

Process Migration - II

Models and Approaches



1

Outline

- Introduction to Process Migration
- Models and Possibilities
- Load Balancing using Process Migration
- **Migrating Resource Segment**
- Challenges for creating Global Reference

20 June 2024

2

Migrating Resource Segment



- The resource segment cannot always be simply transferred along with the other segments without being changed.
 - e.g., say a process holds a reference to a specific TCP port through which it was communicating with other (remote) processes.
 - Such reference is held in resource segment.
 - When the process migrates, it has to give up the port and request a new one at the destination.

20 June 2024

3

Migrating Resource Segment



- In some other cases, transferring a reference may not be a problem.
 - e.g., a reference to a file by means of an absolute URL will remain valid irrespective of the machine where the process that holds the URL resides.

20 June 2024

4

Process to Resource Binding



- Binding by Identifier
 - A process refers to a resource by its identifier.
- Binding by Value
 - A process refers to only the value of a resource.
- Binding by Type
 - A process indicates it needs only a resource of a specific type

20 June 2024

5

Binding by Identifier



- Here, process requires precisely the referenced resource, and nothing else.
 - e.g., a process uses a URL to refer to a specific Web site
 - e.g., a process refers to an FTP server by means of that server's Internet address.
- References to local communication endpoints also lead to a binding by identifier.

20 June 2024

6

Binding by Value



- Here, the execution of the process would not be affected if another resource provides that same value.
 - when a program relies on standard libraries, such as in C or Java.
 - the exact location of such libraries in the local file system may differ for between sites.
 - Not the specific file, but its content is required for proper execution of the process.

20 June 2024

7

Binding by Type



- This is the weakest among the three types of binding
- This is exemplified by references to local devices, such as printers, scanners and like that...

20 June 2024

8

Resource to Machine Binding



- Unattached
 - can be easily moved to another machine
 - Data-file only used by the migrating program
- Fastened
 - can be moved at a higher cost
 - local databases and complete Web sites
- Fixed
 - can't be moved to another machine
 - local devices, local communication endpoint

20 June 2024

9

Binding and Migration Strategies



- Combining three types of process-to-resource bindings, and three types of resource-to-machine bindings, leads to nine combinations
- Let's consider each of these and decide the strategies.

20 June 2024

10

Binding and Migration Strategies



- Let's consider binding by Identifier
 - When resource is unattached, move it along with the migrating code.
 - However, if the resource is shared by other processes, create a global reference, i.e., reference across machine boundaries.
 - An example of such a reference is a URL.
 - Also establish a global reference, when the resource is fastened or fixed.

20 June 2024

11

Binding and Migration Strategies



- Let's consider binding by Value
 - Consider a fixed resource.
 - Combination of a fixed resource and binding by value may occur, for example, if a process assumes that memory is shared between processes
 - Establishing a global reference in this case would mean that we need to implement distributed shared memory mechanism.
 - Obviously, This is not really a viable solution.

20 June 2024

12

Binding and Migration Strategies



- Fastened resources referred by value, could be typically runtime libraries.
 - Normally, copies of such resources are available on target machine, or should be copied before code migration takes place.
- Establishing global reference is better when huge amount of data is involved.
 - If the resource is unattached, then copy (or move) it to the new destination.
- If it is shared between processes, then establish a global reference.

13

Binding and Migration Strategies



- The last case deals with bindings by Type.
 - Irrespective of the resource-to-machine binding, the obvious solution is to rebind the process to a locally available resource of the same type.
- Only when such a resource is not available, will we need to copy or move the original one to the new destination, or establish a global reference.

20 June 2024

14

Binding and Migration Strategies



		Resource-Machine binding		
Process-Resource binding		Unattached	Fastened	Fixed
	By identifier	MV (or GR)	GR (or MV)	GR
	By value	CP (or MV, GR)	GR (or CP)	GR
	By type	RB (or MV, CP)	RB (or GR, CP)	RB (or GR)

MV: Move the resource **CP: Copy value of the resource**
GR: Create a Global Reference **RB: Rebind to local resource**

20 June 2024

15

Outline



- Introduction to Process Migration
- Models and Possibilities
- Load Balancing using Process Migration
- Migrating Resource Segment
- **Challenges for creating Global Reference**

20 June 2024

16

Challenges for creating GR



- Creating a global reference may be prohibitively expensive.
 - Consider, for example, a program that generates high-quality images for a dedicated multimedia workstation.
 - Fabricating high-quality images in real time is a compute-intensive task. So, the program may be moved to a high-performance compute server.

20 June 2024

17

Challenges for creating GR



- However, creating a global reference to the multimedia workstation means setting up of a path between the two machines.
- Besides, significant processing is involved at both ends to meet the bandwidth requirements of transferring the images.
- The net result could be that moving the program to the server is aborted, only because the cost of creating the global reference is too high.

20 June 2024

18

Challenges for creating GR



- Let's consider a process for migration that's using a local communication endpoint.
- Here, we are dealing with a fixed resource to which the process is bound by identifier.

20 June 2024

19

Challenges for creating GR



- There could be two alternate solutions.
 - One solution is to let the process set up a connection to the source machine after it has migrated and install a separate process at the source machine that simply forwards all incoming messages.
 - The main drawback of this approach is that whenever the source machine malfunctions, communication with the migrated process may fail.

20 June 2024

20

Challenges for creating GR



- The other solution is to have all processes that communicated with the migrating process, change their global references, and send messages to the new communication endpoint at the target.
- This has a very high overhead for changing the global references of every other process that communicates with the migrating process

20 June 2024

21

Thanks for your kind attention

Questions??

22