HETEROGENEOUS MULTIPROCESSOR MAPPING FOR REAL-TIME STREAMING SYSTEMS

Jing Lin, Akshaya Srivasta, Prof. Andreas Gerstlauer, and Prof. Brian L. Evans

Department of Electrical and Computer Engineering
The University of Texas at Austin

May 27, 2011
Introduction

- Synchronous data flow (SDF) models
  - Static schedule: 1-2-3-4-3-4-3-4-5
  - Model for many real-time streaming applications, which desire high throughput and low latency
- Multiprocessor system-on-chips (MPSoCs)
Problem Definition

- Mapping SDF models to MPSoCs

- Partition:

- Schedule:
Problem Definition

- Mapping SDF models to MPSoCs

- Partition:

- Schedule:

Period = 1 / Throughput
Latency = (End of the n-th exec. of Sink) – (Start of the n-th exec. of Source)
## Prior Work

<table>
<thead>
<tr>
<th>Publication</th>
<th>General SDF</th>
<th>Processor Heterogeneity</th>
<th>Objectives</th>
<th>Solution Form</th>
<th>Main Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Lee1987]</td>
<td>Yes</td>
<td>No</td>
<td>Throughput</td>
<td>Single solution</td>
<td>Linear programming</td>
</tr>
<tr>
<td>[Bha1996]</td>
<td>Yes</td>
<td>No</td>
<td>Throughput</td>
<td>Single solution</td>
<td>Linear programming</td>
</tr>
<tr>
<td>[Zhu2009]</td>
<td>Yes</td>
<td>Yes</td>
<td>Buffer size</td>
<td>Schedule only**</td>
<td>Constraint programming</td>
</tr>
<tr>
<td>[Bon2010]</td>
<td>No*</td>
<td>Yes</td>
<td>Throughput</td>
<td>Single solution</td>
<td>Graph-based solution</td>
</tr>
<tr>
<td>[Zit2000]</td>
<td>Yes</td>
<td>No</td>
<td>Multiple</td>
<td>Pareto front***</td>
<td>Evolutionary algorithm</td>
</tr>
</tbody>
</table>

* Homogeneous SDF graph only
** Partition is assumed to be given
*** A set of points that are Pareto optimal
Our Goal

• Mapping general SDF to heterogeneous MPSoCs
• Multi-objective optimization
  • Throughput
  • Latency
  • Processor cost (e.g. price, area)

A multi-objective optimization framework that jointly optimizes throughput, latency and processor cost for general multiprocessor SDF mapping.
Global Optimization

- An integer linear programming (ILP) Model
- Optimize partitioning and scheduling simultaneously
- Objective:
  \[
  \text{Minimize } \lambda_1 \cdot \text{Period} + \lambda_2 \cdot \text{Latency} + \lambda_3 \cdot \text{Cost}
  \]
- Constraints
  - SDF semantics
  - Static partitioning
  - Execution time profile
  - Sequential execution of actors mapped to the same processor
  - Stable periodic schedule
Global Optimization

- **Actor - \( i \); Processor - \( j \); Time - \( t \in \{0,1,\ldots T\} \)
- **Decision variables**
  - \( S_i(t), E_i(t) \): Number of started/ended executions of actor \( i \) up to time \( t \)
  - \( A_{ij} \): Indicator of whether actor \( i \) is bound to processor \( j \)
  - \( \text{start}(t) \): Indicator of the start of stable periodic phase
- **Constraints**
  - Execution precedence: \( c_{i_1,i_2} S_{i_2}(t) \leq p_{i_1,i_2} E_i(t) + o_{i_1,i_2} \)
  - Execution time: \( S_i(t) = \sum_j A_{ij} E_i(t + d_{ij}) \)
  - Sequential execution: \( \sum_j A_{ij} (S_i(t) - E_i(t)) \leq 1 \)
  - Periodicity of the schedule: \( W_i(T) - \sum_t W_i(t) \text{start}(t) = n_i \sum_j A_{ij} d_{ij} \)
  - Definition of objectives: \( \text{Period} = T - \sum_t t \cdot \text{start}(t); \text{Cost} = \sum_j \text{Alloc}_j \cdot p c_j \)
  - \( \text{Latency} = \sum_t (U(t) - V(t)) + \sum_j A_{ij} d_{ij} + (S_1(T) - S_i(T)) \cdot \text{Period} \)

  \( \text{Time interval between Source's 1st start and Sink's 1st end in the periodic phase} \)

  \( \text{Difference in iteration numbers} \)
Global Optimization

- **Actor - i; Processor - j; Time - t ∈ \{0,1,…T\}
- **Decision variables
  - $S_i(t)$, $E_i(t)$: Number of started/ended executions of actor $i$ up to time $t$
  - $A_{ij}$: Indicator of whether actor $i$ is bound to processor $j$
  - $start(t)$: Indicator of the start of stable periodic phase
- **Constraints
  - Execution precedence: $c_{i_1,i_2}S_{i_2}(t) \leq p_{i_1,i_2}E_{i_1}(t) + o_{i_1,i_2}$
  - Execution time: $S_i(t) = \sum_j A_{ij}E_i(t + d_{ij})$
  - Sequential execution: $\sum_j A_{ij}(S_i(t) - E_i(t)) \leq 1$
  - Periodicity of the schedule: $W_i(T) - \sum_t W_i(t)\cdot start(t) = n_i \sum_j A_{ij}d_{ij}$
  - Definition of objectives: $\text{Period} = T - \sum_t t \cdot start(t); \text{Cost} = \sum_j \text{Alloc}_j \cdot p_{c_j}$

Express indicator functions with two constraints

Linearize product terms: add one variable and three constraints

Express indicator functions with two constraints

Time interval between Source’s 1\textsuperscript{st} start and Sink’s 1\textsuperscript{st} end in the periodic phase

Difference in iteration numbers
Global Optimization

- Actor - $i$; Processor - $j$; Time - $t \in \{0,1,\ldots,T\}$

- **Decision variables**
  - $S_i(t), E_i(t)$: Number of started/ended executions of actor $i$ up to time $t$
  - $A_{ij}$: Indicator of whether actor $i$ is bound to processor $j$
  - $\text{start}(t)$: Indicator of the start of stable periodic phase

- **Constraints**
  - Execution precedence: $c_{i_1,i_2} s_{i_2}(t) \leq p_{i_1,i_2} E_{i_1}(t) + o_{i_1,i_2}$
  - Execution time: $S_i(t) = \sum_j A_{ij} E_i(t + d_{ij})$
  - Sequential execution: $\sum_j A_{ij} (S_i(t) - E_i(t)) \leq 1$
  - Periodicity of the schedule: $W_i(T) - \sum_t W_i(t) \cdot \text{start}(t) = n_i \sum_j A_{ij} d_{ij}$
  - Definition of objectives: $\text{Period} = T - \sum_t t \cdot \text{start}(t)$; $\text{Cost} = \sum_j \text{Alloc}_j \cdot \text{pc}_j$

**Latency** = $\sum_t \left( U(t) - V(t) \right) + \sum_j A_{ij} d_{ij} + (S_1(T) - S_i(T)) \cdot \text{Period}$

- Time interval between Source’s 1$^\text{st}$ start and Sink’s 1$^\text{st}$ end in the periodic phase
- Difference in iteration numbers

Express indicator functions with two constraints

Linearize product terms: add one variable and three constraints
Heuristic Optimization

• Maximum throughput partition
  • For fixed partition, the best throughput is determined by the critical processor
  • Empirically, the best throughput is achievable given long enough startup phase and proper scheduling
  • Just optimize partitioning for the best throughput and cost

• Two-stage optimization process

  Stage I: Partitioning
  Maximize throughput and minimize cost

  Stage II: Scheduling
  Minimize latency under throughput constraint

• Throughput and cost are prioritized over latency
Heuristic Optimization

- Two ILPs

- Multi-Objective Evolutionary Algorithm (MOEA)
  - The population consists of a subset of all possible partitions
  - Converge to a set of multi-objective optimal partitions, i.e. Pareto front
Heuristic Optimization

- MOEA with Scheduling ILP

<table>
<thead>
<tr>
<th>Generate a single solution</th>
<th>Generate a Pareto front</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput/Cost Computation</td>
<td>Throughput/Cost Computation</td>
</tr>
<tr>
<td>Two-Objective MOEA</td>
<td>Three-Objective MOEA</td>
</tr>
<tr>
<td>Throughput/Cost 2-D Pareto Front</td>
<td>Throughput/Cost/Latency 3-D Pareto Front</td>
</tr>
<tr>
<td>Scheduling ILP</td>
<td>Scheduling ILP</td>
</tr>
<tr>
<td>Best Mapping</td>
<td>Best Mapping</td>
</tr>
</tbody>
</table>
## Global vs. Heuristic Optimization

<table>
<thead>
<tr>
<th></th>
<th>Global Optimization</th>
<th>Heuristic Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optimality</strong></td>
<td>Global optimal</td>
<td>Sub-optimal</td>
</tr>
<tr>
<td><strong>Computational</strong></td>
<td>NP hard</td>
<td>MOEA: $O(M^2 \log M)$ per iteration*</td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td></td>
<td>Scheduling ILP: polynomial time</td>
</tr>
<tr>
<td><strong>Design Space</strong></td>
<td>Generate a single mapping;</td>
<td>Generate a single mapping</td>
</tr>
<tr>
<td><strong>Exploration</strong></td>
<td>Generate a Pareto-front by fine tuning the weights</td>
<td>or a three-objective Pareto front</td>
</tr>
</tbody>
</table>

* M is the population size
Experimental Results

• Programming Tools
  • ILP: CPLEX Concert Technology for C++
  • MOEA: MOGALib framework in C++

• Run-time comparison
  • random cyclic/acyclic SDF graphs mapped to 3 processors
Experimental Results

• Design space exploration for an MP3 decoder

• Convergence to Pareto front
  • ~1 hour execution time

*Solution of global ILP with $\lambda_1 = 0.8$ and $\lambda_2 = 0.2*
Conclusion

- Mapping SDF models onto heterogeneous MPSoCs
- Global ILP
- Heuristics by MOEA
- Generate a single mapping or a Pareto front
References

Thank you for your attention!