CSE 4117

Parallel and Distributed Processing

rmShoeb
CSE-16

Distributed Systems - Maarten van Steen and Andrew S. Tanenbaum, first edition

- Chapter - 1: Introduction
  - 1.1: Definition of Distributed System
  - 1.2: Goals
    - 1.2.1: Connecting Users and Resources
    - 1.2.2: Transparency
    - 1.2.3: Openness
    - 1.2.4: Scalability
  - 1.3: Hardware Concepts
    - 1.3.1: Multiprocessors
    - 1.3.2: Homogeneous Multicomputer Systems
    - 1.3.3: Heterogeneous Multicomputer Systems
  - 1.4: Software Concepts
    - 1.4.1: Distributed Operating Systems
      - Uniprocessor Operating Systems
      - Multiprocessor Operating Systems
      - Multicomputer Operating Systems
      - Distributed Shared Memory Systems
    - 1.4.2: Network Operating Systems
    - 1.4.3: Middleware
      - A Comparison between Systems
  - 1.5: The Client-Server Model
    - 1.5.1: Clients and Servers
    - 1.5.2: Application Layering
      - User-Interface Level
      - Processing Level
      - Data Level
- Chapter - 2: Communication
  - 2.2: Remote Procedure Call
    - 2.2.1: Basic RPC Operation
      - Conventional Procedure Call
      - Client and Server Stub
    - 2.2.2: Parameter Passing
      - Passing Value Parameters
      - Passing Reference Parameters
      - Parameter Specification and Stub Generation
    - 2.2.3: Extended RPC Models
      - Asynchronous RPC
- Chapter - 3: Processes
  - 3.4: Code Migration

        - 3.4.1: Approaches to Code Migration
            - Reasons for Migrating Code
            - Models for Code Migration
        - 3.4.2: Migration and Local Resources
- Chapter - 5: Synchronization
    - 5.1: Clock Synchronization
        - 5.1.1: Physical Clocks
        - 5.1.2: Clock Synchronization Algorithms
            - Cristian's Algorithm
            - The Berkeley Algorithm
    - 5.2: Logical Clocks
        - 5.2.1: Lamport Timestamps
    - 5.4: Election Algorithms
        - 5.4.1: The Bully Algorithm
        - 5.4.2: A Ring Algorithm
    - 5.5: Mutual Exclusion
        - 5.5.1: A Centralized Algorithm
        - 5.5.2: A Distributed Algorithm
        - 5.5.3: A Token Ring Algorithm
        - 5.5.4: A Comparison of the Three Algorithms

Distributed Operating System - Andrew S. Tanenbaum

- Chapter - 1: Introduction to Distributed Systems
    - 1.1: What is a Distributed System
    - 1.2: Goals
        - 1.2.1: Advantages of Distributed Systems over Centralized Systems
        - 1.2.2: Advantages of Distributed Systems over Independant PC's
        - 1.2.3: Disadvantages of Distributed Systems
    - 1.3: Hardware Concepts
        - 1.3.1: Bus-Based Multiprocessors
        - 1.3.2: Switched Multiprocessors
        - 1.3.3: Bus-Based Multicomputers
        - 1.3.4: Switched Multicomputers
- Chapter - 2: Communication in Distributed Systems
    - 2.4: Remote Procedure Call
        - 2.4.1: Basic RPC Operation
        - 2.4.2: Parameter Passing
        - 2.4.3: Dynamic Binding
        - 2.4.4: RPC Semantics in the Presence of Failures
        - 2.4.5: Implementation Issues
            - Critical Path

- Chapter - 3: Synchronizations in Distributed Systems
  - 3.1: Clock Synchronization
    - 3.1.1: Logical Clocks
    - 3.1.2: Physical Clocks
    - 3.1.3: Clock Synchronization Algorithms
      - Cristian's Algorithm
      - The Berkeley Algorithm
  - 3.2: Mutual Exclusion
    - 3.2.1: A Centralized Algorithm
    - 3.2.2: A Distributed Algorithm
    - 3.2.3: A Token Ring Algorithm
    - 3.2.4: A Comparison of the Three Algorithms
  - 3.3: Election Algorithms
    - 3.3.1: The Bully Algorithm
    - 3.3.2: A Ring Algorithm
  - 3.5: Deadlocks in Distributed Systems
    - 3.5.1: Distributed Deadlock Detection
      - Centralized Deadlock Detection
      - Distributed Deadlock Detection
    - 3.5.2: Distributed Deadlock Prevention
- Chapter - 4: Processes and Processors in Distributed Systems
  - 4.5: Fault Tolerance
    - 4.5.1: Component Faults
    - 4.5.2: System Failures
    - 4.5.3: Synchronous versus Asynchronous Systems
    - 4.5.4: Use of Redundancy
    - 4.5.5: Fault Tolerance Using Active Replication
    - 4.5.6: Fault Tolerance Using Primary Backup

Computer Organization and Architecture: Designing for Performance - William Stallings, tenth edition

- Chapter - 17: Parallel Processing
  - 17.1: Multiple Processor Organizations
  - 17.2: Symmetric Multiprocessors
  - 17.3: Cache Coherence and the MESI Protocol
  - 17.4: Multithreading and Chip Multiprocessors
  - 17.5: Clusters
  - 17.6: Nonuniform Memory Access
  - 17.7: Cloud Computing
- Chapter - 18: Multicore Computers
  - 18.1

    - 18.2
    - 18.3
    - 18.4
- Chapter - 19: General Purpose Graphics Processing Units
    - 19.1
    - 19.2
    - 19.3
    - 19.4
    - 19.5

Computer Architecture and Parallel Processing - Kai Hwang and Faye A. Briggs

- Chapter - 1: Introduction to Parallel Processing
    - 1.1: Evolution of Computer Systems
        - 1.1.2: Trends Towards Parallel Processing
    - 1.2: Parallelism in Uniprocessor Systems
        - 1.2.1: Basic Uniprocessor Architecture
        - 1.2.2: Parallel Processing Mechanisms
            - Multiplicity of Functional Units
            - Parallelism and Pipelining within the CPU
            - Overlapped CPU and I/O Operations
            - Use of Hierarchical Memory System
        - 1.2.3: Balancing of Subsystem Bandwidth
            - Bandwidth balancing between CPU and Memory
            - Bandwidth balancing between Memory and I/O devices
        - 1.2.4: Multiprogramming and Time Sharing
            - Multiprogramming
            - Time Sharing
    - 1.3: Parallel Computer Structures
        - 1.3.1: Pipeline Computers
        - 1.3.2: Array Computers
        - 1.3.3: Multiprocessor Systems
    - 1.4: Architectural Classification Schemes
        - 1.4.1: Multiplicity of Instruction-Data Streams
            - SISD Computer Organization
            - SIMD Computer Organization
            - MISD Computer Organization
            - MIMD Computer Organization
- Chapter - 7: Multiprocessor Architecture and Programming
    - 7.2: Interconnection Networks
        - 7.2.1: Time Shared or Common Buses
        - 7.2.3: Multistage Networks for Multiprocessors