Koneru Lakshmaiah Education Foundation
(Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Off-Campus: Bachupally-Gandimaisamma Road, Bowrampet, Hyderabad, Telangana - 500 043.
Phone No: 7815926816, www.klh.edu.in

Case Study ID: 15- Distributed Operating Systems

1. Title

Architecture and Design of Distributed Operating Systems

2. Introduction

- **Overview**
  This case study explores the architecture and principles of distributed operating systems, focusing on systems like Amoeba, Plan 9, and Hadoop. Distributed operating systems are designed to manage a collection of independent computers and make them appear as a single coherent system to the user.
- **Objective**
  The objective is to analyze the architecture, design principles, and functionality of distributed operating systems, understanding how they handle resource management, communication, and reliability across a distributed network.

3. Background

- **Organization/System /Description**
- Distributed operating systems (DOS) enable multiple independent computers to work together as one system. This case study examines specific examples, such as the Amoeba operating system, which provides a seamless distributed environment; Plan 9, which emphasizes resource sharing; and Hadoop, known for its distributed file system and data processing capabilities.
- **Current Network Setup**
  In traditional setups, each system operates independently, with limited sharing of resources. Distributed systems, on the other hand, integrate these resources under a single operating environment, enhancing performance and fault tolerance.

4. Problem Statement

- **Challenges Faced:**

  The primary challenges in distributed operating systems include:
- **Scalability:** Ensuring that the system can grow without performance degradation.
- **Fault Tolerance:** Managing failures without affecting the overall system.
- **Synchronization:** Coordinating processes across different nodes to maintain consistency.
- **Resource Management:** Efficiently allocating and managing resources across a distributed environment.

5. Proposed Solutions

- **Approach**
  The architecture of distributed systems focuses on decoupling the operating system from the underlying hardware, using a network of computers to perform tasks. Techniques such as process migration, remote procedure calls (RPC), and distributed file systems are employed.
- **Technologies/Protocols Used**
  - **Amoeba:** Microkernel architecture, capabilities for resource access control.
  - **Plan 9:** Namespace-based architecture, protocol 9P for resource access.
  - **Hadoop:** Hadoop Distributed File System (HDFS), MapReduce for distributed data processing.

6. Implementation

- **Process**
  - **Design Phase:** Understanding the unique requirements of distributed systems and choosing appropriate architectures like microkernels or monolithic kernels.
  - **Development Phase:** Implementing the chosen architecture using suitable programming languages and tools.
  - **Testing Phase:** Ensuring the system handles scalability, fault tolerance, and synchronization effectively.

  Implementation
  The implementation will be carried out in phases, starting with an assessment of the current system, followed by the deployment of the proposed solutions, and finally, a review and fine-tuning of the configurations.

- **Timeline**
  - **Week 1-2:** Research and Design
  - **Week 3-4:** Development
  - **Week 5-6:** Testing and Optimization

7. Results and Analysis

- **Outcomes**
  Amoeba: Demonstrated high scalability and fault tolerance but faced challenges in complexity.
  Plan 9: Effective resource sharing through namespaces, but limited adoption due to its unique model.
  Hadoop: Excellent performance in large-scale data processing, widely adopted in big data environments.

  Analysis
  Each system offers distinct advantages depending on the use case. Amoeba excels in flexibility, Plan 9 in resource sharing, and Hadoop in data processing. However, all systems require careful planning and management to address their inherent challenges.

8. Security Integration

- **Security Measures**

  Amoeba: Capability-based security model to control resource access.
  Plan 9: Simplified security through namespace control.
  Hadoop: Secure authentication with Kerberos and encrypted data transfer.

9. Conclusion

- **Summary**

  Distributed operating systems offer significant advantages in terms of scalability and fault tolerance, but they also present challenges in complexity, synchronization, and security. The case studies of Amoeba, Plan 9, and Hadoop illustrate various approaches to overcoming these challenges.

- **Recommendations**

  1. **For Future Development:** Emphasize security and ease of use to increase adoption.
  2.For Existing Systems: Continuous monitoring and optimization of resources

  processes to maintain performance.

10. References

- **Research Papers:**
  - **Amoeba: "Amoeba: A Distributed Operating System for the 1990s."**
  - **Plan 9: "The Design and Implementation of the Plan 9 Operating System."**
  - **Hadoop: "The Hadoop Distributed File System: Architecture and Design."**

NAME: G.LAVANYA

ID-NUMBER: 2320090004

SECTION-NO: 07

Koneru Lakshmaiah Education Foundation
(Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Off-Campus: Bachupally-Gandimaisamma Road, Bowrampet, Hyderabad, Telangana - 500 043.
Phone No: 7815926816, www.klh.edu.in