Process Assignment, Scheduling and Load Balancing

Study-Ready Notes

Compiled by Andrew Photinakis

October 2nd, 2025

Contents

1 Introduction to Process Assignment and Scheduling

1.1 Basic Concepts

- **Process Assignment**: Mapping processes to processing elements (PEs)
  - Considers: Process characteristics, Hardware/Software characteristics
  - Answers the question: **Where?**
- **Scheduling**: Determining when to start executing each task
  - Types: Undirected, Directed
  - Answers the question: **When?**

[Summary] Process assignment determines where processes run, while scheduling determines when they execute. Both consider system and process characteristics for optimal performance.

1.2 Programming Models

- **Definition**: Abstractions of CPU hardware, memory, and communication architectures
- **Types**: SPMD (Single Program Multiple Data), MPMD (Multiple Program Multiple Data), Shared Memory, Message Passing
- **Selection Criteria**:
  - Application problem characteristics
  - Available parallel computer architecture
  - Knowledge of parallel algorithms and programming languages

[Summary] Programming models provide abstractions for parallel computing, with selection depending on application needs and available hardware.

2 Critical Factors in Parallel Computing

2.1 Granularity

- **Definition**: Ratio of computation to communication
- **Coarse vs Fine Granularity**:
  - Coarse: Large computational work between communication events
  - Fine: Frequent communication with smaller computation chunks
- **Impact**: Higher computation/communication ratio → better speedup and efficiency

2.2 Overheads

- **Types**:
  - Synchronization costs
  - Data communication costs
- **Significance**: Overheads reduce effective parallel speedup

2.3 Scalability

- **Definition**: Ability to maintain performance improvement with increasing processors
- **Influencing Factors**:
  - Memory-CPU bandwidth
  - Network communication capabilities
  - Application algorithm design
  - Programming language characteristics
  - Process assignment strategy

[Summary] Granularity, overheads, and scalability are critical factors affecting parallel performance, with granularity balancing computation and communication trade-offs.

3 Processor and System Characteristics

3.1 Homogeneous vs Heterogeneous Systems

- **Homogeneous Processors**:
  - All processors identical in type and capability
  - Uniform computing and communication costs
- **Heterogeneous Processors**:
  - Processors with varying capabilities (speed, resources, software)
  - Different computing and communication costs
  - May require specific processors for certain processes

3.2 Network Characteristics

- **Homogeneous/Heterogeneous Networks**:
  - Communication bandwidth may vary
  - Considerations: Mobility, disconnection issues

3.3 Cost Modeling

- **Total Cost Formula**:

$$\text{Total Cost} = \text{Computing Costs} + \text{Communication Costs}$$

- **Example Calculation**:
  - Given process costs on different processors
  - Communication cost between processors = 1 unit
  - Assignment: (B,C,D) $\rightarrow$ P1; (A,E,F) $\rightarrow$ P2
  - Total Cost = Sum of computation costs + communication costs

[Summary] System heterogeneity affects assignment decisions, with total cost being the sum of computation and communication expenses.

4 Problem and System Knowledge Requirements

4.1 Problem Analysis

- **Focus Areas**:
  - Parallelize most time-consuming processes
  - Avoid parallelizing trivial processes
  - Identify bottlenecks: I/O, data dependencies

4.2 System Analysis

- **Processor Characteristics**:
  - Speed, memory capacity
- **Topology Types**:
  - Mesh, Tree, Hypercube, 3-D Mesh
- **Communication Patterns**:
  - Processor-Processor
  - Processor-Memory
  - Memory-Memory

5 Decomposition Strategies

5.1 Domain Decomposition

- **Approach**: Divide data into discrete chunks
- **Applications**: Matrix operations, Image processing
- **Goal**: Maintain high $\frac{\text{cost of computing (R)}}{\text{cost of communication (C)}}$ ratio
- **Considerations**: Match system (R,C) with application (r,c)

5.2 Functional Decomposition

- **Approach**: Assign different functions to different processors
- **Applications**: Signal processing (pipelined filter stages)
- **Key Principle**: Maintain matching R/C to r/c ratios for improved parallelism

[Summary] Domain decomposition partitions data, while functional decomposition partitions functions, both aiming to optimize computation-communication ratios.

6 Process Assignment and Scheduling Types

6.1 Static Scheduling

- **Characteristics**:
  - Problem and process complexities known *a priori*
  - Fixed assignments determined before execution
- **Applications**: Traditional parallel systems with predictable workloads

6.2 Dynamic Scheduling

- **Characteristics**:
  - Processor availability changes over time
  - Adapts to system state changes
  - Combined with process migration and load balancing
- **Applications**:
  - Cloud systems (shared resources)
  - Mobile systems (mobility, battery constraints)

[Summary] Static scheduling works with known parameters, while dynamic scheduling adapts to changing system conditions.

7 Process Scheduling Approaches

7.1 Centralized vs Decentralized

- **Centralized Scheduling**:
  - Single controller makes all decisions
  - Pros: Consistent, global view
  - Cons: Single point of failure, scalability issues
- **Decentralized Scheduling**:
  - Distributed decision making
  - Pros: Scalable, fault-tolerant
  - Cons: Coordination overhead, potential inconsistencies

7.2 Primary Issues: Problem Perspective

- **Workload Distribution**:
  - Distributing jobs and metadata
  - Queue length management
- **Session State Management**:
  - Node stickiness (affinity)
  - Cost of task reallocation
  - Session state distribution

7.3 Primary Issues: System Perspective

- **Node Selection Policies**:
  - Random selection
  - Round Robin
  - Shortest queue
  - Threshold-based (queue length ¿ threshold)
- **Workload Metrics**:
  - Queue Length
  - CPU Utilization
  - Response Time, Capacity, Network latency
  - Probe limit

8 Optimal Scheduling Analysis

8.1 Mathematical Formulation

- **System Model**:
  - N tasks: $p_1, p_2, p_3, \ldots, p_N$
  - 2 processors: A and B
  - Assignment: A: $p_1, p_2, \ldots, p_k$; B: $p_{k+1}, p_{k+2}, \ldots, p_N$
- **Cost Components**:
  - Computation cost for process $p_i = r_i$
  - Communication cost between p_i and $p_j = c_{i,j}$ if on different processors
  - Communication cost = 0 if on same processor

8.2 Cost Analysis

- **Total Cost Formula**:

$$\text{Total Cost} = r \times \max(k, N-k) + c \times [k \times (N-k)]$$

- **Special Cases**:
  - All processes on one processor ($k = N$):

$$\text{Cost} = r \times N$$

  - Equal division ($k = N/2$):

$$\text{Cost} = \frac{r \times N}{2} + c \times \left(\frac{N}{2} \times \frac{N}{2}\right) = \frac{1}{2}\left(r \times N + \frac{c \times N^2}{2}\right)$$

8.3 Parallelization Decision

- **Condition for Parallelization**:

$$r \times N > \frac{1}{2}\left(r \times N + \frac{c \times N^2}{2}\right)$$

  Simplifies to:

$$\frac{r}{c} > \frac{N}{2}$$

- **Decision Rule**:
  - Equal distribution if $\frac{r}{c} > \frac{N}{2}$
  - Use single processor if $\frac{r}{c} \leq \frac{N}{2}$
  - Decision independent of number of processors

8.4 Assumptions and Limitations

- **Key Assumptions**:
  - Total communication among all processes
  - No overlap between computation and communication
  - All processes communicate with each other
- **Real-world Considerations**:
  - Subset of processes communicate
  - Computation and communication can often overlap

[Summary] Optimal scheduling depends on the computation-to-communication cost ratio, with parallelization beneficial only when $r/c > N/2$.

9 Clustering in Process Scheduling

9.1 Clustering Concept

- **Objective**: Group processes to minimize communication costs
- **Computation Cost**: $\sum_{i=1}^{n} r_i$
- **Communication Cost**: $\sum c_{in} + \sum c_{out}$

9.2 Clustering Strategies

- **Input/Output Focus**:
  - Consider incoming and outgoing communication costs
  - Balance computation and communication within clusters

10 Dynamic Load Balancing

10.1 Basic Concepts

- **Definition**: Equitable distribution of load among processors
- **Goal**: Minimize difference between most heavily and lightly loaded processors
- **Key Principle**: Adjust based on monitored system state

10.2 Dynamic Scheduling Algorithms

- **Sender-Initiated Algorithms**:
  - Transfer policy: When queue length exceeds threshold
  - Selection policy: Which process to transfer
  - Location policy: Cost and distance considerations
- **Receiver-Initiated Algorithms**:
  - Triggered when queue length falls below threshold

10.3 Dynamic System Challenges

- **Sources of Dynamism**:
  - Uncertainty in task execution times
  - Dynamic task arrival and departure
  - Changing processor availability
  - Network condition variations
  - Task priority changes
  - Processor and network faults
- **Exacerbating Factors**:
  - Cloud systems (resource sharing)
  - Mobile systems (mobility, energy constraints)

[Summary] Dynamic load balancing adapts to changing system conditions using sender or receiver initiated approaches to maintain equitable load distribution.

11 Load Balancing Methods and Challenges

11.1 Approaches to Load Balancing

- **Centralized vs Decentralized**:
  - Centralized: Single controller, less scalable
  - Decentralized: Distributed control, more scalable
- **Information Collection**:
  - How to obtain processor state information
  - Centralized vs decentralized collection
  - Periodic vs event-driven updates
  - Threshold-based monitoring

11.2 Key Questions in Load Balancing

- **Transfer Policy**: Whether to move processes
- **Location Policy**: Where to move processes
- **Process Selection**: Which processes to move
- **Decision Architecture**: Centralized or distributed control

11.3 Cost Considerations

- **Processing Overhead**:
  - Data collection for load monitoring
  - Decision making computations
- **Network Overhead**:
  - Distribution of load information
  - Process migration costs
  - Job redistribution

12 Static vs Dynamic Load Balancing

12.1 Static Load Sharing

- **Characteristics**:
  - Fixed policy, no adaptation to system state
  - Simple, low cost implementation
  - Handles session state easily
  - Cannot adjust to dynamic changes

12.2 Dynamic Load Balancing

- **Advantages**:
  - Adapts to changing system conditions
  - Better resource utilization
  - Handles unpredictable workloads
- **Disadvantages**:
  - Complex implementation
  - Significant overhead costs
  - Session state management challenges

13 Process Migration

13.1 Migration Process

- **Steps**:
  - Migration request initiation
  - Process suspension on source host
  - State transfer to destination host
  - Process resumption on destination
  - File server coordination

13.2 Migration Scenarios

- **Work Stealing**:
  - Idle processor seeks work from busy ones
  - "No more work → Find work elsewhere"
- **Load Distribution**:
  - Balance load across multiple processors
  - Handle processor saturation scenarios

14 Distributed Scheduling Framework

14.1 System Components

- **Load Information Management**:
  - Collects local node load information
  - Disseminates information to other nodes
- **Distributed Scheduling**:
  - Makes migration decisions
  - Determines when, where, and which processes to migrate
- **Migration Mechanism**:
  - Executes the actual process transfer

14.2 Implementation Considerations

- **Local Information Collection**: Monitoring node status
- **Information Dissemination**: Sharing load data
- **Migration Directives**: Decision rules for process movement

Study Aids

Key Formulas

- Total Cost = Computing Costs + Communication Costs
- Optimal scheduling condition: $\frac{r}{c} > \frac{N}{2}$
- Clustering costs: Computation = $\sum r_i$, Communication = $\sum c_{in} + \sum c_{out}$

Important Concepts

- Granularity: Computation-to-communication ratio
- Homogeneous vs Heterogeneous systems
- Static vs Dynamic scheduling
- Centralized vs Decentralized control
- Sender vs Receiver initiated load balancing

Exam Questions

1. Compare and contrast static and dynamic scheduling approaches, including their advantages and limitations.
2. Derive the condition for optimal parallelization in a 2-processor system and explain its practical implications.
3. Describe three node selection policies for load balancing and discuss situations where each would be most appropriate.
4. Explain the process migration mechanism and discuss the costs involved in dynamic load balancing.
5. How does system heterogeneity affect process assignment decisions? Provide examples.

[Mnemonic] "GCD SL" - Granularity, Cost, Dynamism for Scheduling and Load balancing
[Concept Map]

- Process Assignment → Where (Mapping)
- Scheduling → When (Timing)
- Load Balancing → How (Distribution)
- Static vs Dynamic approaches
- Centralized vs Decentralized control
- Homogeneous vs Heterogeneous systems