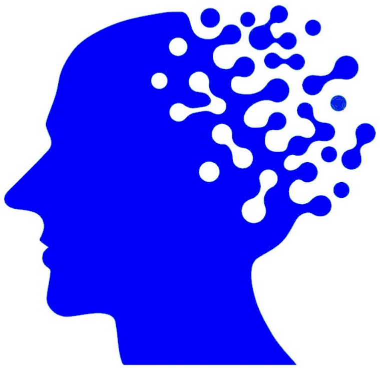



Spring 2023

Distributed Computing
- Distributed Sorting Algorithms

Dr. Rajendra Prasath

Indian Institute of Information Technology Sri City, Chittoor

19-20th January 2023 (http://rajendra.2power3.com)

> Distributed Computing?

➤ How will you design a Distributed Algorithm?

➤ Learn to Solve using Distributed Algorithms

Overview

> About this Lecture

What do we learn today?

- This covers a model of distributed computations that every algorithm designer needs to know

  - **Challenges and Goals**
  - **A model of distributed executions**

with an application to

  - **Distributed Sorting on a line network**

Let us explore these topics → → →

Today's Focus

Recap: Distributed Systems

A Distributed System:

- A collection of independent systems that appears to its users as a single coherent system
- A system in which hardware and software components of networked computers communicate and coordinate their activity only by passing messages
- A computing platform built with many computers that:
  - Operate concurrently
  - Are physically distributed (have their own failure modes)
  - Are linked by a network
  - Have independent clocks

Recap: Characteristics

- Concurrent execution of processes:
  - Non-determinism, Race Conditions, Synchronization, Deadlocks, and so on
- No global clock
  - Coordination is done by message exchange
  - No Single Global notion of the correct time
- No global state
  - No Process has a knowledge of the current global state of the system
- Units may fail independently
  - Network Faults may isolate computers that are still running
  - System Failures may not be immediately known

Recap: Flynn's Classification

- Single Instruction Single Data (SISD) Stream
  - Traditional von Neumann architecture
- Single Instruction Multiple Data (SIMD) Stream
  - Scientific Applications, Vector Processors, array processors and so on
- Multiple Instruction Single Data (MISD) Stream
  - Visualization is an example
- Multiple Instruction Multiple Data (MIMD) Stream
  - Distributed Systems

Recap: Message Passing Systems

Basic Primitive Operations

- Send
  - *send* message from process A to Process B

    A → B

- Receive
  - *receive* message at Process B from Process A

    B ← A

- Compute at A and / or B
  - do the specific computations at A and / or B

Challenges / Goals of DS

What are the challenges / goals of distributed systems?

- Heterogeneity
- Openness
- Security
- Scalability
- Failure Handling
- Concurrency
- Transparency

A Distributed Program

- A distributed program is composed of a set of n asynchronous processes, $p_1, p_2, \ldots, p_i, \ldots, p_n$
- The processes do not share a global memory and communicate solely by passing messages
- The processes do not share a global clock that is instantaneously accessible to these processes
- Process execution and message transfer are asynchronous
- Without loss of generality, we assume that each process is running on a different processor
- Let C_{ij} denote the channel from process p_i to p_j and let m_{ij} denote a message sent by p_i to p_j
- The message transmission delay is finite and unpredictable

A Model of Distributed Executions

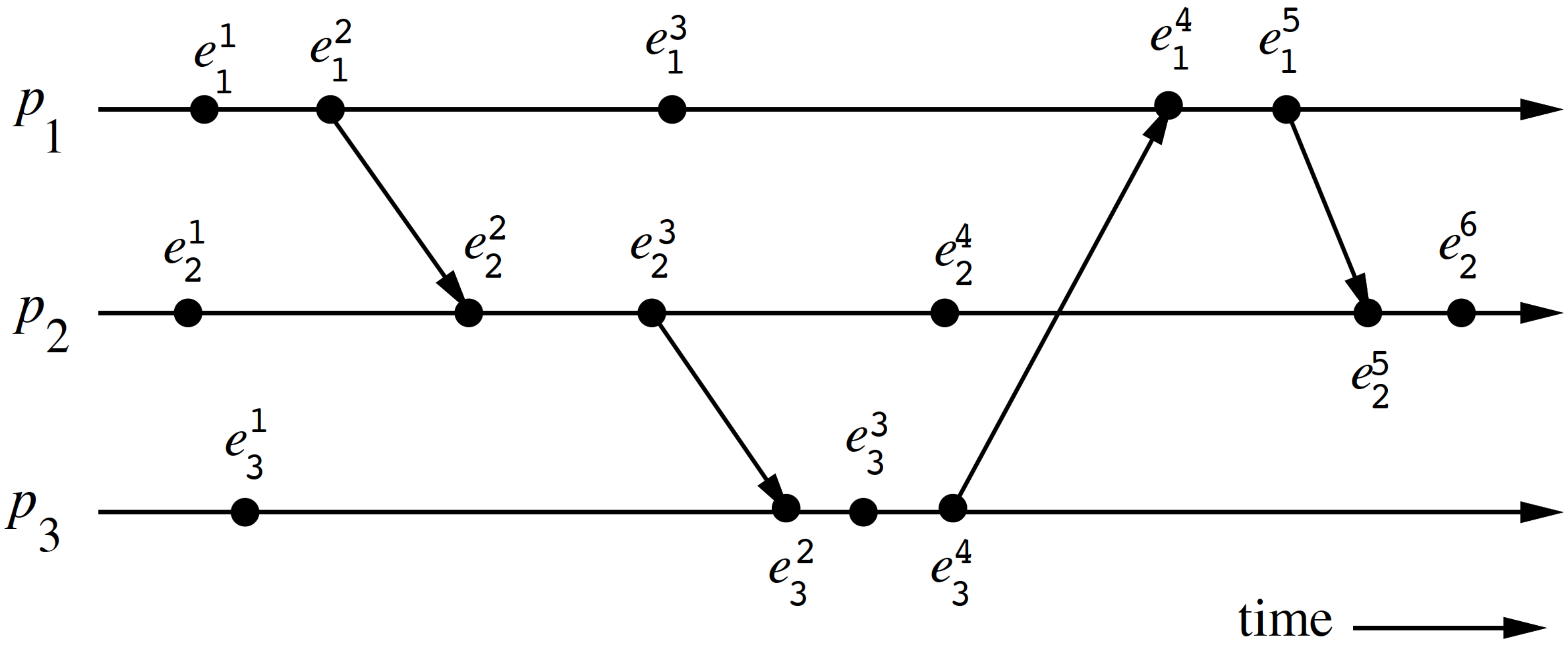
- The execution of a process consists of a sequential execution of its actions.
- The actions are atomic and modeled as three types of events: internal events, message send events, and message receive events
- Let e_i^x denote the x^{th} event at process p_i .
- For a message m, let send(m) and receive(m) denote send and receive events, respectively.
- The occurrence of events changes the states of respective processes and channels.
- Internal event → changes state of the process
- Send and Receive events change the state of the process that sends / receives the message & the state of the channel on which the message is sent / received respectively

A State-Time diagram

- The evolution of a distributed execution is depicted by a space-time diagram
- A horizontal line represents the progress of a specific process
- A dot indicates an event
- A slant arrow indicates a message transfer

- Since an event execution is atomic (indivisible and instantaneous), it is justified to denote it as a dot on a process line

A State-Time diagram – An Example



- **For Process p_1:**

  Second event is a message send event

  First and Third events are internal events

  Fourth event is a message receive event

Distributed Sorting – An example

Why Sorting?

Fundamental problem in computing

Distributed Sorting (DS):

- Initially, each process P_i has an element s_i for sorting
- n Elements are arranged in a **Line network**
- Position of each element has to be rearranged to satisfy the condition

$$s_i \leq s_i+1$$

in each process P_i , $1 \leq i < n$, at the final state

- Find a strategy to minimize the amount of communication (in terms of the number of message exchanges)

Odd-Even Transposition Sort

Odd-Even Transposition Sorting - (n) rounds

(odd - i): $P_{i\ (=odd,\ )}\ (v_i) \leftarrow\rightarrow P_{i+1}\ (v_{i+1})$, if $v_i > v_{i+1}$
(even - i): $P_{i\ (=even,\ )}\ (v_i) \leftarrow\rightarrow P_{i+1}\ (v_{i+1})$, if $v_i > v_{i+1}$

Requires knowledge about Global position

Example:

- Consider Sorting of 5 elements

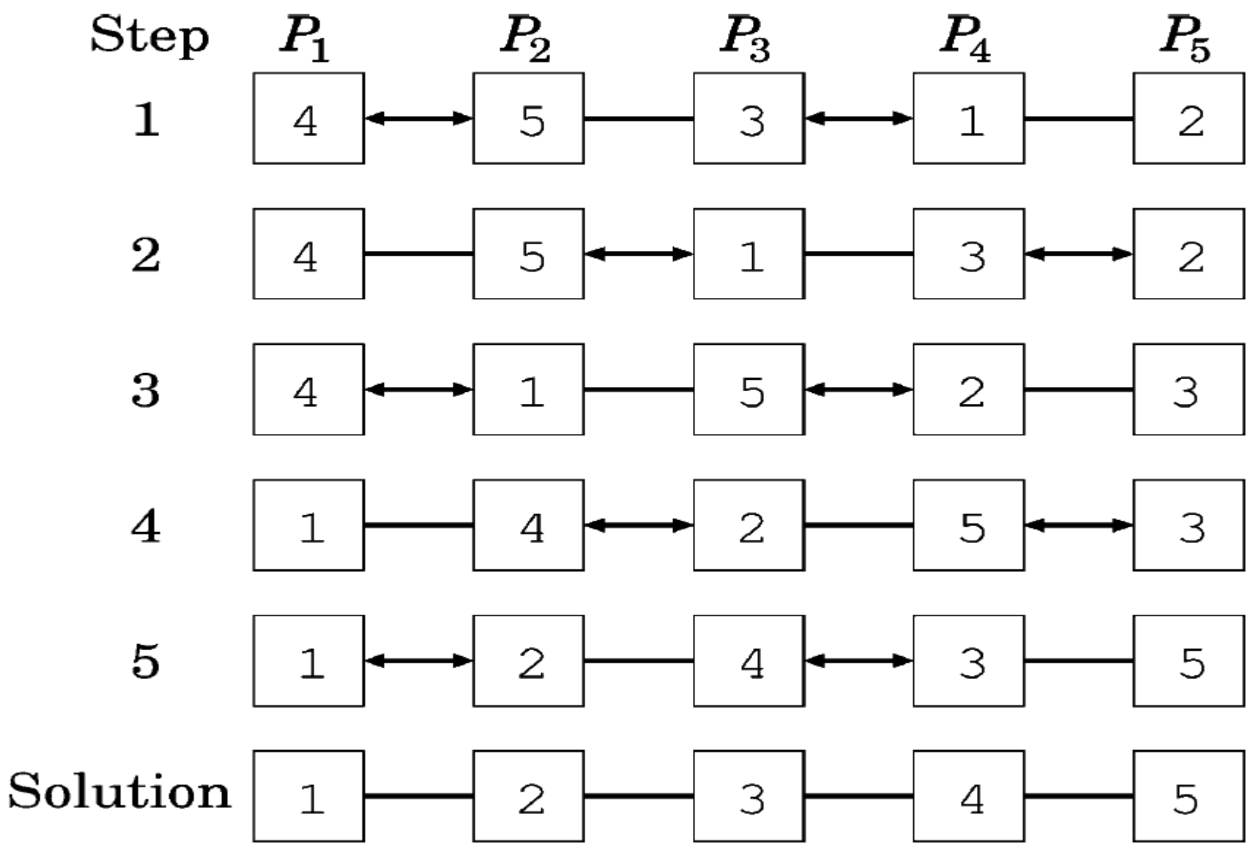
  4 5 3 1 2

- Each element is kept with a process P_i

- Line network – the underlying network that connects all processes

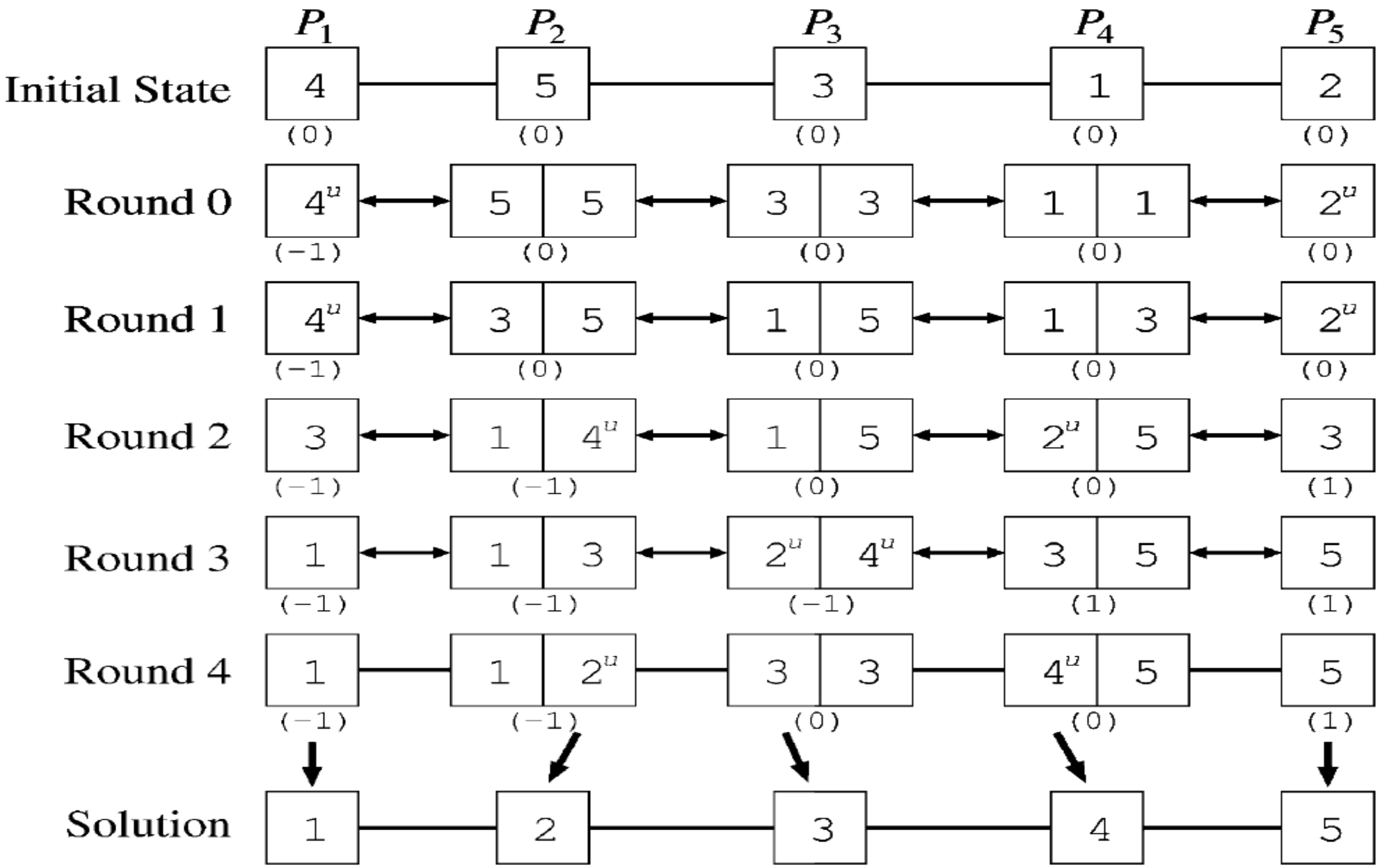
Odd-Even Transposition Sort

An Illustrative Example:



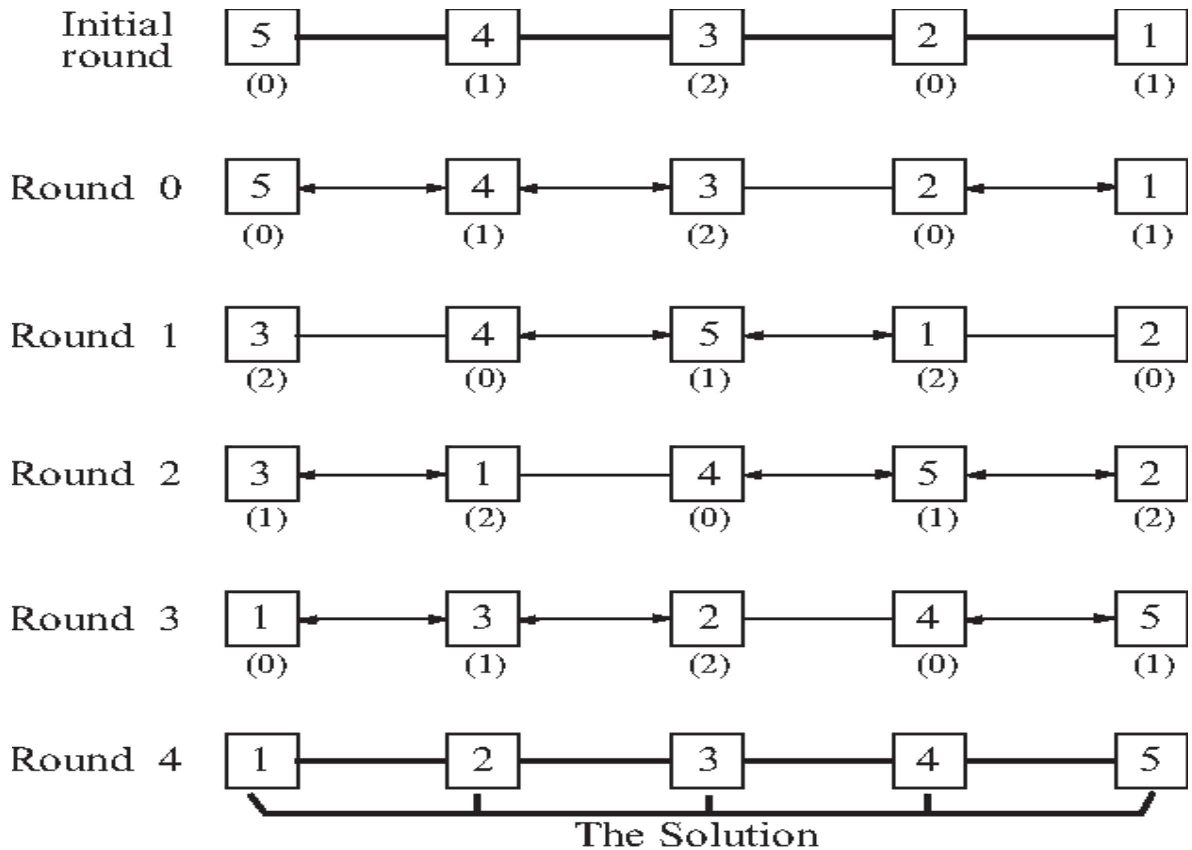
Distributed Sorting – Sasaki's (n-1) round

[Sasaki, 2002]



No Global position; Make copies of elements at intermediate nodes; Rule to select Final Solution; Computing n at runtime

Distributed Sorting – An alternative (n-1) round



DO NOT MAKE copies of elements at intermediate nodes;
No Rule to Select the Final Solution;
No Global position; Computing n at runtime

A Model of Distributed Executions

- The events at a process are linearly ordered by their order of occurrence.
- The execution of process p_i produces a sequence of events $e_i^1, e_i^2, \ldots, e_i^x, e_i^{x+1}, \ldots$ and is denoted by H_i where

$$H_i = (h_i, \rightarrow i)$$

h_i is the set of events produced by p_i and binary relation $\rightarrow i$ defines a linear order on these events

- **Linear Relation:** Mathematically, the independent variable is multiplied by the slope coefficient, added by a constant, which determines the dependent variable
- Relation $\rightarrow i$ expresses causal dependencies among the events of p_i

A Model of Distributed Executions (contd)

- The send and the receive events signify the flow of information between processes and establish causal dependency from the sender process to the receiver process
- Define a relation $\rightarrow_{msg}$ that captures the causal dependency due to message exchanges as follows:

  For every message m that is exchanged between two processes, we have

$$send(m) \rightarrow_{msg} receive(m)$$

- Relation $\rightarrow_{msg}$ defines causal dependencies between the pairs of corresponding send and receive events

A Few Applications

- ➔ Mobile Systems
- ➔ Sensor networks
- ➔ Pervasive Computing
  - ➔ Smart workplace
  - ➔ Intelligent Home
- ➔ Peer-to-peer computing
- ➔ Distributed Agents
- ➔ Distributed Data Mining
- ➔ Grid Computing
- ➔ Security aspects in Distributed Systems

Summary

- Goals and Challenges of DS
  - Fundamental aspects while building distributed applications
- A model of Distributed Computations
  - Primitives of Distributed Communications
    - Message Passing is the main focus
  - Properties of distributed Computations
  - Distributed Sorting
  - Events and their ordering
    - How to handle Causal Precedence ?
    - Lamport's Logical Clocks ?
    - Many more to come up … stay tuned in !!

Penalties

- Every Student is expected to strictly follow a fair Academic Code of Conduct to avoid penalties

- Penalties is heavy for those who involve in:
  - Copy and Pasting the code
  - Plagiarism (copied from your neighbor or friend - in this case, both will get "0" marks for that specific take home assignments)
  - If the candidate is unable to explain his own solution, it would be considered as a "copied case"!!
  - Any other unfair means of completing the assignments

Help among Yourselves?

- **Perspective Students** (having CGPA above 8.5 and above)
- **Promising Students** (having CGPA above 6.5 and less than 8.5)

- **Needy Students** (having CGPA less than 6.5)
  - Can the above group help these students? (Your work will also be rewarded)

- You may grow a culture of **collaborative learning** by helping the needy students

How to reach me?

- ➔ Please leave me an email:

  rajendra [DOT] prasath [AT] iiits [DOT] in

- ➔ Visit my homepage @
  - ➔ https://www.iiits.ac.in/people/regular-faculty/dr-rajendra-prasath/

    (OR)

  - ➔ http://rajendra.2power3.com

Assistance

- You may post your questions to me at any time
- You may meet me in person on available time or with an appointment
- You may ask for one-to-one meeting

Best Approach

- You may leave me an email any time
  (email is the best way to reach me faster)

Questions
It's Your Time

Contact Information

Dr. Rajendra Prasath
IIIT Sri City, Chittoor