MAS DSE 230
Scalable Analytics

Big Data
Distributed Processing
Big Data Analytics

Mai H. Nguyen

REVIEW: COMPUTER SYSTEMS & PARALLELISM

- Basics of Computer Systems
  - Hardware & Software
  - Computer Instruction Cycle
  - Memory Hierarchy
  - Virtualization
- Parallelism
  - Parallel Processing
  - Task & Data Parallelism
  - Speedup

COMPUTER HARDWARE & SOFTWARE

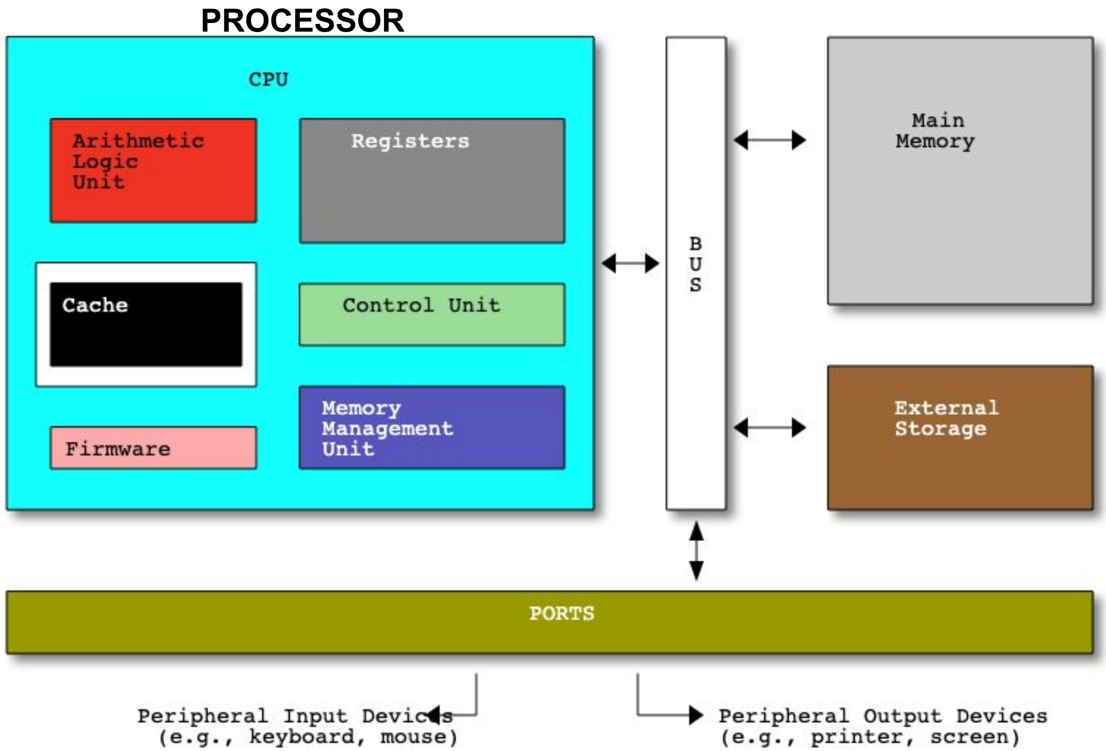
https://www.crampal.in/difference-between-software-and-hardware/

Hardware:
Physical parts of computer

Software:
Programs (instructions) to perform tasks on computer

KEY HARDWARE COMPONENTS

Processor
Executes instructions as specified in program to manipulate data



Main Memory
Stores data and programs for fast access

External Storage
Stores data and programs; slower but more persistent than Main Memory

Network Interface Controller
Sends/Retrieves data over network to/from interconnected computers/devices

https://www.refsmmat.com/courses/751/notes/architecture.html

MAIN TYPES OF COMPUTER SOFTWARE

- Firmware
  - Specially designed for device to help control functionality of device
  - e.g.: TV, remote control, appliances
- System Software
  - Controls and manages operations of computer hardware
  - Operating System: Manages computer's resources to enable application software to execute efficiently
    - e.g.: Linux, MacOS, Windows
- Application Software
  - Implements end user applications
  - e.g.: email, spreadsheet, Web browser, communications

MEMORY HIERARCHY

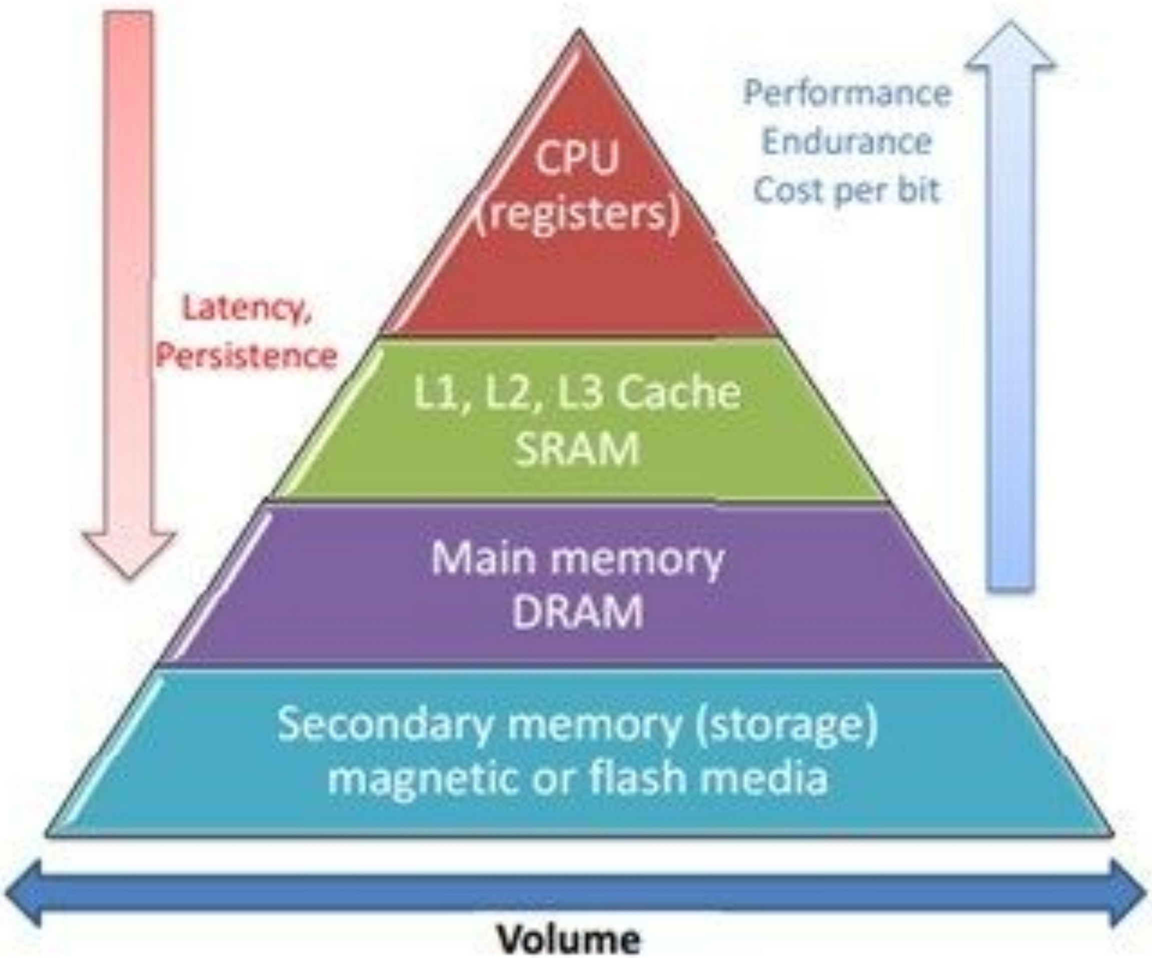
Access Speed

~100GB/s

~10GB/s

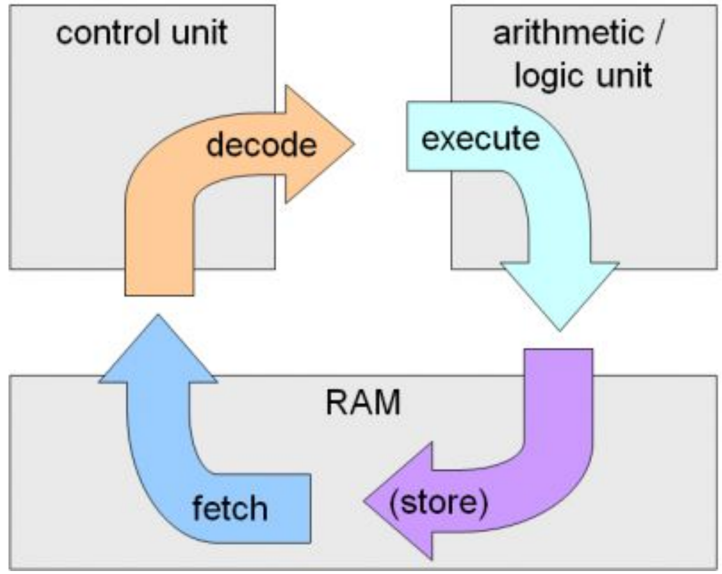
~GB/s

~200MB/s



https://www.researchgate.net/figure/The-memory-hierarchy-pyramid_fig1_319529366

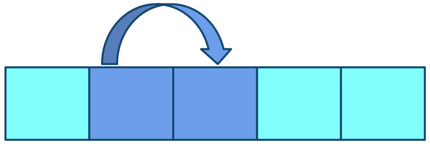
COMPUTER INSTRUCTION CYCLE



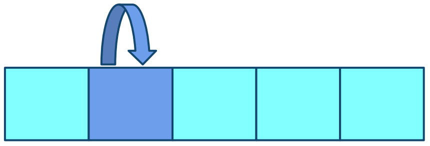
- Modern processors can run millions of instructions per second
- But when data has to be fetched from memory, CU and ALU are idle -> **memory stall**
- Careful use of different levels of memory is essential for overall system performance
  - Want to maximize cache hits to optimize processor utilization

LOCALITY OF REFERENCE

- **Locality of Reference**
  - Many programs tend to access memory locations in a somewhat predictable manner
  - 2 types: spatial and temporal
- **Spatial** locality (locality in space)
  - Items with nearby locations tend to be referenced close together in time

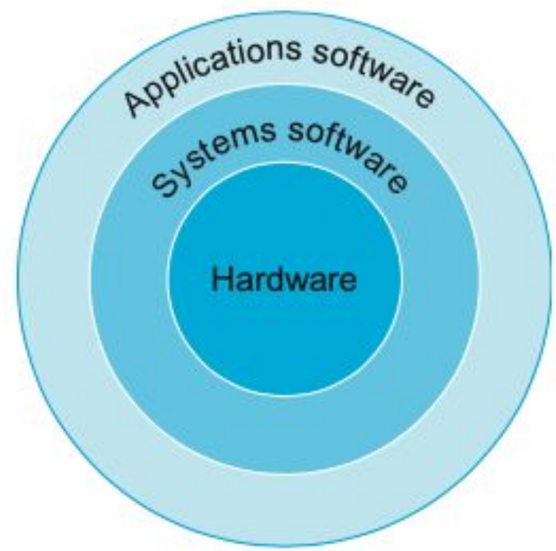


- **Temporal** locality (locality in time)
  - Recently referenced items are likely to be referenced again in the near future



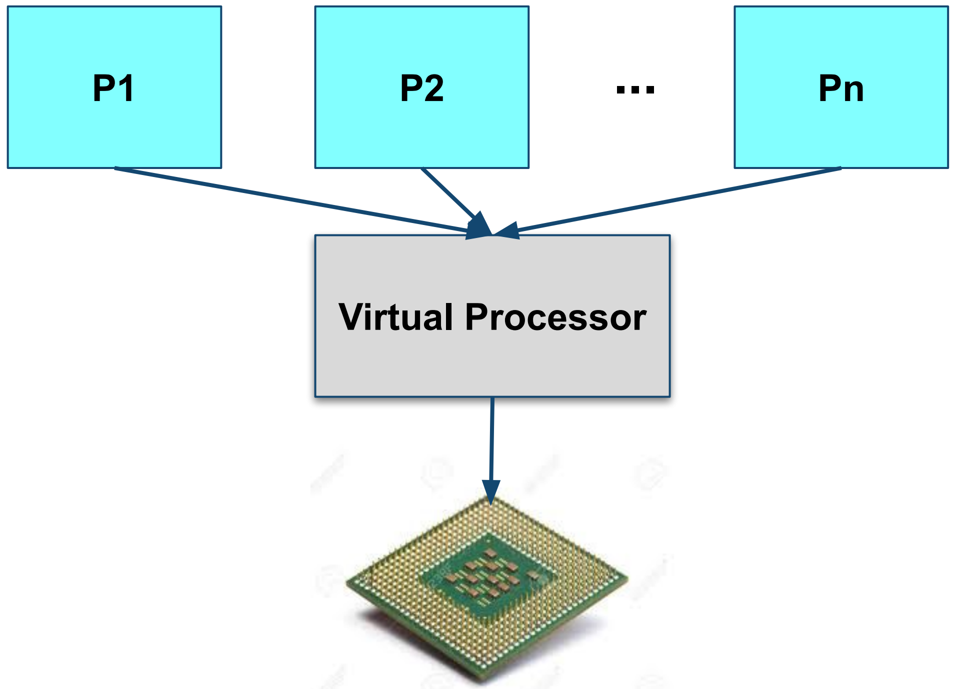
OPERATING SYSTEM

- Operating System (OS)
  - Systems software that manages hardware and software resources of computer system
  - Provides consistent way for application software to use computer hardware effectively, efficiently, and securely
- Functionality provided
  - Process management
  - Main memory management
  - File management
  - Networking
  - Device management
- Common OS
  - MacOS, Windows, Linux



Patterson & Hennessy, *Computer Organization and Design*

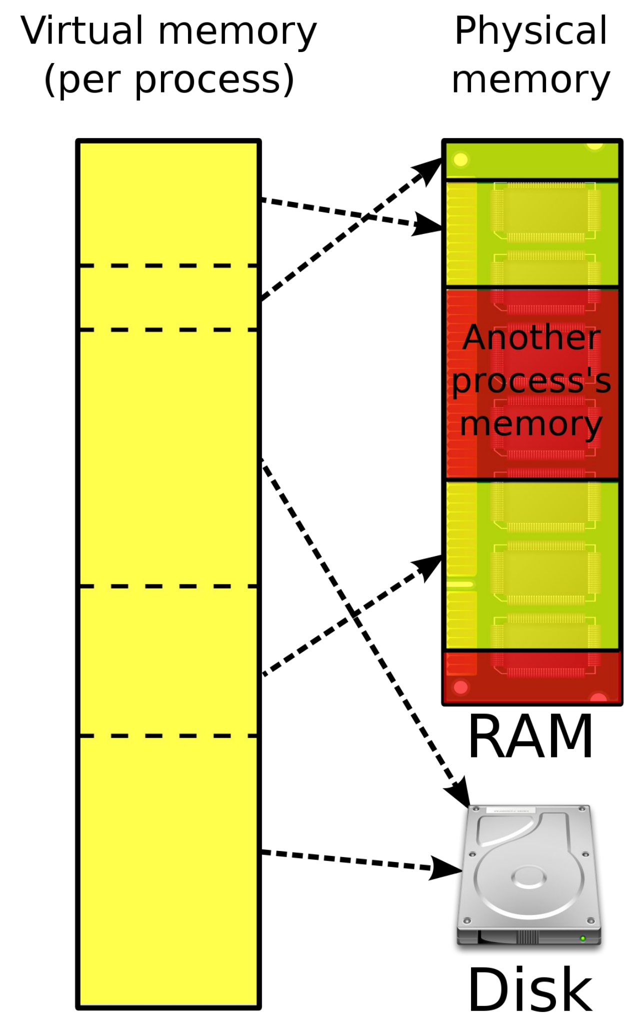
PROCESS VIRTUALIZATION



- OS enables process isolation
  - Each process sees its “own” processor
  - Each process is isolated from other processes
- User can run multiple apps at once on single machine

VIRTUAL MEMORY

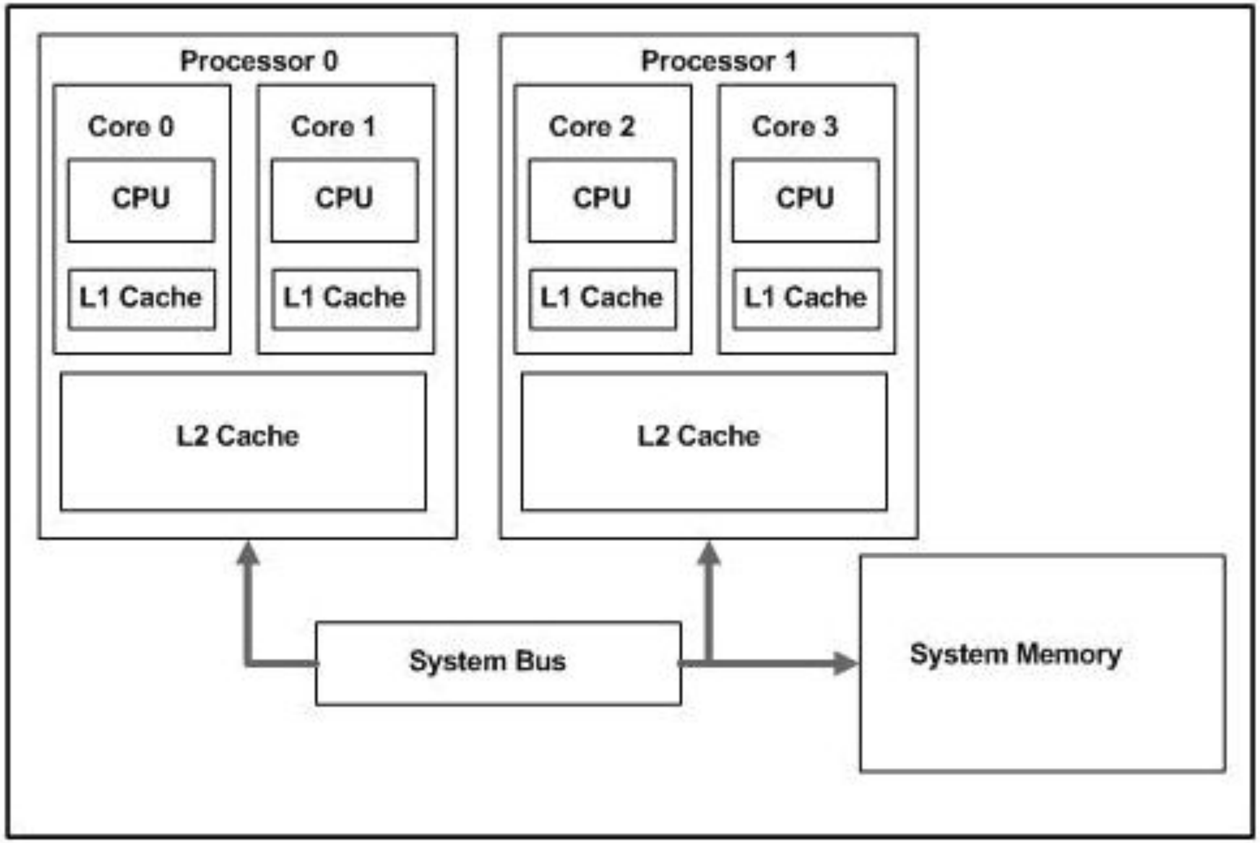
- Memory (also hardware) can also be virtualized by OS
- **Virtual memory**
  - Allows multiple processes to safely share available memory
  - Allows main memory to be extended through secondary storage
- Virtual memory allows multiple processes to *safely* and *efficiently* share available memory



https://en.wikipedia.org/wiki/Virtual_memory

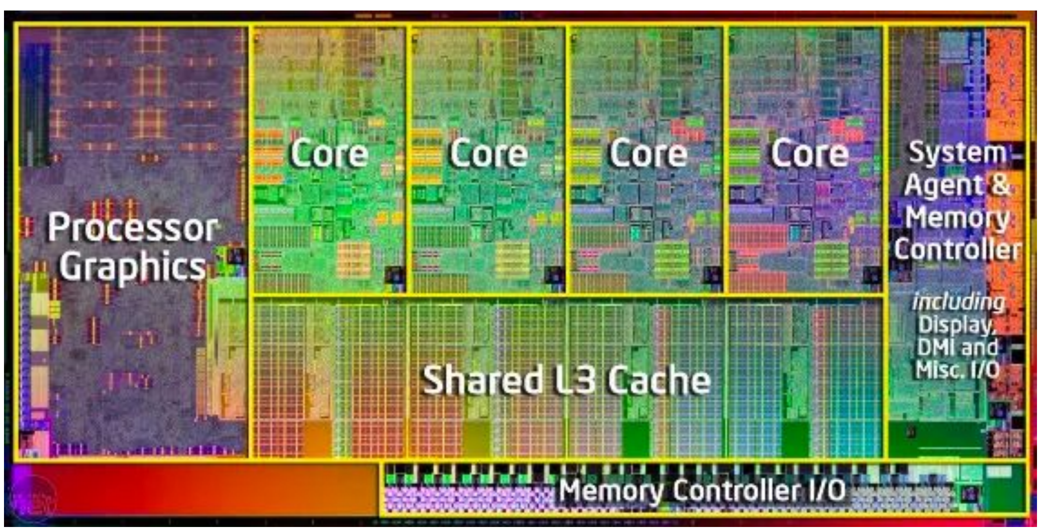
PARALLEL PROCESSING

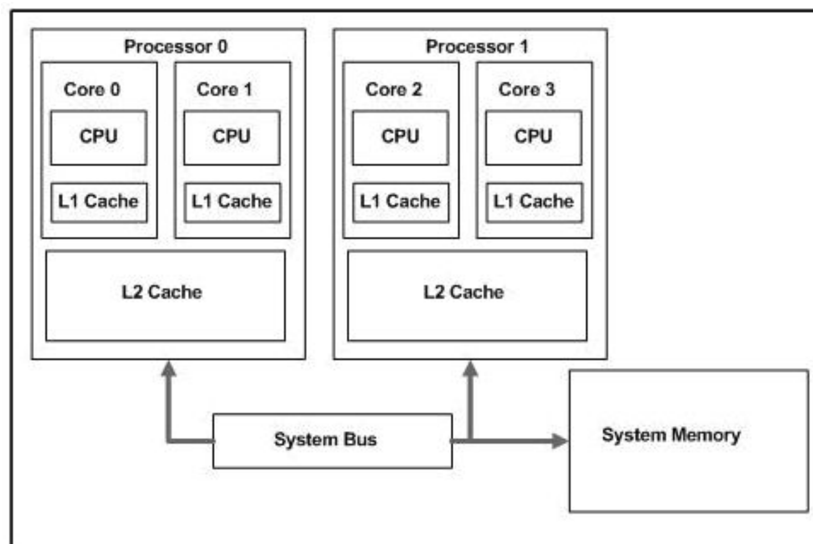
- Split workload across multiple cores / processors / nodes in order to speed up processing



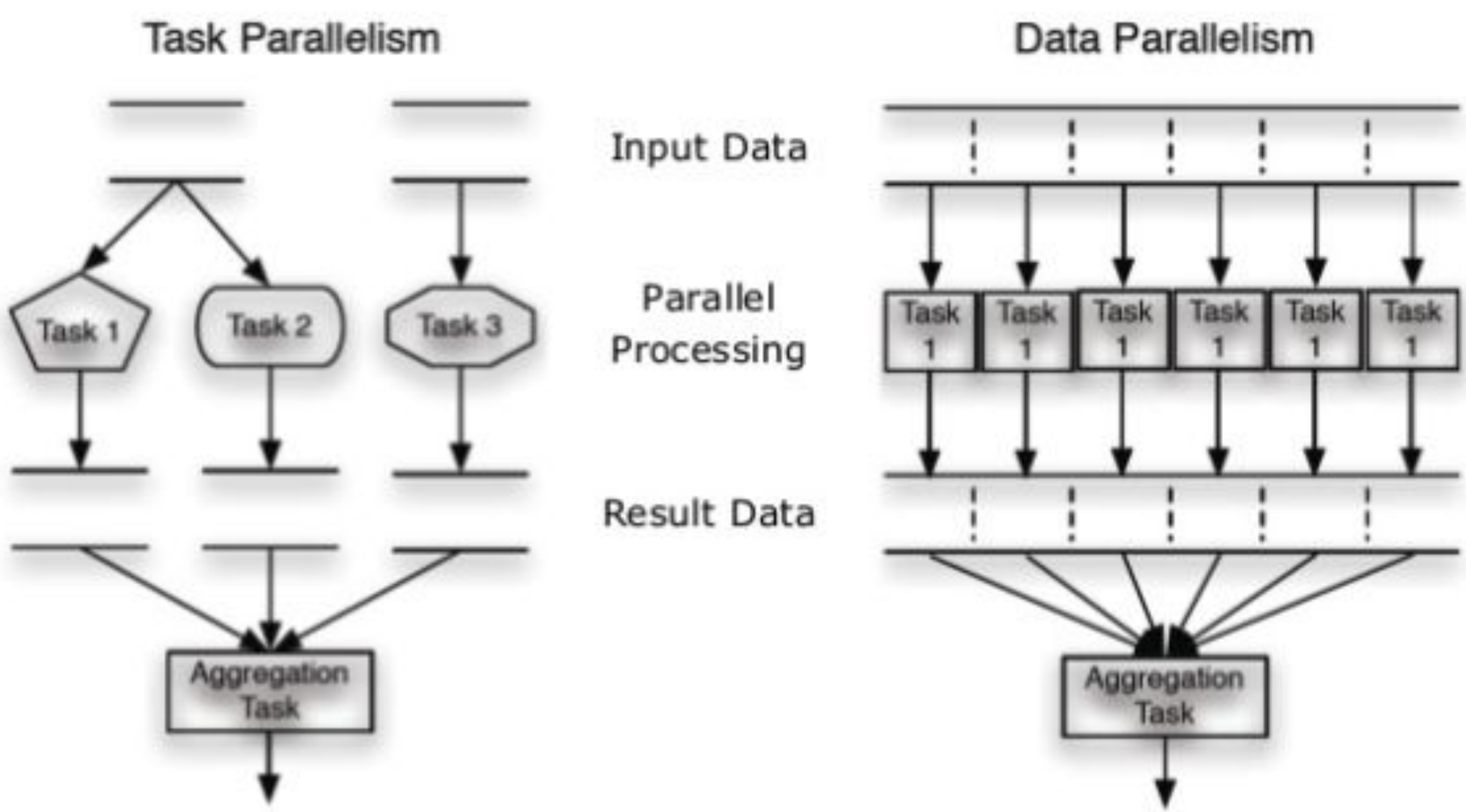
MULTI-PROCESSING

- Modern computers often have multiple *cores* per processor
  - Can also have multiple *processors*
- **Multiprocessing:** Executing multiple processes simultaneously on multiple cores/processors





TASK PARALLELISM VS. DATA PARALLELISM



https://www.slideshare.net/PeterTrger/openhpi-parallel-programming-concepts-week-6

SPEEDUP

- Parallel Computing
  - Processing large-scale data using multiple processors/nodes
- Scaling/Scalability
  - Ability of a computer system to process more data when the amount of resources is increased
- Speedup
  - How much faster a parallel algorithm is compared to a corresponding sequential algorithm

$$\text{Speedup} = \frac{\text{Execution time with 1 core/ processor / worker}}{\text{Execution time with N cores / processors / workers}}$$

AMDAHL'S LAW & GUSTAFSON'S LAW

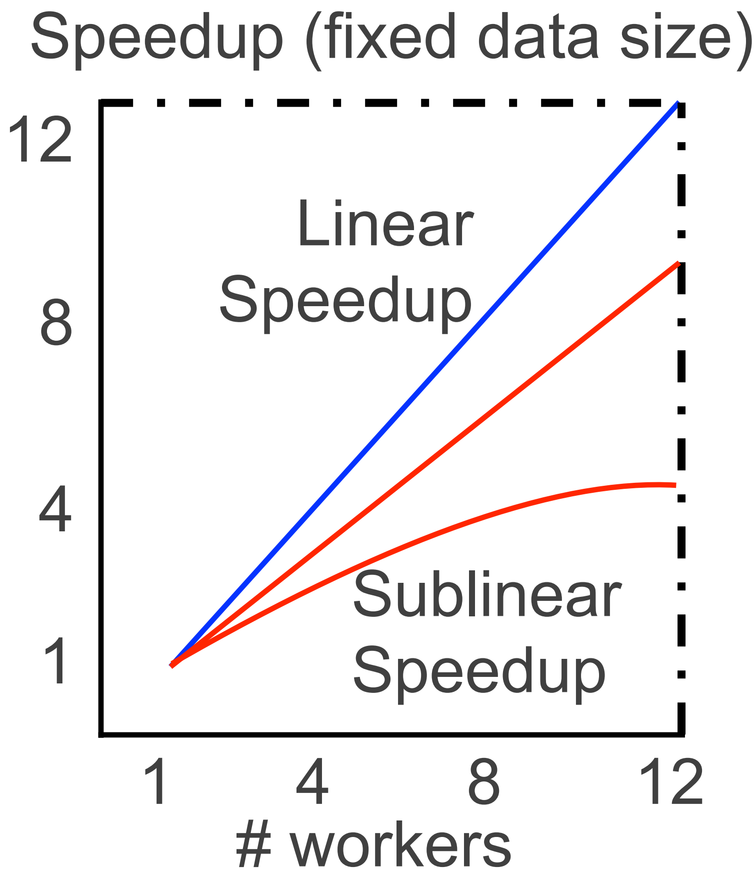
- **Amdahl's Law**
  - Gives upper limit of speedup for problem of *fixed* size
  - In practice, problem size scales with amount of available resources

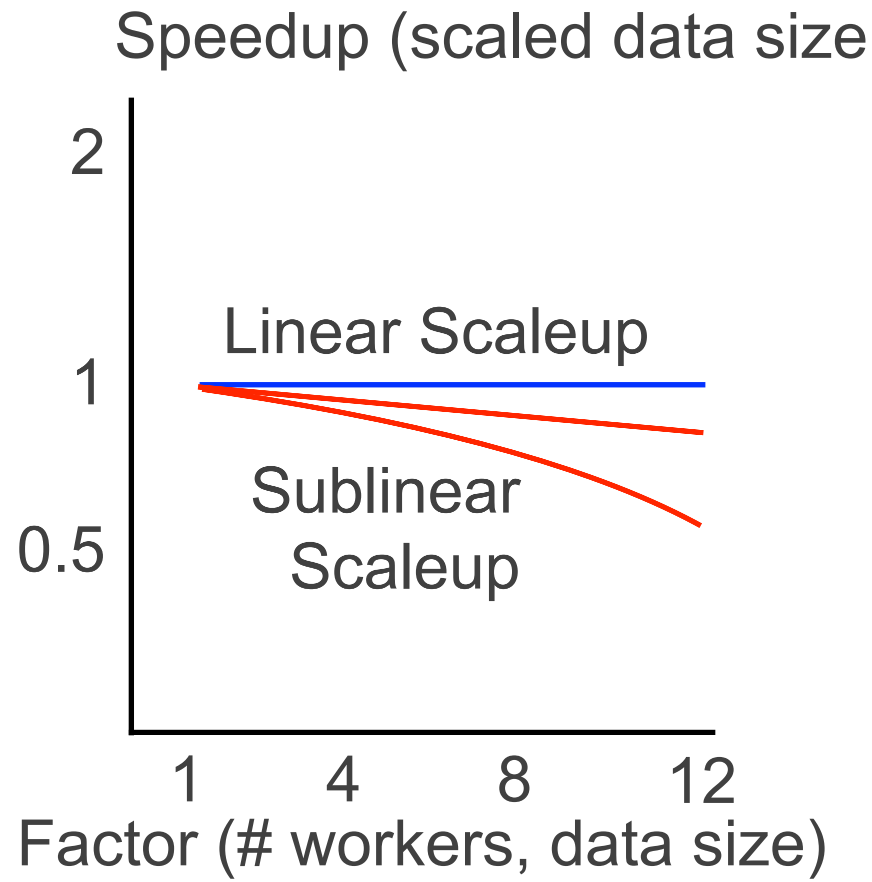
- **Gustafson's Law**
  - Reformulate so that solving larger problem in same amount of time is possible
  - Parallel part scales linearly with amount of resources, and serial part does not increase with respect to problem size

STRONG VS WEAK SCALING

- **Strong Scaling**
  - How execution time varies with number of processors for a fixed *total* problem size
  - Speedup for a *fixed* problem size wrt number of processors
  - How much does parallelism reduce execution time of a fixed problem?
  - Governed by Amdahl's law
- **Weak Scaling**
  - How execution time varies with number of processors for fixed problem size *per processor*
  - Speedup for a *scaled* problem size wrt number of processors
  - How much more data can we process in same amount of time through parallelism?
  - Governed by Gustafson's law

QUANTIFYING PARALLELISM





Speedup plot / **Strong scaling** **Scaleup** plot / **Weak scaling**

Arun Kumar, DSC102

REVIEW: COMPUTER SYSTEMS & PARALLELISM

- Basics of Computer Systems
  - Hardware & Software
  - Computer Instruction Cycle
  - Memory Hierarchy
  - Virtualization
- Parallelism
  - Parallel Processing
  - Task & Data Parallelism
  - Speedup

SESSION 2 TOPICS

- Big Data
- Distributed Processing
- Big Data Analytics

BIG DATA & DISTRIBUTED PROCESSING

- Big Data Overview
- Scalable Systems
- Hadoop
- Spark
- PySpark Exercise
- Assignment

BIG DATA & DISTRIBUTED PROCESSING

- Big Data Overview
- Scalable Systems
- Hadoop
- Spark
- PySpark Exercise
- Assignment

WHAT IS BIG DATA?

http://www.digitalzenway.com/2011/12/data-diet-a-resolution-you-can-stick-to/

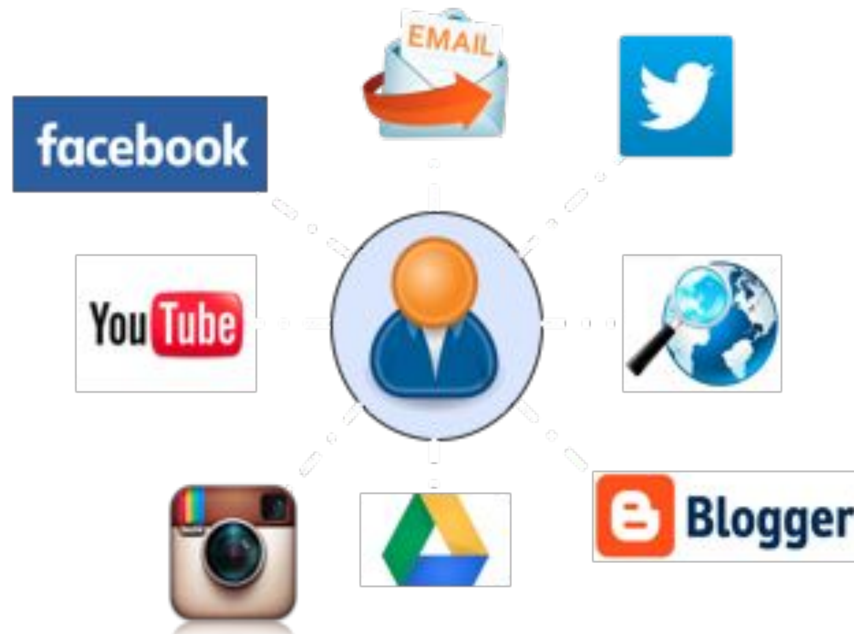
- “Growing torrent” of data
- Data
  - Comes in large volumes
  - Continuous
  - Complex

WHERE DOES BIG DATA COME FROM?

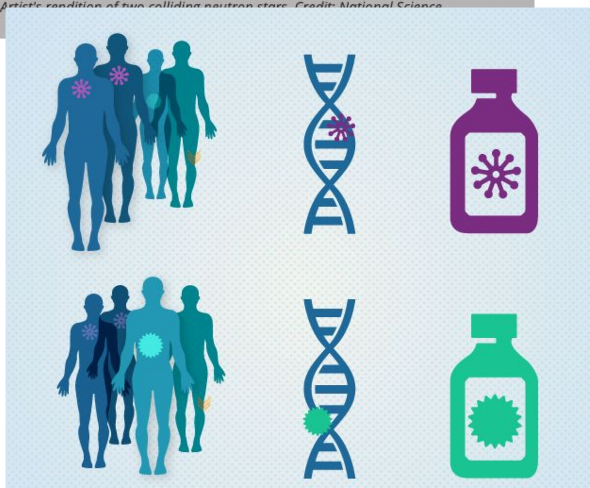
Machines

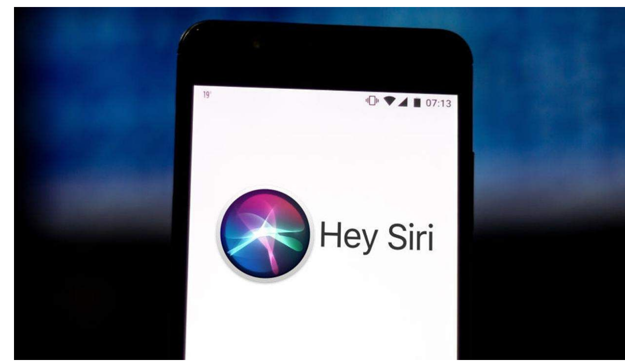
Sensors

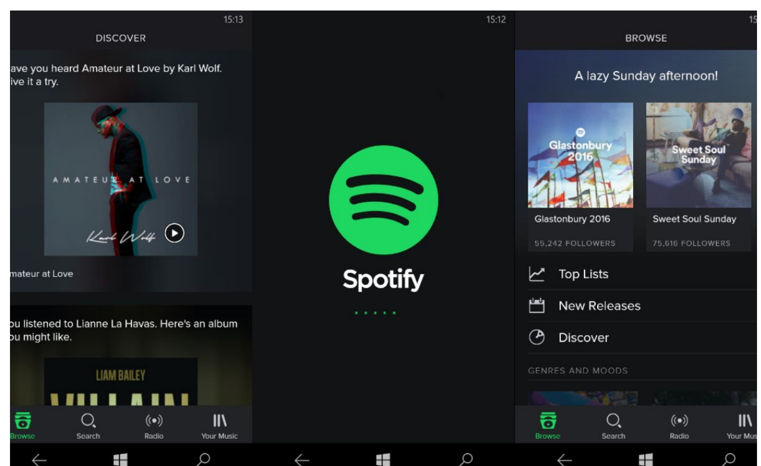
People



HOW IS BIG DATA USED?







WHY BIG DATA NOW?

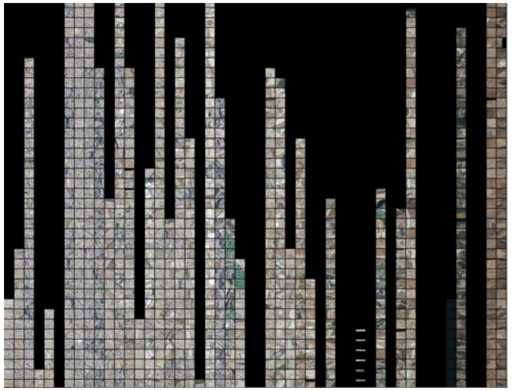
- Advances in processing power, storage capacity, mobile computing, interconnectivity
    - Create unprecedented data
    - Can store and process more data
- Data-driven applications in all areas
    - Science: bioinformatics, image analysis
    - Medicine: drug design, healthcare
    - Retail: targeted advertisement, dynamic pricing
    - Finance: fraud detection, risk analysis
    - Manufacturing: preventive maintenance, supply chain management
    - Law enforcement: crime pattern detection
    - Others ...

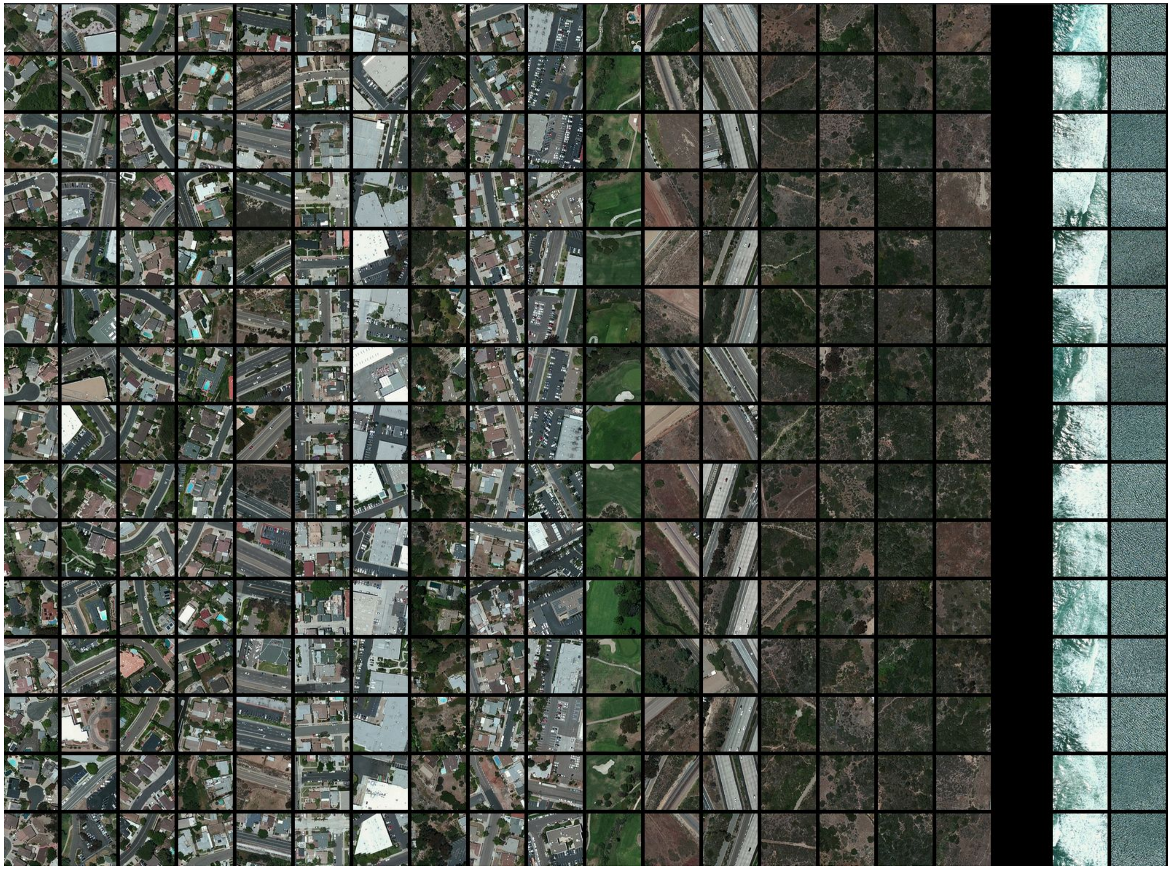
HOW MUCH DATA?

- How much data is big data?

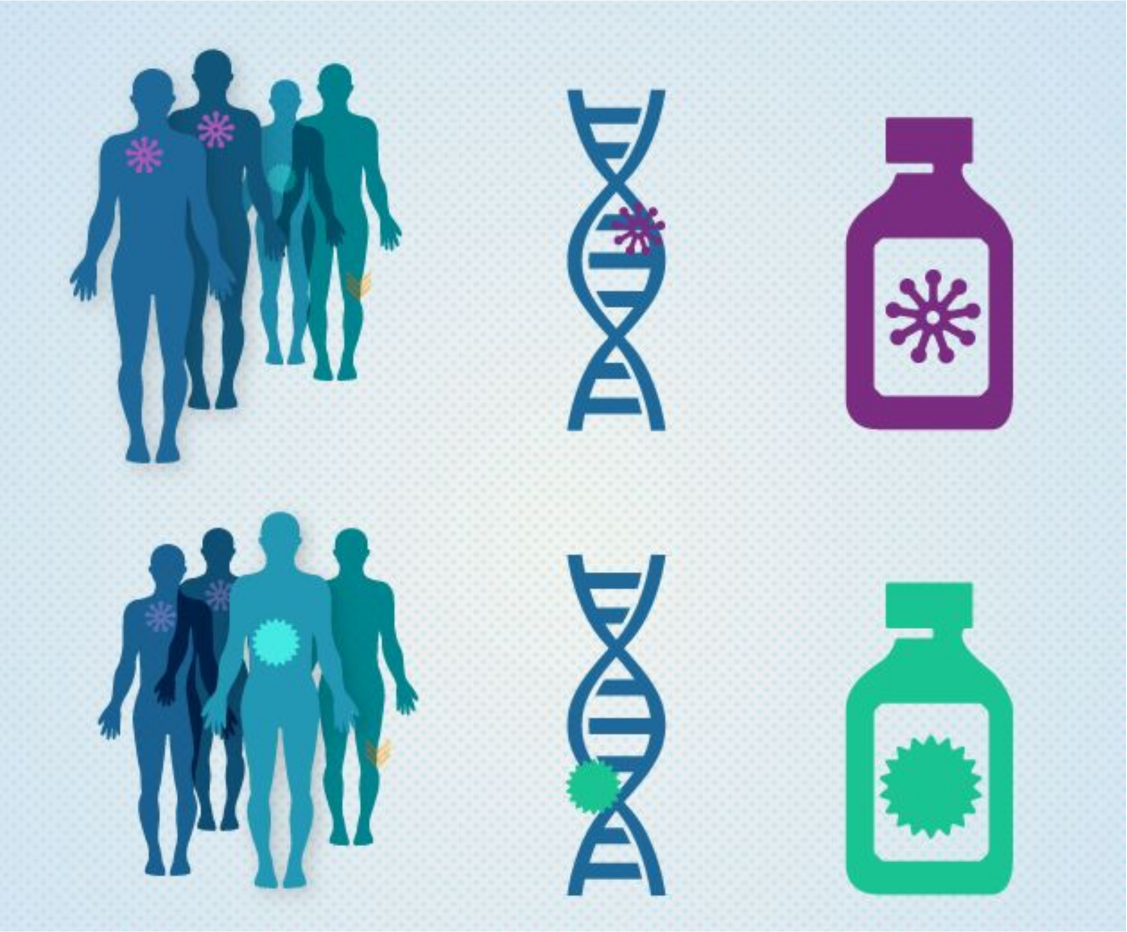
SATELLITE IMAGE ANALYSIS

- MODIS Satellite Instruments
  - Capture images of Earth's surface every 1-2 days
  - 219 TB / year





PRECISION MEDICINE



https://www.cancer.gov/news-events/cancer-currents-blog/2015/precision-medicine-initiative-2016

- Patients with tumors that share the same genetic change receive the drug that targets that change, no matter the type of cancer
- ~3GB genome per human; 900PB+ for nation

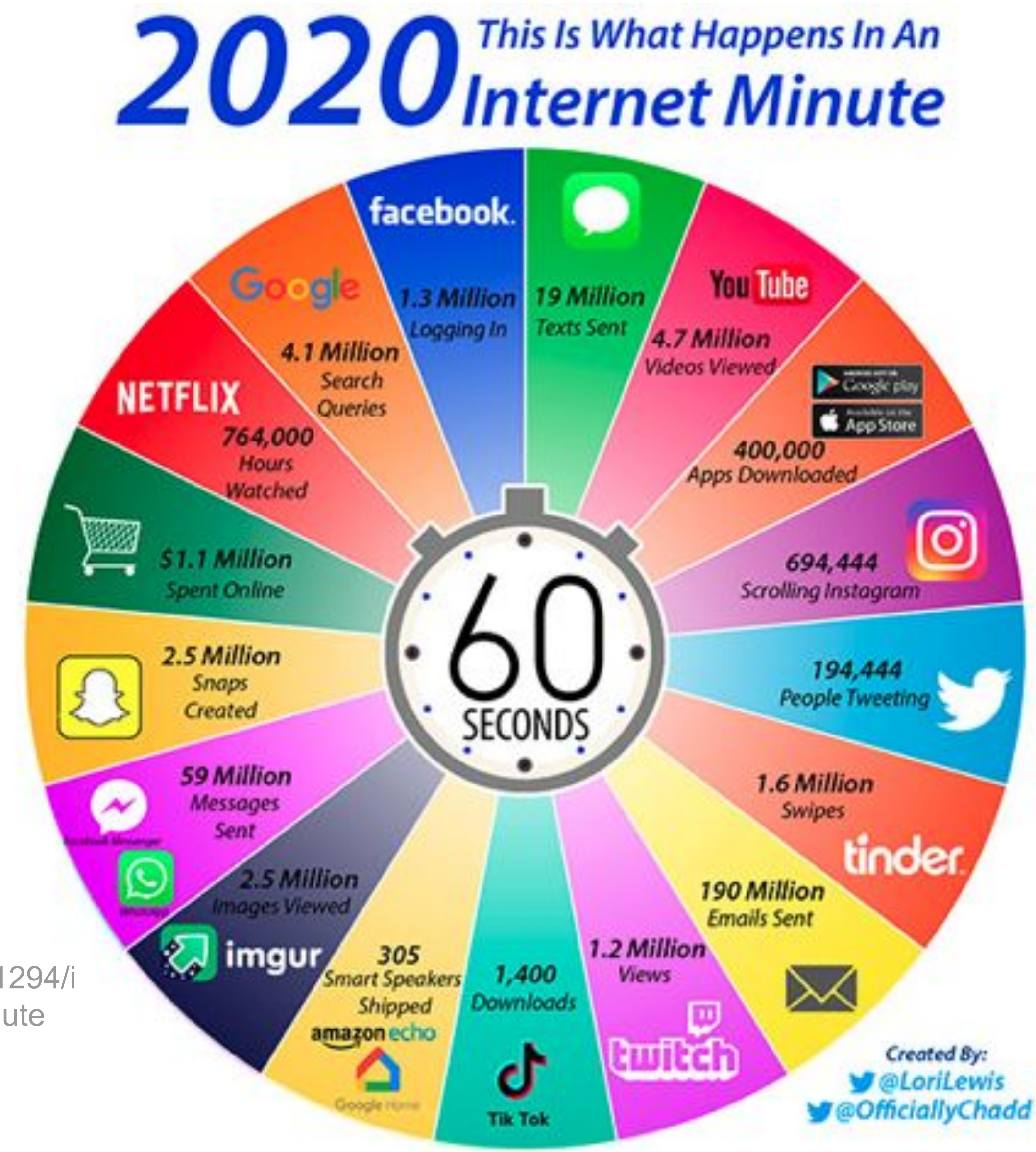
ASTRO-PHYSICS

Artist's rendition of two colliding neutron stars. Credit: National Science Foundation/LIGO/Sonoma State University/A. Simonnet

LIGO: Laser Interferometer Gravitational-Wave Observatory
Generates TBs of data *daily*!

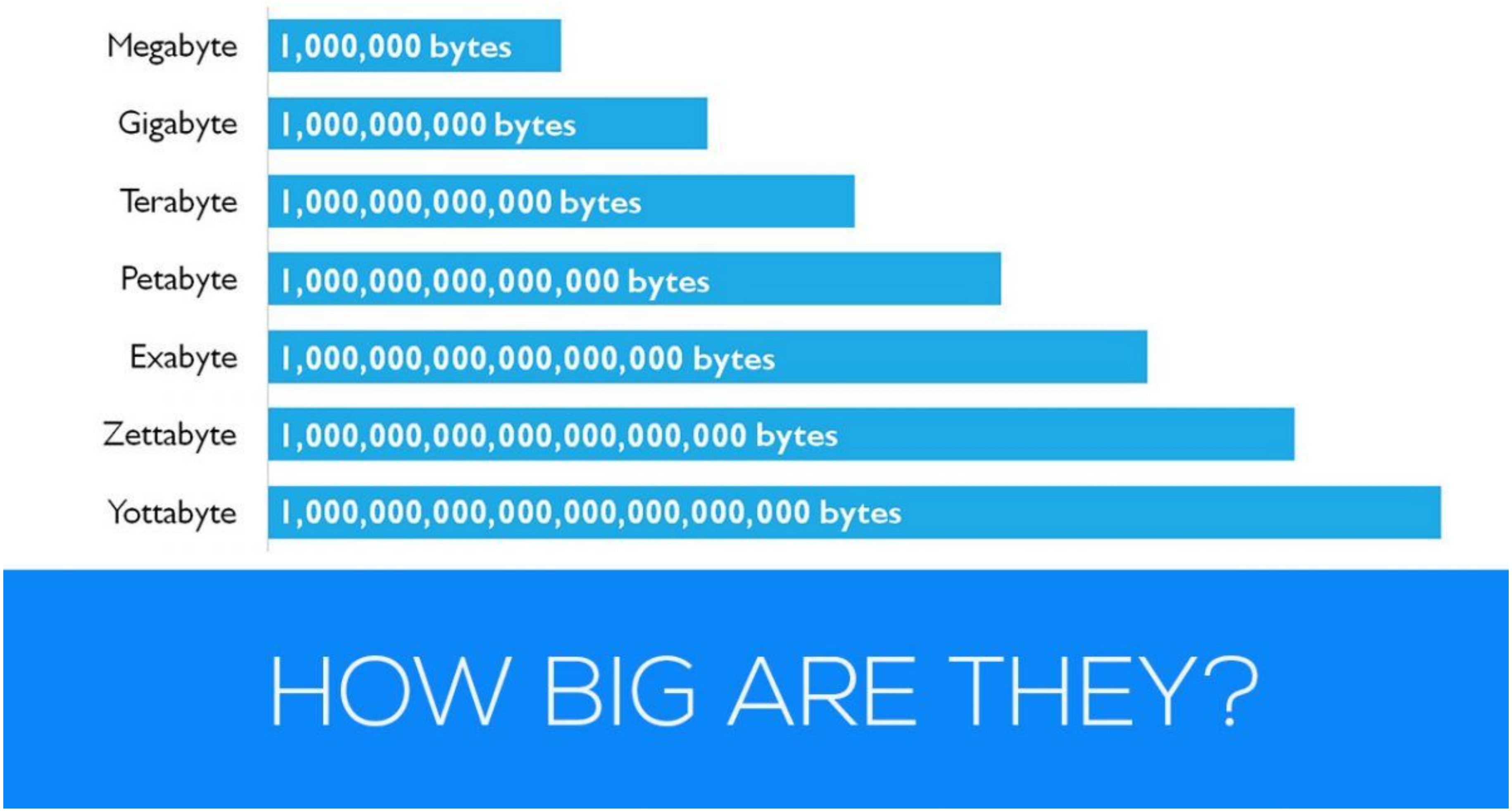
BIG DATA ON THE INTERNET

How much data is generated every minute on the Internet‘



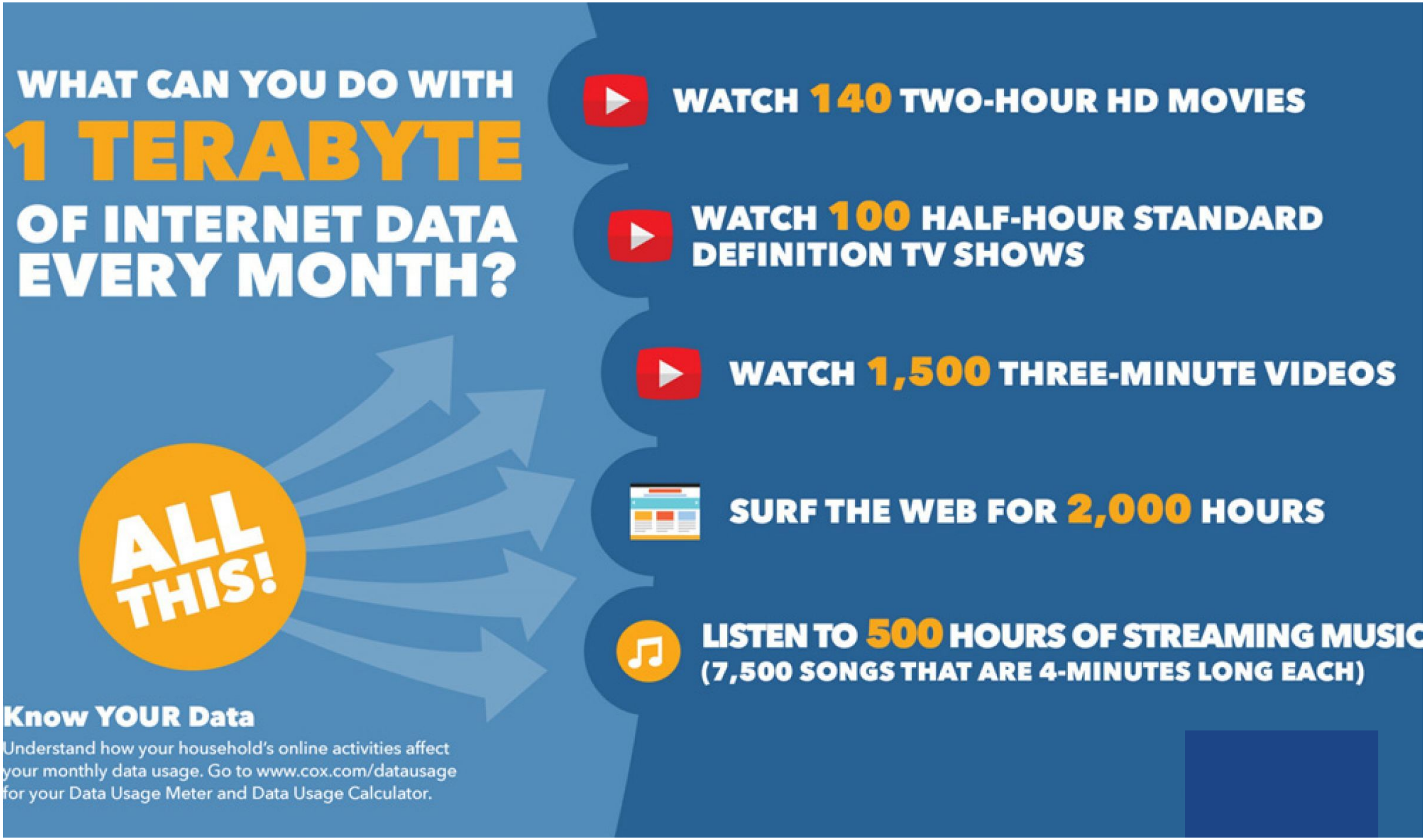
https://www.allaccess.com/merge/archive/31294/infographic-what-happens-in-an-internet-minute

HOW MUCH DATA?



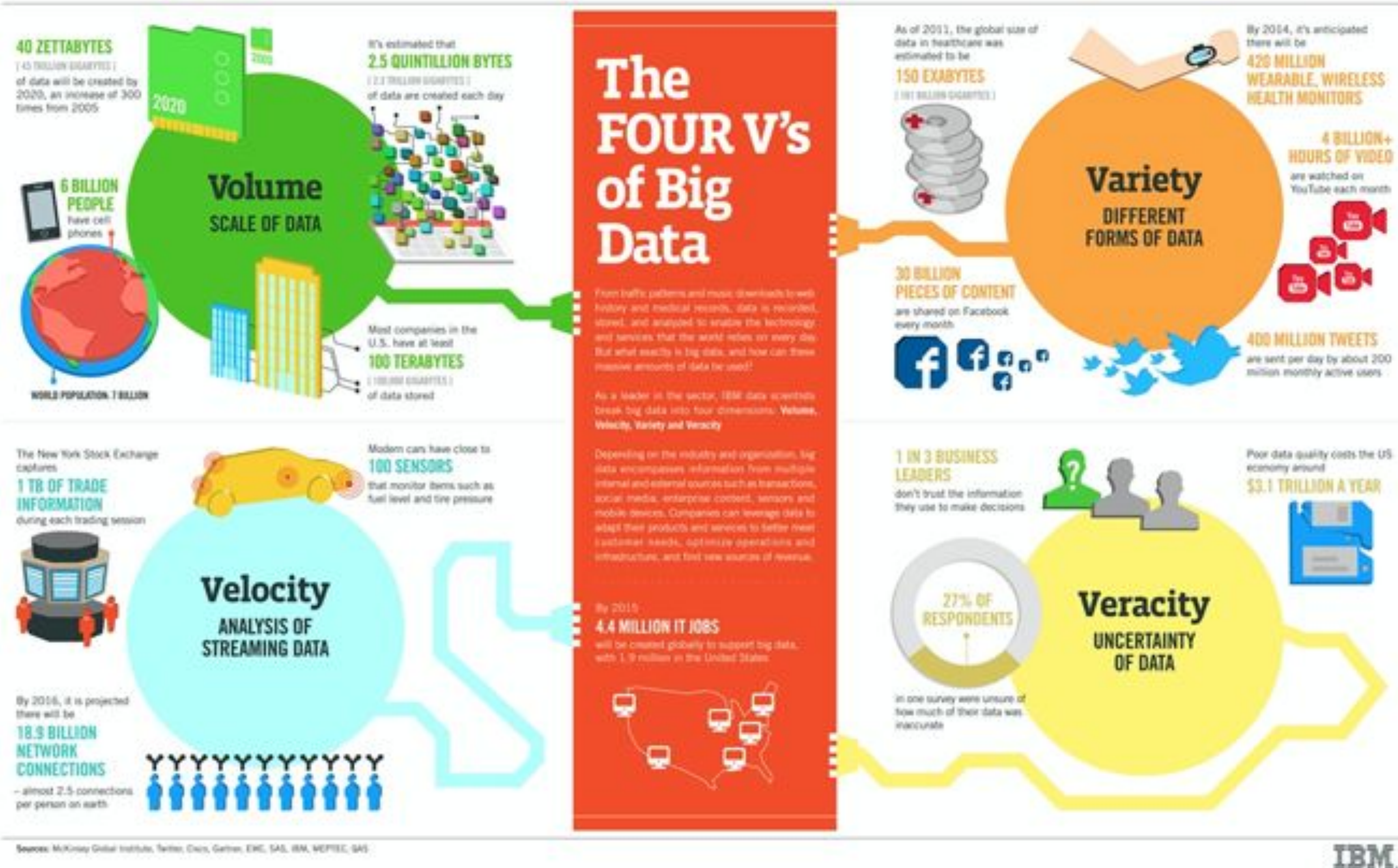
https://www.technotification.com/2017/08/gigabytes-terabytes-petabytes.html

HOW MUCH DATA?



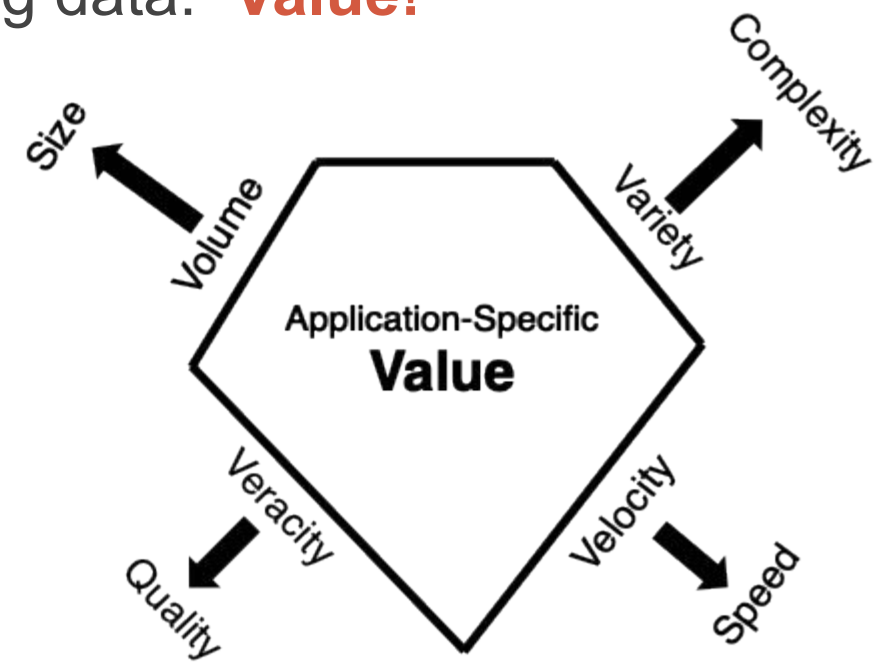
https://www.noozhawk.com/article/what_is_a_terabyte_and_what_can_you_do_with_it_20171117

CHARACTERISTICS OF BIG DATA



CHARACTERISTICS OF BIG DATA

- Goal of processing data is to extract value from data
- Not sufficient to collect data
- Need to analyze data to make sense of it and gain insights
- So 5th 'V' of big data: **Value!**



BENEFITS OF BIG DATA

- Higher sales
- Targeted ads
- Better customer satisfaction
- Customer retention
- Increased efficiency
- Better demand prediction
- Data-driven risk management
- Improved safety
- ...

ANALYZING BIG DATA

- Requires Big Data techniques and tools!

BIG DATA & DISTRIBUTED PROCESSING

- Big Data Overview
- Scalable Systems
- Hadoop
- Spark
- PySpark Exercise
- Assignment

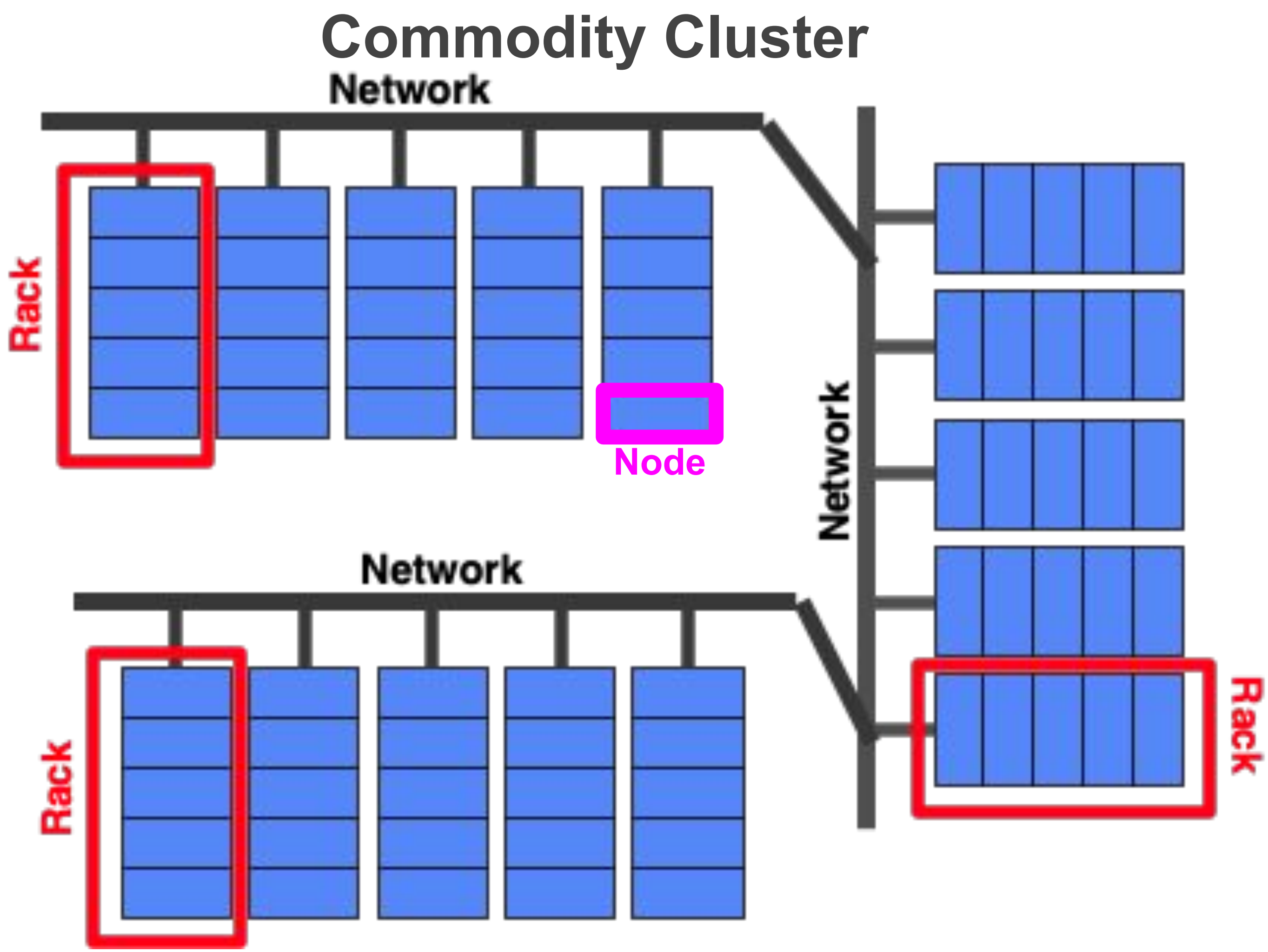
SCALABLE SYSTEMS

- Key components
  - Distributed Computing
    - Processing of large data volumes
    - Scalability
    - Fault tolerance
    - Support for various workloads
  - Distributed File System
    - Data Partitioning
    - Data Replication

DISTRIBUTED COMPUTING

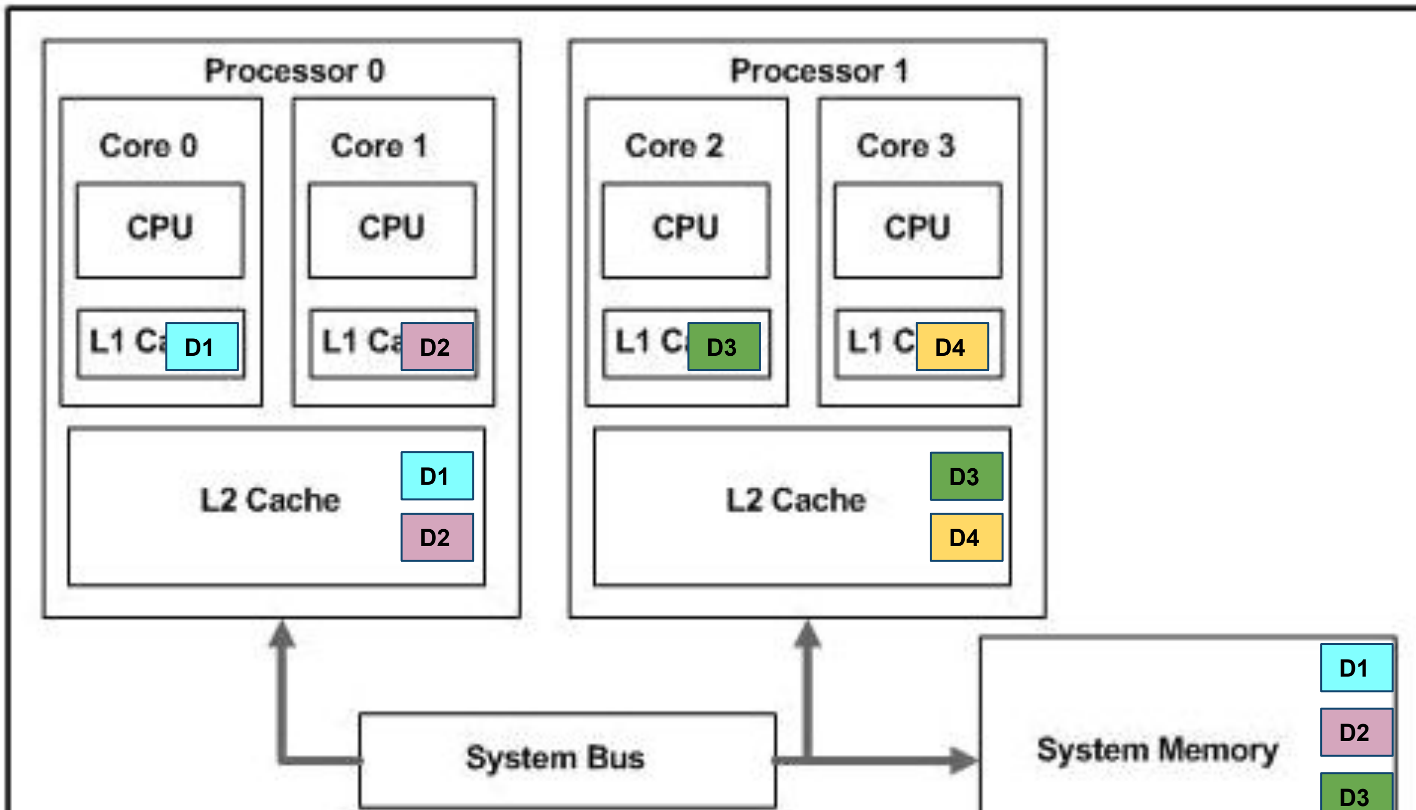
- Distributed Computing
  - Processing is performed on multiple nodes (systems)
- Parallel Computer
  - Large number of single computing nodes with specialized capabilities via a network
    - e.g., SDSC Expanse is supercomputer
  - Specialized => Expensive
- Commodity cluster
  - Large number of low-cost computers with generic computing nodes used in parallel
  - Generic => Cost-effective

DISTRIBUTED COMPUTING



PROCESSING LARGE DATA VOLUMES

- Processing is performed on multiple cores/processors/nodes
- Data parallelism



https://software.intel.com/content/www/us/en/develop/articles/software-techniques-for-shared-cache-multi-core-systems.html?wapkw=smart+cache

SCALABILITY

- **Scalability**
  - Ability of a computer system to accommodate more data when the amount of resources is increased
- **Scaling Up**
  - Adding resources (processors, memory, etc.) to single node
  - Requires specialized hardware (e.g., supercomputer)
  - aka **Vertical Scaling**
- **Scaling Out**
  - Adding more nodes
  - Achievable with cluster of commodity systems
  - aka **Horizontal Scaling**

FAULT TOLERANCE

- Ability of system to recover from failures and continue operating
- Points of failure in distributed system:
  - node, rack, connection, etc.
- When processing large-scale data, restarting is not practical!
- Approaches
  - Data redundancy
    - Periodically save snapshot of data & results (aka checkpoint)
    - Continue processing from last checkpoint
  - Data-parallel job restart
    - Restart process on failed partition

WORKLOADS

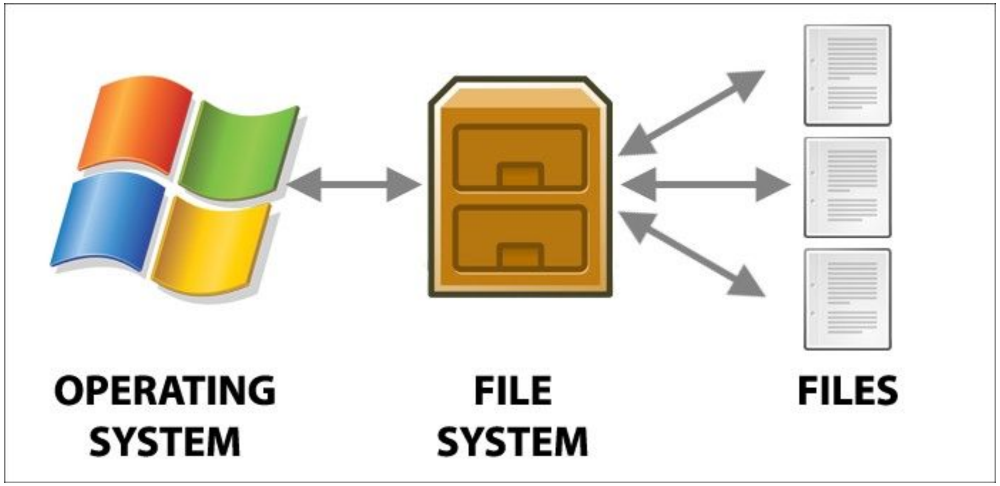
- Scalable systems for processing big data should be extensible to various workloads
- Handle different data types
  - numeric, text, images, audio, geospatial, etc.
- Handle different types of processing
  - batch vs streaming
  - static vs dynamic
  - calculate-once vs. iterative
  - etc.

SCALABLE SYSTEMS

- Key components
  - Distributed Computing
    - Processing of large data volumes
    - Scalability
    - Fault tolerance
    - Support for various workloads
  - Distributed File System
    - Data Partitioning
    - Data Replication

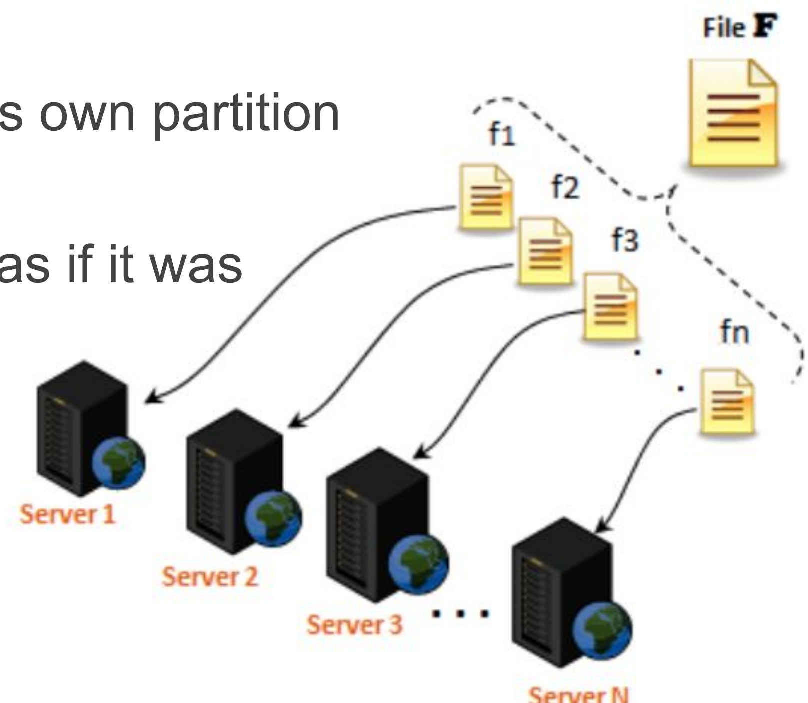
FILE SYSTEM

- Data for/from computing is stored in files on secondary storage
- File system
  - Keeps track of data
  - Organizes data so data can be stored and retrieved efficiently



DISTRIBUTED FILE SYSTEM

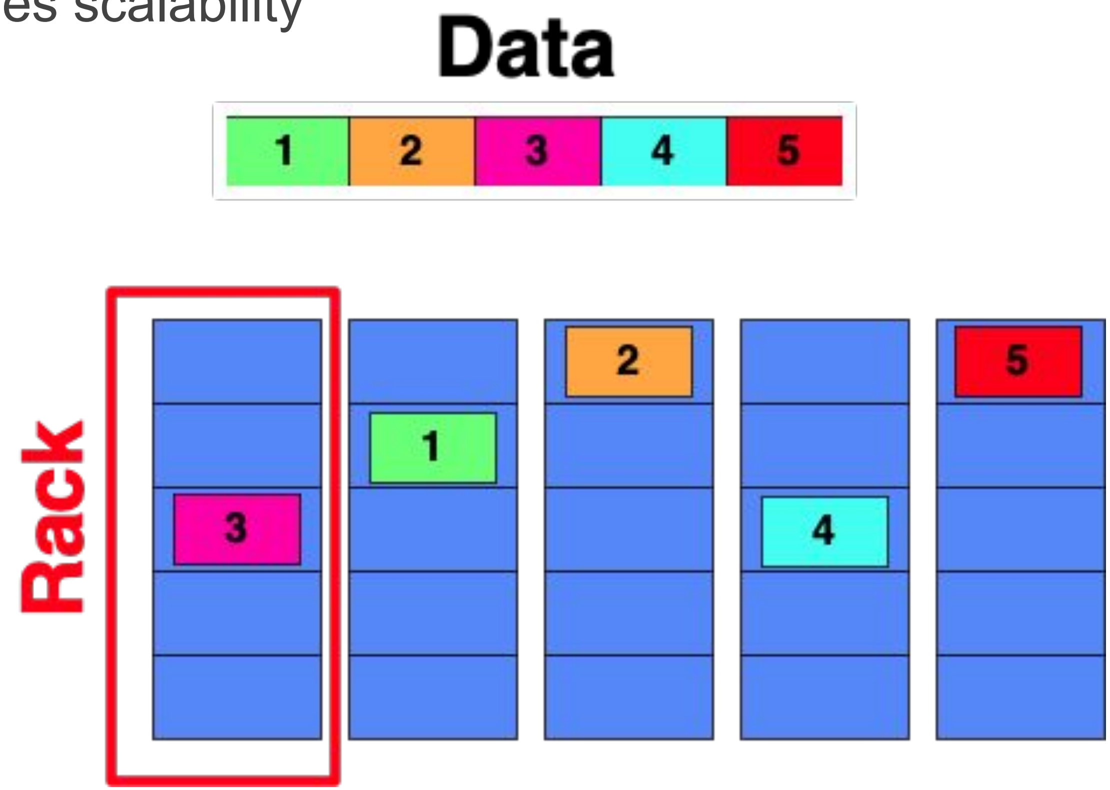
- For efficient processing of very large data file
  - **Partition** data across many computer systems (aka **sharding**)
- **Distributed file system (DFS)**
  - Manages data that is distributed across many networked systems
  - Each local file system manages its own partition
  - Works on top of local file systems
  - Data is accessed and processed as if it was stored on local client machine
  - *Virtualization*: Gives illusion of a single local file
    - Generalization of virtual memory on single system



https://www.linkedin.com/pulse/distributed-file-system-google-replica-sridhar-ramasamy

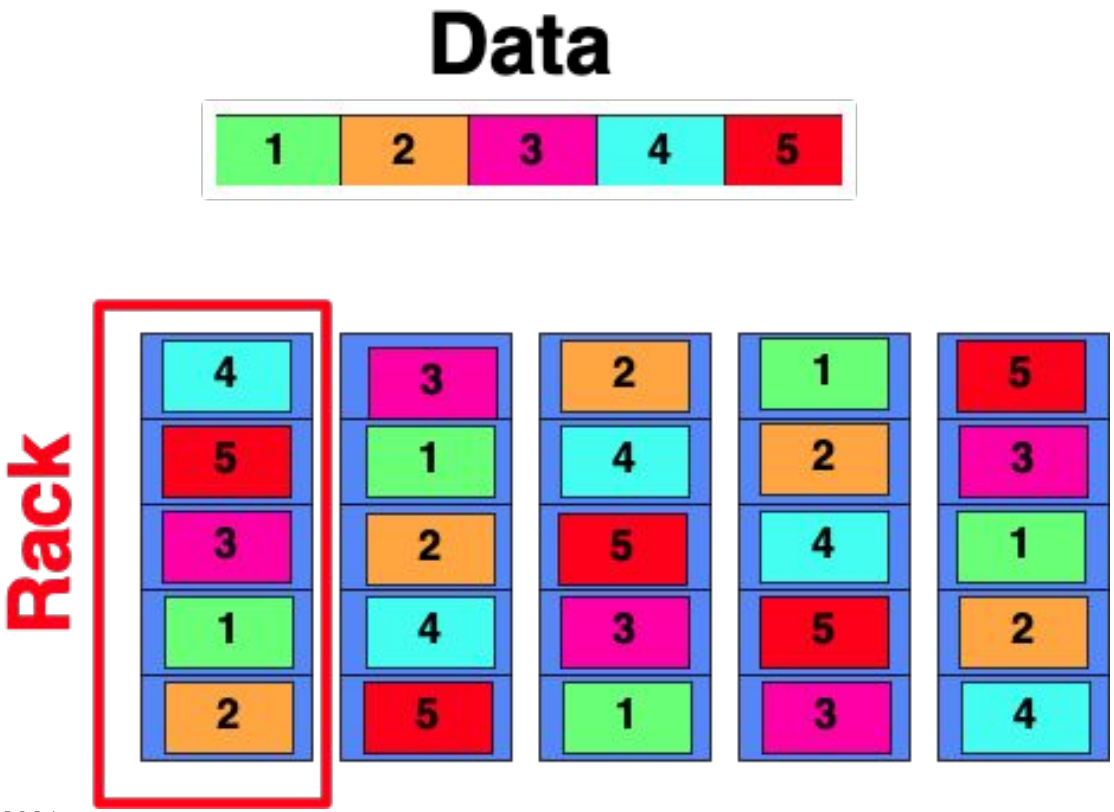
DISTRIBUTED FILE SYSTEM

- **Data Partitioning**
  - Divide large dataset and distribute subsets across nodes
  - Enables handling of large data files via data parallelism
  - Provides scalability



DISTRIBUTED FILE SYSTEM

- **Data Replication**
  - Data partitions are copied, and copies are distributed across nodes
  - Enables fault tolerance and high concurrency



SCALABLE SYSTEMS

- Key components
  - Distributed Computing
    - Processing of large data volumes
    - Scalability
    - Fault tolerance
    - Support for various workloads
  - Distributed File System
    - Data Partitioning
    - Data Replication

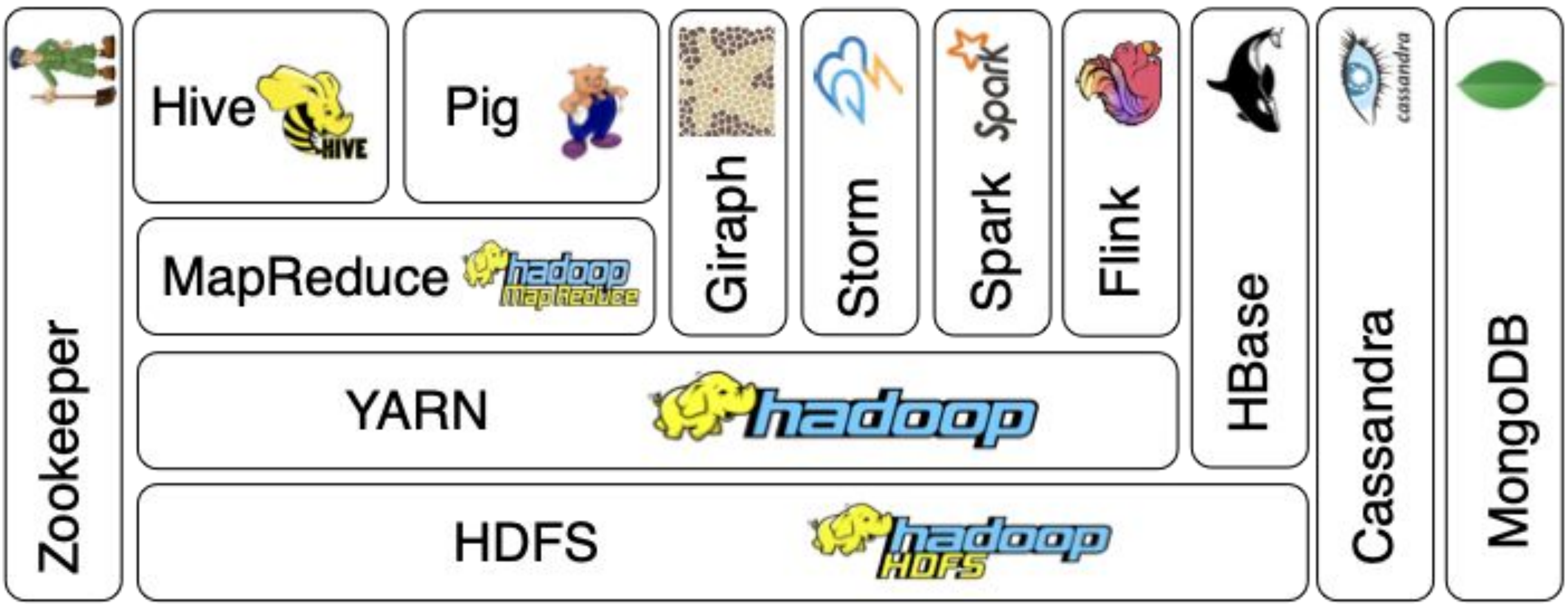
BIG DATA & DISTRIBUTED PROCESSING

- Big Data Overview
- Scalable Systems
- Hadoop
  - History
  - HDFS
  - YARN
  - MapReduce
  - Hadoop Ecosystem
- Spark
- PySpark Exercise
- Assignment

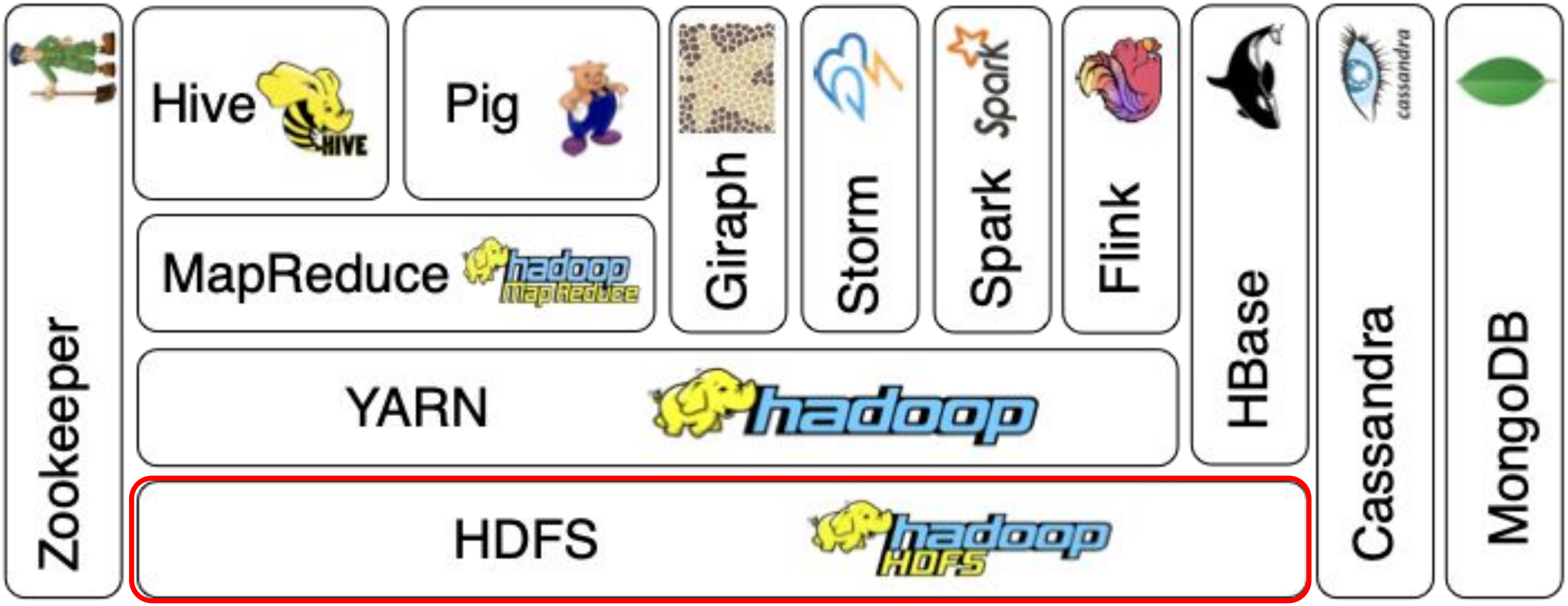
HADOOP

- System for distributed processing of large data sets across clusters of computers
  - Data partitioning, fault tolerance, etc. all handled by the Hadoop library under the covers
  - Scalable platform on commodity clusters
- History
  - Google published Google File System paper in 2003
  - Google published MapReduce paper in 2004
  - Yahoo created Hadoop in 2005

HADOOP ECOSYSTEM



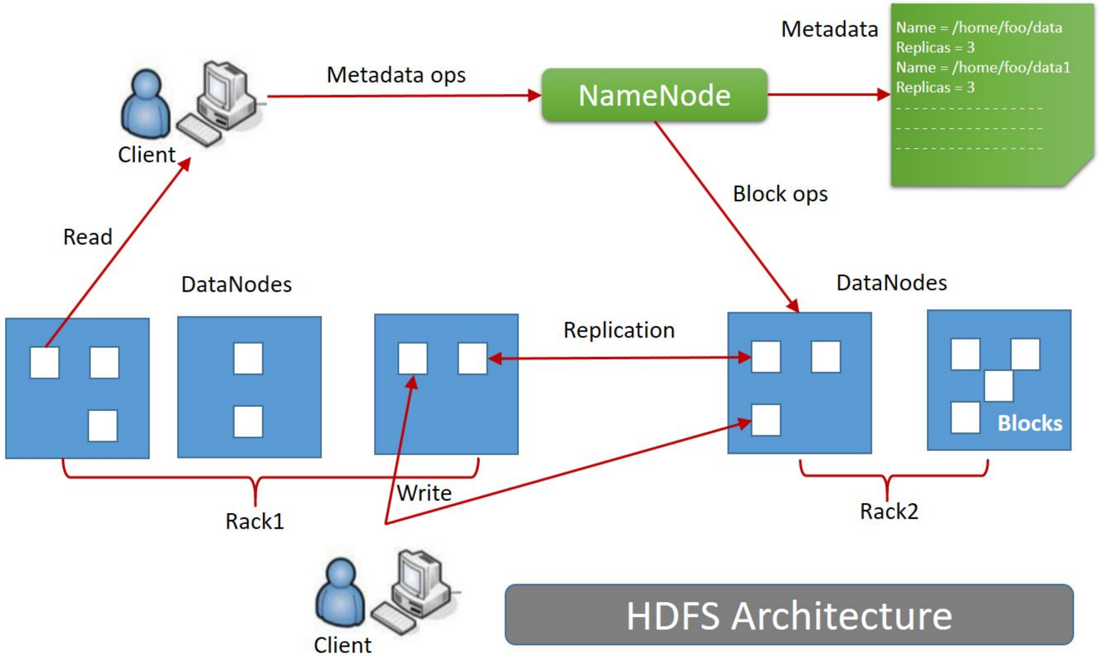
HADOOP DISTRIBUTED FILE SYSTEM (HDFS)



HDFS

- Distributed file system in Hadoop ecosystem
- Open-source spinoff of Google File system (GFS)
- Highly scalable; can do 10s of 1000s of nodes, PB files
- Design features
  - Designed for clusters of commodity nodes
  - Provides *scalable* storage for many scalable systems
  - *Parallel* reads/writes of partitioned data “blocks”
  - Replication of blocks improves *fault tolerance*

HDFS

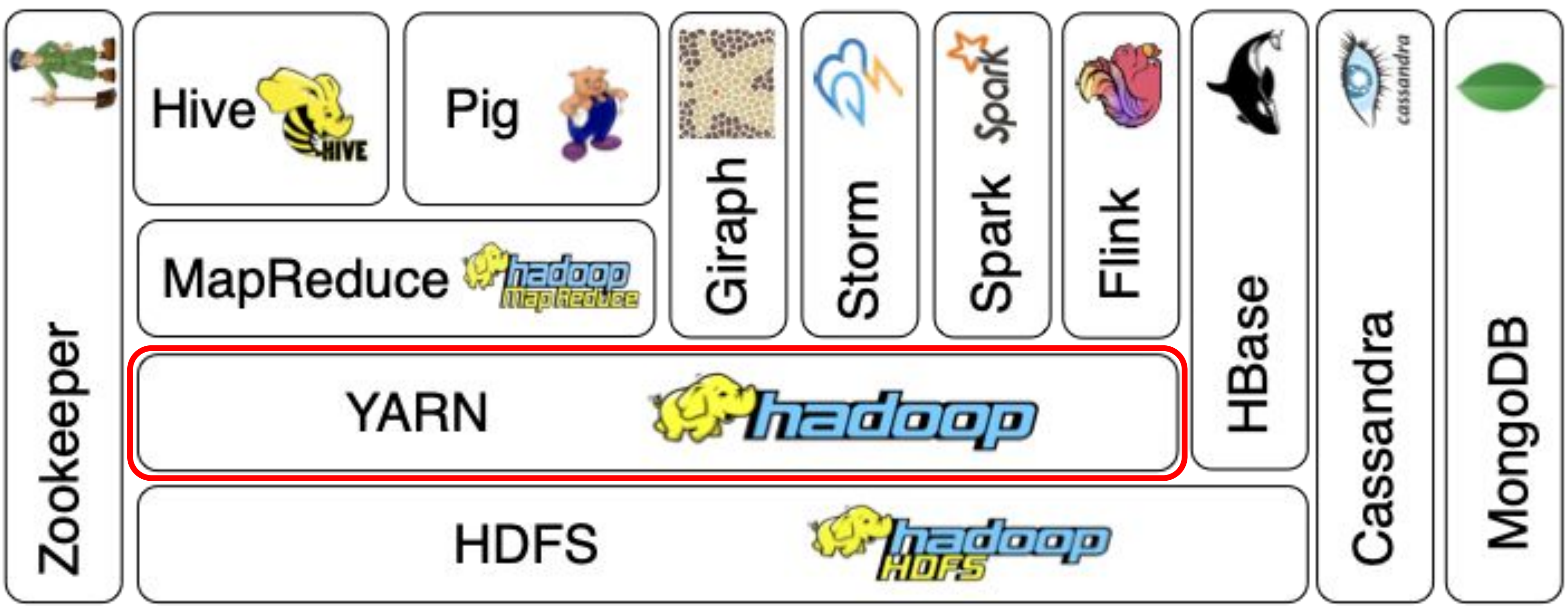


https://www.oreilly.com/library/view/distributed-computing-in/9781787126992/3275691a-477f-4e3a-a00c-9a64bda93b16.xhtml

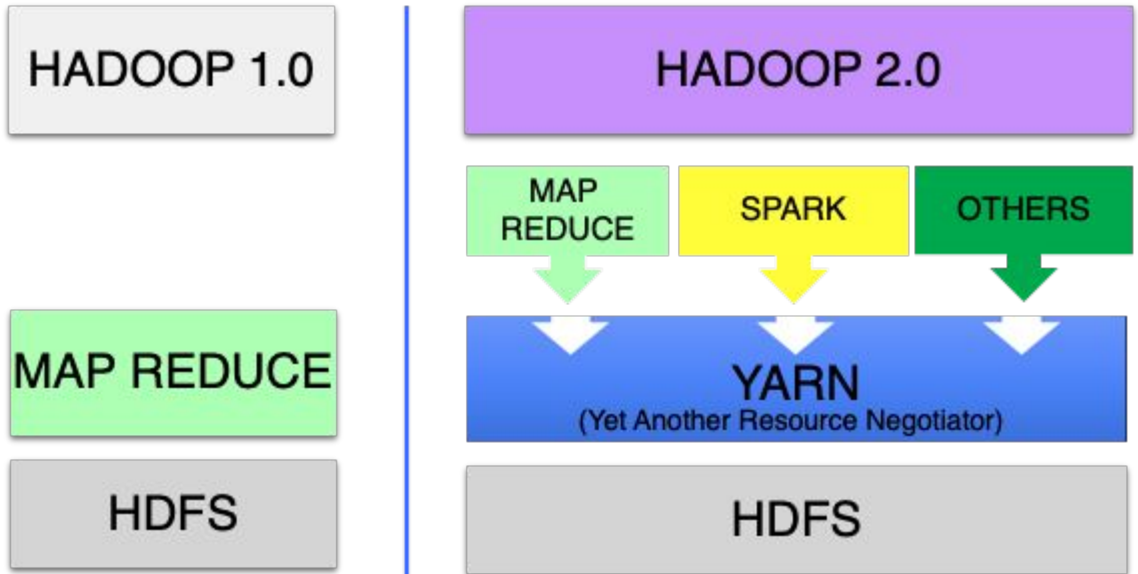
- **NameNode**: One per cluster
  - Coordinates operations of HDFS
  - Manages metadata related to datafile
  - Maps data blocks to DataNodes and issues commands to DataNodes
- **DataNode**: One per node
  - Provides storage for data blocks, which are replicated on multiple nodes
  - Gets commands from NameNode to create, store, delete, replicate data blocks

YARN

- Yet Another Resource Negotiator (YARN)
- Provides job scheduling and cluster resource management
- Enables different types of applications to run in Hadoop



YARN

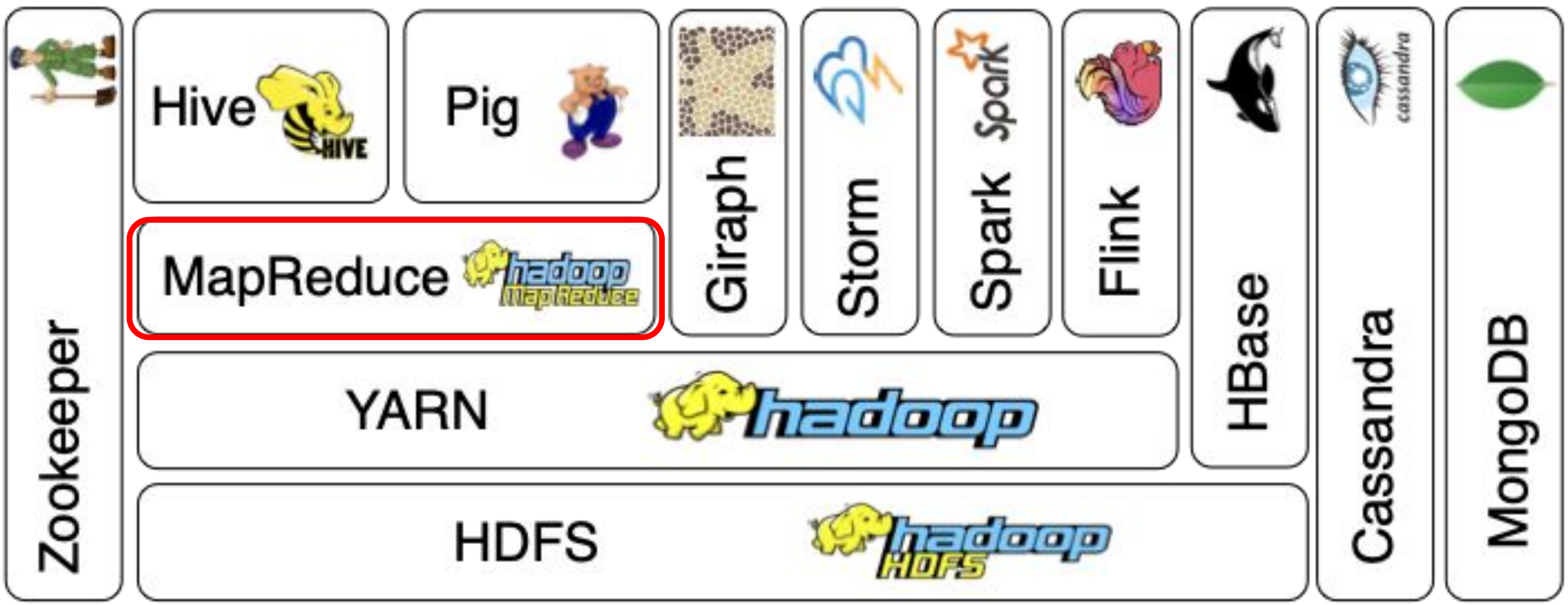


- Hadoop 1.0
  - No resource manager!
  - All applications had to use MapReduce

- Hadoop 2.0
  - Resource management decoupled from data processing and job scheduling & monitoring
  - Allows non-MapReduce applications to run in Hadoop
  - Provides standard platform for variety of applications
  - Much higher overall efficiency

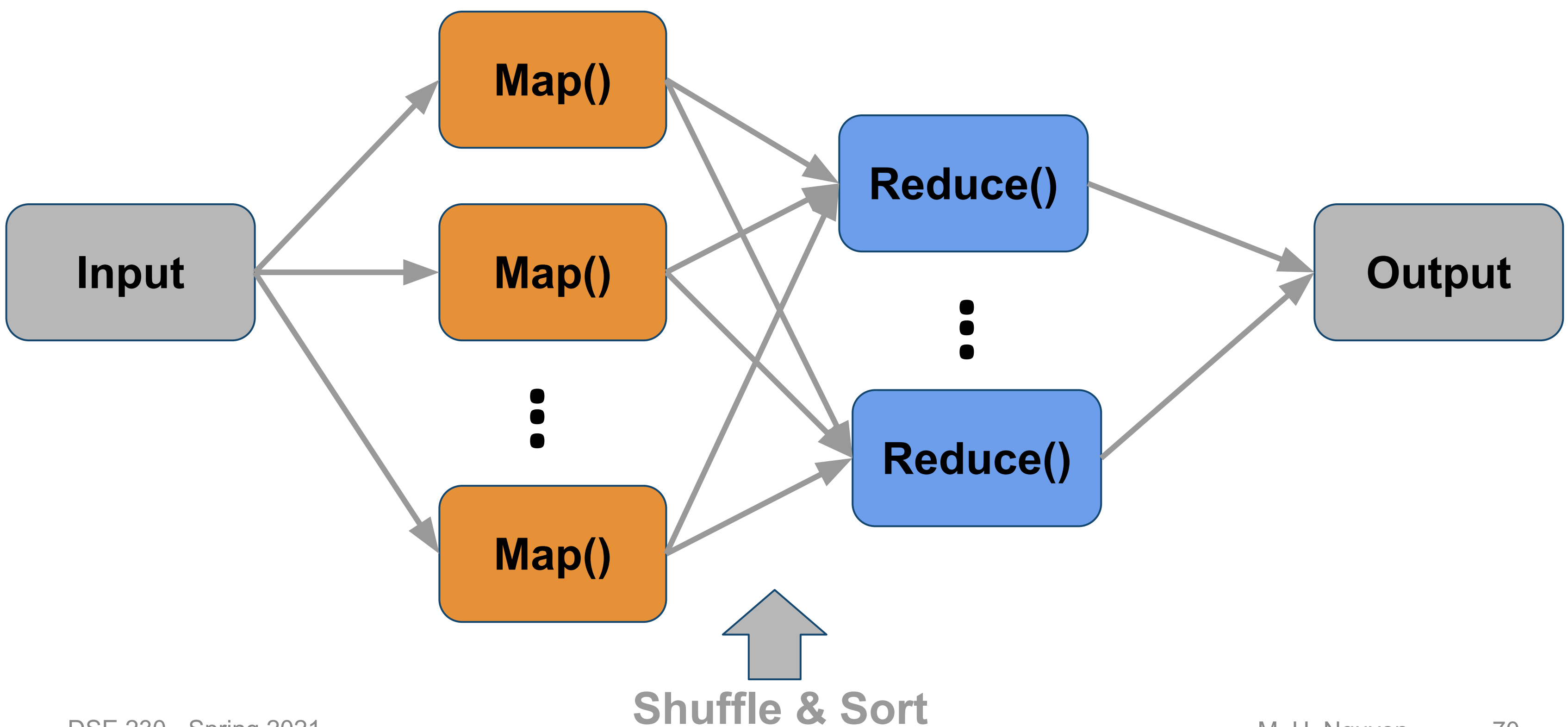
MapReduce

- Programming model for parallel processing on distributed system
- System implementation handles orchestration of data distribution, parallelization, synchronization, etc.
- Programmer doesn't have to worry about low-level mechanisms of parallel programming

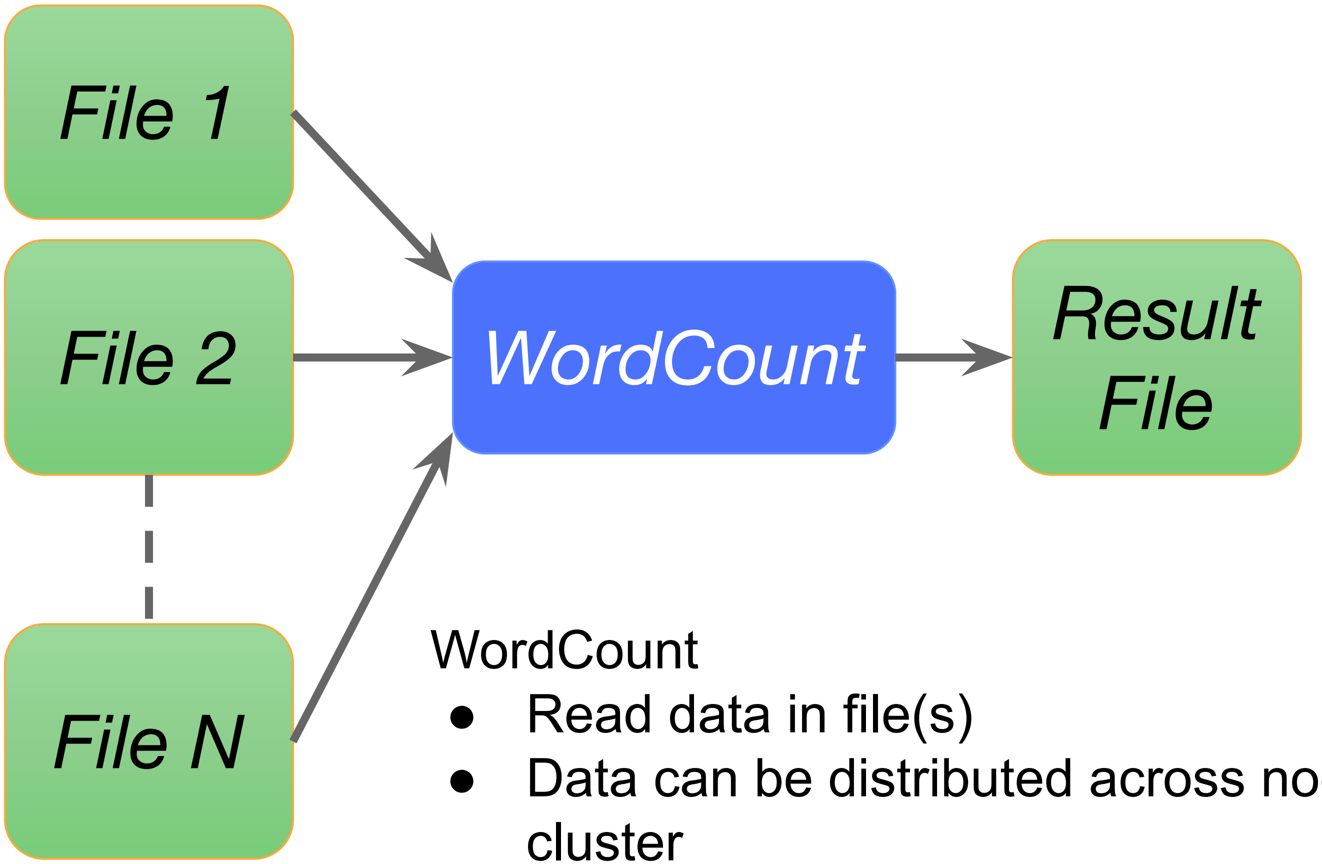


MapReduce

- **Map**: Apply operation to all data elements
- **Reduce**: Summarize elements

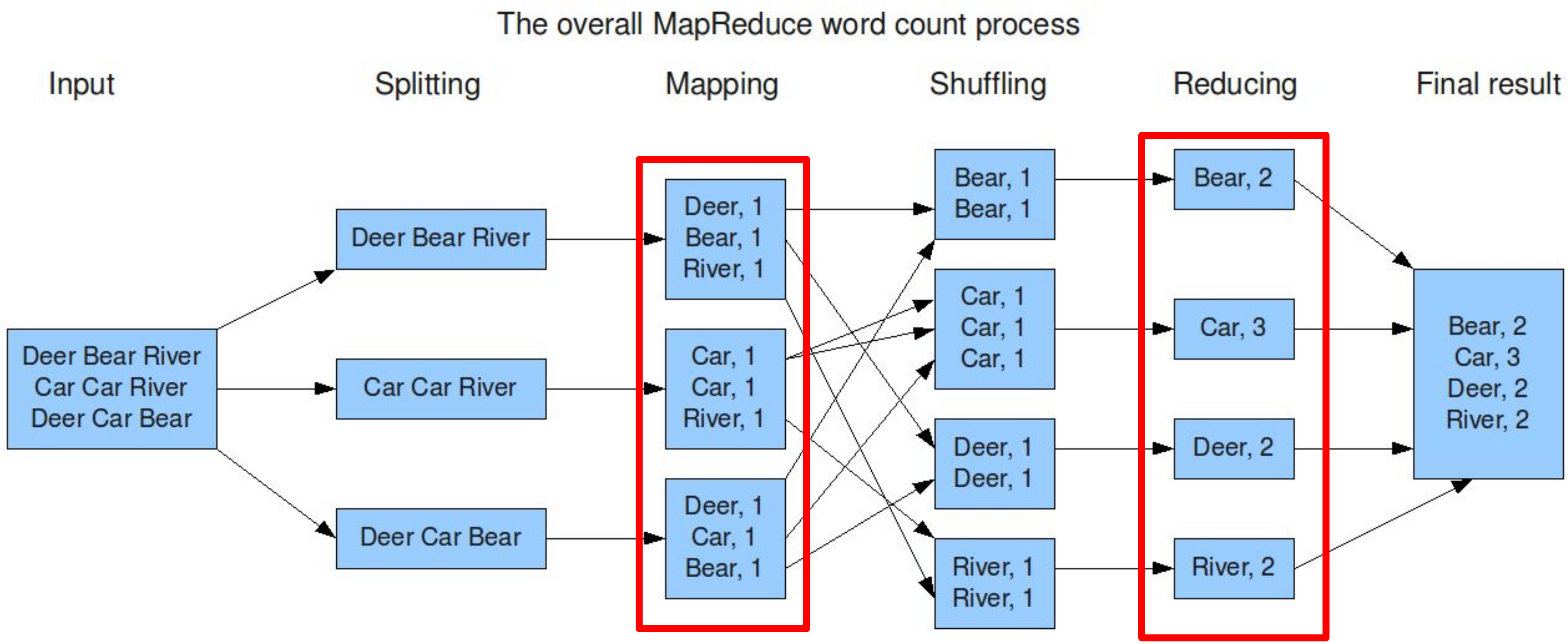


MapReduce: WordCount



WordCount
- Read data in file(s)
- Data can be distributed across nodes in cluster
- Count number of occurrences of each word

MapReduce: WordCount in Detail



https://www.todaysoftmag.com/article/1358/hadoop-mapreduce-deep-diving-and-tuning

Data is partitioned across nodes

Map generates key-value pairs

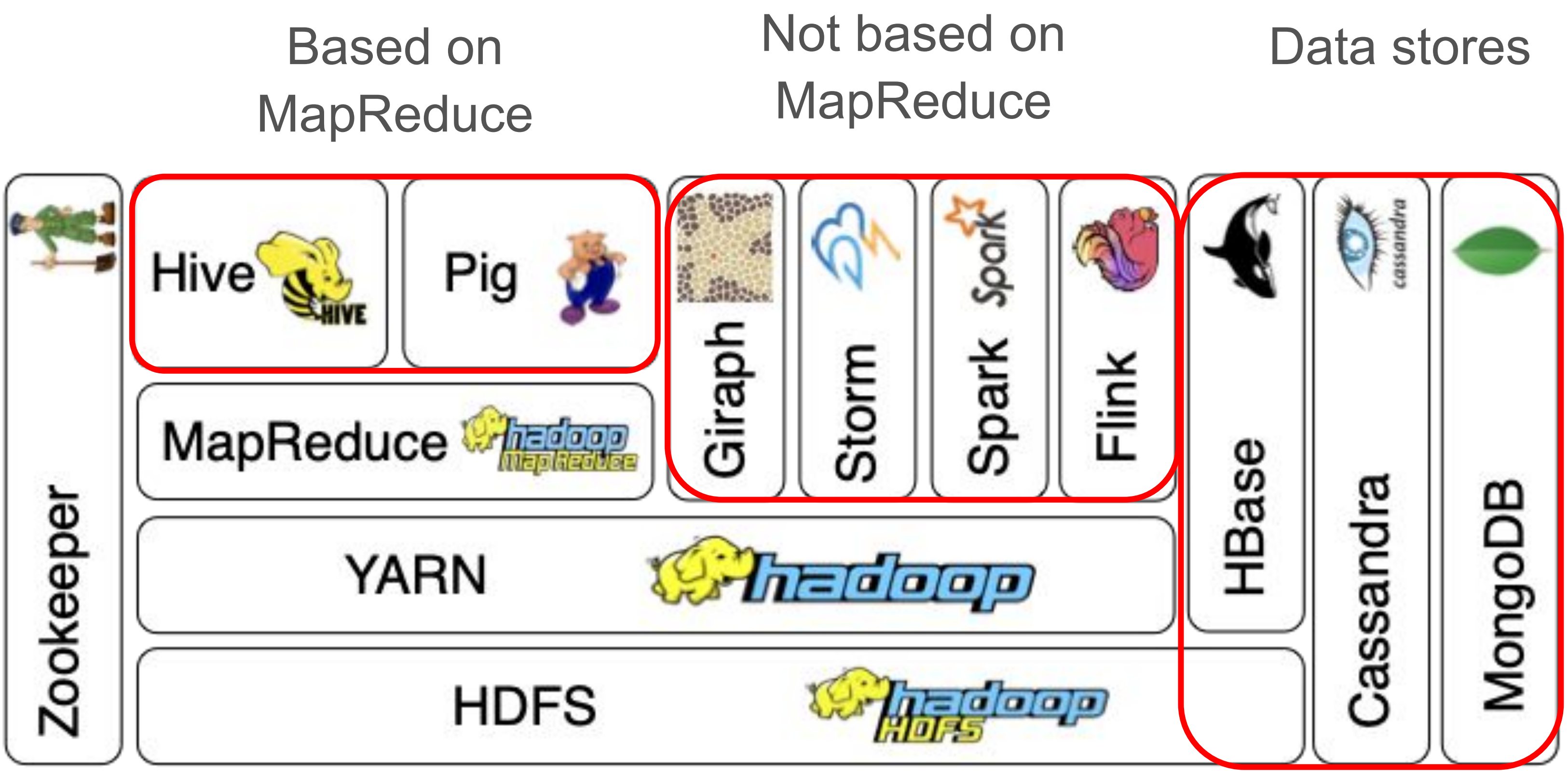
Pairs with same key moved to same node

Reduce sums values for each key

MapReduce

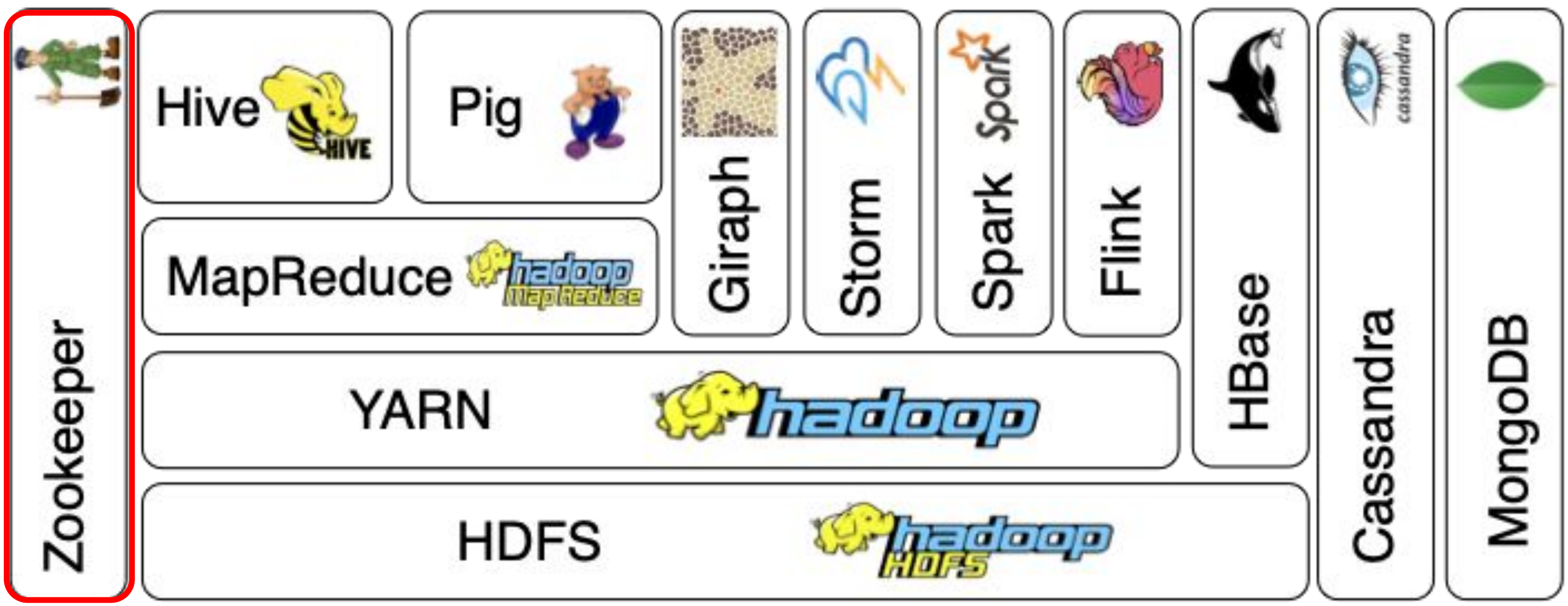
Node
Map
A
Node
Map
B
Node
Map
C
Node
Map
D
Node
word1, 1
word1, 1
Node
word2, 1
word2, 1
word2, 1
Node
wordN, 1
wordN, 1
(you, http://you1.fake)
(apple, http://apple1.fake)
(apple, http://apple2.fake)
(is, http://apple2.fake)
(is, http://apple2.fake)
(rose, http://apple2.fake)
(red, http://apple2.fake)
Reduce Results for "apple"
Key
Value
(apple -> http://apple1.fake,
http://apple2.fake)
apple

HIGH-LEVEL FUNCTIONALITY



HIGH-LEVEL FUNCTIONALITY

- Zookeeper: coordinates services in distributed environment



OTHER TOOLS

- Large community support
- Download separately or part of pre-built image
  - Cloudera, Hortonworks, MapR

SESSION 2 QUIZ

QUIZ

What are the main Vs of Big Data as discussed in Class?

A. volume, velocity
B. veracity, value
C. variety, value
D. A & B
E. A, B, & C

QUIZ

How big is a TB of data?

A. 10^12 bytes
B. 10^9 bytes
C. 1,000,000 bytes
D. Approximately equivalent to one 3-minute video

QUIZ

Which of the following is *false*:

A distributed system...

A. can support processing large data volumes
B. can handle fault tolerance
C. can only execute in a cluster of systems
D. can enable scalability
E. can leverage data parallelism

QUIZ

What is MapReduce?

A. A system implementation of Hadoop
B. A programming model that allows you to process large-scale data in parallel in a cluster environment
C. A resource manager in the Hadoop 2 ecosystem
D. A distributed file system that consists of Map, Split, and Reduce steps
E. A distributed platform created by Hadoop

QUIZ

In a distributed system, fault tolerance ...

A. Is not necessary since restarting a job can be accomplished by any of the nodes in the system
B. Happens rarely since there are many physical nodes in the system
C. Is difficult to achieve in a commodity cluster
D. Refers to the ability of the system to continue operating even when a node fails

HADOOP RESOURCES

- Hadoop: http://hadoop.apache.org/
- MapReduce: Simplified Data Processing on Large Clusters. Jeffrey Dean and Sanjay Ghemawat. In OSDI 2004.
- MapReduce Tutorial: http://bit.ly/2rS2B5j
- MapReduce for relational queries: http://bit.ly/2rkSRj8