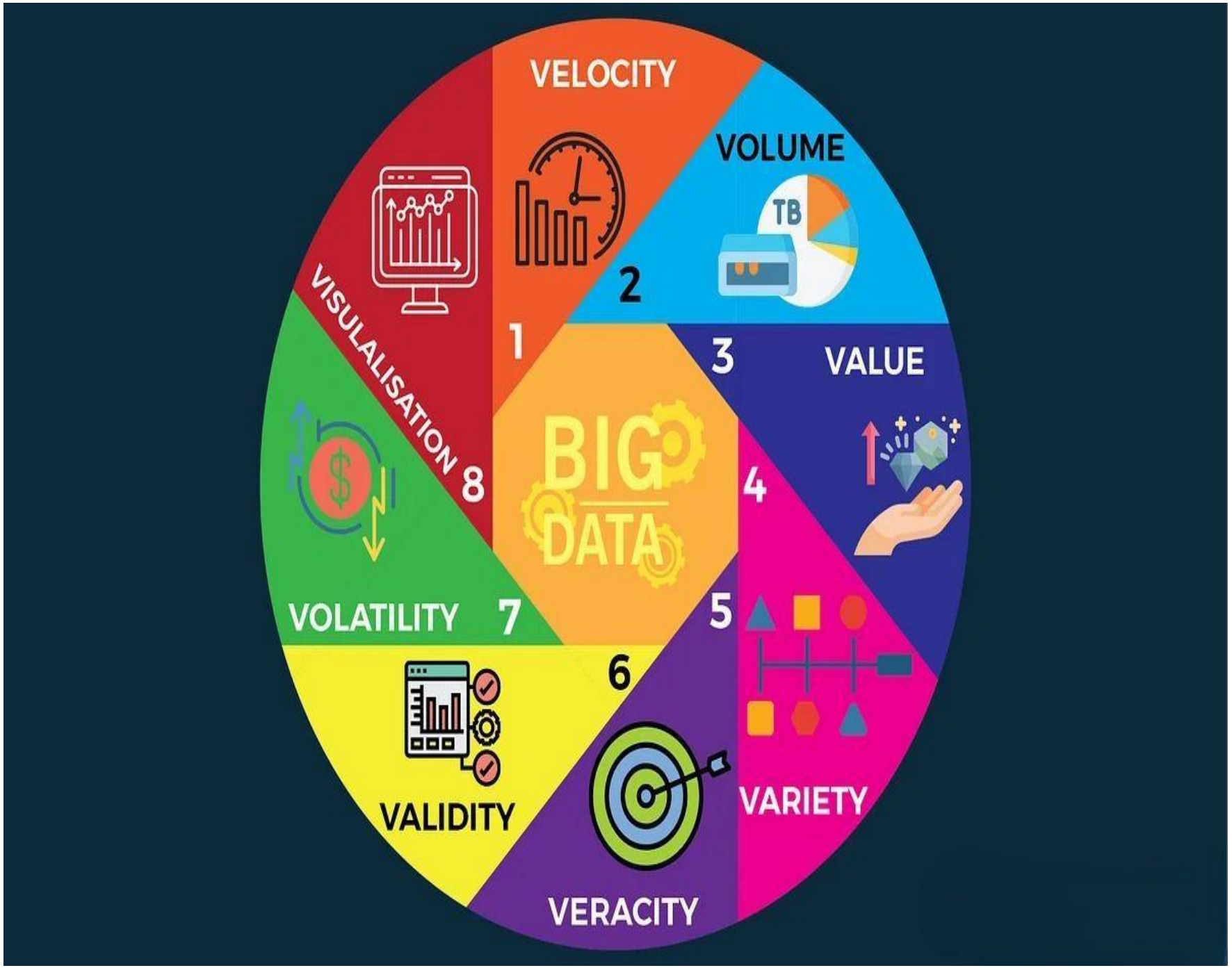
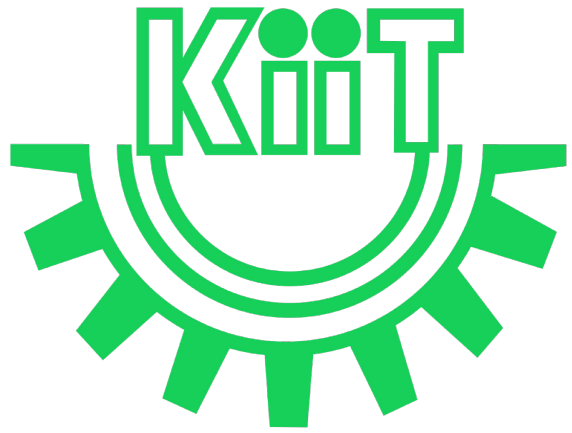


CS 3032: Big Data

Lec-3

In this Discussion . . .

- Key Big Data terminologies



Challenges preventing Businesses from capitalizing Big Data

Challenge	Description
● **Need For Indices Around Disparate Data Resources**	Data collections are becoming more significant and more diverse. It is a sizable challenge to incorporate them into a logical system that might overlook some activities.It's likely to produce openings and lead to wrong messages and insights.
● **Intense Deficiency of Experts Who Understand Big Data Analysis**	With Big Data in picture, Companies must look for data scientists. However, there exist a deficiency of experts who understand Big Data analysis. There is a short absence of data scientists when compared with the massive amount of data being created.

Challenges preventing Businesses from capitalizing Big Data

Challenge	Description
● **Obtaining Meaningful Insights Through Utilizing Big Data Analytics**	Company groups need to receive substantial insights from Big Data analytics. Also, the right segment must have access to this data; else, this massive gap can become a big headache for Companies.
● **Uncertainty Of Data Management Landscape**	With the growth of Big Data, brand-new technologies and companies have come up in the market in the last few years. However, companies still need to ascertain which technologies will benefit them else there might be issues introducing new problems and potential dangers.

Challenges preventing Businesses from capitalizing Big Data

Challenge	Description
• **Recruiting and maintaining large data capacity**	❏ There's no doubt a significant deficit of skilled and proficient folks in massive data. However, we've got data scientists, data miners. ❏ We also have data analysts and giant data specialists working annually. ❏ Most of these find themselves deviating from their favorite career. Or else they end up committing insights that fail to repair the problem under judgment.
• **Information Storage And Quality**	❏ With data generation rising quickly, storing such an enormous amount of data is becoming a real challenge for everyone. ❏ Popular data storage options, like data ponds/warehouses, are usually used to save massive quantities of unstructured and structured data in their native format. ❏ But experts faced challenges in data analysis as difficulty arises when a data pond/warehouse tries to combine inconsistent and sensitive data from diverse sources affecting the data quality.

Challenges preventing Businesses from capitalizing Big Data

Challenge	Description
• **Generating insights in some timely manner**	❏ Companies don't just want to put their big data. They want to use this enormous info to achieve business aims. ❏ That they can extract insights into their tremendous information; they can also act on those insights instantly. ❏ Everyone wants decision-making to become faster to react to improvements in the industry immediately.

• **Organizational resistance**	❏ It is more than just the technical aspects of extensive data that might be challenging: people could also be an issue. Concerning the impediments to this culture shift, economists pointed to three Big obstacles in Their institutions:
	  ❏ Inadequate organizational orientation
	  ❏ Deficiency of facility management adoption and understanding
	  ❏ Business resistance or lack of understanding
	❏ For companies to capitalize on the possibilities supplied by important info, they'll get to do a couple of things differently. And that kind of change might be tremendously difficult for big companies.

Top challenges facing Big Data

Challenge	Description
❑ **Scale**	❑ Storage is one primary concern that needs to be addressed to handle the need for scaling rapidly and elastically. The need of the hour is a storage that can best withstand the onslaught of large volume, velocity, and variety of big data. Should we scale vertically or horizontally?
❑ **Security**	❑ Most of the NoSQL (Not only SQL) big data platforms have poor security mechanism (lack of proper authentication and authorization mechanisms) when it comes to safeguarding big data.
❑ **Schema**	❑ Rigid schema have no place. The need of the hour is dynamic schema and static (pre-defined) schemas are passed

Top challenges facing Big Data

Challenge	Description
❑ **Data Quality**	❑ How to maintain data quality – data accuracy, completeness, timeliness etc. Is the appropriate metadata in place?
❑ **Partition Tolerant**	❑ How to build partition tolerant systems that can take care of both hardware and software failures?
❑ **Continuous availability**	❑ The question is how to provide 24/7 support because almost all RDBMS and NoSQL big data platforms have a certain amount of downtime built in.

Technologies to help meet the challenges posed by Big Data

- Cheap and abundant storage
- Faster processors to help with quicker processing of big data
- Affordable open-source, distributed big data platforms
- Parallel processing, clustering, visualisation, large grid environments, high connectivity, and high throughputs rather than low latency.
- Cloud computing and other flexible resource allocation agreements

Key Big Data Terminologies

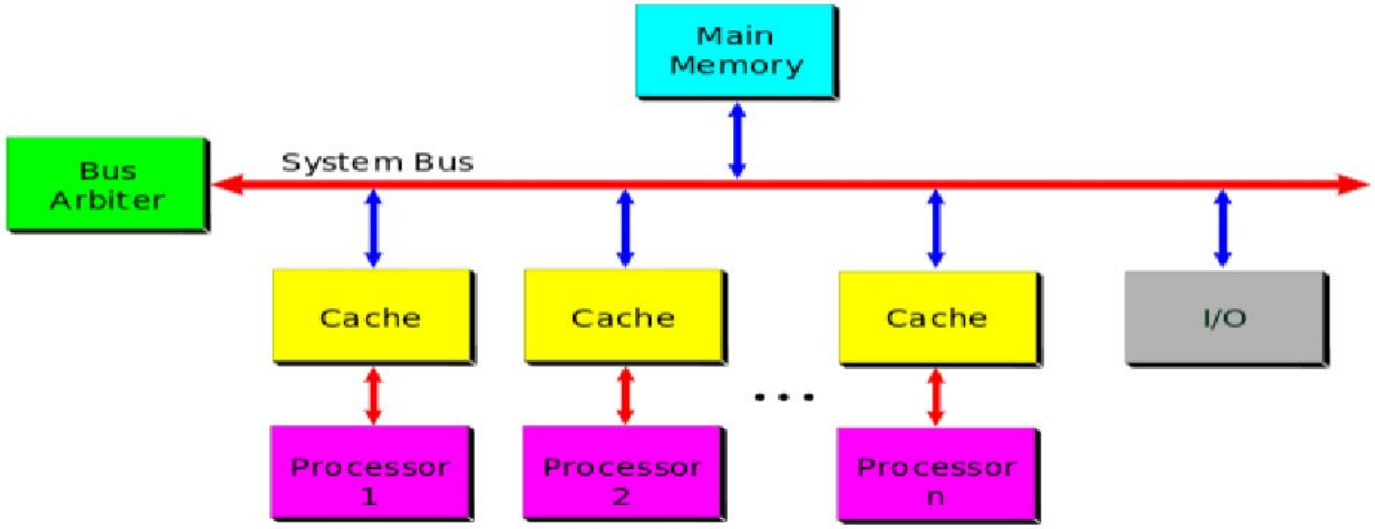
- **In-Memory Analytics:**
  - In-memory analytics is an approach to querying data when it resides in a computer's random access memory (RAM) instead of querying data stored on physical disks. This results in **vastly shortened query response times**, allowing business intelligence (BI) and analytic applications **to support faster business decisions, rapid deployment, better insights and minimal IT involvement**.
  - In-memory Analytics makes everything **Instantly Available** due to lower cost of RAM or Flash Memory, and data can be stored and processed at lightning speed.

Key Big Data Terminologies

- **In-Database Processing:**
  - Also called as *In-Database analytics*. It works by fusing data warehouses with analytical systems.
  - Typically the data from various enterprise Online Transaction Processing (OLTP) systems after cleaning up (deduplication, scrubbing etc.) through the process of ETL is stored in the Enterprise Data Warehouse or data marts. The huge datasets are then exported to analytical programs for complex and extensive computations.
  - With in-database processing, **all of the computations are done from a single program**. **This saves time, because the time needed for exporting is removed, and it speeds up the database to produce real-time results.**

Key Big Data Terminologies

- **Symmetric Multiprocessor System (SMP):**
  - In SMP, there is a single common main memory that is shared by two or more identical processors.
  - The processors have full access to all I/O devices and are controlled by a single operating system instance.
  - Each processor has its own high-speed memory, called cache memory and are connected using a system bus.

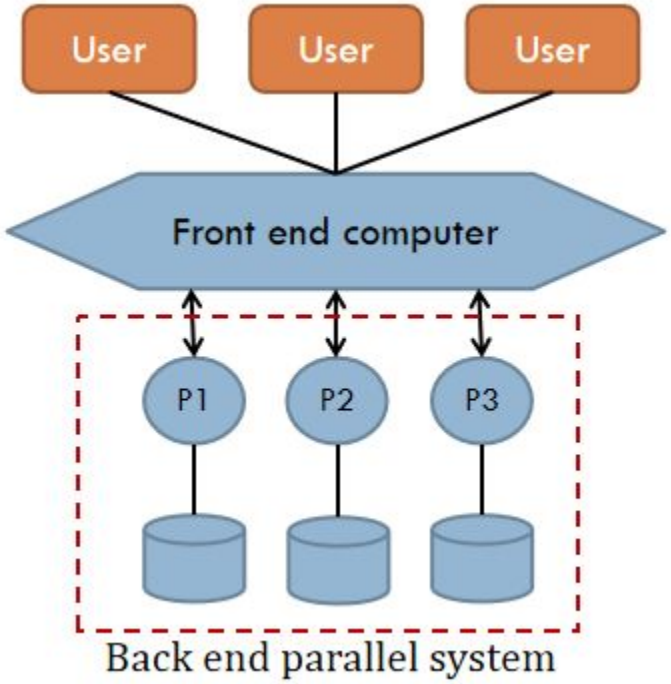


Key Big Data Terminologies

- **Parallel Systems:**
  - Companies need to handle huge amount of data with high data transfer rate. The client server and centralized system is not much efficient. **The need to improve the efficiency** gave birth to the **concept of Parallel Databases**.
  - Parallel database system improves performance of data processing using multiple resources in parallel, like multiple CPU and disks which are used parallely.
  - It also performs many parallelization operations like, data loading and query processing.

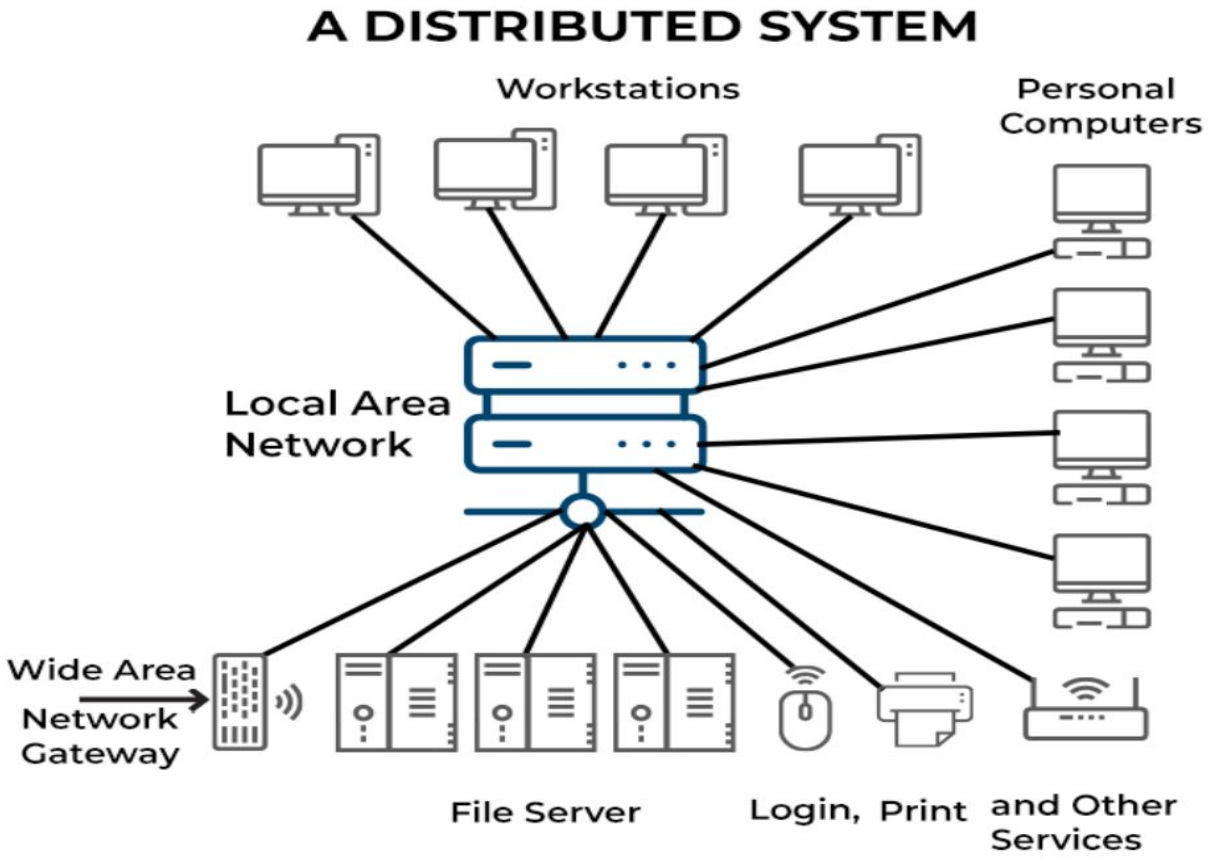
Key Big Data Terminologies

- **Parallel Systems:**
  - The processors co-operate for query processing. The user is unaware of the parallelism since he/she has no access to a specific processor of the system.



Key Big Data Terminologies

- **Distributed Systems:**
  - Distributed computing is a system of software components spread over different computers but running as a single entity. A distributed system can be an arrangement of different configurations, such as mainframes, computers, workstations, and minicomputers.



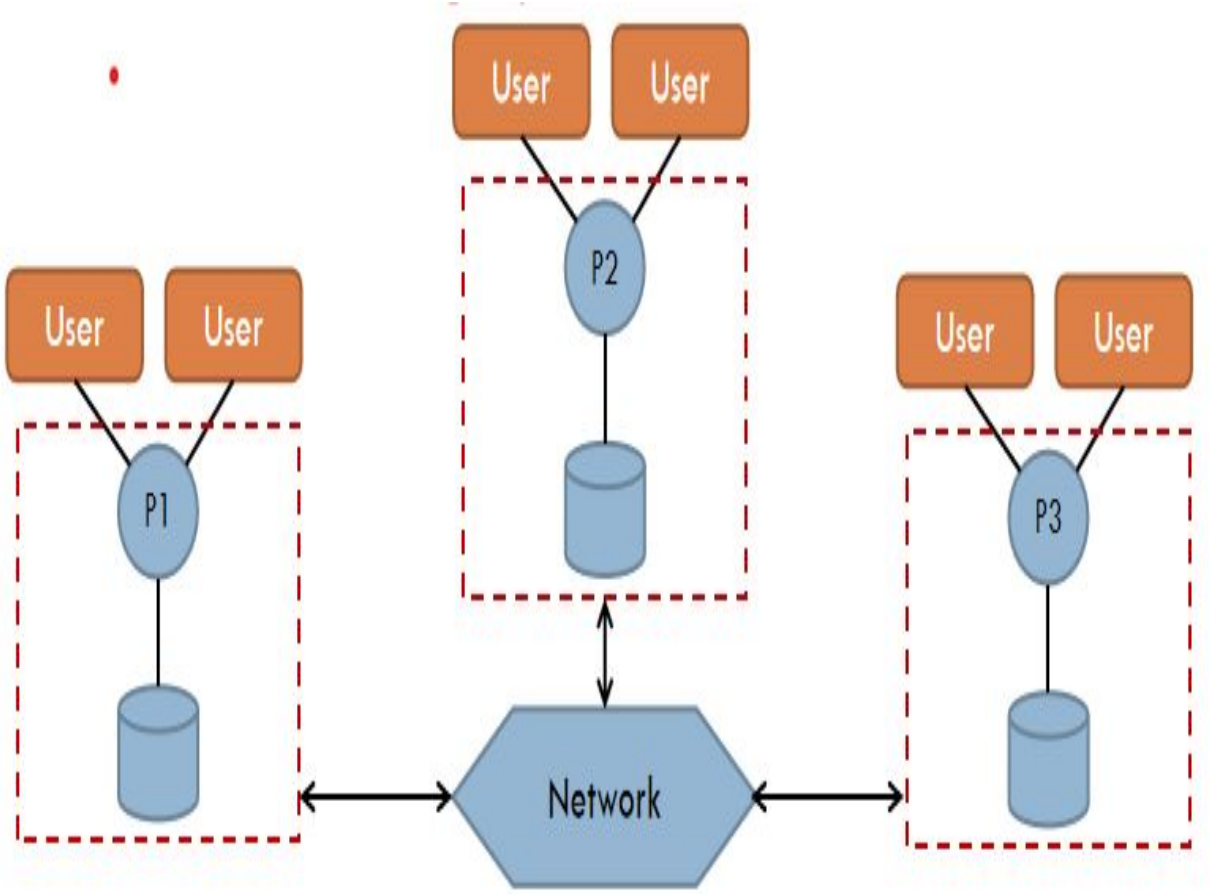
Key Big Data Terminologies

- **Distributed Systems:**
  - Distributed systems consist of many nodes that work together toward a single goal.
  - Known to be loosely coupled and are composed of individual machines.
  - Each of the machine can run their individual application and serve their own respective users. The data is usually distributed across several machines, thereby necessitating quite a number of machines to be accessed to answer a user query.

Key Big Data Terminologies

- **Distributed Systems:**

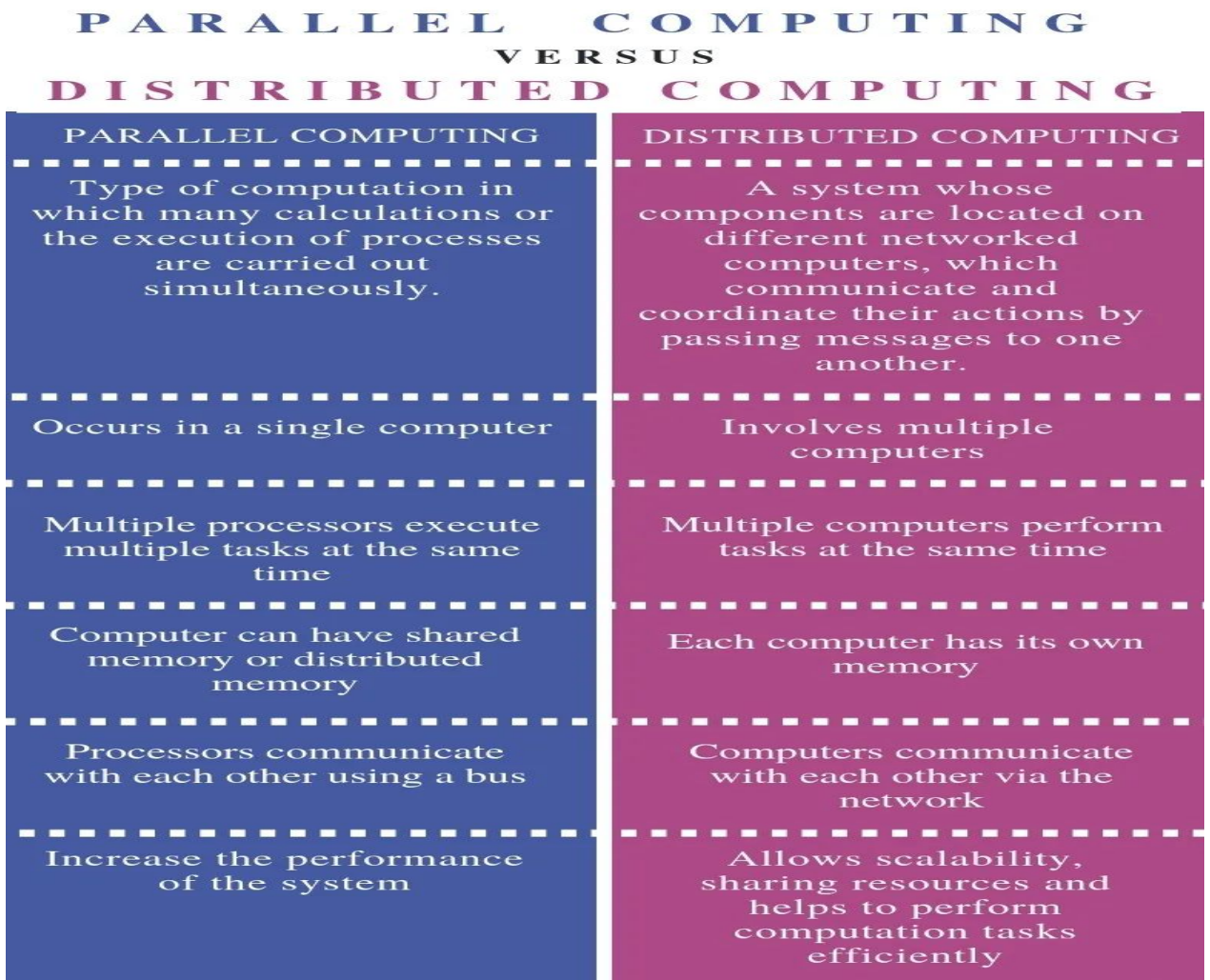
Distributed systems consist of several components spread across different computers but operate as a single network work together toward a single goal.



Key Big Data Terminologies

- **Distributed Vs. Parallel Computing:** The main difference between parallel and distributed computing is that parallel computing allows multiple processors to execute tasks simultaneously while distributed computing divides a single task between multiple computers to achieve a common goal.

Distributed computing in local network (called cluster computing). Distributed computing in wide-area network (grid computing)



Key Big Data Terminologies- System Architectures

- A DBMS's system architecture specifies what shared resources are directly accessible to CPUs. It affects how CPUs coordinate with each other and where they retrieve and store objects in the database.
- A single-node DBMS uses what is called a **shared everything** architecture. This single node executes workers on a local CPU(s) with its own local memory address space and disk.

Key Big Data Terminologies- System Architectures: Shared Memory

- An alternative to shared everything architecture in distributed systems is ***shared memory*** where CPUs have access to common memory address space via a fast interconnect. CPUs also share the same disk.
- In practice, most DBMSs do not use this architecture, as it is provided at the OS / kernel level. It also causes problems, since scope of each process memory is the same memory address space, which can be modified by multiple processes.
- Each processor has a global view of all the in-memory data structures. Each DBMS instance on a processor has to “know” about the other instances.

Key Big Data Terminologies- System Architectures: Shared Disk

- In a ***shared disk*** architecture, all CPUs can read and write to a single logical disk directly via an interconnect, but each have their own private memories. The local storage on each compute node can act as caches. This approach is more common in cloud-based DBMSs.
- The DBMS's execution layer can scale independently from the storage layer. Adding new storage nodes or execution nodes does not affect the layout or location of data in the other layer.

Key Big Data Terminologies- System Architectures: Shared Disk

- Nodes must send messages between them to learn about other node's current state. That is, since memory is local, if data is modified, changes must be communicated to other CPUs in the case that piece of data is in main memory for the other CPUs.
- Nodes have their own buffer pool and are considered stateless. A node crash does not affect the state of the database since that is stored separately on the shared disk. The storage layer persists the state in the case of crashes.

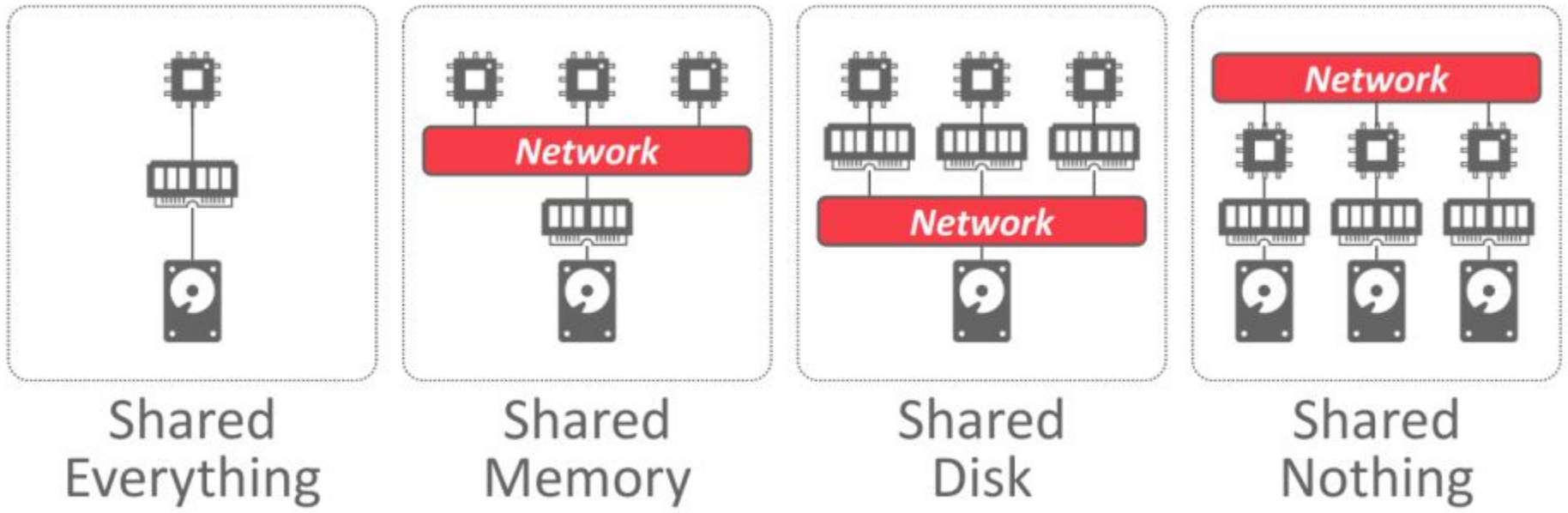
Key Big Data Terminologies- System Architectures: Shared Nothing

- In a ***shared nothing*** environment, each node has its own CPU, memory, and disk. Nodes only communicate with each other via network.
- Before the rise of cloud storage platforms, the shared nothing architecture used to be considered the correct way to build distributed DBMSs.
- It is more difficult to increase capacity in this architecture because the DBMS has to physically move data to new nodes.

Key Big Data Terminologies- System Architectures: Shared Nothing

- It is also difficult to ensure consistency across all nodes in the DBMS, since the nodes must coordinate with each other on the state of transactions.
- The advantage, however, is that shared nothing DBMSs can potentially achieve better performance and are more efficient than other types of distributed DBMS architectures.

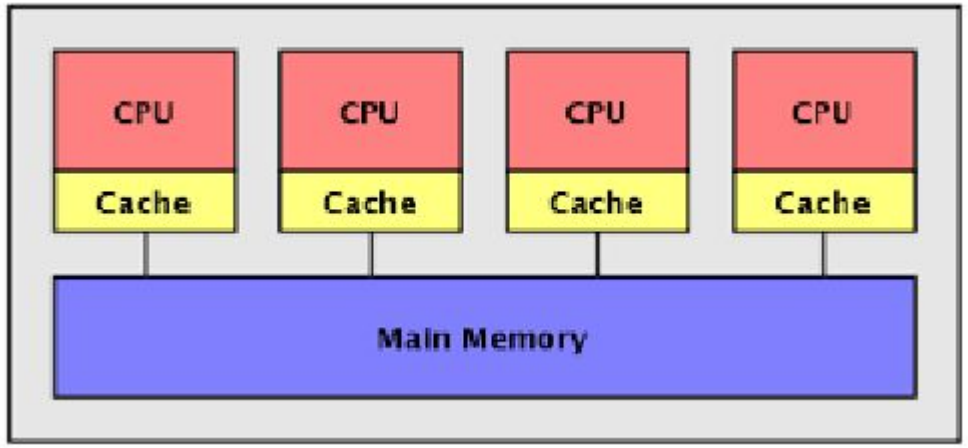
Key Big Data Terminologies- System Architectures: Shared Everything Vs. Shared Memory Vs. Shared Disk Vs. Shared Nothing

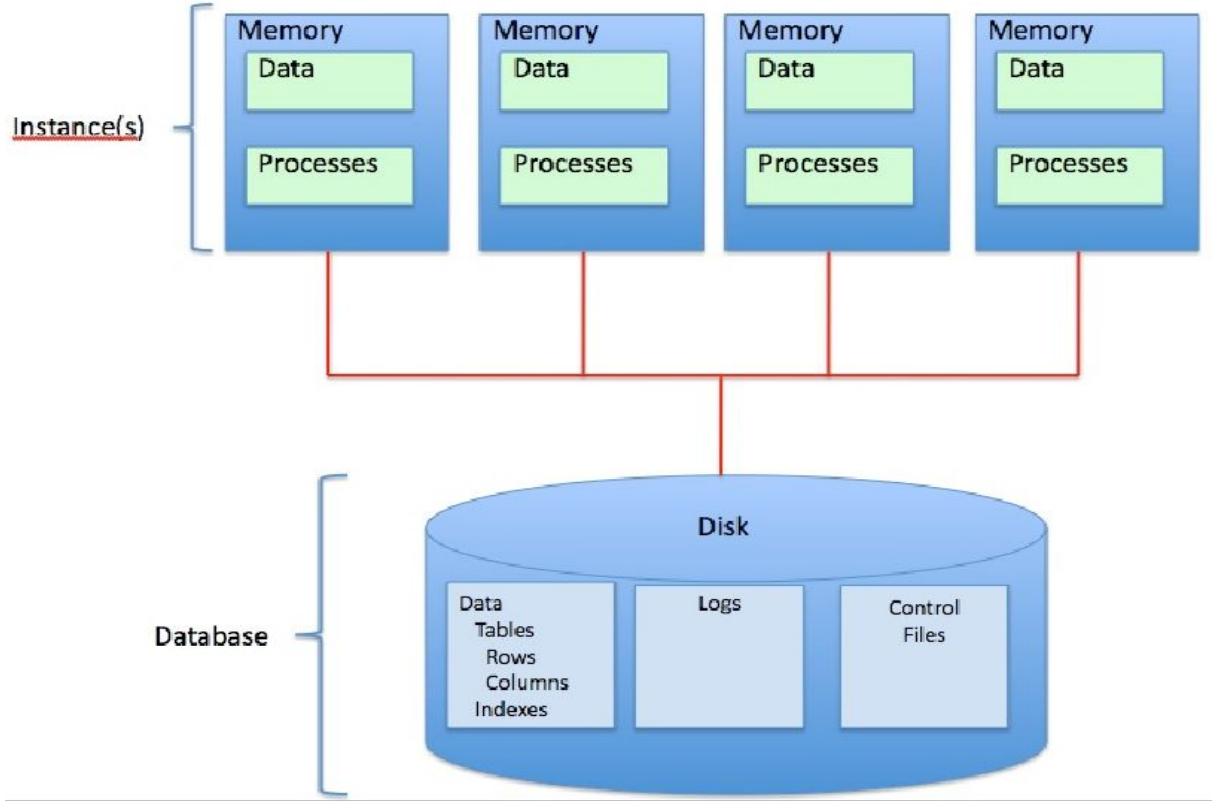


Four system architecture approaches ranging from sharing everything (used by non distributed systems) to sharing memory, disk, or nothing.

Key Big Data Terminologies- System Architectures: Shared Memory Vs. Shared Disk Vs. Shared Nothing

- In a shared memory (SM) architecture, a common central memory is shared by multiple processors. In a shared disk (SD) architecture, multiple processors share a common collection of disks while having their own private memory.





Key Big Data Terminologies- System Architectures: Shared Memory Vs. Shared Disk Vs. Shared Nothing

- In a shared nothing (SN) architecture, neither memory nor disk is shared among multiple processors.
- **Advantages:**
  - ***Fault Isolation***: provides the benefit of isolating fault. A fault in a single machine or node is contained and confined to that node exclusively and exposed only through messages.
  - ***Scalability***: If the disk is a shared resource, synchronization will have to maintain a consistent shared state and it means that different nodes will have to take turns to access the critical data. This imposes a limit on how many nodes can be added to the distributed shared disk system, thus compromising on scalability.

Key Big Data Terminologies- CAP Theorem

- In the past, when we wanted to store more data or increase our processing power, the common option was to **scale vertically** (**get more powerful machines**) or **further optimize the existing code base**.
- However, with the advances in parallel processing and distributed systems, it is **more common to expand horizontally**, **or have more machines to do the same task in parallel.**
- **However, in order to effectively pick the tool of choice like Spark, Hadoop, Kafka, Zookeeper and Storm in Apache project**, a **basic idea of CAP Theorem is necessary**.

Key Big Data Terminologies- CAP Theorem

- **The CAP theorem is often called the Brewer's Theorem after its originator, Eric Brewer.** It states that **a networked shared-data systems in distributed computing environment can only guarantee/strongly support two of the following three properties**: **Consistency**, **Availability** and **Partition Tolerant**, i.e., **one must be sacrificed.**

Consistency	❑ A guarantee that every node in a distributed cluster returns the same, most recent, successful write. ❑ Consistency refers to every client having the same view of the data.

Key Big Data Terminologies- CAP Theorem

- **The CAP theorem is often called the Brewer's Theorem after its originator, Eric Brewer.** It states that **a networked shared-data systems in distributed computing environment can only guarantee/strongly support two of the following three properties**: **Consistency**, **Availability** and **Partition Tolerant** – **one must be sacrificed.**

Availability	❏ A guarantee that every request receives a response about whether it was successful or failed. Whether you want to read or write you will get some response back. ❏ Reads and write always succeed. In other words, each non-failing node will return a response in a reasonable amount of time.

Key Big Data Terminologies- CAP Theorem

- **The CAP theorem is is often called the Brewer's Theorem after its originator, Eric Brewer.** It states that **a networked shared-data systems in distributed computing environment can only guarantee/strongly support two of the following three properties**: **Consistency**, **Availability** and **Partition Tolerant** – **one must be sacrificed.**

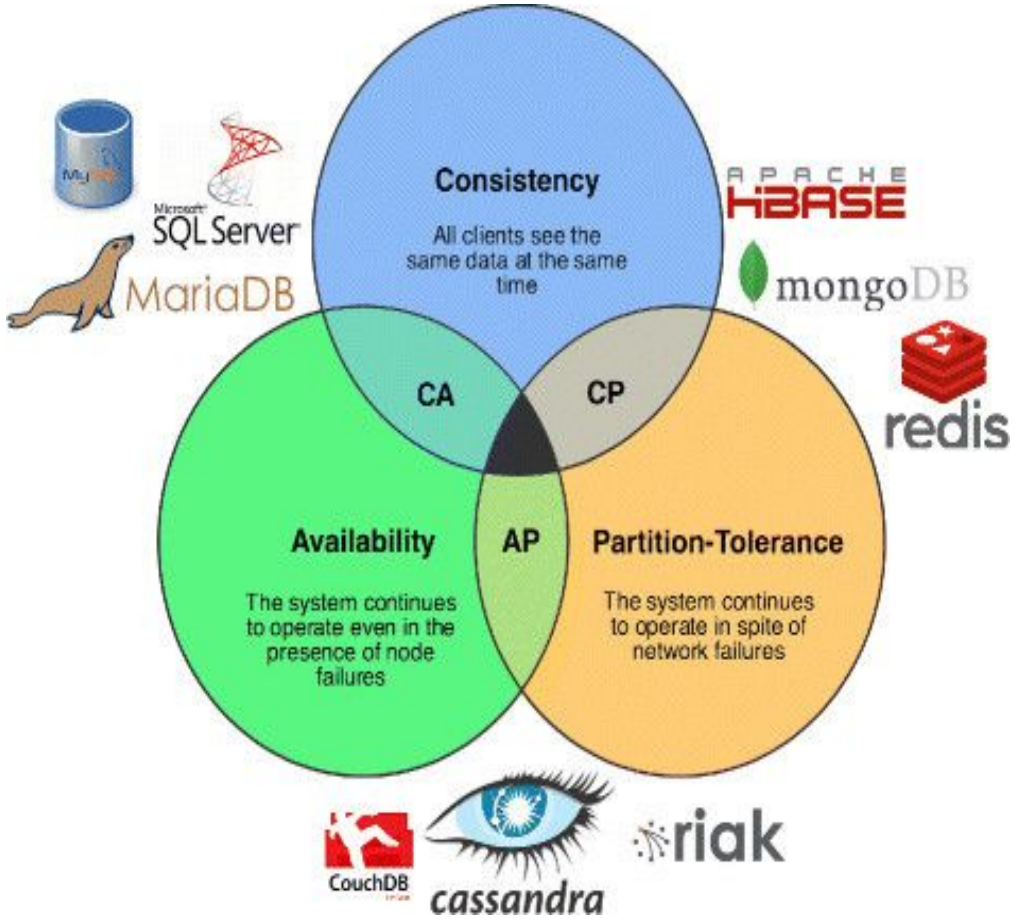
Partition Tolerant	❏ The system continues to operate despite arbitrary message loss or failure of part of the system. ❏ Irrespective of communication cut down among the nodes, system still works.

Key Big Data Terminologies- CAP Theorem

- *Often CAP theorem is misunderstood*. **It is not any 2 out of 3.**
- **Key point here** is **P is not visible to your customer. It is Technology solution to enable C and A. Customer can only experience C and A**.
- ***P is driven by wires, electricity, software and hardware and none of us have any control and often P may not be met. If P is existing, there is no challenge with A and C (except for latency issues). The problem comes when P is not met. Now we have two choices to make.***
- **The C and A in ACID represent different concepts than C and in A in the CAP theorem.**

Key Big Data Terminologies- CAP Theorem

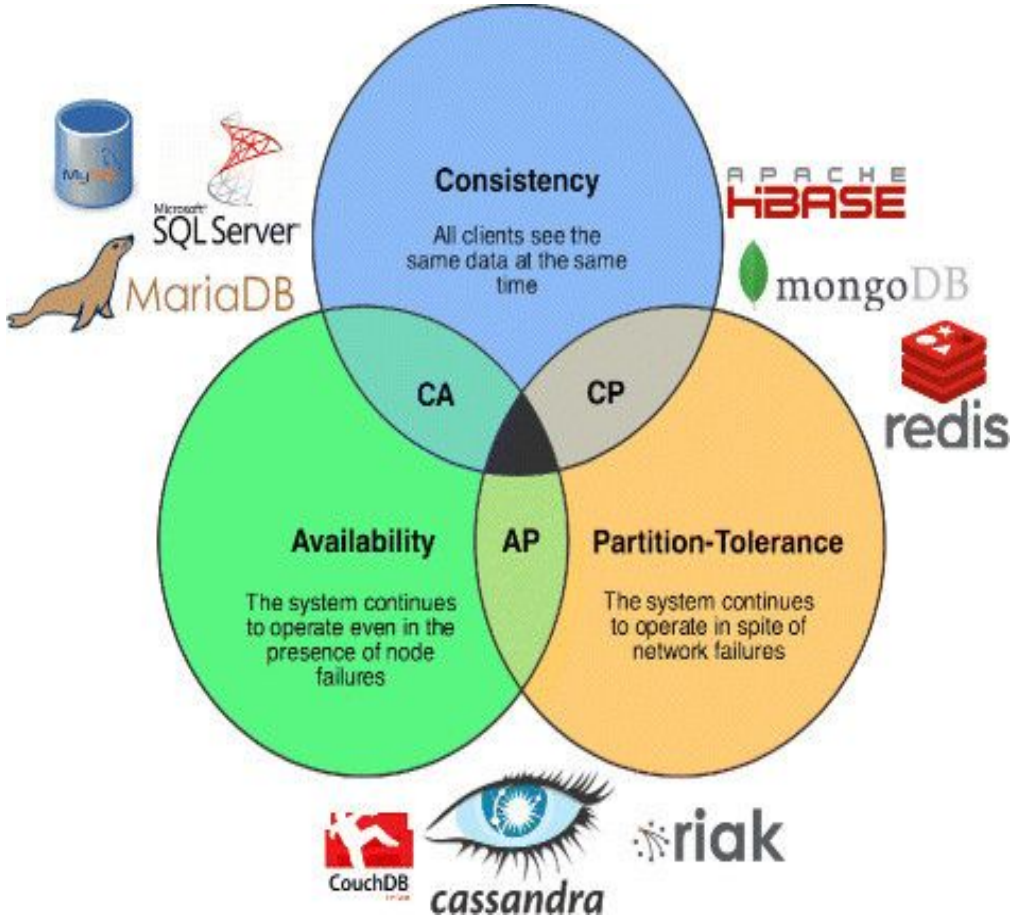
The CAP theorem categorizes systems into **three categories**:



1. **CP (Consistent and Partition Tolerant)** - At first glance, the CP category is confusing, i.e., a system that is consistent and partition tolerant but never available. CP is referring to a category of systems where availability is sacrificed only in the case of a network partition.

Key Big Data Terminologies- CAP Theorem

The CAP theorem categorizes systems into **three categories**:



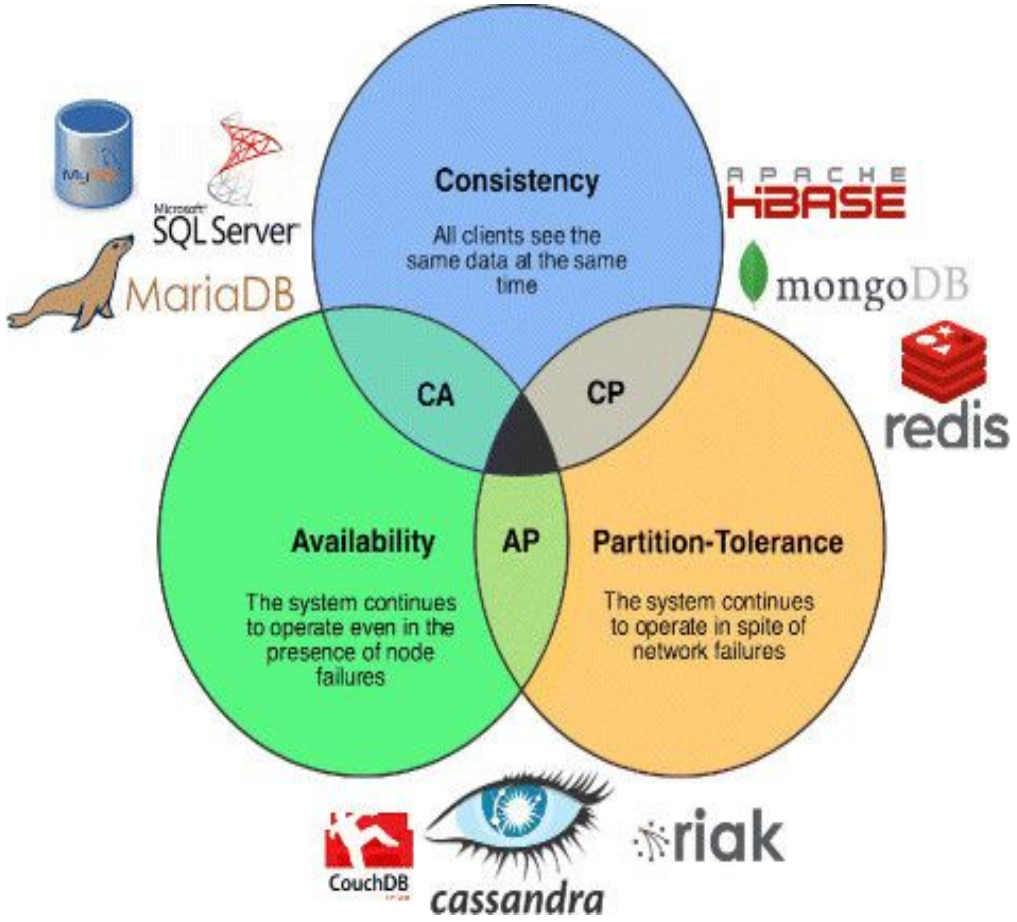
2. **CA** (**Consistent and Available**) - CA systems are consistent and available systems in the absence of any network partition.

   Often a single node DB servers are categorized as CA systems. Single node DB servers do not need to deal with partition tolerance and are thus considered CA systems.

   The only discrepancy in this theory is that single node DB systems are not a network of shared data systems and thus do not fall under the preview of CAP.

Key Big Data Terminologies- CAP Theorem

The CAP theorem categorizes systems into **three categories**:



3. **AP** (**Available and Partition Tolerant**) - These are systems that are available and partition tolerant but cannot guarantee consistency.

References

1. https://www.techiexpert.com/what-big-data-challenges-faced-by-business/
2. https://www.techtarget.com/searchbusinessanalytics/definition/in-memory-analytics
3. https://www.easytechjunkie.com/what-is-in-database-processing.htm
4. https://en.wikipedia.org/wiki/Symmetric_multiprocessing
5. https://www.tutorialride.com/parallel-databases/parallel-databases-tutorial.htm
6. https://pediaa.com/what-is-the-difference-between-parallel-and-distributed-computing/
7. https://15445.courses.cs.cmu.edu/fall2022/notes/21-distributed.pdf
8. https://www.linkedin.com/pulse/understanding-cap-theorem-sanjeev-singh
9.