Inferred Call Path Profiling (ICPP)

Todd Mytkowicz, Devin Coughlin, and Amer Diwan
University of Colorado at Boulder
Context helps developers

main

A

B

foo

$> ./a.out
segfault
Context helps developers

Calling context is expensive
Challenge: low cost, high context

- Low Cost, Low Context: Hardware Counters
- Low Cost, High Context: Where we want To be!
- High Cost, Low Context: Software Instrumentation
- High Cost, High Context: 

Amount of Context

Cost
Approach

**Phase 1:** capture calling context

calling context: 48

call path: main – A – foo

**Phase 2:** map calling context to call path
Phase 1: calling context with ICPP

- **main:** 8
  - **A:** 16
  - **B:** 8
  - **foo:** 8
Phase 1: calling context with ICPP
Phase 1: calling context with ICPP
Phase 1: calling context with ICPP

stack height

- main
- A
- foo
Phase 1: calling context with ICPP

PC/SP is calling context without computation
Problem: *Ambiguity*

More than one call path maps to same PC/SP
Problem: *Ambiguity*

More than one call path maps to same PC/SP

stack height

```
main

A

foo

8
16
24
32
40

main

B

foo

8
16
24
32
40
```
68% of call paths map to a unique PC/SP
Activation Record Resizing (ARR)
Activation Record Resizing (ARR)

stack height

```
8
16
24
32
40
```

```
8
16
24
32
40
```
Implementing ARR

<table>
<thead>
<tr>
<th>Before ARR:</th>
<th>After ARR:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;Perl_av_store&gt;</code>:</td>
<td><code>&lt;Perl_av_store&gt;</code>:</td>
</tr>
<tr>
<td>402de8: push %rbp</td>
<td>402de8: push %rbp</td>
</tr>
<tr>
<td>402de9: mov %rsp, %rbp</td>
<td>402de9: mov %rsp, %rbp</td>
</tr>
<tr>
<td>402dec: sub $16, %rsp</td>
<td>402dec: sub $32, %rsp</td>
</tr>
</tbody>
</table>

Random search to disambiguate binary
Approach

**Phase 1:** capture calling context

**Phase 2:** map calling context to call path
Phase 2: mapping calling context to call paths

Calling Context:
PC = foo
SP = 64 bytes

What is the call path?

Instrumentation run to build map from PC/SP to call paths
Experimental Methodology

- Intel ICC compiler on 2.4GHz workstation
- SPEC C/C++ CPU 2006
  - Some have too many paths:
    - xalan, gcc, gobmk
  - One use setjmp/longjmp for co-routines
    - omnetpp
Experimental Methodology(II)

- Instrumentation on train input
- Evaluate on ref input
  - Does a call path map to a unique PC/SP?
  - Path could be incorrect
  - Path could be missing
Does a call path map to a unique PC/SP?

C: 75% precise    C++: 73% precise
What about overhead?
Where does overhead come from?

- Excess stack utilization
- Hardware sampling PC/SP pairs
Overhead of ICPP
Overhead of ICPP

Negligible overhead (geomean = 0.2%)
Related Work

- Inoue and Nakatani '09
- Exhaustive Instrumentation:
  - [Ball and Larus '94, Graham et al '82, Bond and McKinley '07]
- Selective Instrumentation
  - [Bernat and Miller '07, Zhuang et al '06]
- Stack Walking
  - [Froyd '05]
High

Low

Cost

Low

High

Amount of Context

ICPP

Software Instrumentation

Hardware Counters