Process Management in DS

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

Cont…

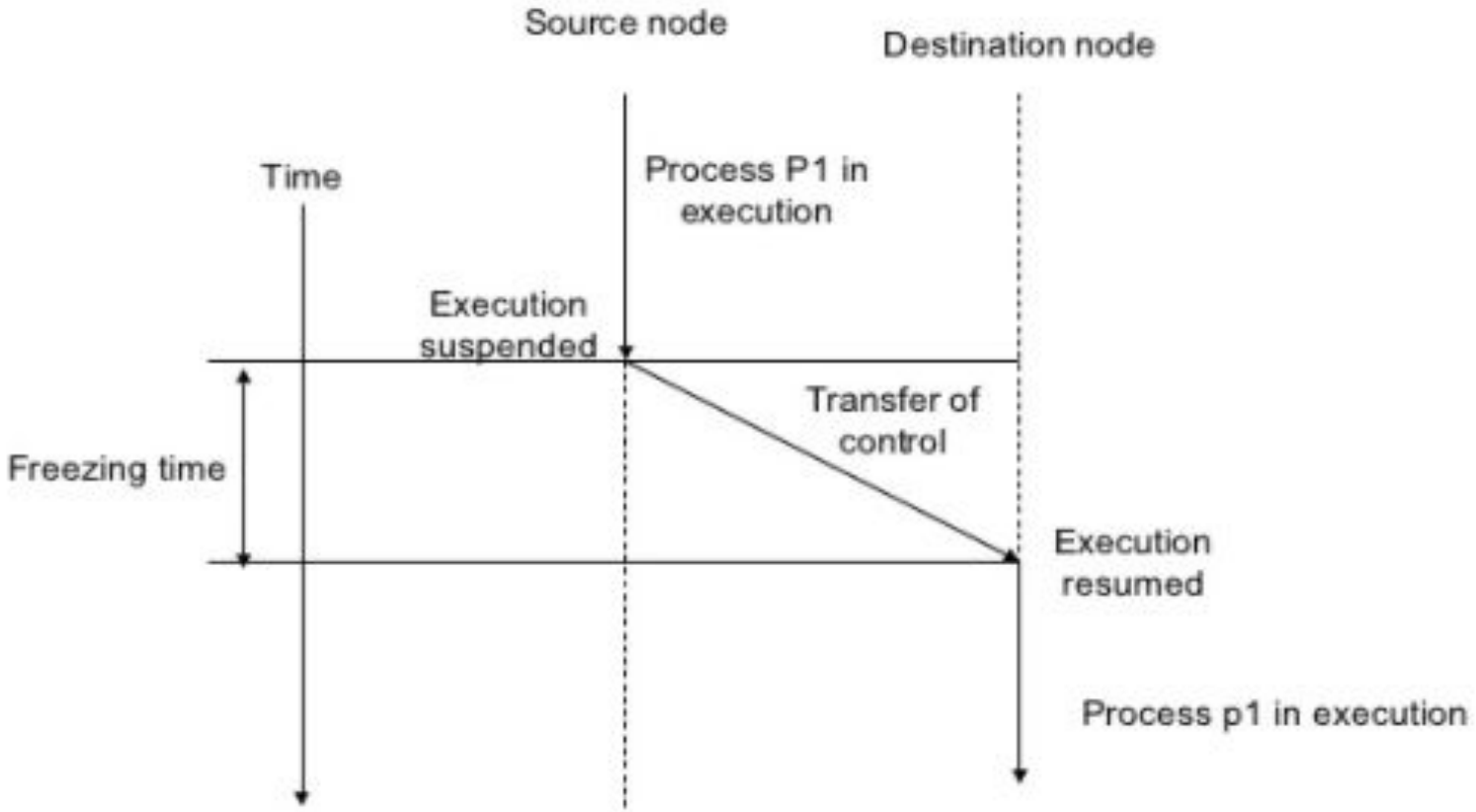
- **Process allocation** deals with the process of deciding which process should be assigned to which processor.
- **Process migration** deals with the movement of a process from its current location to the processor to which it has been assigned.
- **Threads** deals with fine-grained parallelism for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

Cont… : Flow of execution of a migration process



Cont…

- 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 coprocesses of a job

Transparency

Level of transparency:

- Object Access Level
- System Call & Interprocess Communication level

Cont… : 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.

Cont… : 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 massages meant for the migrant process.
- Handling communication between cooperating processes that have been separated as a result of process migration.

Cont… : 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

Cont… :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.

Cont… : 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.

Cont… : 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.

Cont…

- 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.

Cont… : 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.

Cont…

- 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.

- **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
- **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).

Cont…

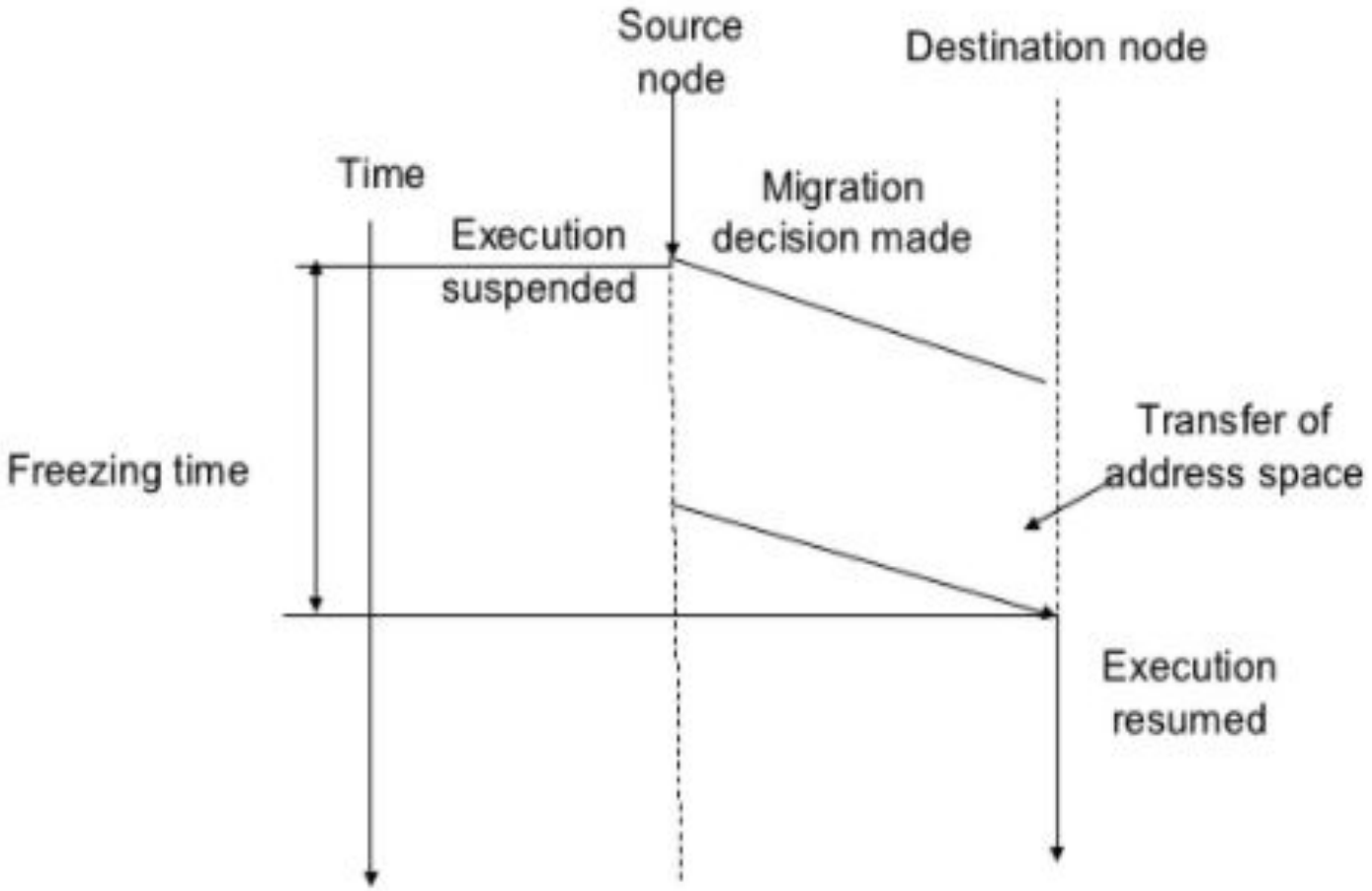
- Mechanisms for address space transfer:
  - Total freezing
  - Pretransferring
  - Transfer on reference

Cont… : 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.

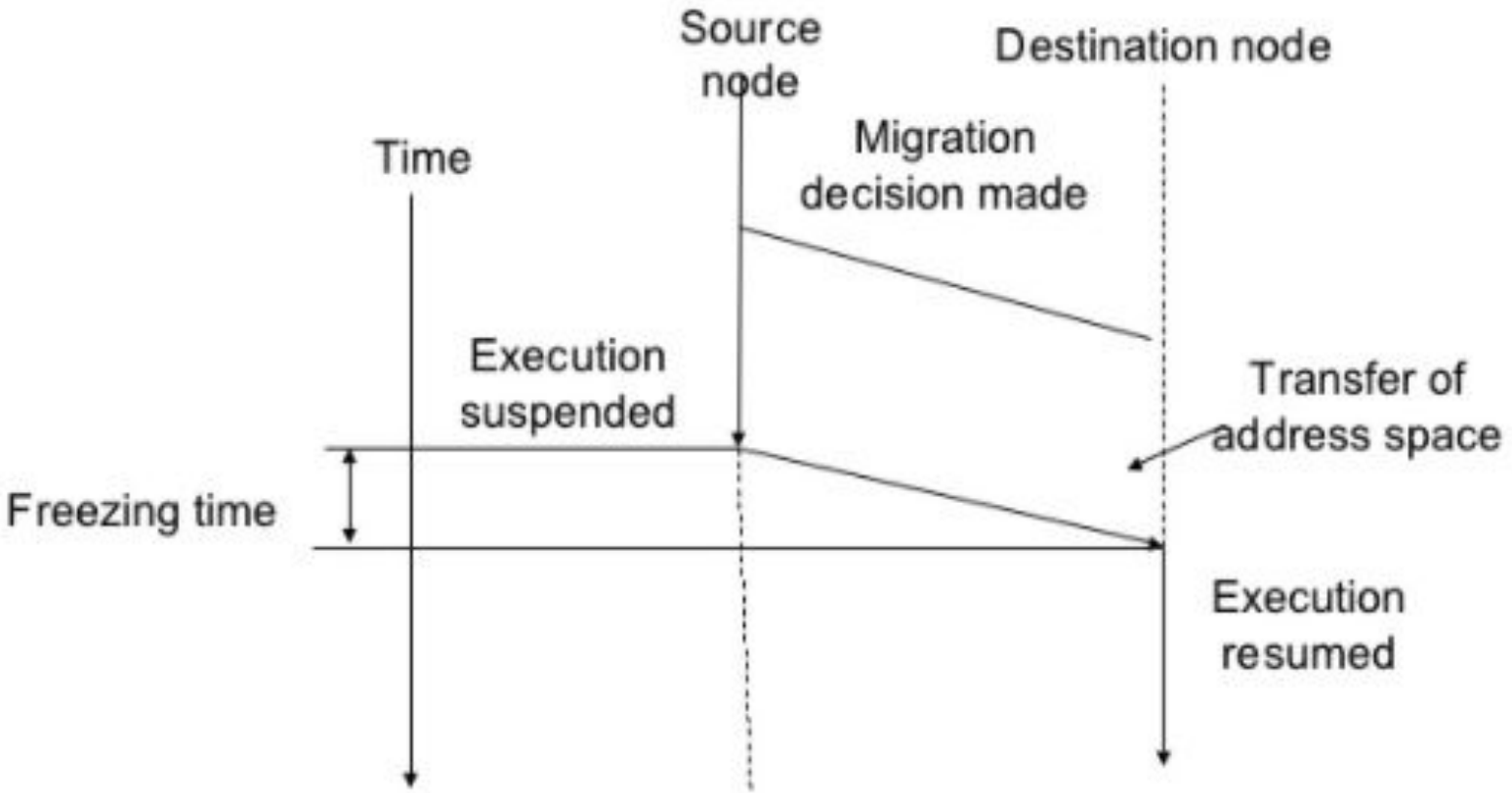
Cont…


Source node
Destination node
Time
Migration decision made
Execution suspended
Freezing time
Transfer of address space
Execution resumed

Cont … : 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.
- 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.

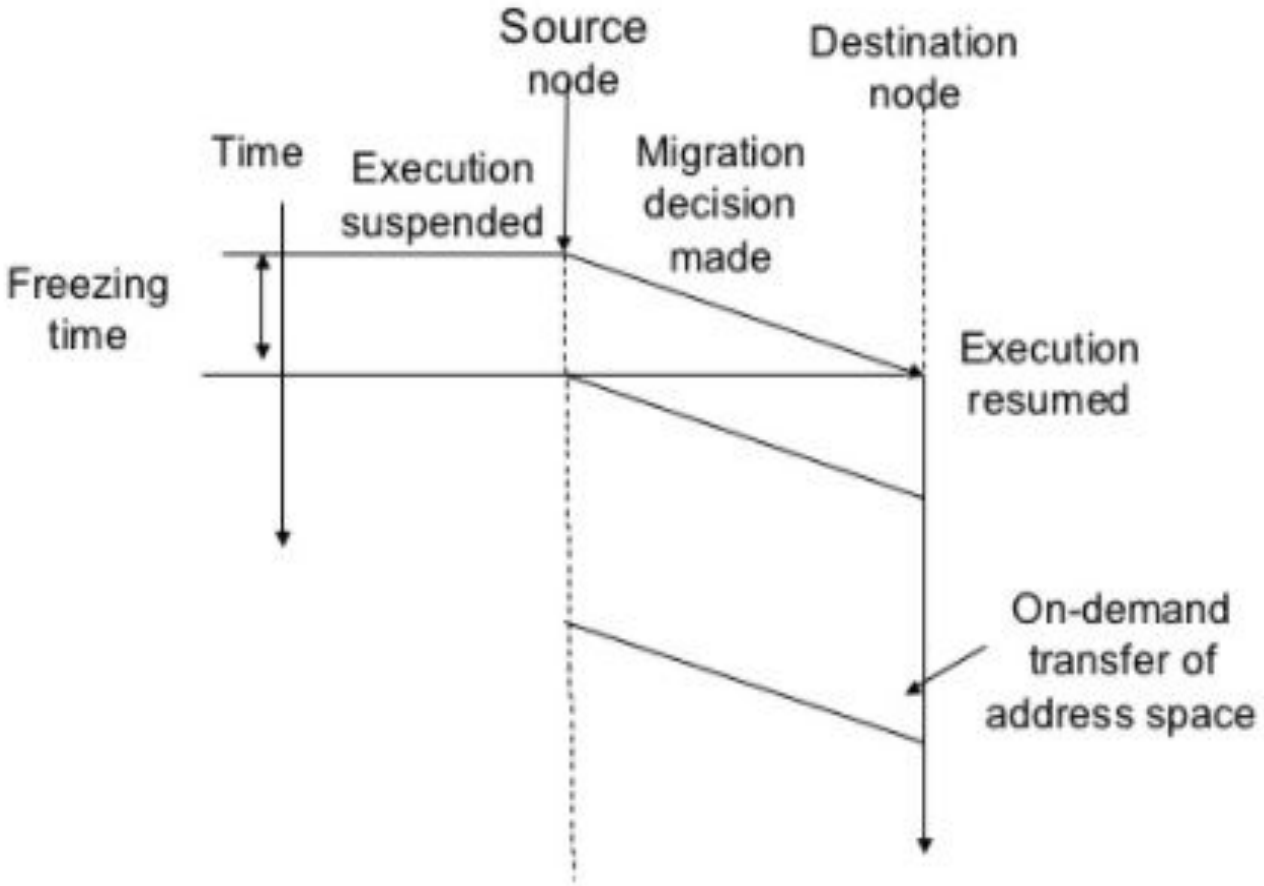
Cont…


Source node
Destination node
Time
Migration decision made
Execution suspended
Transfer of address space
Freezing time
Execution resumed

Cont… : 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.
- 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.

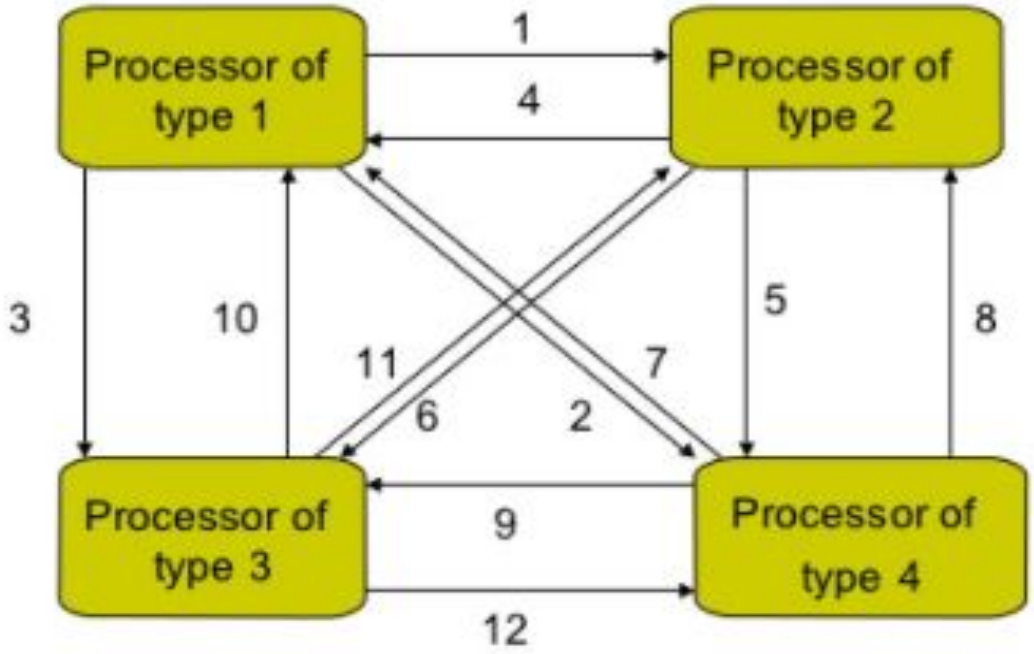
Cont…


Source node
Destination node
Time
Execution suspended
Migration decision made
Freezing time
Execution resumed
On-demand transfer of address space

Process Migration in Heterogeneous Systems

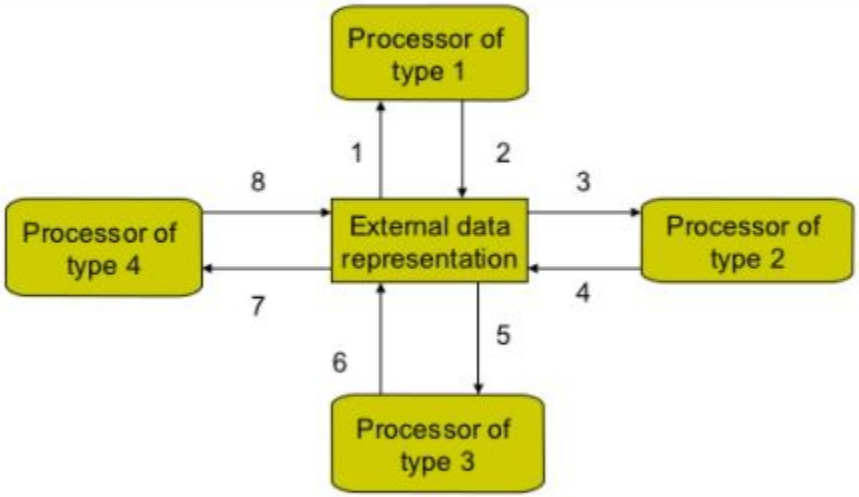
- All the concerned data must be translated from the source CPU format to the destination CPU format before it can be executed on the destination node.
- A heterogeneous system having n CPU types must have n(n-1) pieces of translation software.
- Handles problem of different data representations such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

Cont…



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

Cont…



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

Cont… : 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.

Cont…

- 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.

Cont… : Handling the Mantissa

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

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

Cont…

- Problem:
  - Loss of precision due to multiple migrations.
- Solution:
  - Remote computations can be viewed as ‘extra precision’ calculation.

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.

Cont…

- *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.

Cont…

- *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.

59

Models of Code Migration