REPUBLIC OF CAMEROON

Peace- Work-Fatherland

REPUBLIQUE DU CAMEROUN

Paix-Travail-Patrie

THE UNIVERSITY OF BAMENDA

UNIVERSITE DE BAMENDA

THE COLLEGE OF TECHNOLOGY

(COLTECH)

ECOLE DE TECHNOLOGIE

COMPUTER ENGINEERING

Software Engineering, Computer Network and System Maintenance

OPERATION SYSTEMS (CA):

DISTRIBUTED SYSTEMS

GROUP 06

Presented By:

Name	Matricule	Option	Attendance (3 Times)	Participation (On 5)
MBAH LESKY TAGWANG (GL)	UBA21PB015	SWE	3/3	5/5
CHRISTAIN SAFEH	UBA21PH048	CNSM	3/3	5/5
WANJAH ACHU HOPE	UBA21PB035	SWE	3/3	5/5
AJONGAKO BELTUS	UBA21P0478	CNSM	3/3	4/5
CHEMUTAH DENZEL NUMFOR A	UBA21PB012	SWE	3/3	4/5
CHE CRAIG ABONGBAN	UBA21PH009	CNSM	3/3	4/5

2021/2022

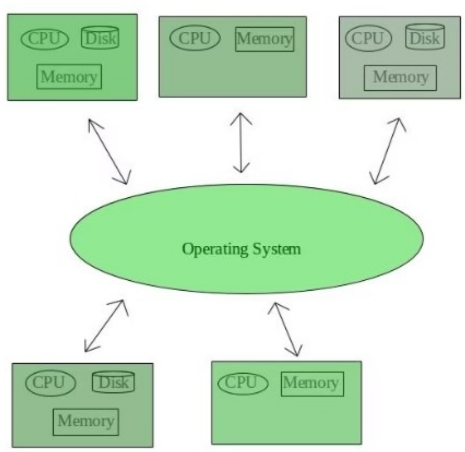
DISTRIBUTED SYSTEMS

Introduction

A distributed system is a collection of various independent components located on different computers or computing devices. These machines split up the work, coordinating their efforts to complete a job more efficiently than if a single device had been responsible for the task.

This system looks like an ordinary centralized operating system but runs on multiple, independent central processing units. These systems are also referred to as loosely **coupled systems** where each processor has its own local memory and processors communicate through various communication lines like high-speed buses or telephone lines (diagram below).

The distributed system involves a collection of autonomous computer systems, capable of communicating and cooperating with each other through a LAN/WAN. A distributed system, provides a virtual machine abstraction to its users and a wide sharing of resources like computational capacity, Input/output, storage and files. the users of this system won't know on which machine their programs are running and where their files are stored.



Functions/Characteristics of Distributed Systems

- **Resource Sharing:** hardware, software, files and data can be shared.
- **Openness:** how open is the software designed to be developed and shared with each other. This means that the services which the system provides are openly displayed through interface
- **Concurrency:** multiple machines can process the same function at the same time.
- **Scalability:** The ability to grow as the size of the workload increases and processing capabilities multiply is an essential feature of distributed systems, accomplished by adding additional processing units or nodes (computers) to the network as needed.
- **Fault Tolerance/Availability:** If one node fails, the remaining nodes can continue to operate without disrupting the overall computation.
- **Transparency:** An external programmer or end user sees a distributed system as a single computational unit rather than as its underlying parts, allowing users to interact with a single logical device rather than being concerned with the system's architecture.
- **Heterogeneity:** In most distributed systems, the nodes and components are often asynchronous, with different hardware, middleware, software and operating systems. This allows the distributed systems to be extended with the addition of new components

Applications/Examples of Distributed Systems

- **Distributed Databases Systems:** distributed database is a database that is located over multiple servers and/or physical locations. The data can either be replicated or duplicated across systems. Most popular applications use a distributed database and need to be aware of the homogenous or heterogenous nature of the distributed database system. A homogenous distributed database means that each system has the same database management system and data model. They are easier to manage and scale performance by adding new nodes and locations. Heterogenous distributed databases allow for multiple data models, different database management systems. Gateways are used to translate the data between nodes and usually happen as a result of merging applications and systems.

- **Distributed Real-Time Systems:** Many industries use real-time systems that are distributed locally and globally such as;
  - Airlines use flight control systems
  - Hotel reservation systems
  - Multiuser video conferencing systems
  - Uber and Lyft use dispatch systems
  - Cryptocurrency processing systems
  - manufacturing plants use automation control systems
  - Graphical and Video-rendering Systems
  - Multiplayer video games
  - logistics and e-commerce companies
    use real-time tracking systems.
- **Peer-To-Peer Networks Systems:** Peer-to-peer networks have evolved and e-mail and then the Internet as we know it continue to be the biggest, ever growing example of distributed systems. As the internet changed from IPv4 to IPv6, distributed systems have evolved from “LAN” based to “Internet” based
- **Telecommunication Networks:** have been around for over a century and it started as an early example of a peer to peer network. Cellular networks are distributed systems. As telephone networks have evolved to VOIP, it continues to grow in complexity as a distributed network.
- **Distributed Artificial Intelligence:** Distributed artificial intelligence is a way to use large scale computing power and parallel processing to learn and process very large data sets using multi-agents.
- **Graphical and Video-rendering Systems**

Examples of Distributed Operating Systems

- AIX operating for IBM RS/6000 computers
- Solaris operating system for SUN multiprocessor workstations.
- Mach/OS is a multitasking and multithreading UNIX compatible OS

- OSF/1 operating system.
- LOCUS
- MICROS
- IRIX operating system
- DYNIX operating system

Distributed System Architecture

- Distributed systems must have a network that connects all components (machines, hardware, or software) together so they can transfer messages to communicate with each other.
- That network could be connected with an IP address or use cables or even on a circuit board.
- The messages passed between machines contain forms of data that the systems want to share like databases, objects, and files.
- (What is a Distributed System? - An Introductory Guide, n.d.; What is a distributed Operating System?, n.d.; Distributed Operating System - javatpoint, n.d.; Tanenbaum; Serafini)
- The way the messages are communicated reliably whether it's sent, received, acknowledged or how a node retries on failure is an important feature of a distributed system.

Types of Distributed System Architectures

Distributed applications and processes typically use one of four architecture types below:

Client-Server

In the early days, distributed systems architecture consisted of a server as a shared resource like a printer, database, or a web server. It had multiple clients (for example, users behind computers) that decide when to use the shared resource, how to use and display it, change data, and send it back to the server.

In this system, the client requests the server for a resource. On the other hand, the server provides this resource to the client. One client contacts only a single server at a time. Whereas a single server can deal with multiple clients simultaneously. The clients and servers connect through a computer network in the system.

Today, distributed systems architecture has evolved with web applications into:

Three-Tier

In this architecture, the clients no longer need to be intelligent and can rely on a middle tier to do the processing and decision making. Most of the first web applications fall under this category. The middle tier could be called an agent that receives requests from clients, that could be stateless, processes the data and then forwards it on to the servers.

Multi-Tier

Enterprise web services first created n-tier or multi-tier systems architectures. This popularized the application servers that contain the business logic and interacts both with the data tiers and presentation tiers.

Peer-To-Peer

There are no centralized or special machine that does the heavy lifting and intelligent work in this architecture. All the decision making and responsibilities are split up amongst the machines involved and each could take on client or server roles. All the work equally divides among the nodes. Furthermore, these nodes can share data or resources as per the requirement. Again, they require a network to connect.

Advantages of Distributed Systems

- The ultimate goal of a distributed system is to enable the **scalability, performance** and **high availability** of applications.
  Major benefits include:
- **Unlimited Horizontal Scaling** - machines can be added whenever required.
- **Low Latency/Geo-distribution** - having machines that are geographically located closer to users, it will reduce the time it takes to serve users.
- **Fault Tolerance/Reliability** - if one server or data center goes down, others could still serve the users of the service.
- **Enhanced Speed:** Heavy traffic can slow down single servers when traffic gets heavy, impacting performance for everyone. The scalability of distributed databases and other distributed systems makes them easier to maintain and also sustain high-performance levels.

Disadvantages of Distributed Systems

- **Complexity** is the biggest disadvantage of distributed systems. There are more machines, more messages, more data being passed between more parties which leads to more opportunity there for failure. If a system is not carefully designed and a single node crashes, the entire system can go down. While distributed systems are designed to be fault tolerant, that fault tolerance isn't automatic or foolproof.
- **Increased complexity:** Distributed systems are more complex to design, manage and understand than traditional computing environments.
- **Data Integration & Consistency**: being able to synchronize the order of changes to data and states of the application in a distributed system is challenging, especially when their nodes are starting, stopping or failing.
- **Risk of network failure:** Distributed systems are beholden to public networks in order to transmit and receive data. If one segment of the internet becomes unavailable or overloaded, distributed system performance may decline**.** Messages may not be delivered to the right nodes or in the incorrect order which lead to a breakdown in communication and functionality.

- **Management Overhead:** more intelligence, monitoring, logging, load balancing functions need to be added for visibility into the operation and failures of the distributed systems
- **Synchronization process challenges:** Distributed systems work without a global clock, requiring careful programming to ensure that processes are properly synchronized to avoid transmission delays that result in errors and data corruption. In a complex system such as a multiplayer video game - synchronization can be challenging, especially on a public network that carries data traffic.
- **Imperfect scalability:** Doubling the number of nodes in a distributed system doesn't necessarily double performance. Architecting an effective distributed system that maximizes scalability is a complex undertaking that needs to consider load balancing, bandwidth management and other issues.
- **Security:** Distributed systems are as vulnerable to attack as any other system, but their distributed nature creates a much larger attack surface that exposes organizations to threats.

Why do we need distributed systems now?

Modern computing wouldn't be possible without distributed systems. They're essential to the operations of wireless networks, cloud computing services and the internet. If distributed systems didn't exist, neither would any of these technologies.

In most cases, we need distributed systems for enterprise-level jobs that don't have the complexity of an entire telecommunications network. Distributed systems provide scalability and improved performance in ways that some other systems can't, and because they can draw on the capabilities of other computing devices and processes, distributed systems can offer features that would be difficult or impossible to develop on a single system.

This includes things like performing an off-site server and application backup. If the master catalog doesn't see the segment bits it needs for a restore, it can ask the other off-site node or nodes to send the segments. Virtually everything you do now with a computing device takes advantage of the power of distributed systems, whether that's sending an email, playing a game or reading an article on the web.

Almost any type of application or service will incorporate some form of distributed computing. The need for always-on, available-anywhere computing is driving this trend, particularly as users increasingly turn to mobile devices for daily tasks. Looking ahead, distributed systems are certain to cement their importance in global computing as enterprise developers increasingly rely on distributed tools to streamline development, deploy systems and infrastructure, facilitate operations and manage applications.

References

- Andrew S. Tanenbaum, R. V. (1985). Distributed Operating Systems. In R. V. Andrew S. Tanenbaum, *Computer Surveys* (pp. 438-440).
- *Distributed Operating System - javatpoint*. (n.d.). Retrieved from JavaTpoint: https://www.javatpoint.com/distribbuted-operating-system
- *Distributed Operating System: Definition, Types and Characteristics*. (n.d.). Retrieved from toppr: https://www.toppr.com/guides/computer-science/computer-fundamentals/operating-system/distributed-operating-system/
- Serafini, M. (n.d.). Distributed and Operating Systems. 1-16, 22-29.
- Tanenbaum, A. S. (n.d.). Distributed Operating Systems. *Distributed Operating Systems*.
- Thakur, D. (n.d.). *Definition of Distributed Operating System - Computer Notes*. Retrieved from Computer Notes: https://ecomputernotes.com/fundamental/disk-operating-system/distributed-operating-system
- *Types of Operating Systems - GeeksforGeeks*. (n.d.). Retrieved from GEEKSFORGEEKS: https://www.google.com/amp/s/www.geeksforgeeks.org/types-of-operating-systems/amp/
- *What Are Distributed Systems? An Introduction | Splunk*. (n.d.). Retrieved from Splunk: https://www.splunk.com/en_us/data-insider/what-are-distributed-systems.html#:~:text=A%20distributed%20system%20is%20a,been%20responsible%20for%20the%20task
- *What is a distributed Operating System?* (n.d.). Retrieved from Tutorialspoint: https://www.tutorialspoint.com/what-is-a-distributed-operating-system
- *What is a Distributed System? - An Introductory Guide*. (n.d.). Retrieved from CONFLUENT: https://www.confluent.io/learn/distributed-systems/