CPET 581 Cloud Computing: Technologies and Enterprise IT Strategies

Lecture 8

Cloud Programming & Software Environments

Part 1 of 2

Spring 2015

A Specialty Course for Purdue University's M.S. in Technology Graduate Program: IT/Advanced Computer App Track

Paul I-Hai Lin, Professor

Dept. of Computer, Electrical and Information Technology

Purdue University Fort Wayne Campus

References

1. Chapter 6. Cloud Programming and Software Environments, Book "Distributed and Cloud Computing," by Kai Hwang, Geoffrey C. Fox a,d Jack J. Dongarra, published by Mogan Kaufmman/ Elsevier Inc.

Features of Cloud and Grid Platforms

- Important Cloud **Platform Capabilities**
  - Physical or virtual computing Platform
  - Massive data storage service, distributed file system
  - Massive database storage service
  - Massive data processing method and programming model
  - Workflow and data query language support
  - Programming interface and service deployment (Web interface, special API: J2EE, PHP, ASP, Rails)
  - Runtime support
  - Support services (MapReduce)

Features of Cloud and Grid Platforms

- **Infrastructure Cloud Features**
  - Accounting
  - Appliances (VM,, Message Passing Interface – MPI)
  - Authentication and authorization
  - Data transport
  - Operating systems: Apple, Android, Linux, Windows
  - Program library
  - Registry
  - Security
  - Scheduling
  - Gang scheduling (multiple data-parallel tasks in a scalable fashion; provided automatically by MapReduce)
  - Software as a Service (SaaS)
  - Virtualization

Gang Scheduling Algorithm

References

- Gang Scheduling, http://en.wikipedia.org/wiki/Gang_scheduling
  - In computer science, gang scheduling is a scheduling algorithm for parallel systems that schedules related threads or processes to run simultaneously on different processors.
- Gang Scheduling at SLURM@LLNL, https://computing.llnl.gov/linux/slurm/gang_scheduling.html
  - Support time-sliced gang scheduling
- A gang scheduling design for multi-programmed parallel computing environments, June 15, 2005, http://link.springer.com/chapter/10.1007%2FBFb0022290

Features of Cloud and Grid Platforms

- Cloud Capabilities and Platform Features
- Azure's Platform Features
  - **Azure Table, Queues, Blobs (Binary Large Objects: images, audios, multimedia objects), SQL Database, Web and Worker roles.**
  - **Webinars, http://azure.microsoft.com/en-us/overview/webinars/**
  - **How to Use and Benefits from Azure Virtual Machines with Microsoft's Corey Sanders.**
- Amazon's Platform Features
  - **IaaS, SimpleDB, queues, notification, monitoring, content delivery network, relational database, MapReduce (Hadoop)**
- Google
  - **Google App Engine (GAE)**

Features of Cloud and Grid Platforms

- **Workflow** – links multiple cloud and non-cloud services in real applications on demand.
  - Open Source Workflow Management System
    - **Pegasus – workflow management system, http://pegasus.isi.edu/**
    - **Taverna – workflow management system (Open source & domain independent tools for designing and executing workflows), http://www.taverna.org.uk/**
    - **The Kepler project – open source, scientific workflow application, https://kepler-project.org/**
  - Commercial systems:
    - Pipeline Pilot, http://accelrys.com/products/pipeline-pilot/
    - AVS (Advanced Visual System), Data Visulization http://www.avs.com/
    - LIMS environment (Laboratory Information Management System)
- **Data Transport**
- **Security, Privacy, and Availability**

Features of Cloud and Grid Platforms

- **Data Features and Databases**
  - Program Library
  - Blob and Drives
  - DPFS
    - Google File System (MapReduce)
    - HDFS (Hadoop Distributed File System)
    - Cosmos (Dryal)
  - SQL and Relational Databases
  - Table and NoSQL Non-Relational Databases
  - Queuing Services

Features of Cloud and Grid Platforms

- **Programming and Runtime Support**
  - Worker and Web Roles
  - MapReduce
  - Cloud Programming Model
  - SaaS

Amazon Cloud Computing Products & Services, http://aws.amazon.com/products/

- **Compute**
- **Storage & Content Delivery**
- **Database**
- **Networking**
- **Administration & Security**
- **Analytics**
- **Application Services**
- **Deployment & Management**
- **Mobile Services**
- **Enterprise Application**

Amazon Cloud Computing Products & Services, http://aws.amazon.com/products/

- **Compute:**
  - EC2 – provides resizable compute capacity in the cloud
  - Lambda – a compute service that runs your code in response to events and automatically manages the computer resources for you
  - Auto Scaling
  - Elastic Load Balancing
  - Virtual Private Cloud

Amazon Cloud Computing Products & Services, http://aws.amazon.com/products/

- **Storage & Content Delivery**
  - S3 (Simple Storage Service) – can be used to storage and retrieve any amount of data
  - Glacier – a low-cost storage service that provides secure and durable storage for data archiving and backup
  - EBS (Elastic Block Store)
  - Elastic File System (EFS)
  - Import/Export
  - CloudFront – Provides a way to distribute content to end users with low latency and high data transfer speeds
  - Storage Gateway – securely integrates on-premises IT environments with cloud storage for backup and disaster recovery

Amazon Cloud Computing Products & Services, http://aws.amazon.com/products/

- **Database**
  - RDS – Amazon Relational Database Services (RDS) provides familiar SQL databases while automatically managing administrative tasks
  - ElastiCache – improves application performance by allowing you to retrieve information from an in-memory caching system
  - DynamoDB – Scalable NoSQL data store that manages distributed replicas of your data for high availability
  - Redshift – data warehouse service

Amazon Cloud Computing Products & Services, http://aws.amazon.com/products/

- **Networking**
  - AWS VPC (Virtual Private Cloud)
  - AWS Direct Connect
  - AWS Route 53 (Domain Name System)
  - Elastic Load Balancing
- **Administration & Security**
  - AWS Directory Service
  - AWS Identity and Access Management
  - AWS CloudTrail, AWS Config
  - AWS CloudHSM (Cloud Hardware Security Module)
  - AWS Key Management Service (KMS)
  - AWS Cloud Watch, AWS Trusted Advisor

Amazon Cloud Computing Products & Services, http://aws.amazon.com/products/

- **Analytics**
  - Amazon EMR (Elastic MapReduce)
  - Amazon Kinesis (real-time streaming data ingestion and processing)
  - Amazon Redshift
  - AWS Data Pipeline
  - Amazon Machine Learning
- **Application Services**
  - Amazon SQS (Simple Queue Service)
  - SWF (Simple Workflow Service), AppStream, Elastic Transcoder, SES (Simple Email Service)
  - Amazon CloudSearch, SNS (Simple Notification Service), Flexible Payment Service

Amazon Cloud Computing Products & Services, http://aws.amazon.com/products/

- **Deployment & Management**
  - Elastic Beanstalk, OpsWorks
  - CloudFormation, CodeDeploy
- **Mobile Services**
  - Amazon Cognito, Mobile Analytics
  - SNS (Simple Notification Service)
- **Enterprise Application**
  - Amazon WorkSpaces, WorkDocs
- **AWS Support**
  - Trusted Advisor
- **AWS Marketplace**

Azure Platform Features

- **Microsoft Azure**: **Services**, http://azure.microsoft.com/en-us/services/
  - **Azure Active Directory, API Management, Application Insights, App Service, Automation,**
  - **Backup, Batch, BizTalk Services (B2B, EAI capabilities)**
  - **CDN (Content Delivery Network), Cloud Services**
  - **Data Factory, DocumentDB, Event Hubs, ExpressRoute**
  - **HDInsight, Key Vault**
  - **Machine Learning, Managed Cache, Media Services, Mobile Management, Mobile Services, Multi-Factor Authentication**
  - **Notification Hubs**
  - **Operational Insights, Redis Cache, Remote App, Scheduler**
  - **Azure Search, Service Bus, Site Recovery, SQL Database, Storage, StorSimple, Stream Analytics, Traffic Manager**
  - **Virtual Machines, Virtual Network, Visual Studio Online**

Azure Platform Features

- **Microsoft Azure**: **Services**, http://azure.microsoft.com/en-us/services/ (2015/4/9)
  - Compute:
  - Web & Mobile:
  - Data & Storage
  - Analytics
  - Internet of Things
  - Networking
  - Media & CDN
  - Hybrid Integration
  - Identity & Access Management
  - Developer Services
  - Management

Azure Platform Features

- Microsoft Azure: **Services**, http://azure.microsoft.com/en-us/services/ (2015/4/9)
- **Compute**:
  - Virtual Machines, Cloud Services, Batch, RemoteApp
- **Web & Mobile**:
  - App Service, Web App, Mobile App, Logic App, API Management, Notification Hubs, Mobile Engagement
- **Data & Storage**
  - SQL Database, DocumentDB, Redis Cache, Storage, StoSimple, Azure Search
- **Analytics**
  - HDInsight, Machine Learning, Stream Analytics, Data Factory, Events Hubs
- **Internet of Things**
  - Event Hubs, Stream Analytics, Machine Learning, Notification Hubs

Azure Platform Features

- Microsoft Azure: **Services**, http://azure.microsoft.com/en-us/services/ (2015/4/9)
- **Networking**
  - Virtual Network, Express Route, Traffic Manager
- **Media & CDN**
  - Media Services, Content Delivery Network
- **Hybrid Integration**
  - BizTalk Services, Service Bus, Backup, Site Recovery
- **Identity & Access Management**
  - Azure Active Directory, Multi-Factor Authentication
- **Developer Services**
  - Visual Studio Online, Application Insights
- **Management**
  - Preview Portal, Scheduler, Automation, Operational Insights, Key Vault

6.2 Parallel and Distributed Programming Paradigms

- A distributed computing system consisting of a set or networked nodes or workers. The system issues for running a typical parallel program in either a parallel or a distributed manner would include the following:
  - **Partitioning**
  - **Computation partitioning**
  - **Data partitioning**
  - **Mapping**
  - **Synchronization**
  - **Communication**
  - **Scheduling**

MapReduce Framework

- Apache Hadoop 2.6.0 – MapReduce Tutorial, http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
- A software framework that support parallel and distributed computing on large data sets.
- Providing users with two interfaces in the form of two functions
  - Map()
  - Reduce()

MapReduce: Simplified Data Processing on Large Custers, http://research.google.com/archive/mapreduce.html, De. 2004

By Jeffrey Dean and Sanjay Ghemawat

Abstract

MapReduce is a **programming model** and an associated implementation for processing and generating large date sets. Users specify a **map function** that processes a key/value pair to generate a set of intermediate key/value pairs, and a **reduce function** that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in this paper.

Programs written in this functional style are automatically **parallelized and executed on a large cluster of commodity machines**. The run-time system takes care of the details of partitioning the input data, scheduling the program execution across a set of machines, handling ,machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed system to easily utilize the resources of a large distributed system.

MapReduce: Simplified Data Processing on Large Custers, http://research.google.com/archive/mapreduce.html, De. 2004

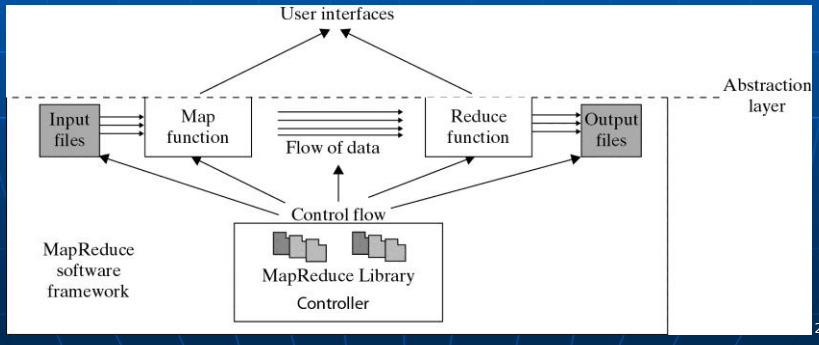
Abstract (continue)

Our implementation of MapReduce runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines. Programmers find the system easy to use: hundreds of MapReduce programs have been implemented and upwards of one thousand jobs are executed on Google's cluster every day.

- Appeared in:
  OSDI'04: Sixth Symposium on Operating System Design and Implementation,
  San Francisco, CA, December, 2004.
- Download: PDF Version
- Slides: HTML Slides

MapReduce Framework

- A software framework that support parallel and distributed computing on large data sets.
- Providing users with two interfaces in the form of two functions: Map(), Reduce()
- Provides an abstraction layer with the data flow and flow of control to users, and hides the implementation of all data flow steps: data partitioning, mapping, synchronization, communication, and scheduling



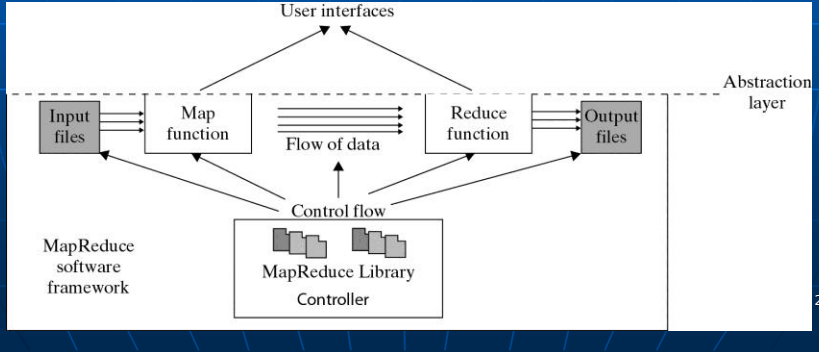
25

MapReduce Framework

Allover structure of a user's program:

- Map Function(….) { … }
- Reduce Function(…) {…}
- Main Function(…)

{ Initialize Spec object
…..
MapRedue(Spec, & Results)
}



26

MapReduce Logical Dataflow

- Map Function's Input and Output
  - **The Input data to the Map function is in the form of a (key, value) pair**
  - **The Output data from the Map function is structured (key, value) pair called Intermediate (key, value) pairs**

```
map(String key, String value):
  // key: document name
  // value: document contents
  for each word w in value:
    EmitIntermediate(w, "1");
```

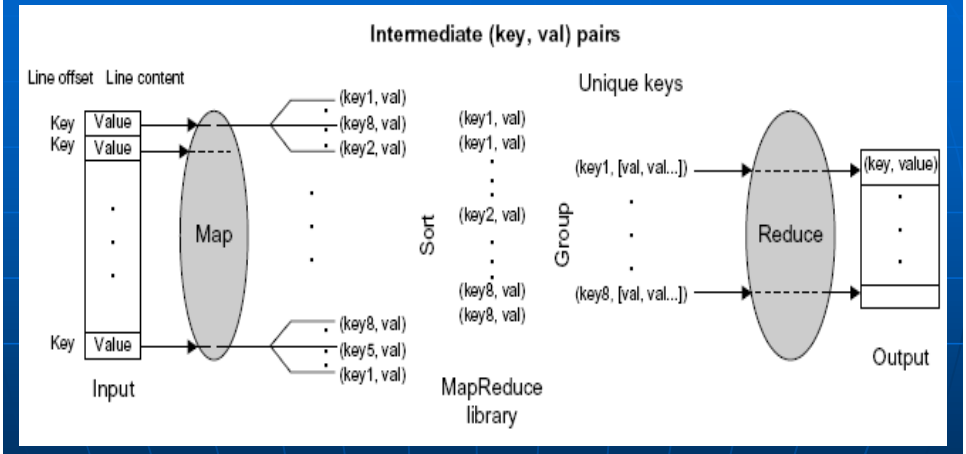
  - **Process all input pairs to the Map function in parallel**
  - **See Figure 6.2**

MapReduce Logical Dataflow

- Reduce function sums together all counts emitted for a particular word

```
reduce(String key, Iterator values):
  // key: a word
  // values: a list of counts
  int result = 0;
  for each v in values:
    result += ParseInt(v);
  Emit(AsString(result));
```

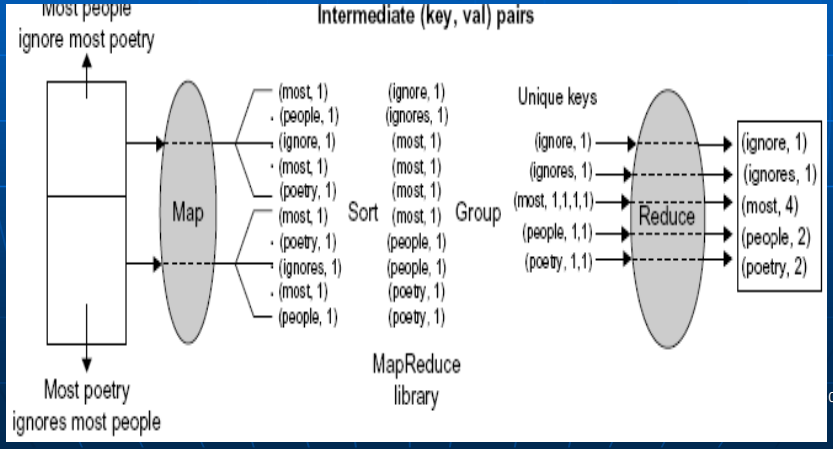
Figure 6.2 Logical Data Flow in 5 Processing Steps in MapReduce Processing Stages



(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.

Figure 6.3 A Word Counting Example on <Key, Count> Distribution

- One well-known MapReduce problem: Word count, to count the number of occurrences of each word in a collection of document.
- A file contains only two lines: (1) Most people ignore most poetry, (2) Most poetry ignores most people



Example: Document Indexing

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

Google Reveals New MapReduce Stats

http://googlesystem.blogspot.com/2008/01/google-reveals-more-mapreduce-stats.html

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

- Google Reveals New MapReduce Stats, http://googlesystem.blogspot.com/2008/01/google-reveals-more-mapreduce-stats.html
- Hadoop on Google Cloud Platform, https://cloud.google.com/hadoop/running-a-mapreduce-job

Figure 6.4 Use of MapReduce partitioning function to link the Map and Reduce workers

- MapReduce Actual Data and Control Flow: 1) Data partitioning, 2) Computation partitioning, 3) Determining the master and workers, 4) Reading the input date (data distribution), 5) Map function, 6) Combiner function, 7) Partitioning Function

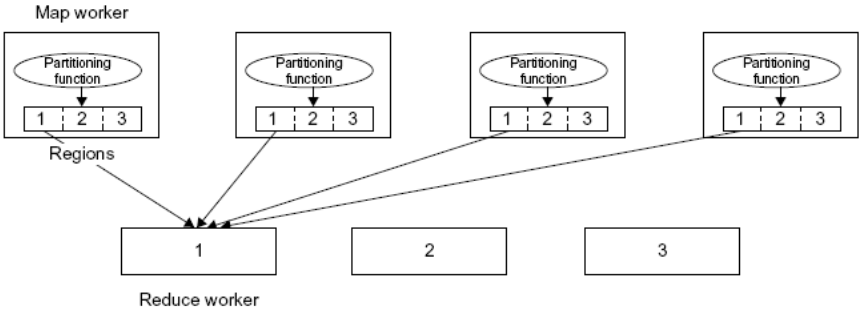


FIGURE 6.4

MapReduce *partitioning* function.

MapReduce Actual Data and Control Flow

1. Data Partitioning
2. Computation Partitioning
3. Determining the Master and Workers
4. Readings the Input Data (data distribution)
5. Map function
6. Combiner function
7. Partitioning function
8. Synchronization
9. Communication
10. Sorting and Grouping
11. Reduce function

Figure 6.5 Data flow implementation of many functions in the Map workers and the reduced workers through multiple sequence of partitioning, combining, synchronization and communication, sorting and grouping, and reduce operations

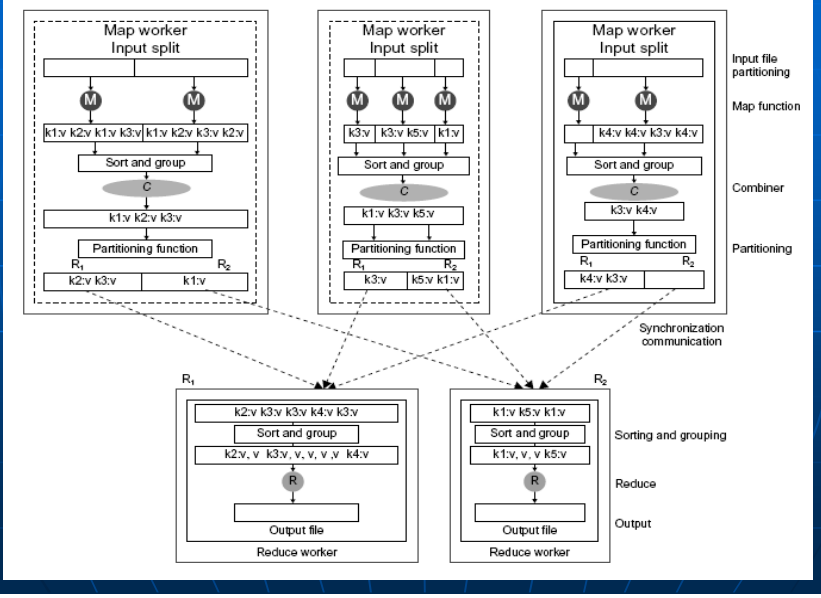


Figure 6.6 Control Flow Implementation of MapReduce

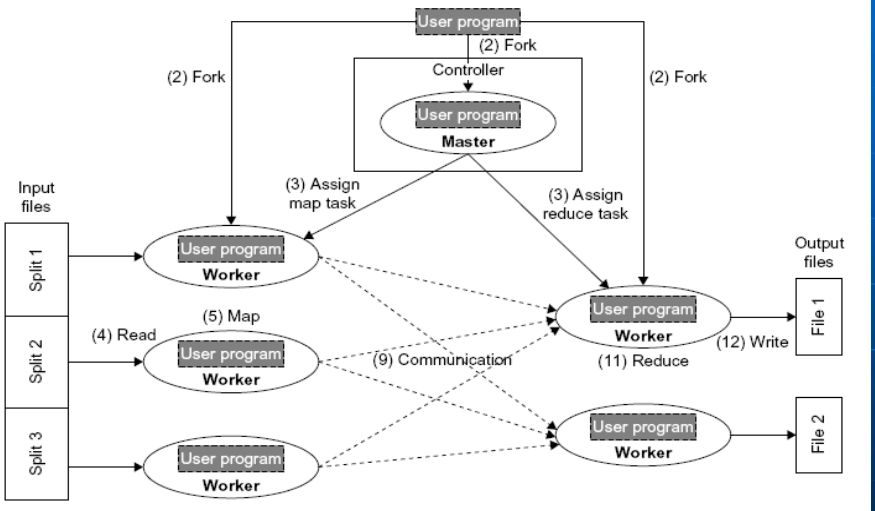


FIGURE 6.6

Control flow implementation of MapReduce.

(Courtesy of Yahoo! Pig Tutorial [54])

Table 6.5 Comparison of MapReduce Type Systems

	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/Subscribe messaging	Files, TCP

Google MapReduce

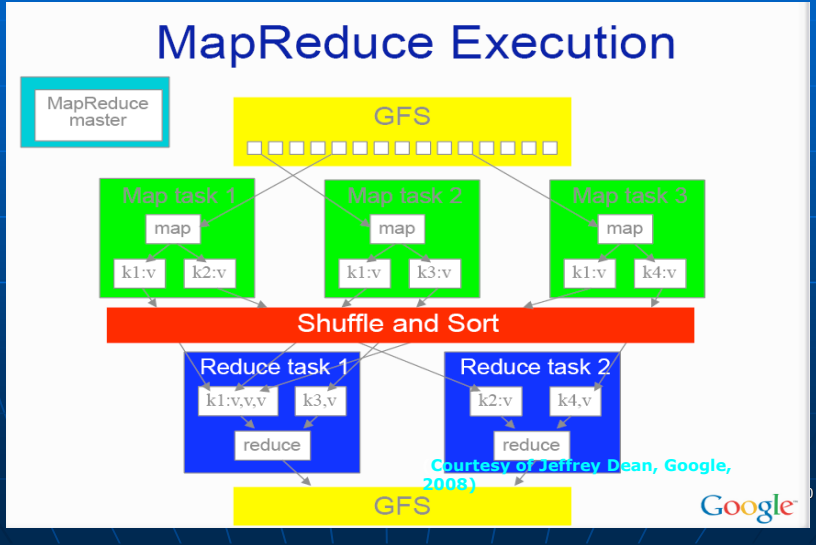
- The MapReduce software framework was first proposed and implemented in C language by Google.
- Default GFS block size is 64 MB

Data Processing: MapReduce

- Google's batch processing tool of choice
- Users write two functions:
  - Map: Produces (key, value) pairs from input
  - Reduce: Merges (key, value) pairs from Map
- Library handles data transfer and failures
- Used everywhere: Earth, News, Analytics, Search Quality, Indexing, …

Google MapReduce

- The MapReduce software framework was first proposed and implemented in C language by Google.
- Default GFS block size is 64 MB



Hadoop Library from Apache

- A open source implementation of MapReduce coded and released in Java (rather than C) by Apache
- The Hadoop implementation of MapReduce uses the HDFS (Hadoop Distributed File System)
- The Hadoop core is divided into two fundamental layers:
  - The MapReduce engine and HDFS
- A software platform originally developed by Yahoo to enable user write and run applications over vast distributed data.
- Attractive Features in Hadoop:
  - Scalable
  - Economical: an open-source MapReduce
  - Efficient
  - Reliable

References

- **Apache Hadoop, https://hadoop.apache.org**
  - Nov. 18, 2014, release 2.6.0, https://hadoop.apache.org/#Download+Hadoop
  - Hadoop Wiki, https://wiki.apache.org/hadoop/Hadoop2OnWindows
  - MapRaduce Tutorial – Apache Hadoop, https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
  - Apache Hadoop 2.6.0 – MapReduce Tutorial, http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
  - An Introduction to Hadoop with Hive and Pig,
  - http://hortonworks.com/hadoop-tutorial/hello-world-an-introduction-to-hadoop-hcatalog-hive-and-pig/
- HDFS Architecture, https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

References

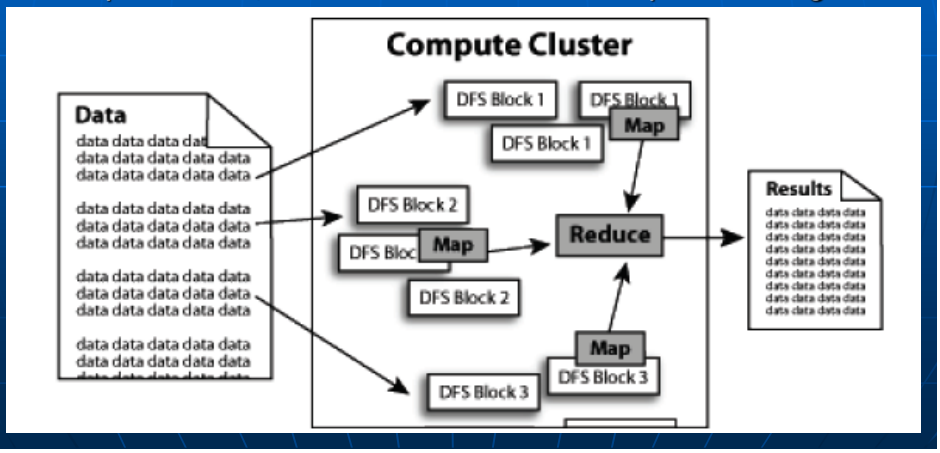
- An Introduction to Hadoop with Hive and Pig, http://hortonworks.com/hadoop-tutorial/hello-world-an-introduction-to-hadoop-hcatalog-hive-and-pig/
- Apache Pig, https://pig.apache.org/
  - Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

References

- Apache Hive, https://cwiki.apache.org/confluence/display/Hive/Home
- The Apache Hive data warehouse software facilitate querying and managing large datasets residing in distributed storage.
- Built on top of Apache Hadoop, it provides
  - **Tools to enable easy data extract/transform/load (ETL)**
  - **A mechanism to impose structure on a variety of data formats**
  - **Access to files stored either directly in Apache HDFS™ or in other data storage systems such as Apache HBase™**
  - **Query execution via MapReduce**

Apache Hadoop Architecture

- HDFS – has a master/slave architecture containing
  - **A single NameNode as the master and**
  - **A number of DataNodes as workers (slaves)**
- HDFS Fault Tolerance: Block replication, Replica replacement, and Heartbeat and blockreport messages



Apache Hadoop Architecture

- HDFS – A master/slave architecture
- HDFS fault tolerance
- HDFS high throughput access to large data sets (files)
- HDFS operation
  - **Reading a file**
  - **Writing a file**

Figure 6.11 HDFS (Hadoop DFS) and MapReduce Architecture

- Top layer: MapReduce engine manages the data flow and control flow of MapReduce jobs over HDFS.
- JobTrackerc- the Master
- A Number of TaskTrackers: Workers (slaves)
  - Manage the execution of Map and/or /Reduce tasks
  - Example: A TrackerNode with N CPUs, each supporting M threads, has M * N simultaneous execution slots

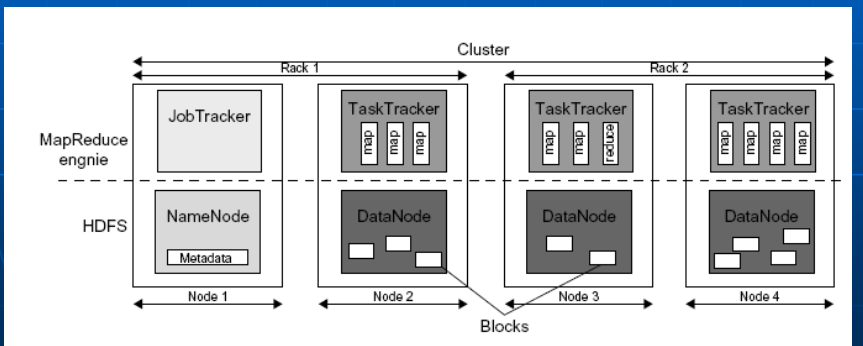
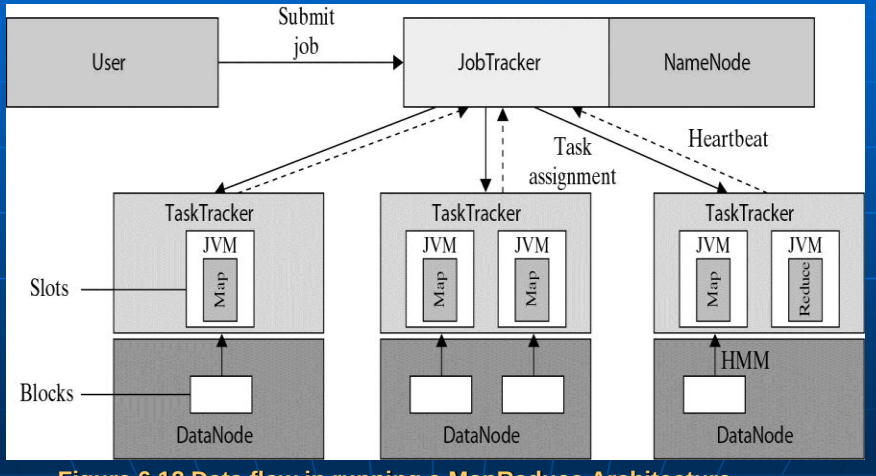


FIGURE 6.11
HDFS and MapReduce architecture in Hadoop.

Running a Job in Hadoop

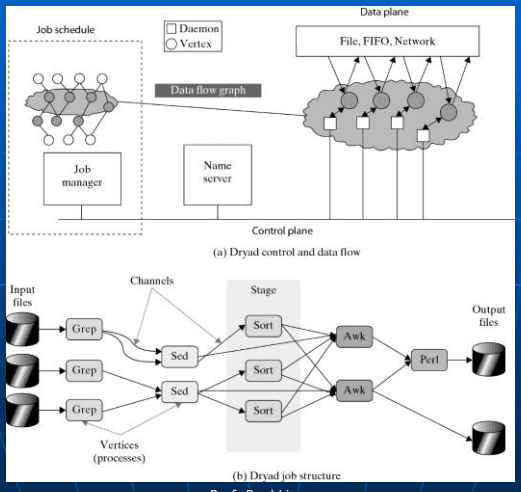
- Data Flow of running a MapReduce job in Hadoop [63]
- Job Submission | Task Assignment | Task Execution | Task Running check



- Figure 6.12 Data flow in running a MapReduce Architecture

Dryad and DryadLINQ (Language Integrated Query Extension) from Microsoft

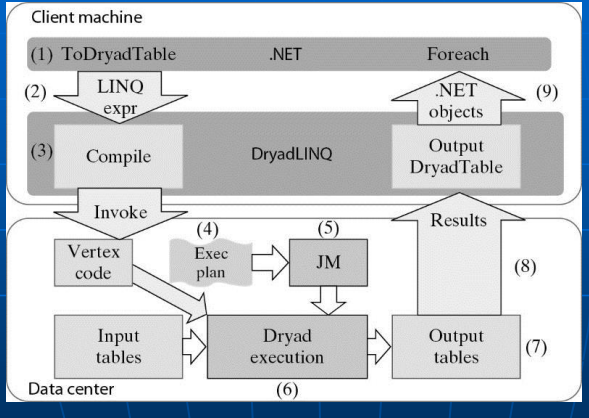
- Microsoft Dryad, http://research.microsoft.com/en-us/projects/dryad/
- Dryad is more flexible than MapRedue
- Dryad program or job is defined by a DAG (directed acyclic graph)



- Figure 6.13 Dyrad framework and its job structure and data flow

Microsoft DryadLINQ (Language Integrated Query)

- Microsoft DryadLINQ project, http://research.microsoft.com/en-us/projects/DryadLINQ/
- JM – Job Manager



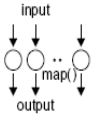
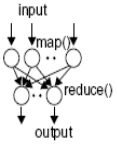
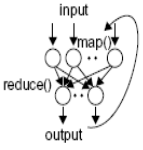
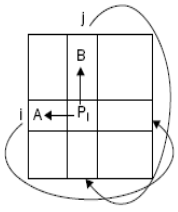
- Figure 6.14 LINQ-expression execution in DryadLINQ

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 Extension

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

Map-Only	Classic MapReduce	Iterative MapReduce	Loosely Synchronous
			

• Document conversion (e.g., PDF->HTML)
• Brute force searches in cryptography
• Parametric sweeps
• Gene assembly
• PolarGrid Matlab data analysis (www.polargrid.org)
	• High-energy physics (HEP) histograms
	• Distributed search
	• Distributed sort
	• Information retrieval
	• Calculation of pairwise distances for sequences (BLAST)
		• Expectation maximization algorithms
		• Linear algebra
		• Data mining including
		  • Clustering
		  • K-means
		  • Deterministic annealing clustering
		  • Multidimensional scaling (MDS)
			• Many MPI scientific applications utilizing a wide variety of communication constructs including local interactions
			• Solving differential equations and particle dynamics with short-range forces

← Domain of MapReduce and Iterative Extensions →			MPI

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