Intel® VTune™ Performance Analyzer

Performance Tools Lab
Intel Corporation
Agenda

- Overview of the VTune™ Performance Analyzer
- Sampling O.S. and User-Defined performance monitoring counters
- Summary
Some Performance Tool Goals

- The act of measuring performance shouldn’t change the performance of the software
  - Accurate, representative results with no instrumentation
  - Low intrusive performance measurements
  - Interrupt based sampling driven by CPUs registers
- Programmers need to see System-wide performance
  - Identify the software that is consuming most of the CPU
    - Application, shared objects, O.S. layer, device driver, …
- Source code should be the normal view of the performance data
  - Each executable statement annotated with perf. data
VTune™ Performance Analyzer - Products

- VTune™ Performance Analyzer 7.2
  - “Traditional” VTune analyzer
  - Remote capability, inc. Linux*
  - Command line interface (Windows*) included
- VTune Performance Analyzer 3.0 for Linux*
  - GUI-based tool runs on Eclipse
  - Stand alone viewers, command line tool also
- VTune™ Analyzer Driver Kit
  - Rebuild VTune™ Analyzer Linux driver for non-standard kernels (ex: errata kernels, modified kernels)
  - Red Hat, SuSE production distributions supported
Performance Analysis Technologies

- **Sampling - Identify Performance Bottlenecks**
  - Interrupt based sampling
    - CPU events (Event Based Sampling)
    - O.S. events (Perfmon counters)
  - Lower Overhead, less data

- **Call Graph - Examine flow of control through the app**
  - Which functions took the longest
  - Which functions were blocked the longest
  - Calling sequence critical path
  - Higher Overhead, more data
Sampling – O.S. Events

- **Monitor the Counters in Registry**
  - Sample the counters periodically
  - Show a real-time graph of the data
- **Allows addition of user defined counters**
- **When a counter value does something bad:**
  - What was the app doing when the counter value changed?
    - Highlight that part of the graph on the overall run
    - Do the VTune™ Analyzer’s normal drill down to source
```c
float divd_rout(float par1, float par2)
{
    int i;
    float ret_value;
    for(i = 0; i < 10; i++)
        ret_value = par1 / par2;
    return ret_value;
}
```

```
float test_ororl(float divd, float divs, int signif)
{
}
```

### Function Summary

<table>
<thead>
<tr>
<th>Address</th>
<th>Size</th>
<th>Function</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>-------</td>
<td>----</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>Cx1000</td>
<td>0x67</td>
<td>test_if</td>
<td>------</td>
</tr>
<tr>
<td>Cx1067</td>
<td>0x6E</td>
<td>test_if1</td>
<td>------</td>
</tr>
<tr>
<td>Cx10D5</td>
<td>0x2F</td>
<td>test_memset</td>
<td>------</td>
</tr>
</tbody>
</table>

Counter Monitor Results - Thu Aug 04 08:43:52...

- Clockticks (700)
  - 0.00%
  - 27.27%
  - 25.58%
  - 10.36%
Summary

- Free evals, supported platforms, technical papers, ...
- VTune™ Analyzer User Forums
  - [http://softwareforums.intel.com/ids](http://softwareforums.intel.com/ids)