CaSMap: Agile Mapper for Reconfigurable Spatial Architectures by Automatically Clustering Intermediate Representations and Scattering Mapping Process

Xingchen Man, Jianfeng Zhu, Guihuan Song, Shouyi Yin, Shaojun Wei, Leibo Liu*
Beijing National Research Center for Information Science and Technology (BNRist)
School of Integrated Circuits, Tsinghua University, Beijing, China
*Corresponding author: liulb@tsinghua.edu.cn
Domain-Specific Architecture (DSA)

Artificial Intelligence

Internet of Things

Information Security

Signal Processing


This slide has been designed using resources from Flaticon.com
DSA Candidate—Reconfigurable spatial architecture (RSA)

RSA features:

- **Myriad** programmable resources: compute and communication resources
- **Volatile** ISAs: different ISAs in different versions/domains
- **NP-Complete** Mapping problem\(^1\): difficult to find the optimal orchestration

Compiler is necessary:

- Avoid expensive manual mapping
- Adapt to new algorithms/architectures
- Helpful in architecture design space exploration

---

LLVM is good for CPUs:
- Infrastructural: lower the barrier
- Retargetable: decouple compiler design
- Modular: avoid reinventing the wheels

Post-Moore compiler challenges:
- Hardware diversity
- Parallelism excavation
- Algorithm complexity

LLVM Three Phase Compiler Framework
Reconfigurable spatial architecture (RSA) compiler framework

Mapper:
- Place all the operations
- Maintain all the dependencies
- **Optimize** performance/resource

Combinatorial optimization problem:
- Objective: \( \min F(x) \)
- Constraints: \( s.t. \, G(x) \geq 0 \)
  \( x \in D \)
# State-of-The-Art Mapping Methods

<table>
<thead>
<tr>
<th>Categories</th>
<th>Methods</th>
<th>Hyper-parameters</th>
<th>Heuristic Type</th>
<th>Representative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic searching</td>
<td>Meta-heuristic (e.g., simulated annealing)</td>
<td>Cooling factor, Transition, etc.</td>
<td>Auto</td>
<td>DRESC&lt;sup&gt;[1]&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Hyper-heuristic (machine learning)</td>
<td>NN structure, Training, etc.</td>
<td>Auto</td>
<td>LISA&lt;sup&gt;[2]&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Heuristic (architecture-specific)</td>
<td>None</td>
<td>Expertise</td>
<td>TAEM&lt;sup&gt;[3]&lt;/sup&gt;</td>
</tr>
<tr>
<td>Systematic backtracking</td>
<td>Integer linear programming (ILP)</td>
<td>None</td>
<td>None</td>
<td>AA-ILP&lt;sup&gt;[4]&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Satisfiability modulo theory (SMT)</td>
<td>None</td>
<td>None</td>
<td>SMT-based Mapper&lt;sup&gt;[5]&lt;/sup&gt;</td>
</tr>
</tbody>
</table>


<sup>[2]</sup> Zhaoying Li, et al. LISA: Graph Neural Network based Portable Mapping on Spatial Accelerators. In 2022 IEEE HPCA


State-of-The-Art Mapping Methods

<table>
<thead>
<tr>
<th>Categories</th>
<th>Methods</th>
<th>Merits</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic searching</td>
<td>Meta-heuristic (e.g., simulated annealing)</td>
<td>Generality, Good speed (loose constraints)</td>
<td>Unstable solution quality</td>
</tr>
<tr>
<td></td>
<td>Hyper-heuristic (machine learning)</td>
<td>Fast speed, Good solution quality</td>
<td>Tedious tuning and training</td>
</tr>
<tr>
<td></td>
<td>Heuristic (architecture-specific)</td>
<td>Fast speed, Good solution quality</td>
<td>High development cost</td>
</tr>
<tr>
<td>Systematic backtracking</td>
<td>Integer linear programming (ILP)</td>
<td>Generality, Completeness, Optimal solution quality</td>
<td>Slow speed</td>
</tr>
<tr>
<td></td>
<td>Satisfiability modulo theory (SMT)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## RSA Mapper Framework Requirements

<table>
<thead>
<tr>
<th>Features</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portability</td>
<td>Take low-level HW/SW IRs as input</td>
</tr>
<tr>
<td>Ease of development</td>
<td>Utilize constraint programming</td>
</tr>
<tr>
<td>Nonexpert heuristic</td>
<td>Obtain heuristics by analyzing IRs</td>
</tr>
<tr>
<td>Modularity</td>
<td>Decompose heuristics and mapping algorithm</td>
</tr>
<tr>
<td>High quality</td>
<td>Systematic approach</td>
</tr>
</tbody>
</table>

### RSA Mapper Framework Outline

- **Library**
  - Strategy\_1
  - ..., Strategy\_m

- **Low-Level HW IR**
  - Library

- **Low-Level SW IR**
  - Constraint-based Mapping Algorithm
  - Heuristics
Only using single-level HW/SW IRs is the key limiting factor in prior systematic approaches. Clustering IRs into multi-level IRs can mitigate the complexity of the mapping problem.

Opportunities in RSA mapping:
- Communications are mostly localized due to depopulated interconnection. → Cluster HW IR
- Operations on the critical path have a great effect on performance. → Cluster SW IR

Benefits of clustering IRs:
- Obtain heuristics in the form of multi-level HW/SW IRs.
- Scatter the mapping process into multi-level decomposed mapping models.
Contributions

- An agile mapper, CaSMap, which has multi-level self-contained incremental mapping models with multi-level HW/SW IRs.

- Rich strategies of clustering HW/SW IRs and scattering mapping algorithms for designing an agile mapper.

- 80.5% problem scale reduction and 83x speedup compared with the state-of-the-art waterfall ILP mapper.
CaSMap Overview

- Input: original mapping model based on low-level HW/SW IRs
CaSMap Overview

- Clustering IRs: obtain heuristics in the form of multi-level HW/SW IRs
CaSMap Overview

- Multi-level mapping models
CaSMap Overview

- Iterative execution
CaSMap—Hardware IR clustering

Connection unit clustering
- Cluster routing nodes according to connectivity
- **Eliminate redundancy** of low-level HW IR

Functional unit clustering
- Cluster functional units based on communication distance
  \[
  \forall u, v \in F(H), CD(u, v) = \min_{p=uv} \text{delay}(p)
  \]
- Communication distance between any two FUs in the same HW cluster is smaller than the threshold (CDTH)
- Can be modeled as a **graph coloring problem**

High-level hardware IRs can be obtained via a two-step clustering
CaSMap—Software IR clustering

Operator fusion
- Clustering **bottleneck** operator combination (e.g., multiply accumulate (MAC))

Throughput-sensitive clustering
- Clustering operations on the **critical path**
- Separate loosely connected operations

Temporal clustering
- Scheduling approach (e.g., modulo scheduling)

Throughput-sensitive clustering (a) is able to achieve a better mapping than arbitrary clustering (c)
CaSMap—IR Clustering

Multi-level IRs:
- New variables, constraints, and objective functions
- Coupling constraints

Example:
- Original constraints: $\forall p \in Ops, \sum_{q \in FUs} f_{p,q} = 1$ (1)
- Coupling constraints: $\forall p \in Ops, c_{p,j} = \sum_{q \in j} f_{p,q}$ (2)
- New constraints: $\forall p \in Ops, \sum_{j \in FUCs} c_{p,j} = 1$ (3)

New variables brought by IR clustering
CaSMap—Multi-level mapping model

Original waterfall mapping model with single-level IRs:
- Tons of variables are strongly coupled with each other

Multi-level IRs can decompose the waterfall model:
- High-level: mapping clusters
- Low-level: mapping details

Fig. 5 Original waterfall mapping model
CaSMap—Systematic backtracking

Searching space and exploration:
- **Entire** mapping space
- ILP solver

Iterative execution:
- Change IR clustering strategies
- Constraint reduction or conversion

An example of iterative execution when meeting failures
Evaluation—Setup

Input software:
- Microbench\cite{1}, ExPRESS\cite{2}, Embench-IoT\cite{3}

Target architecture topology:
- MorphoSys\cite{4}, HReA\cite{5}, HyCUBE\cite{6}, REVEL\cite{7}

Software IR:
- Software IR: dataflow graph (DFG)

Hardware IR:
- Modulo routing resource graph (MRRG)\cite{8}

---

**DFG features**

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>Vertices</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>accum</td>
<td>21</td>
<td>25</td>
</tr>
<tr>
<td>cap</td>
<td>42</td>
<td>47</td>
</tr>
<tr>
<td>conv2</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>conv3</td>
<td>28</td>
<td>31</td>
</tr>
<tr>
<td>mac</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>mac2</td>
<td>40</td>
<td>46</td>
</tr>
<tr>
<td>matmult</td>
<td>26</td>
<td>28</td>
</tr>
<tr>
<td>mults1</td>
<td>34</td>
<td>38</td>
</tr>
<tr>
<td>mults2</td>
<td>42</td>
<td>48</td>
</tr>
<tr>
<td>sum</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>arf</td>
<td>54</td>
<td>86</td>
</tr>
<tr>
<td>h2v2</td>
<td>68</td>
<td>71</td>
</tr>
<tr>
<td>mulf</td>
<td>97</td>
<td>108</td>
</tr>
</tbody>
</table>

---

**Target topologies and hardware IR clustering example**

Evaluation—Problem scale and time cost

Problem scale metric:
- Number of nonzero elements in coefficient matrix

Compared with waterfall ILP mapper [1] (WILP):
- 80.5% less problem scale
- 83 times faster (8348.3 s → 99.8 s)

Clustering time cost:
- Less than 0.1%

Problem scale and time cost with different clustering strategies on different architecture topologies

Evaluation—Mapping quality

Initiation interval (II) and routing resource occupation (RRO):

- Identical **optimal IIs** to waterfall ILP mapper [1] (WILP)
- 5.5% more and 25.5% less RRO compared with WILP* and simulated annealing mapper [2]

---


*Relative Optimal Gap = 20%
Evaluation—Generality

Time cost, initiation interval (II), and routing resource occupation (RRO):

- 4.5x faster than waterfall ILP mapper [1]
- Identical optimal IIs to WILP
- 6.1% more compared with WILP *
- No more than two iterations


*Relative Optimal Gap = 20%
Conclusion

Challenge
- Compilation for RSAs is difficult because of the mapping problem’s NP-Completeness.
- Heuristic and systematic approach suffer from low quality and high time cost respectively.

Our method
- We present CaSMap, an agile mapper framework for various hardware and software of RSAs.
- We propose strategies for clustering hardware and software IRs and setting up a multi-level mapping model as the foundation of the framework.
CaSMap: Agile Mapper for Reconfigurable Spatial Architectures by Automatically Clustering Intermediate Representations and Scattering Mapping Process

Xingchen Man, Jianfeng Zhu, Guihuan Song, Shouyi Yin, Shaojun Wei, Leibo Liu*
Beijing National Research Center for Information Science and Technology (BNRist)
School of Integrated Circuits, Tsinghua University, Beijing, China
*Corresponding author: liulb@tsinghua.edu.cn

THANK YOU!

This presentation and recording belong to the authors. No distribution is allowed without the authors' permission.