# MPP, DS and Hadoop

Jitendra Kumar Samriya

Semester 8th

A massively parallel processing (MPP) system consists of a large number of small homogeneous processing nodes interconnected via a high-speed network.

- An MPP database is a type of database or data warehouse where the data and processing power are split up among several different nodes (servers), with one leader node and one or many compute nodes.

The processing nodes in an MPP machine are independent—they typically do not share memory, and typically each processor may run its own instance of an operating system,

although there may be systemic controller applications hosted on leader processing nodes that instruct the individual processing nodes in the MPP configuration on the tasks to perform.

# Main principles

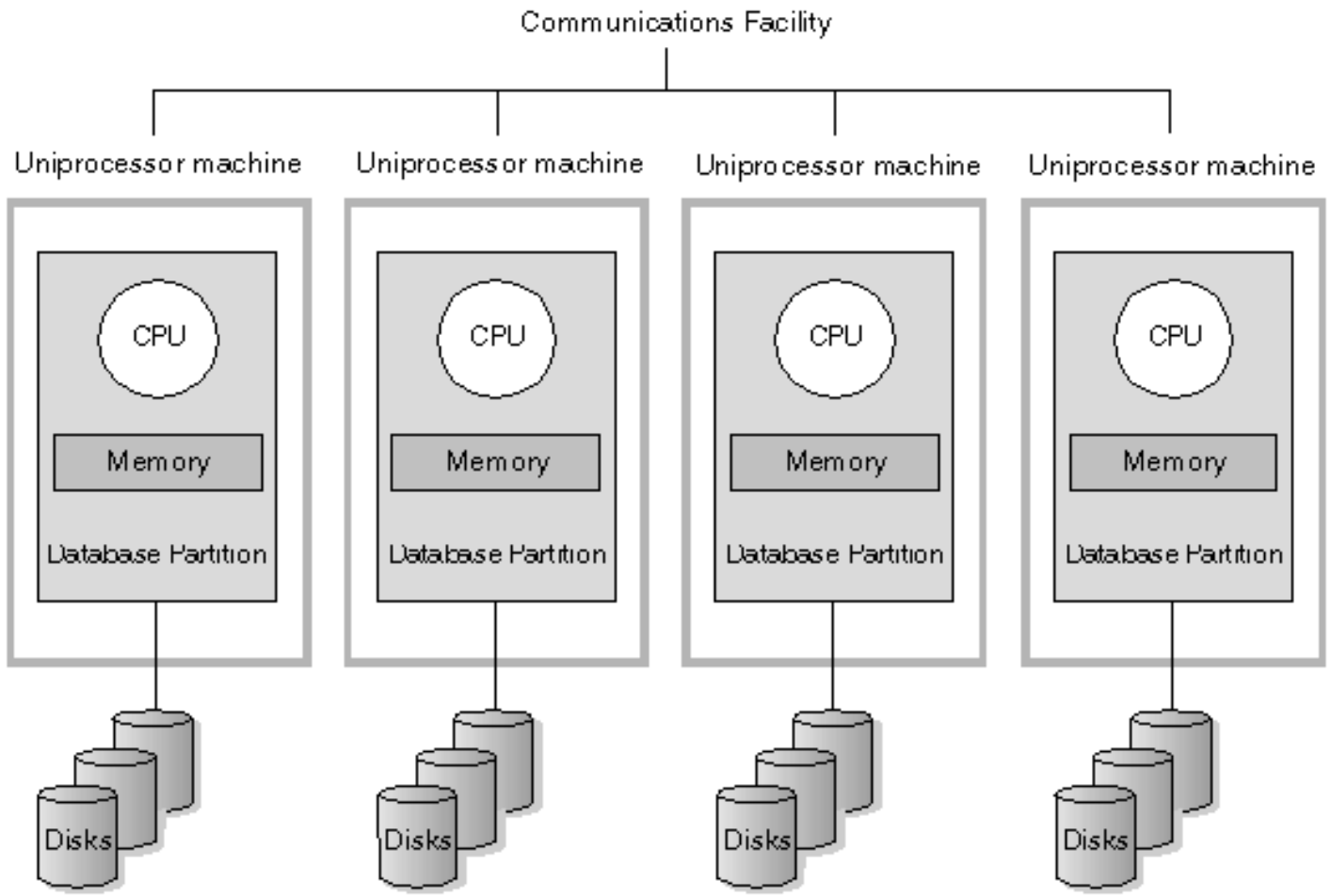
. Shared Nothing

. Data Sharding

. Data Replication

. Distributed Transactions

. Parallel Processing


Communications Facility
Uniprocessor machine
Uniprocessor machine
Uniprocessor machine
Uniprocessor machine
CPU
CPU
CPU
CPU
Memory
Memory
Memory
Memory
Database Partition
Database Partition
Database Partition
Database Partition
Disks
Disks
Disks
Disks

MPP is useful for

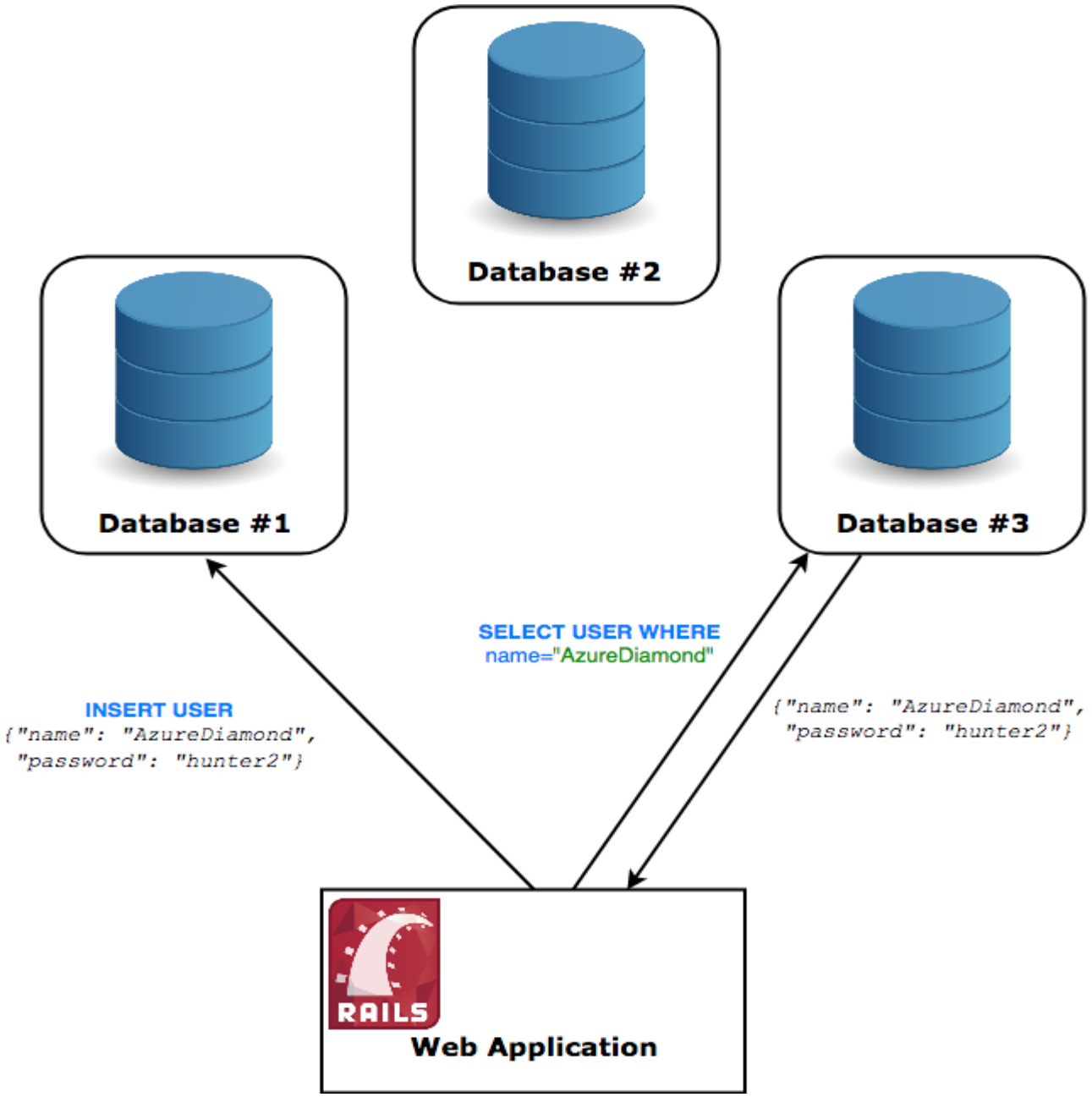
Relational data
Batch processing
Ad hoc analytical SQL
Low concurrency
Applications requiring ANSI SQL

# Distributed System DS

A distributed system in its most simplest definition is a group of computers working together as to appear as a single computer to the end-user.

These machines have a shared state, operate concurrently and can fail independently without affecting the whole system’s uptime.

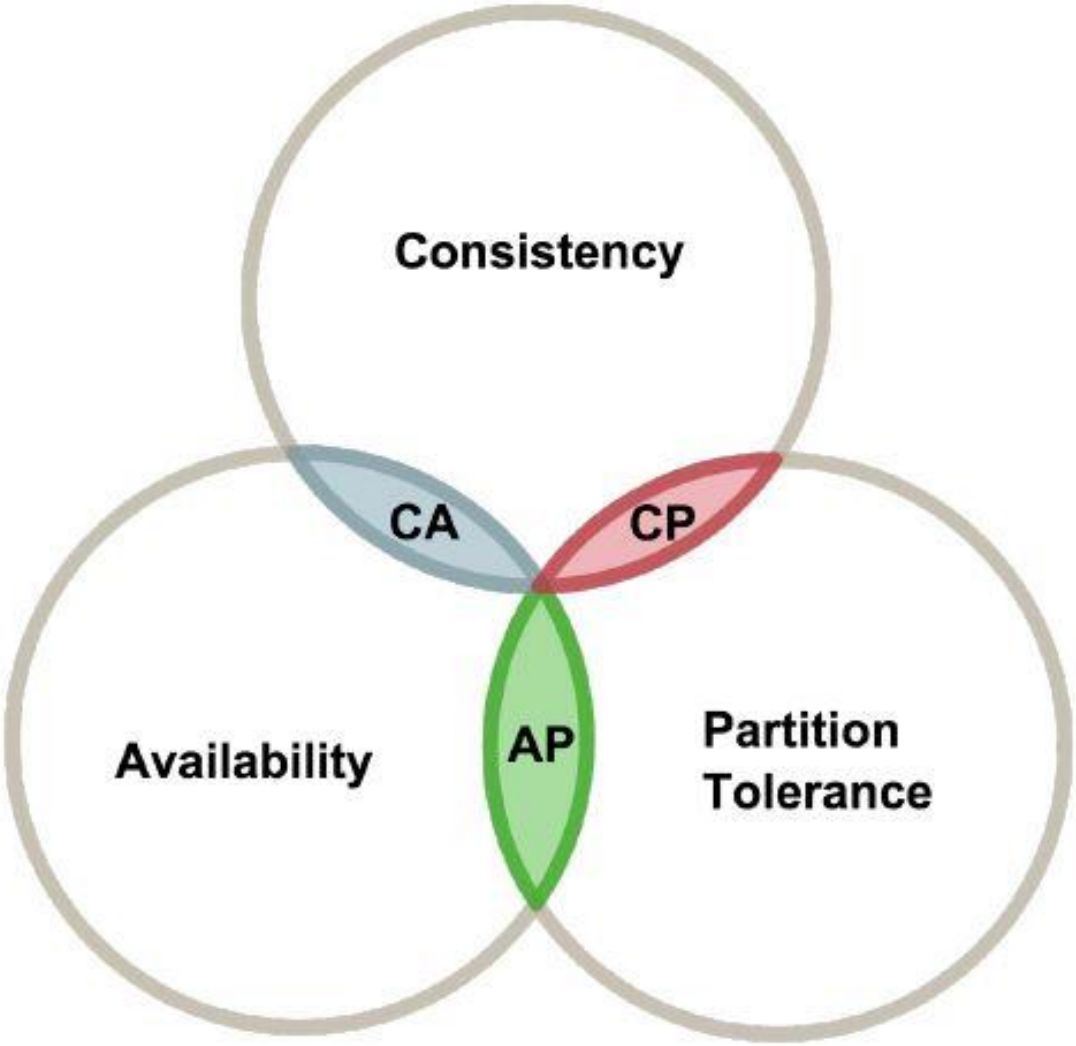
I propose we incrementally work through an example of distributing a system so that you can get a better sense of it all


Database #2
Database #1
Database #3
SELECT USER WHERE
name="AzureDiamond"
INSERT USER
{"name": "AzureDiamond",
"password": "hunter2"}
{"name": "AzureDiamond",
"password": "hunter2"}
RAILS
Web Application

## Why distributed computing is needed for big data

Example: you can distribute a set of programs on the same physical server and use messaging services to enable them to communicate and pass information. It is also possible to have many different systems or servers, each with its own memory, that can work together to solve one problem.

# CAP Theorem



Proven way back in 2002, the CAP theorem states that a distributed data store cannot simultaneously be consistent, available and partition tolerant.

**Consistency** — What you read and write sequentially is what is expected

**Availability** — the whole system does not die — every non-failing node always returns a response.

**Partition Tolerant** — The system continues to function and uphold its consistency/availability guarantees in spite of network partitions

you cannot have consistency and availability without partition tolerance.

# Hadoop

Hadoop is an open source, Java based framework used for storing and processing big data. The data is stored on inexpensive commodity servers that run as clusters. Its distributed file system enables concurrent processing and fault tolerance.

## Main Components

. HDFS
. YARN
. MapReduce
. HBase
. Hive / Hive+Tez

## How Hadoop Improves on Traditional Databases

Hadoop solves two key challenges with traditional databases:

1. Capacity: Hadoop stores large volumes of data.

By using a distributed file system called an HDFS (Hadoop Distributed File System), the data is split into chunks and saved across clusters of commodity servers. As these commodity servers are built with simple hardware configurations, these are economical and easily scalable as the data grows.

2. Speed: Hadoop stores and retrieves data faster.

Hadoop uses the MapReduce functional programming model to perform parallel processing across data sets. So, when a query is sent to the database, instead of handling data sequentially, tasks are split and concurrently run across distributed servers. Finally, the output of all tasks is collated and sent back to the application, drastically improving the processing speed.

# 5 Benefits of Hadoop for Big Data

For big data and analytics, Hadoop is a life saver. Data gathered about people, processes, objects, tools, etc. is useful only when meaningful patterns emerge that, in-turn, result in better decisions. Hadoop helps overcome the challenge of the vastness of big data:

**Scalability** — Unlike traditional systems that have a limitation on data storage, Hadoop is scalable because it operates in a distributed environment. As the need arises, the setup can be easily expanded to include more servers that can store up to multiple petabytes of data.
**Low cost** — As Hadoop is an open-source framework, with no license to be procured, the costs are significantly lower compared to relational database systems. The use of inexpensive commodity hardware also works in its favor to keep the solution economical.
**Speed** — Hadoop's distributed file system, concurrent processing, and the MapReduce model enable running complex queries in a matter of seconds.
**Data diversity** — HDFS has the capability to store different data formats such as unstructured (e.g. videos), semi-structured (e.g. XML files), and structured. While storing data, it is not required to validate against a predefined schema. Rather, the data can be dumped in any format. Later, when retrieved, data is parsed and fitted into any schema as needed. This gives the flexibility to derive different insights using the same data.

**Resilience** — Data stored in any node is also replicated in other nodes of the cluster. This ensures fault tolerance. If one node goes down, there is always a backup of the data available in the cluster.

# Thank You.