Process Switches and Branch Prediction Accuracy

ELEC 525 – Adv Microprocessor Architecture – Spring 2005

David Chen, Bennett Lau, Jeff Shafer
Hypothesis

- Running multiple processes pollutes the history in branch predictors due to aliasing. This decreases accuracy.
- Branch prediction accuracy can be improved over existing techniques.
  - The history table can be partitioned to allow each process (or a subset of the most frequently executed processes) to have their own history.
Obtaining Instruction Traces

- Used Simics – Whole system simulator
  - Simulated OS: RedHat Linux 7.3
  - Modified OS kernel task scheduler
    - Calls Simics “magic function” with new process ID
  - Modified Simics trace module to catch magic function and output ID inline with instruction trace
  - Installed application suite: Apache web server, MySQL database, SPEC 2000
Branch Predictor Simulator

- Written in Perl
- Parses instruction trace, predicts branches, and compares to actual result in trace file
- Two algorithms
  - Two-Level Global Predictor
    - 12 branch history window, 4K level two history, GShare, 2-bit counter
  - Same predictor plus separate history tables for each process
- Instrumented for history table comparison before & after process switch
Branch Predictor Simulator

- Original results from prior presentation:

- Corrected Simulator:

- Experimented with history table size, window length, and initial history value to optimize accuracy
What is Aliasing?

Process  
A  

GShare by Process  

GShare

Instruction Stream  
B  

A  

...  

...  

...  

{ Aliasing! }
Does Aliasing Exist?

Clearly Aliasing Exists – Is it destructive?
### Accuracy and Utilization

<table>
<thead>
<tr>
<th></th>
<th>ID Branch Predictor Accuracy (%)</th>
<th>G Branch Predictor Accuracy (%)</th>
<th>Accuracy Difference (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache 1</td>
<td>88.68</td>
<td>88.48</td>
<td>0.20</td>
</tr>
<tr>
<td>Apache 2</td>
<td>88.18</td>
<td>87.92</td>
<td>0.26</td>
</tr>
<tr>
<td>Apache 3</td>
<td>88.20</td>
<td>87.79</td>
<td>0.41</td>
</tr>
<tr>
<td>KDE 1</td>
<td>86.85</td>
<td>86.10</td>
<td>0.75</td>
</tr>
<tr>
<td>KDE 2</td>
<td>90.48</td>
<td>88.99</td>
<td>1.49</td>
</tr>
<tr>
<td>KDE 3</td>
<td>90.34</td>
<td>89.70</td>
<td>0.68</td>
</tr>
<tr>
<td>KDE 4</td>
<td>91.16</td>
<td>90.73</td>
<td>0.43</td>
</tr>
<tr>
<td>MySQL</td>
<td>87.62</td>
<td>84.76</td>
<td>2.86</td>
</tr>
<tr>
<td>Pine</td>
<td>86.33</td>
<td>85.77</td>
<td>0.56</td>
</tr>
<tr>
<td>SPEC2000 Gzip</td>
<td>87.04</td>
<td>86.93</td>
<td>0.11</td>
</tr>
<tr>
<td>SPEC2000 Mgrid</td>
<td>86.31</td>
<td>86.16</td>
<td>0.15</td>
</tr>
<tr>
<td>VI</td>
<td>87.94</td>
<td>86.04</td>
<td>1.90</td>
</tr>
</tbody>
</table>

- Removing aliasing improves prediction accuracy (Thus, the aliasing was destructive overall)
Where Do We Improve Performance?
Why Does Performance Change?

MySQL Accuracy Graph

Accuracy (%) vs. Branch Count (in 100s)

- Process Switch
- ID
- G
- Diff

1700 1720 1740 1760 1780 1800
Do Other Traces Show Improved Performance?

- This behavior was present on all traces
- Some show more periods of convergence
- Directly related to task switching frequency
How Long Does Convergence Take?

- Longer convergence time leads to greater potential performance improvement

![Graph showing performance comparison of branch predictors for KDE trace by process.](image1)

![Graph showing average performance comparison of branch predictors for KDE trace.](image2)
Is This True for Other Traces?

Most traces converge (~80% of our traces)
Not All Traces Converge

Performance Comparison of Branch Predictors for VI trace by process

Average Performance Comparison of Branch Predictors for VI trace
**Optimized View**

- Some processes have poor branch predictor performance and dominate the averaged view.
- Eliminating those processes shows convergence more clearly.

![Average Performance Comparison of Branch Predictors for VI trace](image)
Conclusions

- Modern machines do frequent process switches
- Traditional global branch prediction accuracy suffers after process switches
- Convergence period is short (~2000 branches)
- Overall accuracy improvement is dependent on frequency of task switches
- All accuracy improvements are on top of the existing branch predictor accuracy