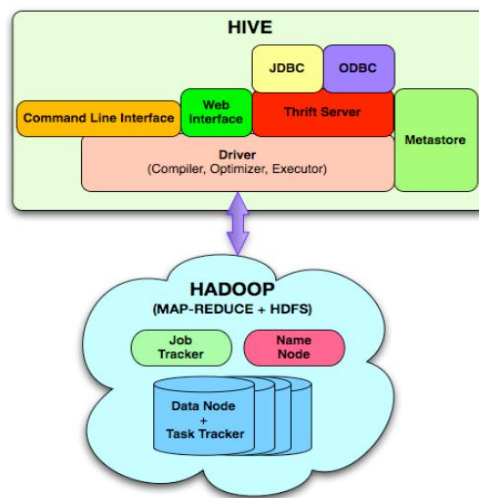


- A system for managing and querying structured data built on top of Hadoop
- Three main components:
 - MapReduce for execution
 - Hadoop Distributed File System for storage
 - Metadata in an RDBMS
- Hive QL based on SQL
 - Easy for users familiar with SQL

30



Hive Architecture



32

Fig. 1: Hive System Architecture



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?

CS54100