Faculty of Computer
and Information Sciences
Ain Shams University

كلية الحاسبات و المعلومات
جامعة عين شمس

Data Science

Code:

Instructor

Prof. Dr. Abeer M. Mahmoud

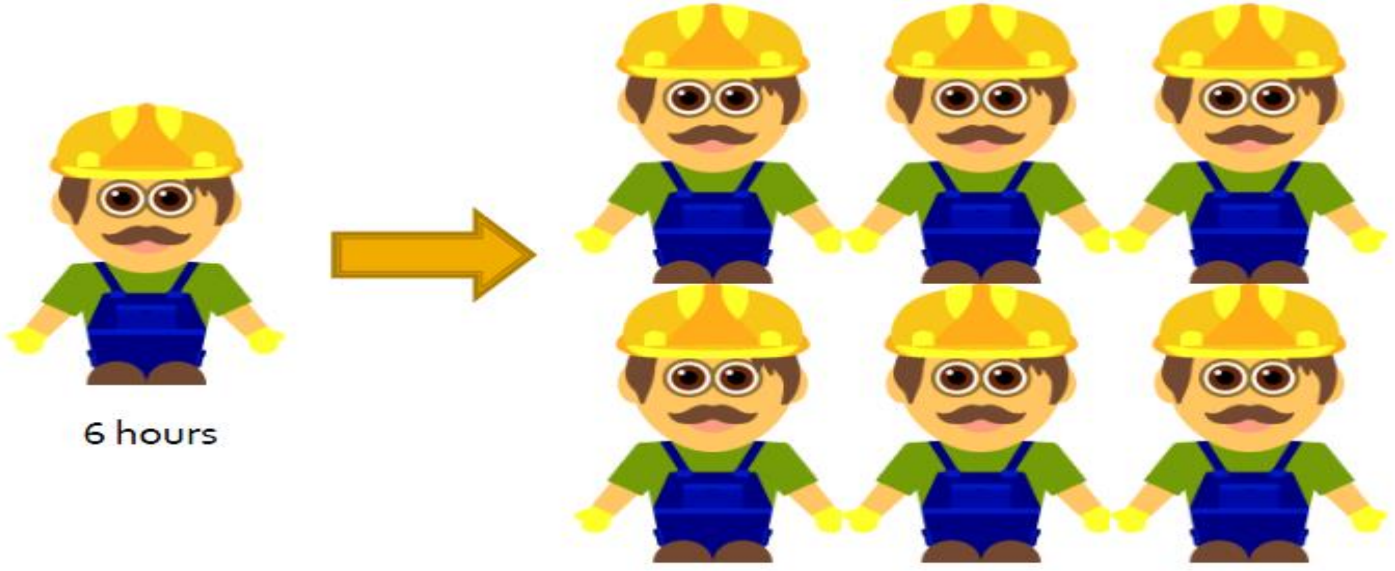
Professor of Computer Science-faculty of Computer and Information Sciences
- Ain Shams University

- What is parallel computing?
- Why you should care?
- Motivation
- Inevitability of parallel computing
  - *Application demands*
  - *Technology and architecture trends*
- How is parallelism expressed in a program
- Challenges
- Sequential VS parallel
- ……..

What is Parallel Computer?

“Informal definition”

- multiple processors cooperating concurrently to solve one problem.



What is Parallel Computer?

"A parallel computer is a collection of processing elements that can communicate and cooperate to solve large problems fast"

Almasi/Gottlieb

"communicate and cooperate"

- Nodes and interconnect architecture
- Problem partitioning and orchestration (Co-ordination of events in a process)

"large problems fast"

- Programming model
- Match of model and architecture

What is Parallel Computer?

Some broad issues:

- Resource Allocation:
  - – how large a collection?
  - – how powerful are the elements?
- Data access, Communication and Synchronization
  - – how are data transmitted between processors?
  - – how do the elements cooperate and communicate?
  - – what are the abstractions and primitives for cooperation?
- Performance and Scalability
  - – how does it all translate into performance?
  - – how does it scale? *A service is said to be scalable if when we increase the resources in a system, it results in increased performance in a manner proportional to resources added*

Why is Parallel Computer?

- ❖ Tremendous advances in microprocessor technology, ex: clock rates of processors increased from 40MHz (e.g MIPS R3000, 1988) to 2.0 GHz (e.g pentium 4, 2002) to nowadays, 8.429GHz (AMD's Bulldozer based FX chips, 2012)

- ❖ Also, processor are now capable of executing multiple instruction in the same cycle

  - ✓ The fundamental sequence of steps that a CPU performs. Also known as the "fetch-execute cycle," it is the time in which a single instruction is fetched from memory, decoded and executed.

  - ✓ The first half of the cycle transfers the instruction from memory to the instruction register and decodes it. The second half executes the instruction

Why is Parallel Computer?

- ❖ Also, ability of memory system to feed data to processor at required rate increased

- ❖ In addition, significant innovations in architecture and software have addressed the alleviation of bottlenecks posed by dataPath and memory

  ✓ Hence, multiplicity of datapaths to increas access to storage elements (memory & disk)

Why Parallel Computing?

Inevitability of parallel computing

Fueled by application demand for performance

- Scientific: weather forecasting, pharmaceutical design, genomics
- Commercial: OLTP, search engine, decision support, data mining
- Scalable web servers

Enabled by technology and architecture trends

- limits to sequential CPU, memory, storage performance
  - o parallelism is an effective way of utilizing growing number of transistors.
- low incremental cost of supporting parallelism

1-Applications demand

Engineering

- Earthquake and structural modeling
- Computation fluid dynamics (airplane design)
- Design and simulation of micro- and nano-scale systems.

Defense

- Nuclear weapons -- test by simulation
- Cryptography

Computational Sciences

- Bioinformatics: Functional and structural characterization of genes and proteins.
- Astrophysics: exploring the evolution of galaxies.
- Weather modeling, flood/tornado prediction..

Commercial

- Data mining and analysis for optimizing business and marketing decisions.
- Database and Web servers for online transaction processing

Computers

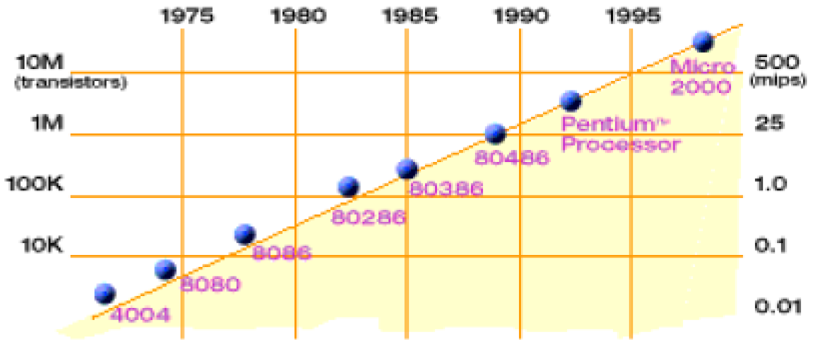
1-Applications demand

- Embedded systems increasingly rely on distributed control algorithms.
- Network intrusion detection, cryptography, etc.
- Optimizing performance of modern automobile.
- Networks, mail-servers, search engines…
- Visualization architectures & entertainment
- Simulation

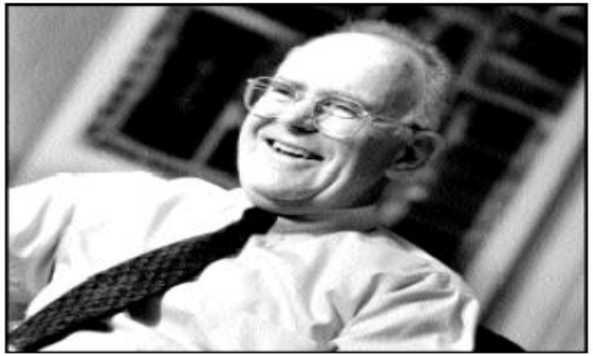
Traditional scientific and engineering paradigm:
1) Do theory or paper design.
2) Perform experiments or build system.
Limitations:
– Too difficult -- build large wind tunnels.
– Too expensive -- build a throw-away passenger jet.
– Too slow -- wait for climate or galactic evolution.
– Too dangerous -- weapons, drug design, climate experimentation.

2-Technology and architecture trends

Technology Trends: *μ-processor Capacity*



Microprocessors have become smaller, denser, and more powerful.

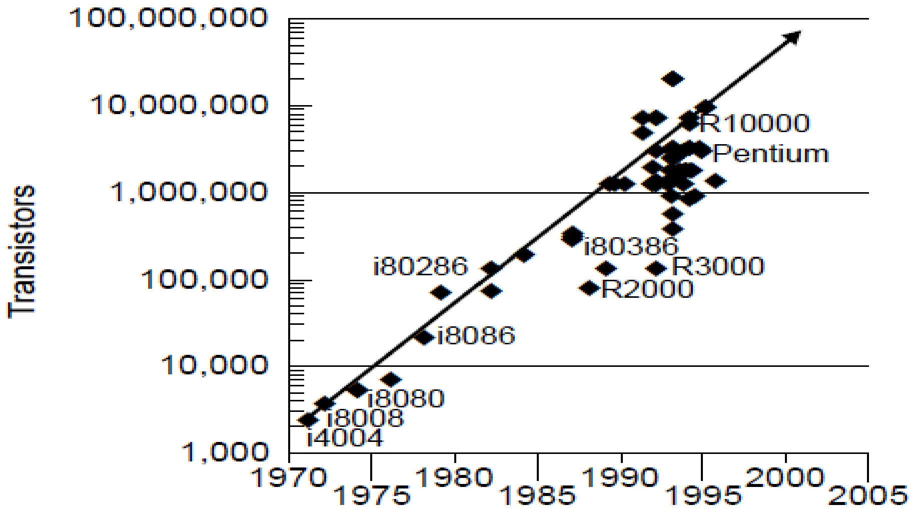


Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months. ➔"Moore's Law"

Slide source: Jack Dongarra

2-Technology and architecture trends

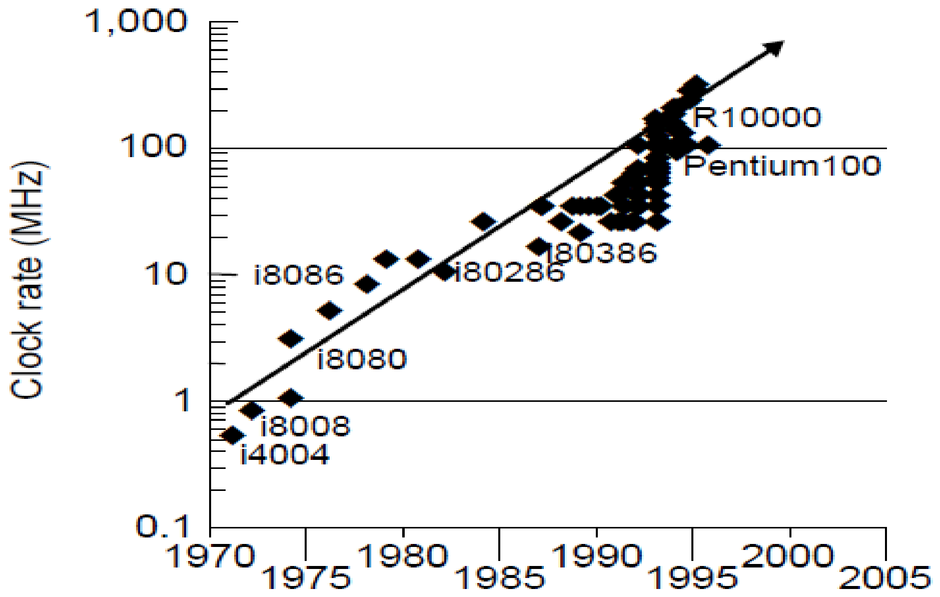
Technology Trends: *Transistor Count*



- 40% more functions can be performed by a CPU per year

2-Technology and architecture trends

Technology Trends: *Clock Rate*



- 30% per year ---> today's PC is yesterday's Supercomputer

Units of Measure in HPC

Definition

High-performance computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly.

units are:

- Flop/s: floating point operations
- Bytes: size of data

Units of Measure in HPC

Typical sizes are millions, billions, trillions…

Mega Mflop/s = 10^{6} flop/sec Mbyte = 10^{6} byte (also 2^{20} = 1048576)

Giga Gflop/s = 10^{9} flop/sec Gbyte = 10^{9} byte (also 2^{30} = 1073741824)

Tera Tflop/s = 10^{12} flop/sec Tbyte = 10^{12} byte (also 2^{40} = 10995211627776)

Peta Pflop/s = 10^{15} flop/sec Pbyte = 10^{15} byte (also 2^{50} = 1125899906842624)

Exa Eflop/s = 10^{18} flop/sec Exa = 10^{18} byte

Parallel Algorithms

History

- Precise mapping of tasks to specific topologies such as Mesh , hypercube

Recently

- Programmability & portability of algorithms , design and implementation

How is parallelism expressed in a program

IMPLICITLY

- define tasks only, rest implied; or define tasks and work decomposition rest implied;
- OpenMP is a high-level parallel programming model, which is mostly an implicit model.

EXPLICITLY

- define tasks, work decomposition, data decomposition, communication, synchronization.

- MPI is a library for fully explicit parallelization.

"It is either All or nothing".

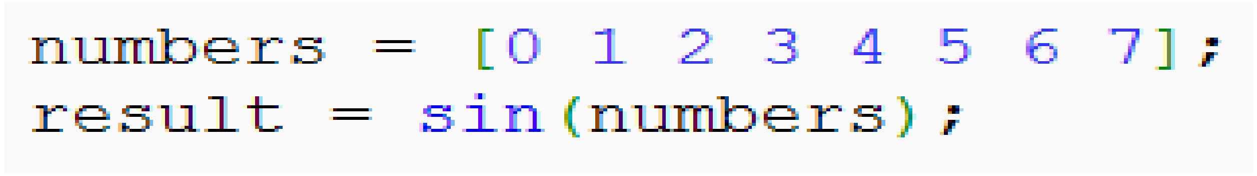
How is parallelism expressed in a program

IMPLICITLY

- It is a characteristic of a programming language that allows a compiler or interpreter to automatically exploit the parallelism inherent to the computations expressed by some of the language's constructs. A pure implicitly parallel language does not need special directives, operators or functions to enable parallel execution.

Programming languages with implicit parallelism include Axum, HPF, Id, LabVIEW, **MATLAB M-code**,

Ex taking the sine or logarithm of a group of numbers, a language that provides implicit parallelism might allow the programmer to write the instruction thus:



```
numbers = [0 1 2 3 4 5 6 7];
result = sin(numbers);
```

How is parallelism expressed in a program

IMPLICITLY

Advantages

- A programmer does not need to worry about task division or process communication, focusing instead in the problem that his or her program is intended to solve.

- It generally facilitates the design of parallel programs .

Disadvantages

- It reduce the control that the programmer has over the parallel execution of the program, resulting sometimes in less-than-optimal parallel efficiency.

- Sometimes mad debugging difficult.

EXPLICITLY

- it is the representation of concurrent computations by means of primitives in the form of special-purpose directives or function calls.
- Most parallel primitives are related to process synchronization, communication or task partitioning.

How is parallelism expressed in a program

EXPLICITLY

Advantages

- the absolute programmer control over the parallel execution.
- A skilled parallel programmer takes advantage of explicit parallelism to produce very efficient code.

Disadvantages

- programming with explicit parallelism is often difficult, especially for non computing specialists, because of the extra work involved in <u>planning the</u> <u>task division and synchronization of concurrent processes.</u>

❖ Current generation of platforms consist of

1. Inexpensive clusters of workstation.
2. Multiprocessor workstations
3. servers

❖ Programming models for these platforms also evolved over time

are accepted as standards to variety of platforms

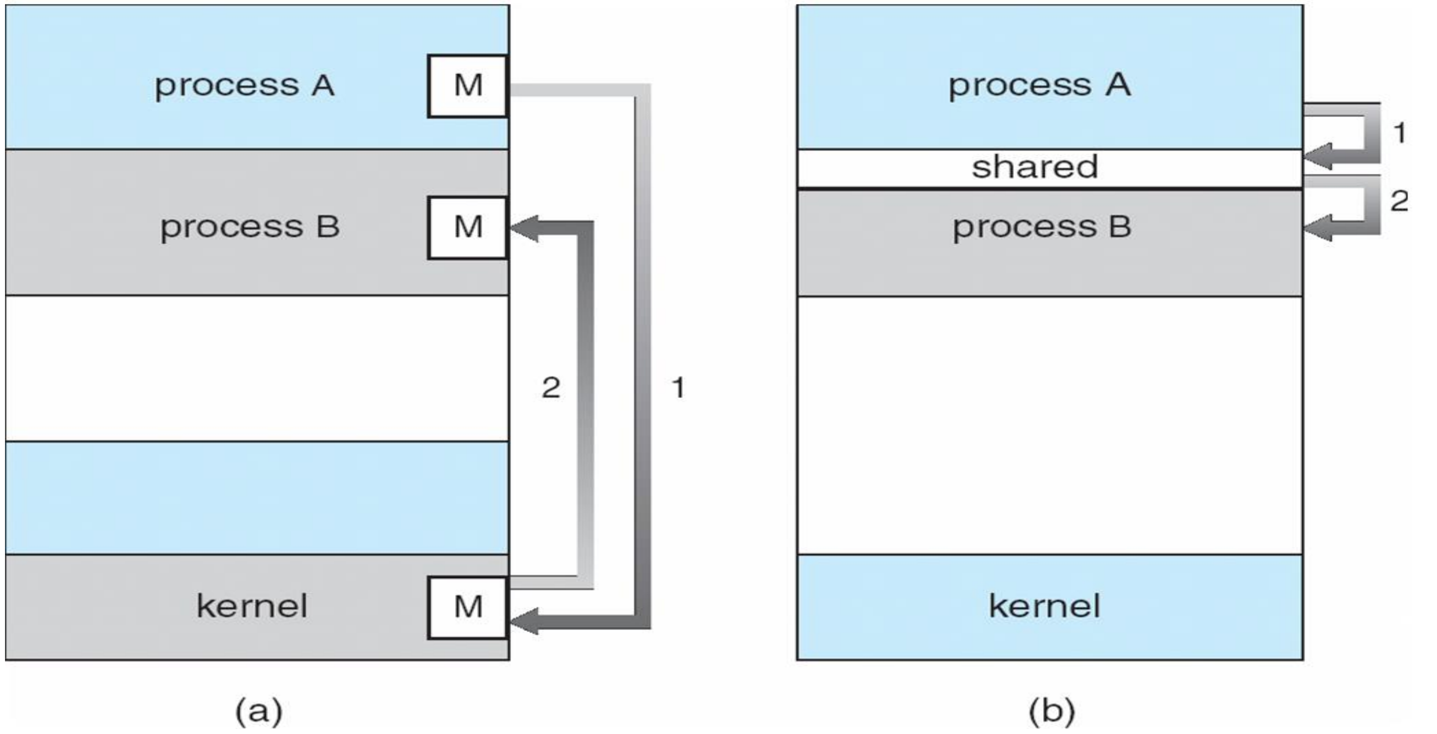
- ✓ Most machines relied on custom APIs for messaging and loop-based parallelism
- ✓ Current models standardize these API across platforms.
- ✓ PVM and MPI (message passing libraries)
- ✓ POSIX threads (thread library)
- ✓ OpenMP (directive based models)

Interprocess Communication (Cont.)

- Two models of IPC: process can exchange information by

Shared memory:	Message passing
• **reading and writing data to the shared region.** • Shared memory is **faster** than message passing …. • Because message passing systems requires more system calls (interrupts) .	• By **messages exchanged** between the cooperating processes. • Useful for exchanging smaller amounts of data, because no conflicts need be avoided. • Easier to implement than shared memory .

Communications Models



Message passing model

Shared memory model

Computer-System Architecture

Computer-System Architecture

- **Single-processor systems :** one CPU

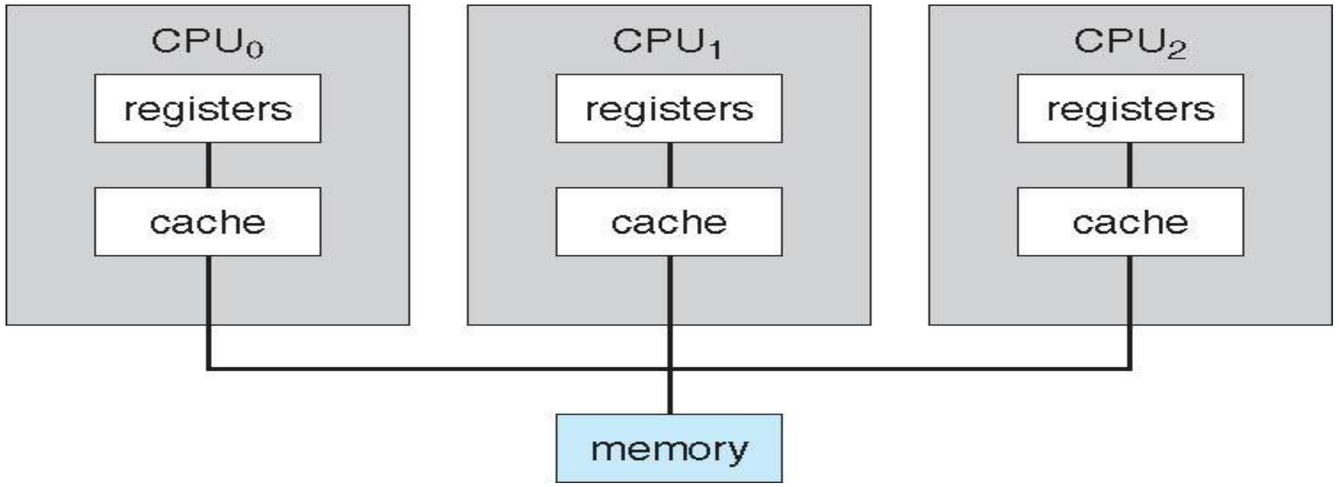
- **MULTIPROCESSOR SYSTEMS**
  - Also known as parallel systems, tightly-coupled systems
  - have two or more processors in close communication, sharing the computer bus and some times the clock, memory, and peripheral devices.
  - **Advantages include:**
    1. Increased throughput : more work is performed
    2. Economy of scale : cost less
    3. Increased reliability : the failure of one processor ***will not halt the system***, only slow it down.

Computer-System Architecture (cont..)

- **MULTIPROCESSOR SYSTEMS – (CONT.):**

Asymmetric Multiprocessing	**Symmetric Multiprocessing (SMP)**
○ Each processor is assigned a specific task. ○ master-slave relationship ○ A **master processor** controls the system; It schedules and allocates work to the **slave processors**.	○ Each processor performs all tasks within the operating system. (peer =equal rank) ○ Each processor has its own set of registers & cache ○ All processors share physical memory.

Symmetric Multiprocessing Architecture



Computer-System Architecture (cont..)

❑ **SMP pros (++):**

– *many processes can run simultaneously —N processes can run if there are N CPUs—*

❑ **SMP Cons (--):**

– OS must carefully control I/O to ensure that the data reach the appropriate processor.

– since the CPUs are separate, one may be sitting **idle** while another is overloaded, resulting in inefficiencies.

- All modern operating systems—including Windows, Mac OS X, and Linux—now provide support for SMP.

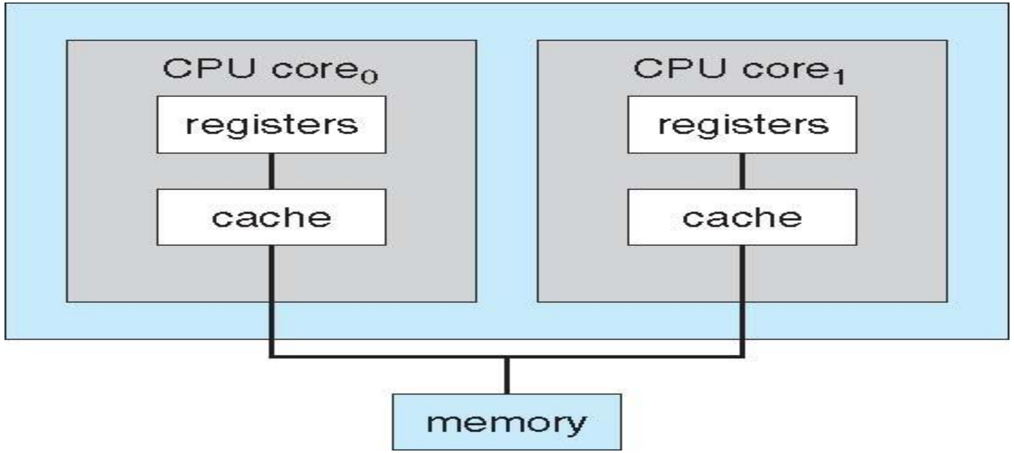
Computer-System Architecture (cont..)

- ○ Processors were originally developed with only one core.
- ○ **The core :** is the part of the processor that actually performs the reading and executing of the instruction.

- ○ Single-core processors can process only one instruction at a time

- **Multi-cores chips.**
  - ○ It is a recent trend in CPU design is to include **multiple computing cores** on a **single chip.**

A Dual-Core Design



- Dual-core processor contains two cores (Such as Intel Core Duo).
- Multi-core systems are especially suited for server systems such as database and Web servers.

Multi-cores chips can be more efficient than multiple chips with single cores because:

1. one chip communication is faster than between-chip communication
2. one chip with multiple cores uses significantly less power than multiple single-core chips.

Framework , System, Model and Programming language??

a **programming language** is a foundation

system is a collection of organized things; a whole composed of relationships among its members

Model is a theoretical concepts that underscore the dimensions and components of empirical investigations.

A framework, or software framework, is **a platform for developing software applications**. It provides a foundation on which software developers can build programs for a specific platform
Think of it as a template of a working program that can be selectively modified by adding code.

DELLTechnologies

Data Science and Big Data Analytics v2

Topics : Data Science and Big Data Analytics Course

Introduction to Big Data Analytics + Data Analytics Lifecycle	Review of Basic Data Analytic Methods Using R	Advanced Analytics – Theory and Methods	Advanced Analytics - Technology and Tools	The Endgame, or Putting it All Together + Final Lab on Big Data Analytics
Big Data Overview State of the Practice in Analytics The Data Scientist Big Data Analytics in Industry Verticals Data Analytics Lifecycle	Using R to Look at Data - Introduction to R Analyzing and Exploring the Data Statistics for Model Building and Evaluation	K-means Clustering Association Rules Linear Regression Logistic Regression Naive Bayesian Classifier Decision Trees Time Series Analysis Text Analysis	Analytics for Unstructured Data (MapReduce and Hadoop) The Hadoop Ecosystem In-database Analytics – SQL Essentials Advanced SQL and MADlib for In-database Analytics	Operationalizing an Analytics Project Creating the Final Deliverables Data Visualization Techniques + Final Lab – Application of the Data Analytics Lifecycle to a Big Data Analytics Challenge

DELL Technologies

Advanced analytics—technology and tools

Advanced analytics—technology and tools

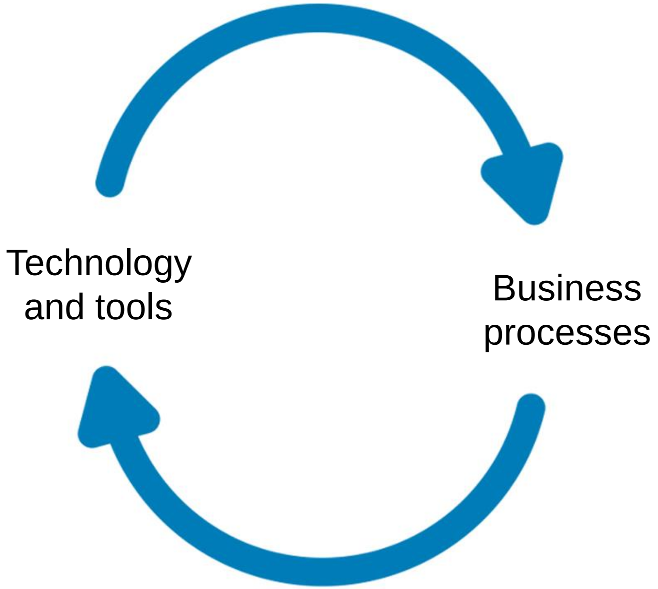
Upon completing this module, you should be able to:

- ✓ Perform analytics on unstructured data using the MapReduce Programming paradigm.
- ✓ Use Hadoop, Hive, and Pig for unstructured data analytics.
- ✓ Effectively use advanced SQL functions and the Greenplum extensions for in-database analytics.
- ✓ Use MADlib to solve analytics problems in-database.

Lesson: Introduction to advanced analytics—technology and tools

Challenges with Big Data beyond analytics

- Infrastructure
  - Storage
  - Backups/restores
  - Compute
  - Network
- Architectural complexities
- Security
  - Data governance
  - Regulatory compliance
- Data quality
- Ethics



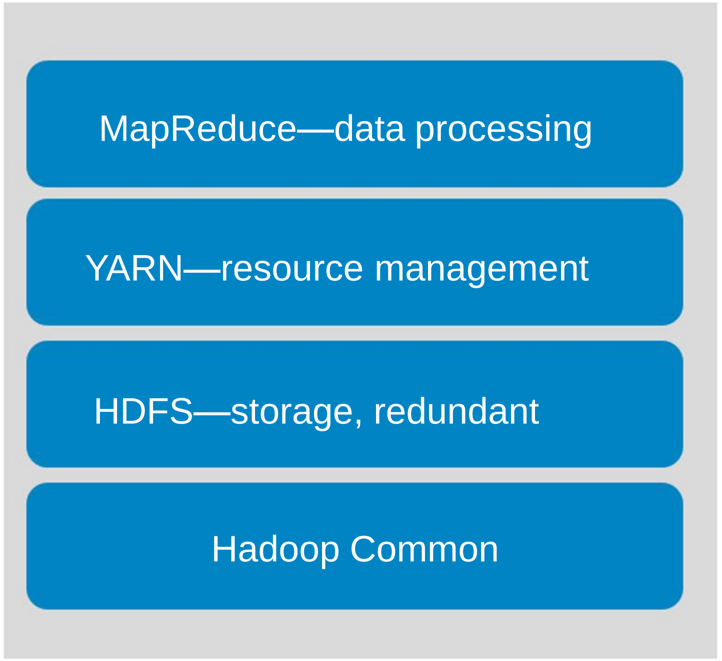
Data governance (DG) is the process of managing the availability, usability, integrity and security of the data

When a business follows state, federal, and international laws and regulations relevant to its operations.

What is Apache Hadoop?

- Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.*
- The Apache Hadoop software library is a framework that allows for the distributed processing of large datasets across clusters of computers using simple programming models.*
- This library enables us to use parallel processing capability to handle huge volumes of data using flexible infrastructure.
- To summarize, Hadoop offers:
  - A scalable, flexible, and reliable distributed computing Big Data framework for a cluster of systems.
  - Storage capacity.
  - Local computing power—applying commodity hardware.
- Hadoop is not:
  - A database.
  - Simply a data warehouse tool. But it can be used as one.
- Hadoop is written in Java™.

Four main components of Apache Hadoop



- MapReduce
- Yet Another Resource Negotiator (YARN™)
- Hadoop Distributed File System (HDFS™)
- Hadoop Common module is a Hadoop Base API —a jar file— for all Hadoop components. All other components work on top of this module.

The components are explained in detail in the next few slides.

Hadoop Distributed File System

- Distributed file system designed to run on commodity hardware for storing large files of data with streaming data access patterns
- Highly fault tolerant
- Default storage for the Hadoop cluster
- File system namespace
- Data/File on HDFS is stored in chunks (128 MB default) called blocks

Hadoop Distributed File System—assumption/goals

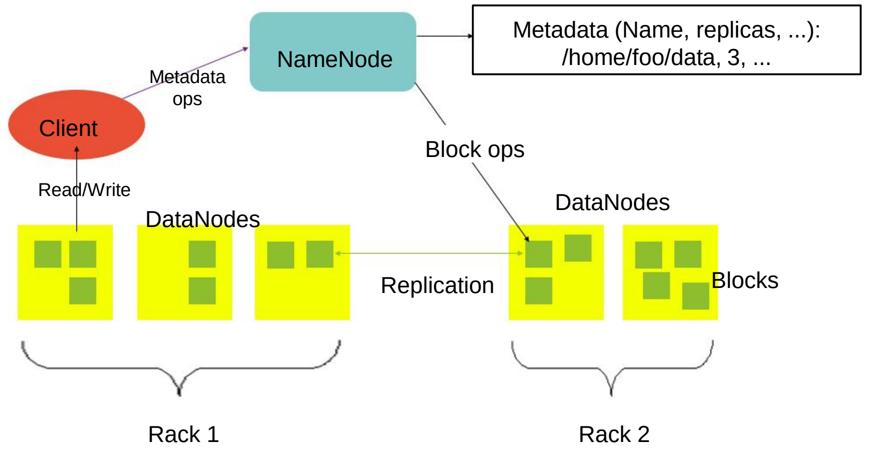
- Hardware failure is a norm rather than an exception
- Streaming data access
- Large dataset processing
- Simple coherency model
- Moving computation is cheaper than moving data
- Portability across heterogeneous hardware and software platforms

Process near nodes

Hadoop Distributed File System—architecture

Uses a master/slave architecture

HDFS clusters can contain multiple NameNodes and some DataNodes



NameNode

- A master server that manages the file system namespace and regulates access to clients.
- NameNode maps the entire file system and metadata structure—such as permissions, modification timestamp, and so on—into memory.
- It executes file system namespace operations such as opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes.
- Information is stored persistently on the disk in the form of two files: namespace image and edit log.
  - **Namespace image** file contains the Inodes and the list of blocks which define the metadata.
  - **Edit log** contains any modifications that have been performed on the content of the image file.

DataNodes

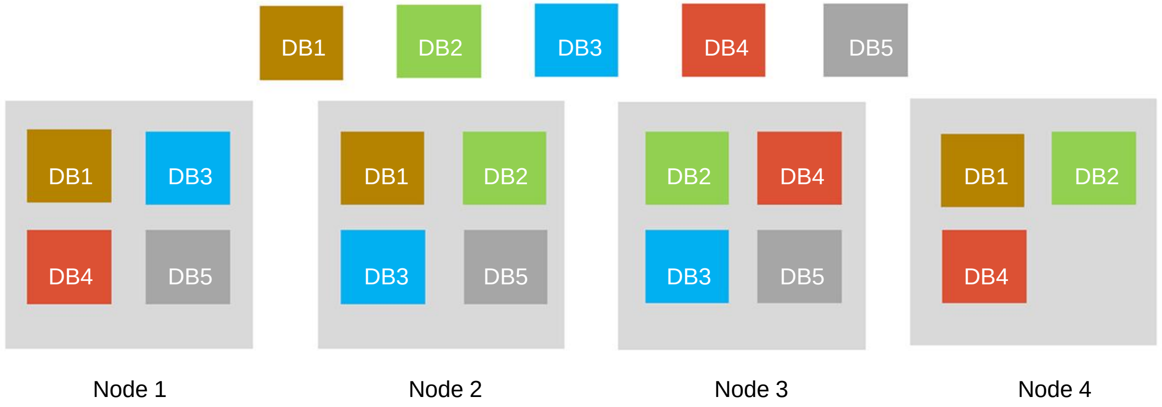
- A file is split into one or more blocks, and these blocks are stored in multiple DataNodes.
- DataNodes are responsible for *serving read* and *write requests* from the file system's clients.
- The DataNodes perform block creation, deletion, and replication upon instruction from the NameNode.
- Reports back to NameNode with the list of stored blocks.
- Bringing computation to data is often more efficient than the reverse.
- DataNodes perform most CPU-intensive and I/O-intensive jobs.

Block replication

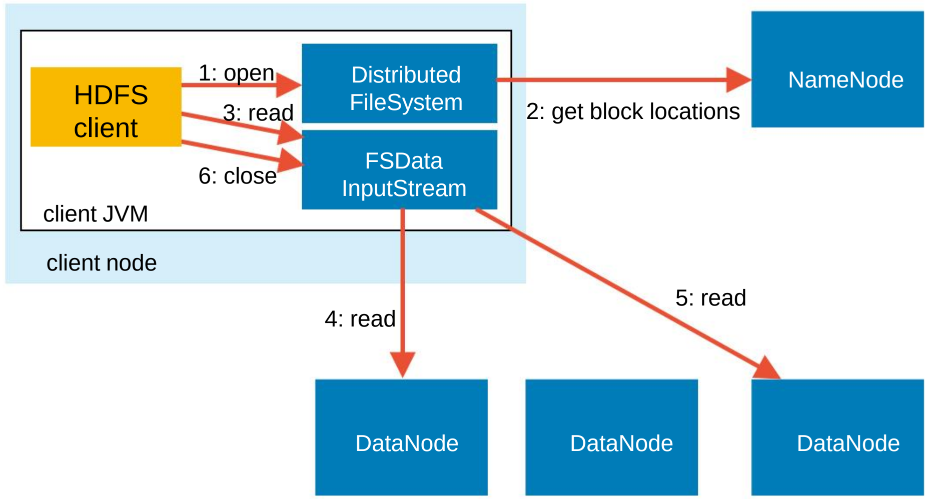
Data is replicated more than once in a Hadoop cluster for fault tolerance and availability.

Every block of data is replicated on more than one node so, even if a node fails, the data is available on another node.

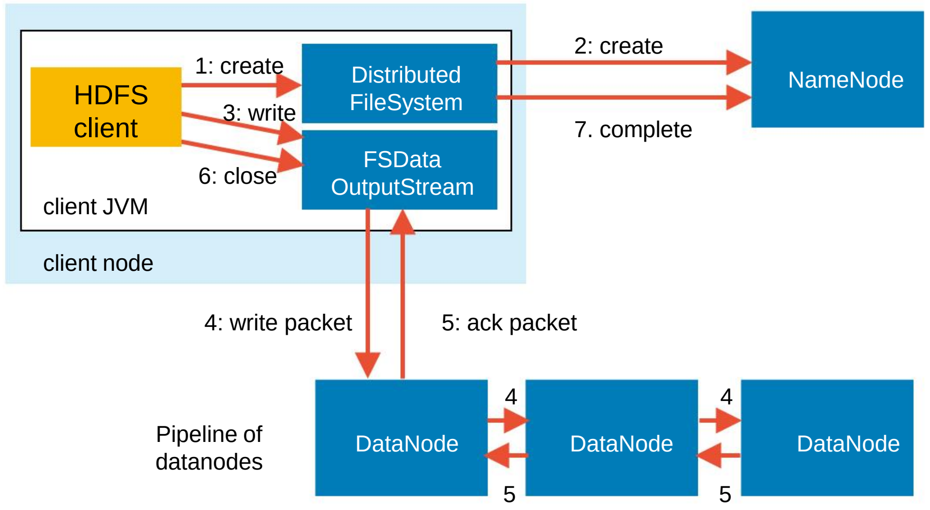
The replication factor is the number of times a block is replicated. The default is 3 for HDFS, which means every block is replicated three times on three different nodes—see example below; DB stands for data block.



Hadoop Distributed File System—file read



Hadoop Distributed File System—file write



Hadoop Distributed File System—not ideal in the following situations

a very high volume of data messages with minimal delay (latency). These networks are designed to support operations that require near real-time access to rapidly changing data.

- Low-latency reads
  - High throughput rather than low latency for small chunks of data
- Large number of small files
  - Better for large files but not for millions of small files
- Multiple writes
  - Single write per file
  - Writes only at the end of the file

Processing versus time

Introduction to MapReduce

"In pioneer days, they used oxen for heavy pulling. When one ox couldn't budge a log, they didn't try to grow a larger ox…

We shouldn't be trying to grow bigger computers, but to add more systems of computers."

- Grace Hopper

The Hadoop ecosystem with the HDFS and MapReduce paradigm helps you add more oxen.

What MapReduce is

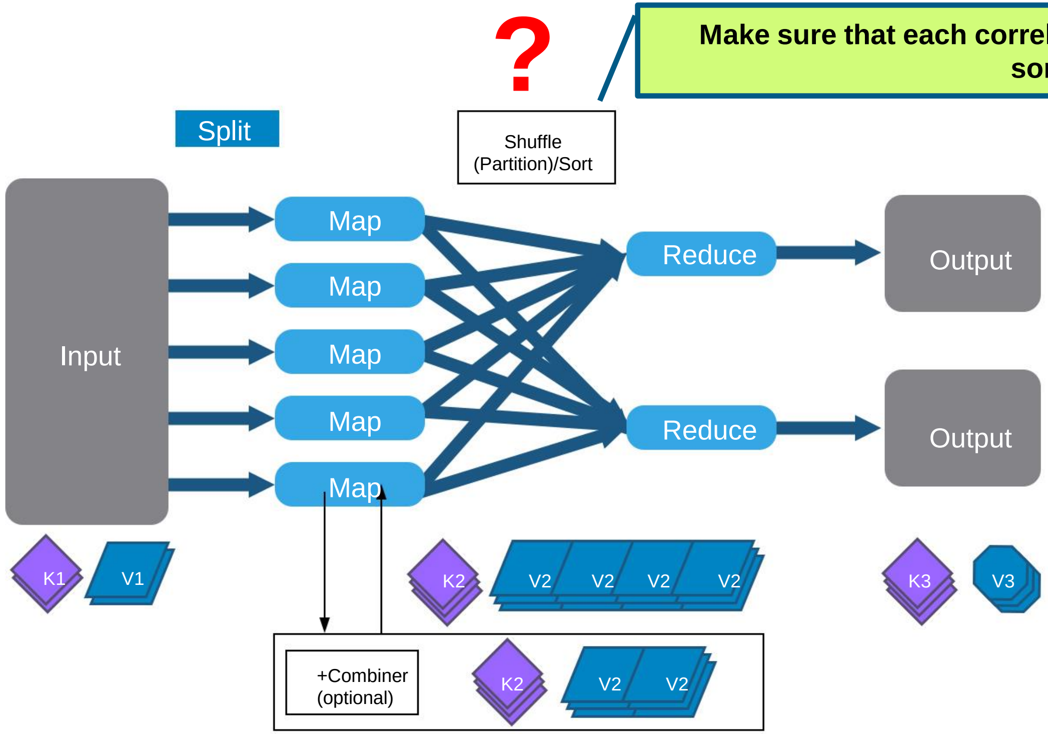
- A software paradigm for writing applications that process vast amounts of data, multi-terabyte datasets, in-parallel on large clusters—thousands of nodes—of commodity hardware in a reliable, fault-tolerant manner
- Java-based programming paradigm
- A combination of the Map and Reduce models that can be applied to wide variety of business cases
- Handles scheduling and fault tolerance

When to use MapReduce

- Problems that are “embarrassingly parallel”
- Examples
  - Word count
  - Reverse index
  - tf-idf
  - Distributed *grep* and distributed object recognition—"Where's Waldo?"
  - Distributed associative aggregation—marginalization, sum; mean if you track both numerator and denominator; min or max; count

MapReduce Steps

Make sure that each correlated keys are grouped and sorted

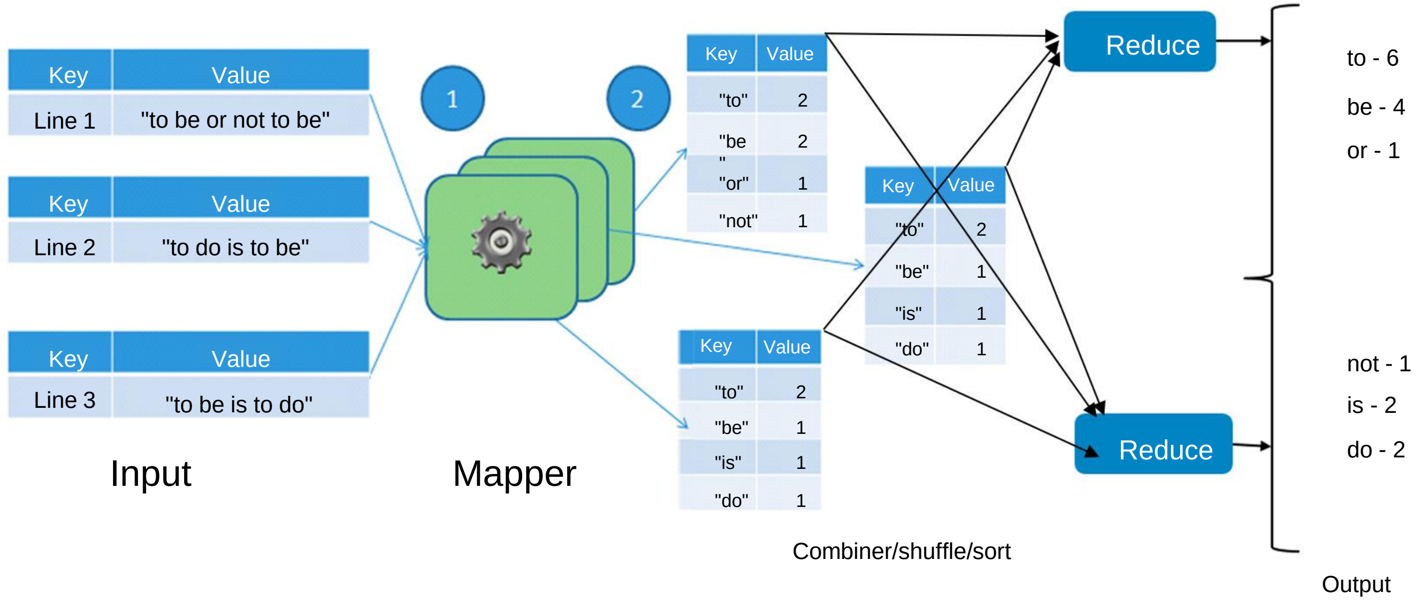


MapReduce paradigm

- Data processing system with two phases—Map and Reduce
- Map
  - Performs a map function on key input key-value pairs to generate intermediate key-value pairs
- Reduce
  - Performs a reduce function on intermediate key-value groups to generate output key-value pairs

- Process
  - Input: a set of key-value pairs
  - User supplies two functions
    - Map $(K_1,V_1) \rightarrow \text{list}(K_2,V_2)$
    - Reduce $(K_2,\text{list}(V_2)) \rightarrow (K_3,V_3)$
  - (K_2,V_2) is an intermediate key-value pair
    - Intermediate key-value groups are created by sorting map output which is input to combiner or reduce function.

MapReduce—count words in document



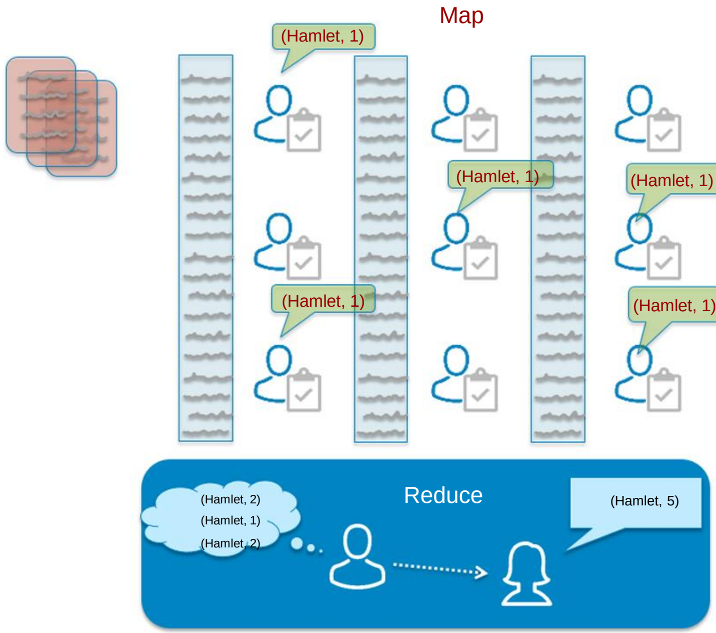
Another word count example

This example is the “Hello, World” of MapReduce.

Distribute the text of millions of documents over hundreds of machines.

Mappers can be word-specific. They run through the stacks and shout “One!” every time they see the word “Hamlet”.

Reducers listen to all the Mappers and total the counts for each word.



Where MapReduce is used—some examples

- Index building in search engines
- Article clustering for news
- Statistical machine translation
- Data mining
- Spam detection
- Ad optimization

Example—social networking—eDiscovery

Problem: An energy company has been indicted. The company has identified 517,424 emails that are circulating within the company that is the cause of this charge.

Objective: Identify all members involved in the conversations that resulted in indictment of the company.

Identify the directionality of the email exchange between each pair.

Date: Thu, 4 May 2000 09:55:00 -0700 (PDT)

From: walt.zimmerman@enron.com

To: michael.burke@enron.com, dana.gibbs@enron.com, lori.maddox@enron.com, susan.ralph@enron.com

Subject: Update on Steve Todoroff Prosecution—CONFIDENTIAL/SUBJECT TO ATTORNEY-CLIENT PRIVILEGE

Cc: steve.duffy@enron.com, stanley.horton@enron.com, jdegeeter@velaw.com

Almost one month ago, Special Agent Carl Wake of the FBI called me about the Steve Todoroff investigation. He indicated that the FBI had recently learned of the article about EOTT's NGL theft that appeared in the business section of the Houston Chronicle. Mr. Wake said it might be a matter the FBI would like to investigate. I told Mr. Wake that EOTT was currently working with the Harris County District Attorney on the prosecution of this matter, and I thanked him for the FBI's interest. He told me that the FBI might want to work with the Harris County District Attorney in investigating this matter, and he stated that there may be investigative information that the FBI can obtain more quickly than the Harris County District Attorney. Mr. Wake requested a copy of the materials we had provided to the Harris County District Attorney.

In order to avoid damage to the good rapport we have established with Assistant District Attorney Bill Moore, I asked John DeGeeter to call Bill Moore and advise him of the contact that had been made by the FBI. Bill Moore agreed to call Carl Wake and work with Mr. Wake on his request for the materials provided by EOTT.

Carl Wake called me again yesterday. He has been working with Bill Moore. Mr. Wake stated it was too early to speculate as to what charges would be brought. He did say that our materials clearly indicated federal wire fraud and possibly mail fraud. He said that where there is wire fraud, there is usually money laundering.

The purpose of Mr. Wake's call yesterday was to inquire about the status of some interview summaries that John DeGeeter and I have prepared and collected at the request of Bill Moore. Mr. Wake requested that EOTT send a copy of the summaries to him when we sent the summaries to Bill Moore. Those summaries were sent out today.

I gathered from my calls with Carl Wake that the FBI is very interested in taking an active part in this investigation. In order to build on the relationship we have established with Bill Moore, we will continue to direct our inquiries about the investigation to Mr. Moore until he tells us to do otherwise.

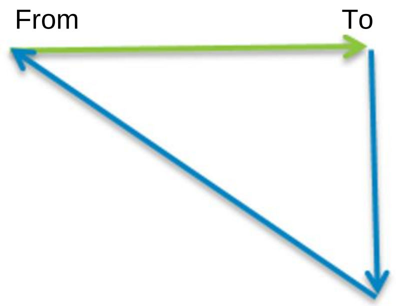
Social triangle—first directed edge

Mapper1

- Maps two regular expression searches:
  - To: Michael, Dan, Lori, Susan
  - From: Walt
- Emits the outbound directed edge of the social graph:
  - <Key, Value> = <Walt, [Michael, Dan, Lori, Susan]>

Reducer1

- Gets the output from the mapper with different values:
  - <Key, Value> = <Walt, [Michael, Dan, Lori, Susan]>
  - <Key, Value> = <Walt, [Lori, Susan, Jeff, Ken]>
- Unites the values for the second directed edge:
  - <Key, Value> = <Walt, [Dan, Jeff, Ken, Lori, Michael, Susan]>



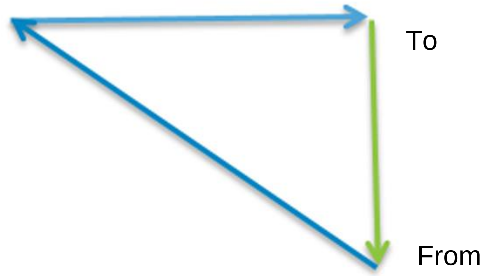
Social triangle—second directed edge

Mapper2

- Reverses the previous map:
  - To: Michael, Dan, Lori, Susan
  - From: Walt
- Emits the inbound directed edge of the social graph:
  - <Key, Value> = <Susan, Walt>; <Lori, Walt>; <Dan, Walt>; and so on

Reducer2

- Gets the output from the mapper with different values:
  - <Key, Value> = <Susan, Walt>
  - <Key, Value> = <Susan, Jeff>
- Unions the values for the third directed edge:
  - <Key, Value> = <Susan, [Jeff, Ken, Walt]>



Social triangle—third directed edge

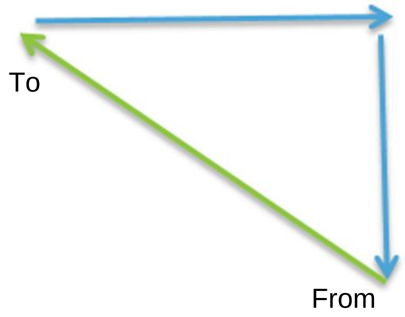
Mapper3

- Join [inbound] and [outbound] lists by key:
  - Walt, [Jeff, Ken, Lori, Susan], [Jeff, Lori, Stanley]
- Emits <Person, Person> pair with level of association:
  - <Key, Value> = <Walt: Jeff reciprocal>; <Walt:Stanley directed>, and so on

Reducer3

- Reducer unions the output of the mappers and presents rules:
  - <Key, Value> = <Walt::Jeff, reciprocal>
  - <Key, Value> <Walt::Stanley, directed>

The third reducer can shape the data any way that serves the business objective.



Hadoop operational modes

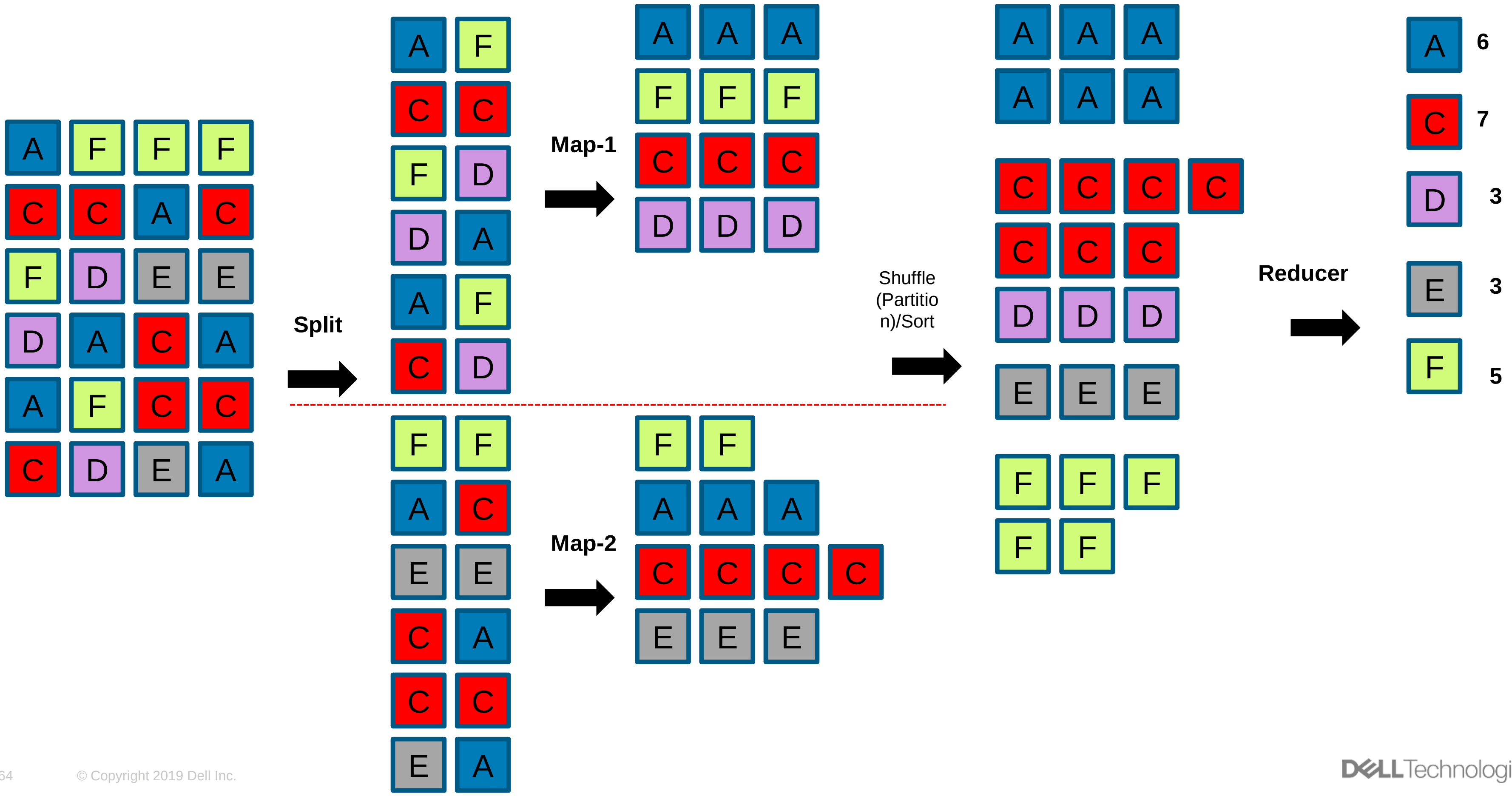
- Java MapReduce Mode
  - Write mapper, combiner, reducer functions in Java using Hadoop Java APIs
  - Read records one at a time
- Streaming mode
  - Uses *nix pipes and standard input and output streams
  - Any language—Python, Ruby, C, Perl, Tcl/Tk, and so on
  - Input can be a line at a time, or a stream at a time

Your Turn

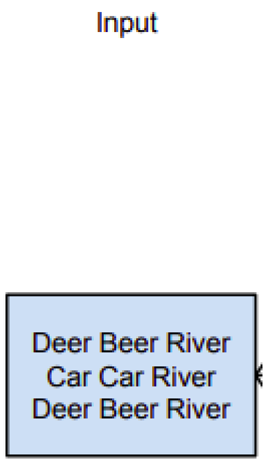
A	B	B	B
C	C	A	C
B	D	E	E
D	A	C	A
A	B	C	C
C	D	E	A

Apply Map Reduce using two mapper vertically and one reduce and output the result

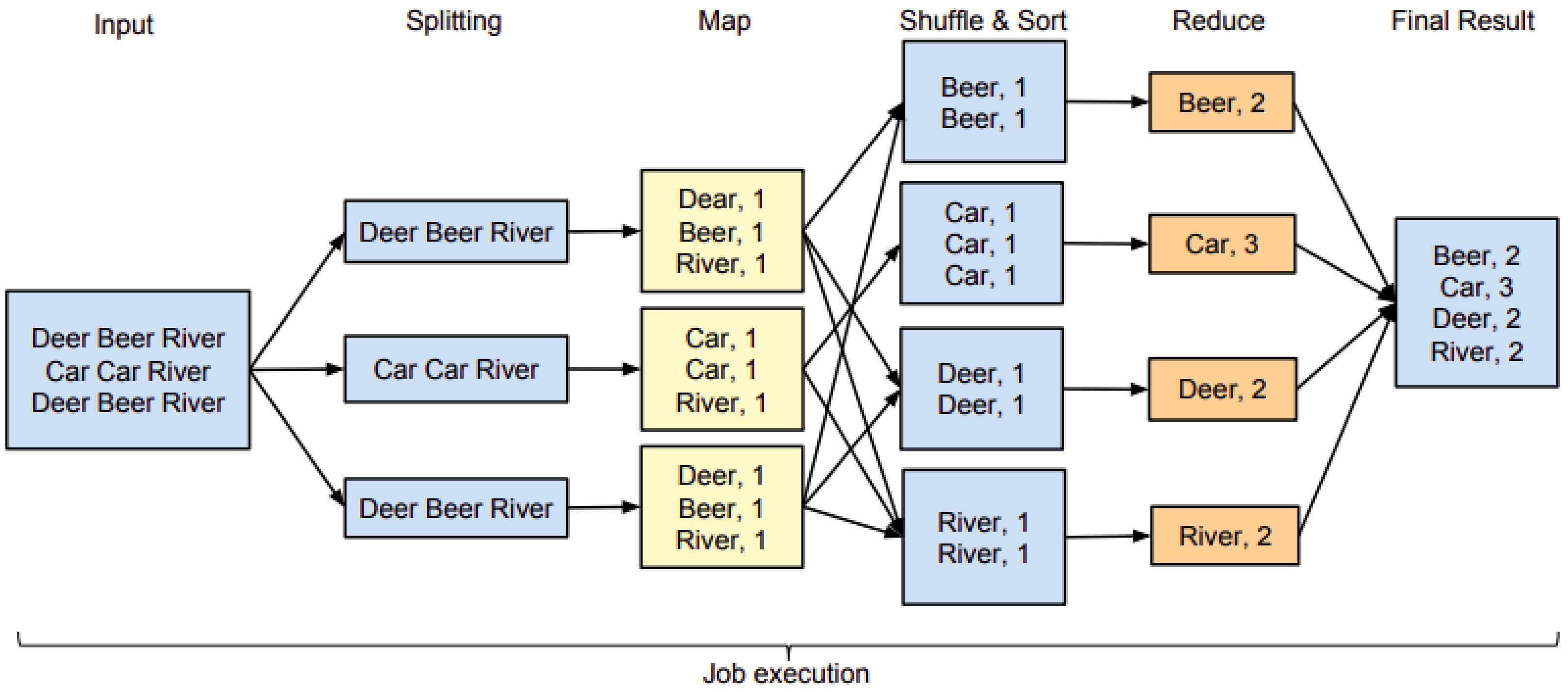
Your Turn


A F F F
C C A C
F D E E
D A C A
A F C C
C D E A
Split
A F
C C
F D
D A
A F
C D
F F
A C
E E
C A
C C
E A
Map-1
A A A
F F F
C C C
D D D
Map-2
F F
A A A
C C C C
E E E
Shuffle
(Partitio
n)/Sort
A A A
A A A
C C C C
C C C
D D D
E E E
F F F
F F
Reducer
A 6
C 7
D 3
E 3
F 5

Your Turn



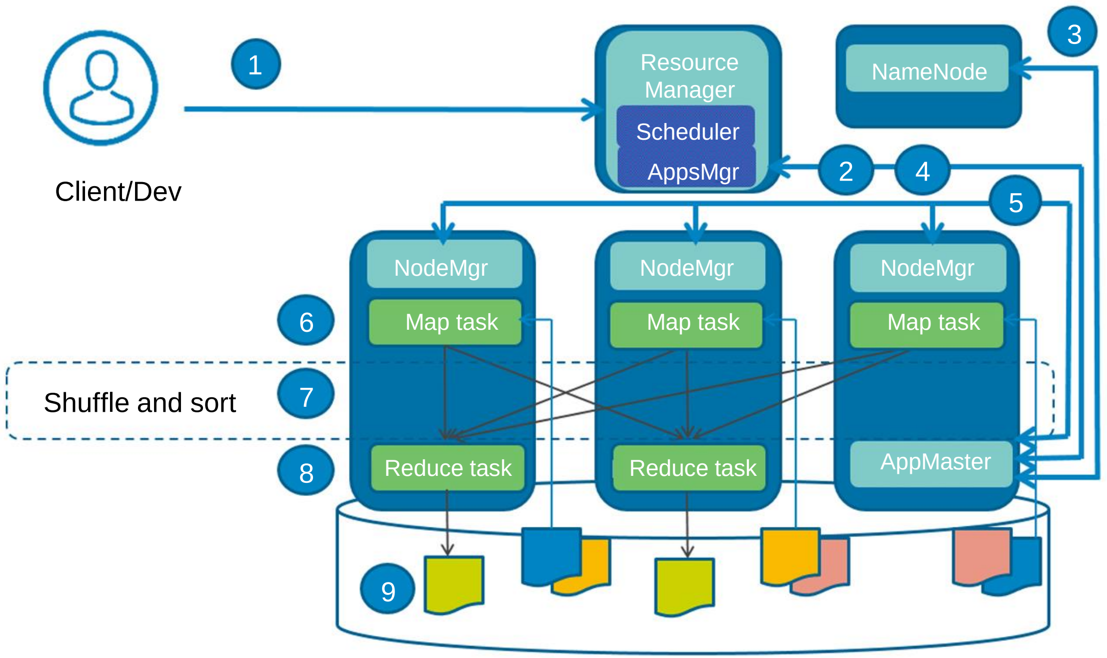
Apply Map Reduce using three mapper and four reduce and output the result


Input
Splitting
Map
Shuffle & Sort
Reduce
Final Result
Deer Beer River
Car Car River
Deer Beer River
Deer Beer River
Car Car River
Deer Beer River
Dear, 1
Beer, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Beer, 1
River, 1
Beer, 1
Beer, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Beer, 2
Car, 3
Deer, 2
River, 2
Beer, 2
Car, 3
Deer, 2
River, 2
Job execution

YARN—Yet Another Resource Negotiator

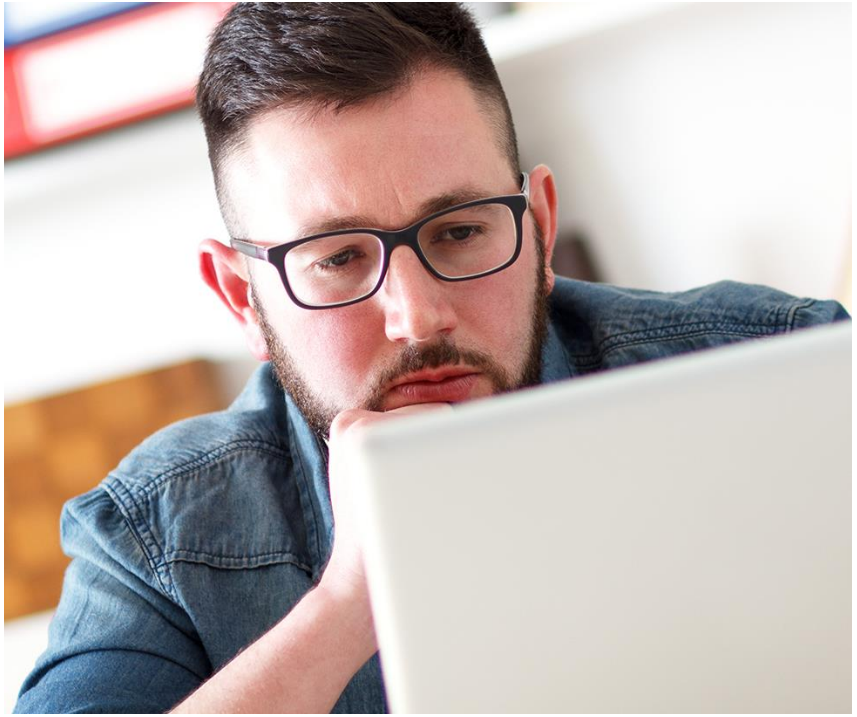
- **Y**et **A**nother **R**esource **N**egotiator (YARN) is a Hadoop ecosystem component that provides the resource management.
- YARN is also one the most important component of the Hadoop ecosystem. YARN is called as the operating system of Hadoop, as it is responsible for managing and monitoring workloads.
- It allows multiple data processing engines such as real-time streaming and batch processing to handle data stored on a single platform.
- Main features of YARN are:
  - Flexibility
  - Efficiency
  - Shared
  - Scalability

YARN—architecture/components



Check your knowledge

1. Why is a combiner function optional in the MapReduce framework?
2. What is the purpose of the NameNode in HDFS?
3. Identify an embarrassingly parallel situation from your current work.
4. What are two benefits of YARN?



Lesson—summary

During this lesson, the following topics were covered:

- Applying the MapReduce/Hadoop Processing Framework in Big Data analytics problems for unstructured data
- Differentiating among the various elements of Hadoop
- Naming the components of HDFS
- Identifying the parts of a Hadoop streaming script