Process Migration

Motivations:

- load sharing/balancing - migrate jobs to less utilized machines
- resource locality - migrate jobs to the machine where needed resources exist
- resource sharing - make distinctive resources available to all jobs
- fault tolerance - migrate jobs away from failing machines
- system administration - migrate jobs temporarily for administrative reasons
- mobility/ubiquitous computing - migrate jobs to follow the user

Architectures:

- MPP - massively parallel processors/clusters
- NOW - networks of workstations

Steps in Migrating a Process

1. Select process to migrate
2. Detach process from source node
3. Redirect communication
4. Extract state
**Steps in Migrating a Process**

5. Create new process at destination node

6. Transfer state

7. Create forwarding reference

8. Resume migrated process

**Factors in Process Migration**

- **complexity** - added functionality imposed on operating system
- **performance** - strongly related to state transfer
- **transparency** - visibility to the process of migration
- **fault resilience** - tolerance of the migrated task to failure of its source node
- **scalability** - sustained performance with increasing system size
- **heterogeneity** - ability to incorporate diverse architectures
Trade-offs Among Factors

Mosix

upper kernel
linker
lower kernel
source node

upper kernel
linker
lower kernel
destination node

migrated process

transparency ++
complexity --
fault resilience --

State Migration Strategies

Eager (all)

source node

destination node

Eager (dirty)

source node

destination node

network paging store
State Migration Strategies

**copy-on-reference**

- Source node
- Destination node
- Network paging store

**flushing**

- Source node
- Destination node
- Network paging store

**precopy**

- Source node
- Destination node
## Barriers to Use of Migration

<table>
<thead>
<tr>
<th>Barrier</th>
<th>Reason to Overcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of applications</td>
<td>• Grid computations</td>
</tr>
<tr>
<td></td>
<td>• Huge data repositories</td>
</tr>
<tr>
<td>Lack of infrastructure</td>
<td>• Both NT and Unix are suitable for extension to support migration;</td>
</tr>
<tr>
<td></td>
<td>• both more supportive of distributed applications</td>
</tr>
<tr>
<td>Not a user requirement</td>
<td>• Mobile hardware</td>
</tr>
<tr>
<td></td>
<td>• Network speed differential</td>
</tr>
<tr>
<td></td>
<td>• Grid applications</td>
</tr>
<tr>
<td>Sociological factor</td>
<td>• Abundance reduces “ownership”</td>
</tr>
<tr>
<td></td>
<td>• Ubiquity enhances desire for “remote” access</td>
</tr>
</tbody>
</table>