What We Can Do in the Cloud (2)

-Tutorial for Cloud Computing Course-

Mikael Fernandus Simalango

WISE Research Lab

Ajou University, South Korea

Overview

- Riding Google App Engine
- Taming Hadoop
- Summary

Google App Engine

Riding Google App Engine

WISE

Riding Google App Engine

- Google App Engine in a nutshell:
  - A platform to create and deploy web applications on Google's infrastructure
    - An example of PaaS
  - Free and scalable (to certain extent)
  - Apps should be written in Java or Python
- For first-timers:
  - http://appengine.google.com

Riding Google App Engine (cont'd)

- **Quickstart:**
  1. Create a Google account (if you don't have one)
  2. Login to appengine
  3. Verify your account by SMS

Google app engine

Verify Your Account by SMS

To create applications with Google App Engine, you need a verification code. Select the country and carrier number. The verification code will be sent to it via SMS. Note you will only need to verify your account once.

Country and Carrier:

Other (Not Listed)

If your country and carrier are not on the list, select Other (Not Listed). What carriers are supported?

Mobile Number:

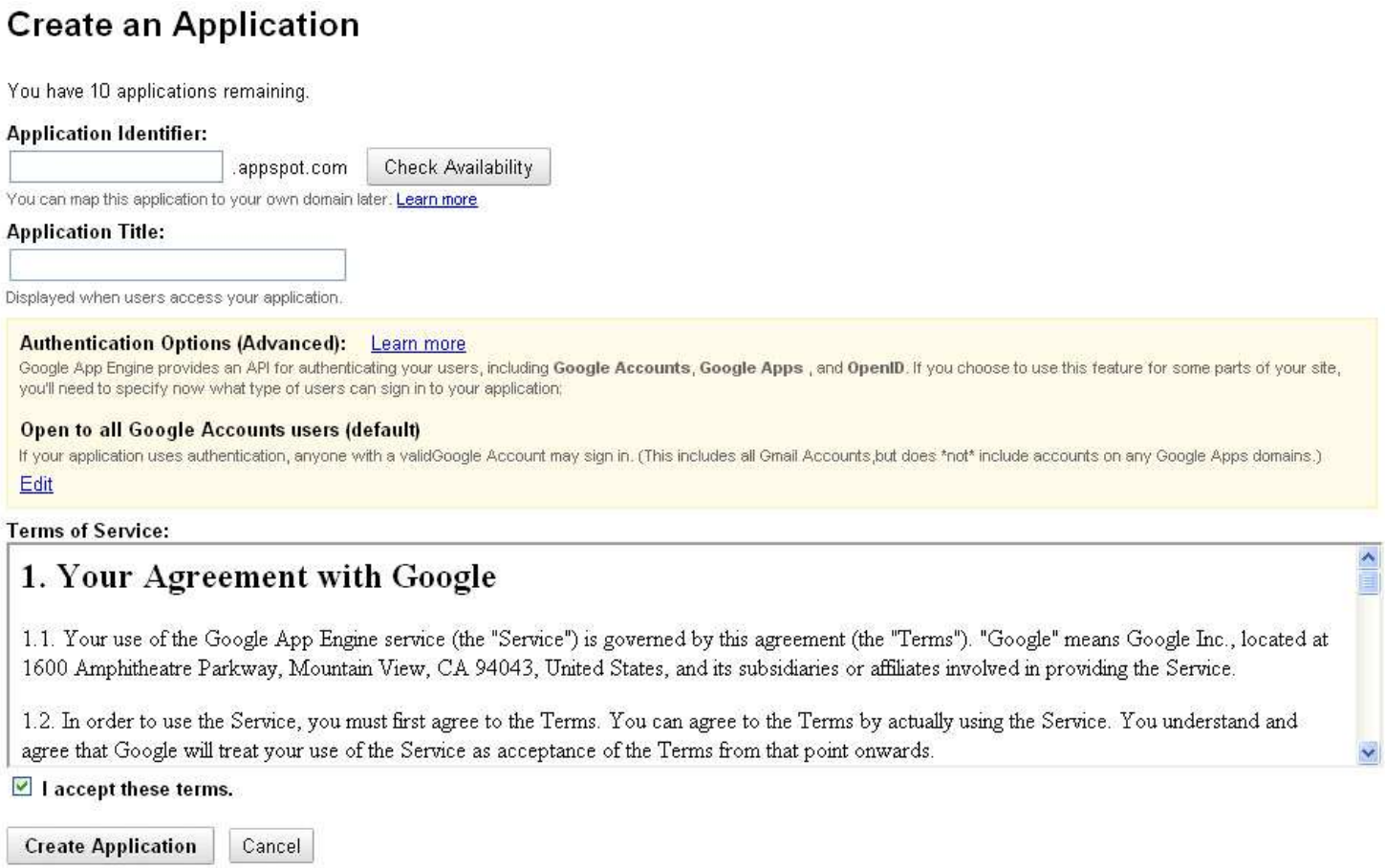
Include your country code and full phone number. eg. +1 650 555 1212

Send

Riding Google App Engine (cont'd)

- **Quickstart (cont'd):**

  4. Pick a name for your application



Riding Google App Engine (cont'd)

- **Quickstart (cont'd):**

  5. Read the documentation to grab more info about how-to

     Url: http://code.google.com/appengine/docs/

     Also check about Java compatibility in GAE

     Url: http://groups.google.com/group/google-appengine-java/web/will-it-play-in-app-engine?pli=1

  6. We are good to go. Let's start our first Google App Engine-based application.

We will use Java and Eclipse IDE in this showcase

Riding Google App Engine (cont'd)

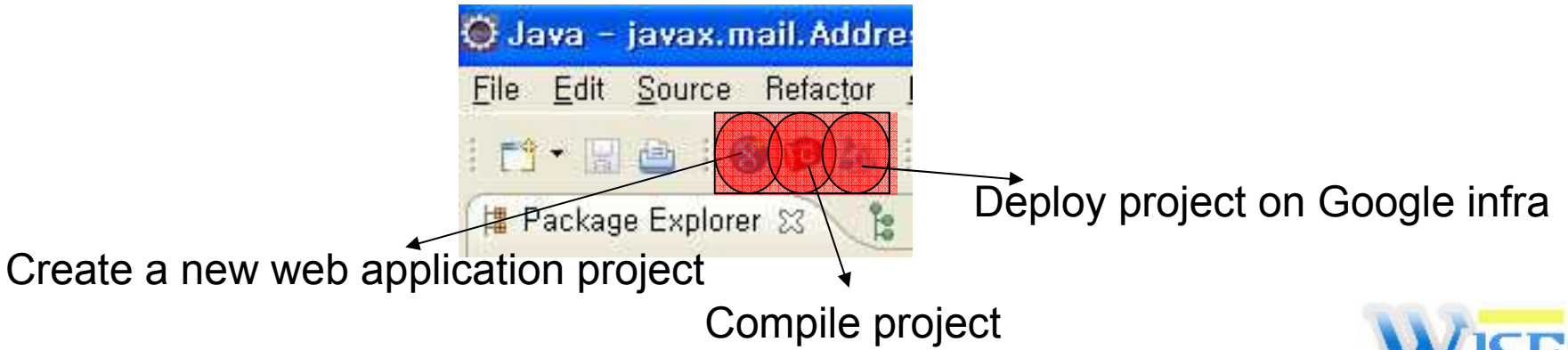
- Development stage:
  - Step 1a: Download and install JDK 1.6 from Sun's website (if you haven't done so)
  - Step 1b: Download and install Eclipse (if you haven't done so)
    - Eclipse 3.5 Galileo is available at WISE FTP server (ftp://wise.ajou.ac.kr)

Riding Google App Engine (cont'd)

- Development stage:
  - Step 2: Read this tutorial about developing in Java for GAE
    - Part 1: http://www.ibm.com/developerworks/java/library/j-gaej1/
    - Part 2: http://www.ibm.com/developerworks/java/library/j-gaej2/
    - Part 3: http://www.ibm.com/developerworks/java/library/j-gaej3.html

Riding Google App Engine (cont'd)

- Development stage:
  - Step 3: Download GAE plugin for Eclipse
    - Click **Help** > **Install New Software** in Eclipse
    - Input url: http://dl.google.com/eclipse/plugin/3.5
    - Check the option to download the SDK
    - After the plugin is downloaded, you will see these icons in the menu bar



Riding Google App Engine (cont'd)

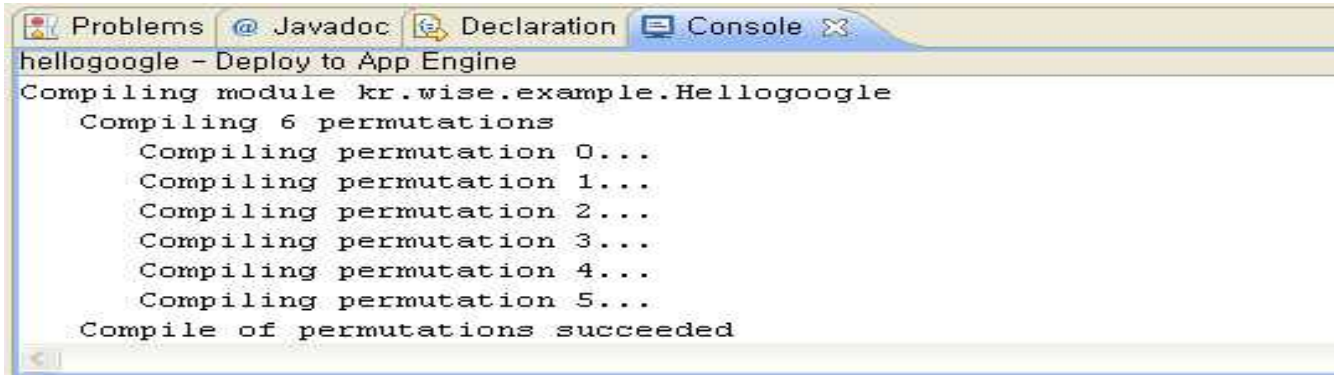
- Development stage:
  - Step 4a: Create a GAE project

Riding Google App Engine (cont'd)

- Development stage:
  - Step 4b: Observe the packages and files created in the project tree

Riding Google App Engine (cont'd)

- Development stage:
  - Step 5: Modify the default code and deploy on Google



Riding Google App Engine (cont'd)

- Development stage:
  - Step 6: Take a look at the deployed web application by visiting the application url

Riding Google App Engine (cont'd)

- Development stage:
  - Step 7: Try to deploy more advanced examples provided in the tutorial
  - Step 8: Plan the application you will build for your project:
    - It should "exploit" the features associated with a cloud application service
      - Scalability
      - High availability
      - Economies of scale (if you go with the Google App Engine for Business)
    - Recommendation:
      - Exploit Google data storage functionality using JDO (Java Data Objects) or JPA (Java Persistence API)

Taming Hadoop

Taming Hadoop

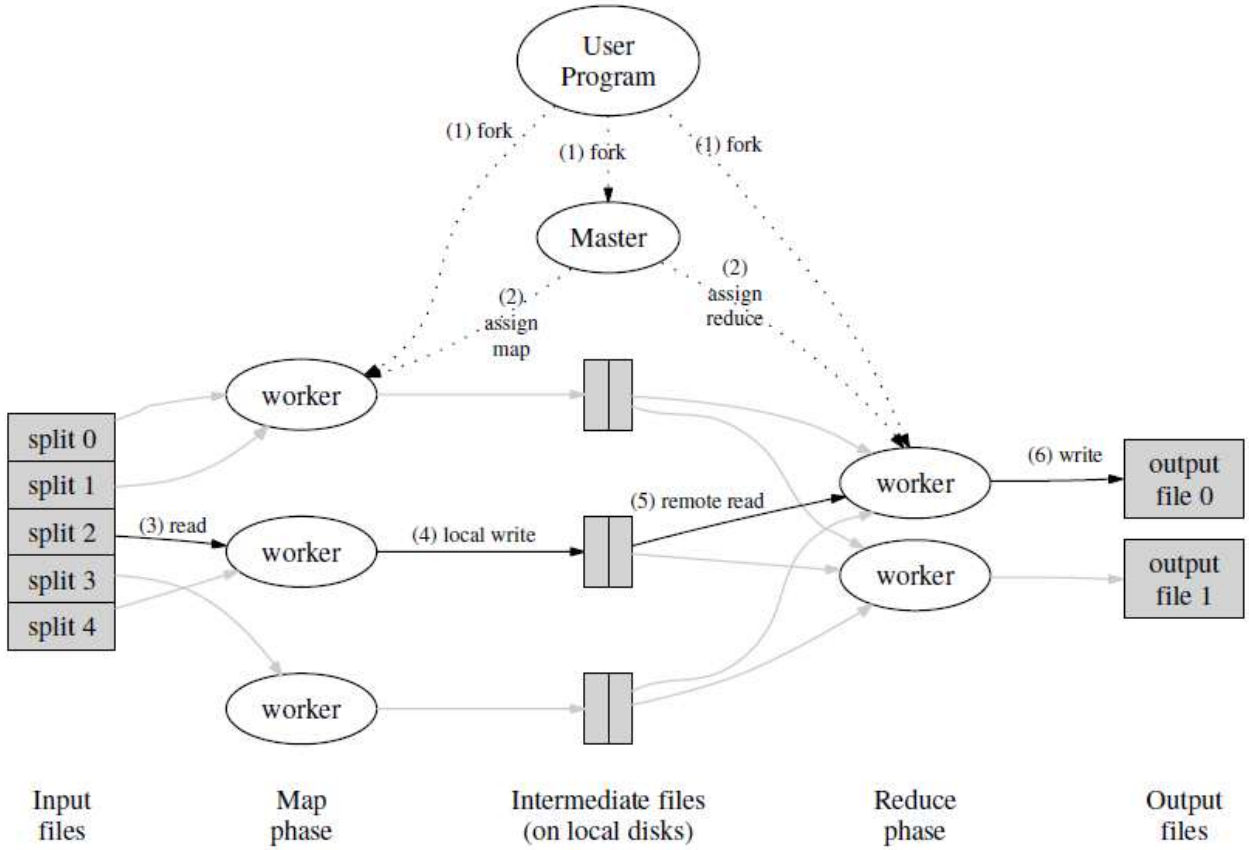
- Hadoop in brief:
  - Platform consisting of a collection of open-source software for reliable, scalable, distributed computing
  - Initially an Apache project (http://hadoop.apache.org) but several forks or distros are now available (e.g.: Cloudera Hadoop)
  - Hadoop != MapReduce but MapReduce⊂ Hadoop

Taming Hadoop (cont'd)

- Subprojects (or subplatforms) in Hadoop:
  - **Hadoop common**: common utilities that support other subplatforms
  - Chukwa: data collection system for managing large distributed systems
  - **HBase**: scalable, distributed database supporting big tables
  - **HDFS**: primary storage system used by Hadoop applications
  - **Hive**: data warehouse infrastructure that provides data summarization and SQL-like ad hoc querying
  - **MapReduce**: implementation of Google's MapReduce programming model
  - **Pig**: high-level data-flow language and execution framework for parallel computation
  - ZooKeeper: high-performance coordination service for distributed applications

Taming Hadoop (cont'd)

- MapReduce programming model*:

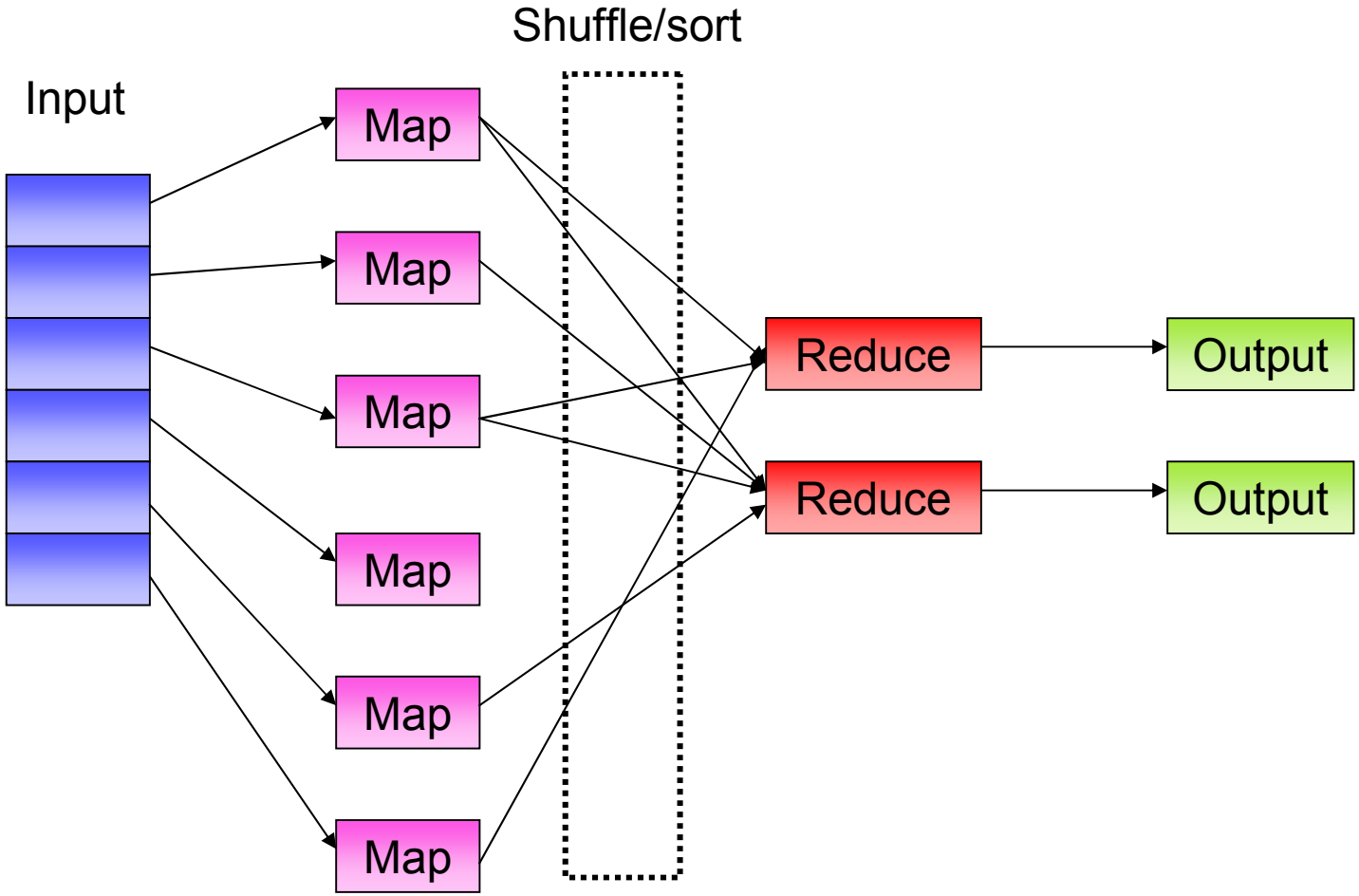


(*)J. Dean, S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Proc. of 6th OSDI, 2004

Taming Hadoop (cont'd)

- Two phases in MapReduce:
  - Map: user provides a map function that processes a key/value pairs and output intermediate key/value pairs
  - Reduce: a (user-supplied) function merges all intermediate values associated with the same intermediate key

Taming Hadoop (cont'd)



Taming Hadoop (cont'd)

- How Hadoop adopts MapReduce model:

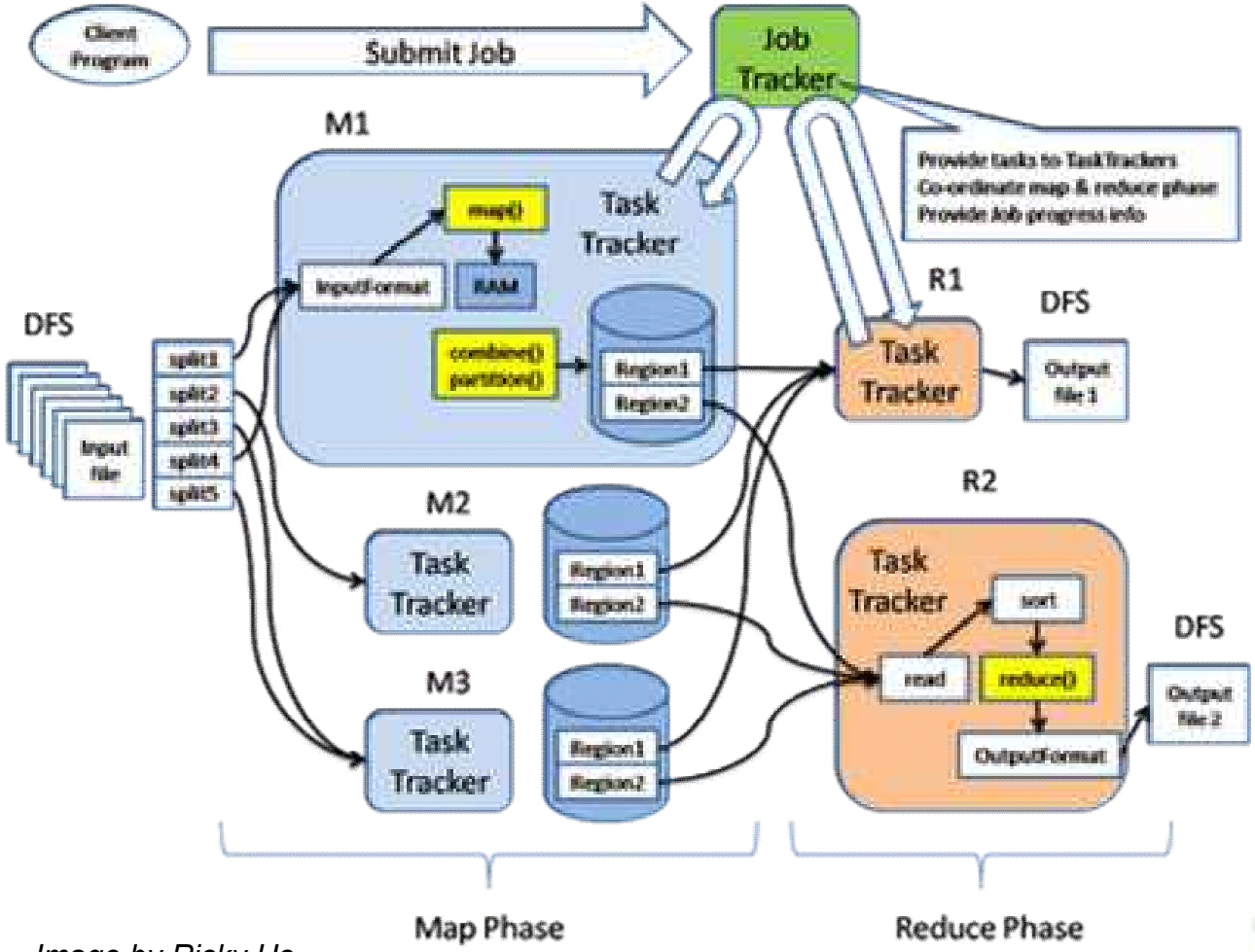


Image by Ricky Ho

Taming Hadoop (cont'd)

- Some terms in Hadoop MapReduce (MR):
  - **Job**: all jar files (or classes) which contains a MR program
  - **JobTracker**: a Hadoop service that controls the distribution of MapReduce tasks to nodes in the cluster
  - **Task**: program that executes individual map and reduce
  - **TaskTracker**: a Hadoop service that starts and tracks MR task in a networked environment. It communicates with JobTracker for task assignment and reporting results
  - **HDFS**: Hadoop Distributed File System
  - **NameNode**: directory namespace manager and "inode table" for HDFS
  - **DataNode**: a class (and program) in a cluster node that stores a set of blocks for DFS deployment

Taming Hadoop (cont'd)

- Hadoop MapReduce sample applications:
  - WordCount
  - LogAnalyzer
  - Protein sequence analyzer

Taming Hadoop (cont'd)

- WordCount application:
  - We provide input files and Hadoop will output words found along with its recurrence
  - We use two samples:
    - WordCount with MapReduce java library
    - WordCount without MapReduce java library
      - We specify the Mapper and Reducer functions
      - Execute the job with HadoopStreaming

Taming Hadoop (cont'd)

- Step 1: copy input files to HDFS

```
:~/cloudcomp/hadoop/java_samples
[          java_samples]$ $HADOOP_HOME/bin/hadoop dfs -copyFromLocal /home/mikael/cloudcomp/hadoop/gutenberg-books /user/mikael/wordcount/input
[          java_samples]$ $HADOOP_HOME/bin/hadoop dfs -ls /user/mikael/wordcount/input
Found 1 items
drwxr-xr-x   - mikael wheel          0 2010-09-16 14:34 /user/mikael/wordcount/input/gutenberg-books
[          java_samples]$ $HADOOP_HOME/bin/hadoop dfs -ls /user/mikael/wordcount/input/gutenberg-books
Found 8 items
-rw-r--r--   4 mikael wheel     343694 2010-09-16 14:34 /user/mikael/wordcount/input/gutenberg-books/art_of_war.txt
-rw-r--r--   4 mikael wheel    2686016 2010-09-16 14:34 /user/mikael/wordcount/input/gutenberg-books/count_of_monte_cristo.txt
-rw-r--r--   4 mikael wheel    1391706 2010-09-16 14:34 /user/mikael/wordcount/input/gutenberg-books/da_vinci.txt
-rw-r--r--   4 mikael wheel     305851 2010-09-16 14:34 /user/mikael/wordcount/input/gutenberg-books/machiavelli.txt
-rw-r--r--   4 mikael wheel     674762 2010-09-16 14:34 /user/mikael/wordcount/input/gutenberg-books/outline_of_science.txt
-rw-r--r--   4 mikael wheel     704169 2010-09-16 14:34 /user/mikael/wordcount/input/gutenberg-books/pride_and_prejudice.txt
-rw-r--r--   4 mikael wheel    1238989 2010-09-16 14:34 /user/mikael/wordcount/input/gutenberg-books/the_republic.txt
-rw-r--r--   4 mikael wheel    1573044 2010-09-16 14:34 /user/mikael/wordcount/input/gutenberg-books/ulysses.txt
[          java_samples]$
```

Taming Hadoop (cont'd)

- Step 2: Hadoop MapReduce with native library

```
:~/cloudcomp/hadoop/java_samples
[           java_samples]$ $HADOOP_HOME/bin/hadoop jar wordcount.jar kr.wise.ex
ample.wordcount.WordCount /user/mikael/wordcount/input/gutenberg-books /user/mik
ael/wordcount/outputc
10/09/16 20:57:07 INFO input.FileInputFormat: Total input paths to process : 8
10/09/16 20:57:07 INFO mapred.JobClient: Running job: job_201008150343_0006
10/09/16 20:57:08 INFO mapred.JobClient:  map 0% reduce 0%
10/09/16 20:57:19 INFO mapred.JobClient:  map 62% reduce 0%
10/09/16 20:57:22 INFO mapred.JobClient:  map 100% reduce 0%
10/09/16 20:57:31 INFO mapred.JobClient:  map 100% reduce 100%
10/09/16 20:57:33 INFO mapred.JobClient: Job complete: job_201008150343_0006
10/09/16 20:57:33 INFO mapred.JobClient: Counters: 17
10/09/16 20:57:33 INFO mapred.JobClient:   Job Counters
10/09/16 20:57:33 INFO mapred.JobClient:     Launched reduce tasks=1
10/09/16 20:57:33 INFO mapred.JobClient:     Launched map tasks=8
10/09/16 20:57:33 INFO mapred.JobClient:     Data-local map tasks=8
10/09/16 20:57:33 INFO mapred.JobClient:   FileSystemCounters
10/09/16 20:57:33 INFO mapred.JobClient:     FILE_BYTES_READ=4559500
10/09/16 20:57:33 INFO mapred.JobClient:     HDFS_BYTES_READ=8918231
10/09/16 20:57:33 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=7604484
10/09/16 20:57:33 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=1353231
10/09/16 20:57:33 INFO mapred.JobClient:   Map-Reduce Framework
10/09/16 20:57:33 INFO mapred.JobClient:     Reduce input groups=121393
10/09/16 20:57:33 INFO mapred.JobClient:     Combine output records=208560
10/09/16 20:57:33 INFO mapred.JobClient:     Map input records=182642
```

Taming Hadoop (cont'd)

- Step 3: Hadoop MapReduce with user-supplied Mapper and Reducer functions

```
:~/cloudcomp/hadoop/python_samples
[            python_samples]$ ll
total 8
-rwxr-xr-x. 1 mikael wheel 228 Sep 15 21:23 mapper_simple.py
-rwxr-xr-x. 1 mikael wheel 576 Sep 15 21:32 reducer_simple.py
[            python_samples]$ $HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/contrib/s
treaming/hadoop-0.20.1+169.89-streaming.jar -file mapper_simple.py -mapper mappe
r_simple.py -file reducer_simple.py -reducer reducer_simple.py -input /user/mika
el/wordcount/input/gutenberg-books/* -output /user/mikael/wordcount/outputp
packageJobJar: [mapper_simple.py, reducer_simple.py, /var/lib/hadoop-0.20/cache/
mikael/hadoop-unjar4521896047048909333/] [] /tmp/streamjob4982408366407420152.ja
r tmpDir=null
10/09/16 21:08:17 INFO mapred.FileInputFormat: Total input paths to process : 8
10/09/16 21:08:17 INFO streaming.StreamJob: getLocalDirs(): [/var/lib/hadoop-0.2
0/cache/mikael/mapred/local]
10/09/16 21:08:17 INFO streaming.StreamJob: Running job: job_201008150343_0007
10/09/16 21:08:17 INFO streaming.StreamJob: To kill this job, run:
10/09/16 21:08:17 INFO streaming.StreamJob: /usr/lib/hadoop-0.20//bin/hadoop job
  -Dmapred.job.tracker=magi.ajou.ac.kr:8021 -kill job_201008150343_0007
10/09/16 21:08:17 INFO streaming.StreamJob: Tracking URL: http://localhost.local
domain:50030/jobdetails.jsp?jobid=job_201008150343_0007
10/09/16 21:08:18 INFO streaming.StreamJob:  map 0%  reduce 0%
10/09/16 21:08:28 INFO streaming.StreamJob:  map 88%  reduce 0%
10/09/16 21:08:31 INFO streaming.StreamJob:  map 100%  reduce 0%
10/09/16 21:08:37 INFO streaming.StreamJob:  map 100%  reduce 29%
10/09/16 21:08:40 INFO streaming.StreamJob:  map 100%  reduce 74%
10/09/16 21:08:43 INFO streaming.StreamJob:  map 100%  reduce 91%
10/09/16 21:08:49 INFO streaming.StreamJob:  map 100%  reduce 100%
```

Summary

- It is necessary to plan your project well
  - Problem to attack
  - Project complexity
  - Resource availability (knowledge, hands-on experience, time, infrastructure, etc)
- You can build your project by using infrastructure and platform on WISE or other public ones
  - Eucalyptus Community Cloud
  - Amazon Web Services
  - Google App Engine
  - Github
  - etc

THANK YOU