
DISTRIBUTED SYSTEMS
Krishna Turaga

AGENDA

- Introduction
- Characteristics
- Types Of Distributed Computing
- Distributed System Categories
- CAP Theorem
- Advantages
- Disadvantages
- Questions

Distributed Systems

- A distributed system is a collection of separate and independent components, called nodes,
- A group of Nodes are networked and work together coherently by coordinating and communicating through messages passing or events, to fulfill one end goal.
- They may connect to offer some service, share data or simply store data.
- The complexities are hidden away from the end user, making the whole system appear as one computer.
- These machines have a shared state, operate concurrently, and can fail independently without affecting the whole system's uptime.

Characteristics

- Resource Sharing
- Scalability
- Fault tolerance
- No shared clock.
- No shared memory.
- Concurrency.
- Heterogeneity and Loose Coupling

Types Of Distributed Computing

- Cluster Computing
- Grid Computing
- Cloud computing

Cluster Computing

- Homogenous
- Management is centralized
- Used for high performance and minimum downtime

Grid Computing

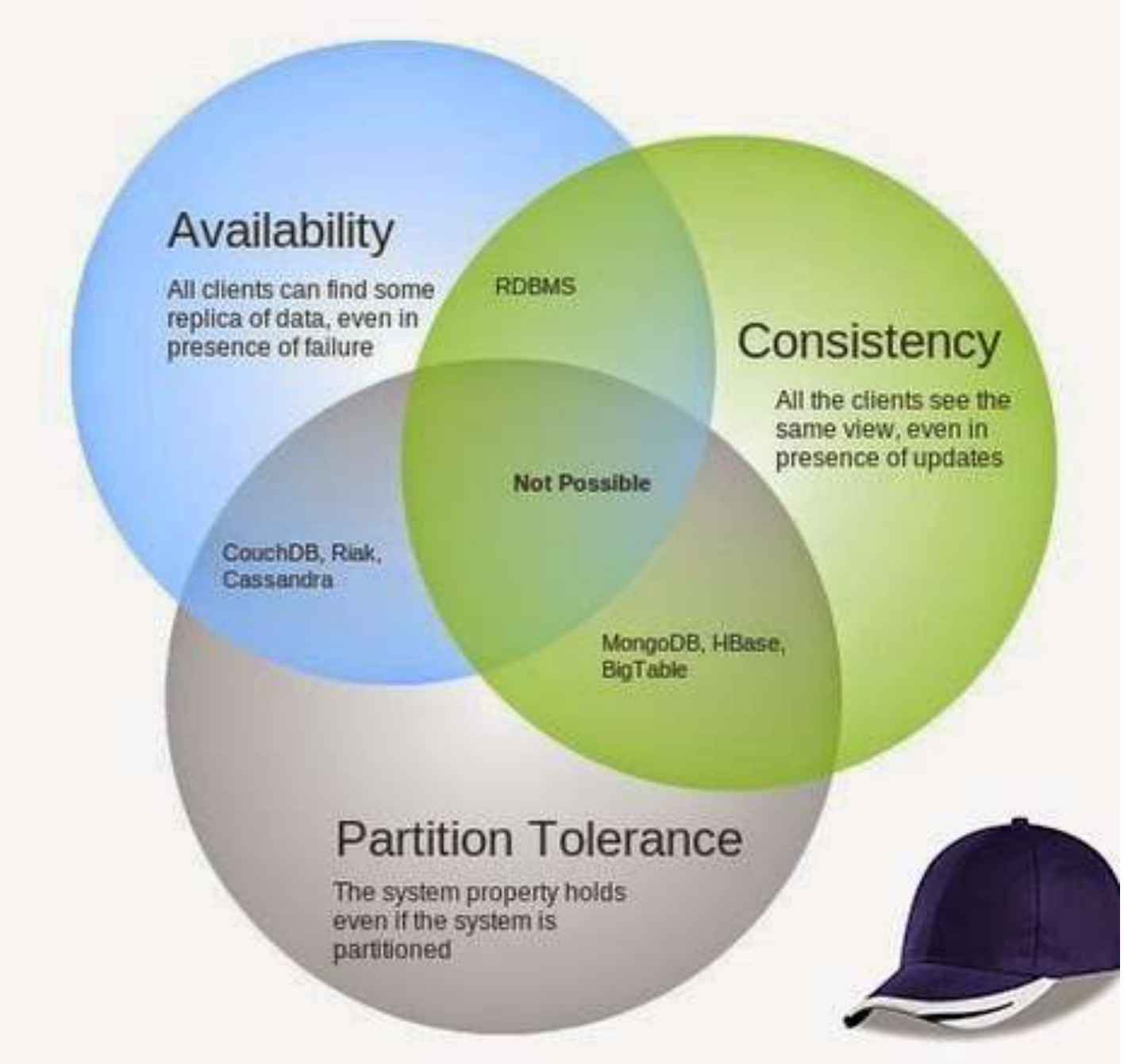
- Heterogenous and geographically dispersed
- Management is decentralized
- Used for large repository of data and high computing power

Cloud computing

- Software as a service (SaaS)
- Platform as a service(PaaS)
- Infrastructure as a service (IaaS)

Distributed System Categories

1. Distributed Data Stores
2. Distributed Computing
3. Distributed File Systems
4. Distributed Messaging
5. Distributed Applications
6. Distributed Ledgers



CAP Theorem

CAP Theorem is a concept that a distributed database system can only have 2 of the 3: Consistency, Availability and Partition Tolerance

Distributed Data Stores

- MongoDB
- Cassandra
- Hbase
- CouchDB

Distributed Computing

- Key to the influx of Big Data processing
- It is the technique of splitting an enormous task (e.g aggregate 100 billion records), of which no single computer is capable of practically executing on its own, into many smaller tasks, each of which can fit into a single commodity machine.
- You split your huge task into many smaller ones, have them execute on many machines in parallel, aggregate the data appropriately, and you have solved your initial problem.
- MapReduce
- Apache Hadoop
- Apache Spark
- Apache Kafka
- Apache Hive

Distributed File Systems

- HDFS (Hadoop Distributed File System)
- GFS (Google File System)
- IPFS (Interplanetary File System)

Distributed Messaging

- RabbitMQ
- Apache Kafka
- Google Pub/Sub
- Apache ActiveMQ
- Amazon SQS

Distributed Applications

- Parallel computation:
  - Scientific computing, including cluster computing and grid computing and various volunteer computing projects
  - Distributed rendering in computer graphics
- Real-time process control:
  - Aircraft control systems
  - Industrial control systems
- Network applications:
  - World wide web and peer-to-peer networks
  - Massively multiplayer online games and virtual reality communities
- Telecommunication networks:
  - Telephone networks and cellular networks
  - Computer networks such as the Internet

Distributed Ledgers

- Blockchain
- Bitcoin
- Ethereum
- Decentralized Authentication using Sovrin / Civic etc

Advantages & Disadvantages

- Advantages
  - Reliability
  - Scalability
  - Fault Tolerance
  - Increased Performance
- Disadvantages
  - Difficult to detect failures
  - Performance bottlenecks
  - Redundancy
  - Inconsistency

Math Applications in Distributed Systems

- Graph databases
- Graph Compute Engines
- Grid Computing
- Randomization Techniques
- Probabilistic Techniques
- Markov-chain based reliability analysis for distributed systems

Any questions?