COSMIC
System Software for Efficient Xeon Phi Coprocessor Sharing

COSMIC+ is NEC’s system software that enables seamless Xeon Phi coprocessor sharing. It is completely transparent to applications and all other system software components. COSMIC is useful in organizations where several users share one or more Xeon Phi-based servers. It can reduce capital cost by efficiently utilizing fewer servers.

WITHOUT COSMIC

Amy must make sure her jobs don’t oversubscribe the Phi. She runs them one after another.

Bob does not know Amy is using the Phi. He starts his job at the same time.

Job 2 must wait for Job 1 to finish

Overruns memory, and crashes

LINUX + MPSS

User directives may conflict with each other
Jobs crash when oversubscribing memory

WITH COSMIC

Amy and Bob can start Xeon Phi jobs concurrently without worrying about each other or crashes due to resource oversubscription.

Resolves conflicts between users
No oversubscription-related job crashes

LINUX + MPSS

WITHOUT COSMIC: Average device utilization around 40% *

WITH COSMIC: Average device utilization around 70% *

+COSMIC is pre-commercialization and the name is subject to change.

* Measured using 64 randomly arriving jobs on a server with 1 8GB Xeon Phi. Study reported in HPDC 2013.
**COSMIC:** middleware turbo-charges Xeon Phi stack for coprocessor sharing

### Job Management with Xeon Phi Sharing
- Eliminates Xeon Phi out-of-memory crashes by intelligent memory management of multiple tenants
- Shares coprocessors to achieve better overall utilization and minimize waiting
- Load balances devices and cores: resolves conflicts between directives of different users

### CapEX/OpEX Reduction
- Multi-tenancy allows consolidation of coprocessors to reduce system footprint and cost
- Fewer coprocessors and higher utilization per coprocessor reduces power consumption

### Simple Deployment
- Simple add-on to Xeon Phi SW stack
- Supports native or offload processing
- Completely transparent to user applications and Xeon Phi SW stack
- Requires no changes (or recompilation) of user applications

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>Without COSMIC</th>
<th>With COSMIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job scheduling</td>
<td>FCFS</td>
<td>Resource-driven scheduling improves turnaround time and throughput</td>
</tr>
<tr>
<td>Thread Oversubscription Control</td>
<td>NA</td>
<td>Pro-active thread management improves performance</td>
</tr>
<tr>
<td>Memory Oversubscription Control</td>
<td>NA</td>
<td>Pro-active memory management avoids swaps and crashes</td>
</tr>
<tr>
<td>Load balancing of coprocessor devices</td>
<td>User-managed</td>
<td>Automatic load balancing</td>
</tr>
<tr>
<td>Load balancing of cores within a device</td>
<td>User-managed</td>
<td>Automatic load balancing of jobs across device cores</td>
</tr>
<tr>
<td>Avoid conflicts between user directives</td>
<td>NA</td>
<td>Automatic</td>
</tr>
<tr>
<td>Manage interference between apps</td>
<td>NA</td>
<td>Resource-aware scheduling</td>
</tr>
<tr>
<td>Manage policies for thread and memory management, and for load balance</td>
<td>NA</td>
<td>Administrator sets and controls policies</td>
</tr>
</tbody>
</table>

**CONTACT:** Yoshiaki Hirotani (y-hirotani@aj.jp.nec.com)