Adjustable Contiguity of Run-Time Task Allocation in Networked Many-Core Systems

Mohammad Fattah, Pasi Liljeberg, Juha Plosila, Hannu Tenhunen
Outline

- Introduction.
- Motivation.
- Contiguity Adjustable Square Allocation.
- Results.

Source code: http://users.utu.fi/mofana/CASqA.html
Introduction
Many-Core Systems
- Connected via NoC

Applications: set of communicating tasks.

Applications enter and leave the system at Run-Time.

**Question:** How to map an Application?

**Answer:** Contiguously
- Less network power
- Less network congestion
Motivation
Forced to limit the system to **only contiguous** allocations.

**Store-and-Forward**

- **Execution**
- **Turn-Around**

**Wormhole**

- Non-contiguous allocation: **higher throughput** with **more network power**

**Execution**
- **Turn-Around**

Source code: http://users.utu.fi/mofana/CASqA.html
What about the middle ground?

Does the execution/turnaround time steadily increase/decrease?

How to quantify and implement the contiguity?

Source code: http://users.utu.fi/mofana/CASqA.html
Contiguity Adjustable Square Allocation

Source code: http://users.utu.fi/mofana/CASqA.html
0.0 ≤ α ≤ 1.0:
  o level of “desired contiguity” / “allowed contiguous solutions”.

α adaptively determines the maximum exploration radius.

For an application with |T| tasks:

1. Start from a first node
2. until |T| available nodes are found:
   o Increase the exploration radius (allocation dispersion)

Source code: http://users.utu.fi/mofana/CASqA.html
\( R_{\text{max}}:\)  
- Maximum allowed exploration radius.  
- Initial value:

\[
R_{\text{max}} = \left\lfloor \frac{\sqrt{|T|}}{2} \right\rfloor
\]

\( \tau \) (expansion threshold):
- Initial value: \( |T| \times \alpha \).  
- if less than \( \tau \) nodes are needed: \( R_{\text{max}} \) ++
- \( \tau \times= \alpha \) on each increase.

\( \alpha=0 \rightarrow \tau=0 \rightarrow \) don’t explore beyond initial \( R_{\text{max}} \).
\( \alpha=1 \rightarrow \tau=|T| \rightarrow \) expand exploration until allocated.

Source code: http://users.utu.fi/mofana/CASqA.html
A 9-task application enters the system with the following run-time configuration.

Initial

\[ R_{max} = \left\lfloor \frac{\sqrt{9}}{2} \right\rfloor = 1 \]

If \( \alpha \geq 0.56 \)

\( \Rightarrow \tau > 5 \)

\( \alpha = 0.5, \text{ initial } \tau = 4.5. \)

Source code: http://users.utu.fi/mofana/CASqA.html
ICEB is defined as a unified metric to model both congestion and energy dissipation of the network.

ICEB metric is used by CASqA to arrange task placement within a selected area.

First node of allocation is selected using our previous approach, SHiC.

The source code can be downloaded at: : http://users.utu.fi/mofana/CASqA.html
Experimental Results
- A 16x16 mesh interconnect.
- Cycle-Accurate System-C network model.
- Message-Passing.
- Communication-Intensive applications with 4 to 35 tasks.
- Random sequence of application.
- Upon a mapping failure, new *first nodes* are selected until the current application being mapped.

Source code: http://users.utu.fi/mofana/CASqA.html
The same throughput can be achieved with smaller $\alpha$. 

Source code: http://users.utu.fi/mofana/CASqA.html
Results - Network Metrics

- Latency
- Energy

37% reduction

Source code: http://users.utu.fi/mofana/CASqA.html
## Results - Comparison

<table>
<thead>
<tr>
<th>Mapping</th>
<th>NMRD</th>
<th>$E_{norm}$</th>
<th>$L_{avg}$</th>
<th>%Congestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASqA^{0.5}</td>
<td>1.13</td>
<td>1</td>
<td>41</td>
<td>30.52</td>
</tr>
<tr>
<td>CASqA^{0.5**}</td>
<td>1.13</td>
<td>1.25</td>
<td>48.23</td>
<td>37.7</td>
</tr>
<tr>
<td>CoNA</td>
<td>1.69</td>
<td>1.48</td>
<td>50.32</td>
<td>38.4</td>
</tr>
<tr>
<td>NN</td>
<td>1.97</td>
<td>1.56</td>
<td>55.53</td>
<td>41.98</td>
</tr>
</tbody>
</table>

** arbitrary task placement within the selected area (without ICEB).
Investigation of the middle ground of the spectrum between strictly contiguous and unlimited non-contiguous allocation.

The communication distance is yet a matter in todays on-chip networks.

CASqA provides the tune to adjust the power/throughput optimal dispersion level, $\alpha$.

The optimum $\alpha$ depends on the application specifications.

Source code: http://users.utu.fi/mofana/CASqA.html