On the Use of Trace Sampling for Architectural Studies of Desktop Applications

SIGMETRICS 1999
Atlanta, GA

Patrick Crowley & Jean-Loup Baer
Department of Computer Science & Engineering
University of Washington
{pcrowley, baer}@cs.washington.edu
Motivation

• Trace-based simulation is expensive
• Desktop applications are large and can execute billions of instructions
• Often, approximate results or trends are OK

Question: *When can sampling be used to efficiently (i.e., ~10x reduction) drive architectural studies?*
Application Traces

- 5 interactive desktop applications

<table>
<thead>
<tr>
<th>Program</th>
<th>Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acrord32</td>
<td>Adobe Acrobat Reader 3.0</td>
<td>PDF document viewer</td>
</tr>
<tr>
<td>netscape</td>
<td>Netscape Navigator 3.1</td>
<td>Web browser</td>
</tr>
<tr>
<td>photoshop</td>
<td>Adobe Photoshop 4.0</td>
<td>Image editing package</td>
</tr>
<tr>
<td>powerpnt</td>
<td>MS PowerPoint 7.0</td>
<td>Presentation software</td>
</tr>
<tr>
<td>winword</td>
<td>MS Word 7.0</td>
<td>Word processor</td>
</tr>
</tbody>
</table>

- Binary instrumentation used to record memory references*
- Full traces total ~2GB in size
- sampled traces ~200MB

(*OS kernel activity not included)
Two types of errors:

- Sampling Errors
  - Result of parameter choice (i.e., sample size & sampling ratio)
  - Sampling interval, e.g. 5M references
  - Sampling ratio = Sample size/Sampling interval = 0.1
  - Sample size, e.g. 0.5M references

- Errors due to unknown references
  - Cache state is unknown at start of sample
    - Is the first reference to a line a hit or a miss?

Trace Sampling

"sample" the trace
## Trace Sampling Techniques

- Techniques for mitigating error due to unknown references

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cold</strong></td>
<td>Assumes that each unknown reference misses (i.e., an effectively empty cache).</td>
</tr>
<tr>
<td><strong>half</strong></td>
<td>Uses the first half of each sample to “prime” the cache</td>
</tr>
<tr>
<td><strong>stitch</strong></td>
<td>Uses the end state of the previous sample as the start state for the current sample</td>
</tr>
<tr>
<td><strong>INITMR</strong></td>
<td>Estimates the miss ratio of unknown references [Wood et al. 91, Kessler et al. 94]</td>
</tr>
<tr>
<td><strong>true-sample</strong></td>
<td>Starts each sample with the correct cache state</td>
</tr>
</tbody>
</table>
Miss Ratio Determination

Miss Rates for acrord32
(direct-mapped, instruction cache)

- All techniques work up to 32KB (stitch is best)
- stitch and INITMR are reliable up to 64KB
Architectural Study: Victim Cache Trends

**stitch** results for acrord32
(direct-mapped, data cache)

- Results from stitch are good
- Trends clearly demonstrated
See the paper for ...

- More sampling techniques
- Trace sampling for branch prediction
- Trace sampling for analytical cache models
- Large set of results

Conclusions

- **stitch** & **INITMR** are reliable for caches up to 64KB in size
- Sampling *can* be used effectively to drive architectural studies