Modeling Instruction Placement on a Spatial Architecture

Martha Mercaldi
Steven Swanson, Andrew Petersen, Andrew Putnam, Andrew Schwerin
Mark Oskin and Susan Eggers

University of Washington
Why Spatial Architectures?

Scalability?
Complexity?
Power?
Why Spatial Architectures?

Scalability
- Short wires

Complexity
- Simple, replicated unit

Power
- Turn off unneeded tiles

What should execute where?
Instruction Placement

On a spatial architecture, where should execution occur?
Why model placement?

Enable exploration -
- of placements
- of microarchitecture

Guide for development of placement algorithms [ASPLOS 06]
Talk Outline

Motivation

WaveScalar Background

Sub-model Construction & Evaluation

Unified Model Construction & Evaluation
WaveScalar Processor

Dataflow execution model

Tiled microarchitecture
Processing Element

5-stage pipeline
Holds 64 instructions
1 execution unit
1 cycle operand latency
PEs in a Pod

2 Processing Elements

Execution stages linked
Domain

4 Pods
Crossbar interconnect
EXE to EXE: 4 cycles
Cluster

4 Domains
Network switch
Local L1 Data Cache
Store Buffer
EXE to EXE: 7 cycles
WaveScalar Processor
Application Execution
Talk Outline

Motivation

WaveScalar Background

Sub-model Construction & Evaluation
  Methodology
  Example

Unified Model Construction & Evaluation
Model Inputs & Output

- mapping: instruction -> PE
- execution profile
- architectural parameters

MODEL

mapping quality
Internal Model Structure

- Mapping: instruction -> PE
- Execution profile
- Architectural parameters
- Operand Latency
- Resource Contention
- Coherence Overhead
- etc.
- Mapping quality
Sub-model Methodology

How might placement effect performance?

- Operand Latency
- Resource Contention
- Network Bandwidth
- Coherence overhead
Sub-model Methodology

How much does X effect performance?

1. Generate a sampling of placements

2. Run idealized simulation
   (To measure contribution of X, idealize everything except X)

3. Contribution = Variance in IPC / Average IPC
Sub-model Methodology

For a placement, what is the cost wrt. X?

Takes three inputs
- placement
- profile
- microarchitectural parameters

Produces cost for X
Sub-model Methodology

How good is the submodel?

Measure correlation between sub-model output to simulated IPC

(Still using idealized simulator)

Perfect correlation: -1.0
Sub-model Example: Operand Latency

Producer-consumer distance determines operand latency

In simulator, idealized:
- Interconnect bandwidth
- Execution resources
- Data & instruction caches

Contribution
- $\text{Contribution} = \frac{\text{Variance(IPC)}}{\text{Average(IPC)}}$
- $= 0.84$
Sub-model Example: Operand Latency

Cost depends on type of communication

- Intra-pod
  - Latency = 0
- Intra-domain
  - Latency\(_{i,j}\) = 4
- Inter-domain
  - Latency\(_{i,j}\) = 7 + ||C\(_i\) - C\(_j\)||

\(T_{i,j}\) = dynamic number of operand tokens
Latency = \(\sum_{i,j} (T_{i,j} \times \text{Latency}\(_{i,j}\))\)
### Sub-model Example: Operand Latency

<table>
<thead>
<tr>
<th>Sub-model</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>art</td>
<td>-0.90</td>
</tr>
<tr>
<td>equake</td>
<td>-0.92</td>
</tr>
<tr>
<td>fft</td>
<td>-0.88</td>
</tr>
<tr>
<td>gzip</td>
<td>-0.93</td>
</tr>
<tr>
<td>lu</td>
<td>-0.86</td>
</tr>
<tr>
<td>mcf</td>
<td>-0.80</td>
</tr>
<tr>
<td>twolf</td>
<td>-0.89</td>
</tr>
<tr>
<td>vpr</td>
<td>-0.84</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>-0.88</strong></td>
</tr>
</tbody>
</table>

**Graph:**
- X-axis: Operand Latency Metric
- Y-axis: Simulated Operand Latency IPC (art)
- Data points for various sub-models indicating the correlation between operand latency andIPC.
Sub-model Summary

<table>
<thead>
<tr>
<th>Sub-model</th>
<th>Contribution (sub-model importance)</th>
<th>Correlation (sub-model quality)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand latency</td>
<td>0.84</td>
<td>-0.88</td>
</tr>
<tr>
<td>Interconnect bandwidth</td>
<td>0.01</td>
<td>--</td>
</tr>
<tr>
<td>PE contention</td>
<td>1.21</td>
<td>-0.76</td>
</tr>
<tr>
<td>Cache coherence overhead</td>
<td>0.34</td>
<td>-0.84</td>
</tr>
</tbody>
</table>
Talk Outline

Motivation

WaveScalar Background

Sub-model Construction & Evaluation

Unified Model Construction & Evaluation
Sub-model Unification

<table>
<thead>
<tr>
<th>Contribution (sub-model importance)</th>
<th>Correlation (sub-model quality)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand Latency</td>
<td>0.84 0.35</td>
</tr>
<tr>
<td>Interconnect Bandwidth</td>
<td>0.01 0.00</td>
</tr>
<tr>
<td>PE Contention</td>
<td>1.21 0.51</td>
</tr>
<tr>
<td>Cache coherence overhead</td>
<td>0.34 0.14</td>
</tr>
</tbody>
</table>

TotalScore =

0.35 \times \text{OperandLatencyScore} +
0.51 \times \text{PeContentionScore} +
0.14 \times \text{CoherenceOverheadScore}
Internal Model Structure

mapping: instruction -> PE

execution profile

architectural parameters

Operand Latency x 0.35
Resource Contention x 0.51
Coherence Overhead x 0.14

mapping quality
Unified Model: Evaluation

The diagram illustrates the Unified Model's evaluation using a scatter plot. The x-axis represents the Model Estimate, while the y-axis shows IPC (Scaled). Each data point corresponds to a different benchmark, with different symbols and colors for each benchmark:

- art: -0.81
- equake: -0.98
- fft: -0.77
- gzip: -0.98
- lu: -0.93
- mcf: -0.86
- twolf: -0.92
- vpr: -0.97

The plot shows the correlation between the IPC (Scaled) and the Model Estimate, with a linear trend indicating the effectiveness of the Unified Model.
Unified Model: Evaluation

How does model predict performance of new application?

- Use cross-validation
- Split data into training and test sets
  - Example:
    - Training: all benchmarks except fft
    - Test: fft
- Derive model from training data
- Measure correlation on test data
## Combined Model: Evaluation

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Test Set</th>
<th>Correlation Coeff. (on test set)</th>
</tr>
</thead>
<tbody>
<tr>
<td>all except art</td>
<td>art</td>
<td>-0.76</td>
</tr>
<tr>
<td>all except equake</td>
<td>equake</td>
<td>-0.89</td>
</tr>
<tr>
<td>all except fft</td>
<td>fft</td>
<td>-0.74</td>
</tr>
<tr>
<td>all except gzip</td>
<td>gzip</td>
<td>-0.83</td>
</tr>
<tr>
<td>all except lu</td>
<td>lu</td>
<td>-0.77</td>
</tr>
<tr>
<td>all except mcf</td>
<td>mcf</td>
<td>-0.95</td>
</tr>
<tr>
<td>all except twolf</td>
<td>twolf</td>
<td>-0.76</td>
</tr>
<tr>
<td>all except vpr</td>
<td>vpr</td>
<td>-0.89</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>-0.82</td>
</tr>
</tbody>
</table>
Conclusion

Application placement demands analytical model

Model that predicts application placement performance based on multiple factors

Predictions shows -0.82 correlation with simulated performance
For more information:

http://wavescalar.cs.washington.edu
Supporting Material
Sub-model Example: PE Contention

1. Proposed sub-model
   Oversubscription of PE instruction cache hurts performance.

2. Measure Contribution
   In simulator, idealized:
   - Interconnect bandwidth
   - Interconnect latency
   - Data & instruction caches
   Contribution
     = Variance(IPC) / Average(IPC)
     = 1.21

3. Construct sub-model
   PeCapacity = 64
   \( l_p = \text{number of instructions mapped to PE } p \)
   \( \text{Contention}_p = \max(0, l_p - \text{PeCapacity}) \)
   PeContention = \( \sum_p (\text{Contention}_p) \)
Sub-model Example: PE Contention

<table>
<thead>
<tr>
<th>Sub-model</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>art</td>
<td>-0.69</td>
</tr>
<tr>
<td>equake</td>
<td>-0.84</td>
</tr>
<tr>
<td>fft</td>
<td>-0.74</td>
</tr>
<tr>
<td>gzip</td>
<td>-0.83</td>
</tr>
<tr>
<td>lu</td>
<td>-0.65</td>
</tr>
<tr>
<td>mcf</td>
<td>-0.83</td>
</tr>
<tr>
<td>twolf</td>
<td>-0.79</td>
</tr>
<tr>
<td>vpr</td>
<td>-0.67</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>-0.76</strong></td>
</tr>
</tbody>
</table>

The graph shows the simulated contention IPC for the mcf sub-model with various contention metrics.
Sub-model Example: Cache Coherence Overhead

1. Proposed sub-model

*Instruction placement determines location of cache line requests for distributed L1 data cache.*

2. Measure Contribution

In simulator, idealized:
- Interconnect bandwidth
- Interconnect latency
- PE resourced

Contribution

\[
\text{Contribution} = \frac{\text{Variance(IPC)}}{\text{Average(IPC)}} = 0.34
\]

3. Construct sub-model

\[
C_a = \text{number of clusters accessing line } a
\]
\[
N_a = \text{total number of accesses to line } a
\]
\[
\text{misses}_a = \begin{cases} 
1 & \text{if } C_a = 1 \\
C_a & \text{if } C_a > 1
\end{cases}
\]
\[
\text{hits}_a = \begin{cases} 
N_a - 1 & \text{if } C_a = 1 \\
N_a - C_a & \text{if } C_a > 1
\end{cases}
\]

CoherenceOverhead = Average miss rate for all \(a\)
Sub-model Example: Cache Coherence Overhead

<table>
<thead>
<tr>
<th>Sub-model</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>art</td>
<td>-0.92</td>
</tr>
<tr>
<td>equake</td>
<td>-0.99</td>
</tr>
<tr>
<td>fft</td>
<td>-0.33</td>
</tr>
<tr>
<td>gzip</td>
<td>-0.95</td>
</tr>
<tr>
<td>lu</td>
<td>-0.64</td>
</tr>
<tr>
<td>mcf</td>
<td>-0.97</td>
</tr>
<tr>
<td>twolf</td>
<td>-0.92</td>
</tr>
<tr>
<td>vpr</td>
<td>-1.0</td>
</tr>
<tr>
<td>Average</td>
<td>-0.84</td>
</tr>
</tbody>
</table>
Dataflow Execution Model

• Not a new idea [Dennis 1975]
• Code is a graph
  – Vertices = instructions
  – Edges = operands
• Execution governed by “dataflow firing rule”