A Case for Unlimited Watchpoints

Joseph L. Greathouse†, Hongyi Xin*, Yixin Luo †‡, Todd Austin†

†University of Michigan  *Carnegie Mellon University  ‡Shanghai Jiao Tong University

ASPLOS, London, UK
March 5, 2012
Goal of This Work

MAKE SOFTWARE FAST
Goal of This Work

MAKE

SOFTWARE

FAST

dynamic

analysis
Goal of This Work

MAKE SOFTWARE FAST ER

dynamic analysis
Dynamic Software Analysis

- Bounds Checking
- Data Race Detection
- Taint Analysis
- Deterministic Execution
- Transactional Memory
- Speculative Parallelization
**Dynamic Software Analysis**

- **Bounds Checking**
  - 10-80x

- **Taint Analysis**
  - 2-30x

- **Transactional Memory**
  - 2-50x

- **Data Race Detection**
  - 2-300x

- **Deterministic Execution**
  - 2-10x

- **Speculative Parallelization**
  - 2-4x
Real Goal of This Work

Generic Hardware to Accelerate Many Dynamic Software Analyses
Real Goal of This Work

Generic Hardware to Accelerate Many Dynamic Software Analyses

WATCHPOINTS
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data

```
A B C D E F G H
```

LD 2
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data

Diagram:

```
0 1 2 3 4 5 6 7
A B C D E F G X
```

Note: WR X→7
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data

R-Watch 2-4
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data

W-Watch 6-7
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data

![Diagram showing hexadecimal values with a red X at LD 2 and locations A, B, C, D, E, F, G, X marked.](image)
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data

```
0 A 1 B 2 C 3 D 4 E 5 F 6 G 7 X
```

WR X→7
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data

- SW knows it’s touching important data
Hardware-Assisted Watchpoints

- HW Interrupt when touching watched data

- SW knows it’s touching important data
  - AT NO OVERHEAD

![Diagram](image)
Dynamic Software Analysis

- Bounds Checking
- Data Race Detection
- Taint Analysis
- Deterministic Execution
- Transactional Memory
- Speculative Parallelization
Dynamic Software Analysis

Bounds Checking  Data Race Detection

Taint Analysis  Deterministic Execution

Transactional Memory  Speculative Parallelization
Watchpoint-Based Taint Analysis

- Taint analysis works on shadow values

\[ x = \text{tainted()} \]

Data
Shadow data
Watchpoint-Based Taint Analysis

- Taint analysis works on shadow values

\[ y = x \times 1024 \]

\[ x = \text{tainted()} \]

Propagate

Data

Shadow data
Watchpoint-Based Taint Analysis

- Taint analysis works on shadow values

\[ y = x \times 1024 \]

\[ x = \text{tainted()} \]

\[ \text{validate}(x) \rightarrow \text{Clear} \]

Data
Shadow data

\[ y = x \times 1024 \rightarrow \text{Propagate} \]
Watchpoint-Based Taint Analysis

- Taint analysis works on shadow values

```plaintext
\[ y = x \times 1024 \]
\[ w = x + 42 \]
```

**Data**
- Shadow data

**Watchpoint-Based Taint Analysis**

- Propagate
- Clear

\[ x = \text{tainted()} \]
\[ \text{validate}(x) \]
Watchpoint-Based Taint Analysis

- Taint analysis works on shadow values

Set R/W watchpoints on tainted values
- No tainted data? → **Run at full speed**
Watchpoint-Based Data Race Detection

- Find inter-thread data sharing, check locks
  - No sharing, no possible data race
  - Turn off detector until HW finds sharing!
Watchpoint-Based Data Race Detection

- Find inter-thread data sharing, check locks
  - No sharing, no possible data race
  - Turn off detector until HW finds sharing!
Watchpoint-Based Data Race Detection

- Find inter-thread data sharing, check locks
  - No sharing, no possible data race
  - Turn off detector until HW finds sharing!
Watchpoint-Based Data Race Detection

- Find inter-thread data sharing, check locks
  - No sharing, no possible data race
  - Turn off detector until HW finds sharing!

![Diagram of data race detection with a fault indicated]
Watchpoint-Based Data Race Detection

- Find inter-thread data sharing, check locks
  - No sharing, no possible data race
  - Turn off detector until HW finds sharing!
Watchpoint-Based Data Race Detection

- Find inter-thread data sharing, check locks
  - No sharing, no possible data race
  - Turn off detector until HW finds sharing!
Watchpoint-Based Data Race Detection

- Find inter-thread data sharing, check locks
  - No sharing, no possible data race
  - Turn off detector until HW finds sharing!
Watchpoint-Based Data Race Detection

- Find inter-thread data sharing, check locks
  - No sharing, no possible data race
  - Turn off detector until HW finds sharing!

Inter-Thread Sharing

FAULT
Needed Watchpoint Capabilities

- Large Number

Z \rightarrow \text{???} \rightarrow V W X Y
Needed Watchpoint Capabilities

- Large Number
- Fine-grained
Needed Watchpoint Capabilities

- Large Number
- Fine-grained
- Per Thread
Needed Watchpoint Capabilities

- Large Number
- Fine-grained
- Per Thread

![Diagram of watchpoints and faults]
Needed Watchpoint Capabilities

- Large Number
- Fine-grained
- Per Thread
- Ranges
Needed Watchpoint Capabilities

- Large Number
- Fine-grained
- Per Thread
- Ranges
**Needed Watchpoint Capabilities**

- Large Number
- Fine-grained
- Per Thread
- Ranges

![Diagram showing watchpoint fault and false faults](image)
Needed Watchpoint Capabilities

- Large Number
- Fine-grained
- Per Thread
- Ranges
Needed Watchpoint Capabilities

- Large Number
- Fine-grained
- Per Thread
- Ranges
- Others in Paper
Existing Watchpoint Solutions

- **Watchpoint Registers**
  - Limited number (4-16), small reach (4-8 bytes)
Existing Watchpoint Solutions

- **Watchpoint Registers**
  - Limited number (4-16), small reach (4-8 bytes)

- **Virtual Memory**
  - Coarse-grained, per-process, *only* aligned ranges
Existing Watchpoint Solutions

- **Watchpoint Registers**
  - Limited number (4-16), small reach (4-8 bytes)

- **Virtual Memory**
  - Coarse-grained, per-process, *only* aligned ranges

- **ECC Mangling**
  - Per physical address, all cores, no ranges
Meeting These Requirements

- Unlimited Number of Watchpoints
  - Store in memory, cache on chip
- Fine-Grained
  - Watch full virtual addresses
- Per-Thread
  - Watchpoints cached per core/thread
  - TID Registers
- Ranges
  - Range Cache
Range Cache

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Watchpoint?</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0xffff_ffff</td>
<td>Not Watched</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
### Range Cache

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Watchpoint?</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0xffff_ffff</td>
<td>Not Watched</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

---

Set Addresses

0x5 – 0x2000

R-Watched
Range Cache

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Watchpoint?</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0x4</td>
<td>Not Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x5</td>
<td>0x2000</td>
<td>R Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x2001</td>
<td>0xffff_ffff</td>
<td>Not Watched</td>
<td>1</td>
</tr>
</tbody>
</table>

Set Address:

0x5 – 0x2000
R-Watched
## Range Cache

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Watchpoint?</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0x4</td>
<td>Not Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x5</td>
<td>0x2000</td>
<td>R Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x2001</td>
<td>0xffff_ffff</td>
<td>Not Watched</td>
<td>1</td>
</tr>
</tbody>
</table>

**Load Address**

0x400
Range Cache

Start Address

<table>
<thead>
<tr>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
</tr>
<tr>
<td>0x5</td>
</tr>
<tr>
<td>0x2001</td>
</tr>
</tbody>
</table>

\[ \leq 0x400? \]

End Address

<table>
<thead>
<tr>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x4</td>
</tr>
<tr>
<td>0x2000</td>
</tr>
<tr>
<td>0xffff ffff</td>
</tr>
</tbody>
</table>

\[ \geq 0x400? \]

Watchpoint?  Valid

<table>
<thead>
<tr>
<th>Watchpoint</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Watched</td>
<td>1</td>
</tr>
<tr>
<td>R Watched</td>
<td>1</td>
</tr>
<tr>
<td>Not Watched</td>
<td>1</td>
</tr>
</tbody>
</table>

Load Address

0x400
## Range Cache

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Watchpoint?</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0x4</td>
<td>Not Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x5</td>
<td>0x2000</td>
<td>R Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x2001</td>
<td>0xffff_ffff</td>
<td>Not Watched</td>
<td>1</td>
</tr>
</tbody>
</table>

≤ 0x400?  
≥ 0x400?

**Load Address**

0x400
## Range Cache

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Watchpoint?</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0x4</td>
<td>Not Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x5</td>
<td>0x2000</td>
<td>R Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x2001</td>
<td>0xffff_ffff</td>
<td>Not Watched</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>≤ 0x400?</th>
<th>≥ 0x400?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Load Address

0x400
Range Cache

<table>
<thead>
<tr>
<th>Start Address</th>
<th>End Address</th>
<th>Watchpoint?</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0x4</td>
<td>Not Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x5</td>
<td>0x2000</td>
<td>R Watched</td>
<td>1</td>
</tr>
<tr>
<td>0x2001</td>
<td>0xffffffff</td>
<td>Not Watched</td>
<td>1</td>
</tr>
</tbody>
</table>

≤ 0x400?  ≥ 0x400?

Load Address 0x400

WP Interrupt
Watchpoint System Design I

- Store Ranges in Main Memory
Watchpoint System Design I

- Store Ranges in Main Memory
Watchpoint System Design I

- Store Ranges in Main Memory
- Per-Thread Ranges, Per-Core Range Cache
Watchpoint System Design I

- Store Ranges in Main Memory
- Per-Thread Ranges, Per-Core Range Cache
Watchpoint System Design I

- Store Ranges in Main Memory
- Per-Thread Ranges, Per-Core Range Cache
- Software Handler on RC miss or overflow
Watchpoint System Design I

- Store Ranges in Main Memory
- Per-Thread Ranges, Per-Core Range Cache
- Software Handler on RC miss or overflow
- Write-back RC works as a write filter
Watchpoint System Design I

- Store Ranges in Main Memory
- Per-Thread Ranges, Per-Core Range Cache
- Software Handler on RC miss or overflow
- Write-back RC works as a write filter
- Precise, user-level watchpoint faults

Diagram:

T1 Memory | T2 Memory | Core 1 | Core 2
Experimental Evaluation Setup

- Pin-based Simulation
  - Every memory access through HW simulator
  - Count pipeline-exposed events
  - Record all other events

- Trace-based timing simulator

- Taint analysis on SPEC INT2000

- Race Detection on Phoenix and PARSEC

- Comparing only shadow value checks
Watchpoint-Based Taint Analysis

- 128 entry Range Cache

![Bar chart showing slowdown times for various benchmarks and tools including MINEMU, Umbra, VM, and RC.]
Watchpoint-Based Taint Analysis

- 128 entry Range Cache

![Bar Chart]

- MINEMU
- Umbra
- VM
- RC

Slowdown (x)

10x 30x 207x 423x 23x 1429x 19x

20% Slowdown

164.gzip 175.vpr 176.gcc 181.mcf 186.crafty 197.parser 252.eon 253.perlbmk 254.gap 255.vortex 256.bzip2 300.twolf GeoMean
The Need for Many Small Ranges

- Some watchpoints better suited for ranges
  - 32b Addresses: 2 ranges x 64b each = 16B
The Need for Many Small Ranges

- Some watchpoints better suited for ranges
  - 32b Addresses: 2 ranges x 64b each = 16B
- Some need large # of small watchpoints
The Need for Many Small Ranges

- Some watchpoints better suited for ranges
  - 32b Addresses: 2 ranges x 64b each = **16B**

- Some need large # of small watchpoints
  - 51 ranges x 64b each = **408B**
  - Better stored as bitmap? 51 bits!
The Need for Many Small Ranges

- Some watchpoints better suited for ranges
  - 32b Addresses: 2 ranges x 64b each = 16B

- Some need large # of small watchpoints
  - 51 ranges x 64b each = 408B
  - Better stored as bitmap? 51 bits!

- Taint analysis has good ranges
- Byte-accurate race detection does not..
Watchpoint System Design II

- Make some RC entries point to bitmaps

Start Addr | End Addr | R  | W  | V
---|---|---|---|---

Watchpoint System Design II

- Make some RC entries point to bitmaps
Watchpoint System Design II

- Make some RC entries point to bitmaps

<table>
<thead>
<tr>
<th>Start Addr</th>
<th>End Addr</th>
<th>R</th>
<th>W</th>
<th>V</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Watchpoint System Design II

- Make some RC entries point to bitmaps

<table>
<thead>
<tr>
<th>Start Addr</th>
<th>End Addr</th>
<th>R</th>
<th>W</th>
<th>V</th>
<th>B</th>
<th>Pointer to WP Bitmap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diagram showing start and end addresses, and flags R, W, V, B pointing to a bitmap.
Watchpoint System Design II

- Make some RC entries point to bitmaps

<table>
<thead>
<tr>
<th>Start Addr</th>
<th>End Addr</th>
<th>Pointer to WP Bitmap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory

Ranges

Core

Range Cache
Watchpoint System Design II

- Make some RC entries point to bitmaps

<table>
<thead>
<tr>
<th>Start Addr</th>
<th>End Addr</th>
<th>Pointer to WP Bitmap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R - W - V - B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - 1 - 1 -</td>
</tr>
</tbody>
</table>

**Memory**

Ranges

Bitmaps

**Core**

Range Cache
Watchpoint System Design II

- Make some RC entries point to bitmaps
Watchpoint System Design II

- Make some RC entries point to bitmaps

<table>
<thead>
<tr>
<th>Start Addr</th>
<th>End Addr</th>
<th>Pointer to WP Bitmap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory

- Ranges
- Bitmaps

Core

- Range Cache
- Bitmap Cache

Accessed in Parallel
Watchpoint-Based Data Race Detection

- RC now 64 entries, added 2KB bitmap cache
Watchpoint-Based Data Race Detection

- RC now 64 entries, added 2KB bitmap cache
Conclusions & Future Directions

- **Watchpoints** a useful generic mechanism

- Numerous SW systems can utilize a well-designed WP system

- In the future:
  - Clear microarchitectural analysis
  - More software systems, different algorithms
Thank You
BACKUP SLIDES
Existing Watchpoint Solutions

- **Watchpoint Registers**
  - Fine-grained, *can* be per-thread
  - Limited number (4-16), small reach (4-8 bytes)

- **Virtual Memory**
  - Virtually unlimited number
  - Coarse-grained, per-process, *only* aligned ranges

- **ECC Mangling**
  - Unlimited, finer-grained
  - Per physical address, no ranges
Width Test