General purpose machine – machine built with no specific app in mind, but rather they are capable of performing computations needed by a diversity of apps

Special purpose machine – machine built for specific apps

Language architecture – interface btw app progs and HLL

ISA – defines the interface bts the basic machine instruction set and I/O control

Computer architecture is built on

- The structure – defines d interconnect-ions of various hardware comps

- The Organization – defines d dynamic in-terface & management of the various components

- The Implementation - defines d detailed design of the hardware components

- The performance - specifies the behavior of the comp system

Transistor - a controlled on/off switch

Scale of integration
- SSI - Small Scale integration
- MSI - Medium Scale integration
- LSI - Large Scale integration
- MLSI - Very Large Scale integration
- WSI - Wafer Scale integration

Performance measure

A user measures the performance of a comp based on the time taken to execute a given job (programming). A metric for assessing the performance of a comp helps in

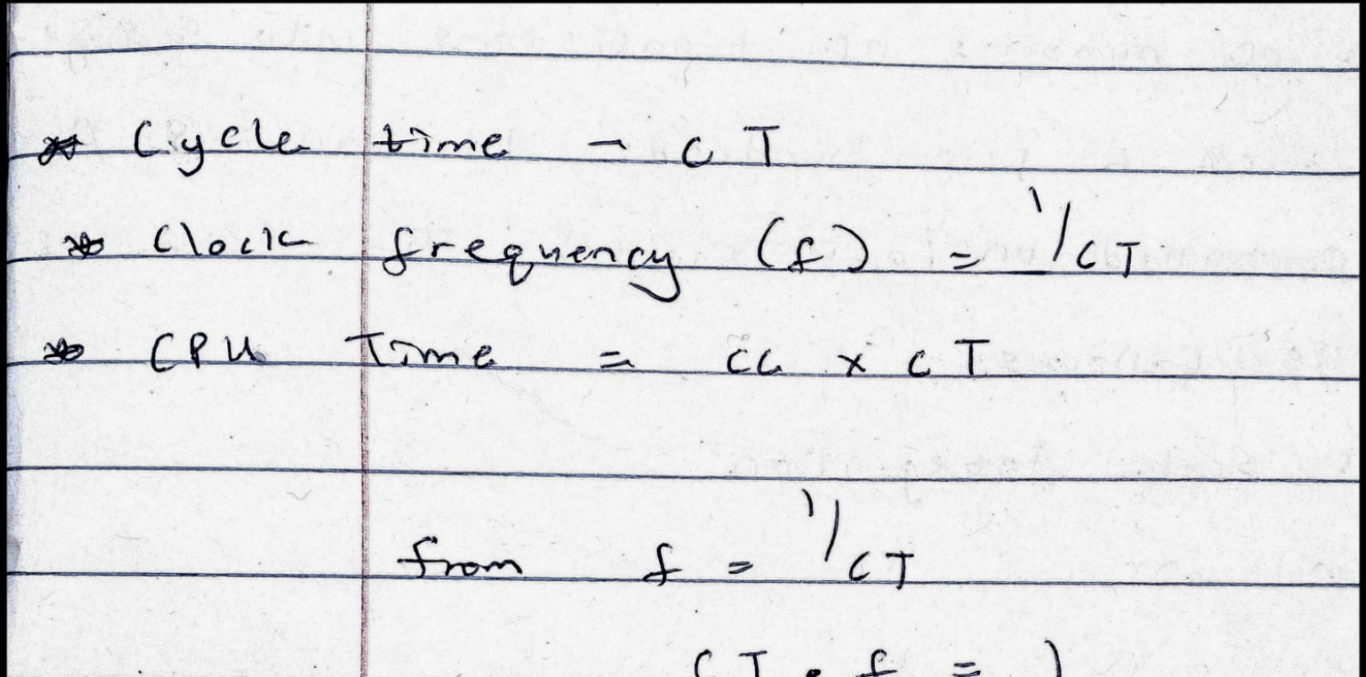
comparing alternative designs

Clock cycle time (CCT) – the time between two consecutive rising edge of a periodic clock signal.

The time required to execute a job is often expressed in terms of clock cycles

Cycle count (CC) – CPU clock signals for exe cuting a job

CPU time – time taken by the CPU to exec a job



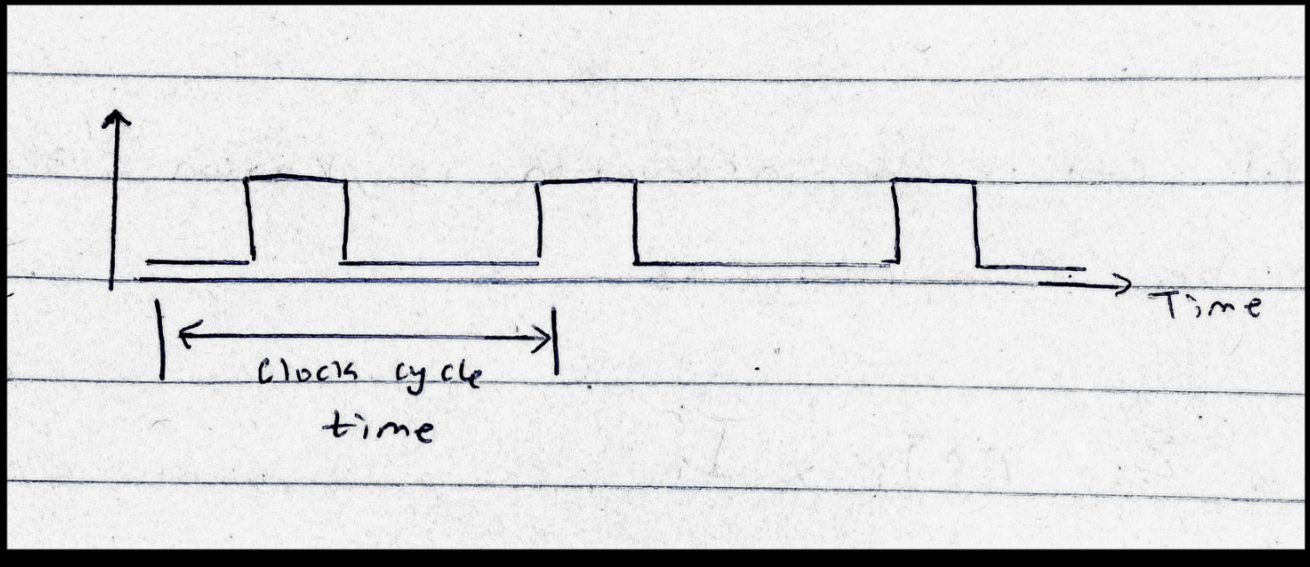
$CT = 1/f$

∴ CPU time = CC × CT

$= CC \times 1/f$

$= CC/f$

It is easier to count the no of instructions executed in a given prog as compared to counting the no of CCC needed for executing that program.



The avg no of clock cycle per instruction (CPI) has been used as an alternative measure

$$CPI = \frac{\text{CCC for the program}}{IC}$$

$$\text{CPU time} = IC \times CPI \times CCT$$

$$= \frac{IC \times CPI}{\text{Clock rate}}$$

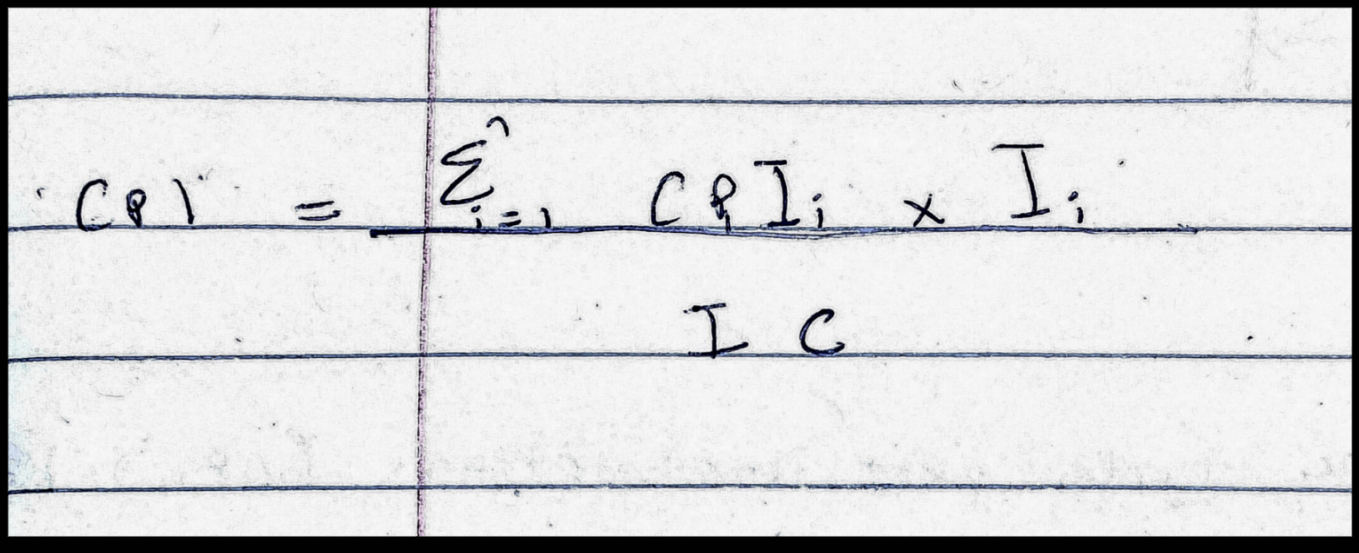
CC – cycle count
CCC – CPU clock cycle
CT – cycle time
CCT – clock cycle time
CPI – cycle clock per instruction
f – clock frequency

IC - instruction count

Instruction set (IS) - of a given machine consists of a no of instruction categories, which include

* Time load
* Branch
* ALU (Simple assignment & arithmetic & logic instructions)

If CPI for each instruction is known, the overall CPI can be computed as



Where

Ii - no of times an instruction of type i is

executed in the prog

CPIi – avg no of clock cycles needed to exe cute such instruction

Example – assume that the clock rate of a CPU is 200MHz. Consider computing the overall CPI for a machine A_9 for which the following performance measures were recorded when executing a set of bench-mark prog

Instruction category	% of occurence	No of cycle per instruction
ALU	38	1
Load & store	15	3
Brach	42	4
others	5	5

Solution

Assuming the execution of 100 instruct-ions, the overall CPI can be computed as

$$CPI_a = \frac{\sum_{i=1}^{n} CPI_i \times I_i}{IC}$$

$$= \frac{(39 \times 1) + (15 \times 3) + (42 \times 4) + (5 \times 5)}{100}$$

$$= \frac{38 + 45 + 168 + 25}{100} = {}^{276}/_{100}$$

$$= 2.76$$

The CPI reflects the organization and the ISA of the processor

The IC reflects the ISA and the computer technology used

The above statement shows the degree of interdependence btw the performance parameters, therefore, it is imperative that both the CPI and d IC are considered in assessing the merit of a given comp

Millions Instruction per second (MIPS) – is a diff performance measure that has been given a lot of attention in recent years.

MIPS (the rate of instruction execution per unit time) is defined as

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6}$$

$$= \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

Example – suppose the same set of bench mark program considered above were exe

cuted on another machine (B) for which the following measures were recorded

Instruction Category	% of occurence	No of cycles per instruction
ALU	35	1
Load & store	30	2
Branch	15	3
Others	20	5

What is the MIPS rating for the machine considered in d previous example (machine A) and machine B_9 assuming a clock rate of 200 mHz.

Solution

From machine A, $CPI_a = 2.76$

Therefore, the $MIPS_a = \frac{200 \times 10^6}{2.76 \times 10^6}$

$$= 72.46 \text{ mips}$$

For machine B

$$CPI_b = \frac{(35 \times 1) + (30 \times 2) + (15 \times 3) + (20 \times 5)}{100}$$

$$= \frac{35 + 60 + 45 + 100}{100}$$

$$= 2.4$$

$$\text{Therefore, } MIPS_b = \frac{200 \times 10^6}{2.4 \times 10^6}$$

$$= 83.33 \text{ mips}$$

We can see from above that MIPS (b) > MIPS (a). Literally, we can say dat machine B is faster than machine A bcos machine B processes more instructions than machine A.

N/B - one has to be careful in using MIPS to compare machines having different IS

Amdalh's law – this law is for speed up due to enhancement. In this case, we consider speed up as a measure of how a machine performs after some enhancements, rela tive to its original performance

$$\text{Speed up} = \frac{\text{Performance after enhancement}}{\text{Performance before enhancement}}$$

Alternatively, we can write

$$\text{Speed up} = \frac{\text{Execution time before enhancement}}{\text{Execution time after enhancement}}$$

Computer network - set of comps sharing resources located on or provided by network nodes. The comps use common communication protocols over digital interconnections, to communicate with each other

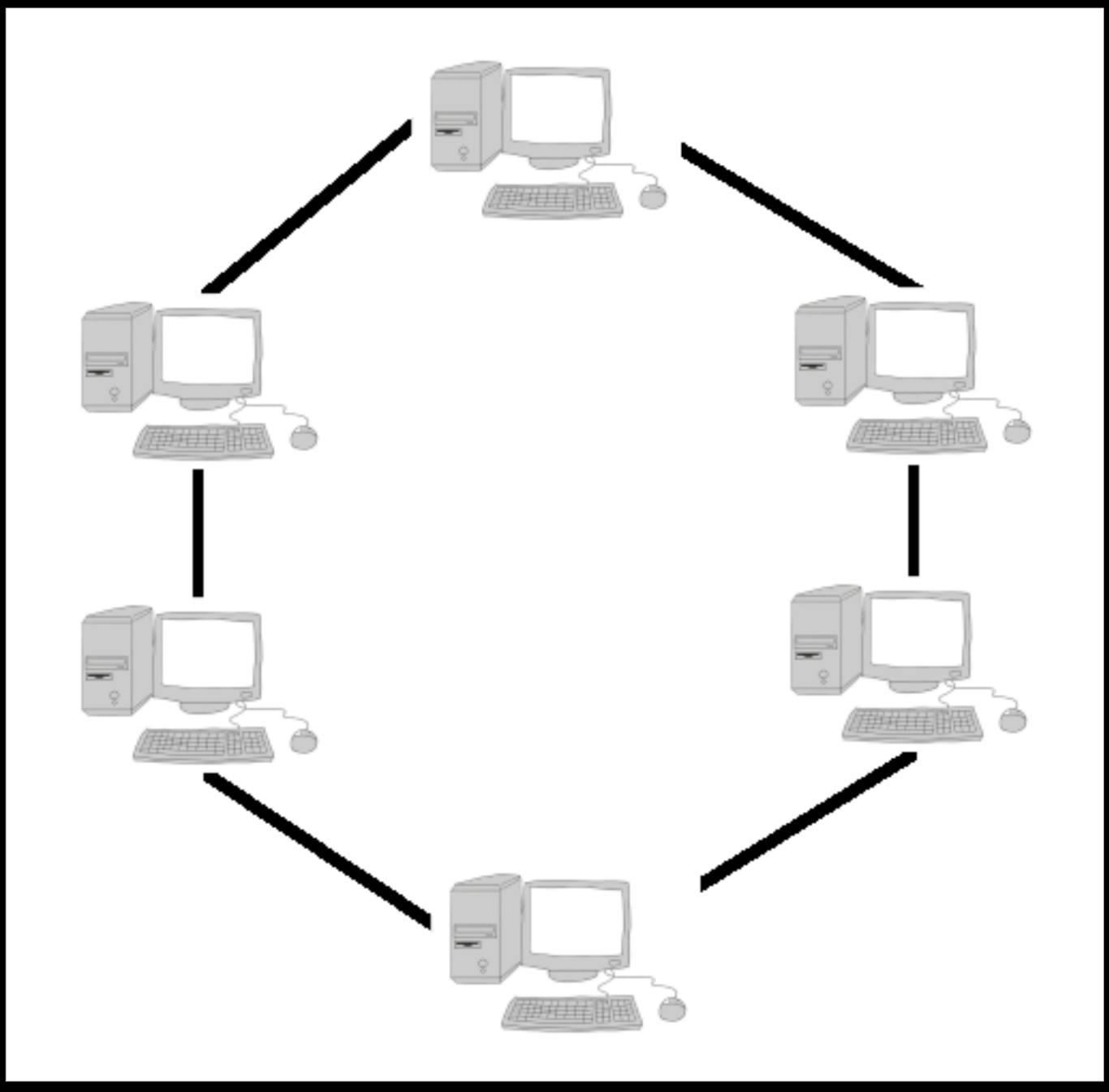
Network topology - the way a network is arranged, including the physical or logical description of how links and nodes are set up to relate to each other

Topologies are categorized as either

- Physical network topology - physical signal transmission medium
- Logical network topology - the manner in which data travels through the network btw devices, independent of physical connection of the devices

Loop/Ring topology - describes a topology of network whose components are serially connected in such a way that the last component is connected to the first

component. Loop topology is also known as arbitrated loop topology and may be used in either LANs or WANs



Why we use Loop/Ring topology (ADITA)

- The possibility of data collision is minimal

as it allows data flow in a unidirectional way.

- A network control server is not required in the ring topology to manage data transmission.

- Data can be sent at a faster rate.

- It is affordable than others as its op cost is economical.

- Any new nodes may be added without difficulty, and topology administration is simplified.

Application of Loop/Ring topology (TIMI)

- This topology may be used in LANs as well as WANs.

- In the telecommunications industry,

ring topology is commonly utilised in SONET (Synchronous optical network) fibre networks.

- Many organizations are also used the ring network as a backup system for their existing network.

- It is used in educational institutions due to its low cost of operation

Advantages of loop topology (DEPP)

- Every device has access to the token and the opportunity to transmit.

- Performs better than a bus topology under heavy network load.

- Does not require a central node to manage the connectivity between the computers.

- Point-to-point line configuration makes it easy to identify and isolate faults.

Disadvantages (BOMM)

- Bandwidth is shared on all links between devices
- More difficult to configure than a Star topology

- Moving, adding and changing the devices can affect the network

- One malfunctioning workstation can create problems for the entire network

Big data - data sets that are too large or complex to be dealt with by traditional data-processing application software.

Big data is a combination of structured, semistructured and unstructured data collected by organizations, that can be mined for information and used in machine learning projects, predictive modeling and other advanced analytics application.

The Four V's of Big data

- Volume – refers to the amount of big data from myriad sources

- Velocity – refers to the enormous speed with which data is generated and processed

- Variety – refers to the type of data
  - Structured
  - Semi-structured
  - Unstructured

- Veracity – refers to the degree to which big data can be trusted

Hadoop - is an open-source software framework for storing data and running apps on clusters of commodity hardware.

Hadoop ecosystem - a platform (or suite) that provides various services to solve big data problems

Components of Hadoop

- Hadoop HDFS (Hadoop Distributed File System) - storage unit of hadoop. HDFS is a distributed file system that provides access to data across Hadoop clusters (is a group of computers that work together).

  HDFS is a key tool that manages and supports analysis of very large volumes (petabytes and zettabytes of data).

Characteristics of HDFS

- It has a high fault tolerance

- It has a high throughput
- It consists of thousands of server machine
- It is designed to support large datasets in batch-style jobs

- **Hadoop MapReduce** – the processing unit of Hadoop. MapReduce is the process of making a list of objects and running an operation over each object in the list (i.e. map) to either produce a new list or calculate a single value (i.e., reduce).

- **Hadoop YARN (Yet Another Resource Negotiator)** – a resource management unit. YARN is a resource manager created by separating the processing engine and the management function of MapReduce.

Uses of YARN

- It monitors and manages workloads
- It maintains a multi-tenant environ
- It manages d high availability features

of Hadoop
- It implements security controls.

Data ingestion is the process of obtaining and importing data for immediate use or storage in a database. Data ingestion is a critical technology that helps organizations make sense of an ever-increasing volume and complexity of data.

Types of data ingestion
- Real time data ingestion - the process of collecting and transferring data from source systems in real time using solutions such as Change Data Capture (CDC).
  It is essential for time-sensitive use cases (stock market trading or power grid monitoring), when organizations have to rapidly react to new information.

- Batch-based data ingestion - d process of collecting and transferring data in

batches according to scheduled intervals.

Batch-based ingestion is useful when companies need to collect specific data points on a daily basis or simply don't need data for real-time decision-making.

- Lambda Architecture-based data ingestion - is a data ingestion setup that consists of both real-time and batch methods.

Benefits of data ingestion (CDDTT)

- Data is readily available
- Data is less complex
- Teams save time and money
- Companies make better decisions
- Teams create better apps and software tools

Data ingestion challenges

- The data ecosystem is increasingly diverse
- Legal requirements are more complex

- Cyber-security challenges grow in size and scope

Apache Sqoop (SQL-to-Hadoop) : big data tool designed to support bulk export and import of data into HDFS from structured data stores (relational DBs, enterprise data warehouses, and NoSQL systems)

Apache Flume - is an open-source system used to collect, aggregate and move large amounts of unstructured data from multiple data sources into HDFS/Hbase (for example) in a distributed fashion via it's strong coupling with the Hadoop cluster.

Difference between Sqoop and Flume

Basis for comparison	Apache Sqoop	Apache Flume
Basic difference	Sqoop is used for loading data from relational databases into HDFS	Flume is used to capture a stream of moving data
Load Type	The data load in Apache Sqoop is not driven by events	Apache Flume is an event-driven system
Architecture	The architecture of Apache Sqoop is connector based	Apache Flume features agent-based architecture
Data Flow	Sqoop is specifically used for parallel data transfer	Flume is used for collecting and aggregating data

Apache Kafka – is a distributed publish-subscribe messaging system and a robust queue that can handle a high volume of data and enables you to pass messages from one end-point to another.

Apache Pig – is a high-level platform or tool used to process large datasets. It provides a high-level of abstraction for processing over the MapReduce.

It provides a high-level scripting lang, known as Pig Latin which is used to develop the data analysis codes

Apache Hive – distributed, fault-tolerant data warehouse system that enables analytics at a massive scale. Hive is built on top of Apache Hadoop, and as a result, Hive is closely integrated with Hadoop.

Apache Spark – open-source, distributed processing system used for big data workloads. It utilizes in-memory caching, and optimized query execution for fast analytic queries against data of any size.

Components of Spark

- Shark (SQL)
- Spark Streaming (Streaming)
- MLLib (Machine Learning)
- GraphX (Graph Computation)
- SparkR (R on Spark)
- BlindDB (Approximate SQL)

These components are built on top of Spark Core Engine. Spark Core Engine allows writing raw Spark programs and Scala programs and launches them.

Difference btw Hadoop ecosystem & Spark ecosystem

Basis of comparison	Hadoop ecosystem	Spark ecosystem
Performance	Hadoop is slower because it stores data on multiple sources and processes it in batches via	Spark is faster because it used random access memory (RAM)

	MapReduce	
Cost	Hadoop runs at a lower cost since it relies on any disk storage type for processing	Spark runs at a higher cost because it relies on in-memory computations for real-time data processing
Processing	Hadoop is ideal for batch processing and linear data processing	Spark is ideal for real-time processing and processing live unstructured data streams
Security	Hadoop uses multiple authentication and access control methods	Spark enhances security with authentication via shared secret or event logging

Spark Architecture overview

Apache Spark has a well-defined layered architecture where all d spark components and layers are loosely coupled.

Apache Spark archi is based on two main abstractions

- Resilient Distributed Dataset (RDD) - are the building blocks of any Spark application. It is a layer of abstracted data over the distributed collection.

- Directed Acyclic Graph (DAG)

Advantages of Spark

- Speed - Spark can be 100x faster than Hadoop for large scale data processing by exploiting in-memory computing and other optimizations

- Ease of Use - Spark has easy-to-use APIs for operating on large datasets.

- A Unified Engine - Spark comes packaged with higher-level libraries, including support for SQL queries, streaming data, machine learning and graph processing