A Performance Estimator ---
PetaSIM
and its relationship to
Application Emulators

Geoffrey Fox and Yuhong Wen
Northeast Parallel Architecture Center (NPAC)
Syracuse University
May 18, 1998
PetaSIM Motivation

- PetaSIM was designed to allow “qualitative” performance estimates where in particular the design of machine is particularly easy to change
- The project will build up a suite of applications which can be used in future activities such as “Petaflop” architectures studies
- Applications are to be derived “by hand” or by automatic generation from Maryland Application Emulators
- Special attention to support of hierarchical memory machines and data intensive applications
- Support parallelism and representation at different grain sizes
- Support simulation of “pure data-parallel” and composition of linked modules
PetaSIM Interpolates Between

• “Back of the envelope” where you get good intuition for why the performance is what it is

\[ \alpha \frac{t_{\text{comm}}}{t_{\text{float}}^\frac{1}{\text{dim}}} \]

where \( \text{dim} \) is geometric dimension

Communication Overhead in classical data parallel (edge over area) with NO memory hierarchy
NO latency

• Detailed simulations
Data Movement in a 5 level Memory Hierarchy

and illustration of hierarchical machine model

Level 2 Cache

Level 1 Cache

8 Nodes in a parallel machine
PetaSIM Design

• We define an object structure for computer (including network) and data

• These object representations can be used in dynamic tools such as scheduling (cf. Legion or Globus) and seamless interfaces

• Architecture Description
  – nodeset & linkset
  – (describe the architecture memory hierarchy)

• Data Description
  – dataset & distribution (not stressed in current version)

• Application Description -- needs further refinement
  – execution script
Architecture of PetaSIM

C++ Simulator ↔ Multi-User Java Server

- Standard Java Applet Client
- Standard Java Applet Client
IBM SP2 Architecture I used in tests

View as a bunch of nodesets joined by a bunch of linksets
Each component defined as “objects” which are valuable outside PetaSIM
IBM SP2 Architecture II used in tests
Nodeset Object Structure

- **Name**: one per *nodeset* object
- **type**: choose from memory, cache, disk, CPU, pathway
- **number**: number of members of this *nodeset* in the architecture
- **grainsize**: size in bytes of each member of this *nodeset* (for memory, cache, disk)
- **bandwidth**: maximum bandwidth allowed in any one member of this *nodeset*
- **floatspeed**: CPU’s float calculating speed
- **calculate()**: method used by CPU *nodeset* to perform computation
- **cacherule**: controls persistence of data in a memory or cache
- **portcount**: number of ports on each member of *nodeset*
- **portname[]**: ports connected to *linkset*
- **portlink[]**: name of *linkset* connecting to this port
- **nodeset_member_list**: list of *nodeset* members in this *nodeset* (for *nodeset* member identification)
Linkset Object Structure

- **Name**: one per *linkset* object
- **type**: choose from updown, across
- **nodesetbegin**: name of initial *nodeset* joined by this *linkset*
- **nodesetend**: name of final *nodeset* joined by this *linkset*
- **topology**: used for across networks to specify linkage between members of a single nodeset
- **duplex**: choose from full or half
- **number**: number of members of this *linkset* in the architecture
- **latency**: time to send zero length message across any member of *linkset*
- **bandwidth**: maximum bandwidth allowed in any link of this *linkset*
- **send()**: method that calculates cost of sending a message across the *linkset*
- **distribution**: name of geometric distribution controlling this *linkset*
- **linkset_member_list**: list of *linkset* members in this *linkset* (for *linkset* member identification)
Distribution Object Structure

• Name: one per distribution object
• type: choose from block1dim, block2dim, block3dim
  – Obviously will add more choices here!
Execution Script

• Currently a few primitives which stress (unlike most languages) movement of data through hierarchies
• send DATAFAMILY from MEM-LEVEL-L to MEM-LEVEL-K
  – These reference object names for data and memory *nodesets*
• move DATAFAMILY from MEM-LEVEL-L to MEM-LEVEL-K
• Use distribution DISTRIBUTION from MEM-LEVEL-L to MEM-LEVEL-K
• compute DATAFAMILY-A, DATAFAMILY-B,… on MEM-LEVEL-L
• synchronize (synchronizes all processors --- loosely synchronous barrier)
Dataset Object Structure

- **Name:** one per *dataset* object
- **choose from** grid1dim, grid2dim, grid3dim, specifies type of *dataset*
- **bytesperunit:** number of bytes in each unit
- **floatsperunit:** update cost as a floating point arithmetic count
- **operationsperunit:** operations in each unit
- **update():** method that updates given *dataset* which is contained in a CPU nodeset and a *grainsize* controlled by last memory nodeset visited
- **transmit():** method that calculates cost of transmission of *dataset* between memory levels either communication or movement up and down hierarchy
  - Methods can use other parameters or be custom
### Jacobi Example -- Nodeset I

<table>
<thead>
<tr>
<th>name</th>
<th>Type</th>
<th>number</th>
<th>grainsize</th>
<th>portlink</th>
<th>(floatspeed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodeset_name</td>
<td>linkset_name</td>
<td>(replicated #times = # links for nodeset in first line)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nodeset_member</td>
<td>link_number</td>
<td>nodeset_member</td>
<td>linkset_member</td>
<td>(pair replicated again)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cpu</th>
<th>CPU  8 32 1 1.56116e-7</th>
</tr>
</thead>
<tbody>
<tr>
<td>mem 1</td>
<td>link1</td>
</tr>
<tr>
<td>cpu0 1</td>
<td>mem0 link10</td>
</tr>
<tr>
<td>cpu1 1</td>
<td>mem1 link11</td>
</tr>
<tr>
<td>cpu2 1</td>
<td>mem2 link12</td>
</tr>
<tr>
<td>cpu3 1</td>
<td>mem3 link13</td>
</tr>
<tr>
<td>cpu4 1</td>
<td>mem4 link14</td>
</tr>
<tr>
<td>cpu5 1</td>
<td>mem5 link15</td>
</tr>
<tr>
<td>cpu6 1</td>
<td>mem6 link16</td>
</tr>
<tr>
<td>cpu7 1</td>
<td>mem7 link17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>mem</th>
<th>Memory 8 134217728 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu</td>
<td>link1</td>
</tr>
<tr>
<td>mem0 2</td>
<td>cpu0 link10</td>
</tr>
<tr>
<td>mem1 2</td>
<td>cpu1 link11</td>
</tr>
<tr>
<td>mem2 2</td>
<td>cpu2 link12</td>
</tr>
<tr>
<td>mem3 2</td>
<td>cpu3 link13</td>
</tr>
<tr>
<td>mem4 2</td>
<td>cpu4 link14</td>
</tr>
<tr>
<td>mem5 2</td>
<td>cpu5 link15</td>
</tr>
<tr>
<td>mem6 2</td>
<td>cpu6 link16</td>
</tr>
<tr>
<td>mem7 2</td>
<td>cpu7 link17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>disks</th>
<th>Disk 8 2147483648 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>mem</td>
<td>Memory 8 134217728 2</td>
</tr>
<tr>
<td>cpu</td>
<td>link1</td>
</tr>
<tr>
<td>mem0 2</td>
<td>cpu0 link10</td>
</tr>
<tr>
<td>mem1 2</td>
<td>cpu1 link11</td>
</tr>
<tr>
<td>mem2 2</td>
<td>cpu2 link12</td>
</tr>
<tr>
<td>mem3 2</td>
<td>cpu3 link13</td>
</tr>
<tr>
<td>mem4 2</td>
<td>cpu4 link14</td>
</tr>
<tr>
<td>mem5 2</td>
<td>cpu5 link15</td>
</tr>
<tr>
<td>mem6 2</td>
<td>cpu6 link16</td>
</tr>
<tr>
<td>mem7 2</td>
<td>cpu7 link17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>disks</th>
<th>Disk 8 2147483648 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>mem</td>
<td>Memory 8 134217728 2</td>
</tr>
<tr>
<td>cpu</td>
<td>link1</td>
</tr>
<tr>
<td>mem0 2</td>
<td>cpu0 link10</td>
</tr>
<tr>
<td>mem1 2</td>
<td>cpu1 link11</td>
</tr>
<tr>
<td>mem2 2</td>
<td>cpu2 link12</td>
</tr>
<tr>
<td>mem3 2</td>
<td>cpu3 link13</td>
</tr>
<tr>
<td>mem4 2</td>
<td>cpu4 link14</td>
</tr>
<tr>
<td>mem5 2</td>
<td>cpu5 link15</td>
</tr>
<tr>
<td>mem6 2</td>
<td>cpu6 link16</td>
</tr>
<tr>
<td>mem7 2</td>
<td>cpu7 link17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>disks</th>
<th>Disk 8 2147483648 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>mem</td>
<td>Memory 8 134217728 2</td>
</tr>
<tr>
<td>cpu</td>
<td>link1</td>
</tr>
<tr>
<td>mem0 2</td>
<td>cpu0 link10</td>
</tr>
<tr>
<td>mem1 2</td>
<td>cpu1 link11</td>
</tr>
<tr>
<td>mem2 2</td>
<td>cpu2 link12</td>
</tr>
<tr>
<td>mem3 2</td>
<td>cpu3 link13</td>
</tr>
<tr>
<td>mem4 2</td>
<td>cpu4 link14</td>
</tr>
<tr>
<td>mem5 2</td>
<td>cpu5 link15</td>
</tr>
<tr>
<td>mem6 2</td>
<td>cpu6 link16</td>
</tr>
<tr>
<td>mem7 2</td>
<td>cpu7 link17</td>
</tr>
</tbody>
</table>
Jacobi Example -- Nodeset II

ctl1 Pathway 8 0 2
disks link3 ctl2 link4
ctl10 2 d0 link30 ctl20 link40
ctl11 2 d1 link31 ctl21 link41
ctl12 2 d2 link32 ctl22 link42
ctl13 2 d3 link33 ctl23 link43
ctl14 2 d4 link34 ctl24 link44
ctl15 2 d5 link35 ctl25 link45
ctl16 2 d6 link36 ctl26 link46
ctl17 2 d7 link37 ctl27 link47

ctl2 Pathway 8 0 3
mem link2 ctl1 link4 network link5
ctl20 3 mem0 link20 ctl10 link40 network0 link50
ctl21 3 mem1 link21 ctl11 link41 network0 link51
ctl22 3 mem2 link22 ctl12 link42 network0 link52
ctl23 3 mem3 link23 ctl13 link43 network0 link53
ctl24 3 mem4 link24 ctl14 link44 network0 link54
ctl25 3 mem5 link25 ctl15 link45 network0 link55
ctl26 3 mem6 link26 ctl16 link46 network0 link56
ctl27 3 mem7 link27 ctl17 link47 network0 link57

network Switch 1 0 1
ctl2 link5
network0 8 ctl20 link50 ctl21 link51 ctl22 link52
ctl23 link53 ctl24 link54 ctl25 link55 ctl26 link56
ctl27 link57
Jacobi Example -- Linkset I

Excerpt from Linkset Definitions
1) name Type Duplex number latency bandwidth
2) nodesetbegin nodesetend
3) linkset_member nodeset_member_begin nodeset_member_end

link1 Updown Full 8 0.0 524288000
mem cpu
link10 mem0 cpu0
link11 mem1 cpu1
link12 mem2 cpu2
link13 mem3 cpu3
link14 mem4 cpu4
link15 mem5 cpu5
link16 mem6 cpu6
link17 mem7 cpu7

link2 Updown Full 8 0.0 83886080
mem ctl2
link20 mem0 ctl20
link21 mem1 ctl21
link22 mem2 ctl22
link23 mem3 ctl23
link24 mem4 ctl24
link25 mem5 ctl25
link26 mem6 ctl26
link27 mem7 ctl27

link3 Updown Full 8 2.0e-4 8388608
ctl1 disks
link30 ctl10 d0
link31 ctl11 d1
link32 ctl12 d2
link33 ctl13 d3
link34 ctl14 d4
link35 ctl15 d5
link36 ctl16 d6
link37 ctl17 d7
Jacobi Example -- Linkset II

- Excerpt from Linkset Definitions:
  - name Type Duplex number latency bandwidth
  - nodesetbegin nodesetend
  - linkset_member nodeset_member_begin nodeset_member_end

<table>
<thead>
<tr>
<th>link4</th>
<th>Updown</th>
<th>Full</th>
<th>8 0.0</th>
<th>83886080</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctl1</td>
<td>ctl2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>link40</td>
<td>ctl10</td>
<td>ctl20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link41</td>
<td>ctl11</td>
<td>ctl21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link42</td>
<td>ctl12</td>
<td>ctl22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link43</td>
<td>ctl13</td>
<td>ctl23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link44</td>
<td>ctl14</td>
<td>ctl24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link45</td>
<td>ctl15</td>
<td>ctl25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link46</td>
<td>ctl16</td>
<td>ctl26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link47</td>
<td>ctl17</td>
<td>ctl27</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>link5</th>
<th>Updown</th>
<th>Full</th>
<th>4.0e-5</th>
<th>83886080</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctl2</td>
<td>network</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>link50</td>
<td>ctl20</td>
<td>network0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link51</td>
<td>ctl21</td>
<td>network0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link52</td>
<td>ctl22</td>
<td>network0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link53</td>
<td>ctl23</td>
<td>network0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link54</td>
<td>ctl24</td>
<td>network0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link55</td>
<td>ctl25</td>
<td>network0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link56</td>
<td>ctl26</td>
<td>network0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link57</td>
<td>ctl27</td>
<td>network0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Jacobi Example

Dataset Definition is very simple (by design):

name type size bytesperunit floatperunit operationperunit

Jacobi Grid2dim 1000000 4 4 10
Jacobi Example -- Execution Script I

compute Jacobi on cpu4
Move Jacobi from d1 to ctl11
move Jacobi from ctl11 to ctl21
move Jacobi from ctl21 to mem1
move Jacobi from mem1 to ctl21
move Jacobi from ctl21 to network0
move Jacobi from network0 to ctl20
move Jacobi from ctl20 to mem0
move Jacobi from mem0 to cpu0
compute Jacobi on cpu0
move Jacobi from mem1 to ctl21
move Jacobi from ctl21 to network0
move Jacobi from network0 to ctl22
move Jacobi from ctl22 to mem2
move Jacobi from mem2 to cpu2
compute Jacobi on cpu2
move Jacobi from mem1 to ctl21
move Jacobi from ctl21 to network0
move Jacobi from network0 to ctl25
move Jacobi from ctl25 to mem5
move Jacobi from mem5 to cpu5
compute Jacobi on cpu5

Move Jacobi from d0 to ctl10
move Jacobi from ctl10 to ctl20
move Jacobi from ctl20 to mem0
move Jacobi from mem0 to ctl20
move Jacobi from ctl20 to network0
move Jacobi from network0 to ctl21
move Jacobi from ctl21 to mem1
move Jacobi from mem1 to cpu1
compute Jacobi on cpu1
move Jacobi from mem0 to ctl20
move Jacobi from ctl20 to network0
move Jacobi from network0 to ctl24
move Jacobi from ctl24 to mem4
move Jacobi from mem4 to cpu4

Also a simpler data parallel version
Jacobi Example -- Execution Script II

Move Jacobi from d5 to ctl15
move Jacobi from ctl15 to ctl25
move Jacobi from ctl25 to mem5
move Jacobi from mem5 to ctl25
move Jacobi from ctl25 to network0
move Jacobi from network0 to ctl21
move Jacobi from ctl21 to mem1
move Jacobi from mem1 to cpu1
compute Jacobi on cpu1
move Jacobi from mem6 to ctl26
move Jacobi from ctl26 to network0
move Jacobi from network0 to ctl27
move Jacobi from ctl27 to mem7
move Jacobi from mem7 to cpu7
compute Jacobi on cpu7

Move Jacobi from d7 to ctl17
move Jacobi from ctl17 to ctl27
move Jacobi from ctl27 to mem7
move Jacobi from mem7 to ctl27
move Jacobi from ctl27 to network0
move Jacobi from network0 to ctl23
move Jacobi from ctl23 to mem3
move Jacobi from mem3 to cpu3
compute Jacobi on cpu3
move Jacobi from mem7 to ctl27
move Jacobi from ctl27 to network0
move Jacobi from network0 to ctl26
move Jacobi from ctl26 to mem6
move Jacobi from mem6 to cpu6
compute Jacobi on cpu6
synchronize
# Pathfinder Performance Estimation Results (Architecture II)

<table>
<thead>
<tr>
<th>Examples</th>
<th>IBM SP2 Running Time</th>
<th>Petasim Estimation Time (appl)</th>
<th>Petasim Execution Time (appl)</th>
<th>Petasim Estimation Time (max)</th>
<th>Petasim Execution Time (max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc16path</td>
<td>26.48</td>
<td>25.38</td>
<td>3.41</td>
<td>21.12</td>
<td>3.4</td>
</tr>
<tr>
<td>proc32path</td>
<td>35.38</td>
<td>34.68</td>
<td>12.18</td>
<td>29.11</td>
<td>12.24</td>
</tr>
<tr>
<td>proc64path</td>
<td>43.80</td>
<td>43.07</td>
<td>46.01</td>
<td>36.86</td>
<td>45.95</td>
</tr>
<tr>
<td>proc100path</td>
<td>60.34</td>
<td>58.89</td>
<td>149.91</td>
<td>52.68</td>
<td>149.87</td>
</tr>
<tr>
<td>path.vary(4_60)</td>
<td>22.99</td>
<td>21.46</td>
<td>13.26</td>
<td>15.45</td>
<td>13.26</td>
</tr>
<tr>
<td>path.vary(8_56)</td>
<td>13.32</td>
<td>13.37</td>
<td>11.86</td>
<td>10.08</td>
<td>11.85</td>
</tr>
<tr>
<td>path.vary(16_48)</td>
<td>10.85</td>
<td>10.63</td>
<td>11.24</td>
<td>9.12</td>
<td>11.25</td>
</tr>
<tr>
<td>path.vary(32_32)</td>
<td>10.93</td>
<td>10.57</td>
<td>10.15</td>
<td>9.89</td>
<td>10.18</td>
</tr>
<tr>
<td>path.vary(48_16)</td>
<td>12.79</td>
<td>12.28</td>
<td>8.13</td>
<td>11.84</td>
<td>8.14</td>
</tr>
<tr>
<td>path.vary(56_8)</td>
<td>18.20</td>
<td>17.45</td>
<td>7.19</td>
<td>17.18</td>
<td>7.20</td>
</tr>
<tr>
<td>path.vary(60_4)</td>
<td>30.02</td>
<td>28.93</td>
<td>6.52</td>
<td>28.68</td>
<td>6.47</td>
</tr>
</tbody>
</table>
Pathfinder Performance Estimation Results

Pathfinder Estimation Results

- Running on SP2
- PetaSIM Estimation Time
- PetaSIM Execution Time

Processor number
Pathfinder Estimation Results (with I/O ratio changed)

- Running on SP2
- PetaSIM Estimation Time
- PetaSIM Execution Time

I/O Ratio vs. Estimation Time

- Data points for different I/O ratios showing trends.
# PetaSIM Estimation Results (Architecture I)

<table>
<thead>
<tr>
<th>Examples</th>
<th>IBM SP2 Running Time</th>
<th>Petasim Estimation Time</th>
<th>Petasim Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc15titan(9K)</td>
<td>114</td>
<td>105.23</td>
<td>7.12</td>
</tr>
<tr>
<td>proc15titan(27K)</td>
<td>347</td>
<td>318.51</td>
<td>21.61</td>
</tr>
<tr>
<td>proc32titan(27K)</td>
<td>N/A</td>
<td>316.07</td>
<td>41.69</td>
</tr>
<tr>
<td>proc15path(9K)</td>
<td>166</td>
<td>159.45</td>
<td>8.49</td>
</tr>
<tr>
<td>proc15path(27K)</td>
<td>497</td>
<td>484.51</td>
<td>25.7</td>
</tr>
<tr>
<td>proc32path(55K)</td>
<td>N/A</td>
<td>710.48</td>
<td>78.99</td>
</tr>
</tbody>
</table>
Interface between Emulator and PetaSIM
Part of Pathfinder Application

Excerpt from Nodeset Definitions
name Type number grainsize portlink (floatspeed)
nodeset_name linkset_name
nodeset_member link_number nodeset_member linkset_member

cpu CPU 8 32 1 1.530000e-06
bus1 link2
cpu0 1 bus10 link20
cpu1 1 bus11 link21
cpu2 1 bus12 link22
cpu3 1 bus13 link23
cpu4 1 bus14 link24
cpu5 1 bus15 link25
cpu6 1 bus16 link26
cpu7 1 bus17 link27

mem Memory 8 134217728 1
bus1 link1
mem0 1 bus10 link10
mem1 1 bus11 link11
mem2 1 bus12 link12
mem3 1 bus13 link13
mem4 1 bus14 link14
mem5 1 bus15 link15
mem6 1 bus16 link16
mem7 1 bus17 link17

.................
Interface between Emulator and PetaSIM -- Pathfinder

Excerpt from Linkset Definitions:

<table>
<thead>
<tr>
<th>name</th>
<th>Type</th>
<th>Duplex</th>
<th>number</th>
<th>latency</th>
<th>bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>linkset_member</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nodeset_begin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nodeset_end</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- link1 Updown Full 8 0.000000e+00 134217728
  - bus1 mem
  - link10 bus10 mem0
  - link11 bus11 mem1
  - link12 bus12 mem2
  - link13 bus13 mem3
  - link14 bus14 mem4
  - link15 bus15 mem5
  - link16 bus16 mem6
  - link17 bus17 mem7

- link2 Updown Full 8 0.000000e+00 134217728
  - bus1 cpu
  - link20 bus10 cpu0
  - link21 bus11 cpu1
  - link22 bus12 cpu2
  - link23 bus13 cpu3
  - link24 bus14 cpu4
  - link25 bus15 cpu5
  - link26 bus16 cpu6
  - link27 bus17 cpu7
Interface between Emulator and PetaSIM -- Pathfinder

Excerpt from Dataset Definitions:

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>size</th>
<th>bytesperunit</th>
<th>floatperunit</th>
<th>operationperunit</th>
</tr>
</thead>
<tbody>
<tr>
<td>data0</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data1</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data2</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data3</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data4</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data5</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data6</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data7</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data8</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data9</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data10</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data11</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data12</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data13</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>data14</td>
<td>grid2dim</td>
<td>46920</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

...
Interface between Emulator and PetaSIM Pathfinder

Excerpt from Execution Script:

move data0 from disks0 to bus30
move data0 from bus30 to bus20
move data0 from bus20 to bus10
move data0 from bus10 to mem0
move data0 from mem0 to bus10
move data0 from bus10 to bus20
move data0 from bus20 to nwa0
move data0 from nwa0 to network0
move data0 from network0 to nwa11
move data0 from nwa11 to bus211
move data0 from bus211 to bus111
move data0 from bus111 to mem11
move data0 from mem11 to bus111
move data0 from bus111 to cpu11
compute data0 on cpu11
## Titan Estimation Results

**(Architecture II)**

<table>
<thead>
<tr>
<th>Examples</th>
<th>IBM SP2 Running Time</th>
<th>Petasim Estimation Time (appl)</th>
<th>Petasim Execution Time (appl)</th>
<th>Petasim Estimation Time (max)</th>
<th>Petasim Execution Time (max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc16titan</td>
<td>66.51</td>
<td>59.95</td>
<td>13.89</td>
<td>59.52</td>
<td>13.84</td>
</tr>
<tr>
<td>proc32titan</td>
<td>94.16</td>
<td>101.30</td>
<td>50.65</td>
<td>100.48</td>
<td>50.43</td>
</tr>
<tr>
<td>proc64titan</td>
<td>112.42</td>
<td>156.52</td>
<td>209.44</td>
<td>154.85</td>
<td>209.63</td>
</tr>
<tr>
<td>proc100titan</td>
<td>154.90</td>
<td>226.56</td>
<td>594.49</td>
<td>224.37</td>
<td>595.71</td>
</tr>
<tr>
<td>proc16fixed</td>
<td>66.83</td>
<td>59.95</td>
<td>13.93</td>
<td>59.52</td>
<td>13.82</td>
</tr>
<tr>
<td>proc32fixed</td>
<td>47.16</td>
<td>50.32</td>
<td>25.24</td>
<td>49.92</td>
<td>25.18</td>
</tr>
<tr>
<td>proc64fixed</td>
<td>27.84</td>
<td>37.63</td>
<td>52.37</td>
<td>37.26</td>
<td>52.23</td>
</tr>
<tr>
<td>proc100fixed</td>
<td>24.37</td>
<td>33.56</td>
<td>94.98</td>
<td>33.31</td>
<td>94.50</td>
</tr>
</tbody>
</table>
# VMScope Performance Estimation Results (Architecture II)

<table>
<thead>
<tr>
<th>Examples</th>
<th>IBM SP2 Running Time</th>
<th>Petasim Estimation Time (appl)</th>
<th>Petasim Execution Time (appl)</th>
<th>Petasim Estimation Time (max)</th>
<th>Petasim Execution Time (max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc16vmscope</td>
<td>17.13</td>
<td>14.59</td>
<td>0.25</td>
<td>14.58</td>
<td>0.26</td>
</tr>
<tr>
<td>proc32vmscope</td>
<td>21.35</td>
<td>17.80</td>
<td>0.67</td>
<td>17.79</td>
<td>0.68</td>
</tr>
<tr>
<td>proc64vmscope</td>
<td>21.43</td>
<td>18.02</td>
<td>2.11</td>
<td>18.00</td>
<td>2.12</td>
</tr>
<tr>
<td>proc100vmscope</td>
<td>18.50</td>
<td>15.45</td>
<td>4.76</td>
<td>15.43</td>
<td>4.84</td>
</tr>
</tbody>
</table>

7/14/98  petasimmay98  http://www.npac.syr.edu
PetaSIM Current Progress Summary

- Architecture Description (*nodeset* & *linkset*)
- Application Description (*dataset* & execution script)
- Link to Application Emulators
- Jacobi hand-written example
- Pathfinder, Titan, VMScope real applications (Generated by UMD’s Emulator)
- Easy modified Architecture and Application description
- Fast and relatively Accurate performance estimation (PetaSIM running on single processor)
- Java applet based user Interface
- About 5000 lines of C++ and 4000 lines of Java (client and server)
Possible Future Work

• Richer set of applications using standard benchmarks and DoD MSTAR

• Relate object model to those used in “seamless interfaces” / metacomputing i.e. to efforts to establish (distributed) object model for computation

• Review very simple execution script -- should we add more complex primitives or regard “application emulators” as this complex script

• Binary format (“compiled PetaSIM”) of architecture and application description ( ASCII format will make execution script very large)
  – Translation tool from ASCII format to binary format (to retain the friendly user interface)

• Upgrade performance evaluation model

• Run performance simulation in parallel (i.e. PetaSIM running on multi-processors)