Outline

- *Problems Unique to Distributed Systems*
- *Implementation Issues*
- *Issue Summary*
- *References*

Definition:

Distributed operating system: [3]

Integration of system services presenting a transparent view of a multiple computer system with distributed resources and control.

Consisting of concurrent processes accessing distributed shared or replicated resources through message passing in a network environment.

Problems Unique to Distributed Systems

- ## Distributed Operating Systems: [3]
  - *Generation: Third Generation Operating System.*
  - *Characteristics: Global view of file system, name space, time, security, computational power.*
  - *Goal: Single computer view of multiple computer system (transparency)*
- ## Distributed Operating System Goals:
  - *Efficiency*
  - *Consistency*
  - *Robustness*

Efficiency [3]

- *Efficiency problem: Communication delays*
  - *Data propagation*
  - *Overhead of communication protocols*
  - *Load distribution*

Consistency [3]

- *Consistency Problem:*
  - *User's perspective:*
    - *Uniformity in using the system*
    - *Predictability of the system's behavior*
  - *System's perspective:*
    - *Integrity maintenance*
      - *Concurrency control*
      - *Failure handling*
      - *Recovery procedures*

Robustness [3]

- *Robustness Problems:*
  - *Fault tolerance*
    - *What to do when a message is lost?*
    - *Handling of exceptional situations and errors*
    - *Changes in the system topology*
    - *Long message delays*
    - *Inability to locate a server*
  - *Security for the users and the system*

Implementation Issues

- *Objects models and identification.*
- *Distributed Coordination.*
- *Interprocess Communication*
- *Distributed Resources.*
- *Fault Tolerance and Security.*

Identification / Name

- **Design Issue Example:** Resource identification [2]

  The resources in a distributed system are spread across different computers and a naming scheme has to be devised so that users can discover and refer to the resources that they need.

  An example of such a naming scheme is the URL (Uniform Resource Locator) that is used to identify WWW pages. If a meaningful and universally understood identification scheme is not used then many of these resources will be inaccessible to system users.

Object Model and Naming Schemes [1]

- *Objects:*
  - *processes, files, memory, devices, processors, and networks.*
- *Object access:*
  - *Each object associate with a defined access operation.*
  - *Accesses via object servers*
- *Identification of a server by:*
  - *Name*
  - *Physical or Logical address*
  - *Service that the servers provide.*
- *Identification Issue:*
  - *Multiple server addresses may exist requiring a server to move requiring the name to be changed.*

Distributed Coordination [1]

- *Processes Coordination required to achieve synchronization:*

- *Synchronization Types:*
  - *Barrier synchronization:*
  - *Condition Coordination:*
  - *Mutual exclusion:*

Synchronization Types

- *Barrier Synchronization:*
  - *Process must reach a common synchronization point before they can continue:*

- *Condition Coordination:*
  - *A process must wait for a condition that will be set asynchronously by other interacting processes to maintain some ordering of execution.*

- *Mutual Exclusion:*
  - *Concurrent processes must have mutual exclusion when accessing a critical shared resource.*

Synchronization Issues:

- *State information sent by messages:*
  - *Typically only partial state information is known about other processes making synchronization difficult.*
  - *Information not current due to transfer time delay.*
- *Decision if process may continue must rely on a message resolution protocol.*
  - *Centralized Coordinator:*
    - *Central point of failure*
- *Deadlocks*
  - *Circular Waiting for the other process*
  - *Deadlock detection and recovery strategies.*

Interprocess Communication [1]

- *Lower level: message passing*
- *Higher level logical communication provides transparency*
  - *Client/server model communication.*
    - *All system communication are seen as a pair of message exchanges between the client and server.*
  - *Remote Procedure Call, (RPC), communication.*
    - *RPC built on top of client/server model.*
    - *Request/reply message passing as used in programming procedure-call concept.*

Interprocess Communication Issues

- *Susceptible to failures in the system due to having to communicate through several protocol layers.*

Distributed Resources [1]

- *Resources:*
  - *Data (Storage)*
  - *Processing capacity (Sum of all processors)*
- *Transparency of Data distribution:*
  - *Distributed file systems*
    - *Single file system view in distributed environment.*
  - *Distributed shared memory*
    - *Single shared memory view of physically distributed memories.*
  - *Issue: Sharing and replication of data/Memory.*
- *Transparency of process allocating:*
  - *Applications are constrained by time, thus scheduling of process must satisfy a real-time requirement.*
    - *Load Distribution Schemes*

Load Distribution Schemes

- *Static Load Distribution:*
  - *Multiprocessor scheduling*
  - *Objective: Minimize the completion time of processes*
  - *Issue: Minimize communication overhead with efficient scheduling.*
- *Dynamic Load Distribution:*
  - *Load sharing*
  - *Objective: Maximize the utilization of processors.*
  - *Issue: Process migration strategy & mechanism.*

Fault Tolerance and Security [1]

- *Failures & Security Threats:*
  - *Openness inherent in Distributed Environments*
- *System Failures:*
  - *Failures: Faults due to unintentional intrusions*
  - *Security Violations: Faults due to intentional intrusions.*
- *Issue: Fault Tolerance*
  - *Faults Transparent to user:*
    - *System Redundancy (Inherent property in Distributed Systems)*
    - *Systems Ability to Recovery. (Rolling back failed processes)*
- *Security Issue: Authentication & Authorization*
  - *Access control over across network with different administrative units & varying security models.*

Summary of Issues [3]

Issue	Affect Service
Communication, Synchronization, distributed algorithms	*Interaction and Control*
Process scheduling, deadlock handling, load balancing	*Performance*
Resource scheduling, file sharing, concurrency control	*Resource*
Failure handling, configuration, redundancy	*System Failures*

Issues Governing Quality of Service

The quality of service offered by a system reflects its performance, availability and reliability. It is affected by a number of factors such as the allocation of processes to processes in the system, the distribution of resources across the system, the network and the system hardware and the adaptability of the system.