Karlsruhe Institute of Technology

Cloud Computing: MapReduce and Hadoop

June 2010

Marcel Kunze, Research Group Cloud Computing

KIT – University of the State of Baden-Württemberg and
National Laboratory of the Helmholtz Association

www.kit.edu

Motivation: Large Scale Data Processing

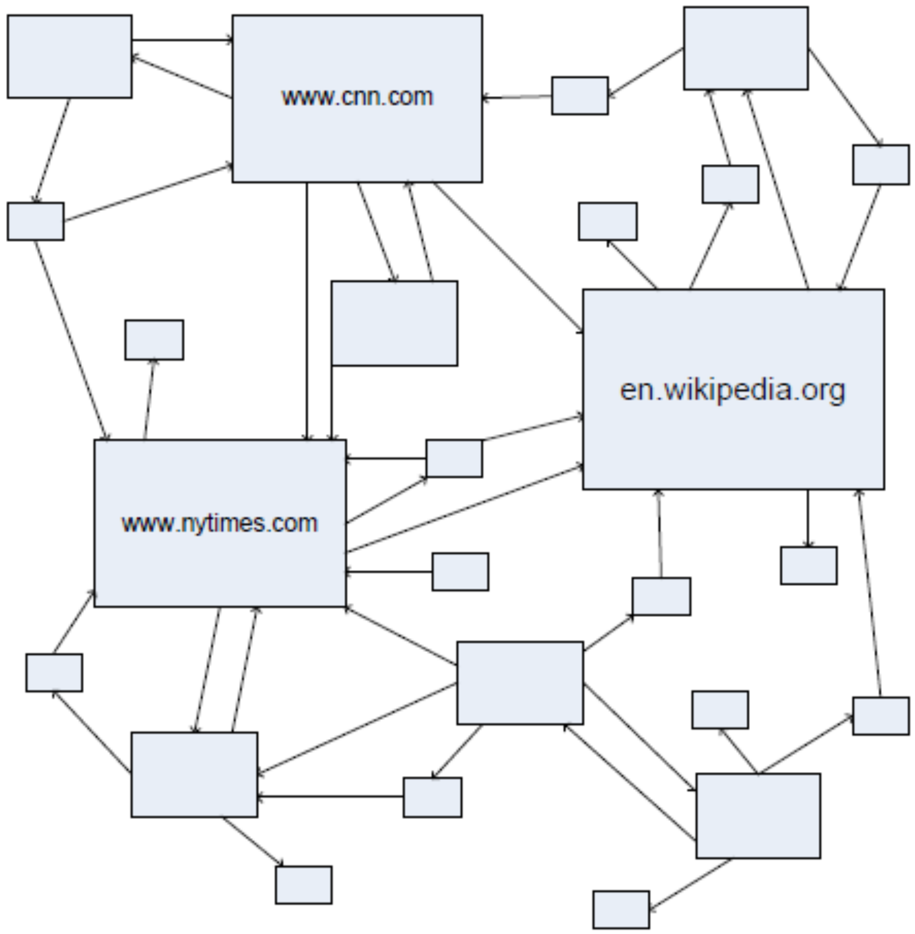
- MapReduce: Algorithm for large scale parallel data processing
  - Want to process lots of data (> 1 TB)
  - Want to parallelize across hundreds/thousands of CPUs
  - … Want to make this easy

- Potential fields of application
  - Web indexing
  - Data mining
  - Log file analysis
  - Machine learning
  - Scientific simulation
  - Bioinformatics research

- Google uses MapReduce everywhere, e.g. to run PageRank

Google Search: PageRank Algorithm

- Find the most popular page for a specific set of key words
- If a user starts at a random web page and surfs by clicking links and randomly entering new URLs, what is the probability that s/he will arrive at a given page?
- The PageRank of a page captures this notion
  - More “popular” or “worthwhile” pages get a higher rank



Source: A.Kimball

PageRank: Formula

- Given page A, and pages T_1 through T_n linking to A, PageRank is defined as:

$$PR(A) = (1-d) + d\ (PR(T_1)/C(T_1) + ... + PR(T_n)/C(T_n))$$

C(P) is the cardinality (out-degree) of page P

d is the damping (“random URL”) factor

Source: A.Kimball

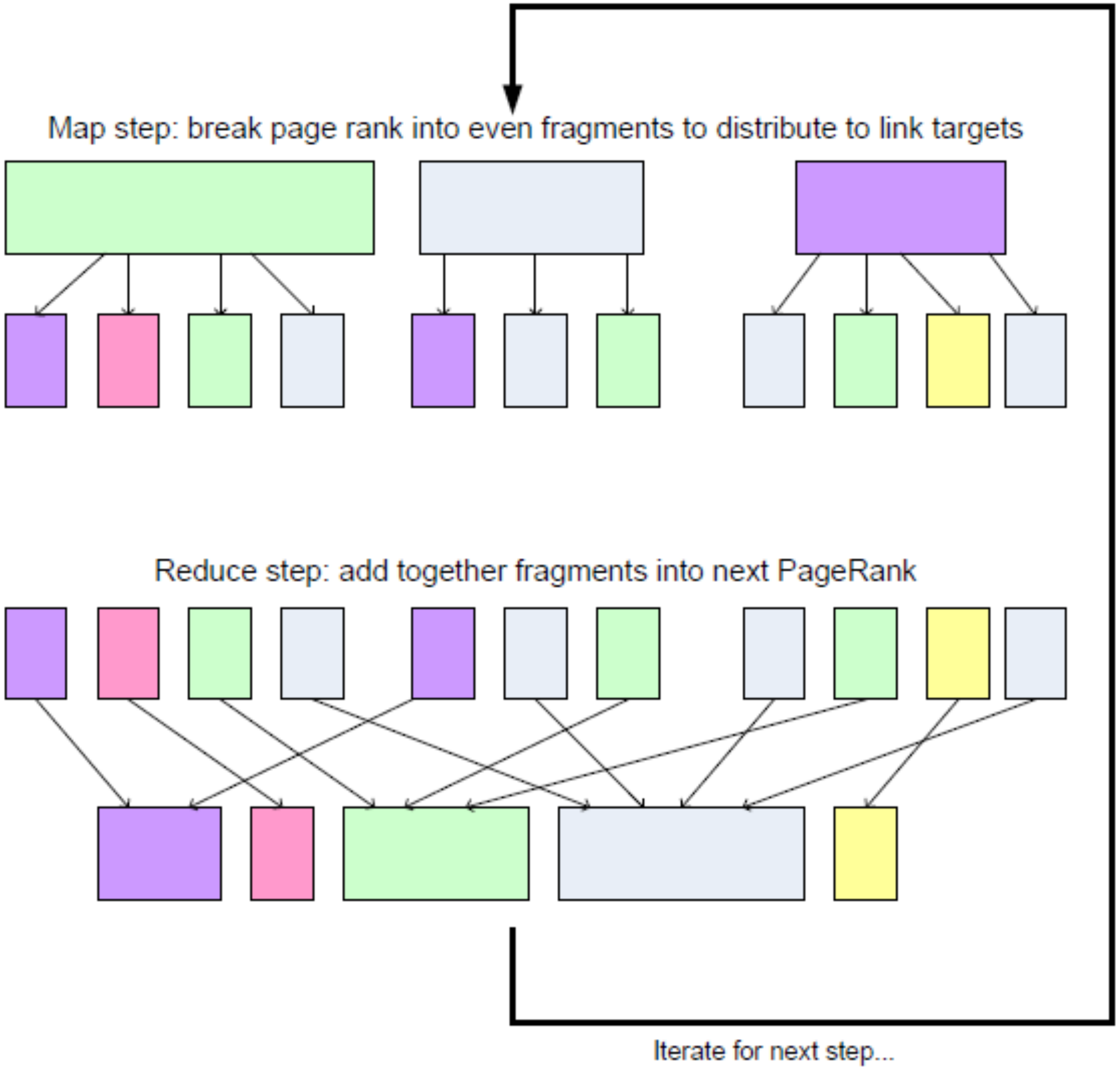
PageRank: Intuition

- Calculation is iterative: PR_{i+1} is based on PR_i
- Each page distributes its PR_i to all pages it links to. Linkees add up their awarded rank fragments to find their PR_{i+1}
- d is a tunable parameter (usually = 0.85) encapsulating the "random jump factor"

$$PR(A) = (1-d) + d\,(PR(T_1)/C(T_1) + \ldots + PR(T_n)/C(T_n))$$

Source: A.Kimball

PageRank: Iteration



Map step: break page rank into even fragments to distribute to link targets

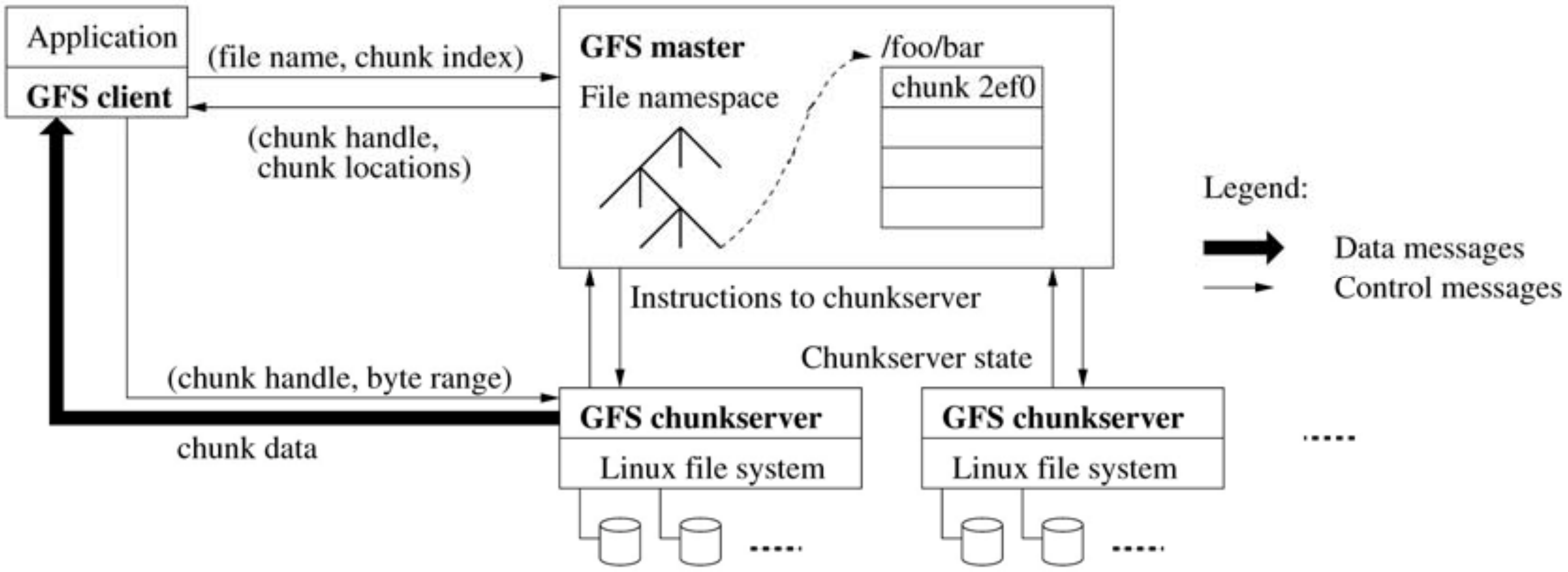
Reduce step: add together fragments into next PageRank

Iterate for next step...

Source: A.Kimball

Google File System (GFS)

- A GFS cluster has one master and many chunkservers
- Files are divided into 64 MB chunks
- Chunks are replicated and stored in the Unix file systems of the chunkservers
- The master holds metadata
- Clients get metadata from the master, and data directly from chunkservers



MapReduce

- Jeffrey Dean and Sanjay Ghemawat
  MapReduce: Simplified Data Processing on Large Clusters
  OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004.
  see http://labs.google.com/papers/mapreduce.html

- Google Tutorial at
  http://code.google.com/intl/de-DE/edu/submissions/mapreduce-minilecture/listing.html

Functional Programming Review

- Functional operations do not modify data structures: They always create new ones
- Original data still exists in unmodified form
- Data flows are implicit in program design
- Order of operations does not matter

- History: LISP programming

Functional Programming Review

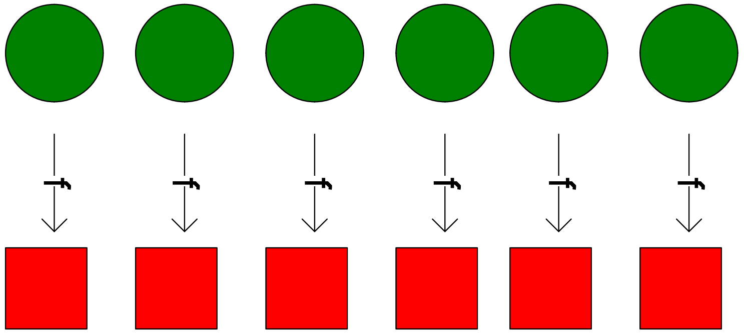
```
fun foo(l: int list) =
  sum(l) + mul(l) + length(l)
```

Order of sum() and mul(), etc does not matter – they do not modify l

Map

map f lst: (’a->’b) -> (’a list) -> (’b list)

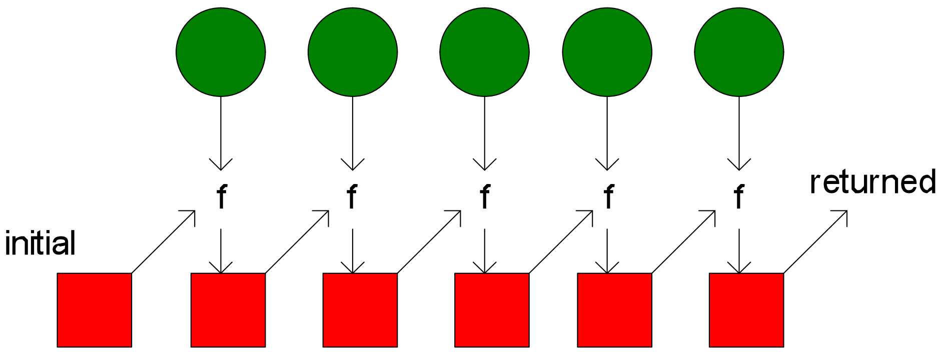
Creates a new list by applying f to each element of the input list; returns output in order.



Fold

fold f x_0 lst: ('a*'b->'b)->'b->('a list)->'b

Moves across a list, applying *f* to each element plus an *accumulator*. f returns the next accumulator value, which is combined with the next element of the list

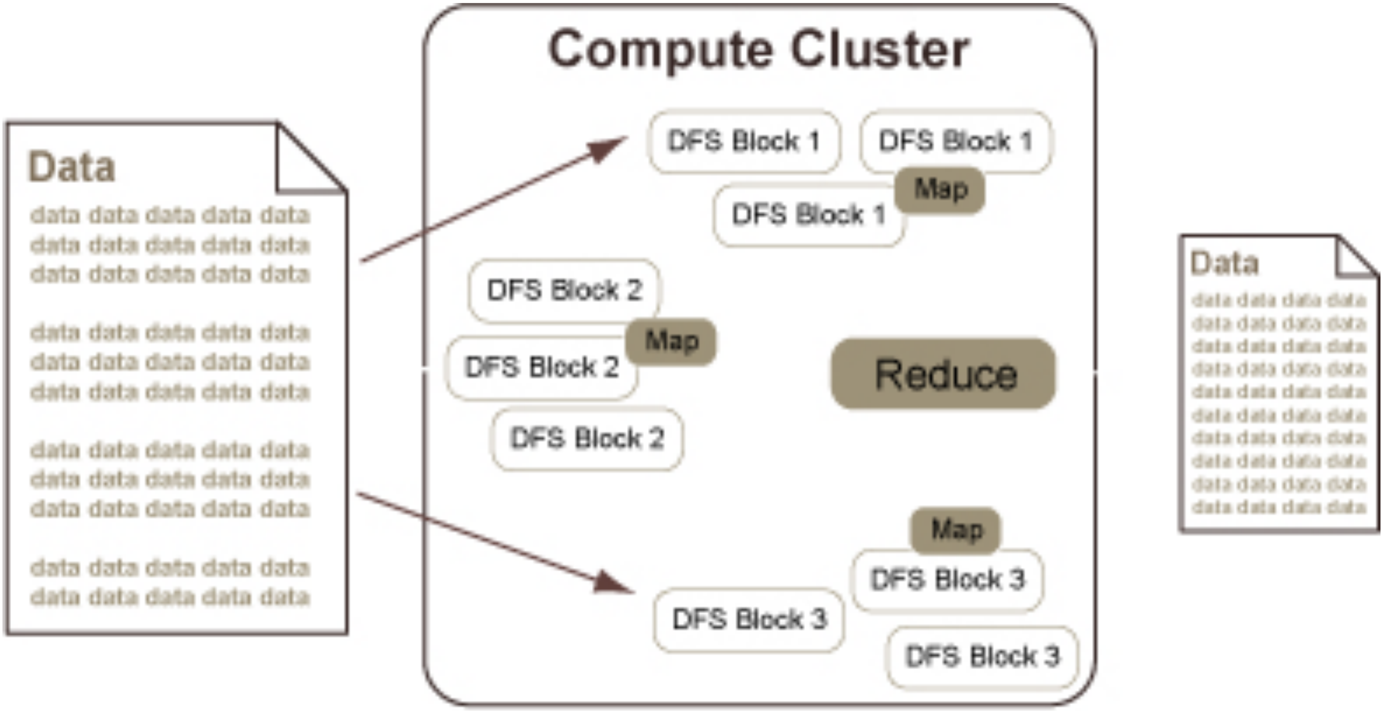


Implicit Parallelism in map

- In a purely functional setting, elements of a list being computed by map cannot see the effects of the computations on other elements
- If order of application of *f* to elements in list is *commutative*, we can reorder or parallelize execution
- This is the “secret” that MapReduce exploits

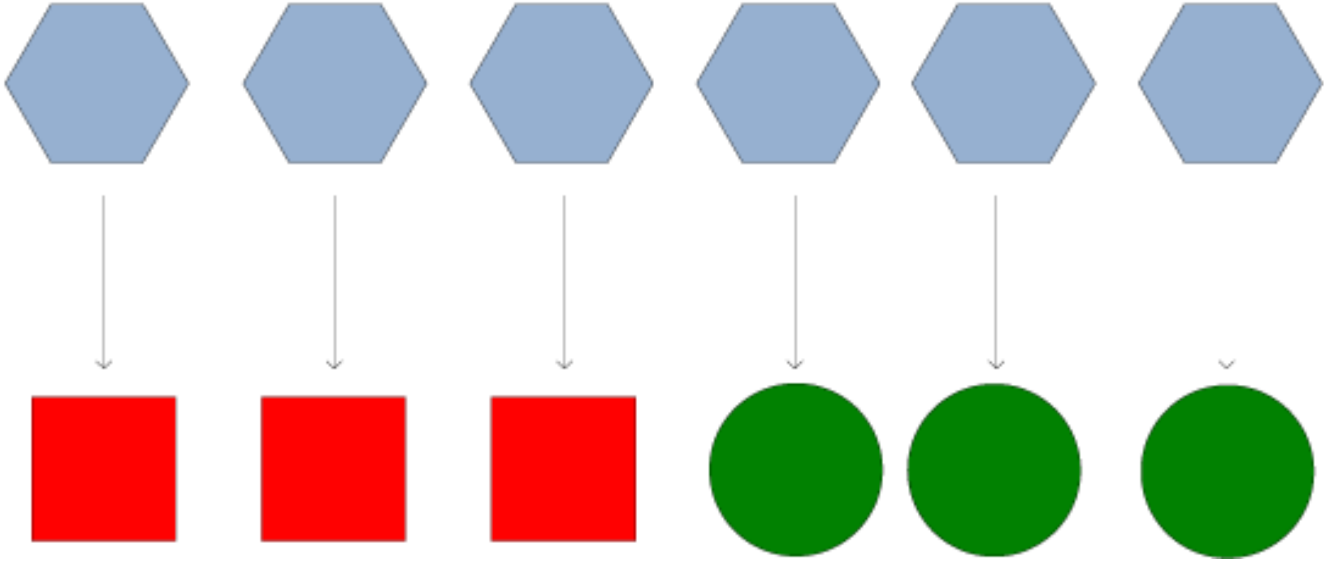
MapReduce Programming Model

- Borrows from functional programming
- Implements two basic functions:
  - map (in_key, in_value) -> (out_key, intermediate_value) list
  - reduce (out_key, intermediate_value list) -> out_value list



Map

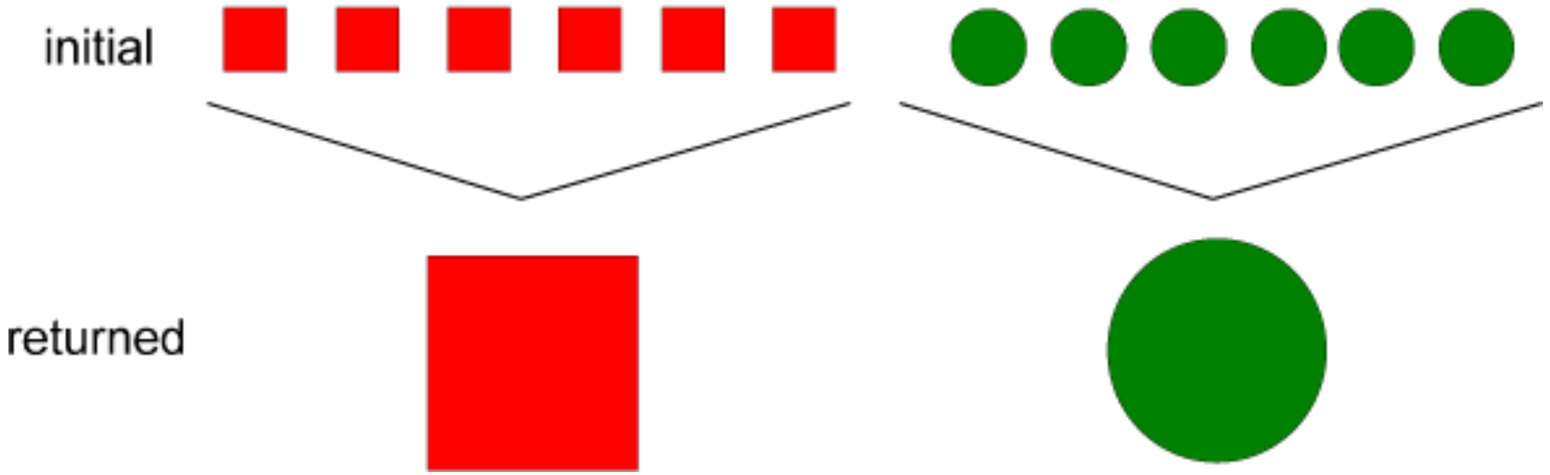
```
map  (in_key, in_value) ->
   (out_key, intermediate_value) list
```



- Records from the data source (lines out of files, rows of a database, etc) are fed into the map function as key*value pairs: e.g., (filename, line).
- map() produces one or more *intermediate* values along with an output key from the input.

Reduce

```
reduce (out_key, intermediate_value list) ->
        out_value list
```

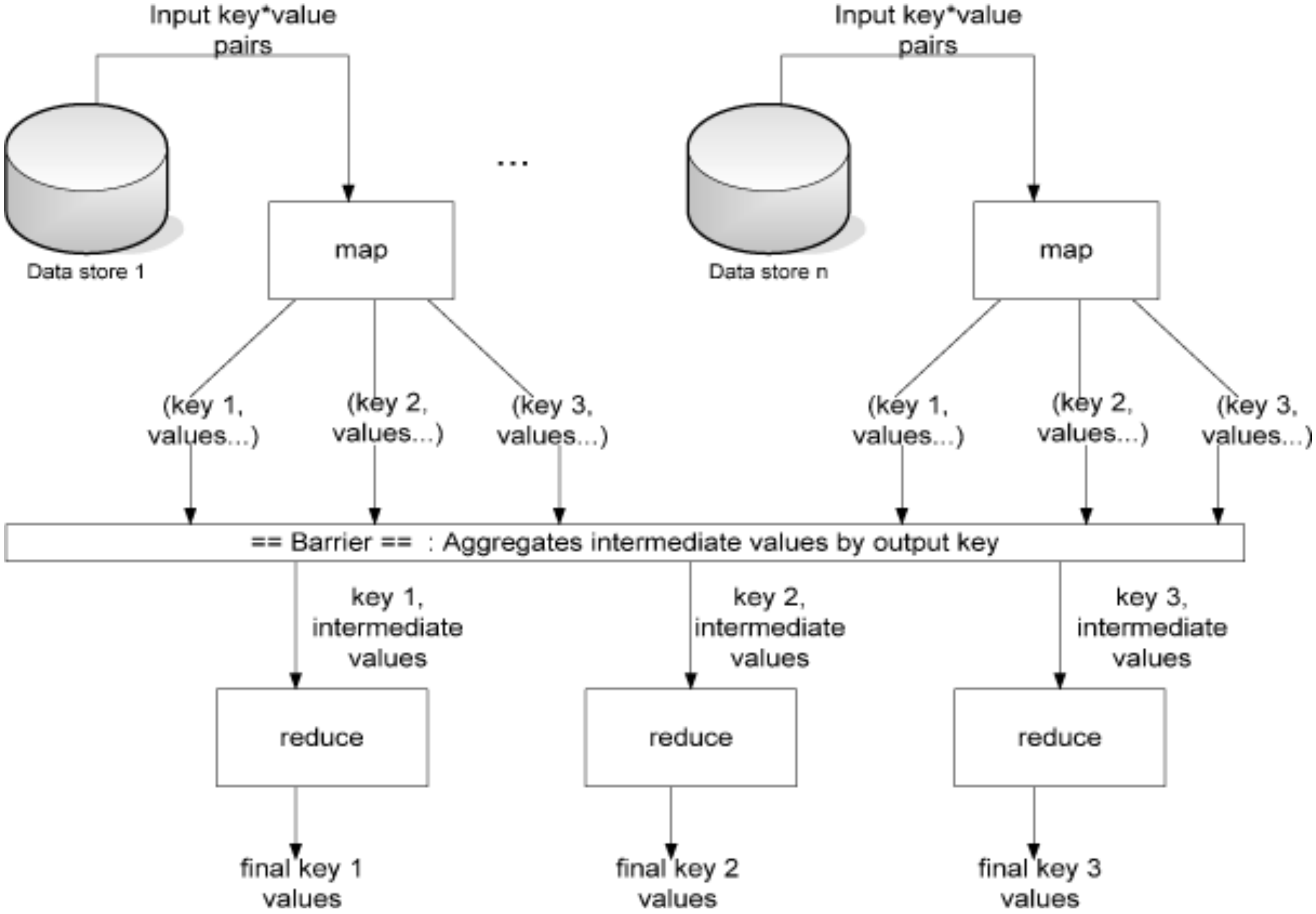


- After the map phase is over, all the intermediate values for a given output key are combined together into a list
- reduce() combines those intermediate values into one or more *final values* for that same output key
- (in practice, usually only one final value per key)

Parallelism

- map() functions run in parallel, creating different intermediate values from different input data sets
- reduce() functions also run in parallel, each working on a different output key
- All values are processed *independently*
- Bottleneck: reduce phase can’t start until map phase is completely finished.

MapReduce Parallel Programming Model



- **Bottleneck: All Map processes have to finish before Reduce starts!**

MapReduce Example

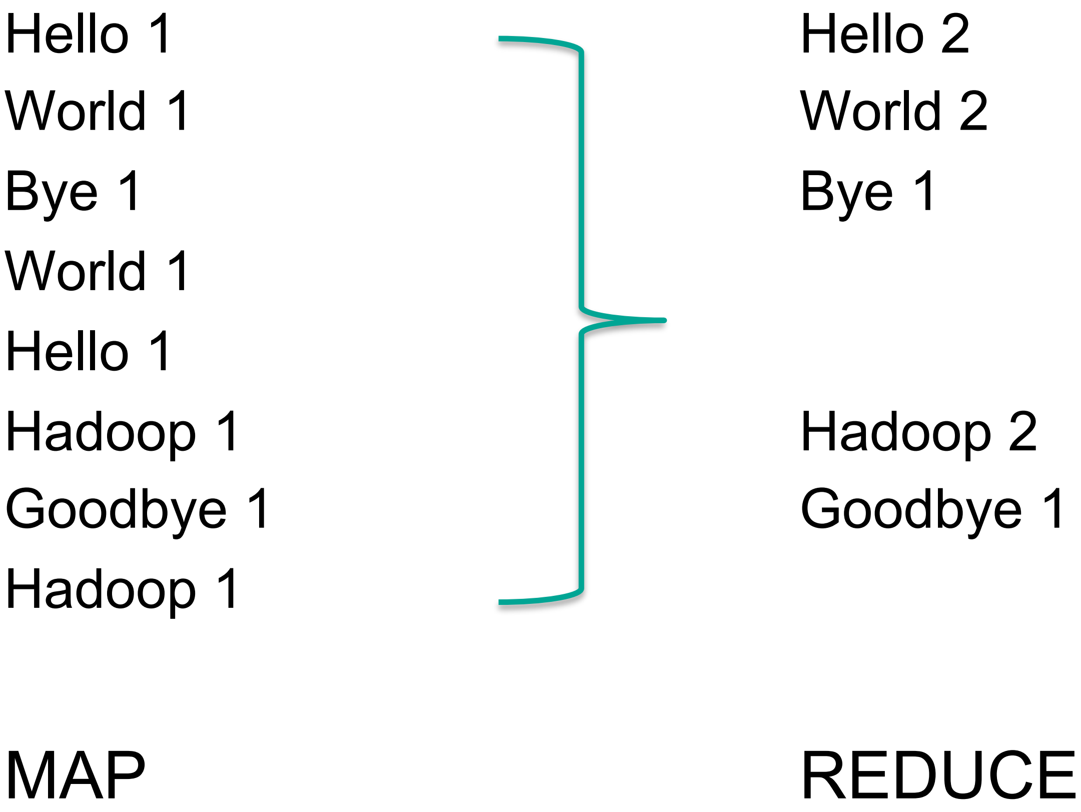
- Counting words in documents

```
map(String input_key, String input_value):
// input_key: document name
// input_value: document contents
    for each word w in input_value:
            EmitIntermediate(w, 1);

reduce(String output_key, Iterator<int> intermediate_values):
// output_key: a word
// output_values: a list of counts
    int result = 0;
    for each v in intermediate_values:
            result += v;
    Emit(result);
```

Wordcount Example

- Hello World Bye World
- Hello Hadoop Goodbye Hadoop



Hello 1
World 1
Bye 1
World 1
Hello 1
Hadoop 1
Goodbye 1
Hadoop 1

Hello 2
World 2
Bye 1

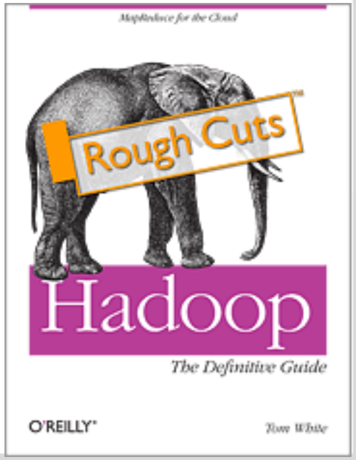
Hadoop 2
Goodbye 1

MAP

REDUCE

Programming the Cloud: Hadoop

- Reproduce the proprietary software infrastructure developed by Google (Started by Doug Cutting 2004)
- Hadoop implements
  - Parallel programming model (MapReduce)
  - Hadoop Distributed File System (HDFS)
  - Parallel database (HBase)
  - Programming environment (Pig)
  - Data warehouse infrastructure (HIVE)
  - Data collection and analysis (Chukwa)
  - Machine based learning (Mahout)
- Largest Cluster at Yahoo!: 32.000 cores and 16 PetaByte storage (1PB/16h)


MapReduce for the Cloud

Rough Cuts

Hadoop

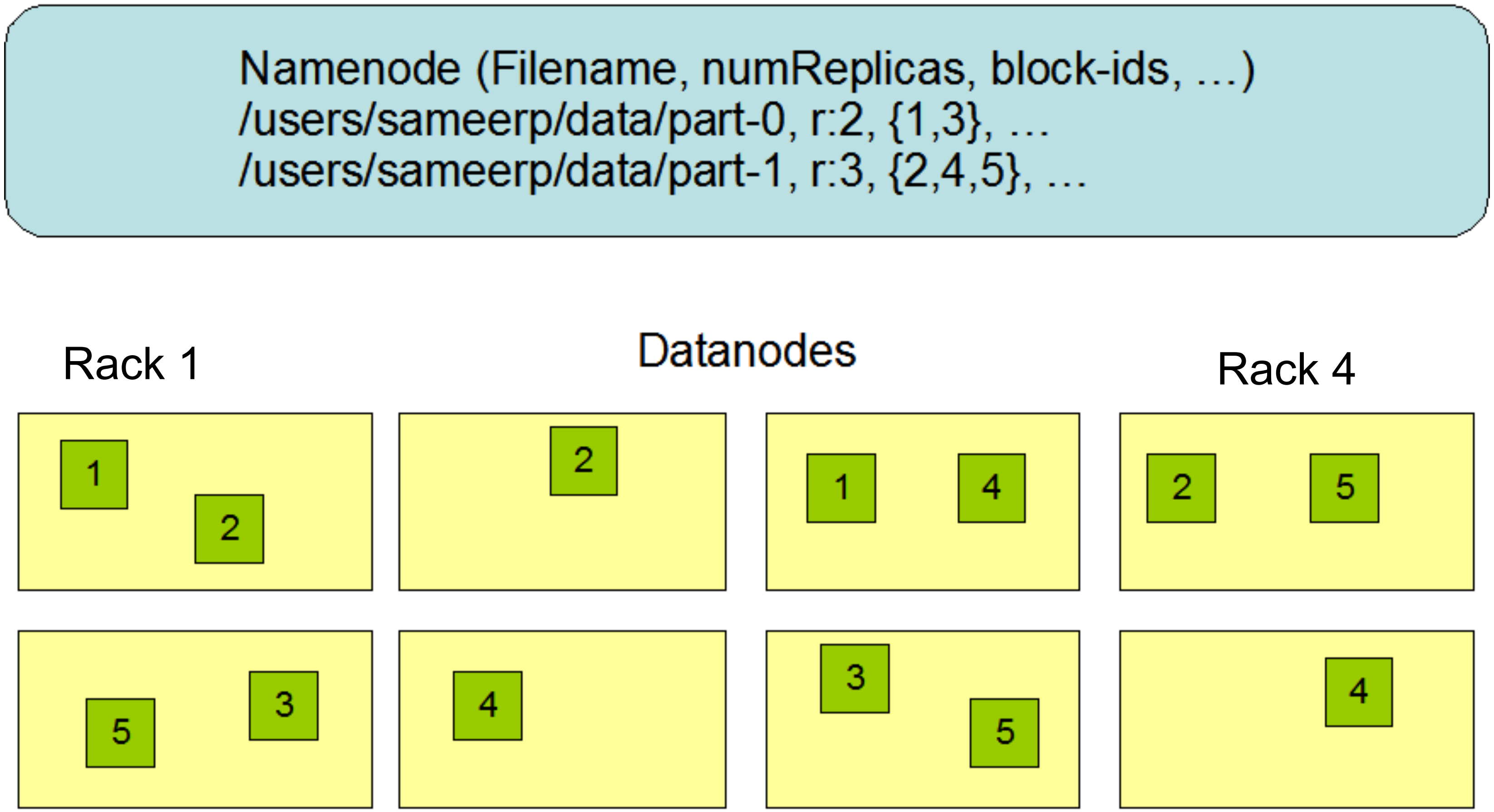
The Definitive Guide

O'REILLY

Tom White

HDFS Architecture

Block Replication



Namenode (Filename, numReplicas, block-ids, …)
/users/sameerp/data/part-0, r:2, {1,3}, …
/users/sameerp/data/part-1, r:3, {2,4,5}, …

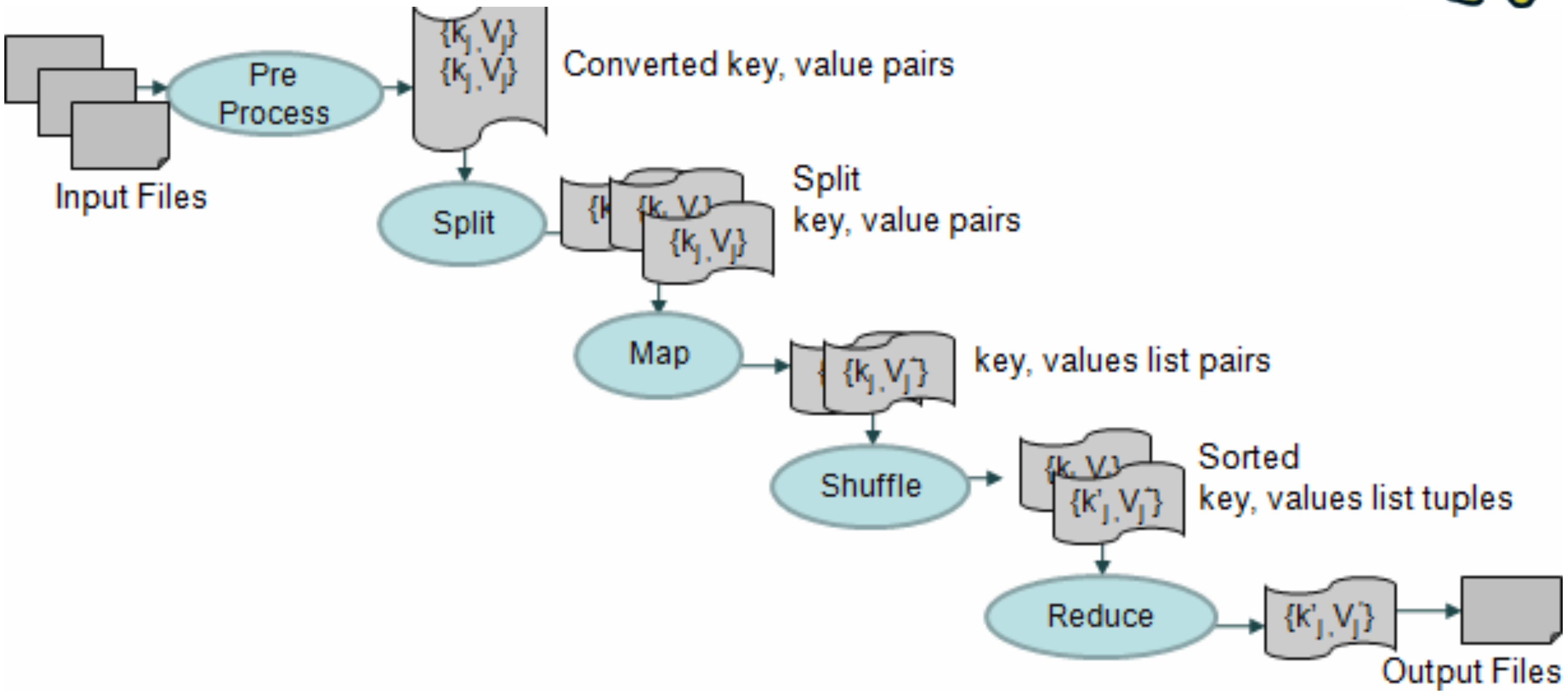
Rack 1

Datanodes

Rack 4

- Namenode holds meta data (File names, block-ids etc.)
- Blocks are replicated over data nodes (Default size: 64 MB)

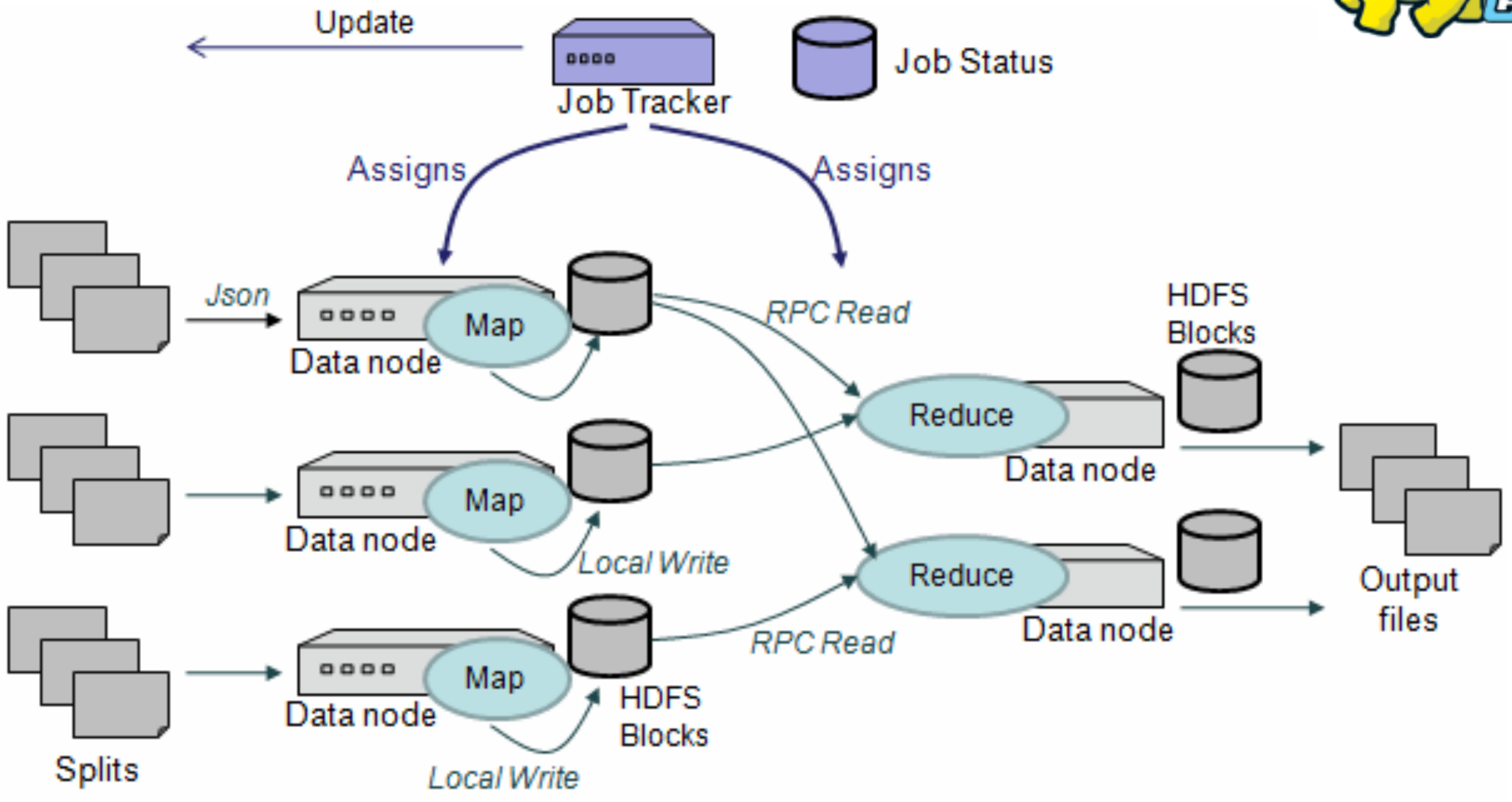
MapReduce



- Map: Generate intermediate Key/Value pairs from input data
  map (k,v) -> list (k1,v1)

- Reduce: Generate output data from intermediate data
  reduce (k1, list(v1)) -> list(v1)

Orchestration of MapReduce Tasks



- Store data in HDFS
- Job Tracker starts MapReduce Job
- Assign tasks to task tracker nodes
- Assemble output files in HDFS

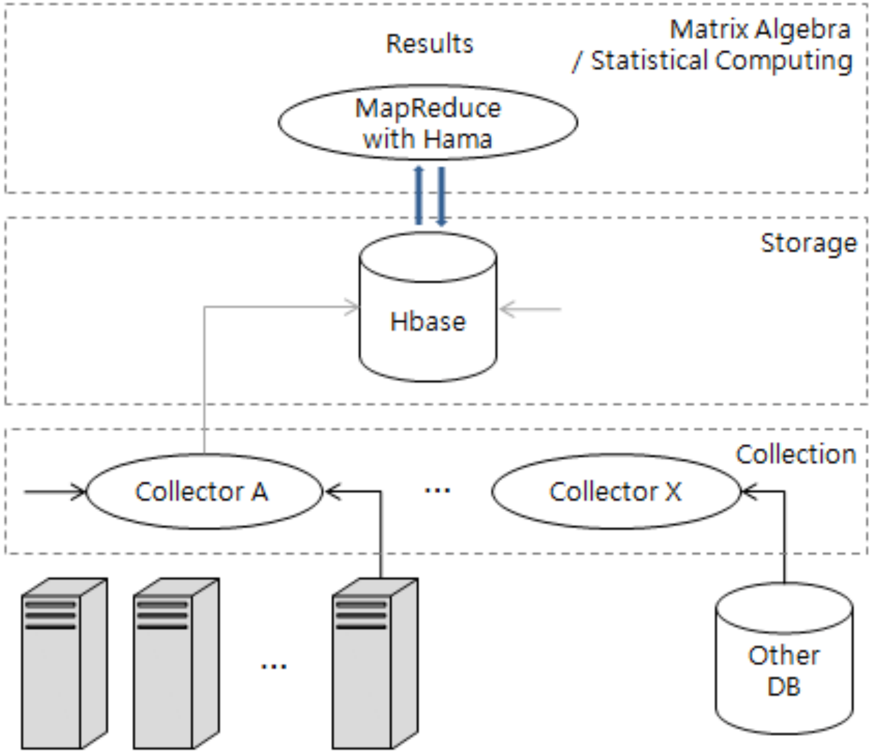
Pig

- Platform for analyzing large data sets
  - High level language to express data analysis programs
  - Parallel infrastructure
- Pig Latin
  - Ease of programming. It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain.
  - Optimization opportunities. The way in which tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency.
  - Extensibility. Users can create their own functions to do special-purpose processing.

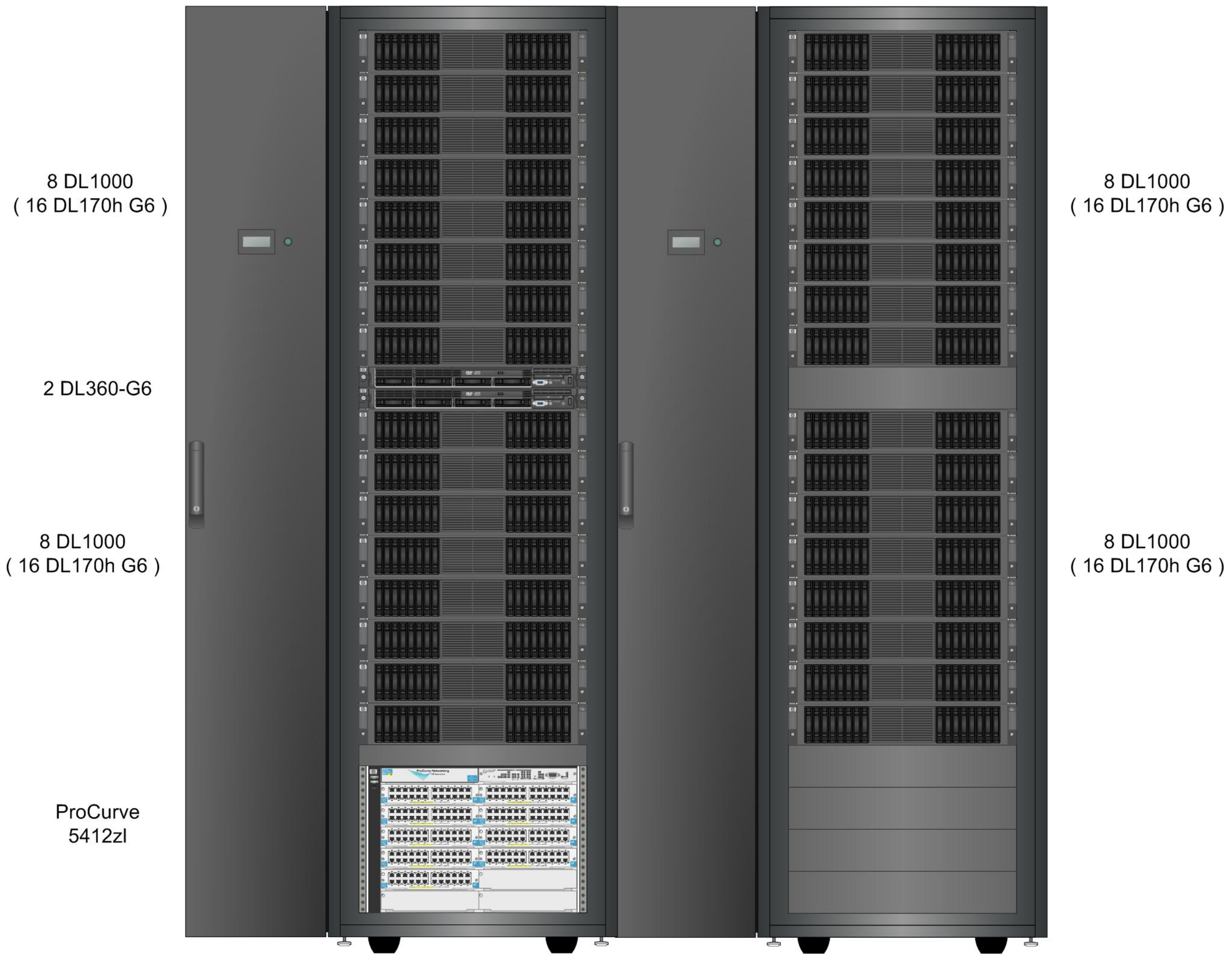
HAMA

Hama (means a hippopotamus in Korean) is a distributed matrix computation package. It is a library of matrix operations for large-scale processing and development environments as well as a Map/Reduce framework for a large-scale numerical analysis and data mining, that need the intensive computation power of matrix inversion, e.g., linear regression, PCA, SVM and etc. It will be useful for many scientific applications, e.g., physics computations, linear algebra, computational fluid dynamics, statistics, graphic rendering and many more.

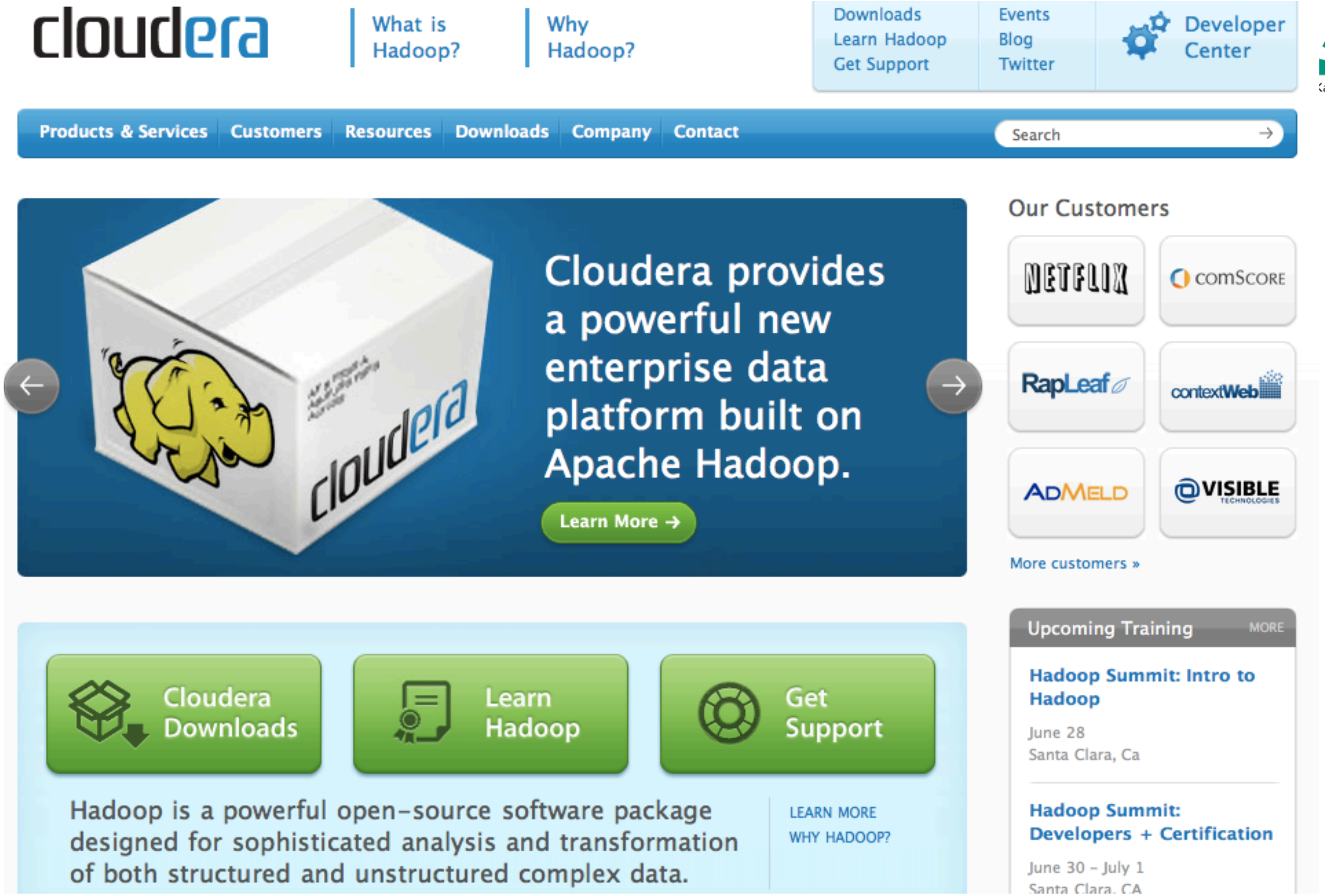
- Scientific simulation and modeling
  - Matrix-vector/matrix-matrix multiply
  - Soving linear systems
  - Scientific graphs
- Information retrieval
  - Sorting
  - Finding eigenvalues and eigenvectors
- Computer graphics and computational geometry
  - Matrix multiply
  - Computing matrix determinate



New Hadoop Cluster at KIT



- Hadoop
- 480 Nehalem cores
- 128 TB + 256 TB storage
- OpenCirrus resource
- Application area:
    - Cloud R&D
    - Bioinformatics



cloudera

What is Hadoop? | Why Hadoop?

Downloads
Learn Hadoop
Get Support

Events
Blog
Twitter

Developer Center

Products & Services | Customers | Resources | Downloads | Company | Contact | Search

Cloudera provides a powerful new enterprise data platform built on Apache Hadoop.

Learn More →

Our Customers

NETFLIX, comScore, RapLeaf, contextWeb, ADMELD, VISIBLE TECHNOLOGIES

More customers »

Cloudera Downloads | Learn Hadoop | Get Support

Hadoop is a powerful open-source software package designed for sophisticated analysis and transformation of both structured and unstructured complex data.

LEARN MORE
WHY HADOOP?

Upcoming Training MORE

Hadoop Summit: Intro to Hadoop
June 28
Santa Clara, Ca

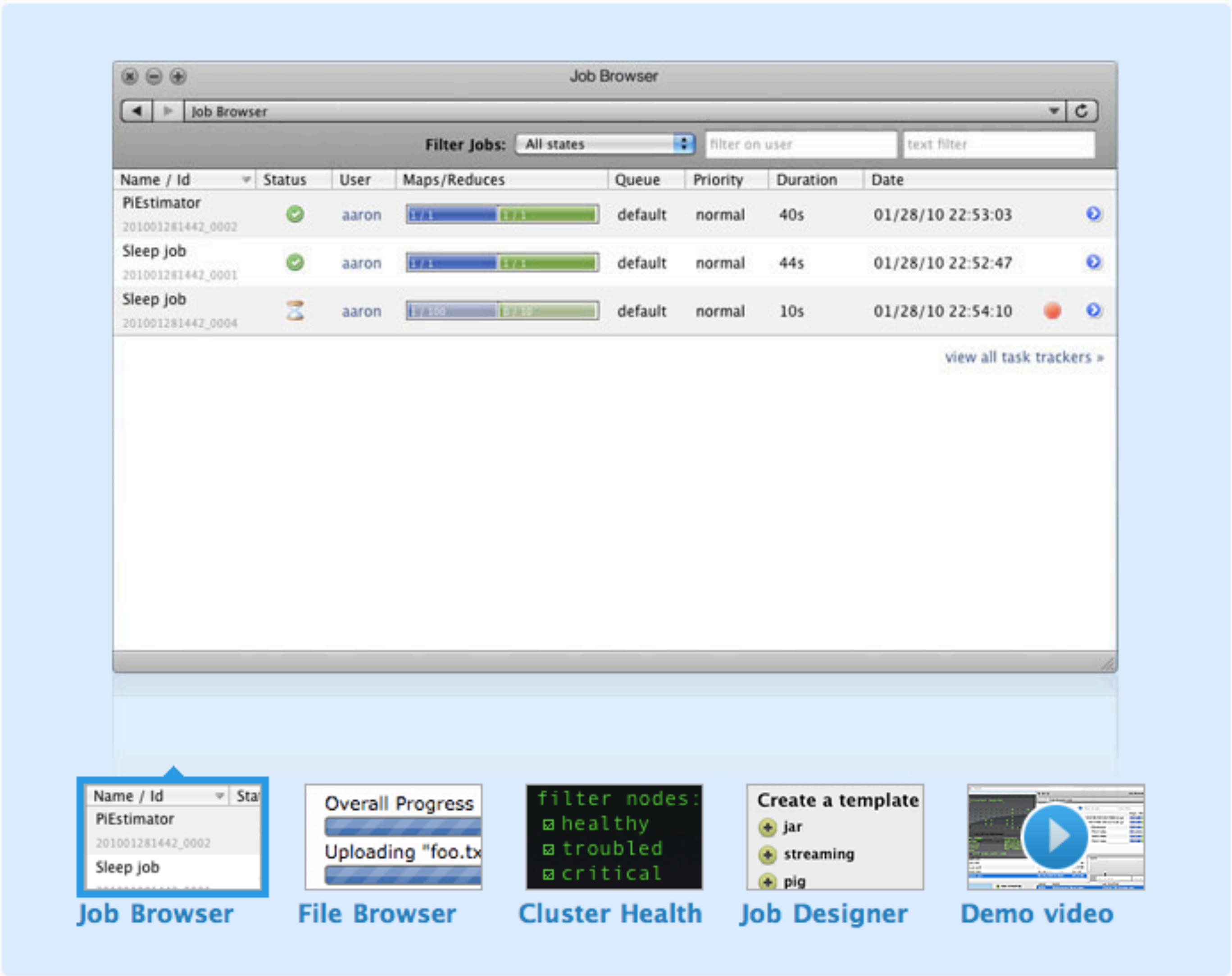
Hadoop Summit: Developers + Certification
June 30 – July 1
Santa Clara, CA

- http://www.cloudera.com/

Cloudera

- Cloudera offers a Hadoop release
  - Free download
  - Enterprise support for Hadoop
- Tutorials and training videos
- Cloudera desktop
- Virtual appliance for VMware Workstation and Fusion based on Ubuntu Linux

Cloudera Desktop



Job Browser

Job Browser

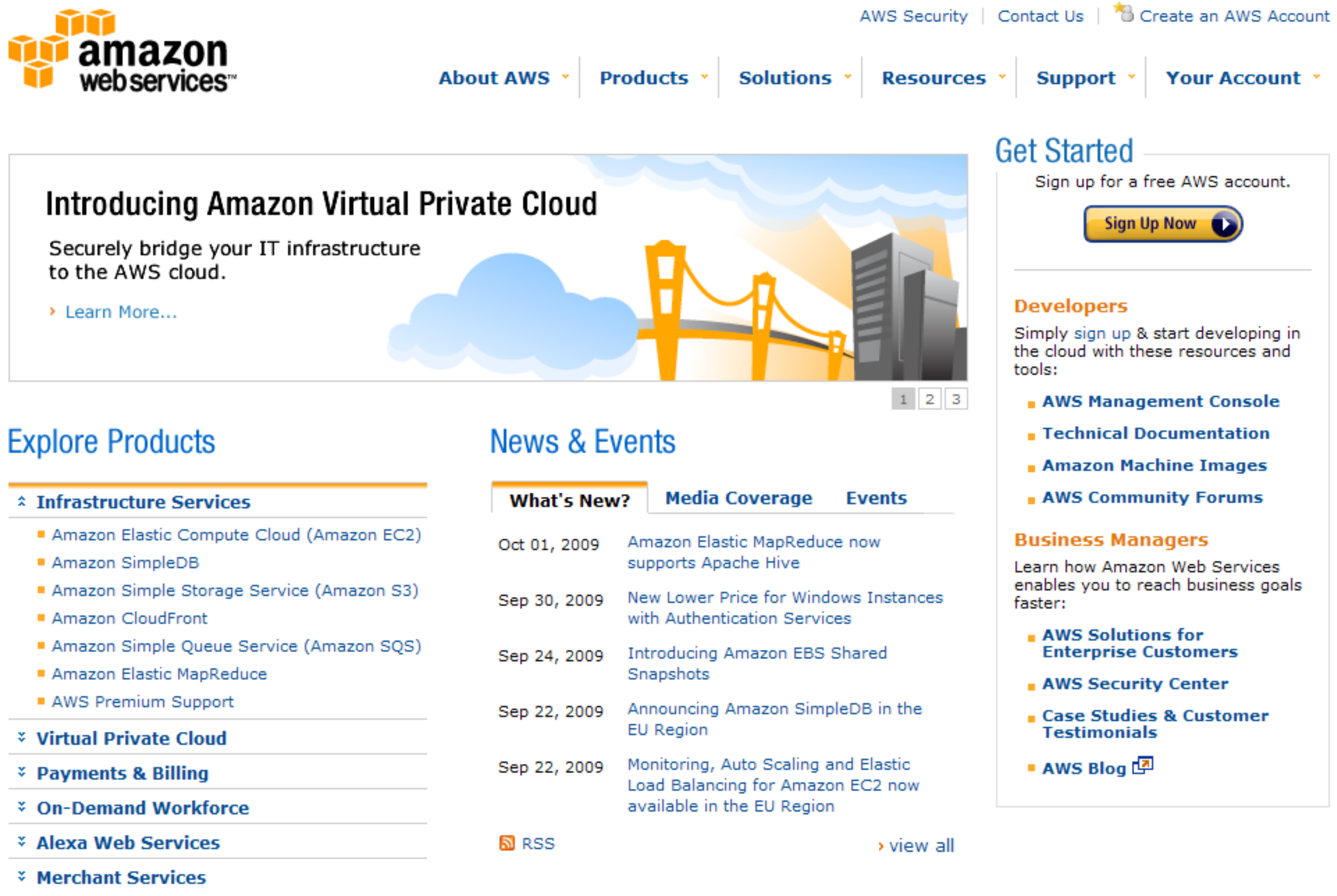
Filter Jobs: All states | filter on user | text filter

Name / Id	Status	User	Maps/Reduces	Queue	Priority	Duration	Date
PiEstimator 201001281442_0002	✓	aaron	1 / 1 1 / 1	default	normal	40s	01/28/10 22:53:03
Sleep job 201001281442_0001	✓	aaron	1 / 1 1 / 1	default	normal	44s	01/28/10 22:52:47
Sleep job 201001281442_0004	⌛	aaron	1 / 100 0 / 10	default	normal	10s	01/28/10 22:54:10 ●

view all task trackers »

Amazon Web Services

http://aws.amazon.com/



Introducing Amazon Virtual Private Cloud

Securely bridge your IT infrastructure to the AWS cloud.

› Learn More...

Explore Products

Infrastructure Services

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon SimpleDB
- Amazon Simple Storage Service (Amazon S3)
- Amazon CloudFront
- Amazon Simple Queue Service (Amazon SQS)
- Amazon Elastic MapReduce
- AWS Premium Support

Virtual Private Cloud

Payments & Billing

On-Demand Workforce

Alexa Web Services

Merchant Services

News & Events

What's New? | Media Coverage | Events

Oct 01, 2009	Amazon Elastic MapReduce now supports Apache Hive
Sep 30, 2009	New Lower Price for Windows Instances with Authentication Services
Sep 24, 2009	Introducing Amazon EBS Shared Snapshots
Sep 22, 2009	Announcing Amazon SimpleDB in the EU Region
Sep 22, 2009	Monitoring, Auto Scaling and Elastic Load Balancing for Amazon EC2 now available in the EU Region

RSS › view all

Get Started

Sign up for a free AWS account.

Sign Up Now

Developers

Simply sign up & start developing in the cloud with these resources and tools:

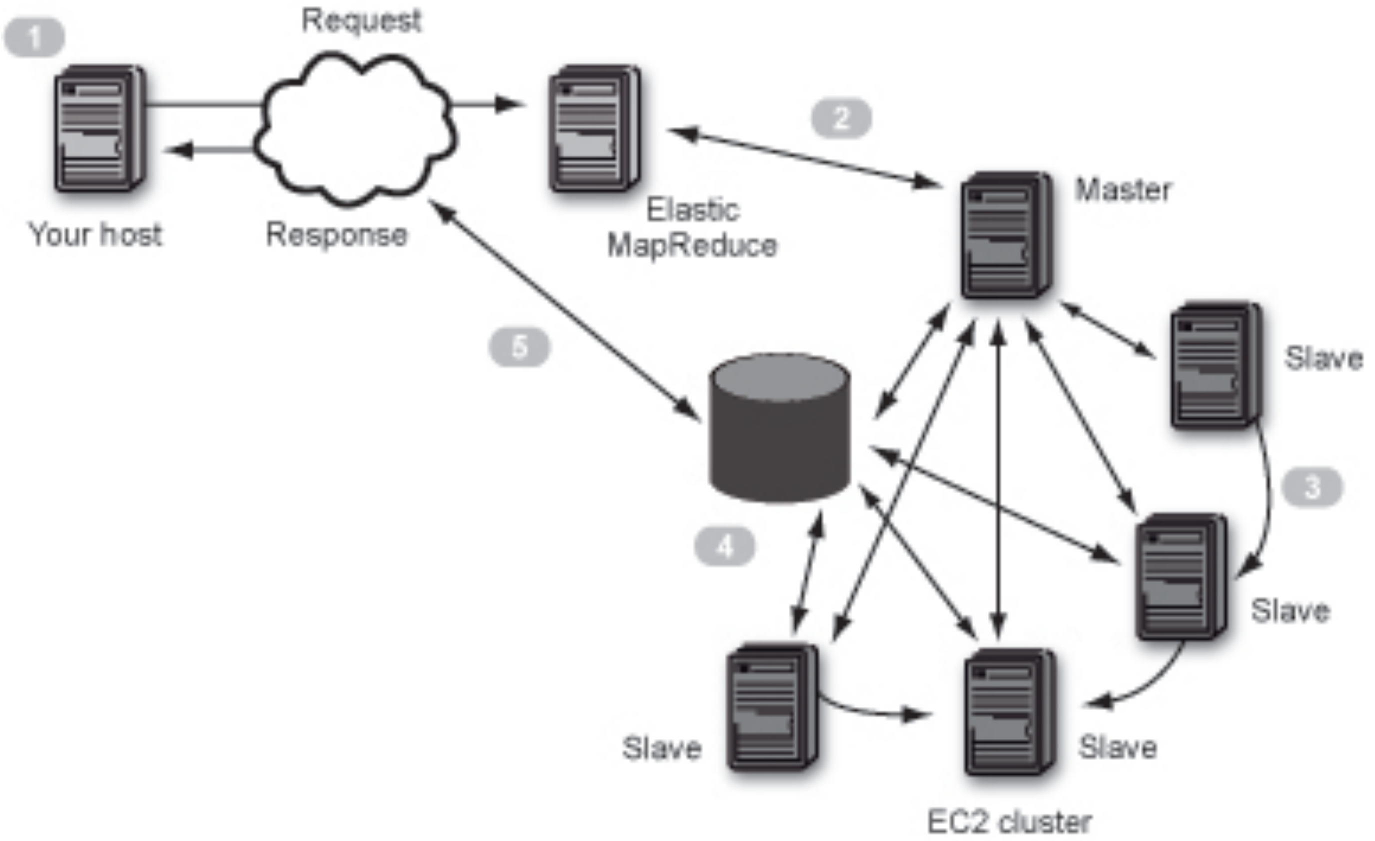
- AWS Management Console
- Technical Documentation
- Amazon Machine Images
- AWS Community Forums

Business Managers

Learn how Amazon Web Services enables you to reach business goals faster:

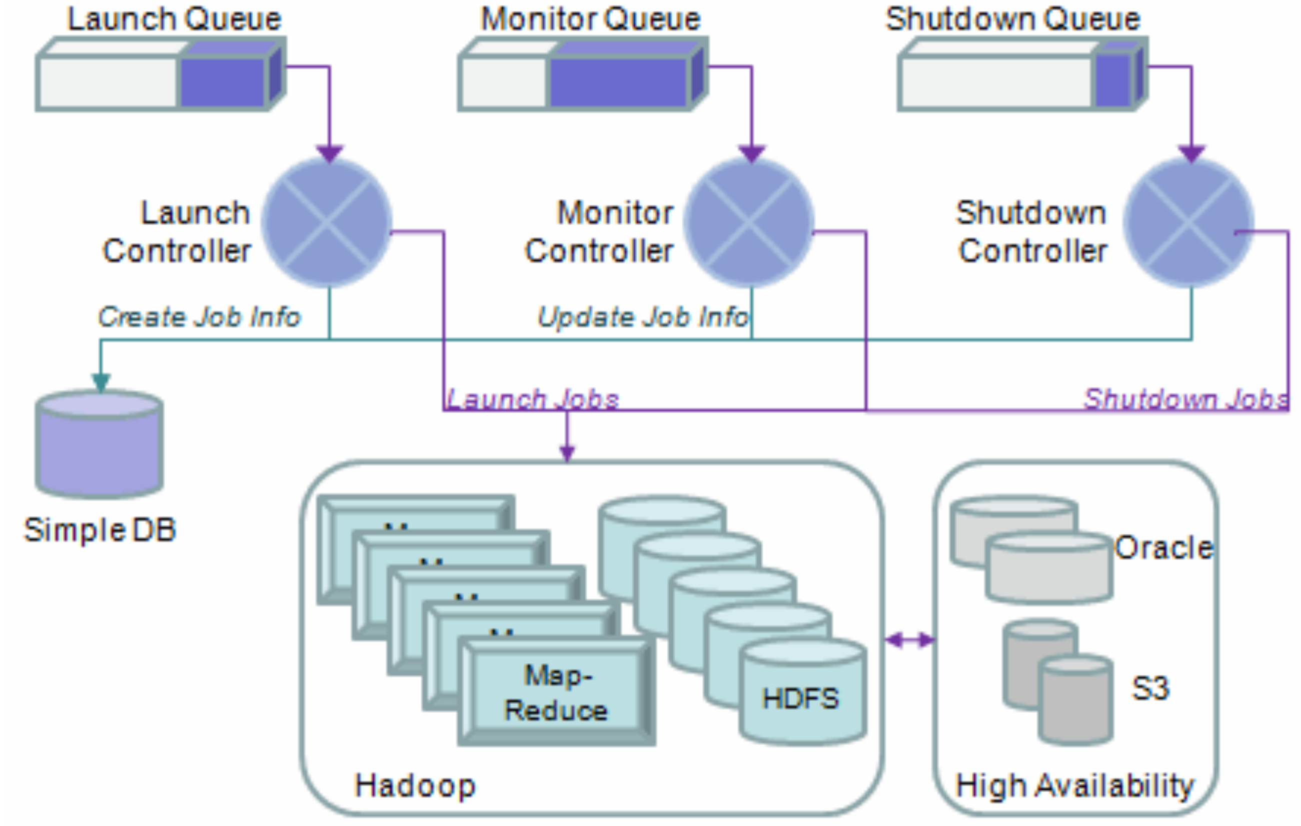
- AWS Solutions for Enterprise Customers
- AWS Security Center
- Case Studies & Customer Testimonials
- AWS Blog

Amazon Elastic MapReduce



1. Load data, Map and Reduce executables to S3
2. Elastic MapReduce starts EC2 Hadoop-Cluster (Master + Slaves)
3. Hadoop generates Jobflow to distribute S3 data to cluster and to process it
4. Results are copied over to S3
5. Message is sent at the end: Retrieve the results from S3 (Browser, wget,…)

Management of Virtual Machines in EC2

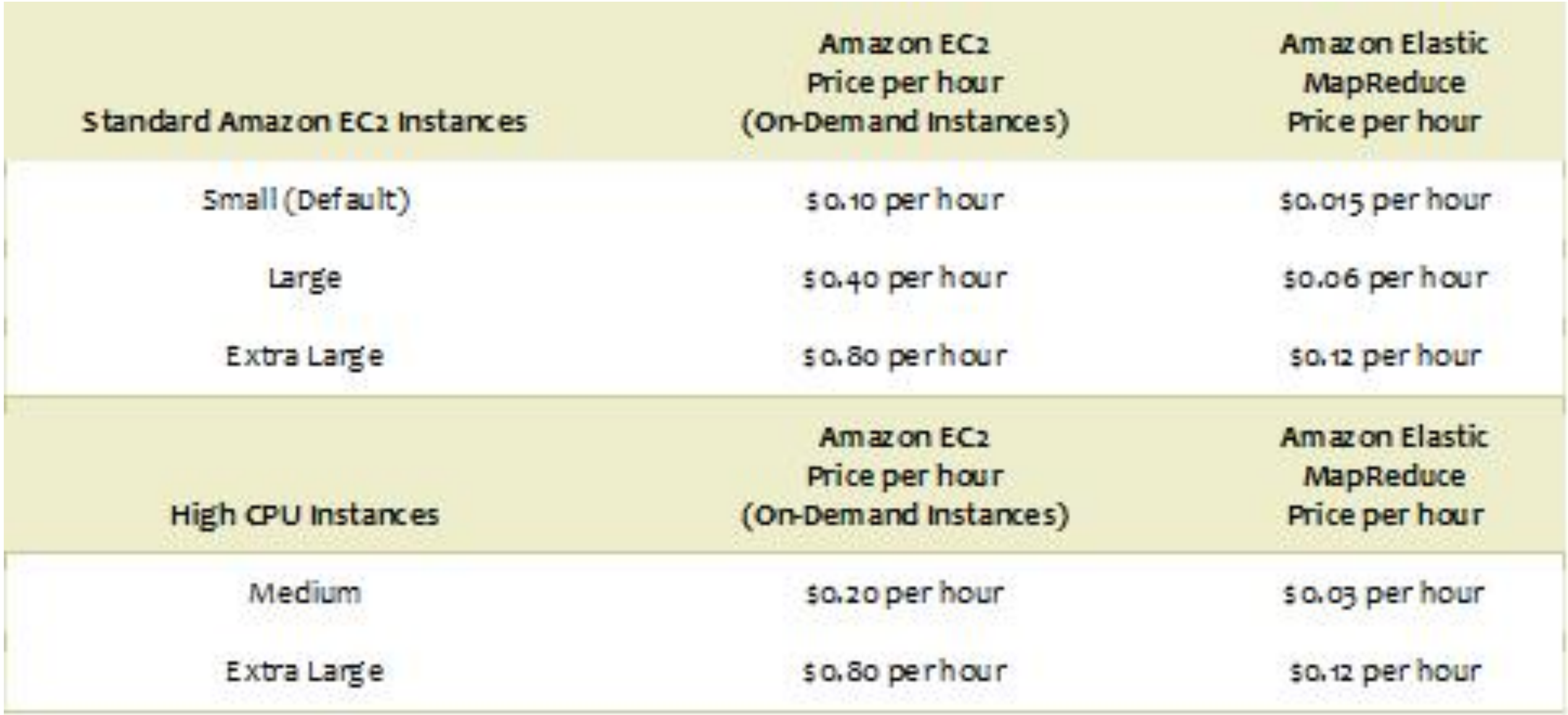


- Request queues: Launch jobs, monitor execution or shut down an existing job.
- Controllers manage and monitor the execution of those requests for jobs.
- Simple DB: Statistics on execution of the controllers and jobs for reporting purpose
- Instances are virtual machines that execute jobs
- HDFS is the default local distributed block-based storage

Amazon Elastic MapReduce API

- RunJobFlow: Creates a job flow request, starts EC2 instances and begins processing.
- DescribeJobFlows: Provides status of your job flow request(s).
- AddJobFlowSteps: Adds additional step to an already running job flow.
- TerminateJobFlows: Terminates running job flow and shutdowns all instances.

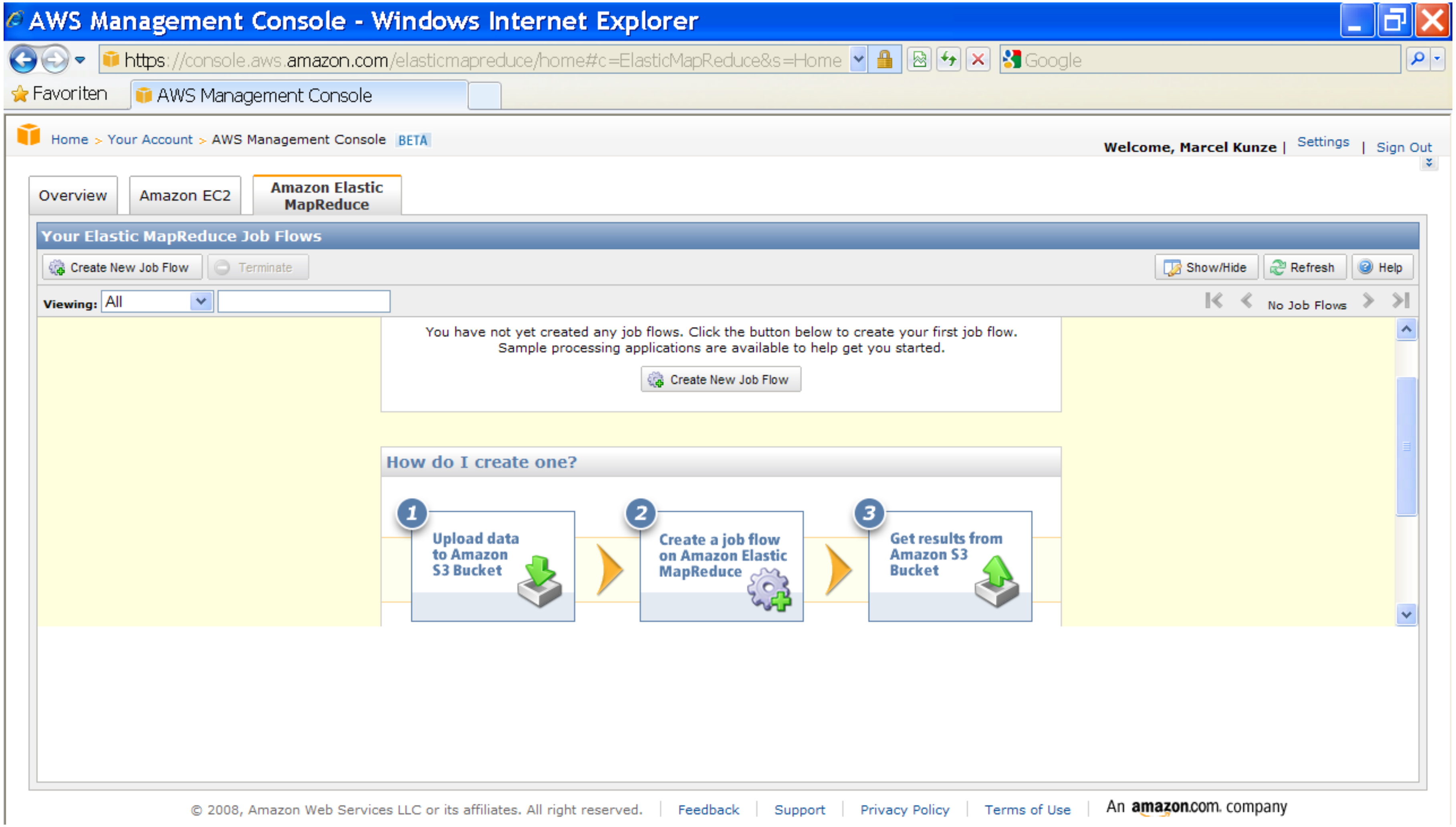
Amazon Elastic MapReduce Pricing



Standard Amazon EC2 Instances	Amazon EC2 Price per hour (On-Demand Instances)	Amazon Elastic MapReduce Price per hour
Small (Default)	$0.10 per hour	$0.015 per hour
Large	$0.40 per hour	$0.06 per hour
Extra Large	$0.80 per hour	$0.12 per hour
High CPU Instances	**Amazon EC2 Price per hour (On-Demand Instances)**	**Amazon Elastic MapReduce Price per hour**
Medium	$0.20 per hour	$0.03 per hour
Extra Large	$0.80 per hour	$0.12 per hour

- **Price in addition to EC2**

Amazon MapReduce Management Console (1)



AWS Management Console - Windows Internet Explorer
https://console.aws.amazon.com/elasticmapreduce/home#c=ElasticMapReduce&s=Home
Google
Favoriten
AWS Management Console
Home > Your Account > AWS Management Console BETA
Welcome, Marcel Kunze | Settings | Sign Out
Overview
Amazon EC2
Amazon Elastic MapReduce
Your Elastic MapReduce Job Flows
Create New Job Flow
Terminate
Show/Hide
Refresh
Help
Viewing: All
No Job Flows

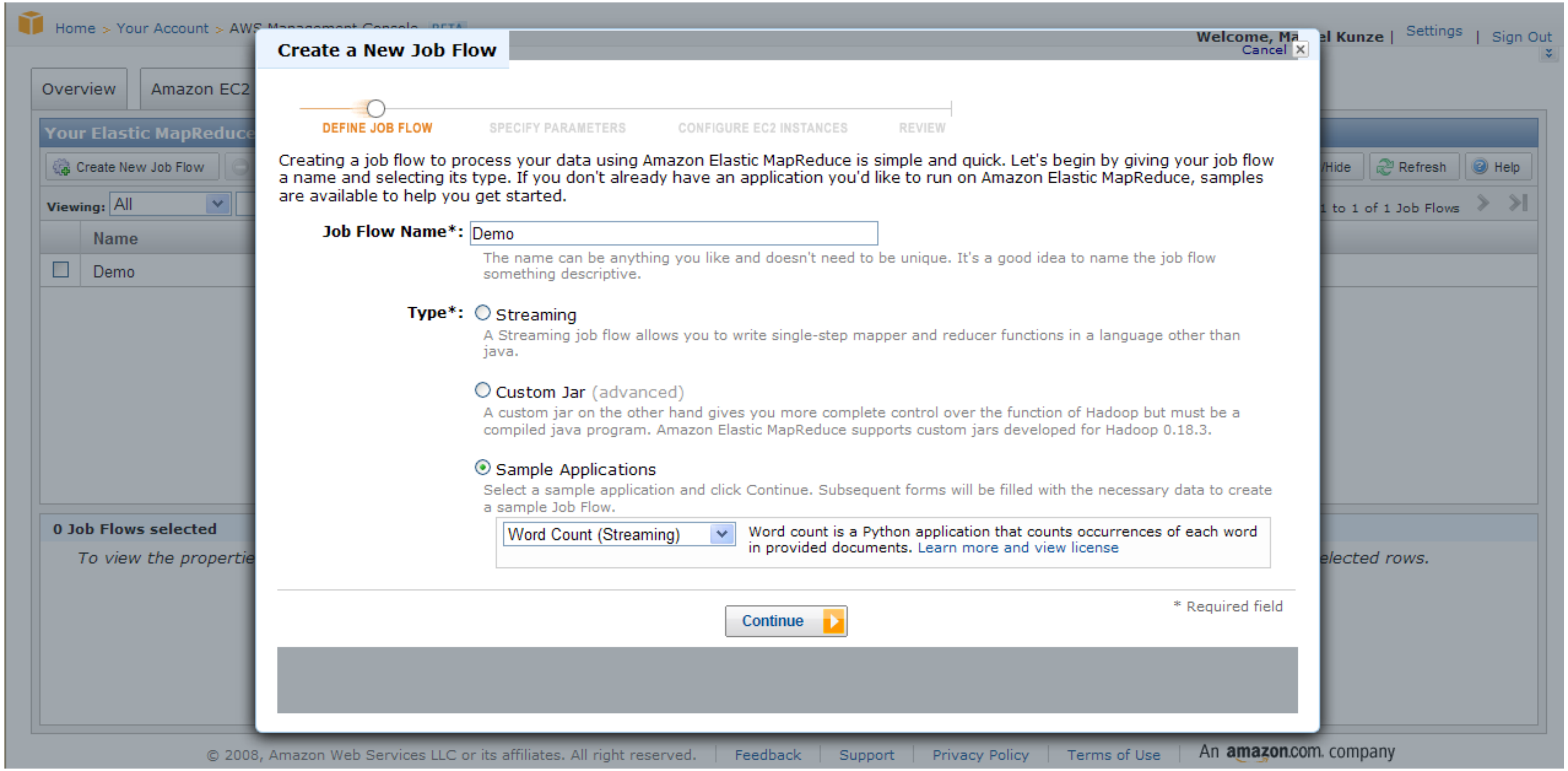
You have not yet created any job flows. Click the button below to create your first job flow.
Sample processing applications are available to help get you started.

How do I create one?

1. Upload data to Amazon S3 Bucket
2. Create a job flow on Amazon Elastic MapReduce
3. Get results from Amazon S3 Bucket

© 2008, Amazon Web Services LLC or its affiliates. All right reserved. | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com. company

Amazon MapReduce Management Console (2)



Create a New Job Flow

DEFINE JOB FLOW · SPECIFY PARAMETERS · CONFIGURE EC2 INSTANCES · REVIEW

Creating a job flow to process your data using Amazon Elastic MapReduce is simple and quick. Let's begin by giving your job flow a name and selecting its type. If you don't already have an application you'd like to run on Amazon Elastic MapReduce, samples are available to help you get started.

Job Flow Name*: Demo

The name can be anything you like and doesn't need to be unique. It's a good idea to name the job flow something descriptive.

Type*:

○ Streaming

A Streaming job flow allows you to write single-step mapper and reducer functions in a language other than java.

○ Custom Jar (advanced)

A custom jar on the other hand gives you more complete control over the function of Hadoop but must be a compiled java program. Amazon Elastic MapReduce supports custom jars developed for Hadoop 0.18.3.

● Sample Applications

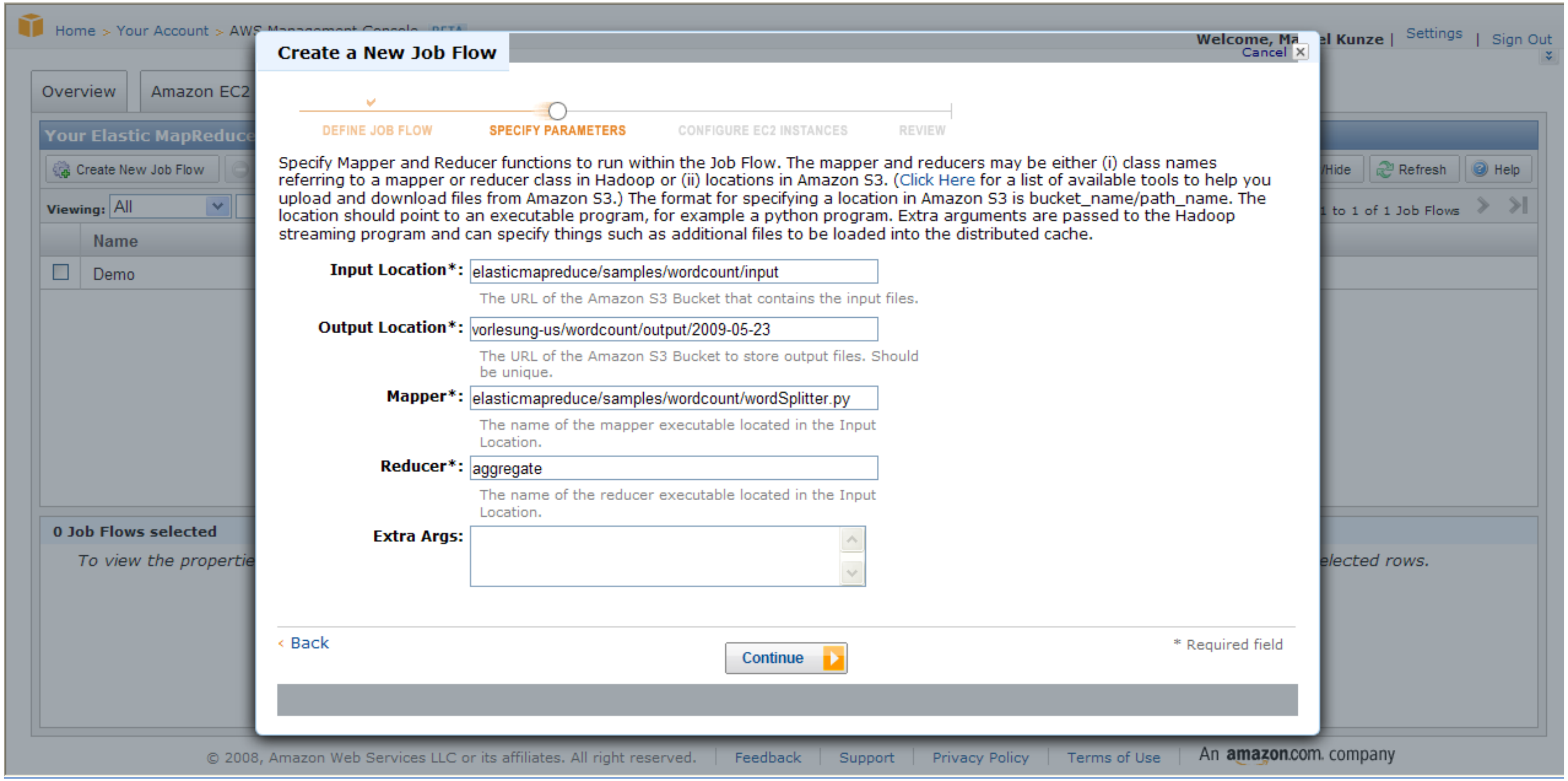
Select a sample application and click Continue. Subsequent forms will be filled with the necessary data to create a sample Job Flow.

Word Count (Streaming) — Word count is a Python application that counts occurrences of each word in provided documents. Learn more and view license

* Required field

Continue

Amazon MapReduce Management Console (3)



Home > Your Account > AWS Management Console BETA
Welcome, Ma… el Kunze | Settings | Sign Out
Overview
Amazon EC2
Your Elastic MapReduce
Create New Job Flow
Viewing: All
Name
Demo
0 Job Flows selected
To view the propertie… elected rows.
Hide
Refresh
Help
1 to 1 of 1 Job Flows
Cancel ✕
DEFINE JOB FLOW
SPECIFY PARAMETERS
CONFIGURE EC2 INSTANCES
REVIEW
‹ Back
Continue
© 2008, Amazon Web Services LLC or its affiliates. All right reserved. | Feedback | Support | Privacy Policy | Terms of Use
An amazon.com. company

Create a New Job Flow

Specify Mapper and Reducer functions to run within the Job Flow. The mapper and reducers may be either (i) class names referring to a mapper or reducer class in Hadoop or (ii) locations in Amazon S3. (Click Here for a list of available tools to help you upload and download files from Amazon S3.) The format for specifying a location in Amazon S3 is bucket_name/path_name. The location should point to an executable program, for example a python program. Extra arguments are passed to the Hadoop streaming program and can specify things such as additional files to be loaded into the distributed cache.

Input Location*: elasticmapreduce/samples/wordcount/input
The URL of the Amazon S3 Bucket that contains the input files.

Output Location*: vorlesung-us/wordcount/output/2009-05-23
The URL of the Amazon S3 Bucket to store output files. Should be unique.

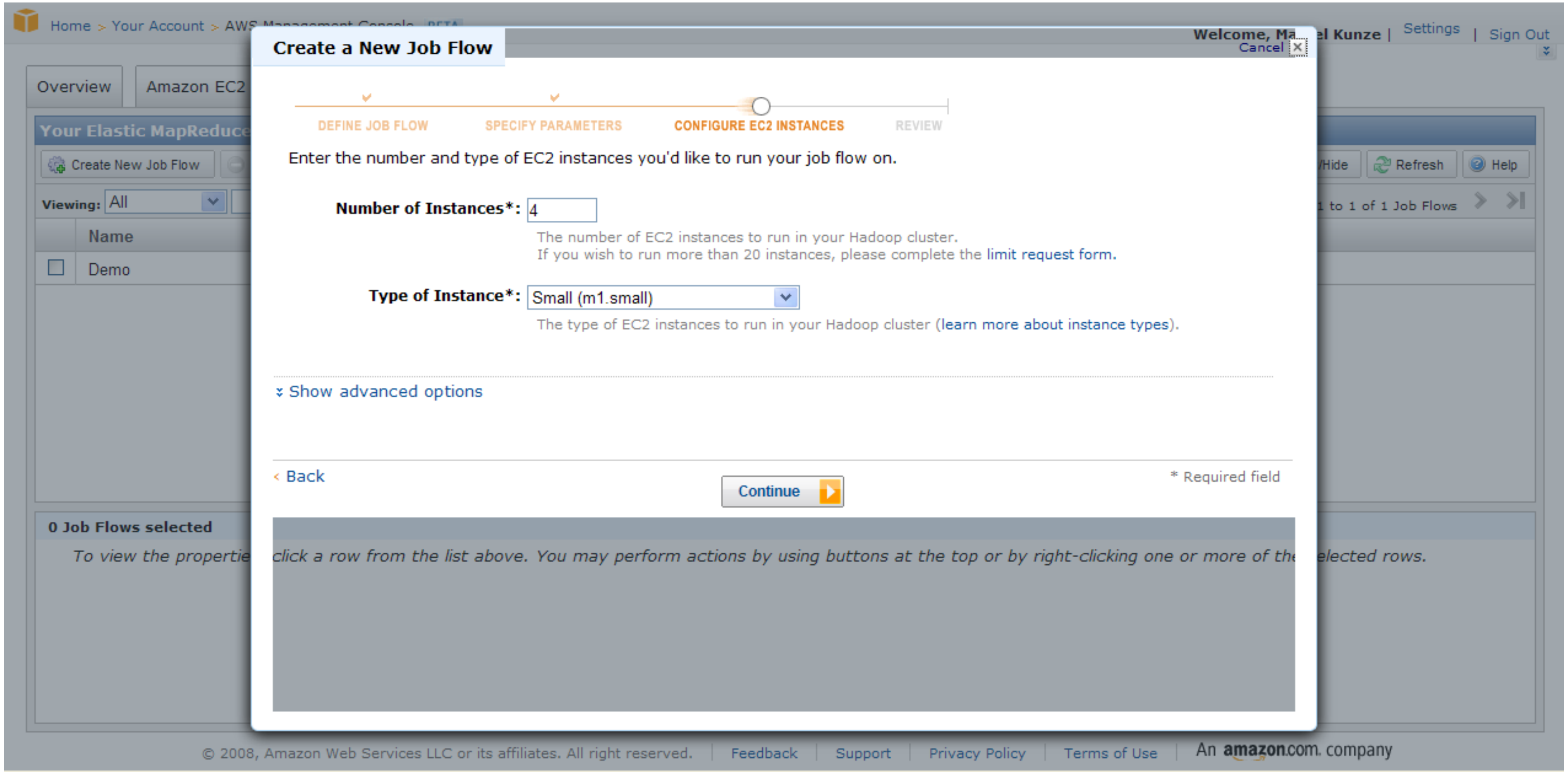
Mapper*: elasticmapreduce/samples/wordcount/wordSplitter.py
The name of the mapper executable located in the Input Location.

Reducer*: aggregate
The name of the reducer executable located in the Input Location.

Extra Args:

* Required field

Amazon MapReduce Management Console (4)



Home > Your Account > AWS Management Console BETA
Welcome, Ma... el Kunze | Settings | Sign Out
Overview
Amazon EC2
Your Elastic MapReduce
Create New Job Flow
Hide
Refresh
Help
Viewing: All
1 to 1 of 1 Job Flows
Name
Demo
0 Job Flows selected
DEFINE JOB FLOW
SPECIFY PARAMETERS
CONFIGURE EC2 INSTANCES
REVIEW
Cancel ✕
Show advanced options
Back
Continue
© 2008, Amazon Web Services LLC or its affiliates. All right reserved. | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com. company

Create a New Job Flow

Enter the number and type of EC2 instances you'd like to run your job flow on.

Number of Instances*: 4

The number of EC2 instances to run in your Hadoop cluster.
If you wish to run more than 20 instances, please complete the limit request form.

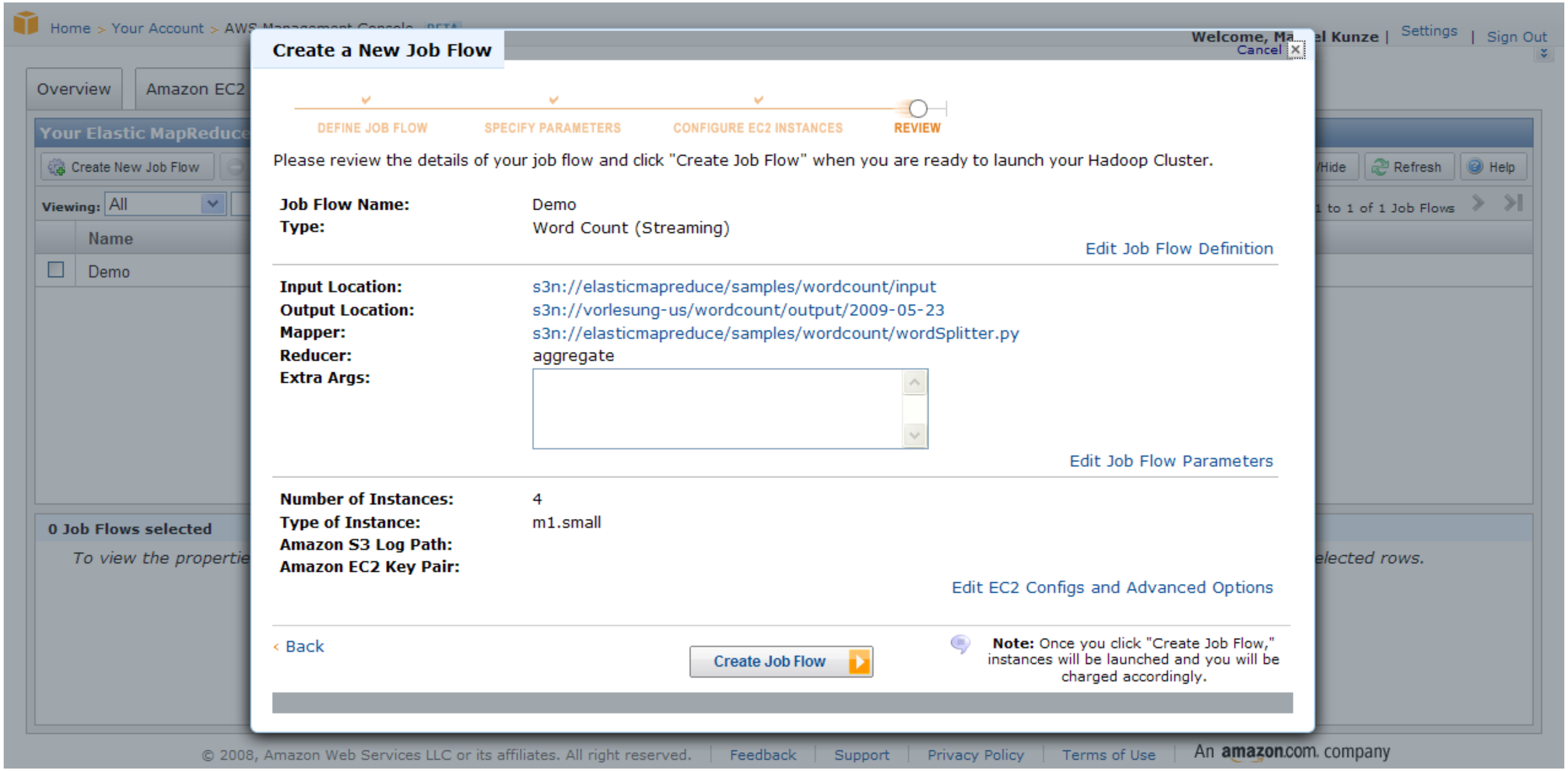
Type of Instance*: Small (m1.small)

The type of EC2 instances to run in your Hadoop cluster (learn more about instance types).

* Required field

To view the properties ... click a row from the list above. You may perform actions by using buttons at the top or by right-clicking one or more of the ... elected rows.

Amazon MapReduce Management Console (5)



Create a New Job Flow

Please review the details of your job flow and click "Create Job Flow" when you are ready to launch your Hadoop Cluster.

Job Flow Name:	Demo
Type:	Word Count (Streaming)

Edit Job Flow Definition

Input Location:	s3n://elasticmapreduce/samples/wordcount/input
Output Location:	s3n://vorlesung-us/wordcount/output/2009-05-23
Mapper:	s3n://elasticmapreduce/samples/wordcount/wordSplitter.py
Reducer:	aggregate
Extra Args:	

Edit Job Flow Parameters

Number of Instances:	4
Type of Instance:	m1.small
Amazon S3 Log Path:	
Amazon EC2 Key Pair:	

Edit EC2 Configs and Advanced Options

Note: Once you click "Create Job Flow," instances will be launched and you will be charged accordingly.

Summary

- MapReduce
  - Useful abstraction to simplify large-scale computations
  - Functional programming paradigm can be applied to large-scale applications
  - Fun to use: focus on problem, let library deal w/ messy details

- Hadoop
  - Re-implements Google software infrastructure as OpenSource
  - Ecosystem of useful services to process large data
  - Cloudera offers Hadoop release with enterprise support
  - Amazon Elastic MapReduce Service

©johnlund.com