

Distributed and Cloud Computing

K. Hwang, G. Fox and J. Dongarra

Chapter 6: Cloud Programming and Software Environments

(suggested for use in 5 lectures in 250 minutes)

Prepared by Kai Hwang
University of Southern California
March 30, 2012

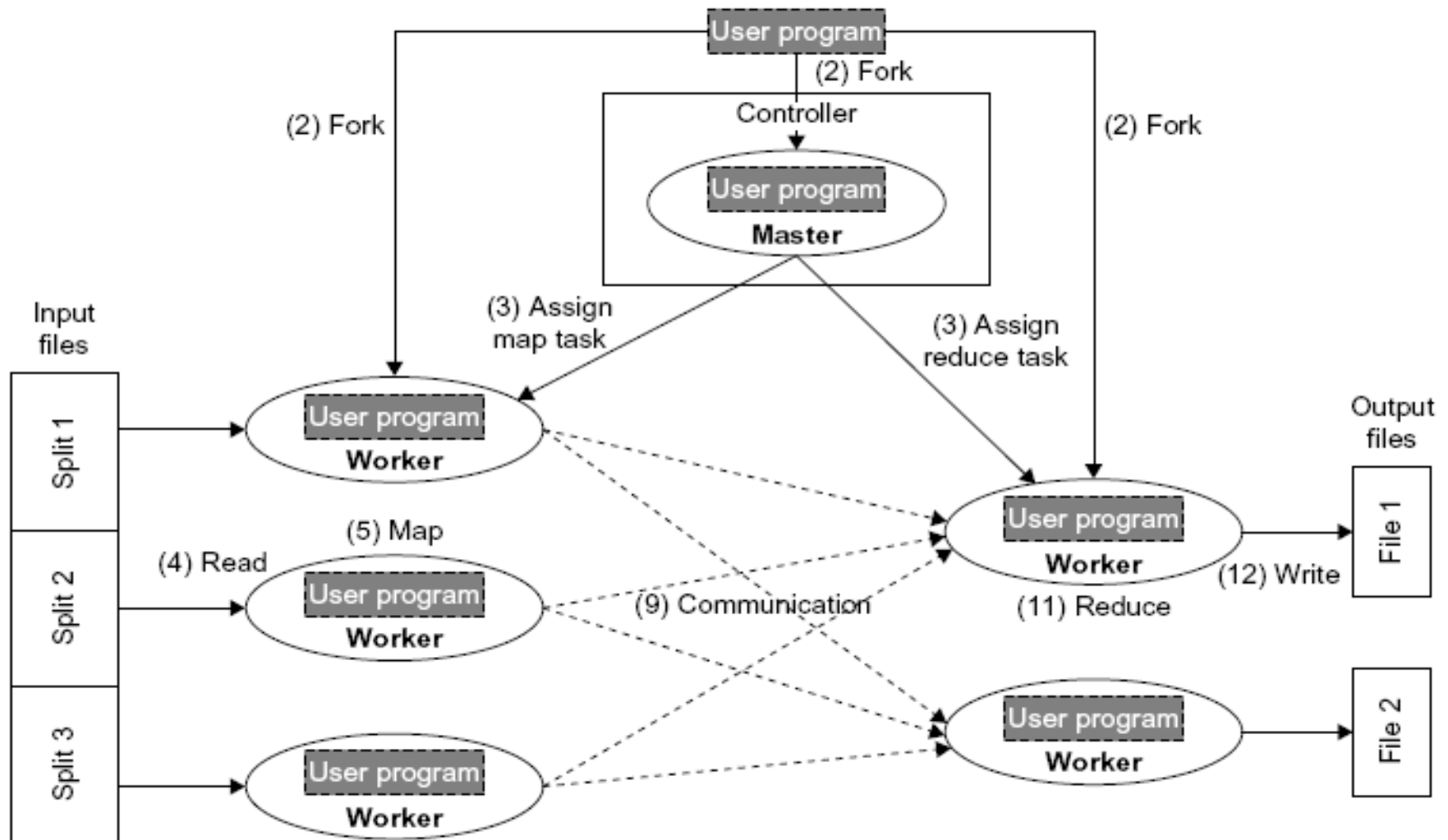


FIGURE 6.6

Control flow implementation of MapReduce.

(Courtesy of Yahoo! Pig Tutorial [54])

Example: Document Indexing

- Input: Set of documents D_1, \dots, D_N
- Map
 - Parse document D into terms T_1, \dots, T_N
 - Produces (key, value) pairs
 - $(T_1, D), \dots, (T_N, D)$
- Reduce
 - Receives list of (key, value) pairs for term T
 - $(T, D_1), \dots, (T, D_N)$
 - Emits single (key, value) pair
 - $(T, (D_1, \dots, D_N))$

MapReduce in Google

Easy to use. Library hides complexity.

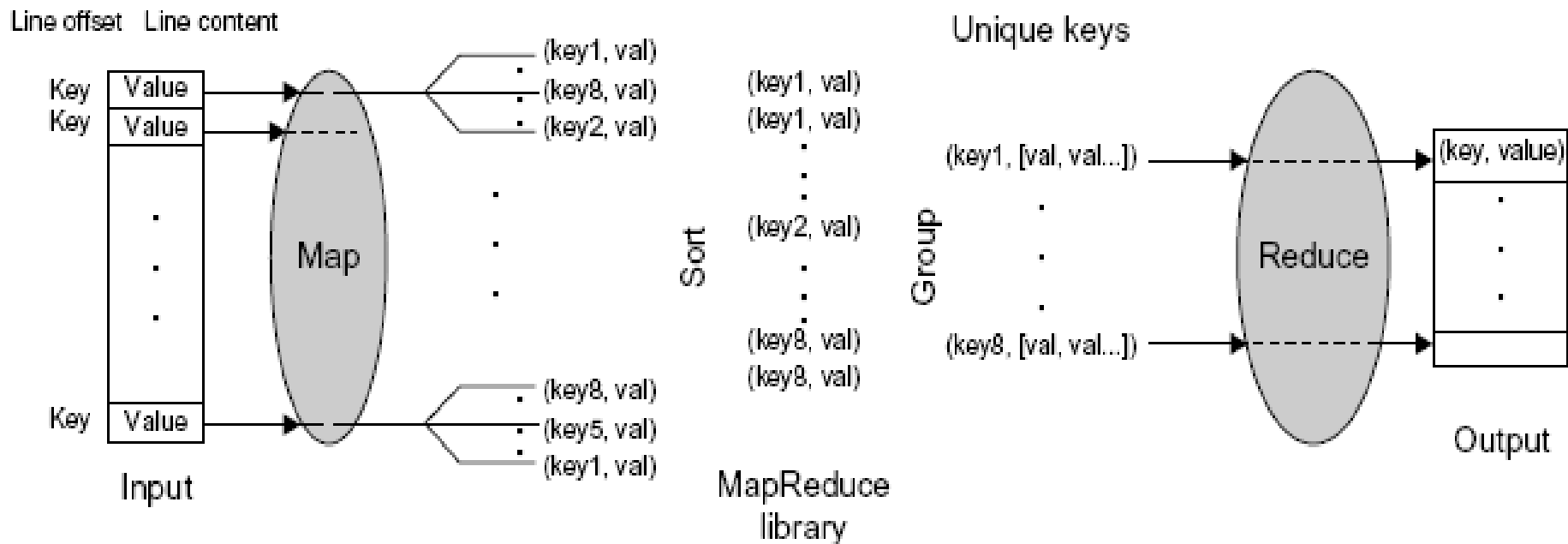
	Mar, '05	Mar, '06	Sep, '07
Number of jobs	72K	171K	2,217K
Average time (seconds)	934	874	395
Machine years used	981	2,002	11,081
Input data read (TB)	12,571	52,254	403,152
Intermediate data (TB)	2,756	6,743	34,774
Output data written (TB)	941	2,970	14,018
Average worker machines	232	268	394

MapReduce and MapReduce++

	Google MapReduce [30]	Apache Hadoop [23]	Microsoft Dryad [26]	Twister [28]	Azure Twister [31]
Program- ming Model	MapReduce	MapReduce	DAG execution, Extensible to MapReduce and other patterns	Iterative MapReduce	Currently just MapReduce-- will extend to Iterative MapReduce
Data Handling	GFS (Google File System)	HDFS (Hadoop Distributed File System)	Shared Directories & local disks	Local disks and data management tools	Azure Blob Storage
Scheduling	Data Locality	Data Locality; Rack aware, Dynamic task scheduling through global queue	Data locality; Network topology based run time graph optimizations; Static task partitions	Data Locality; Static task partitions	Dynamic task scheduling through global queue
Failure Handling	Re-execution of failed tasks; Duplicated execution of slow tasks	Re-execution of failed tasks; Duplicate execution of slow tasks	Re-execution of failed tasks; Duplicate execution of slow tasks	Re-execution of Iterations	Re-execution of failed tasks; Duplicate execution of slow tasks
HLL Support	Sawzall [32]	Pig Latin [33, 34]	DryadLINQ [27]	Pregel [35] has related features	N/A
Environ- ment	Linux Cluster.	Linux Clusters, Amazon Elastic Map Reduce on EC2	Windows HPCS cluster	Linux Cluster EC2	Windows Azure Azure Local Development Fabric
Interme- diate data transfer	File	File, Http	File, TCP pipes, shared-memory FIFOs	Publish/Subscri be messaging	Files, TCP

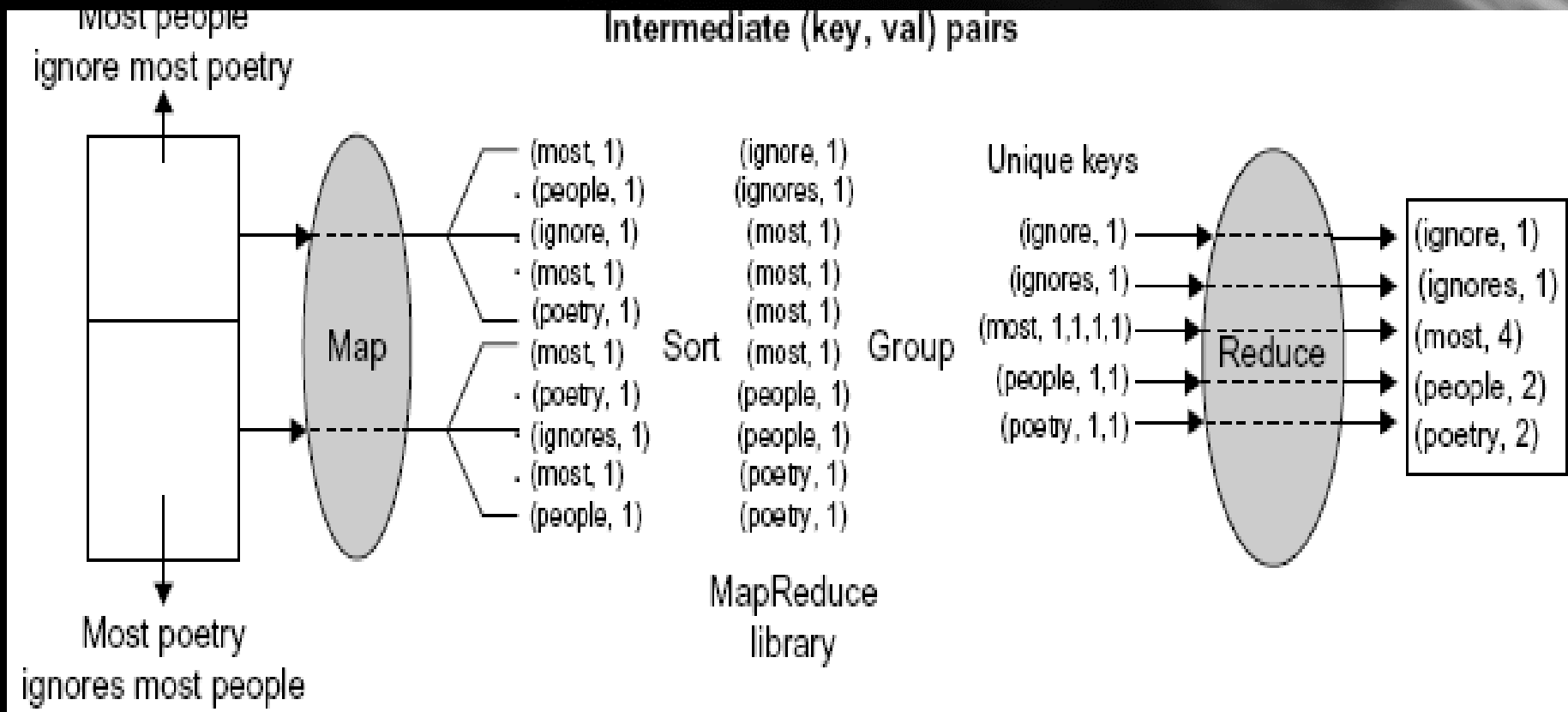
Logical Data Flow in 5 Processing Steps in MapReduce Process

Intermediate (key, val) pairs



(Key, Value) Pairs are generated by the Map function over multiple available Map Workers (VM instances). These pairs are then sorted and group based on key ordering. Different key-groups are then processed by multiple Reduce Workers in parallel.

A Word Counting Example on $\langle \text{Key}, \text{Count} \rangle$ Distribution



Linking the Map Workers and Reduce Workers by Key Matching in Partitioning Functions

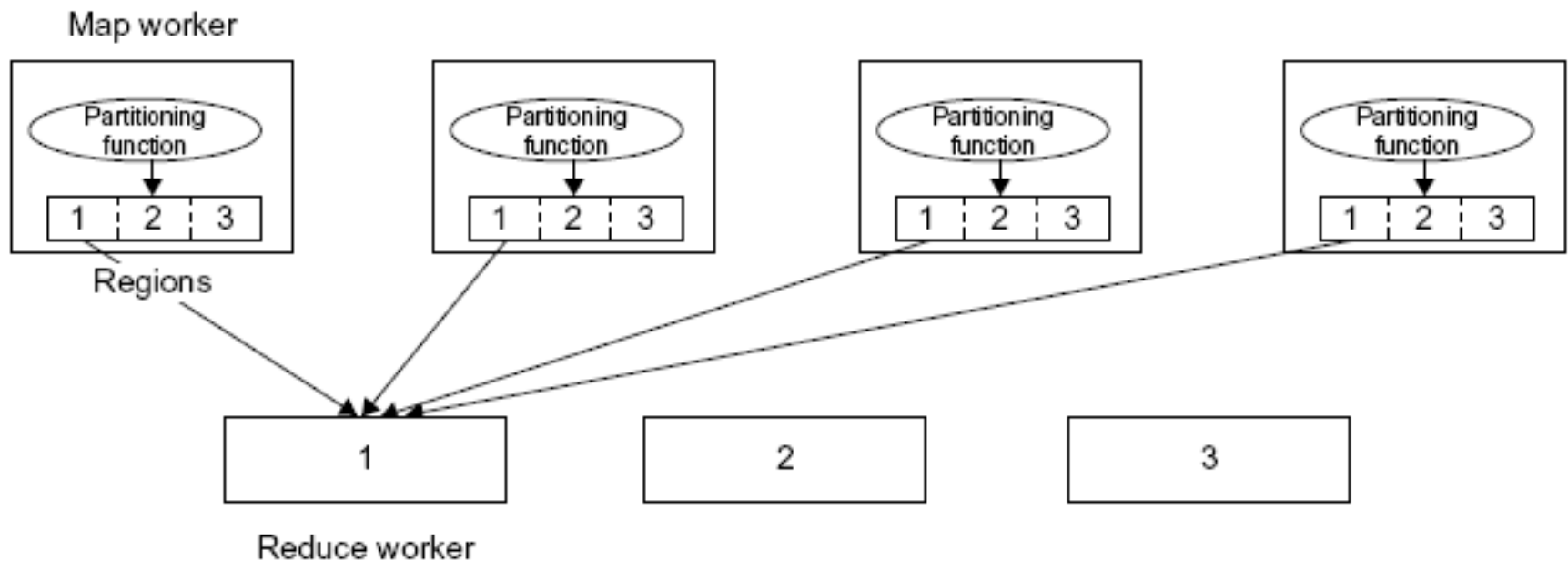
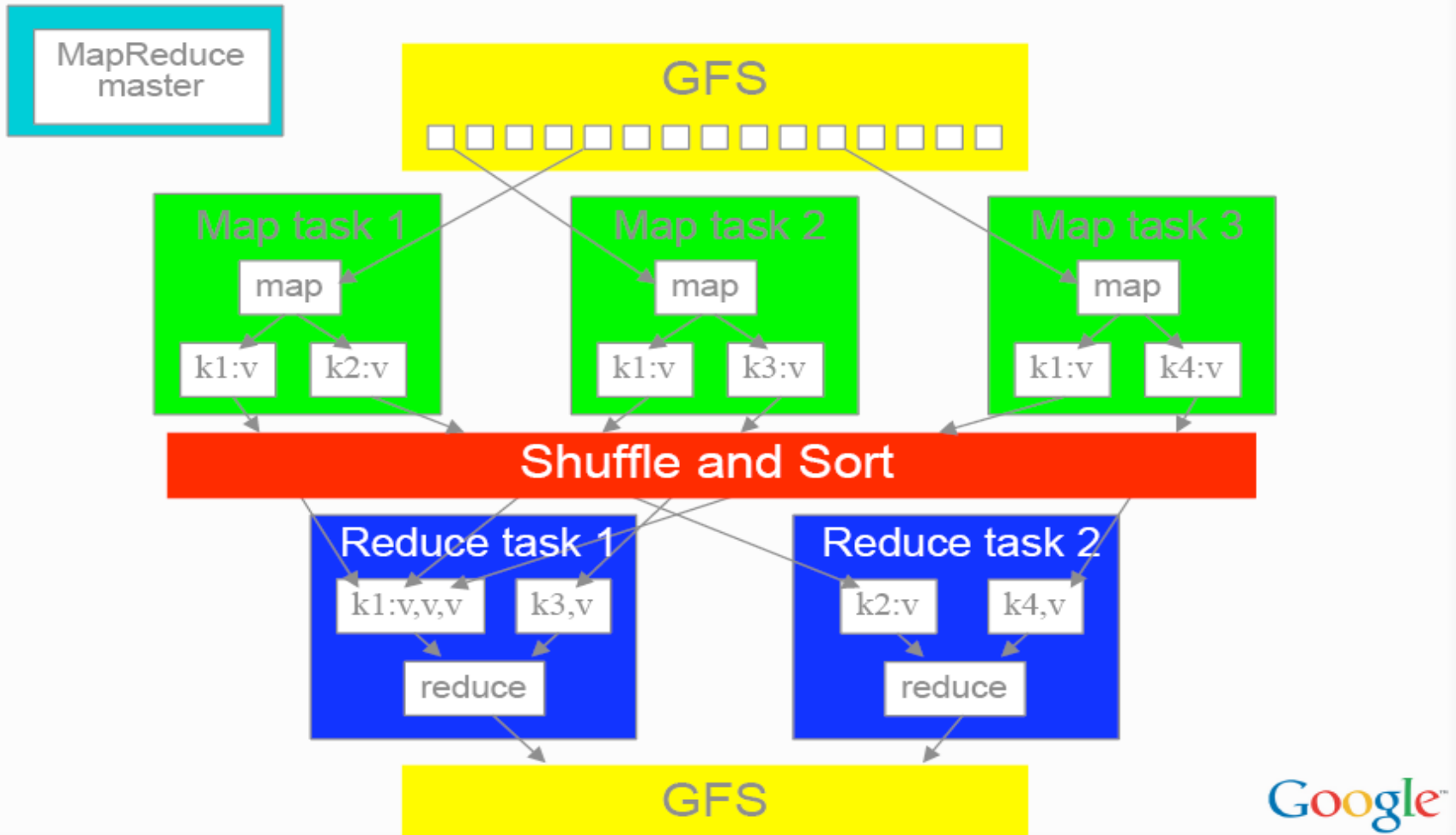


FIGURE 6.4

MapReduce *partitioning* function.

MapReduce Execution



(Courtesy of Jeffrey Dean, Google, 2008)

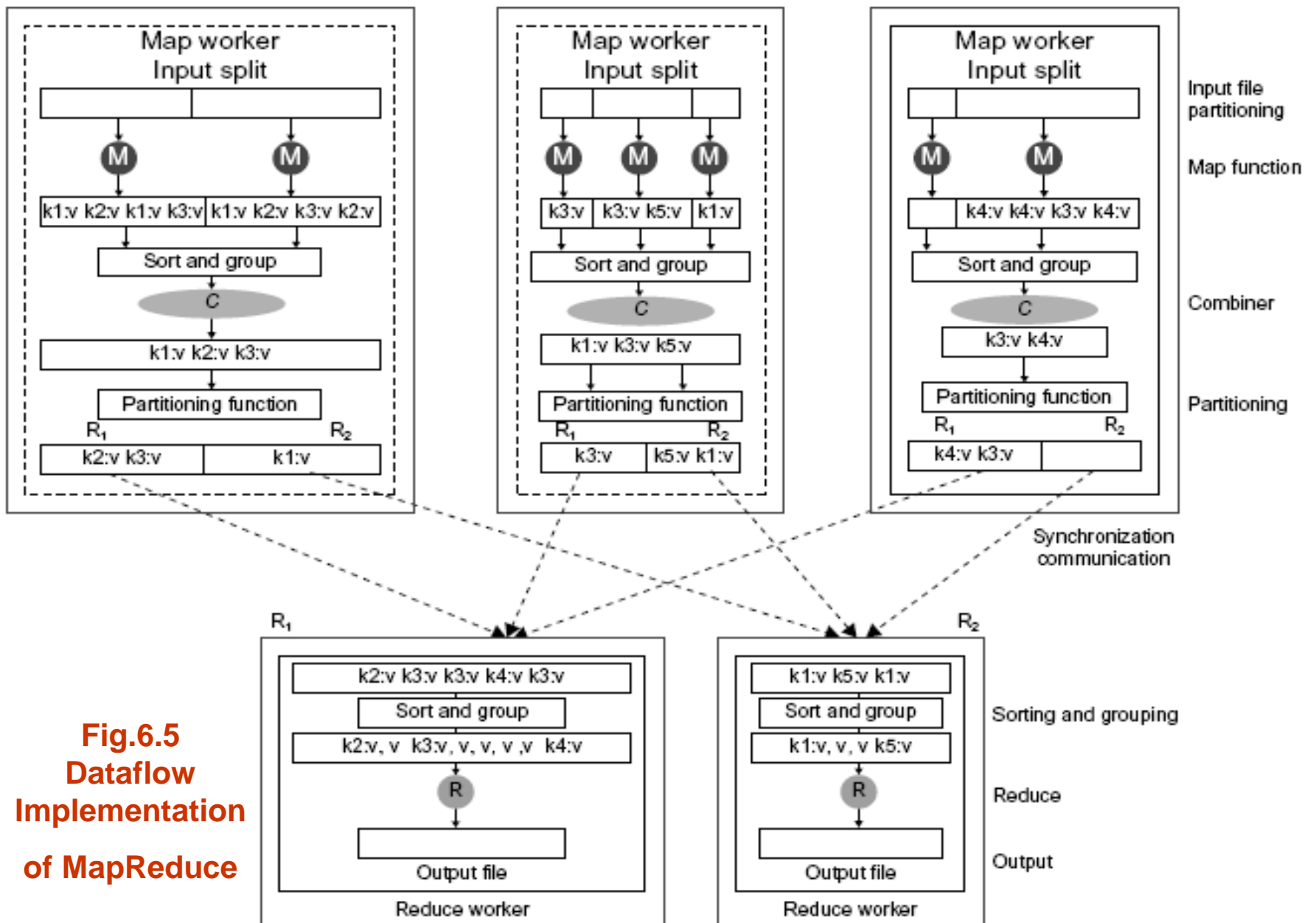
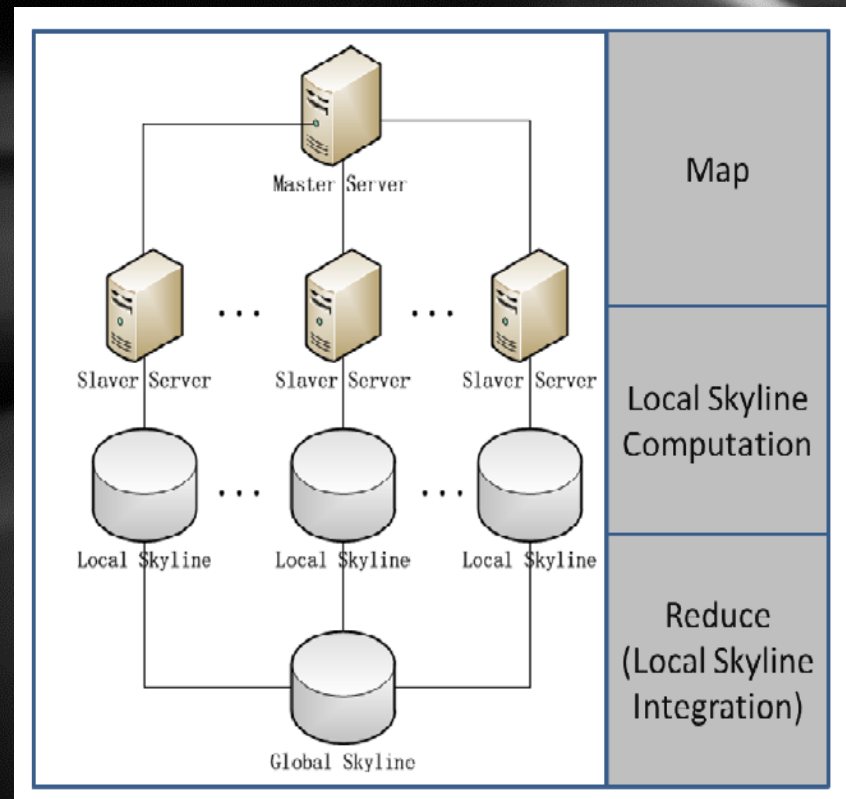
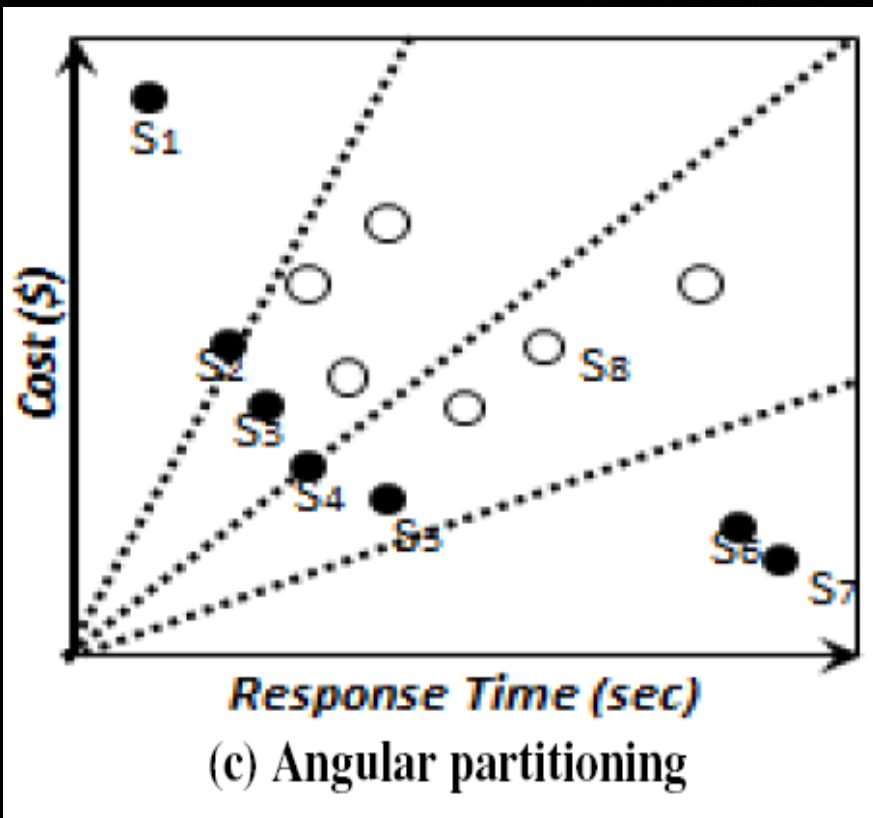


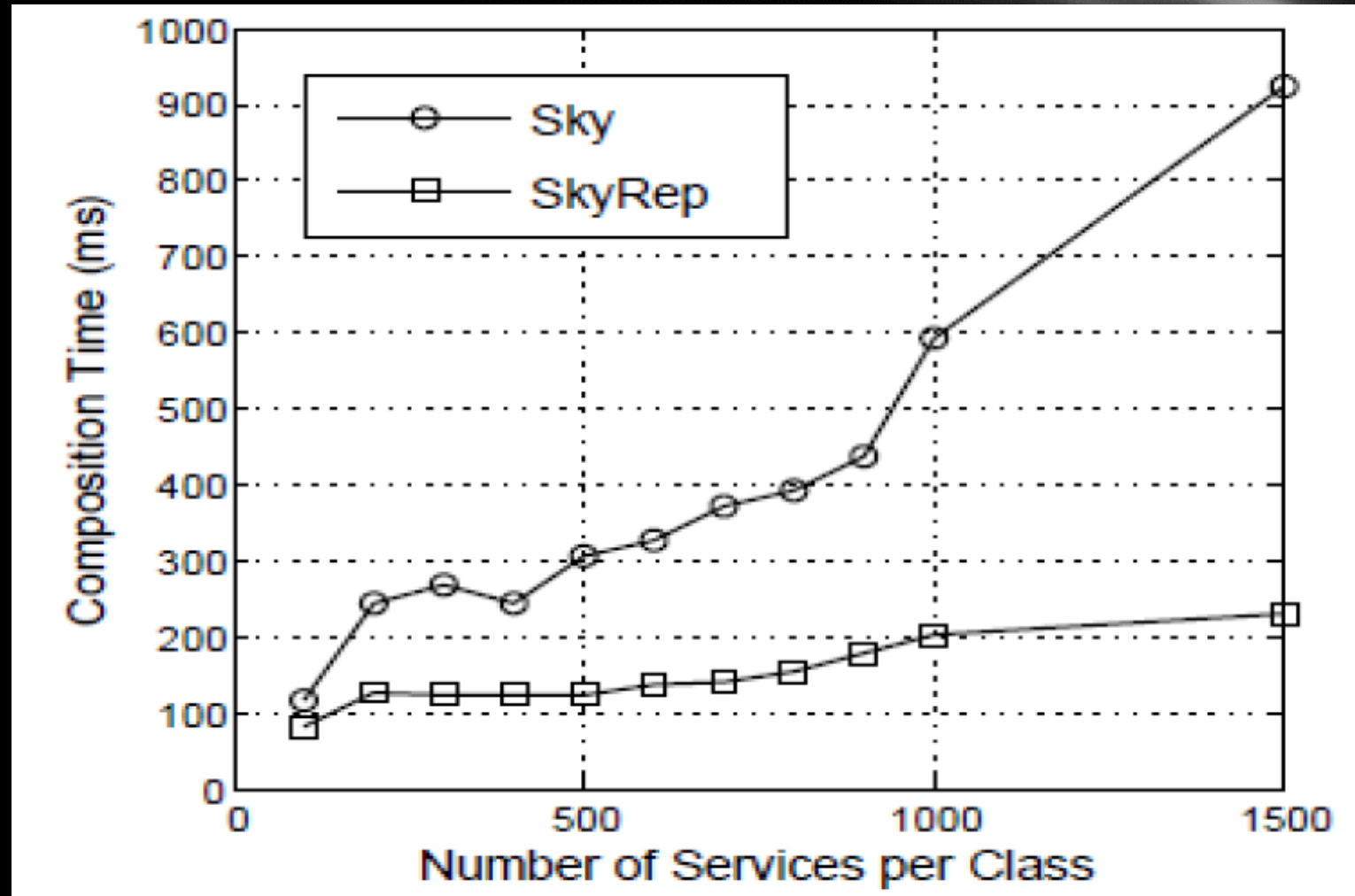
Fig.6.5
Dataflow
Implementation
of MapReduce

Example 2: MapReduce Skyline Composition of Web Services in Inter-Cloud Applications



(Courtesy of L. Chen, K. Hwang, and J. Wu, Jan. 2011)
Copyright © 2012, Elsevier Inc. All rights reserved.

Reduction of Web Service Composition Time from 929 ms to 220 ms using fewer Skyline representatives



Hadoop : A software platform originally developed by Yahoo to enable users write and run applications over vast distributed data.

Attractive Features in Hadoop :

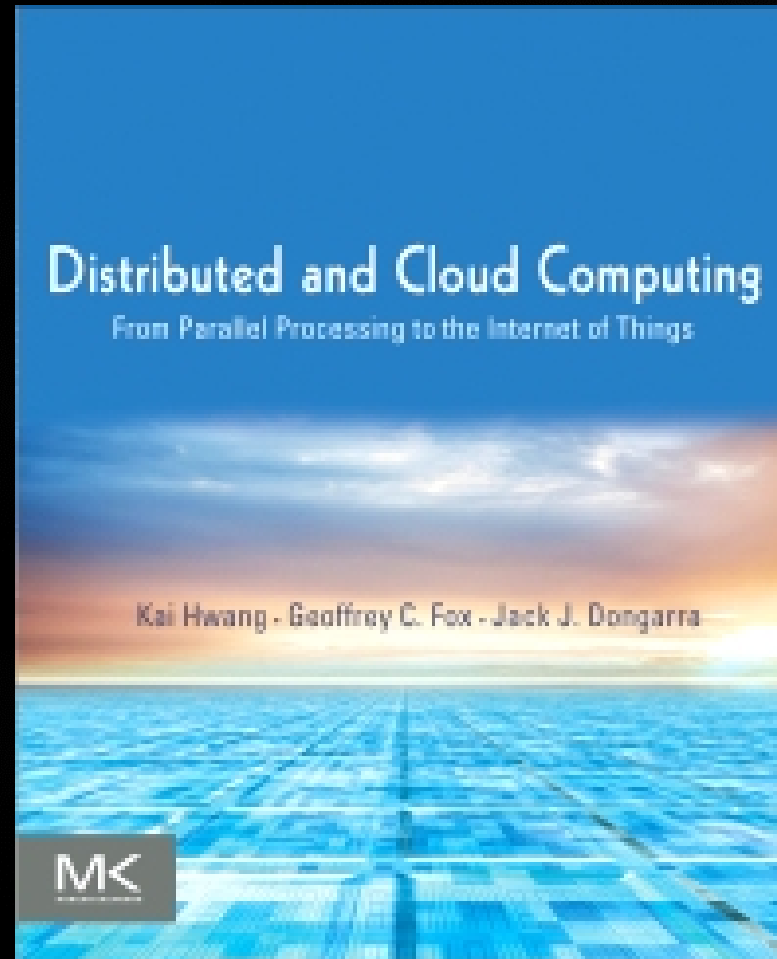
- **Scalable** : can easily scale to store and process petabytes of data in the Web space
- **Economical** : An open-source MapReduce minimizes the overheads in task spawning and massive data communication.
- **Efficient**: Processing data with high-degree of parallelism across a large number of commodity nodes
- **Reliable** : Automatically maintains multiple copies of data to facilitate redeployment of computing tasks on failures

Psi5120 Tópicos de computação em nuvem

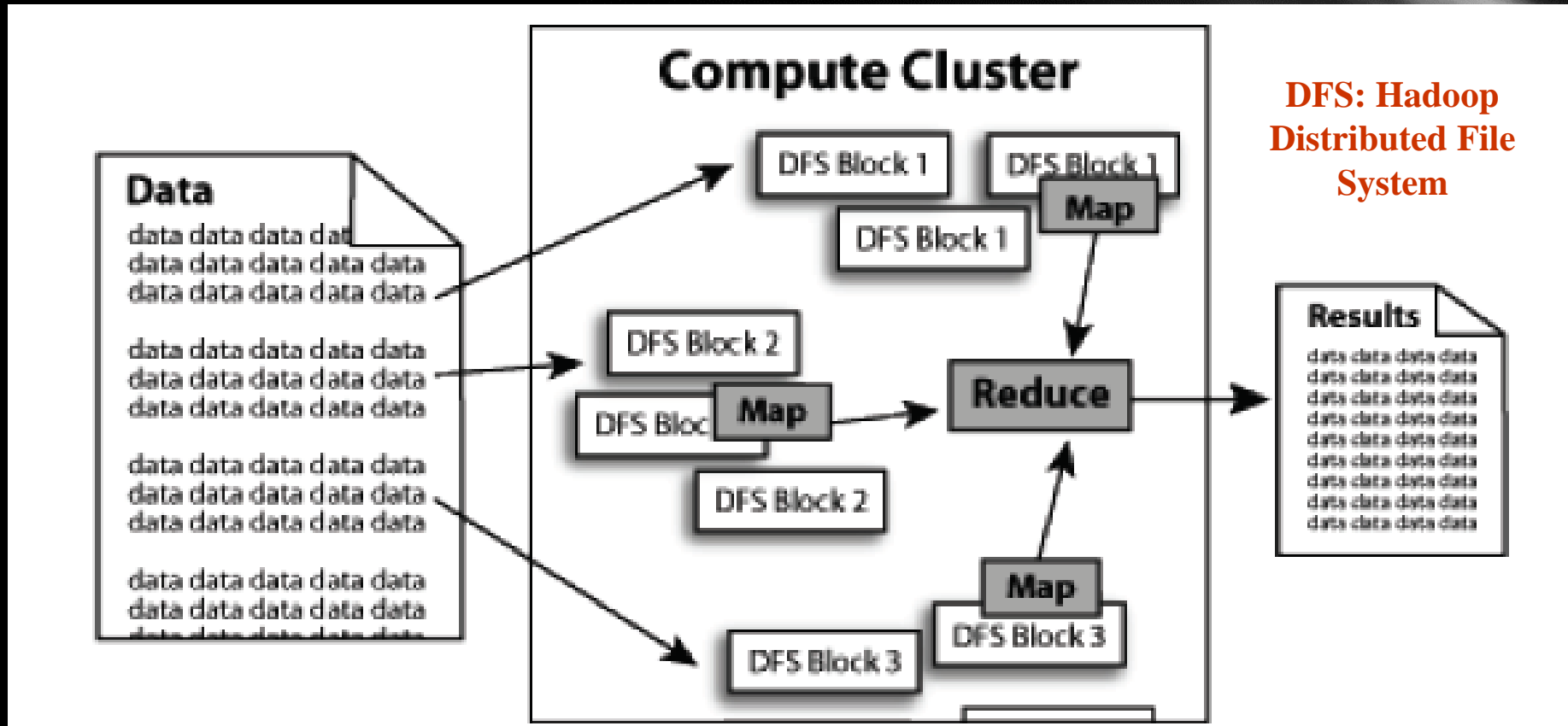
6a. Aula Parte 2

20. Período de 2013

Livro texto



Apache Hadoop Architecture



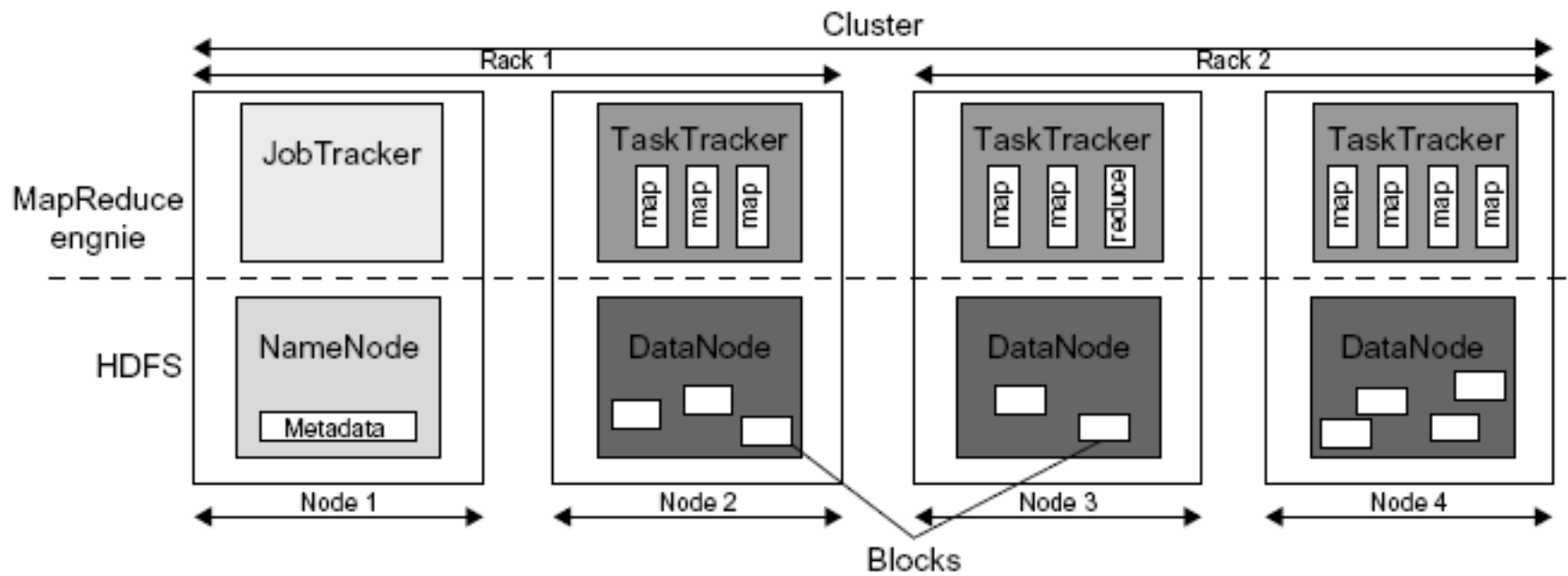


FIGURE 6.11

HDFS and MapReduce architecture in Hadoop.

Secure Query Processing with Hadoop/MapReduce

Query Rewriting and Optimization Principles defined and implemented for two types of data

(i) Relational data: Secure query processing with HIVE

(ii) RDF Data: Secure query processing with SPARQL

Demonstrated with XACML Policies (content, temporal, association)

Joint demonstration with Kings College and U. of Insubria

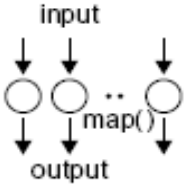
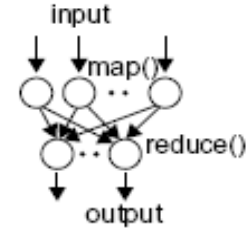
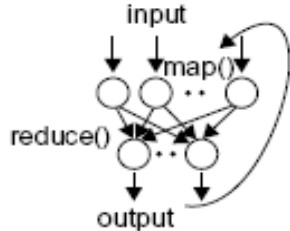
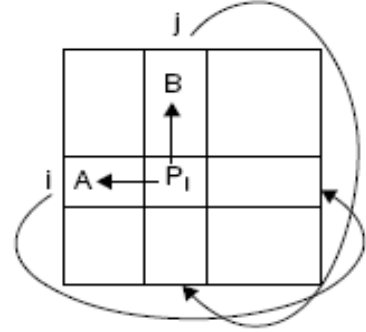
- First demo (2010): Each party submits their data and policies
- Our cloud will manage the data and policies
- Second demo (2011): Multiple clouds

Table 6.7: Comparison of High Level Data Analysis Languages

	Sawzall	Pig Latin	DryadLINQ
Origin	Google	Yahoo	Microsoft
Data Model	Google Protocol Buffer or basic	Atom, Tuple, Bag, Map	Partition File
Typing	Static	Dynamic	Static
Category	Interpreted	Compiled	Compiled
Programming Style	Imperative	Procedural: sequence of declarative steps	Imperative and Declarative
Similarity to SQL	Least	Moderate	A lot!
Extensibility (User defined functions)	No	Yes	Yes
Control Structures	Yes	No	Yes
Execution Model	Record Operations + fixed aggregations	Sequence of MapReduce operations	Directed Acyclic Graphs
Target Runtime	Google MapReduce	Hadoop (Pig)	Dryad

MapReduce and Extensions

Table 6.11 Comparison of MapReduce++ Subcategories along with the Loosely Synchronous Category used in MPI

Map-Only	Classic MapReduce	Iterative MapReduce	Loosely Synchronous
			
<ul style="list-style-type: none"> • Document conversion (e.g., PDF->HTML) • Brute force searches in cryptography • Parametric sweeps • Gene assembly • PolarGrid Matlab data analysis (www.polargrid.org) 	<ul style="list-style-type: none"> • High-energy physics (HEP) histograms • Distributed search • Distributed sort • Information retrieval • Calculation of pairwise distances for sequences (BLAST) 	<ul style="list-style-type: none"> • Expectation maximization algorithms • Linear algebra • Data mining including <ul style="list-style-type: none"> • Clustering • K-means • Deterministic annealing clustering • Multidimensional scaling (MDS) 	<ul style="list-style-type: none"> • Many MPI scientific applications utilizing a wide variety of communication constructs including local interactions • Solving differential equations and particle dynamics with short-range forces
<p>← Domain of MapReduce and Iterative Extensions →</p>			<p>MPI</p>

Next Generation Infrastructure

Truly global systems to span all our datacenters

- Global namespace with many replicas of data worldwide
- Support both consistent and inconsistent operations
- Continued operation even with datacenter partitions
- Users specify high-level desires:
 - “99%ile latency for accessing this data should be <50ms”*
 - “Store this data on at least 2 disks in EU, 2 in U.S. & 1 in Asia”*
- Increased utilization through automation
- Automatic migration, growing and shrinking of services
- Lower end-user latency
- Provide high-level programming model for data-intensive interactive services