The Fuzzy Correlation between Code and Performance Predictability

Murali Annavaram, Ryan Rakvic, Marzia Polito, Jean-Yves Bouguet, Richard Hankins, Bob Davies
Intel Corporation

Acknowledgements to
John Shen, Carole Dulong, Brad Calder
Why Correlate Code and Performance?

• Predict CPI by observing just EIPs (or PCs)
  – Assume, similar EIP sequence -> similar CPI
  – Shown to work well for some CPU2K benchmarks
  – For improving simulation speed and dynamic optimizations

• Do server workloads exhibit this correlation?
  – Large code base and non-loopy code path
  – Processes communicate through inter process communication
  – Use of OS services

• Regression Trees
  – Quantify CPI prediction accuracy using EIPs
  – Find upper bound for correlation
Workloads & Experimental Infrastructure

- Three representative server workloads
  - ODB-C
    - OLTP benchmark on Oracle 10g RDBMS
  - ODB-H
    - DSS benchmark on Oracle 10g RDBMS
  - SPECjAppServer (SjAS)
    - 3-Tier Application
    - Focus on middleware application server running BEA Weblogic
- Workloads tuned for maximum CPU utilization
- Hardware Configuration
  - Itanium-2 processor based system
  - Red Hat 2.1 + kernel 2.4.9-e.10smp
  - 16 GB of DDR memory
  - 34 Ultra320 SCSI 73 GB drives (used in ODB-C and ODB-H)
  - 200 MHz FS Bus
Tool Chain: Step 1: Data Collection

- VTUNE: Non-intrusive performance monitoring of physical systems
- Samples hardware counters
- No code instrumentation/recompilation
- Collects EIP & TSC once every 1M instructions
  - 2% execution overhead
  - Sampling at 100K instructions has negligible effect
- Sampled EIPs are a good approximation for code path
Step 2: Vector Creation

- Execution divided into 100M instruction interval
- Create 1 EIPV per interval
  - 100 VTune samples per EIPV
  - EIPV has sample count of each unique EIP in that interval
  - Any EIP not sampled in an interval has zero count
- Instantaneous CPI associated with EIPVs
  - CPI = (End Time stamp - Begin Time Stamp) / 100M

<table>
<thead>
<tr>
<th>EIPV</th>
<th>EIP0</th>
<th>EIP1</th>
<th>CPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIPV₀</td>
<td>100</td>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>EIPV₁</td>
<td>20</td>
<td>80</td>
<td>2.0</td>
</tr>
<tr>
<td>EIPV₂</td>
<td>100</td>
<td>0</td>
<td>1.1</td>
</tr>
</tbody>
</table>
**Step 3: Regression Tree Analysis**

<table>
<thead>
<tr>
<th>EIPV</th>
<th>EIP₀</th>
<th>EIP₁</th>
<th>EIP₂</th>
<th>CPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIPV₀</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>EIPV₁</td>
<td>80</td>
<td>0</td>
<td>20</td>
<td>1.1</td>
</tr>
<tr>
<td>EIPV₂</td>
<td>0</td>
<td>20</td>
<td>80</td>
<td>2.6</td>
</tr>
<tr>
<td>EIPV₃</td>
<td>80</td>
<td>20</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>EIPV₄</td>
<td>20</td>
<td>20</td>
<td>60</td>
<td>2.0</td>
</tr>
<tr>
<td>EIPV₅</td>
<td>20</td>
<td>20</td>
<td>60</td>
<td>2.1</td>
</tr>
<tr>
<td>EIPV₆</td>
<td>20</td>
<td>0</td>
<td>80</td>
<td>2.5</td>
</tr>
<tr>
<td>EIPV₇</td>
<td>80</td>
<td>20</td>
<td>0</td>
<td>0.7</td>
</tr>
</tbody>
</table>

- Divide EIPVs into clusters where sum of CPI variance minimized
  - CPI optimally drives EIPV clustering; machine dependent
  - By construction, EIPVs in the same cluster will have similar CPI
- Example: The CPI variance of regression tree clusters is smallest for all possible clusters of size 4

- K-means comparison: Clustering using distance between vectors
  - Does not use CPI values in clustering; machine independent
Computing Relative Error Metric

- In our study, limit the tree size to 50 clusters
  - > 50 clusters does not reduce CPI variance
- Compute CPI variance for all K clusters for each tree $T_K$ $(1 \leq K \leq 50)$
- Relative Error (RE) = weighted sum of cluster CPI variance / overall CPI variance
Interpreting Relative Error Metric

• RE represents CPI variance explained by EIPVs
  – RE=0.15 means that 85% of the CPI variance \textit{explained} by EIPVs.
  – If RE~1 then EIPVs have no relationship with CPI

• Small RE + Small tree size (K)
  – workload behavior exhibits a small number of dominant phases

• If regression tree is large
  – Irrespective of RE, EIPVs and CPI relationship not regulated by few dominant phases

• If CPI variance is small (uniform CPI)
  – No need for regression trees
  – Simple average is acceptable
Regression Tree Results - ODB-C and SjAS

- **ODB-C**
  - CPI has no correlation with EIPs

- **SjAS**
  - Only 20% of CPI variance explained by EIPs
A Visual Explanation - ODB-C and SjAS

- Many unique EIPs compared to SPEC
  - 24K in ODB-C, 31K in SjAS, compared to 646 in MCF from CPU2K
- Small CPI variance
  - 0.01 in ODB-C and 0.03 in SjAS
- Performance independent of EIPs
CPI breakdown – ODB-C and SjAS

- L3 misses occur frequently and uniformly
  - 50% of ODB-C CPI, 35% of SjAS CPI due to L3
- L3 misses overwhelmed other microarchitectural bottlenecks
  - CPI determined by L3 miss latency
ODB-H – Q13

- Three functions
  - Sequential scan, join, sort
- Repeated execution on large input data
- Relative error drops to ~0.15
  - 85% of CPI variance explained by EIPs
- Distinct phases
ODB-H – Q18

- Same 3 functions
  - but uses *index scan* instead of sequential scan
  - more cache and branch misses
- CPI varies widely for the same code
- Relative error is 1.1
Quadrant Classification

- Classify benchmarks into four quadrants using CPI and RE
  - Q-I and Q-II have low CPI variance
    - Relative Error is irrelevant
    - Uniform sampling is OK
  - Q-III: Predicting CPI from EIPVs alone cannot achieve accuracy
    - Machine dependent parameters needed to capture CPI variations
  - Q-IV benchmarks benefit from simple EIP based phase detection

<table>
<thead>
<tr>
<th>Bmark</th>
<th>CPI Var</th>
<th>RE</th>
<th>Bmark</th>
<th>CPI Var</th>
<th>RE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODB-C</td>
<td>0.00</td>
<td>1.01</td>
<td>Q18</td>
<td>0.65</td>
<td>1.00</td>
</tr>
<tr>
<td>Q1</td>
<td>0.01</td>
<td>0.78</td>
<td>SjAS</td>
<td>0.03</td>
<td>0.90</td>
</tr>
<tr>
<td>Q19</td>
<td>0.01</td>
<td>0.37</td>
<td>Q7</td>
<td>0.04</td>
<td>0.19</td>
</tr>
<tr>
<td>Q3</td>
<td>0.01</td>
<td>0.36</td>
<td>Q5</td>
<td>0.03</td>
<td>0.18</td>
</tr>
<tr>
<td>Q16</td>
<td>0.01</td>
<td>0.25</td>
<td>Q10</td>
<td>0.02</td>
<td>0.18</td>
</tr>
<tr>
<td>Q15</td>
<td>0.03</td>
<td>0.18</td>
<td>Q9</td>
<td>0.02</td>
<td>0.17</td>
</tr>
<tr>
<td>Q4</td>
<td>0.02</td>
<td>0.16</td>
<td>Q22</td>
<td>0.04</td>
<td>0.15</td>
</tr>
<tr>
<td>Q2</td>
<td>0.01</td>
<td>0.09</td>
<td>Q21</td>
<td>0.02</td>
<td>0.15</td>
</tr>
<tr>
<td>Q13</td>
<td>0.02</td>
<td>0.15</td>
<td>Q13</td>
<td>0.02</td>
<td>0.15</td>
</tr>
<tr>
<td>Q8</td>
<td>0.02</td>
<td>0.12</td>
<td>Q20</td>
<td>0.03</td>
<td>0.12</td>
</tr>
<tr>
<td>Q14</td>
<td>0.08</td>
<td>0.09</td>
<td>Q14</td>
<td>0.08</td>
<td>0.09</td>
</tr>
<tr>
<td>Q12</td>
<td>0.06</td>
<td>0.06</td>
<td>Q12</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>Q6</td>
<td>0.03</td>
<td>0.06</td>
<td>Q6</td>
<td>0.03</td>
<td>0.06</td>
</tr>
<tr>
<td>Q11</td>
<td>0.02</td>
<td>0.04</td>
<td>Q11</td>
<td>0.02</td>
<td>0.04</td>
</tr>
</tbody>
</table>
Summary

• Using regression trees to identify optimal EIP-CPI relationship in three server workloads

• ODB-C and SjAS
  – CPI has no correlation with EIPs
  – Large code segments
  – Uniform CPI dominated by L3 misses

• ODB-H exhibit a range of behaviors
  – Small code path
  – Algorithmic changes significantly impact CPI and EIP relation

• Quadrant based classification
  – No single sampling technique effective for all
  – Shows best-suited sampling to accurately capture CPI variance