# Hadoop and Eclipse

Eclipse Hawaii User's Group
May 26th, 2009

Seth Ladd
http://sethladd.com

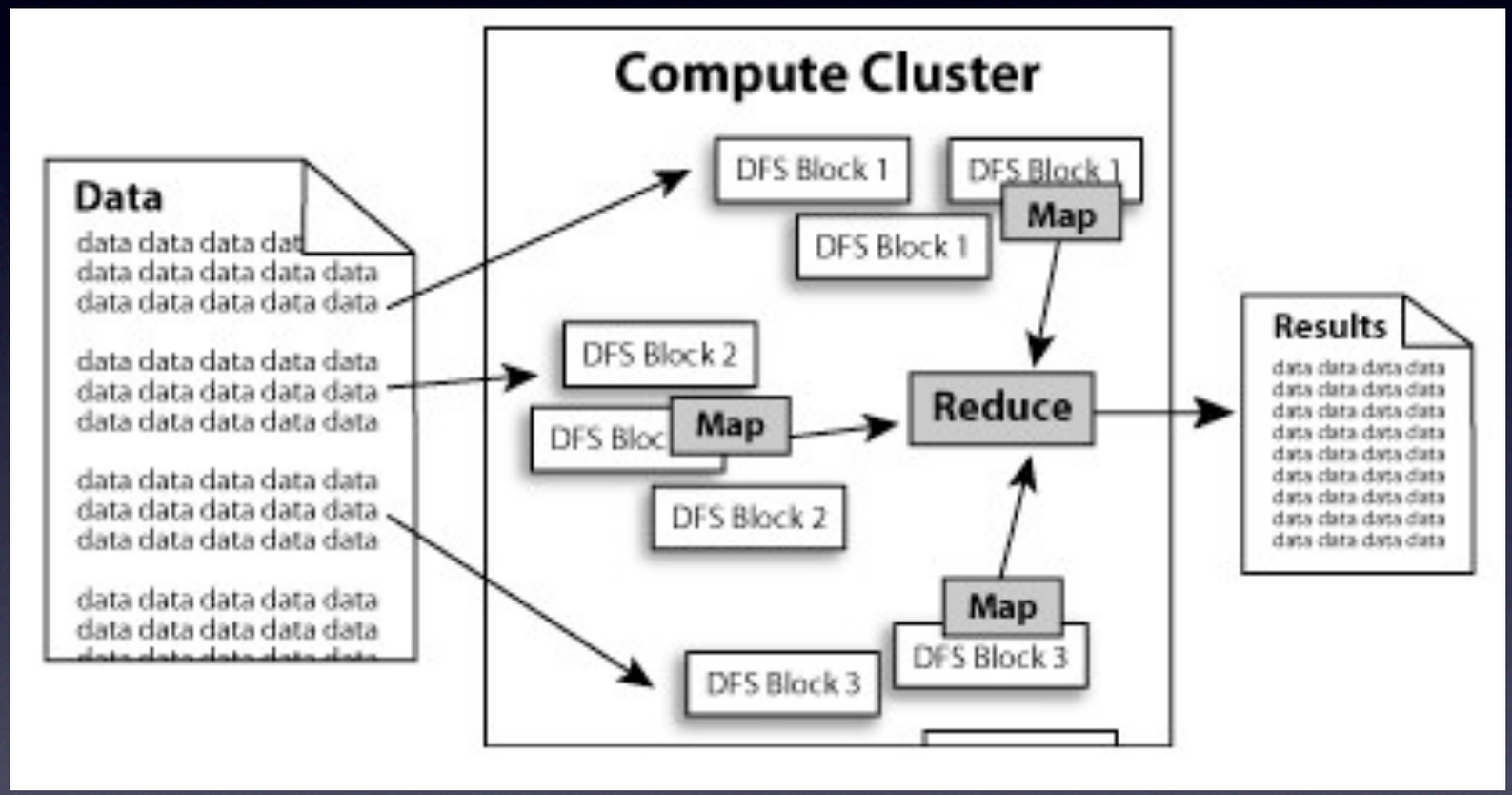
# Goal

- YOU can use the same technologies as The Big Boys
  - Google
  - Yahoo (2000 nodes)
  - Last.FM
  - AOL
  - Facebook (2.5 petabytes in hadoop, 15 TB inserted EACH DAY)
  - Hulu
  - Ning

# Problem

- A lot of data (> 1TB)
- parallel processing is hard
  - parallel algorithms
  - machine management
  - fault tolerance


hadoop


Compute Cluster
Data
data data data dat
data data data data data
data data data data data
data data data data data
data data data data data
data data data data data
data data data data data
data data data data data
data data data data data
data data data data data
DFS Block 1
DFS Block 1
Map
DFS Block 1
DFS Block 2
DFS Bloc
Map
Reduce
DFS Block 2
Map
DFS Block 3
DFS Block 3
Results
data data data data
data data data data
data data data data
data data data data
data data data data
data data data data
data data data data
data data data data

# Jim Gray's Sort Benchmark

| Bytes | Nodes | Maps | Reduces | Replication | Time |
|---|---|---|---|---|---|
| 500,000,000,000 | 1406 | 8000 | 2600 | 1 | 59 seconds |
| 1,000,000,000,000 | 1460 | 8000 | 2700 | 1 | 62 seconds |
| 100,000,000,000,000 | 3452 | 190,000 | 10,000 | 2 | 173 minutes |
| 1,000,000,000,000,000 | 3658 | 80,000 | 20,000 | 2 | 975 minutes |

100 byte records, 10 bytes key

Statistically, nodes were ALWAYS down during sort

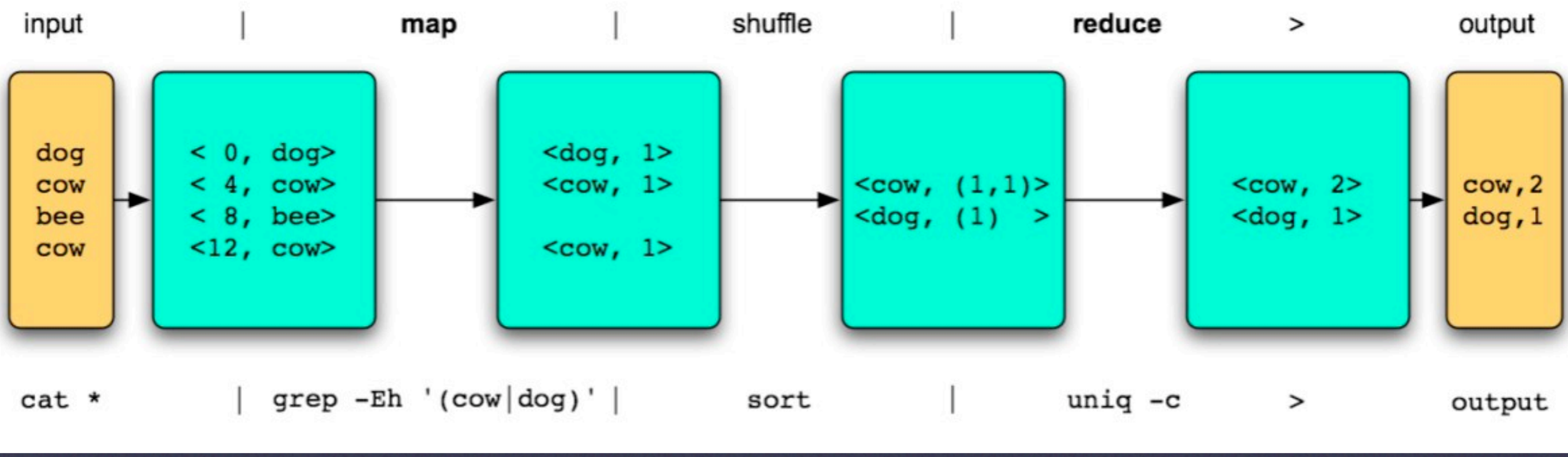
# Use Cases

- Calculating Page Rank - graph traversal
- Targeting Ads
- Analytics
- Machine Learning - clustering, recommendations
- Building indexes
- Aggregating Data

# What is Map Reduce NOT?

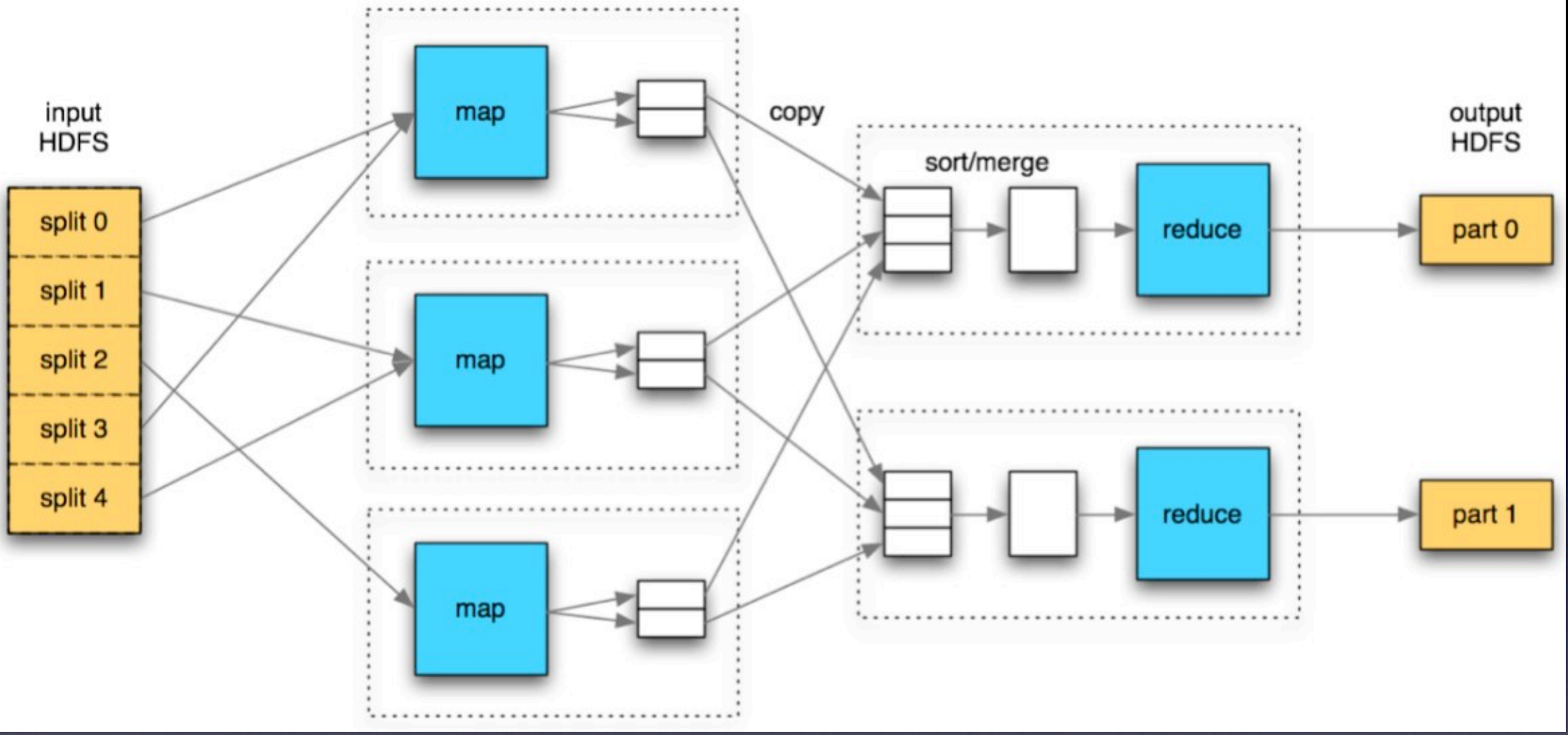
- NOT a relational database
- NOT for "immediate" answers
- NOT for small data sets
- NOT a new idea
  - though, packaging, integration, marketing are new

# What is MapReduce?

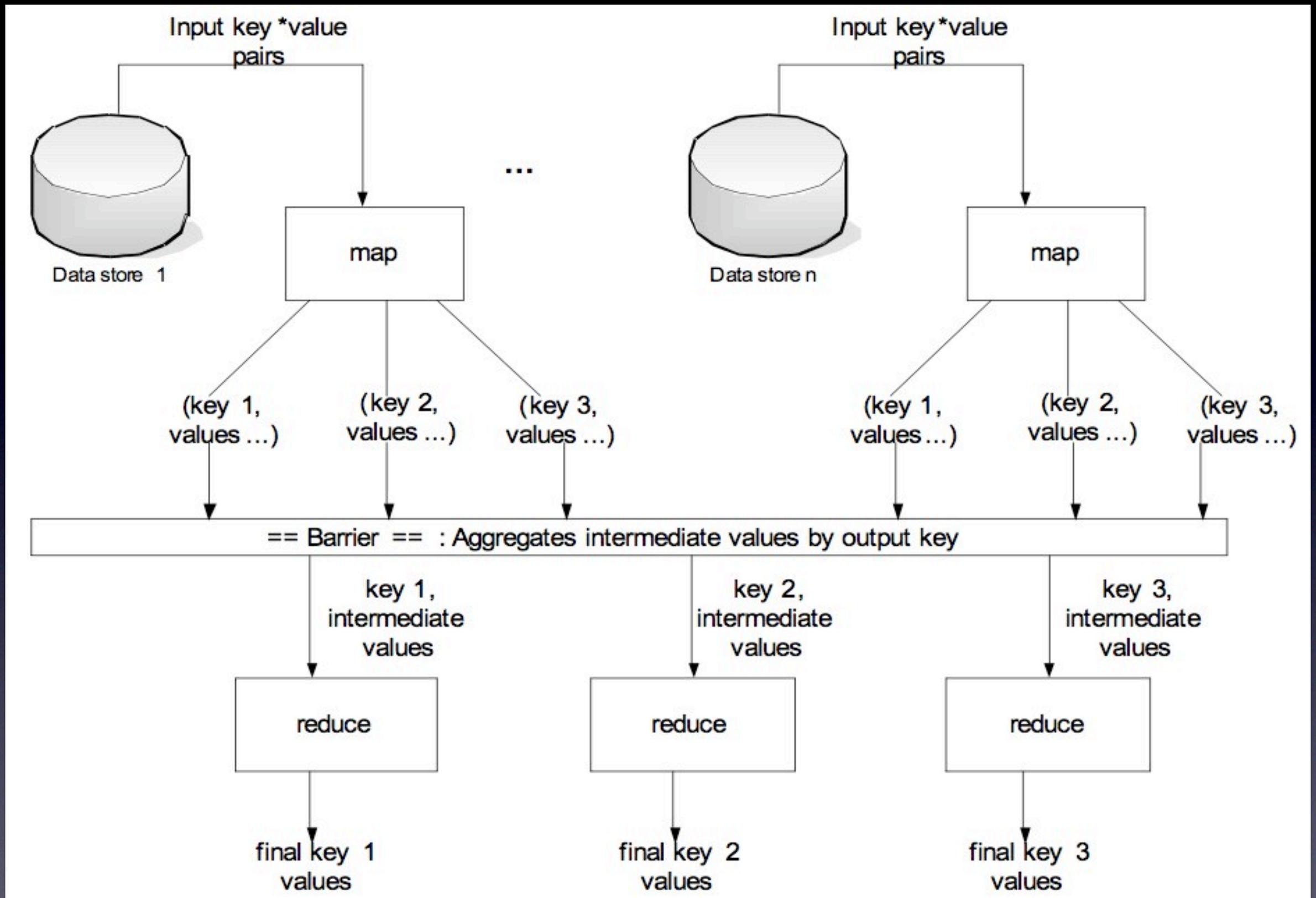



Thanks to Tom White, Buy his Book: Hadoop the Definitive Guide

# Another way of looking at it




Thanks to Tom White, Buy his Book: Hadoop the Definitive Guide




http://code.google.com/edu/submissions/ucsandiego-mapreduce/index.html

# Many Ways to Skin a Petacat

- Java - good for speed, existing libraries
- Pipes - C++
- Streaming - scripting languages
- Dumbo (python), mrtoolkit (ruby) - abstractions in scripting languages
- Pig, Hive - high level abstractions, nested data, joins

# Acknowledges Failure

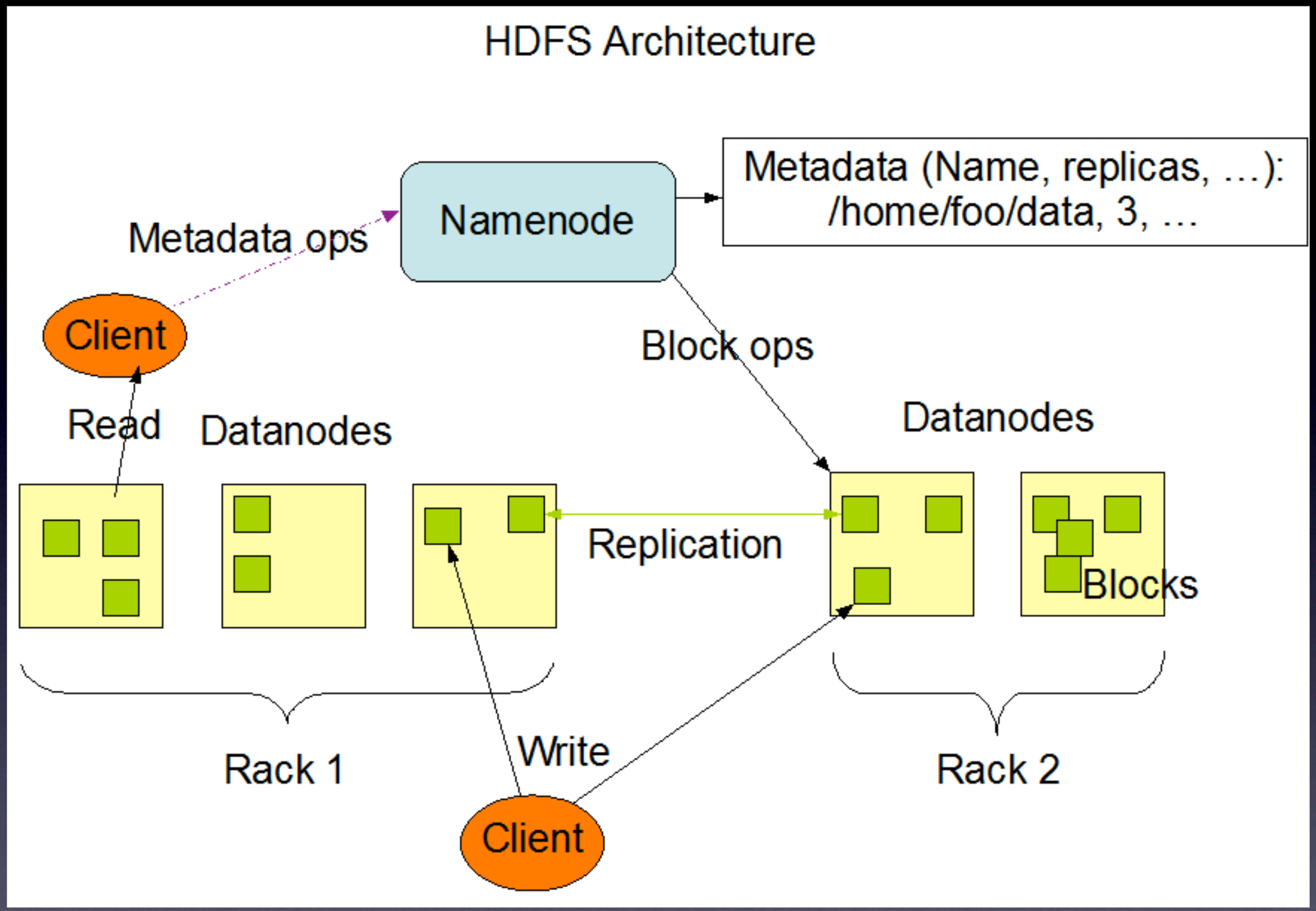
- Blocks copied to multiple nodes
  - usually three copies, one to another rack
- Jobs split into tasks
  - retry tasks

# Components

- Name Node - block lookup
- Secondary Name Node - backup
- Data Node - serves blocks
- Job Tracker - manage jobs
- Task Tracker - “worker”

# HDFS

- Hardware Failure is the Norm
- Streaming Data Access
- Large Data Sets
- Write Once, Read Many
- Moving Computations is Cheaper Than Moving Data
- Rack Aware Replication




http://hadoop.apache.org/core/docs/current/hdfs_design.html

# Unfortunately :(

- Eclipse Ganymede 3.4 does not work with Hadoop plugin
- http://issues.apache.org/jira/browse/HADOOP-3744 for patch

# Bring Up The Cluster

- start-all.sh - or individually
- NOTE: networking must be sane
- http://localhost:50070 - namenode
- http://localhost:50030 - jobtracker

# Interacting

- bin/hadoop
- job -?
- fs -?

# Pig

- High level language for data processing
- Compiles to Map Reduce programs
- So easy, you don't need Eclipse :)

# Word Count in Pig

```
files = LOAD 'Input' AS (line:chararray);
filtered = FILTER files BY line is not null;
words = FOREACH filtered GENERATE FLATTEN(TOKENIZE($0));
grouped = GROUP words BY $0;
counts = FOREACH grouped GENERATE group, COUNT($1);
dump counts;
```

Probably a better way to do this

# Word Count in Java

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.Writable;
import org.apache.hadoop.io.WritableComparable;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
```

```
public class WordCountMapper extends MapReduceBase
    implements Mapper<LongWritable, Text, Text, IntWritable> {

  private final IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(WritableComparable key, Writable value,
      OutputCollector output, Reporter reporter) throws IOException {

    String line = value.toString();
    StringTokenizer itr = new StringTokenizer(line.toLowerCase());
    while(itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      output.collect(word, one);
    }
  }
}
```

# More

```
import java.io.IOException;
import java.util.Iterator;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.WritableComparable;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class WordCountReducer extends MapReduceBase
    implements Reducer<Text, IntWritable, Text, IntWritable> {

  public void reduce(Text key, Iterator values,
      OutputCollector output, Reporter reporter) throws IOException {

    int sum = 0;
    while (values.hasNext()) {
      IntWritable value = (IntWritable) values.next();
      sum += value.get(); // process value
    }

    output.collect(key, new IntWritable(sum));
  }
}
```

# More

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;

public class WordCount {

  public static void main(String[] args) {
    JobClient client = new JobClient();
    JobConf conf = new JobConf(WordCount.class);

    // specify output types
    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);

    // specify input and output dirs
    FileInputPath.addInputPath(conf, new Path("input"));
    FileOutputPath.addOutputPath(conf, new Path("output"));

    // specify a mapper
    conf.setMapperClass(WordCountMapper.class);

    // specify a reducer
    conf.setReducerClass(WordCountReducer.class);
    conf.setCombinerClass(WordCountReducer.class);

    client.setConf(conf);
    try {
      JobClient.runJob(conf);
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

# PigPen

- Eclipse Plugin for Pig
- Buggy
- Syntax highlighting, run from Eclipse
- Avoid for now

# In Depth Example

- Calculate hierarchical populations across 4.38 Gigs of personnel data

# Training


Alan Gates, Yahoo, Map Reduce and Pig

http://alohaonrails.com

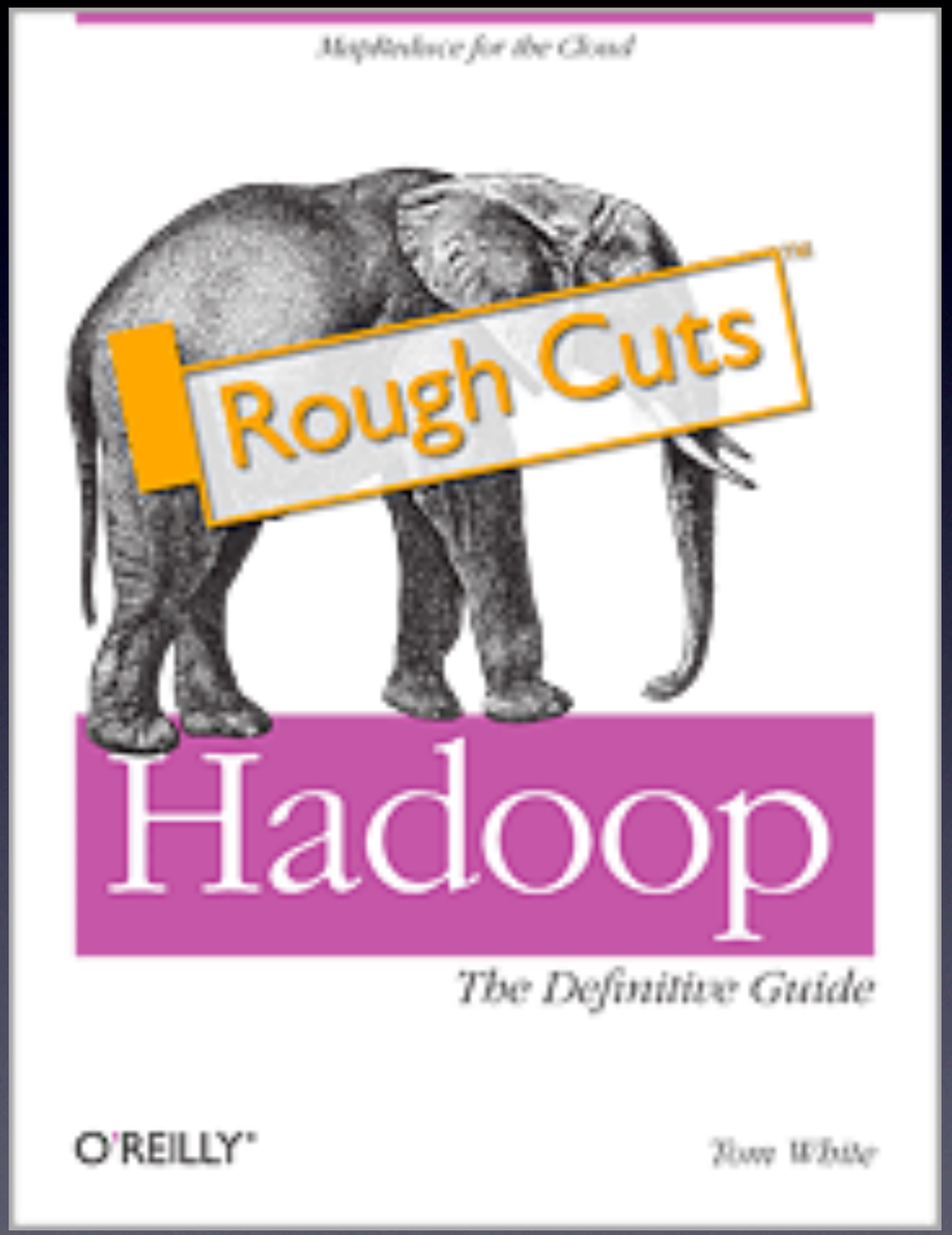
Amazon Webinar: Getting Started with Amazon Elastic MapReduce

http://is.gd/EoWO

# More Training

- Cloudera Training Videos
- Google's Map Reduce Videos on YouTube
  - Google Scalability Conference videos
- Lots of courseware
  - MapReduce in a Week
  - Google Code University
- Hadoop Book
- Many Resources!
- http://delicious.com/sethladd/hadoop

# The Book


MapReduce for the Cloud
Rough Cuts
Hadoop
The Definitive Guide
O'REILLY
Tom White

# The Companies Behind Hadoop

YAHOO!®

# How To Proceed

- Download Hadoop 0.18.3
- Download Pig 0.2.0
- Or... Use Cloudera's RPMs
- Or... Use Cloudera's Scripts for EC2
- Or... Use Amazon's Elastic MapReduce

# If Not Hadoop

- Greenplum
- Aster Data
- Teradata
- Petabyte will be the new Gigabyte very soon!

# Take Away

- Data will continue to grow
- Change your mindset
- Tools and techniques are here NOW
- Hadoop is a rich and exciting place to be