Message Driven Programming with S-Net
Methodology and Performance

Frank Penczek
S.A. Herhut, S.-B. Scholz, A. Shafarenko, J. Yang, C.Y. Chen, N. Bagherzadeh, C. Grelck
What is S-Net
Coordination & Concurrency Engineering
What is S-Net
Applications as Streaming Networks
What is S-Net
Basic Building Blocks
What is S-Net
Basic Building Blocks

{A, B, ...}

\[ \alpha \rightarrow \beta \]

{X, Y, ...}

```
net myNet {
  box MyBox((A,B) -> (X,Y));
} connect MyBox;
```
Raytracing Application
The Principle
Raytracing Application
...in S-Net

- Read Scene
- Solve
- Gen Image
Raytracing Application
...in S-Net

{scn_name} → Read Scene → Solve → Gen Image → {scene} → {raster} → {img}
Raytracing Application
...in S-Net

```plaintext
net raytrace {
    box ReadScene( (scn_name) -> (scene));
    box Solve( (scene) -> (raster));
    box GenImage( (raster) -> (img));
}
}```
Raytracing Application
Parallelised Approach
Raytracing Application
Parallelised Approach in S-Net
net split {
  box Solve( (scene) -> (raster));
} connect Solve;

net split {
  box Solve( (scene) -> (raster));
} connect Solve!
Raytracing Application
Parallelised Approach

Read Scene → Solve → Solve → Merge → Gen Image
Raytracing Application
Extended Read Scene

{scn_name, <num_tasks>}

{scene}

{raster}

{scene}

{raster, <task>, <num_tasks>}

{raster}
S-Net Features
Flow Inheritance

\{\text{scene}, <\text{task}>, <\text{num}\_\text{tasks}>\}

\{\text{raster}, <\text{task}>, <\text{num}\_\text{tasks}>\}

\{<\text{task}>, <\text{num}\_\text{tasks}>\}
Raytracing Application
Parallelised Approach

- Read Scene
- Solve
- Solve
- Merge
- Gen Image
S-Net Combinators

Iteration

{accu}

{sub_res}

Merge

{result}
S-Net Combinators
Star Combinator

```plaintext
net merge {
  box Merge( (accu, sub_res) -> (accu) |
              (result));
  } connect Merge*{result};
```
S-Net Features
Synchronisation

net merge {
  box Merge( (accu, sub_res) -> (accu) | (result));
}

connect ( [[{accu}, {sub_res} | ] .. Merge)*{result};
Raytracing Application
Runtime on Shared Memory

Number of Cores

Runtime in Seconds

- Read Scene
- Solve
- Merge
- Gen Image
Raytracing Application
Speed-Up vs. original C Code

<table>
<thead>
<tr>
<th>Number of Cores</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpeedUp</td>
<td>0.97</td>
<td>1.27</td>
<td>2.13</td>
<td>3.62</td>
<td>6.22</td>
<td>12.11</td>
</tr>
</tbody>
</table>
S-Net Features
Splitting using Distributed Memory

net split {
  box Solve( (scene) -> (raster));
}
} connect Solve!<task>;

net split {
  box Solve( (scene) -> (raster));
}
} connect Solve!@<task>;
Raytracing Application
Runtime on Cluster

- Read Scene
- Solve
- Solve
- Merge
- Gen Image

Bar chart showing runtime in seconds for different numbers of nodes.

- 1 node: 400 seconds
- 2 nodes: 350 seconds
- 4 nodes: 300 seconds
- 6 nodes: 250 seconds
- 8 nodes: 200 seconds
- 10 nodes: 150 seconds
- 12 nodes: 100 seconds
Raytracing Application
Speed-Up vs. original C Code and MPI

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>MPI/SN</th>
<th>Seq/SN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.99</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>1.04</td>
<td>1.23</td>
</tr>
<tr>
<td>4</td>
<td>1.06</td>
<td>2.06</td>
</tr>
<tr>
<td>6</td>
<td>1.04</td>
<td>3.19</td>
</tr>
<tr>
<td>8</td>
<td>1.05</td>
<td>3.57</td>
</tr>
<tr>
<td>10</td>
<td>1.06</td>
<td>4.13</td>
</tr>
<tr>
<td>12</td>
<td>1.06</td>
<td>4.79</td>
</tr>
</tbody>
</table>
Utilising Resources Efficiently
Dynamic Load Scheduling

Problem

Sub-Prob
Sub-Prob
Sub-Prob
Sub-Prob

Sub-Prob

Sub-Prob

Sub-Prob

Sub-Prob

Sub-Prob
Utilising Resources Efficiently
Dynamic Load Scheduling

Problem

Sub-Prob

Sub-Prob

Sub-Prob

Sub-Prob
Utilising Resources Efficiently
Dynamic Load Scheduling
Utilising Resources Efficiently
Dynamic Load Scheduling
Utilising Resources Efficiently
Dynamic Load Scheduling in S-Net

Read Scene

Solve

{cpu} {sub}

Merge

{<cpu>}

University of Hertfordshire
Utilising Resources Efficiently
Dynamic Load Scheduling in S-Net

Read Scene

Solve

Merge

{scn_name, <num_tasks>, <num_cpus>}

{cpu}  {sub}

{scene, <task>, <num_tasks>, <cpu>}
{scene, <task>, <num_tasks>}

University of Hertfordshire
S-Net Combinators

Choice

{scene, <task>, <cpu>}

{raster, <task>}
| {<cpu>}

Solve

{cpu} {scn}

{<cpu>}
| {scene, <task>}

{<cpu>, scene}

net solving {
  ...
} connect (Solve | [[[{cpu},{scene}]]]) *{raster};
Raytracing Application
Parallelised Approach

Read Scene → Solve (cpu, sub) → Merge

{<cpu>}

Raytracing Application
Runtime in Shared Memory

Read Scene

{cpu}

{sub}

Solve

Solve

Solve

{<cpu>}

Merge

RunQme in Seconds

Number of Cores

Runtime in Seconds

University of Hertfordshire
Raytracing Application
Runtime on Cluster + Shared Memory

- Read Scene
- Solve
- Merge

Enter to exit.
S-Net on one Slide
Combinators

A .. B

A*{p}

A | B

A!<t>
Thank you!
End.
S-Net Tool-Chain
Overview

S-Net Compiler

S-Net Module

S-Net Runtime

Box Interface

Executable

Box Compiler

Library / Objects