Scheduling
Message Passing
(what Meta-Computing systems need to do)

CS-594
Dr Graham E. Fagg
Spring 2000
Overview

– Covers details you may have missed in David Walkers MPI / Message Passing class.

– Assumes that you know and understand MPI and PVM.
Closely coupled vs Cluster Computing

Bottom line

– MPI is better at message passing than PVM
– More complex
– Less flexible at anything else
  • I.e. it’s a message passing system not a distributed environment
Scheduling

Not built into MPI as it has no process control

– But maybe an option under MPIRUN
– Yep PVM has it all
  • user controllable pvm_spawn()
  • pvm_rm interface
    – also
      • pvm_tasker interface
      • pvm_hoster interface
Scheduling

Process Management

IBM SP2

POE / LoadLeveler Tasker

SGI Cluster

General Tasker

CRAY

MPP Tasker and PVMD

User Request

PVM GRM

CS-594 Scheduling Message Passing applications
CS-594 Scheduling Message Passing applications

Scheduling

PVM_CORM Console

SM_ADDHOST, SM_DELHOST, SM_CONFIG

User Task

SM_SPAWN

Scheduler

SM_EXECUTE, SM_SPAWN

PVMD

Daemon to daemon messages

PVMD

User Task

SM_TASK, SM_SPAWN (ACK)

SM_NOTIFY

Notifier
Scheduling

Two types to worry about

– At spawn time
  • static allocation based on the environment

– At run time
  • I.e. migration of tasks
    – system level migration
      • Special support needed (Condor)
    – User level
      • check points / restarts
  • Change work load allocated (bag of tasks)
Task allocation in PVM
I.e. pvm_spawn()

Before improving on it, had to figure out how it worked as it wasn’t random but round-robin

Aimed at using spare capacity
– what spare capacity??
What is spare (what is even machine load?)

Condor people claimed 10% utilisation for their systems
– At Reading was more like 40-60% all the time.

Load
– machine average is not a good metric but without more specific help from the kernel it would have to do.

Defined user classes and loading based on observations on the RDG system over a year...
Typical loading

Figure: 4.14 Loading on a Meteorology Department machine ssswsc2.

Figure: 4.15 Loading on a Psychology Department machine sssxsc1
Better Spawn

Added checks for load before starting.
- Based on two methods, central RPC and distributed monitor daemons
- Checked effects of this system on startup performance, and accuracy of placement.

<table>
<thead>
<tr>
<th>Slaves</th>
<th>System</th>
<th>1 (or 2*)</th>
<th>70</th>
<th>Intercept</th>
<th>Slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvm_spawn</td>
<td>seconds</td>
<td>1.246</td>
<td>4.4</td>
<td>1.2</td>
<td>0.046</td>
</tr>
<tr>
<td>central svr</td>
<td>loadsvr_heavy</td>
<td>(*a) 14.884</td>
<td>32.6</td>
<td>14.4</td>
<td>0.261</td>
</tr>
<tr>
<td>distributed</td>
<td>loadsvr_d and mon_d</td>
<td>3.669</td>
<td>17.1</td>
<td>3.5</td>
<td>0.195</td>
</tr>
</tbody>
</table>

*a. Even though the test was via a set of mean averaged results, the first point in this particular was completely out of character with the rest of the data and was discarded to allow for a more representative line fit.*
Application Performance

Comparison of various allocation schemes

Normal pvm spawn
Excess slaves method
SU pvm spawn
Linear speed-up baseline

Number of hosts counting towards final results

Figure: 4.21 Parallel speed-up of PNG Monte-Carlo simulations.
User level migration
User level migration

Space-Time: Tasks vs. Time

CS-594 Scheduling Message Passing applications
Back to load or not

The load was not always a good measure and soon the network was very heterogeneous…

Could you use a benchmark to find the real load?
Modified spawn using effective speed-up

Takes into account benchmark and loading
– Helps on heterogeneous networks

Figure: 4.31 Monte-Carlo speed-up for final version of nu_pvm_spawn.
Very advanced Schedulers

Figure 2.39: Different scheduling algorithms compared.

CS 594: Scheduling Message Passing applications
Fault Tolerance

Multiple methods
  – System level
    • from checkpoint file
  – User level
    • from data check point

  – How do we detect the failure?
    • Notify Message and/or time out