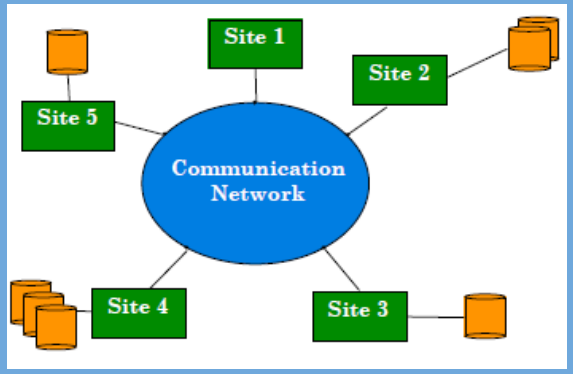
CS 432

Parallel & Distributed Computing



Week 01 – Lecture 1, 2 & 3

Spring 2025-MS

Dr. Shah Khalid

Email: shahkhalid@seecs.edu.pk

Lecture Outline

- Class Introduction
- Class policies details
- Course Details-Aims and Outcomes
- Distributed Computing- Basic Concepts
- Challenges in Distributed Computing

About me

- **Shah Khalid**
  - PhD – (Jiangsu University)

  Research Interest:
  - Information Retrieval, Data Science, Machine Learning, Federated Search, Recommender Systems, Sentiment analysis, Knowledge Graph, Human Action Recognition and Text Summarization
- Consultation Timing - By appointment
  - Faculty Block: A-205 (SEECS)
- Email
  - shah.khalid@seecs.edu.pk
- For Further Details: https://sites.google.com/view/shahkhalid

Some recently Published Research Articles

- **First Author ,** *Real-time feedback query expansion technique for supporting search using citation network analysis*, (*Journal of Information Science*, SCI Index)
- **First Author,** *Multi-objective approach for determining the usefulness of papers in Academic search*, ***Journal: Data Technologies and Applications,*** SCI Index.
- **First Author,** *An effective scholarly search by combining inverted indices and structured search with citation networks analysis*. *IEEE Access* 9 (2021)
- **First Author,** *Summarization of Scholarly Articles Using BERT and BiGRU: Deep Learning-Based Extractive Approach*. *Journal of King Saud University-Computer and Information Sciences* (2023).
- **First Author ,** *Sentiment and Context-Aware Hybrid DNN With Attention for Text Sentiment Classification*. *IEEE Access* 11 (2023).
- "*Human action recognition systems: A review of the trends and state-of-the-art.*" *IEEE Access (2024).*
- *"Depression Detection in Social Media: A Comprehensive Review of Machine Learning and Deep Learning Techniques." IEEE Access (2025).*

For more papers: Dr. Shah Khalid - Google Scholar

Students Introduction

- Brief Introduction of students
  - Name
  - Area of Interest
  - Day Scholar/ Hostelite

Course Books

Text Book:

- ***Distributed systems, principles and paradigms***, by Andrew S. Tanenbaum (Author), Maarten Van Steen (Author), **recent** Addition.
  - https://www.distributed-systems.net/index.php/books/ds4/
- ***Parallel and distributed simulation systems***, Richard Fujimoto
- ***Distributed system concepts and design***, George Coulouris, Jean Dollimore and Tim Kindberg.
- ***Distributed Systems, An algorithmic approach***, Sukumar Ghosh, Chapman & Hall/CRC Computer and Information Science Series, ISBN 10:1-58488-564-5

Reference Books:

- *Selected scientific papers.*
- *Web is the greenest reference book.*

Book Reading is essential for understanding of the lecture

Class Policy- Lecture Resources

LMS:

- Course Outline (Will explain structure of the course, assignments, and project details etc.)
- Lectures
- Assignments- task to do
- Submission as per guidelines

Qalam:

- Attendance (*Strict rules: missing two consecutive classes can result into warning— provide justification with solid proofs for missing out the session*)
- Grading

Course outline [½]

- Distributed/ parallel Computing
- Introduction to distributed systems, challenges
- Distributed system architectures
- Peer-to-Peer Systems
- Lamport logical clocks, vector clocks, event ordering
- Fault tolerance
- Distributed File System
- Solr Distributed indexing
- Introduction to Paxos
- Leader selection, Mutual Exclusion Algorithms
- Amazon Web Services- Cloud Computing
- Google File System

Course outline [2/2]

- Distributed Simulation
  - Programming Discrete event simulation fundamentals
  - Role of Look ahead in simulations
  - Synchronization Algorithms
  - Chandy/Misra/Bryantt Algorithm
  - Jafferson Algorithm
  - Samadi Algorithm for GVT calculation

  - Introduction to OMNeT++
  - Introduction to Message Passing Interface - MPI

Collaborative lectures

- Google cloud platform core infrastructure (Guest Talk)
- Introduction to solr- Distributed Indexing and Searching
- Introduction to Hadoop- Architecture, Big data

Tentative Marks Distribution

Course: 75% (Theory) and Lab + Semester Project 25%

- 25% Assignments and Quizzes
  - Late assignments will not be accepted / graded
  - zero tolerance policy towards plagiarism.
  - While collaboration in this course is highly encouraged, you must ensure that you do not claim other people's work/ ideas as your own.
  - **Quizzes**
    - Quizzes announced (so no retake-10%)
- 30% MSE
- 45 % ESE
- Lab 70% Project 30% (**Project Presentation in 2nd last week)**

Lectures Objectives

- [LO-1] understand distributed system and distributed protocols
- [LO-2] Point out possible flaws of an existing distributed systems
- [LO-3] Explain how existing distributed systems work
- [LO-4] Develop distributed applications/systems

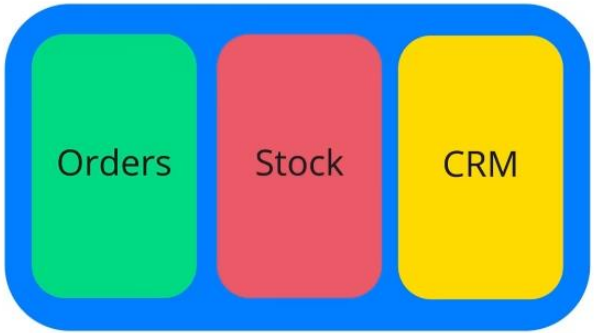
Introduction & Motivation

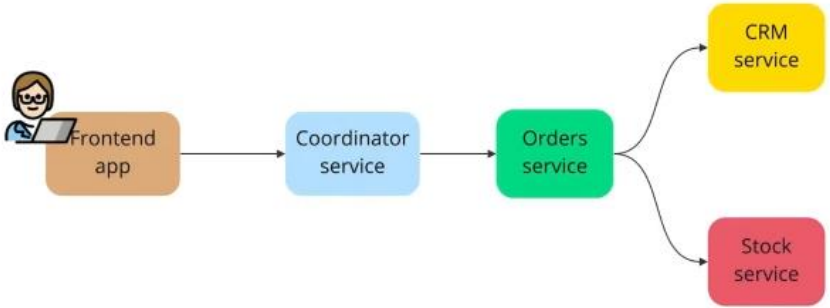
Define: Distributed Systems- Not a Centralized System

- Centralized system: State stored on a single computer
  - Simpler
  - Easier to understand
  - Can be faster for a single user
- Distributed system: State divided over multiple computers
  - More robust (can tolerate failures)
  - More scalable (often supports many users)
  - More complex

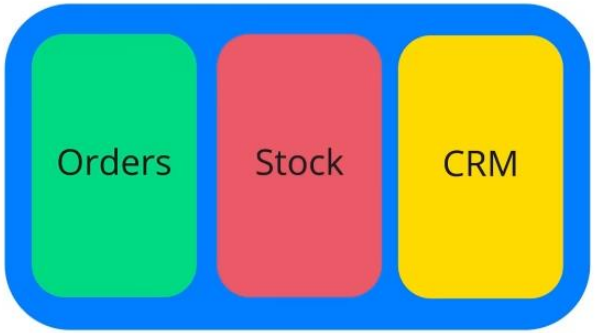
How can a complex system be more robust?

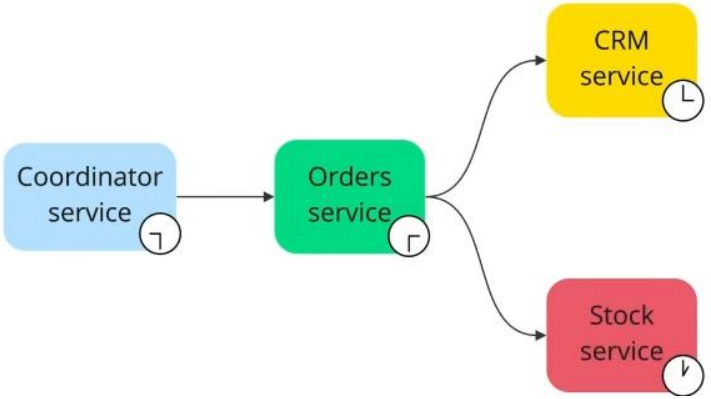
Example Scenario-Software to manage a shop





How Complex?



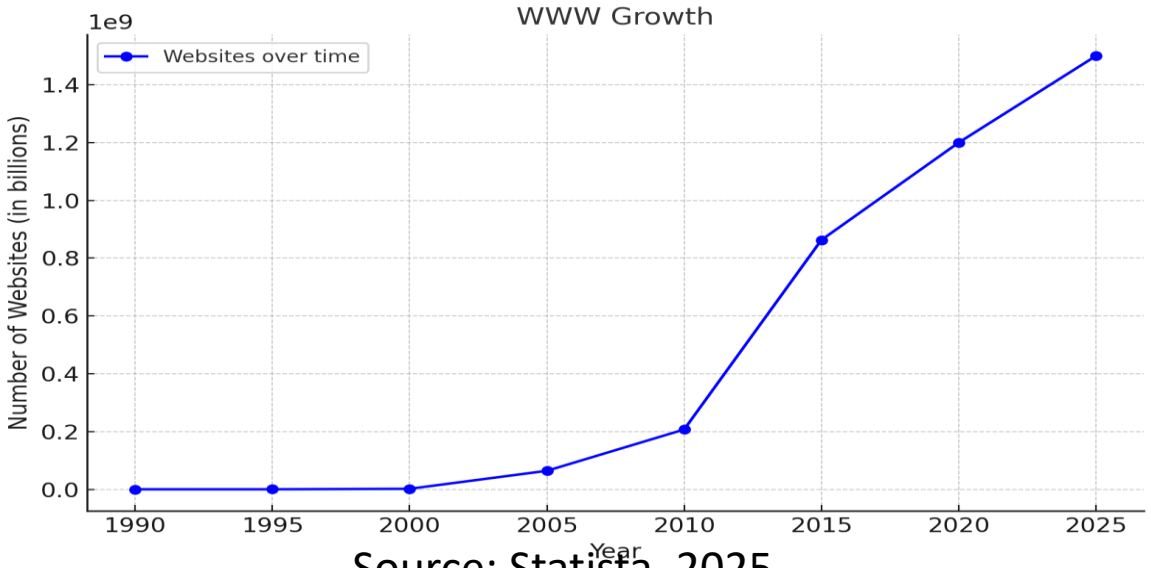


1. Partial Failure
2. Hard to code and test
3. Clock

Network

Why Distributed Computing?

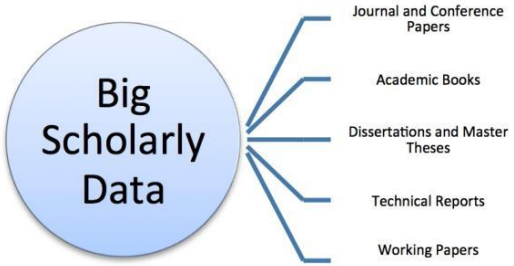
The world is expected to generate over 175 trillion gigabytes of data annually by 2025, equivalent to approximately 60 terabytes per person on Earth.

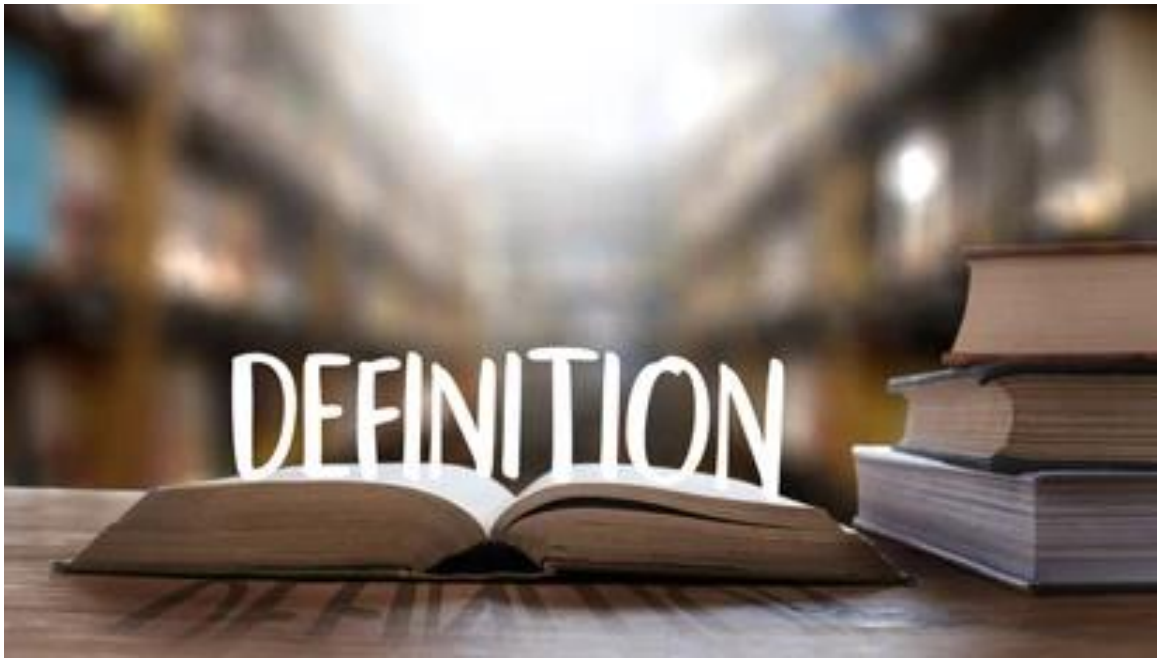


Source: Statista, 2025

Performance Needed For Big Problems

- Scholarly Big data -rapidly growing
  - contains information including millions of authors, papers, citations, figures, tables, as well as scholarly networks and digital libraries
- Facebook -rapidly growing
  - Every 60 seconds, 136,000 photos are uploaded, 510,000 comments are posted, and 293,000 status updates are posted. Facebook **generates 4 petabytes of data per day** — that's a million
- Many more data sources- How to manage?




DEFINITION

Define: Distributed Systems

A distributed system is: “A collection of independent computers that appears to its users as a single coherent system”

OR

“A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable” – Leslie Lamport

Today

- Unlimited computing power and storage space available to companies and users via the cloud
- Everyone has mobile phones which are stronger than average PCs 15 years ago.
- Ubiquitously available internet. . . Actually: Nothing goes without internet anymore
- Everywhere are networked devices such as terminals at train stations/airports/hospitals/banks, video surveillance
- Everything has computer chips in it as strong as computers 30 years ago

Questions about the Internet--www

Not current status

Q: How many computers are in the world?
A: Over 40 million.

Q: How many of them are Web servers?
A: Over 3 million.

Q: How many Web pages in the world?
A: Over 350 million.

Q: What is the most popular formats of Web documents?
A: HTML, GIF, JPG, ASCII files, Postscript and ASP.

Q: What is the average size of Web document?
A: Mean: 5 Kb; Median: 2 Kb.

Q: How many queries does a search engine answer every day?
A: Tens of millions.

Last lecture

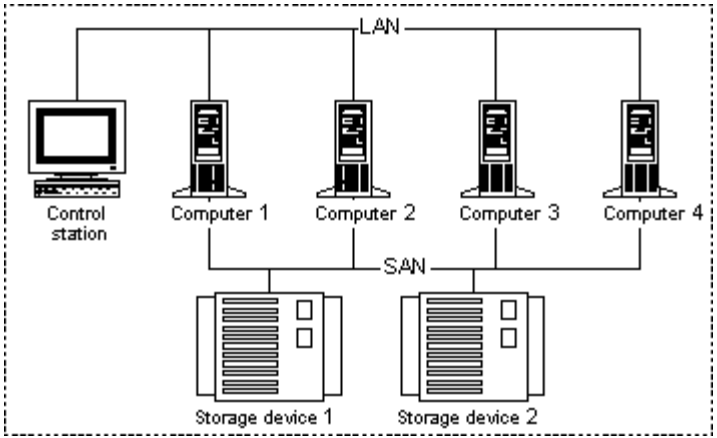
- Introduction to distributed computing and distributed Systems
- **Example Scenario- Today data and its computation**
- Hard to understand, hard to design and can fail in many complex ways

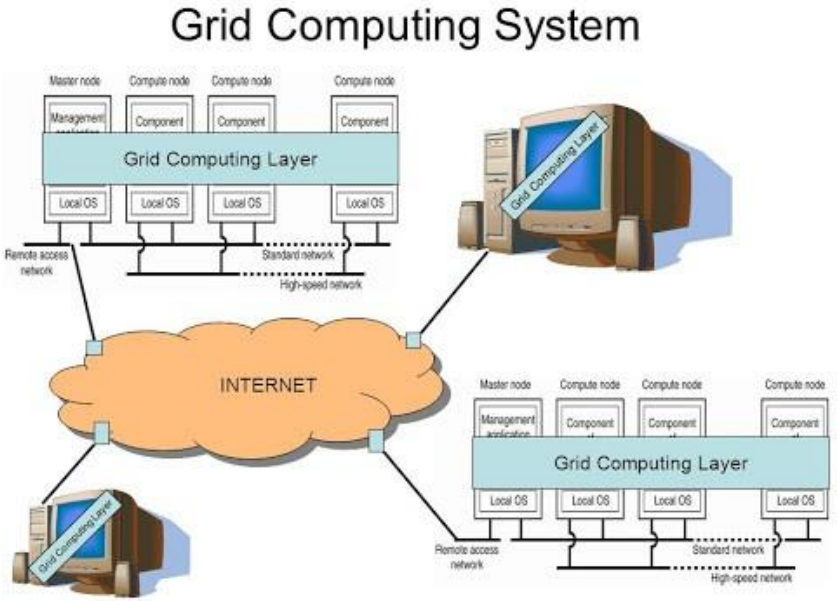
Today lecture

- Existing Distributed Computing architectures
- Different Challenges
  - – need to be studied to make the right trade-offs and pick the right solutions when building them

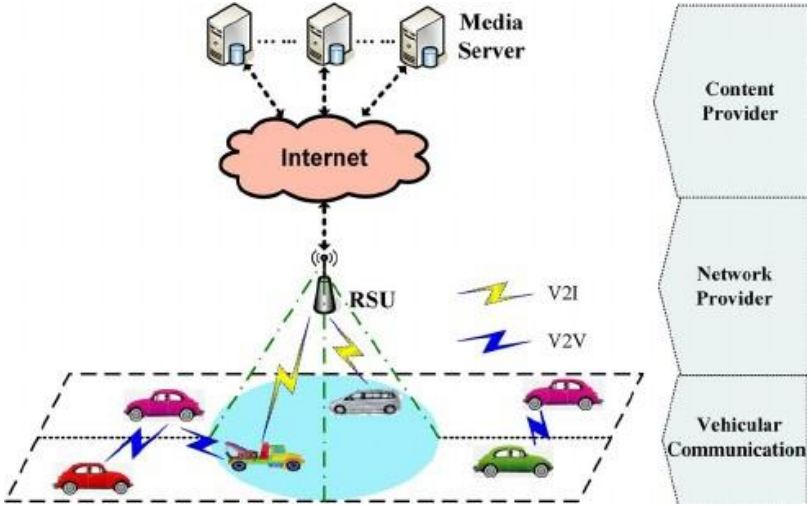
Distributed Systems!

Cluster Computing



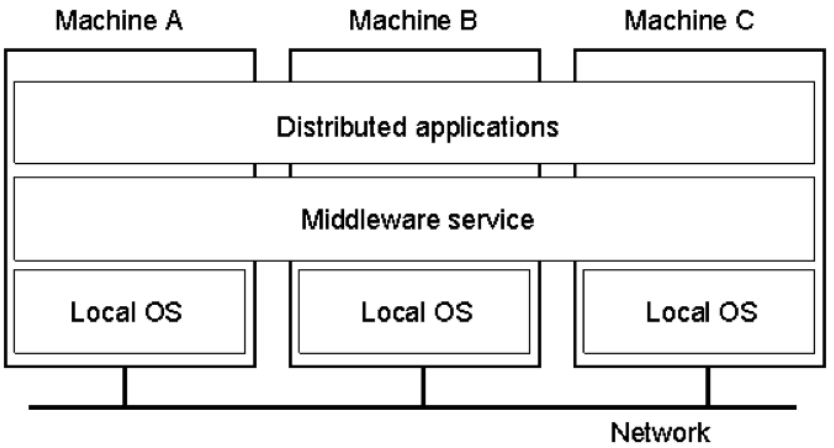


Distributed Systems!

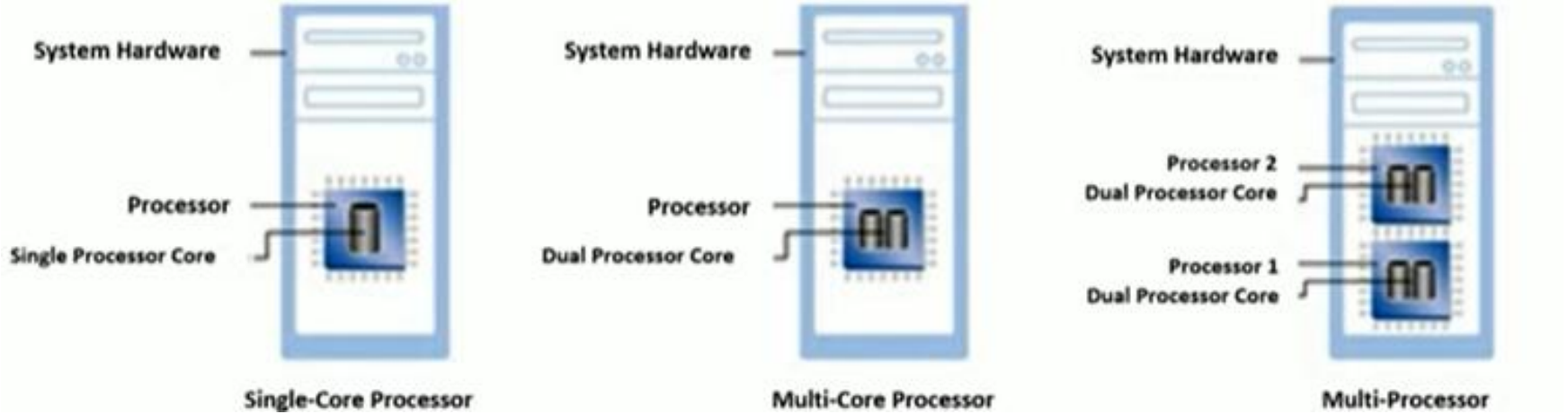


Distributed OPERATING System

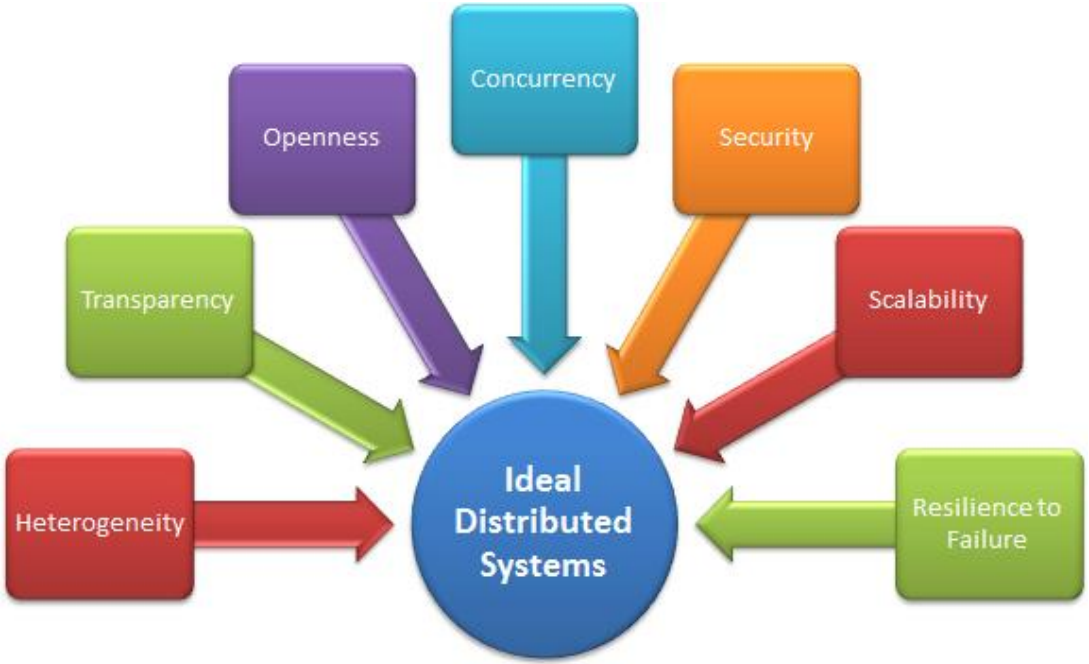
- To support heterogeneous computers and network to build a single view
- Distributed systems organized by means of a layer
- Placed between users app and OS
- Such distributed systems are called **Middleware**
  - ✓ A common layer across machines that facilitates interaction and integration.



Multi-processor



Challenges of Distributed Systems



Goals of Distributed Systems

- Four important goals to meet to build a distributed system
  - Make resource available
  - Distribution transparency
  - Openness
  - Scalability
- **Pitfalls**

1. Make resources Available

- Main goal is making it easier for the user to access/share remote resources
- resource can be anything
  - printer, computers, storage facilities, network etc.

There can be many reasons for resources sharing, any one?

2. Transparency

Definition of transparency is “Hide the fact that its processes and resources are physically distributed”

There are different kinds of transparency exist in distributed system

Can you suggest any?

2. Transparency- cont..

- Access Transparency: Client should be unaware of the distribution of the files, and how these files can be accessed - differences in machine architectures
- Location Transparency: Client should be unaware of the physical location of resources

https://scholar.google.com.pk/citations?user=Sff9RyoAAAAJ&hl=en

2. Transparency- cont..

- Migration Transparency: In distributed systems in which resources can be moved without effecting how these resources can be accessed

- Relocation Transparency: In distributed systems in which resources can be relocated while they are being accessed without user noticing anything

2. Transparency- cont..

- Replication Transparency: Resources are replicated to increase availability and performance

  Replication is hiding the fact that several copies of a resource exist
- Concurrency Transparency: Users and applications should be able to access shared resources without interference between each other

  lead to a consistency issues

2. Transparency- cont..

- Failure Transparency: The distributed system are prone to failures
  Failure transparency is user does not notice that the resource fails to work and that the system subsequently recover from the failure

Transparency	Description
Access	Hide differences in data distribution and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource

Degree of Transparency

- Hide distribution aspects is not a good idea?

How ?

Degree of Transparency- preferable, but it is not always the best option

A trade-off between a high level of transparency and a system's performance is required, For example

- not a good idea to keep a physical resource like a printer hidden from its users
  - Better to print job to a busy nearby computer instead of ideal one at corporate headquarters in a different country
- Communication among processes - Network delay
- Internet applications repeatedly try to contact server before trying another and finally giving up
- Replicas located on different continents, need to consistent, change in one requires seconds to update all

3. Openness

Another important goal of distributed systems

- An open distributed system is a system that offers services according to standard rules

  Interoperability - Two implementation of a system from different manufacturers can work together

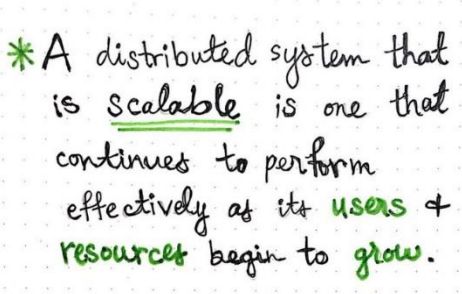
  Portability- Application developed by distributed system A can be executed without modification on system B

4. Scalability

Measured along at least three different dimensions

- Size scalability
- Geographically scalability
- Administratively scalability

Scalable DS=??



- Best scalability: when the **workload and computing resources** are increased or lowered by a factor of K at the same time while the average response time of the system or application remains unchanged

4. Scalability- Cont..

The server becomes a bottleneck as the number of users grows

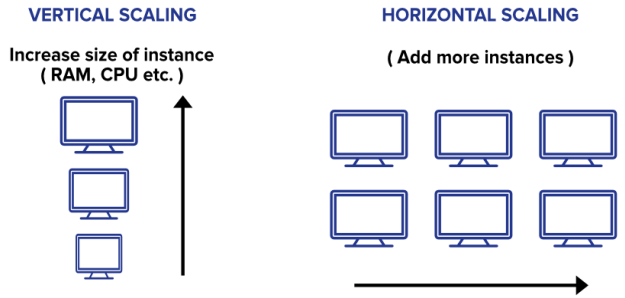
using only a single server is sometimes unavoidable

- centralized services
- centralized data
- centralized algorithms

4. Scalability- Cont..

1. Size scalability

- whenever users and resources grow
  - it should not be carried out at the cost of performance and efficiency of the system.
- The system must respond to the user in the same manner as it was responding before scaling the system.



4. Scalability- Cont..

2. Geographical scalability

- What happens when we increase the distance across the system?
  - Distance is defined as the physical spaces between nodes or users and resources
    - should not affect the communication time between the nodes.

4. Scalability- Cont..

3. Administrative Scalability

- scalability among different administrative domains
- they may have different policies
  - Resource usage
  - Management
  - Payment management and
  - Security

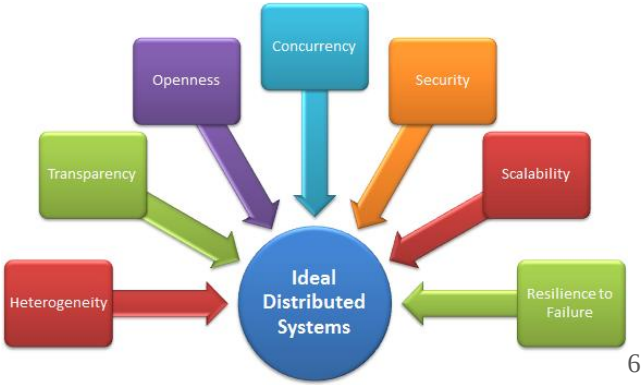
Comprehensive Definition of a Distributed System

A distributed system is a group of loosely-coupled nodes in order to:

- work together in an interoperable and concurrent way;
- communicate and coordinate their actions by passing messages to one another;
- tolerate faults without affecting its operations;
- achieve common goals;
- maximise resource utilisation and performance;
- appear as a single cohesive system.

Last Session

- Distributed / Parallel/ Computing
  - Example Scenario
- Why distributed Computing?
- Challenges Involved
- Four important goals to meet to build a distributed system
  - Make resource available
  - Distribution transparency
  - Openness
  - Scalability



Today Lecture

- Pitfalls- false assumption
- Types of Distributed System
- Distributed Systems Architecture
- Different Architectural Styles
  - ❑ Centralized Architecture
  - ❑ Decentralized Architecture
  - ❑ Hybrid Architecture

5. Pitfalls- Fallacies of distributed systems

Peter Deutsch (Sun microsystem) formulated- these are false assumption people makes about DS

- ❏ network is reliable
- ❏ network is secure
- ❏ network is homogenous
- ❏ latency is zero
- ❏ bandwidth is infinite
- ❏ transport cost is zero
- ❏ there is one administrator

1. Network is reliable

- You cannot assume the **network is reliable** and not worry about network issues.
- The truth is that networks are more reliable than they used to be. However…
- Noy 100% reliable.

When designing and writing your applications, don’t forget to account for network failures.

2. Latency is zero

- Imagine two applications on the same computer talking to each other. The latency, in this case, will be close to zero, but it won't be zero.
- If we introduce a network between the applications, the latency will always be greater than zero.

Latency is an important metric you should be aware of, and monitor for your applications. Latency can have a big impact on user experience and performance.

3. Infinite bandwidth

- At first, it might seem like there's plenty of bandwidth.
- However, when a system has tens or hundreds of services, the amount of communication and data sent back and forth increases significantly.
  - For example, it's predicted that self-driving cars will produce from 400 GB to 5 TB of data an hour.

Design your applications with bandwidth usage in mind.

4. Network is secure

- This fallacy can be fatal.
- Security and embracing a defense-in-depth approach must be a priority when designing your applications.

It’s not a question **if** your system will be attacked; it’s a question of **when** it will be attacked.

5. Topology doesn’t change

- Indeed, topology doesn’t change when you’re running applications on your computer. But…
  - when you deploy the applications to the cloud, the network topology is out of your control.

- The cloud provider upgrades and changes the network equipment, machines are turned off and new ones are created, and so on.

You can’t rely on constant topology in the cloud.

6. There is one administrator

- In the past, it was common to have a single person responsible for maintaining environments, installing and upgrading applications, and so on. However..
  - that approach has changed with the shift to modern cloud architectures and DevOps practices.
- Modern cloud-native applications are composed of many services, working together but developed by different teams. It's practically impossible for a single person to know and understand the whole application, let alone try to fix all the issues.

Put governance in place that makes it easy to troubleshoot any issues that arise.

7. Transport cost is zero & 8. Network is Homogeneous

- Networks are not homogeneous or of the same kind.
- Instead, networks are heterogeneous.
  - You can’t assume that the network hardware always stays the same.

The key point is to focus on standard protocols so that components can communicate, regardless of the hardware.

Types of Distributed Systems

Various types of distributed systems

- Distributed Computing System
- Distributed Information System
- Distributed Embedded System- Pervasive/ubiquitous

1. Distributed Computing System

- Used in performance computation which requires high computing
  - Grid Computing
  - Cluster Computing

1. Distributed Computing System - Grid System

“A Grid computing System is a collection of distributed computing available over a local or wide area network, that appears to an end user or application as one large virtual computing system”

It is an approach that spans not only location but also organizations, and machine architectures.

Internet – getting computers to talk together

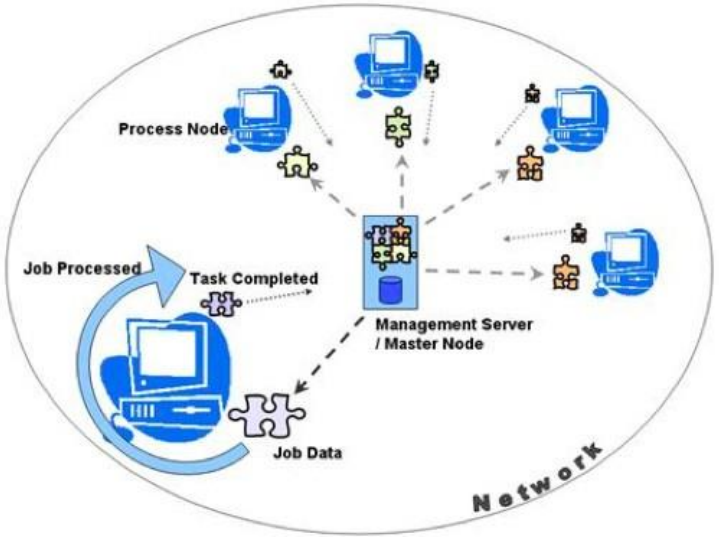
Grid Computing – getting computers work together

1. Distributed Computing System - Grid System

- collection of computers running the same operating system OR as complex as systems comprised of different OS
- server, which handles all the administrative duties for the system (control node, dispatcher)
- Nodes running special grid computing network software - middleware
- Grid middleware: to run a process or application across the entire network of machines.
- Middleware is the workhorse of the grid computing system

1. Distributed Computing System - Grid System

- Control node – dispatcher
  - Scheduling/priority task
  - Monitor systems
  - Resource allocation
- Grid middleware
  - Process launch
  - communicate



For Students

Applications of Grid Computing-

Find its role in the following:

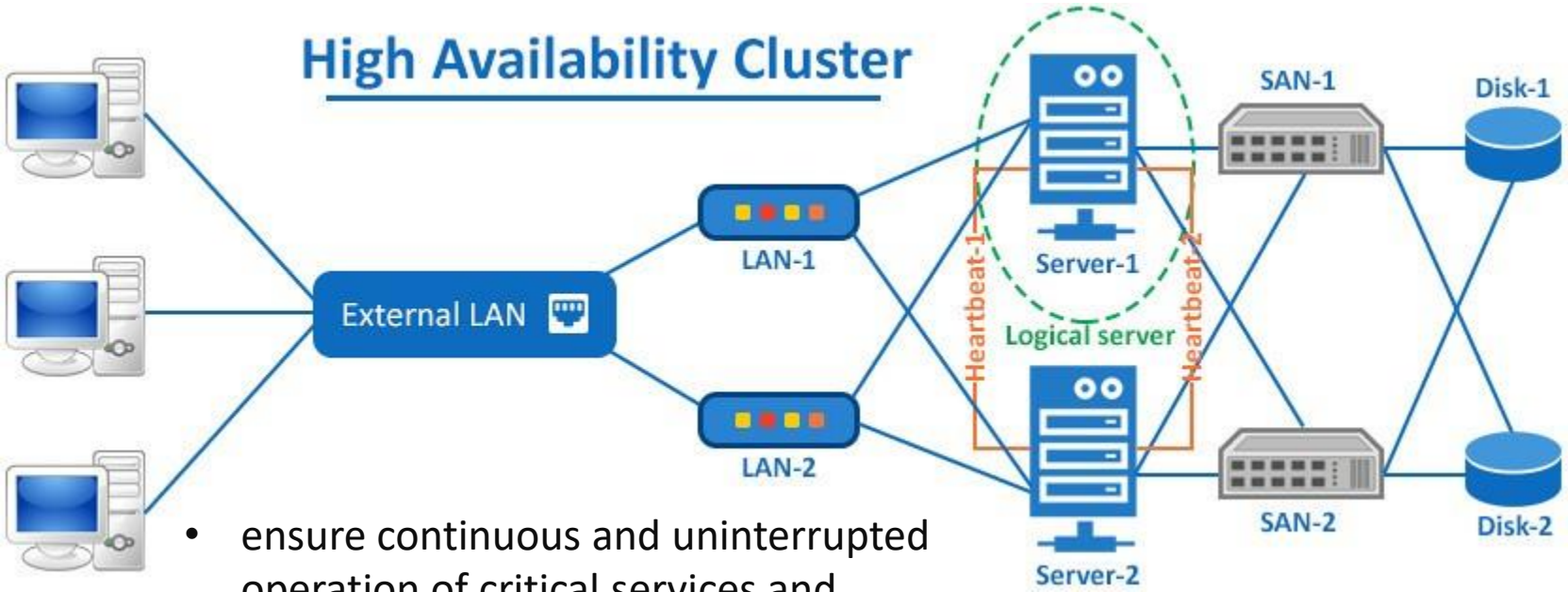
Genetics Researches, Cancer Research, Financial Analysis, Earthquake simulations and analysis, ecommerce back-office data processing task, motion-picture animation, weather, climate modeling, oil exploration researches

Large Hadron Collider (LHC) at cern

Current Status

1. Distributed Computing System - Cluster Computing

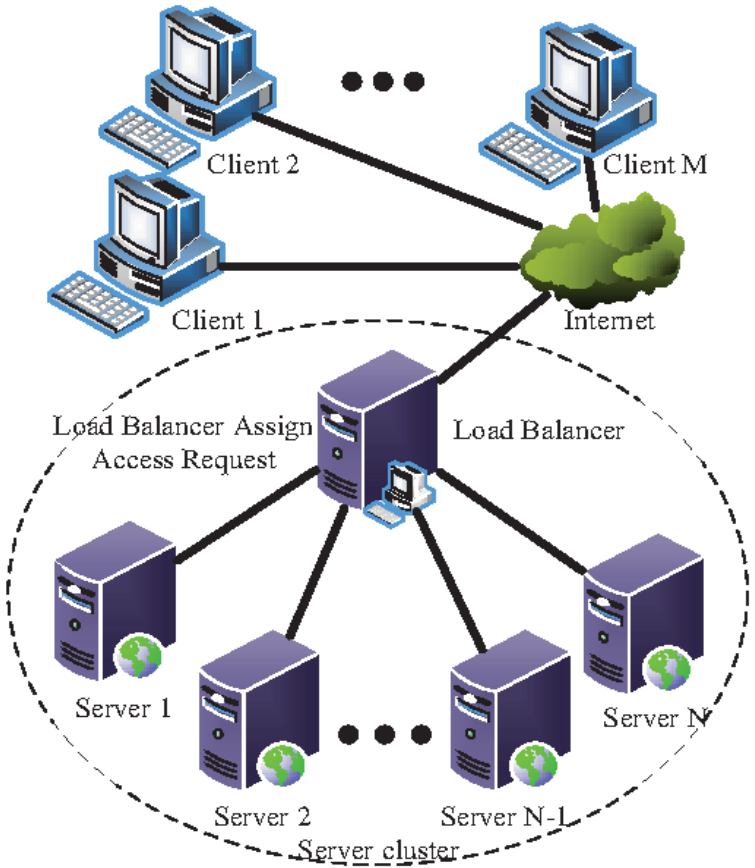
- Collection of systems that work together, can be viewed as a single computer
- Underlying hardware consists of collection of similar PCs
- Connected with high speed networks
- Each node run the same OS
- Definition of cluster is extend further
  - HA (High Availability Cluster)
  - LB (Load-balancing Cluster)



- ensure continuous and uninterrupted operation of critical services and applications.

Load Balancer

- aim to distribute incoming network traffic or computational workload across multiple nodes to optimize resource utilization and enhance performance.



1. Distributed Computing System - Cluster Computing

Why Cluster Computing?

- Performing a complex task
- Fault tolerance
- Processing speed
- Load balancing

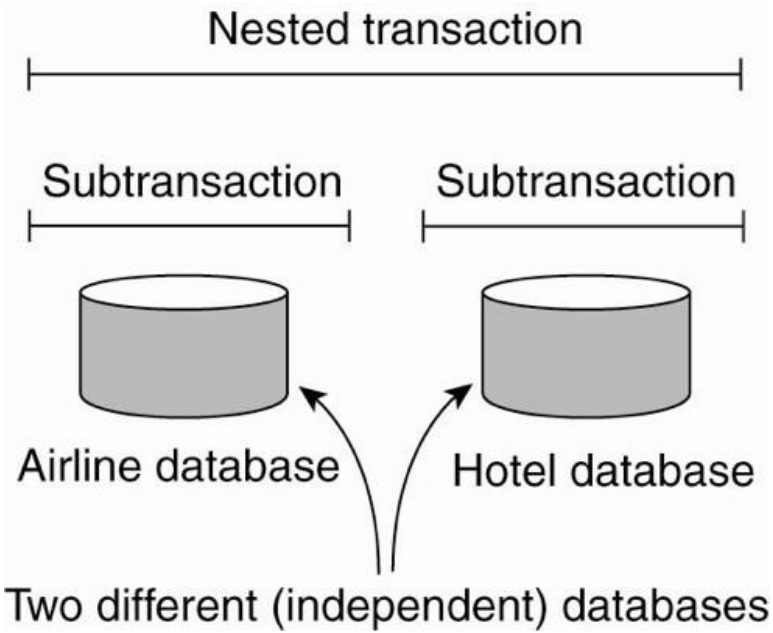
2. Distributed Information Systems

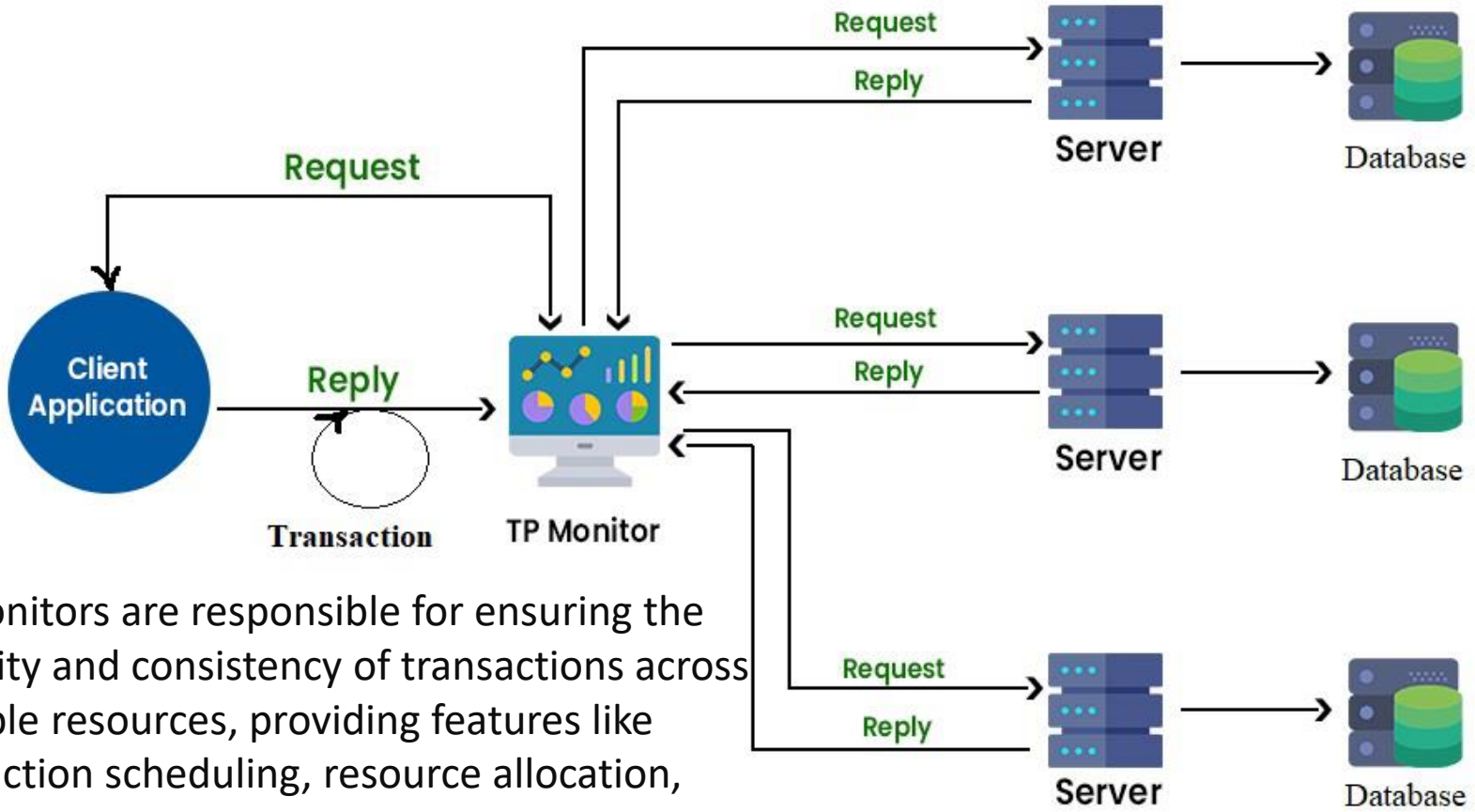
- Typical system includes a database
- Integration of such system is quite difficult
  - Combining different systems into one working solution is challenging because these systems might be built using different technologies.
- Client can wrap number of requests into single request and execute it as a distributed transaction
  - Example: When you purchase an item online, the system:Checks inventory.
    Deducts payment.
    Updates shipping status.
    All these actions are part of one transaction. If one step fails, the entire transaction is rolled back.
- Interoperability is a painful process

Cont..

Transaction processing system

Banking systems use TPS for processing customer transactions, including withdrawals, deposits, and fund transfers, across multiple branches.

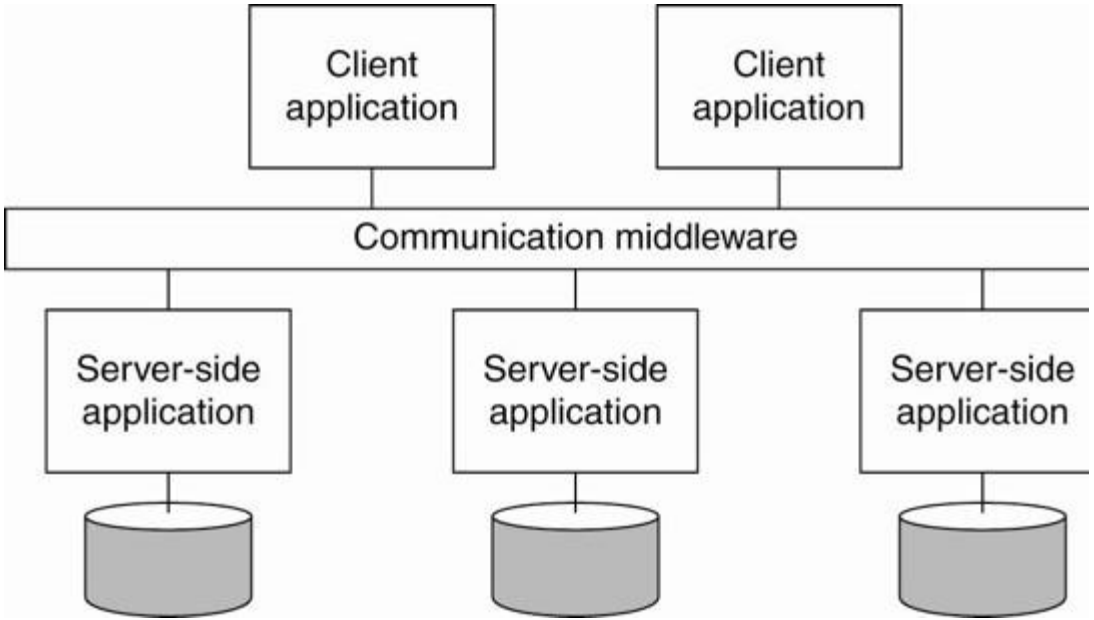




TP Monitors are responsible for ensuring the integrity and consistency of transactions across multiple resources, providing features like transaction scheduling, resource allocation, and error recovery.

Cont..

- Enterprise Application Integration
  - integration of systems and applications across an enterprise
  - process of linking such applications within a single organization together in order to simplify and automate business processes to the greatest extent possible



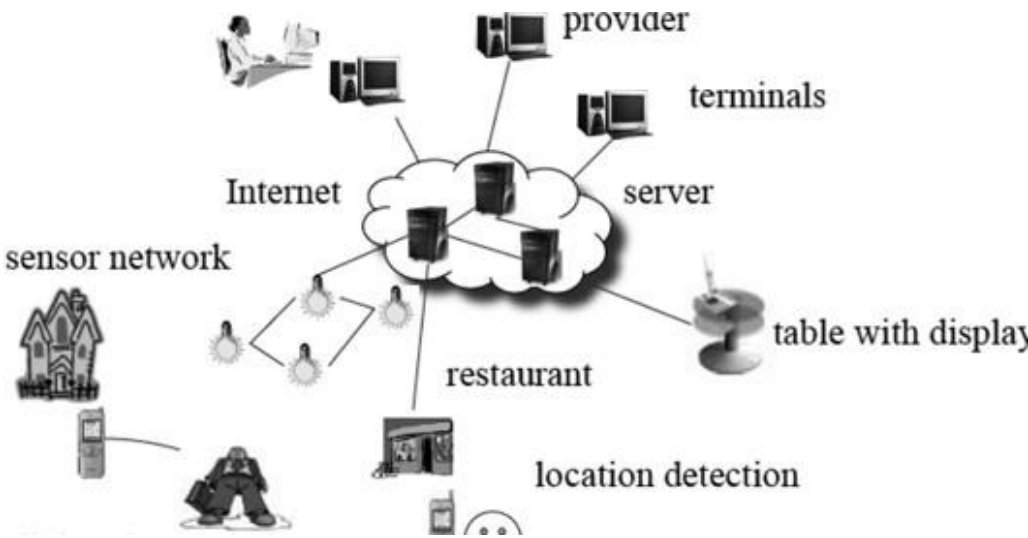
Consider a retail organization that manages its sales and inventory using separate software applications. The sales system handles customer orders, transactions, and invoicing, while the inventory system manages product stock levels, supply chain information, and restocking.

3. Distributed Pervasive Computing

- So far, we considered **stable distributed systems** (fixed nodes good connections)
- But this is not the case for the emerging next generation of distributed systems in which mobile and embedded devices are used
- Some requirements
  - Computing anywhere and anytime
  - **Contextual change:** environment changes should be immediately accounted for.
  - **Ad hoc composition:** Each node may be used in a very different ways by different users. *Requires ease-of-configuration*.
  - **Sharing is the default:** Nodes come and go, providing sharable services and information. *Calls again for simplicity*.

3. Distributed Pervasive Computing

- Emerging trend-Mobile and embedded computing devices
- Embedding microprocessors in a day-to-day objects
- Growing trend of embedding computational capability
- Instability is the default behavior
  - Being small, battery operated having wireless connections
- Lack of administrative control

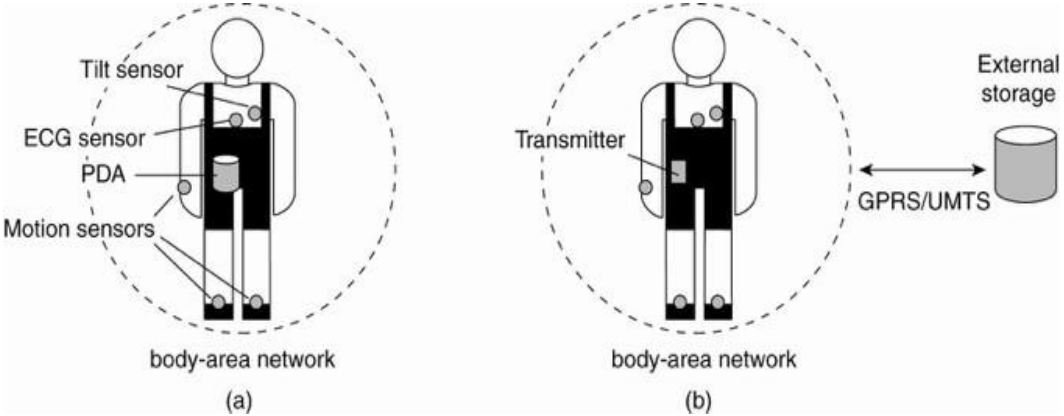


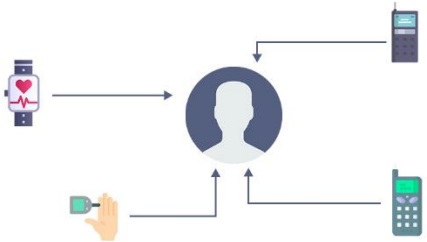
3. Distributed Pervasive Computing

- Distributed Home Systems
  - Popular type of pervasive system
  - Comprises of TV, audio, video equipment, game devices, PDA's as a single system
- Challenges
  - Self-configuring, self-managing
  - Achieved through UPnP standards – obtain IP address

3. Distributed Pervasive Computing

- Electronic Health Care Systems
- System equipped with sensors organized in BAN






Thank
you