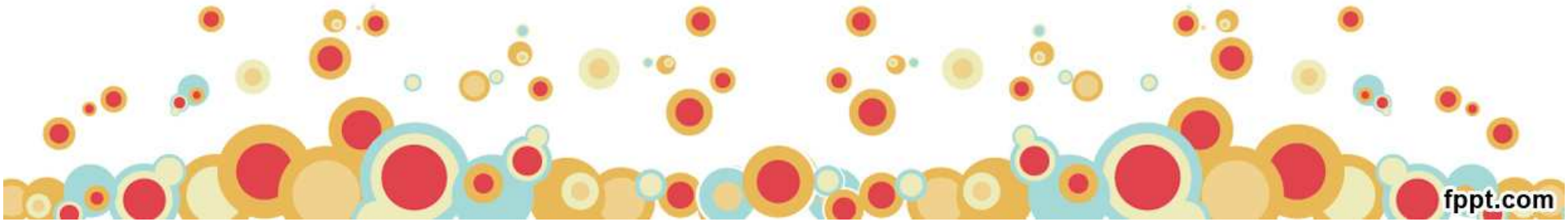
Process Management

Reference: Pradeep K Sinha,

"Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2007

Overview

- Process Migration
  - Features
  - Mechanisms



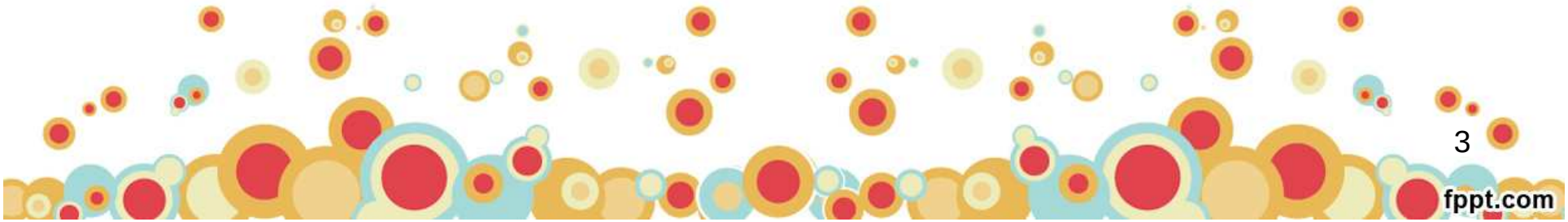
Introduction

- In conventional (centralized) operating system, process management deals with mechanisms and policies for sharing the processor of the system among all processes.
- In a distributed operating system, the main goal of process management is to make the best possible use of the processing resources of the entire system by sharing them among all processes.
- **Three important concepts are used to achieve this goal:**

  -Processor allocation

  -Process migration

  -Threads



Introduction

- **Process allocation** deals with the mechanism of deciding which process should be assigned to which processor.
- **Process migration** deals with the movement of a process from its current location to the processor to which it has been assigned.
- **Threads** deals with fine-grained parallelism for better utilization of the processing capability of the system.

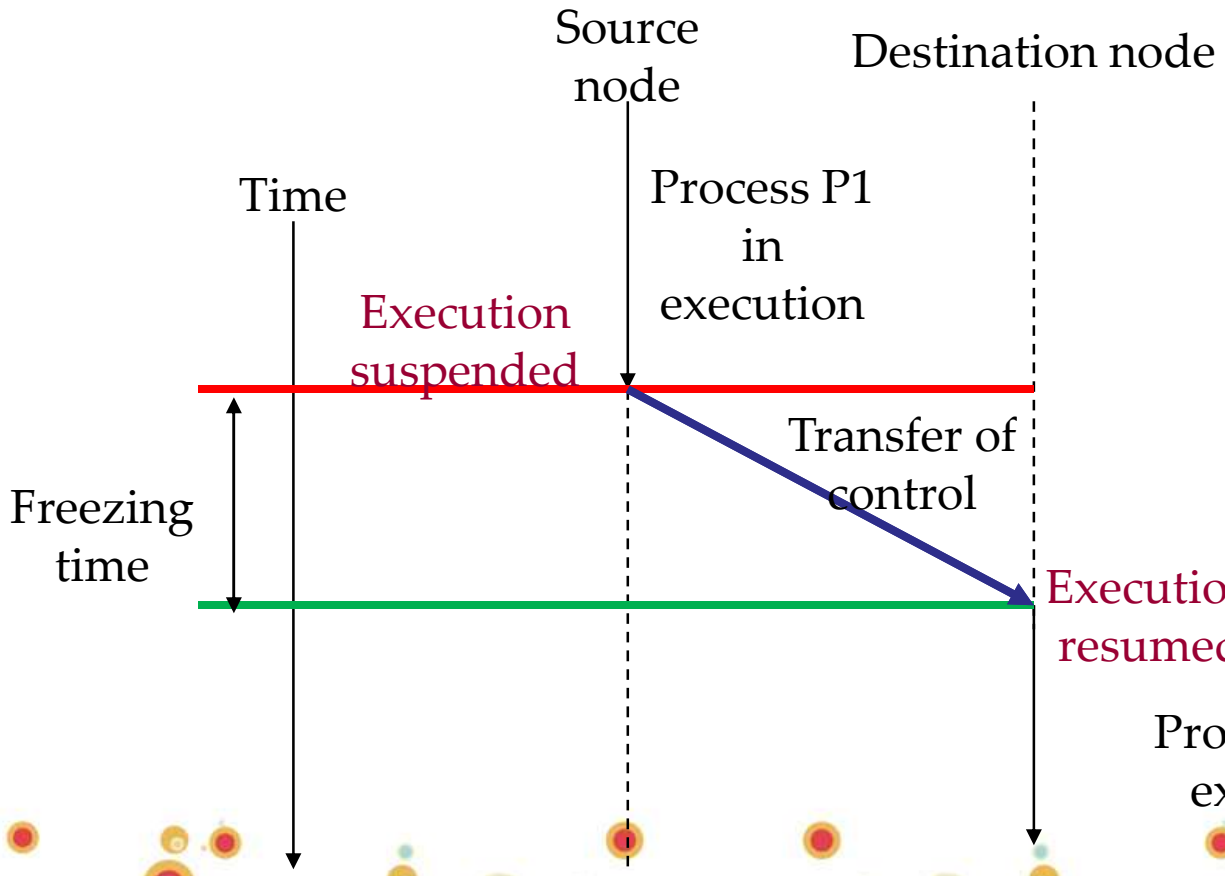
Process Migration

- Relocation of a process from its current location to another node.

- Process may be migrated
  - either before it start executing on its source node
    - Known as **non-preemptive process migration.**

  - or during the course of its execution
    - Known as **preemptive process migration.**

Flow of execution of a migration process



Process Migration

- Preemptive process migration is costlier.

- **Process Migration Policy**
  - Selection of a process that should be migrated.
  - Selection of the destination node to which the selected process should be migrated.

- **Process Migration Mechanism**
  - Actual transfer of the selected process to the destination node.

Desirable features of a good process migration mechanism

- Transparency
- Minimal Interference
- Minimal Residual Dependencies
- Efficiency
- Robustness
- Communication between co-processes of a job

Transparency

Level of transparency:

- Object Access Level
- System Call & Interprocess Communication level

Object Access level Transparency

- Minimum requirement for a system to support non-preemptive process migration facility.
- Access to objects such as files and devices can be location independent.
- Allows free initiation of program at arbitrary node.
- Requires transparent object naming and locating.

System Call & IPC level

- For migrated process, system calls should be location independent.
- For transparent redirection of messages during the transient state of a process.
- Transparency must be provided to support preemptive process migration facility.

Minimal Interference

- Migration of a process should cause minimal interference of progress of the process involved.

- Achieve by minimizing the freezing time of the process being migrated.
  - Freezing time is the time period for which the execution of the process is stopped for transferring its information to the destination node.

Minimal Residual Dependencies

- No residual dependency should be left on previous node.

Efficiency

- Minimum time required for migrating a process.
- Minimum cost of locating an object.
- Minimum cost of supporting remote execution once the process is migrated.

Robustness

- The failure of a node other than the one on which a process is currently running should not in any way affect the accessibility or execution of that process.

Communication between coprocessors of a job

- To reduce communication cost, it is necessary that coprocesses are able to directly communicate with each other irrespective of their locations.

Process Migration Mechanisms

Four major activities

- Freezing the process on its source node and restarting it on its destination node.
- Transferring the process's address space from its source node to its destination node.
- Forwarding messages meant for the migrant process.
- Handling communication between cooperating processes that have been separated as a result of process migration.

Mechanisms for Freezing and Restarting a Process

- **Freezing the process:**
  - The execution of the process is suspended and all external interactions with the process are deferred.

- **Issues:**
  - Immediate and delayed blocking of the process
  - Fast and slow I/O operations
  - Information about open files
  - Reinstating the process on its Destination node

Immediate and delayed blocking of the process

- If the process is not executing a system call, it can be immediately blocked from further execution.
- If the process is executing a system call but is sleeping at an interruptible priority waiting for a kernel event to occur, it can be immediately blocked from further execution.
- If the process is executing a system call but is sleeping at an non-interruptible priority waiting for a kernel event to occur, it cannot be blocked immediately.

Fast and Slow I/O Operations

- Process is frozen after the completion of all fast I/O operations like disk access.
- Slow I/O operation (pipe or terminal) is done after process migration and when process is executed on destination node.

Information about open files

- Includes name and identifier of the file, their access modes and the current positions of their file pointers.
- OS returns a file descriptor to process that is used for all I/O.
- It is necessary to somehow preserve a pointer to the file so that migrated process could continue to access it.

Information about open files

- **Approaches :**
  - Link is created to the file and the pathname of the link is used as an access point to the file after the process migrates.
  - An open file's complete pathname is reconstructed when required by modifying the kernel.
- Keeping Track of file replicas.

Reinstating the process on its Destination Node

- On the destination node, an empty process state is created.
- Newly allocated process may or may not have the same process identifier as the migrating process.

Reinstating the process on its Destination Node

- Once all the state of the migrating process has been transferred from the source to destination node and copied into the empty state, new copy of the process is unfrozen and old copy is deleted.

- The process is restarted on its destination node in whatever state it was in before being migrated.

Address Space Transfer Mechanisms

- The migration of a process involves the transfer of
  - Process's state
  - Process's address space

  from the source node to the destination node.

Address Space Transfer Mechanisms

- **Process State consists of**
  - Execution Status – Register Contents
  - Memory Tables
  - I/O State : I/O Queue, I/O buffers, Interrupts
  - Capability list
  - Process's Identifier
  - Process's user and group identifier
  - Information about Open Files

Address Space Transfer Mechanisms

- **Process address space**
  - code,
  - data and
  - program stack

- The size of the process's address space (several megabytes) overshadows the size of the process's state information (few kilobytes).

Address Space Transfer Mechanisms

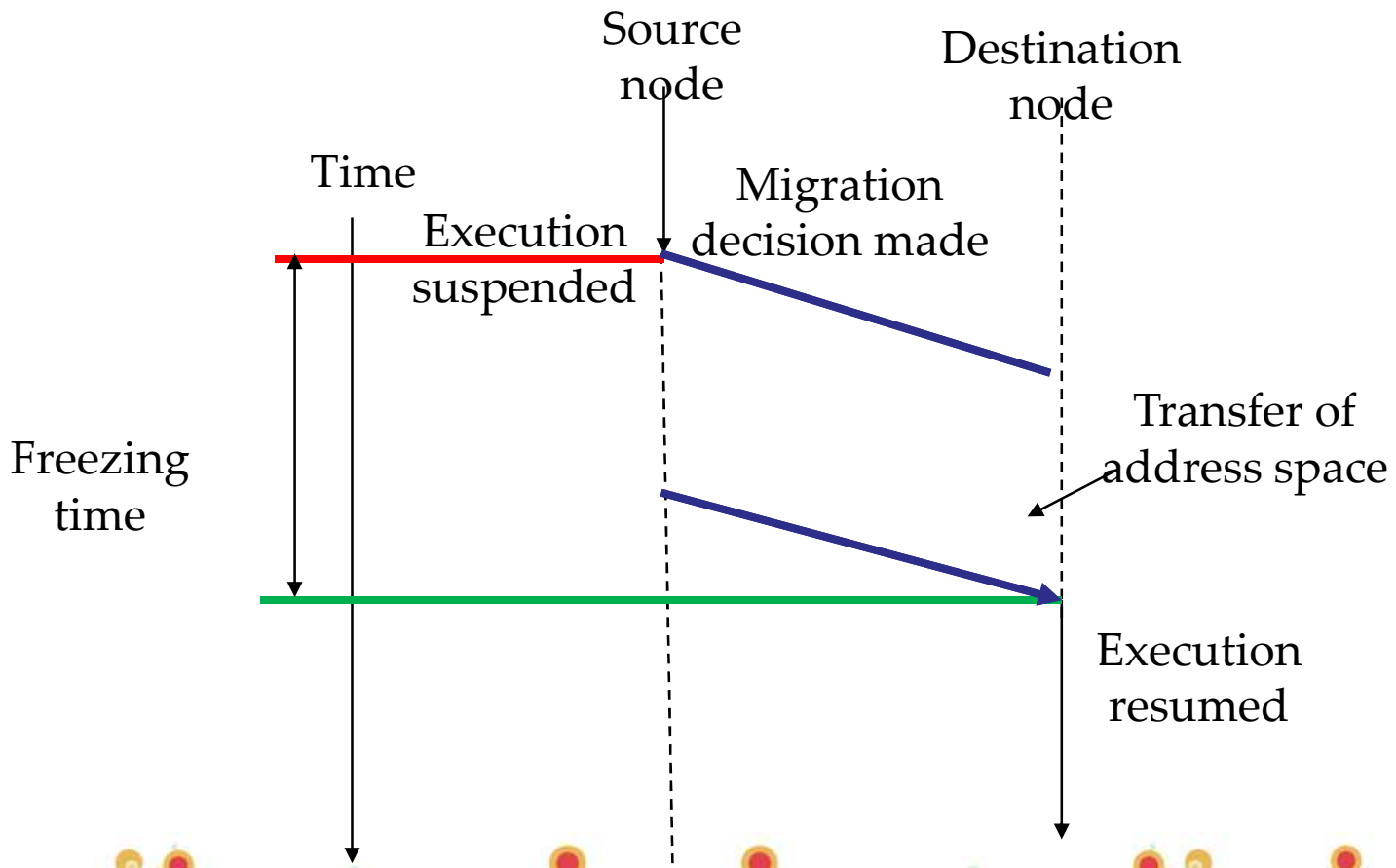
- Mechanisms for address space transfer:
  - Total freezing
  - Pretransferring
  - Transfer on reference

Total Freezing

- A process's execution is stopped while its address space is being transferred.

- Disadvantage:
  - Process is suspended for long time during migration, timeouts may occur, and if process is interactive, the delay will be noticed by the user.

Total Freezing



Pretransferring

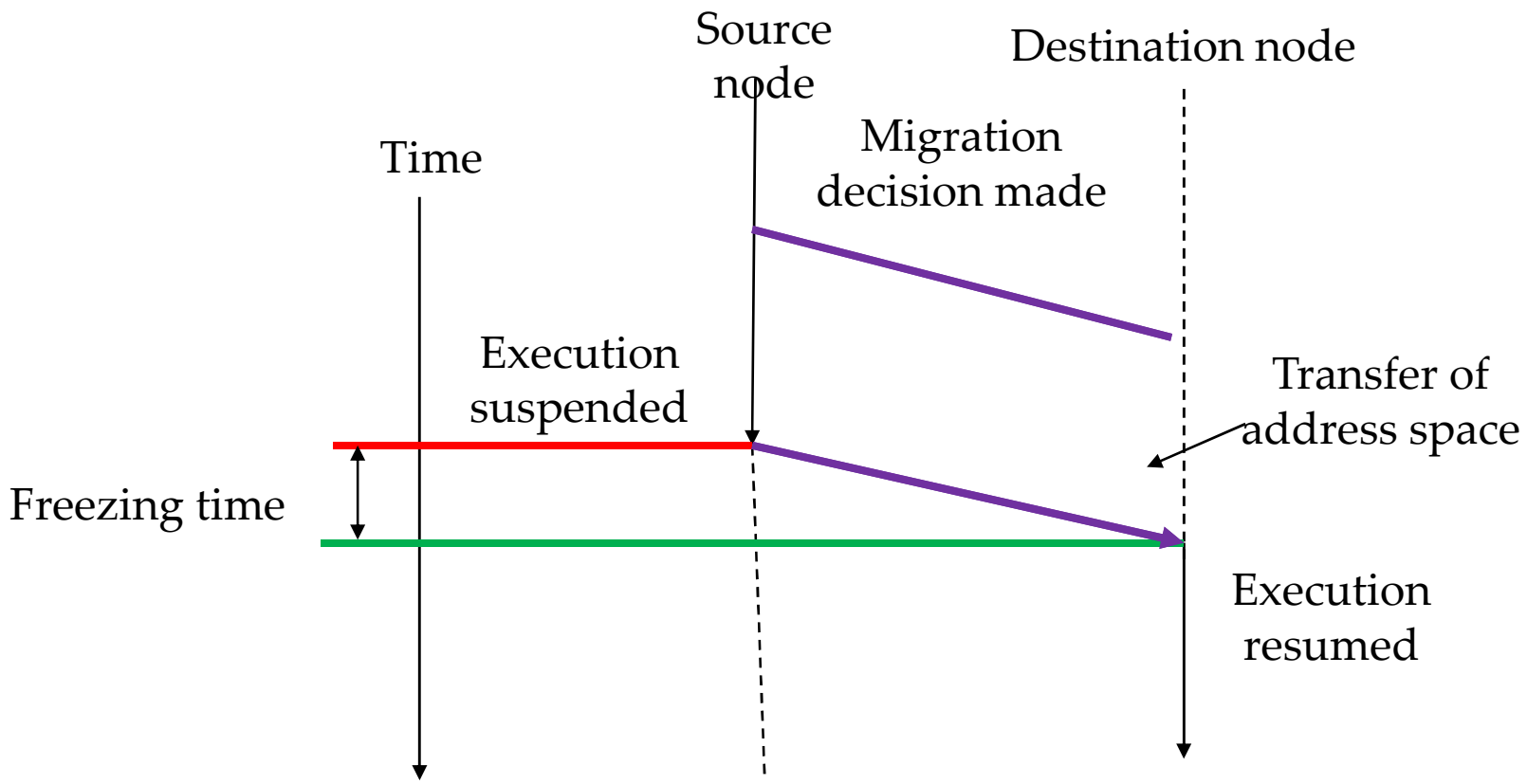
- Also known as precopying.
- The address space is transferred while the process is still running on the source node.
- It is done as an initial transfer of the complete address space followed by repeated transfers of the page modified during the previous transfer.

Pretransferring

- The pretransfer operation is executed at a higher priority than all other programs on the source node.

- Reduces the freezing time of the process but it may increase the total time for migrating due to the possibility of redundant page transfers.

Pretransferring



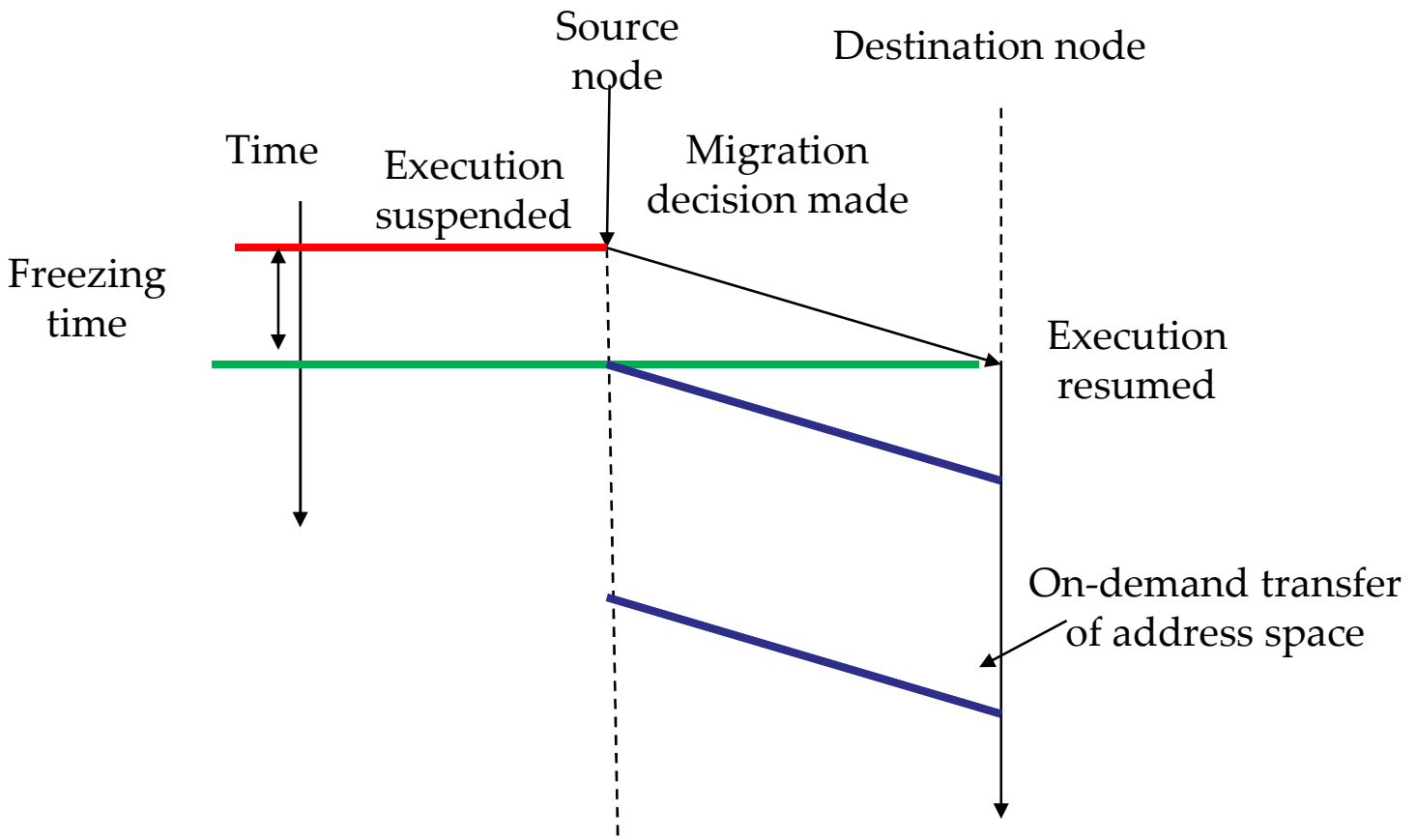
Transfer on Reference

- The process address space is left behind on its source node, and as the relocated process executes on its destination node.
- Attempts to reference memory page results in the generation of requests to copy in the desired blocks from their remote location.
- A page is transferred from its source node to its destination node only when referenced.

Transfer on Reference

- Very short switching time of the process from its source node to its destination node.
- Imposes a continued load on the process's source node and results in the process if source node fails or is rebooted.

Transfer on Reference



Message-forwarding Mechanisms

- In moving a process, it must be ensured that all pending, re-route, and future messages arrive at the process's new location.

- Types of messages:
  1. Messages received at the source node after the process's execution has been stopped on its source node and the process's execution has not yet been started on its destination node.

  2. Messages received at the source node after the process's execution has started on its destination node.

  3. Messages that are to be sent to the migrant process from any other node after it has started executing on the destination node.

Message-forwarding Mechanisms

- Mechanisms:
  - Mechanism of resending the message
  - Origin site mechanism
  - Link traversal mechanism
  - Link update mechanism

Mechanisms of resending the message

- Messages of the types 1 and 2 are returned to the sender as not deliverable or are simply dropped, with the assurance that the sender of the message is storing a copy of the data and is prepared to retransmit it.

- Does not require any process state to be left behind on the process's source node.

Mechanisms of resending the message

- Disadvantage:
  - The message forwarding mechanism of process migration operation is non-transparent to the processes interacting with the migrant process.

Origin Site Mechanism

- Each site is responsible for keeping information about the current locations of all the processes created on it.
- Messages for a particular process are always first sent to its origin site.
- The origin site then forwards the message to the process's current location.

Origin Site Mechanism

- Disadvantage:
  - Failure of the origin site will disrupt the message forwarding mechanism.
  - Continuous load on the migrant process's origin site even after the process has migrated from that node.

Link Traversal Mechanism

- To redirect message type 1, a message queue for the migrant process is created on its source node.
- For 2 and 3 type message, a forwarding address known as link is left at the source node pointing to the destination of the migrant process.
- Two component of link, one is unique process identifier and second is last known location.

Link Traversal Mechanism

- Disadvantage:
  - Several link may have to be traversed to locate a process from a node and if any node in chain of link fails, the process cannot be located.

Link Update Mechanisms

- During the transfer phase of the migrant process, the source node sends link-update messages to the kernels controlling all of the migrant process's communication partners.

- Link update message
  - Tells the new address of each link held by the migrant process.
  - Acknowledged for synchronization purposes.

Mechanisms for Handling Coprocesses

- To provide efficient communication between a process and its sub processes which might have been migrated on different nodes.

- Mechanisms :
  - Disallowing separation of coprocesses.
  - Home node or origin site concept.

Disallowing Separation of coprocesses

- Easiest method of handling communication between coprocesses is to disallow their separation.

- Methods :
  - By disallowing the migration of processes that wait for one or more of their children to complete.
  - By ensuring that when a parent process migrates, its children process will be migrated along with it.

Disallowing Separation of coprocesses

- Disadvantage:
  - It does not allow the use of parallelism within jobs.

Home node or Origin Sites Concept

- Used for communication between a process and its sub process when the two are running on different nodes.
- Allows the complete freedom of migrating a process or its sub process independently and executing them on different nodes of the system.

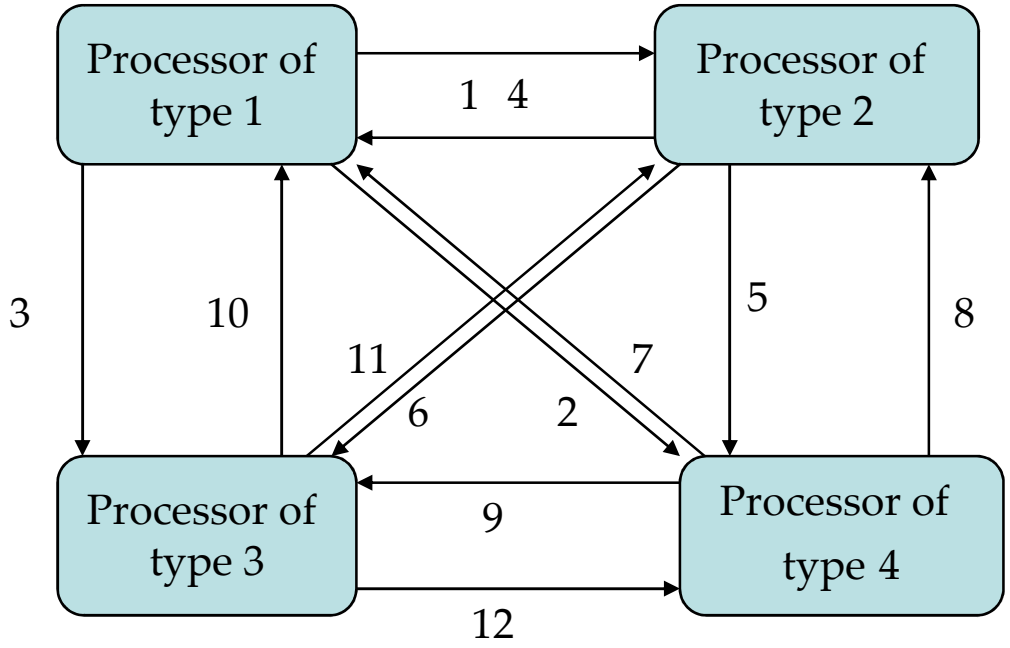
Home node or Origin Sites Concept

- Disadvantage:
  - All communication between a parent process and its children processes take place via the home node.
    - The message traffic and the communication cost increase considerably.

Process Migration in Heterogeneous Systems

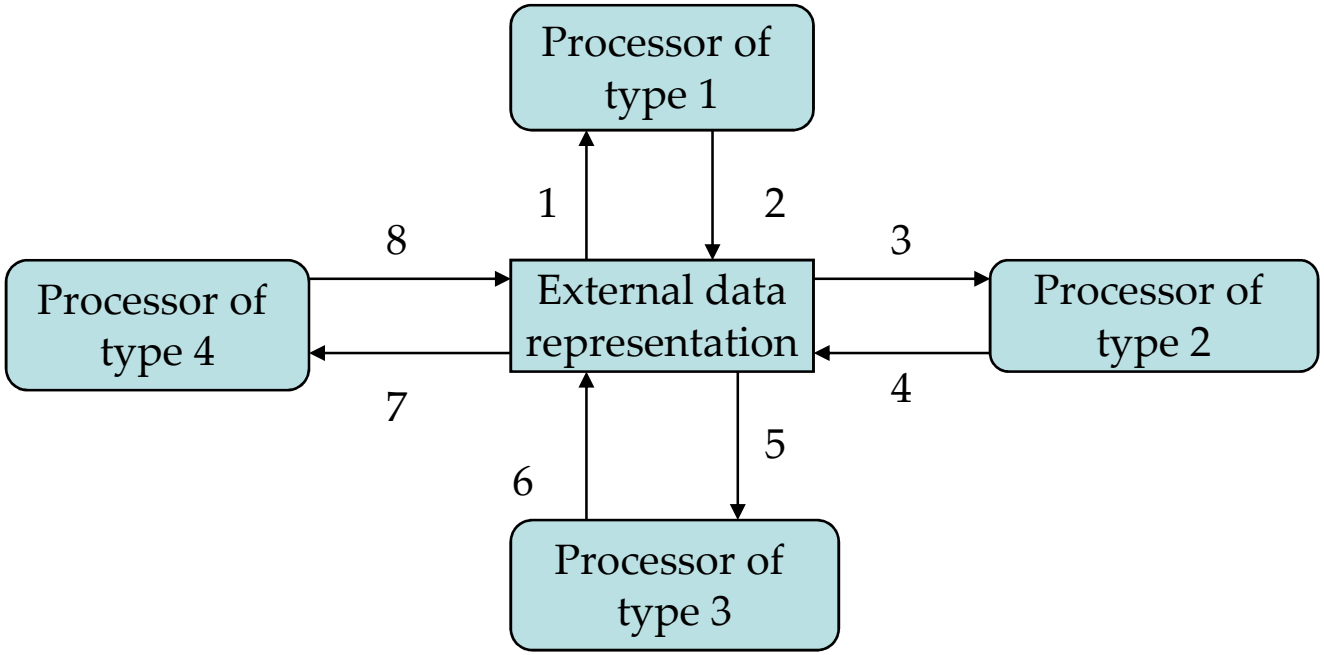
- All the concerned data must be translated from the source CPU format to the destination CPU format before it can be executed on the destination node.
- A heterogeneous system having *n* CPU types must have *n(n-1)* pieces of translation software.
- Handles problem of different data representations such as characters, integers and floating-point numbers.

Process Migration in Heterogeneous Systems



Example: The need for 12 pieces of translation software required in a heterogeneous system having 4 types of processors

Process Migration in Heterogeneous Systems



Example: The need for only 8 pieces of translation software in a heterogeneous system having 4 types of processors when the EDR mechanism is used

Handling Exponent

- The number of bits for the exponent varies from processor to processor.
- The EDR have at least as many bits in the exponent as the largest exponent.
- Overflow or underflow can be managed upon conversion from the external data representation.

Handling Exponent

- Some solutions:
  - Ensuring that numbers used by the programs that migrate have a smaller exponent value than the smallest processor's exponent value in the system.
  - Emulating the larger processor's value.
  - Restricting the migration of the process to only those nodes whose processor's exponent representation is at least as large as that of the source node's processor.

Handling the Mantissa

- Suppose processor A uses 32 bits and processor B uses 64 bits.
  - No problem from A to B migration.

Problem:

- Half-Precision problem from B to A migration.
- Loss of precision due to multiple migrations

- Solution:
  - The external data representation must have sufficient precision to handle the largest mantissa.

Advantages of Process Migration

- *Reducing average response time of processes*
  - To reduce the average response time of the processes, processes of a heavily loaded node are migrated to idle or underutilized nodes.

- *Speeding up individual jobs*
  - A migration of job to different node is done and execute them concurrently.
  - Migrate a job to a node having a faster CPU or to a node at which it has minimum turnaround time.
  - More speed up more migration cost involved.

Advantages of Process Migration

- *Gaining higher throughput*
  - Process migration facility may also be used properly to mix I/O and CPU-bound processes on a global basis for increasing the throughput of the system.

- *Utilizing resources effectively*
  - Depending upon the nature of a process, it can be migrated to suitable node to utilize the system resource in the most efficient manner.

Advantages of Process Migration

- *Reducing network traffic*
  - Migrating a process closer to the resources it is using most heavily.

- *Improving system reliability*
  - Migrate a copy of a critical process to some other node and to execute both the original and copied processes concurrently on different nodes.

- *Improving system security*
  - A sensitive process may be migrated and run on a secure node.

Thank You