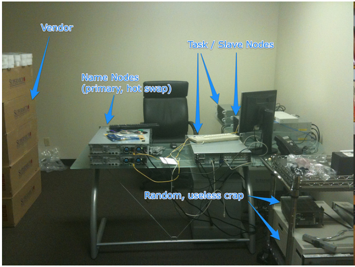
An Overview of Hadoop

Jon Dehdari

The Ohio State University
Department of Linguistics

What is Hadoop?

Hadoop is a software framework for scalable distributed computing



MapReduce

Follows Google's MapReduce framework for distributed computing

- **Scalable** - from one computer to thousands
- **Fault-tolerant** - assumes computers will die
- **Cheap** - uses commodity PCs, no special hardware needed

More active role in distributed computing than Grid Engine, Torque, PBS, Maui, Moab, etc., which are just schedulers

Map & Reduce

Based on the functional programming concepts of higher-order functions:

Map

- **FunProg**: apply a function to every element in a list, return new list
- **MapRed**: apply a function to every row in a file block, return new file block

Reduce (Fold)

- **FunProg**: recursively apply a function to list, return scalar value
- **MapRed**: recursively apply a function to file block, return less-composite value than before

Map & Reduce

Based on the functional programming concepts of higher-order functions:

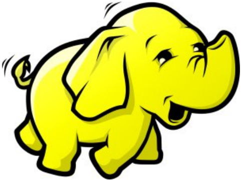
Map

- **FunProg**: apply a function to every element in a list, return new list
- **MapRed**: apply a function to every row in a file block, return new file block

Reduce (Fold)

- **FunProg**: recursively apply a function to list, return scalar value
- **MapRed**: recursively apply a function to file block, return less-composite value than before

Hadoop



- The Hadoop Project is a Free reimplementation of Google's in-house MapReduce and distributed filesystem (GFS)
- Originally written by Doug Cutting & Mike Cafarella, who also created Lucene and Nutch
- Now hosted and managed by the Apache Software Foundation

Who uses Hadoop?

Maybe you've heard of a few of these names:

IBM Watson, Facebook, Yahoo, NSA, Amazon (A9), Adobe, Ebay, Hulu, IBM (Blue Cloud), LinkedIn, New York Times, PARC, Microsoft (Powerset), Twitter, Last.fm, AOL, Rackspace, American Airlines, Apple, Federal Reserve Board of Governors, foursquare, HP, ISI, Netflix, SAP, ...

Facebook crunches 30 petabytes in its Hadoop cluster

https://wiki.apache.org/hadoop/PoweredBy
http://www.theregister.co.uk/2010/12/01/apple_embraces_hadoop/
https://www.microsoft.com/en-us/news/press/2011/oct11/10-12PASS1PR.aspx

Who uses Hadoop?

Maybe you've heard of a few of these names:

IBM Watson, Facebook, Yahoo, NSA, Amazon (A9), Adobe, Ebay, Hulu, IBM (Blue Cloud), LinkedIn, New York Times, PARC, Microsoft (Powerset), Twitter, Last.fm, AOL, Rackspace, American Airlines, Apple, Federal Reserve Board of Governors, foursquare, HP, ISI, Netflix, SAP, ...

Facebook crunches 30 petabytes in its Hadoop cluster

`https://wiki.apache.org/hadoop/PoweredBy`
`http://www.theregister.co.uk/2010/12/01/apple_embraces_hadoop/`
`https://www.microsoft.com/en-us/news/press/2011/oct11/10-12PASS1PR.aspx`

Where does the name come from?

Named after Doug Cutting's son's toy elephant

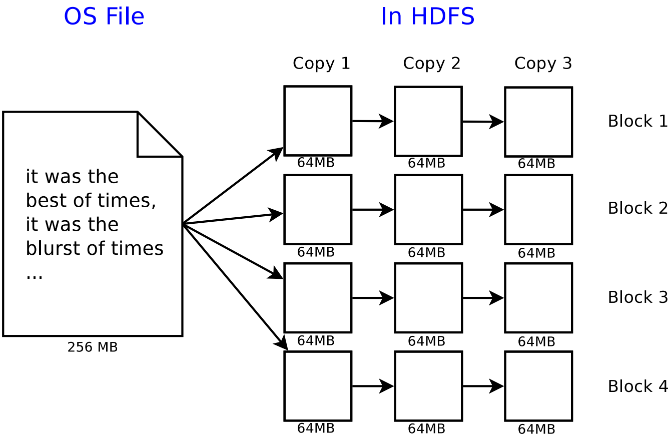
Distributed Filesystem

- Hard drives & RAID arrays are limited to a single machine
- Distributed filesystems work sort of like RAID-1 arrays
- But there are some important differences...

Distributed Filesystem Cont'd

- Files reside in a separate namespace than the OS
- Files are broken-up into blocks (HDFS default: 64 MB)
- Blocks are replicated across multiple nodes (HDFS default: $3x$)
- Due to replication, storage capacity is reduced (to $1/3$ by default)

HDFS



OS File

In HDFS

it was the best of times, it was the blurst of times ...

256 MB

Copy 1

Copy 2

Copy 3

Block 1

Block 2

Block 3

Block 4

64MB

64MB

64MB

64MB

64MB

64MB

64MB

64MB

64MB

64MB

64MB

64MB

HDFS Commands

Most HDFS commands are structured:

```
hadoop fs -[normal_unix_command] -[normal_unix_arguments]
```

For example:

Unix: `ls -R /`

```
hadoop fs -ls -R /
```

Unix: `df -h`

```
hadoop fs -df -h
```

(doesn't account for replication!)

Copy from local filesystem to HDFS

```
hadoop fs -put <localsrc> ... <dst>
```

For more info:

```
hadoop fs
```

HDFS Commands

Most HDFS commands are structured:

```
hadoop fs -[normal_unix_command] -[normal_unix_arguments]
```

For example:

Unix: `ls -R /`

```
hadoop fs -ls -R /
```

Unix: `df -h`

```
hadoop fs -df -h
```

(doesn't account for replication!)

Copy from local filesystem to HDFS

```
hadoop fs -put <localsrc> ... <dst>
```

For more info:

```
hadoop fs
```

HDFS Commands

Most HDFS commands are structured:

```
hadoop fs -[normal_unix_command] -[normal_unix_arguments]
```

For example:

Unix: `ls -R /`

```
hadoop fs -ls -R /
```

Unix: `df -h`

```
hadoop fs -df -h
```

(doesn't account for replication!)

Copy from local filesystem to HDFS

```
hadoop fs -put <localsrc> ... <dst>
```

For more info:

```
hadoop fs
```

HDFS Commands

Most HDFS commands are structured:

```
hadoop fs -[normal_unix_command] -[normal_unix_arguments]
```

For example:

Unix: `ls -R /`

```
hadoop fs -ls -R /
```

Unix: `df -h`

```
hadoop fs -df -h
```

(doesn't account for replication!)

Copy from local filesystem to HDFS

```
hadoop fs -put <localsrc> ... <dst>
```

For more info:

```
hadoop fs
```

Replication

Not a True Backup

Triplicated datanodes provides protection against harddrive failure, but it's not a true backup (ie. it doesn't protect against Raja-isms)

Network Topology

For large installations, HDFS can be made aware of network topology, including nodes, racks, and datacenters, in order to better distribute data

Usage Best Practices

Delete after you're done using a file!!! It uses up three-times the disk space as a normal filesystem

Replication

Not a True Backup

Triplicated datanodes provides protection against harddrive failure, but it's not a true backup (ie. it doesn't protect against Raja-isms)

Network Topology

For large installations, HDFS can be made aware of network topology, including nodes, racks, and datacenters, in order to better distribute data

Usage Best Practices

Delete after you're done using a file!!! It uses up three-times the disk space as a normal filesystem

Replication

Not a True Backup

Triplicated datanodes provides protection against harddrive failure, but it's not a true backup (ie. it doesn't protect against Raja-isms)

Network Topology

For large installations, HDFS can be made aware of network topology, including nodes, racks, and datacenters, in order to better distribute data

Usage Best Practices

Delete after you're done using a file!!! It uses up three-times the disk space as a normal filesystem

The Bottleneck

The main bottleneck is transferring files across a network, so Hadoop keeps computing localized to same node as where the data block resides

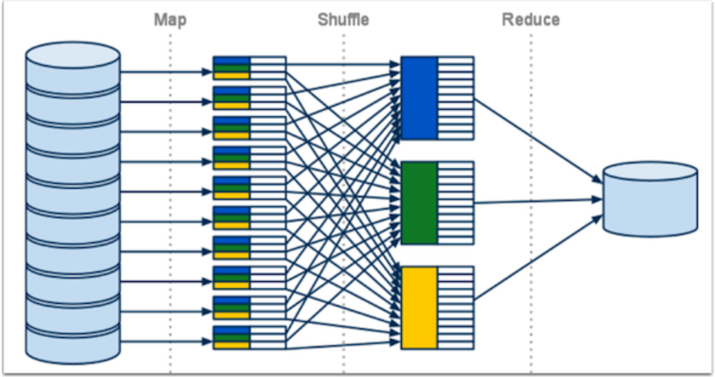
So ideally the beefiest computers should also have the largest HDFS storage capacity (and vice versa for weak computers)

The Bottleneck

The main bottleneck is transferring files across a network, so Hadoop keeps computing localized to same node as where the data block resides

So ideally the beefiest computers should also have the largest HDFS storage capacity (and vice versa for weak computers)

MapReduce



https://developers.google.com/appengine/docs/python/dataprocessing/overview

Streaming

- Easiest way to run a Hadoop job is by using/writing programs that process each line via standard Unix Stdin and Stdout
- Each line should be key+value pairs, separated by tab (by default)

```
hadoop  jar $HADOOP_HOME/hadoop-streaming.jar \
  -input myInputDirs \
  -output myOutputDir \
  -mapper /bin/cat \
  -reducer /bin/wc
```

For more details, see `https://hadoop.apache.org/common/docs/r1.0.3/streaming.html`

Streaming

- Easiest way to run a Hadoop job is by using/writing programs that process each line via standard Unix Stdin and Stdout
- Each line should be key+value pairs, separated by tab (by default)

```
hadoop  jar $HADOOP_HOME/hadoop-streaming.jar \
  -input myInputDirs \
  -output myOutputDir \
  -mapper /bin/cat \
  -reducer /bin/wc
```

For more details, see `https://hadoop.apache.org/common/docs/r1.0.3/streaming.html`

Interface to C++

- Hadoop's C++ interface is called *Pipes*
- Unlike the previous slide, it allows C++ code to communicate with Hadoop processes over sockets, instead of, well, pipes!

Important Methods

```
const std::string& getInputKey();
const std::string& getInputValue();
void emit(const std::string& key, const std::string& value);
```

Full docs:

```
https://hadoop.apache.org/common/docs/current/api/org/apache/
hadoop/mapred/pipes/Submitter.html
```

Interface to C++

- Hadoop's C++ interface is called *Pipes*
- Unlike the previous slide, it allows C++ code to communicate with Hadoop processes over sockets, instead of, well, pipes!

Important Methods

```
const std::string& getInputKey();
const std::string& getInputValue();
void emit(const std::string& key, const std::string& value);
```

Full docs:

https://hadoop.apache.org/common/docs/current/api/org/apache/hadoop/mapred/pipes/Submitter.html

Interface to C++

- Hadoop's C++ interface is called *Pipes*
- Unlike the previous slide, it allows C++ code to communicate with Hadoop processes over sockets, instead of, well, pipes!

Important Methods

```
const std::string& getInputKey();
const std::string& getInputValue();
void emit(const std::string& key, const std::string& value);
```

Full docs:

https://hadoop.apache.org/common/docs/current/api/org/apache/

hadoop/mapred/pipes/Submitter.html

Pig

- Pig is a very cool high-level programming language for Hadoop
- Unlike most high-level languages, it's statically-typed
- Unlike SQL, it's an imperative language
- Can be run interactively, or in batch mode
- Pig programs can be extended using User-Defined Functions in other languages
- Used by Twitter, Yahoo, LinkedIn, Nokia, WhitePages, AOL, Salesforce.com, etc.

Technically Pig is the platform and Pig Latin is the language

Pig Example

```
-- max_temp.pig: Finds the maximum temperature by year

records = LOAD ’input/sample.txt’
  AS (year:chararray, temperature:int, quality:int);

filtered_records = FILTER records BY temperature != 999 AND
  (quality == 0 OR quality == 1 OR quality == 4);

grouped_records = GROUP filtered_records BY year;

max_temp = FOREACH grouped_records GENERATE group,
  MAX(filtered_records.temperature);

DUMP max_temp;
```

From Tom White's *Hadoop: The Definitive Guide*, 2011

Hive

- Hive is a SQL-like declarative row-oriented, batch-oriented system for data analysis, querying, and summarization
- Originally written by Facebook, for its SQL-knowledgeable database folks
- Also used by Netflix, CNet, Digg, eHarmony, Last.fm, Scribd, ...
- Takes the cake for scariest mascot

Hive

- Hive is a SQL-like declarative row-oriented, batch-oriented system for data analysis, querying, and summarization
- Originally written by Facebook, for its SQL-knowledgeable database folks
- Also used by Netflix, CNet, Digg, eHarmony, Last.fm, Scribd, ...
- Takes the cake for scariest mascot

Hive Example

```
CREATE TABLE records
 (year STRING, temperature INT, quality INT)
 ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';

LOAD DATA LOCAL INPATH 'input/sample.txt'
 OVERWRITE INTO TABLE records;

SELECT year, MAX(temperature)
 FROM records
 WHERE temperature != 999
  AND (quality = 0 OR quality = 1 OR quality = 4)
 GROUP BY year;
```

From Tom White's *Hadoop: The Definitive Guide*, 2011

JVM API

Find Maximum Temperature by Year

```java
// NewMaxTemperature Application to find the maximum temperature in the
// weather dataset using the new context objects MapReduce API (from Tom White's book)
import java.io.IOException;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

// vv NewMaxTemperature
public class NewMaxTemperature {

  static class NewMaxTemperatureMapper
    /*[*/extends Mapper<LongWritable, Text, Text, IntWritable>/*]*/ {

    private static final int MISSING = 9999;

    public void map(LongWritable key, Text value, /*[*/Context context/*]*/)
        throws IOException, /*[*/InterruptedException/*]*/ {

      String line = value.toString();
      String year = line.substring(15, 19);
      int airTemperature;
      if (line.charAt(87) == '+') { // parseInt doesn't like leading plus signs
        airTemperature = Integer.parseInt(line.substring(88, 92));
      } else {
        airTemperature = Integer.parseInt(line.substring(87, 92));
      }
      String quality = line.substring(92, 93);
      if (airTemperature != MISSING && quality.matches("[01459]")) {
        /*[*/context.write/*]*/(new Text(year), new IntWritable(airTemperature));
      }
    }
  }

  static class NewMaxTemperatureReducer
    /*[*/extends Reducer<Text, IntWritable, Text, IntWritable>/*]*/ {

    public void reduce(Text key, /*[*/Iterable/*]*/<IntWritable> values,
        /*[*/Context context/*]*/)
        throws IOException, /*[*/InterruptedException/*]*/ {

      int maxValue = Integer.MIN_VALUE;
      for (IntWritable value : values) {
        maxValue = Math.max(maxValue, value.get());
      }
      /*[*/context.write/*]*/(key, new IntWritable(maxValue));
    }
  }

  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.err.println("Usage: NewMaxTemperature <input path> <output path>");
      System.exit(-1);
    }

    /*[*/Job job = new Job();
    job.setJarByClass(NewMaxTemperature.class);/*]*/

    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.setMapperClass(NewMaxTemperatureMapper.class);
    job.setReducerClass(NewMaxTemperatureReducer.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    /*[*/System.exit(job.waitForCompletion(true) ? 0 : 1);/*]*/
  }
}
```

From Tom White's *Hadoop: The Definitive Guide*, 2011

Mahout

Mahout is a scalable machine learning and data mining library

Classification

Logistic Regression, Naive Bayes, Complementary Naive Bayes, Support Vector Machines, Perceptron, Winnow, Neural Networks, Random Forests, Restricted Boltzmann Machines, Online Passive Aggressive (Crammer et al, 2006), Boosting, Hidden Markov Models

Clustering

Canopy Clustering, K-Means Clustering, Fuzzy K-Means, Expectation Maximization, Hierarchical Clustering, Dirichlet Process Clustering, Latent Dirichlet Allocation, Spectral Clustering, Minhash Clustering

Dimension reduction

Singular Value Decomposition, Stochastic Singular Value Decomposition with PCA workflow, Principal Components Analysis, Independent Component Analysis, Gaussian Discriminative Analysis

Regression

Locally Weighted Linear Regression

As well as Evolutionary Algorithms, Collaborative Filtering, and Vector Similarity algorithms

Some Hadoop NLP Tools & Resources

Chaski

Trains phrase-based MT models in a Hadoop cluster (Gao & Vogel, 2010)

SAMT on Hadoop

Venugopal & Zollmann (2009)

Mavuno

Hadoop-Based Text Mining Toolkit (from ISI), does POS tagging, chunking, parsing, NER, etc.

NLP Hadoop Book

Lin, Jimmy, and Chris Dyer. 2010. *Data-Intensive Text Processing with MapReduce*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

`http://lintool.github.com/MapReduceAlgorithms/MapReduce-book-final.pdf`

Processing Graphs

GraphLab

Originally developed for machine learning tasks. Written in C++. Probably what you want to use for graph-based NLP

GoldenOrb

Modeled after Google's Pregel architecture

Apache Giraph

Also modeled after Google's Pregel architecture, not as mature

Other Hadoop Projects

- Avro: A data serialization system.
- Cassandra: A scalable multi-master database with no single points of failure.
- Chukwa: A data collection system for managing large distributed systems. (I don't know what that means either)
- Dumbo: Write & run Hadoop programs in Python (or just use faster language)
- HBase (& Hypertable): Scalable, distributed column-oriented online databases that supports structured data storage for large tables.
- Hive: A data warehouse infrastructure that provides data summarization and ad hoc querying. Lies in between Pig & HBase
- **Mahout**: A Scalable machine learning and data mining library.
- **Pig**: A high-level data-flow language and execution framework for parallel computation.
- Weka-Hadoop: What it sounds like! (Nick Jenkin, 2009, COMP390-09A)
- YSmart: An SQL-to-MapReduce Translator (created by OSU people, now merged with Hive)
- ZooKeeper: A high-performance coordination service for distributed applications. (to go beyond just batch processing)

from `https://hadoop.apache.org`, and other sites

Current Status

- Status of Apache Hadoop Project
- Status of Cloudera Distribution
- Status of Department Cluster

Current Status

- Status of Apache Hadoop Project
- Status of Cloudera Distribution
- Status of Department Cluster

Current Status

- Status of Apache Hadoop Project
- Status of Cloudera Distribution
- Status of Department Cluster