

Introduction

Distributed Systems Challenges



Nabendu Chaki, University of Calcutta, India

1

Brief Outline



- Introduction
- Clock Model and State Recording – I
 - Global state and consistent state
 - Lamport's logical clock model
- Clock Model and State Recording – II
 - Distributed snapshot recording algorithm
- Concurrency Control in DS
 - Mutual exclusion
 - Deadlock handling
- Process Migration Approaches
- Termination Detection in Distributed Systems

2

Brief Outline



- Middleware and Remote Procedure Call
 - Introduction to RPC
 - Message format
- Implementing RPC
 - Call semantics
 - Call binding
 - Complex RPC
- RPC Case Study on SUN Systems
- Distributed File System using Hadoop
- Blockchain and Distributed Storage

3

What is a Process?



- “...each system and end-user task is contained within a **process**. The system creates new **processes** all the time, and **processes** die when a task finishes or something unexpected happens.”
 - Martin Streicher, IBM
- “...a program in execution”
 - Silberschatz, Galvin

4

What is Distributed System?



- It's a networking environment
 - Multiple nodes
 - Multiple processors
 - Multiple processes, threads

5

Where it began?



- Carnegie Mellon University
- Why?



Mahadev Satyanarayanan

6

What is Distributed System?

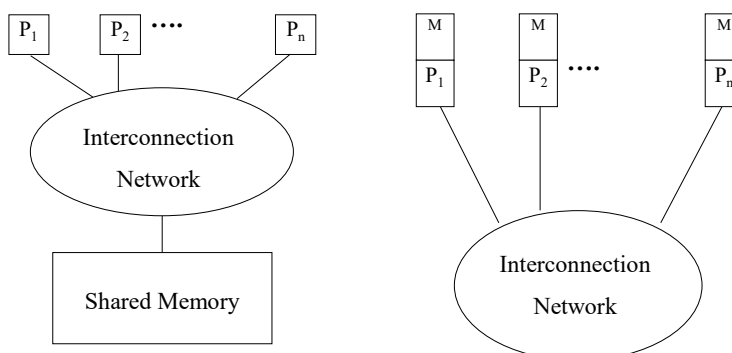


- It's Network ++ 😊
- What are these plus factors?
 - Loosely coupled architecture
 - The hardware perspective
 - The software perspective
 - Distribution transparency
 - User **need** not know resource location in the network
 - User **does** not know resource location in the network

Brendan Burns, Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services; O'Reilly; February 2018. ISBN-13: 978-1491983645.

7

Multi-Processing Organizations

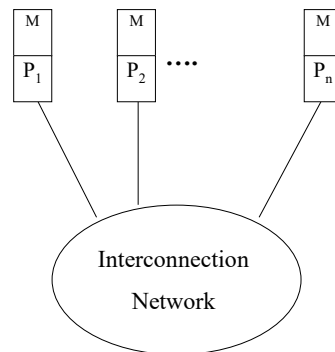


8

Distributed System and Coupling



- A distributed system is loosely coupled



9

Advantages of Loose Coupling



- Improved autonomy for components
- Improved availability
 - Graceful degradation of performance for increasing number of failures
 - Prevents catastrophic failure
- A distributed system is not a collection of isolated nodes
- Network integrates the components

10

Features



- Resource sharing
 - sharing and printing files at remote sites
 - processing information in a distributed database
 - using remote specialized hardware devices
- Computation speedup – load sharing
- Reliability: detect and recover from site failure, function transfer, reintegrate failed site
- Communication – message passing

11

11

Distribution Transparency



- Access transparency
- Location transparency
 - Name transparency
 - User mobility
- Replication transparency
- Failure transparency
- Migration transparency
- Concurrency transparency
- Performance transparency

Pradeep K. Sinha, Distributed Operating Systems - Concepts And Design, Chapter 1, pp. 19-23, IEEE Press, 1996

12

Is it All that Transparent?



- Syntactic transparency
 - User need not use a different syntax for remote services
- Semantic transparency
 - Factors
 - Response time for query
 - Restricted access to services

13

Taxi?



Sharing Enhances Utilization!

30 April 2024

N. Chaki, University of Calcutta, India

14

How sharing helps?



- Increased Efficiency
 - Higher Throughput
 - Improved Availability
 - Easier Maintenance
- Improved sharing of resources
 - Evolution of OS is for greater sharing

15

Sharing and Evolution of OS



- Monitors handling one job at a time
- Batch Monitoring System
- Multi-programming OS
- OS supporting Virtual Memory space
- Multi-tasking OS
- Networking OS
- OS for Distributed Systems

16



Thanks for your kind attention

Questions??