An Introduction to Hadoop, Mahout & HBase

Lukáš Vlček, JBUG Brno, May 2012


hadoop

Hadoop

- Open Source (ASL2) implementation of Google's **MapReduce**[1] and Google **DFS** (Distributed File System) [2] (~from 2006 Lucene subproject)

[1] **MapReduce: Simplified Data Processing on Large Scale Clusters** by Jeffrey Dean and Sanjay Ghemawat, Google labs, 2004

[2] **The Google File System** by Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung, 2003

Hadoop

- MapReduce

  Simple programming model for data processing putting parallel large-scale data analysis into hands of masses.

- HDFS

  A filesystem designed for storing large files with streaming data access patterns, running on clusters of commodity hardware.

- (Common, + other related projects...)

MapReduce programming model

map	(k1,v1)	→	list(k2,v2)
reduce	(k2,list(v2))	→	list(v3)

MapReduce
“Hello World”

- Counting the number of occurrences of each word in collection of documents.

MapReduce
“Hello World”

map(k1,v1) → list(k2,v2)

```
map(key, value){
        // key: document name
        // value: document content
        for each word w in value {
                emitIntermediate(w,1)
        }
}
```

MapReduce
“Hello World”

reduce(k2,list(v2)) → list(v3)

```
reduce(key, values){
        // key: a word
        // value: a list of counts
        int result = 0;
        for each word v in values {
                result += v;
        }
        emit(result);
}
```

MapReduce – benefits

- The model is easy to use

  Steep learning curve.

- Many problems are expressible as MapReduce computations

- MapReduce scales to large clusters

  Such model makes it easy to parallelize and distribute your computation to thousands of machines!

MapReduce – downsides

- The model is easy to use

  People tend to try the simplest approach first.

- Many problems are expressible as MapReduce computations

  MapReduce may not be the best model for you.

- MapReduce scales to large clusters

  It is so easy to overload the cluster with simple code.

More elaborated examples

- Distributed PageRank
- Distributed Dijkstra's algorithm (almost)

Lectures to Google software engineering interns, Summer 2007
http://code.google.com/edu/submissions/mapreduce-minilecture/listing.html

Google PageRank today?

- Aug, 2009: Google moved away from MapReduce back-end indexing system onto a new search architecture, a.k.a. Caffeine.

  http://en.wikipedia.org/wiki/Google_Search#Google_Caffeine

Google Maps?

- “In particular, for large road networks it would be prohibitive to precompute and store shortest paths between all pairs of nodes.”

Engineering Fast Route Planning Algorithms, by Peter Sanders and Dominik Schultes, 2007
http://algo2.iti.kit.edu/documents/routeplanning/weaOverview.pdf

Demand for Real-Time data

- MapReduce batch oriented processing of (large) data does not fit well into growing demand for real-time data.

- Hybrid approach?

  Ted Dunning on Twitter's Storm:
  http://info.mapr.com/ted-storm-2012-03.html
  http://www.youtube.com/channel/UCDbTR_Z_k-EZ4e3JpG9zmhg


mahout

Mahout

- The goal is to build open source scalable machine learning libraries.

  Started by: Isabel Drost, Grant Ingersoll, Karl Wettin

  Map-Reduce for Machine Learning on Multicore, 2006
  http://www.cs.stanford.edu/people/ang//papers/nips06-mapreducemulticore.pdf

Implemented Algorithms

- Classification
- Clustering
- Pattern Mining
- Regression
- Dimension Reduction
- Evolutionary Algorithms
- Recommenders / Collaborative Filtering
- Vector Similarity
- ...

Back to Hadoop

HDFS

HDFS

- Very large files
  - Up to GB and PT.
- Streaming data access
  - Write once, read many times.
  - Optimized for high data throughput.
  - Read involves large portion of the data.
- Commodity HW
  - Clusters made of cheap and low reliable machines.
  - Chance of failure of individual machine is high.

HDFS – don't!

- Low-latency data access
  - HBase is better for low-latency access.
- Lots of small files
  - NameNode memory limit.
- Multiple writes and file modifications
  - Single file writer.
  - Write at the end of file.

HDFS: High Availability

- Currently being added to the trunk

  http://www.cloudera.com/blog/2012/03/high-availability-for-the-hadoop-distributed-file-system-hdfs/

HDFS: Security & File Appends

- Finally available as well, but probably in different branches.

  http://www.cloudera.com/blog/2012/01/an-update-on-apache-hadoop-1-0/

  http://www.cloudera.com/blog/2009/07/file-appends-in-hdfs/

Improved POSIX support

- Available from third party vendors (for example MapR M3 or M5 edition)


APACHE
HBASE

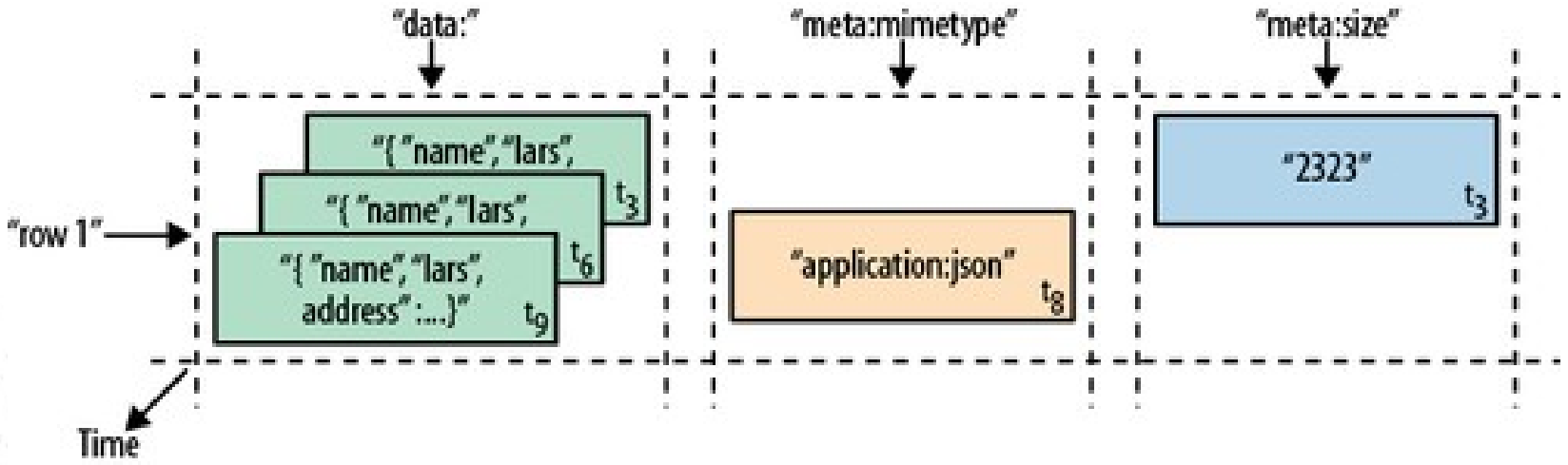
HBase

- Non-relational, auto re-balancing, fault tolerant distributed database
- Modeled after Google BigTable
- Initial prototype in 2007
- Canonical use case *webtable*

Bigtable: A Distributed Storage System for Structured Data, *many* authors, Google labs (2006)

HBase Conceptual view

Row Key	Time Stamp	Column "data:"	Column "meta:" "mimetype"	Column "meta:" "size"	Column "counters:" "updates"
"row1"	t_3	"{ "name" : "lars", "address" : ...}"		"2323"	"1"
	t_6	"{ "name" : "lars", "address" : ...}"			"2"
	t_8		"application/json"		
	t_9	"{ "name" : "lars", "address" : ...}"			"3"



HBase

- Basic operations:
  Get, Put, Scan, Delete
- A {row, column, version} identify a cell
- Allows run Hadoop's MapReduce jobs
- Optimized for high throughput

Use Case: Real-time HBase Analytics

- Nice use case for real-time analysis by Sematext

  http://blog.sematext.com/2012/04/22/hbase-real-time-analytics-rollbacks-via-append-based-updates/

  http://blog.sematext.com/2012/04/27/hbase-real-time-analytics-rollbacks-via-append-based-updates-part-2/

Use Case: Messaging Platform

- Facebook implemented messaging system using HBase

  http://www.facebook.com/notes/facebook-engineering/the-underlying-technology-of-messages/454991608919

Hadoop, Mahout and/or HBase is used by...

- Amazon (A9), Adobe, Ebay, Facebook, Google (university program), IBM, Infochimps, Krugle, Last.fm, LinkedIn, Microsoft, Rackspace, RapLeaf, Spotify, StumbleUpon, Twitter, Yahoo!

  … *many more!*

More Resources

- Hadoop: http://hadoop.apache.org/
- Mahout: http://mahout.apache.org/
- HBase: http://hbase.apache.org/book/book.html

Thank you!

Photo Sources

- http://www.flickr.com/photos/renwest/4909849477/ By renwest, CC BY-NC-SA 2.0
- http://www.flickr.com/photos/asianartsandiego/4838273718/ By Asian Curator at The San Diego Museum of Art, CC BY-NC-ND 2.0
- http://www.flickr.com/photos/zeepack/2932405424/ By ZeePack, CC BY-ND 2.0
- http://www.flickr.com/photos/16516252@N00/3132303565/ By blueboy1478, CC BY-ND 2.0

Backup Slides: Anatomy of MapReduce Execution

- http://code.google.com/edu/parallel/mapreduce-tutorial.html#MRExec

Backup Slides: HDFS Architecture

- http://hadoop.apache.org/common/docs/current/hdfs_design.html