Contention-aware Mapping and Scheduling Optimization for NoC-based MPSoCs

Rongjie Yan, Institute of Software Chinese Academy of Sciences,
Yupeng Zhou, Yige Yan, Minghao Yin, Northeast Normal University,
Anyu Cai, Changwen Li, Beijing University of Technology

Motivation

- We consider spatial and temporal aspects of communication to avoid contention in Network-on-Chip (NoC)-based architectures.

Problem Definition

- We consider applications represented with task graphs (TGs). For example, the model shown is a task graph with 4 tasks and 5 dependency relations (edges).

Hybrid Search Algorithm

- Task Clustering (TC)
- Capacity Constrained Cluster Refinement (CR)
- Spiral Mapping (SM)
- Encoding
- Initialization
- Genetic Process
- Pareto Local Search

Constraint formulation

- We provide a flexible constraint formulation for NoC-based mapping and scheduling in the format of logical formulas.

Experimental Results

- Deal with the mapping and scheduling for NoC-based MPSoCs from a practical view, i.e., optimizing three objectives: makespan, energy consumption and contention probability.

Comparison between MOHA and various methods

<table>
<thead>
<tr>
<th>Case</th>
<th>[T]</th>
<th></th>
<th>MOHA</th>
<th>NSGAII</th>
<th>CPLEX(MLP)</th>
<th>CPLEX(CP)</th>
<th>Z3</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-m</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2(ε)+2(ε)</td>
<td>687.63</td>
<td>2(ε)</td>
<td>53.62</td>
</tr>
<tr>
<td>5-p</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2(ε)+2(ε)</td>
<td>687.63</td>
<td>2(ε)</td>
<td>53.62</td>
</tr>
<tr>
<td>7-m</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2(ε)+2(ε)</td>
<td>687.63</td>
<td>2(ε)</td>
<td>53.62</td>
</tr>
<tr>
<td>7-p</td>
<td>7</td>
<td>6</td>
<td>2</td>
<td>2(ε)+2(ε)</td>
<td>2(ε)</td>
<td>2(ε)</td>
<td>138.96</td>
</tr>
<tr>
<td>10-m</td>
<td>10</td>
<td>9</td>
<td>1</td>
<td>1(ε)+1(ε)</td>
<td>247.59</td>
<td>2(ε)</td>
<td>19.83</td>
</tr>
</tbody>
</table>

Comparison between MOHA and NSGAII on large instances

MOHA can find all the pareto fronts on small-scale benchmarks and outperforms NSGAII on large-scale instances.