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

Title- AI-Enhanced Real-Time System Monitoring

Overview

- This case study explores the architecture and principles of **distributed operating systems (DOS)**, focusing on notable systems like **Amoeba**, **Plan 9**, and **Hadoop**. These systems are designed to manage multiple independent computers as a single coherent system, offering seamless resource sharing, fault tolerance, and scalability across distributed networks.
- **Objective**

The objective of this case study is to **analyze the architecture, design principles, and functionality**

of distributed operating systems. It delves into how these systems manage resources, handle inter-node communication, and ensure reliability across a network

3. Background

- **Organization/System Description: Distributed operating systems enable multiple computers to operate as one, allowing resource sharing and increased fault tolerance. The study focuses on three specific systems:**
- **Amoeba: A microkernel-based system providing seamless distributed environments.**
- **Plan 9: Emphasizes namespace-based resource sharing.**
- **Hadoop: Known for its Hadoop Distributed File System (HDFS) and MapReduce framework for large-scale data processing.**
- **Current Network Setup: Traditional network setups have limited resource sharing between individual computers. However, distributed systems integrate these resources into a single operating environment, enhancing both performance and fault tolerance.**

4. Problem Statement

- **Challenges Faced:**

  in distributed operating systems include:

- **Scalability:** Ensuring system performance is maintained as it grows..
- **Fault Tolerance:** The ability to recover from node failures without disrupting the system.
- **Synchronization:** Coordinating processes across multiple nodes for consistent operation.
- **Resource Management:** Efficient allocation and management of resources.

5. Proposed Solutions

- **Approach**

  The architecture of distributed systems decouples the operating system from hardware, allowing a network of computers to function as a unified system. Techniques like **process migration**, **remote procedure calls (RPC)**, and **distributed file systems** are commonly used.**Technologies/Protocols Used**

  - **Amoeba:** Uses microkernel architecture with capabilities for resource access control.
  - **Plan 9** :Implements a namespace-based architecture with the 9P protocol for resource access.
  - **Hadoop:** Employs HDFS and MapReduce for distributed data processing.

6. Implementation

- **Process**

  - **Design Phase** :Identifying system requirements and selecting appropriate architectures (e.g., microkernel or monolithic).
  - **Development Phase:** Building the system using suitable tools and languages.
  - **Testing Phase:** Evaluating scalability, fault tolerance, and synchronization.

    - **Implementation Phases:**
    - **Week 1-2**: Research and design.
    - **Week 3-4**: Development.
    - **Week 5-6**: Testing and optimization.

7. Results and Analysis

- **Outcomes**
  - **Amoeba**: Showed high scalability and fault tolerance, but was complex to manage.
  - **Plan 9**: Achieved effective resource sharing but had limited adoption due to its unique model.
  - **Hadoop**: Excelled in large-scale data processing and is widely adopted in big data environments.
- **Analysis**: Each system offers unique advantages:
  - **Amoeba**: Flexibility in distributed environments.
  - **Plan 9**: Resource sharing through namespaces.

- **Hadoop**: Superior performance in handling large datasets. However, each system also faces specific challenges that require careful management and planning.

8. Security Integration

- **Security Measures**
  - **Amoeba**: Uses a **capability-based** security model for controlling access to resources.
  - **Plan 9**: Implements simplified security through **namespace control**.
  - **Hadoop**: Ensures security with **Kerberos authentication** and encrypted data transfer.

9. Conclusion

- **Summary**: Distributed operating systems provide significant benefits in scalability and fault tolerance but pose challenges related to complexity, synchronization, and security. The case studies of **Amoeba**, **Plan 9**, and **Hadoop** demonstrate different approaches to overcoming these challenges.
- **Recommendations**:

  For Future Development: Focus on enhancing security and simplifying usage to increase adoption.

  For Existing Systems: Continuous monitoring and resource optimization are key to maintaining performance.

10. References

- **Research Papers:**

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

- This case study offers a comprehensive exploration of distributed operating systems, highlighting key challenges and potential solutions for achieving optimal performance across a network of independent computers.

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

NAME: G.LAVANYA

ID-NUMBER: 2320090004

SECTION-NO: 07

Group Members-

M.Rishika

K.Vedavyas

G.LAVANYA