Simulation-Based Tracing and Profiling for System Software Development

Anselm Busse, Reinhardt Karnapke, and Helge Parzyjegla | SYSTOR 2017 | Haifa 2017-05-22
Motivation

Tracing and profiling is crucial to system software development

- State of the art is in-system tracing (e.g. Perf)
- Shortcomings regarding precision and bias
  - Tracing overhead
  - Data collection and storage
- Results may not be deterministic
- Not everything is traceable
Motivation

Tracing and profiling is crucial to system software development

- State of the art is in-system tracing (e.g. Perf)
- Shortcomings regarding precision and bias
  - Tracing overhead
  - Data collection and storage
- Results may not be deterministic
- Not everything is traceable

An outside view might have better and more precise results!
Simulation Based Tracing Setup

Simulation Host

- gem5 Simulator
- Traced Kernel

Tracing and Profiling for System Software Development

**Figure 8.2:**

**Time:** Calls: 1

**Avg.:**

- I/O: 44\·\·10\·\·50\·\·
- CPU: 44\·\·3\·\·528\·\·
- Mem.: 44\·\·3\·\·528\·\·

**Kernel**

- Time: Calls: 1
  - Avg.: 105\·\·9\·\·50\·\·

**CPU**

- Time: Calls: 1
  - Avg.: 105\·\·9\·\·50\·\·

**Mem.**

- Time: Calls: 1
  - Avg.: 105\·\·9\·\·50\·\·

**Trace Core 1**

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Function (Duration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3499669678000:</td>
<td>__schedule (16500)</td>
</tr>
<tr>
<td>3499669694500:</td>
<td>rcu_note_context_switch (12500)</td>
</tr>
<tr>
<td>3499669707000:</td>
<td>__schedule (8500)</td>
</tr>
<tr>
<td>3499669715500:</td>
<td>__raw_spin_lock_irq (9000)</td>
</tr>
<tr>
<td>3499669724500:</td>
<td>__schedule (25500)</td>
</tr>
<tr>
<td>3499669750000:</td>
<td>f_notify (9500)</td>
</tr>
<tr>
<td>3499669759500:</td>
<td>__raw_read_lock (8000)</td>
</tr>
<tr>
<td>3499669767500:</td>
<td>f_notify (13500)</td>
</tr>
<tr>
<td>3499669781000:</td>
<td>__fw_block (6500)</td>
</tr>
<tr>
<td>3499669787500:</td>
<td>__raw_spin_lock (8000)</td>
</tr>
<tr>
<td>3499669795500:</td>
<td>__fw_block (7000)</td>
</tr>
<tr>
<td>3499669802500:</td>
<td>f_notify (14500)</td>
</tr>
<tr>
<td>3499669817000:</td>
<td>__schedule (15000)</td>
</tr>
<tr>
<td>3499669832000:</td>
<td>f_schedule (9500)</td>
</tr>
<tr>
<td>3499669841500:</td>
<td>topo_get_termination (9000)</td>
</tr>
<tr>
<td>3499669850500:</td>
<td>f_schedule (4500)</td>
</tr>
<tr>
<td>3499669855000:</td>
<td>__raw_spin_lock (8000)</td>
</tr>
<tr>
<td>3499669863000:</td>
<td>f_schedule (3000)</td>
</tr>
<tr>
<td>3499669866000:</td>
<td>f_pipe_clean (29000)</td>
</tr>
<tr>
<td>3499669895000:</td>
<td>f_schedule (3000)</td>
</tr>
<tr>
<td>3499669898000:</td>
<td>f_list_empty (7500)</td>
</tr>
<tr>
<td>3499669905500:</td>
<td>f_schedule (14000)</td>
</tr>
</tbody>
</table>

- ...
Annotated Call Tree
Annotated Call Tree

Simulation-Based Tracing and Profiling for System Software Development
Anselm Busse, Reinhardt Karnapke, Helge Parzyjegla | SYSTOR 2017 | Haifa 2017-05-22

Slide 6
Measurement Precision

**Perf**

**Simulation Based**

![Graph showing runqueue contention over time for Perf and Simulation Based](image_url)
Conclusions and Future Work

Simulation-based tracing and profiling has several advantages

- Little to no measurement bias
- Deterministic execution
- No in-code tracing facilities necessary

Future work may include

- Further performance characteristics (e.g. cache-misses)
- Improved data post-processing
- Extension to a complete tracing and profiling framework