Isocoupling: Reusing Kernel Coupling Values to Predict the Performance of Parallel Applications

Xingfu Wu, Valerie Taylor
Department of Computer Science
Texas A&M University

IPDPS 2004, April 28, 2004
Outline

- **Introduction**
  - Problem Statement
  - Kernel Coupling
- Extending Kernel Coupling Work
- Isocoupling Metric
- Case Study: SP
- Summary and Future Work
Problem Statement

- Performance models and analyses are critical
  - Requires significant development time
- Significant knowledge exists about performance models for small kernels (e.g., FFT, MM)
- Applications are composed of a finite number of kernels
- Understand the interactions between kernels to develop models of full application

Goal
- Use kernel models to develop full application models
Kernel Coupling

- Kernel: a unit of computation that denotes a logical entity within the larger context of an application (e.g. loop, procedure, or component).

- For two kernels $i$ & $j$, three measurements are required:
  - $P_i$: performance of kernel $i$ isolated
  - $P_j$: performance of kernel $j$ isolated
  - $P_{ij}$: performance of kernels $i$ & $j$ coupled
Kernel Coupling

- Compute the coupling value

\[ C_{ij} = \frac{P_{ij}}{P_i + P_j} \]

- The coupling value simply quantifies the interaction between kernel \( i \) and kernel \( j \).
Valerie Taylor  <taylor@cs.tamu.edu> http://prophesy.cs.tamu.edu

**Coupling Categories**

- $C_{ij} = 1$: No Coupling

- $C_{ij} > 1$: Destructive Coupling

- $C_{ij} < 1$: Constructive Coupling
Outline

- Introduction
  - Problem Statement
  - Kernel Coupling
- Extending Kernel Coupling Work
- Isocoupling Metric
- Case Study: SP
- Summary and Future Work
Extending Kernel Coupling to Chains

\[ C_{BCD} = \frac{P_{BCD}}{P_B + P_C + P_D} \]

\[ C_W = \frac{P_W}{\sum_{l \in W} P_l} \]
Summation Method

- Traditional summation method

\[ T = \sum_{i=1}^{n} N_i P_i \]

Where \( N_i \) gives the number of times kernel \( i \) is executed, \( P_i \) is the performance of kernel \( i \) \((i = 1, 2, \ldots, n)\).

- Problem: inaccurate execution time
- How about interactions between kernels?
Extending Coupling Work

In [TW02], we introduced a coefficient for each term to represent the interaction that each kernel has with the others as follows:

\[ T = \sum_{i=1}^{n} \alpha_i N_i P_i \]

Where \( N_i \) gives the number of times kernel \( i \) is executed, \( P_i \) is the performance of kernel \( i \), and \( \alpha_i (i=1,2,\ldots,n) \) is a weighted average of the coupling values associated with a given kernel \( i \).

The weighted average is the ratio of the execution time attributed to each chain to the summation of the execution times of the chains involving in each kernel.
Extending Coupling Work

- Use weighted averages to determine how to combine coupling values
- Example:

  ✷ Given the pair-wise coupling values

  \[
  T = \alpha_1 P_A + \alpha_2 P_B + \alpha_3 P_C
  \]

  \[
  \alpha_1 = \frac{(C_{AB} \times P_{AB}) + (C_{CA} \times P_{CA})}{P_{AB} + P_{CA}}
  \]

  \[
  \alpha_2 = \frac{(C_{AB} \times P_{AB}) + (C_{BC} \times P_{BC})}{P_{AB} + P_{BC}}
  \]

  \[
  \alpha_3 = \frac{(C_{BC} \times P_{BC}) + (C_{CA} \times P_{CA})}{P_{BC} + P_{CA}}
  \]
Extending Coupling Work

The coefficient $\alpha_i (i = 1,2,\ldots,n)$ for each kernel $i$ as follows:

$$
\alpha_i = \frac{\sum_{W \in Q_i} C_W P_W}{\sum_{W \in Q_i} P_W}
$$

Where $Q_i$ be the set of all ordered chains of $k$ ($2 \leq k < n$) kernels involved with kernel $i$, and $W$ is an ordered chain of $k$ kernels.
Outline

- Introduction
  - Problem Statement
  - Kernel Coupling
- Extending Kernel Coupling Work
- Isocoupling Metric
- Case Study: SP
- Summary and Future Work
Isocoupling Metric

- The relationship between the execution time $P_W$ of the chain $W$ and the coupling value $C_W$

$$P_W = C_W \sum_{j \in W} P_j$$

- The total execution time $T$

$$T = \sum_{i=1}^{n} \alpha_i N_i P_i$$
Isocoupling Metric

- The relationship between $T$, the execution time, and the coupling values $C_W$

\[
T = \sum_{i=1}^{n} (N_i P_i \times \frac{\sum_{W \in Q_i} C_W^2 \sum_{j \in W} P_j}{\sum_{W \in Q_i} C_W \sum_{j \in W} P_j})
\]
Isocoupling Metric

- The mathematical analysis of the equations for the execution time in terms of first and second derivatives with respect to $C_{W}$ does not allow one to draw any clear conclusions.

- The coupling values quantify the interaction between kernel pairs or kernel chains. This quantity represents the amount of sharing of data between kernels.

- Use experimental results to address the issue about isocoupling.
Outline

- Introduction
  - Problem Statement
  - Kernel Coupling
- Extending Kernel Coupling Work
- Isocoupling Metric
- Case Study: SP
- Summary and Future Work
Case Study

- Explore the ability to reuse coupling values to predict parallel application performance.
- Explore the three dimensional space consisting of:
  - Different system architectures
  - Different dataset sizes
  - Different number of processors
Case Study

- **Applications**: NAS Parallel Benchmarks
- **Systems**:
  - Linux Supercluster Los Lobas at UNM
  - Linux Cluster Chiba City at ANL
  - IBM SP at ANL
<table>
<thead>
<tr>
<th>System name</th>
<th>System Type</th>
<th>Number of Processors</th>
<th>CPU type</th>
<th>Network</th>
<th>Operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM SP</td>
<td>Distributed share memory</td>
<td>80</td>
<td>120 MHz P2SC</td>
<td>HPS switch</td>
<td>IBM AIX</td>
</tr>
<tr>
<td>Linux Supercluster</td>
<td>Distributed memory</td>
<td>512</td>
<td>Pentium III 733 MHz</td>
<td>Myrinet</td>
<td>Linux</td>
</tr>
<tr>
<td>Linux Cluster</td>
<td>Distributed memory</td>
<td>512</td>
<td>Pentium III 500 MHz</td>
<td>Myrinet</td>
<td>Linux</td>
</tr>
<tr>
<td>SGI Origin2000</td>
<td>Shared memory</td>
<td>64</td>
<td>195 MHz R10000</td>
<td>Gigabit Ethernet</td>
<td>SGI IRIX</td>
</tr>
</tbody>
</table>
SP (Scalar Pentadiagonal solver) is an application benchmark, which solves three sets of uncoupled systems of equations, first in the x, then in the y, and finally in the z dimension.

- **Initialization**
  - INITIALIZATION: Initializes values.
  - COPY FACES: Phase one computation of the right hand side.
  - TXINVR: Phase two computation of the right hand side.

- **Copy Faces**
  - X SOLVE: Solves the problem in the x dimension.
  - Y SOLVE: Solves the problem in the y dimension.
  - Z SOLVE: Solves the problem in the z dimension.

- **TXINVR**
  - ADD: Updates values.

- **Final**
  - FINAL: Verifies the solution integrity

### SP Data Set

<table>
<thead>
<tr>
<th>SP Data Set</th>
<th>Problem Size</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>36 x 36 x 36</td>
<td>400</td>
</tr>
<tr>
<td>A</td>
<td>64 x 64 x 64</td>
<td>400</td>
</tr>
<tr>
<td>B</td>
<td>102 x 102 x 102</td>
<td>400</td>
</tr>
</tbody>
</table>
Different System Architectures

Figure 1. Experimental and predicted execution times for the SP benchmark with data size W executed on the IBM SP.
## Different System Architectures

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summation</td>
<td>15.95%</td>
</tr>
<tr>
<td>IBM SP (prediction-5)</td>
<td>0.07%</td>
</tr>
<tr>
<td>Linux Supercluster (prediction-5)</td>
<td>-32.49%</td>
</tr>
<tr>
<td>Linux Cluster (prediction-5)</td>
<td>-33.87%</td>
</tr>
<tr>
<td>SGI (prediction-5)</td>
<td>58.70%</td>
</tr>
</tbody>
</table>

**Table. Percentage relative errors for Figure 1**
Different System Architectures

Figure 2. Experimental and predicted execution times for the SP benchmark with data size W executed on the Linux Supercluster.
### Table. Percentage relative errors for Figure 2

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summation</td>
<td>93.15%</td>
</tr>
<tr>
<td>Linux Supercluster (prediction-5)</td>
<td>21.00%</td>
</tr>
<tr>
<td>Linux Cluster (prediction-5)</td>
<td>19.19%</td>
</tr>
<tr>
<td>IBM SP (prediction-5)</td>
<td>65.94%</td>
</tr>
<tr>
<td>SGI (prediction-5)</td>
<td>99.87%</td>
</tr>
</tbody>
</table>
Figure 3. Experimental and predicted execution times for the SP benchmark with data size W executed on the Linux Cluster.
<table>
<thead>
<tr>
<th>Method</th>
<th>Average Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summation</td>
<td>117.78%</td>
</tr>
<tr>
<td>Linux Cluster (prediction-5)</td>
<td>36.01%</td>
</tr>
<tr>
<td>Linux Supercluster (prediction-5)</td>
<td>40.62%</td>
</tr>
<tr>
<td>IBM SP (prediction-5)</td>
<td>90.94%</td>
</tr>
<tr>
<td>SGI (prediction-5)</td>
<td>109.80%</td>
</tr>
</tbody>
</table>

Table. Percentage relative errors for Figure 3
Different Dataset Sizes

Figure 4. Experimental and predicted execution times for the SP benchmark with dataset A executed on the IBM SP.
<table>
<thead>
<tr>
<th>Method</th>
<th>4 processors: Relative Error</th>
<th>9 processors: Relative Error</th>
<th>16 processors: Relative Error</th>
<th>25 processors: Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataA-s</td>
<td>29.09%</td>
<td>20.10%</td>
<td>18.04%</td>
<td>14.93%</td>
</tr>
<tr>
<td>dataA-p5</td>
<td>-1.83%</td>
<td>1.08%</td>
<td>1.32%</td>
<td>0.48%</td>
</tr>
<tr>
<td>dataW-ap5</td>
<td>1.42%</td>
<td>2.76%</td>
<td>4.26%</td>
<td>8.13%</td>
</tr>
<tr>
<td>dataB-ap5</td>
<td>5.77%</td>
<td>0.92%</td>
<td>0.03%</td>
<td>-1.58%</td>
</tr>
</tbody>
</table>

Table. Percentage relative errors for Figure 4
Different Number of Processors

Figure 5. Experimental and predicted execution time for the SP benchmark executed on the IBM SP.
## Different Number of Processors

<table>
<thead>
<tr>
<th>Method</th>
<th>Data size W: Relative Error</th>
<th>Data size A: Relative Error</th>
<th>Data size B: Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summation-9</td>
<td>15.81%</td>
<td>20.10%</td>
<td>20.50%</td>
</tr>
<tr>
<td>Prediction-9p5</td>
<td>-0.92%</td>
<td>1.08%</td>
<td>1.38%</td>
</tr>
<tr>
<td>Prediction-4p5</td>
<td>-8.26%</td>
<td>-7.74%</td>
<td>-0.37%</td>
</tr>
<tr>
<td>Prediction-16p5</td>
<td>2.17%</td>
<td>2.96%</td>
<td>2.08%</td>
</tr>
<tr>
<td>Prediction-25p5</td>
<td>8.90%</td>
<td>4.43%</td>
<td>3.13%</td>
</tr>
</tbody>
</table>

**Table. Percentage relative errors for Figure 5**
Summary

- Proposed Isocoupling metric, and used experiments to explore the ability to reuse coupling values to predict parallel application performance
- The experimental results show reusing coupling values is feasible for
  - Similar system architectures
  - Different dataset sizes
  - Different number of processors