道 ... a 'way', 'path', often used to signify the true nature of the world

**TAO: Two-level Atomicity for Dynamic Binary Optimizations**


MPR/Programming Systems Lab *MCD/Microprocessor Architecture
Intel Labs *Intel Architecture Group
Intel Corporation
April 26, 2010
Agenda

- Atomic execution support
- Optimization scope vs rollback penalty
- Two level atomicity
- Preliminary results
- Conclusions & Future Work
Atomicity and Binary Optimizations

• Binary optimizations are very limited without atomicity support
  – Many optimizations are not allowed: cannot reorder load/load, store/store, early load/later store
  – Many hard issues: memory accesses across cache line boundary, non-cacheable memory operation, precise exception, alias speculation, etc

• Atomic regions
  – Increase the optimization opportunities
  – Address many tough issues
  – Simplify the optimizations design
Atomic Region Scope

- **Small** atomicity scope
  - Traces, super-blocks, basic blocks, etc
  - Fast local optimizations, but small opportunities
- **Large** atomicity scope
  - Loops or large DAGs
  - Global optimizations, loop invariant hoisting, software pipelining, long range prefetch, etc
  - May suffer from resource overflow and large amount of work being discarded when rollback.
Small Atomicity Scope: Loop body

B1 has the same EIP as B2 (region entry)

ckpt

Loop body

r3 ← ld [r2]
r4 ← ld [r2 + r0*4 + 4]
r6 ← r3 + r4
st [r2+4] ← r6
r0 ← r0 + 1
if (r6 < r1) goto B2

B2

B1

B3
Small Atomicity Scope: Loop body

B1

ckpt

r3 ← ld [r2]
r4 ← ld [r2 + r0*4 + 4]
r6 ← r3 + r4
st [r2+4] ← r6
r0 ← r0 + 1
if (r6 < r1) goto B2

cmit

B2

r3 ← ld [r2]
r4 ← ld [r2 + r0*4 + 4]
r6 ← r3 + r4
st [r2+4] ← r6
r0 ← r0 + 1
if (r6 < r1) goto B2

ckpt

B3

...
Small Atomicity Scope: Loop body

Remote store may invalidate the load after commit

B1
ckpt
B2
r3 ← ld [r2]
r4 ← ld [r2 + r0*4 + 4]
r6 ← r3 + r4
st [r2+4] ← r6
r0 ← r0 + 1
if (r6 < r1) goto B2

B3
...

B1
t3 ← ld [r2]
ckpt
B2
r3 ← t3
r4 ← ld [r2 + r0*4 + 4]
r6 ← t3 + r4
st [r2+4] ← r6
r0 ← r0 + 1
if (r6 < r1) goto B2

B3
...

Optimizations: LICM + copy propagation
Small Atomicity Scope: Loop body

Commit requires precise architectural state: Partially dead store elimination is invalid
Large Atomicity Scope: Whole loop

B1

B3

B2

r3 ← ld [r2]
r4 ← ld [r2 + r0*4 + 4]
r6 ← r3 + r4
st [r2+4] ← r6
r0 ← r0 + 1
if (r6 < r1) goto B2

Whole loop
Large Atomicity Scope: Whole loop

Optimizations: LICM + PDSE + copy propagation
Rollback Penalty: Small Atomicity Scope

eip: 0x8 B1
... ...

eip: 0x8 B2
cekpt → ...
... ...

ckpt

B2

B1
B2_1
B2_2
...

B2999
B21000

Exception

Execution
Rollback Penalty: Small Atomicity Scope

Execution

Resume from last checkpoint with non-opt. code eip: 0x8
Rollback Penalty: Small Atomicity Scope

Amount of work discarded by rollback = 1 loop iteration
Rollback Penalty: Large Atomicity Scope

**Execution**

![Diagram showing rollback penalty]

- **eip: 0x8 B1**
- **eip: 0x8 B2**
- **B3**

**Exception**
- B1
- B2
- ...
- B2_{999}
- B2_{1000}
Rollback Penalty: Large Atomicity Scope

Execution

Resume from last checkpoint with non-opt. code (first loop iteration)

Resume optimized code execution

Discarded 1000 iterations

Eip: 0x8 (B2₁)
B1
B2₂
...
B2₉₉₉
B2₁₀₀₀
Rollback Penalty: Large Atomicity Scope

Execution

eip: 0x8 B1

ckpt

... ...

B3

... ...

cmit

Eip: 0x8 (B2_1)

B1

B2_1

B2_2

...

B2_999

B2_1000

Eip: 0x8 (B2_2)

B1

B2_3

...

Fail and discard again!

Discarded 999 iterations
Rollback Penalty: Large Atomicity Scope

Large amount of work discarded at rollbacks
Small and Large Atomicity Scopes

- Optimization opportunities
- Rollback Cost

Small Atomicity Scope
Large Atomicity Scope
Small and Large Atomicity Scopes

- Rollback Cost
- Optimization opportunities

Small Atomicity Scope

Large Atomicity Scope

Two Level Atomicity Scope

TAO
Two Level Atomicity

Region checkpoint

Local ckpt

Local cmit

Region commit

B1 (0x8)

B2 (0x8)

r3 \leftarrow ld [r2]
r4 \leftarrow ld [r2 + r0*4 + 4]
r6 \leftarrow r3 + r4
st [r2+4] \leftarrow r6
r0 \leftarrow r0 + 1
if (r6 < r1) goto B2

B3

Loop Body 1st Level

Whole loop 2nd Level

1st Level Buffer

Local cmit

2nd Level Buffer

Reg. cmit

Non-spec. State

1st Level Buffer

Local cmit

2nd Level Buffer

Reg. cmit

Non-spec. State
Two Level Atomicity

Reg. ckpt. 👉 B1 (0x8)

Local ckpt 👉 B2 (0x8)
- r3 ← ld [r2]
- r4 ← ld [r2 + r0*4 + 4]
- r6 ← r3 + r4
- st [r2+4] ← r6
- r0 ← r0 + 1
- if (r6 < r1) goto B2

Local cmit 👉 B3

Reg. cmit 👉...

Reg. cmit 👉 B2-Fixup
- st [r2+4] ← r6

Reg. cmit 👉 B2
- r3 ← ld [r2]

Local ckpt 👉 B2 (0x8)
- r4 ← ld [r2 + r0*4 + 4]
- r6 ← r3 + r4
- r0 ← r0 + 1
- if (r6 < r1) goto B2

Reg. cmit 👉 B3
- st [r2+4] ← r6
Two Level Atomicity

Execution

B1
B2₁
B2₂

Reg. ckpt.
B1 (0x8)

r3 ← ld [r2]

Local ckpt
B2 (0x8)

r4 ← ld [r2 + r0*4 + 4]
r6 ← r3 + r4
r0 ← r0 + 1
if (r6 < r1) goto B2

Local cmít

Reg. cmít
B2-Fixup

Reg. cmít

B3

st [r2+4] ← r6

1st Level

B2₂

2nd Level

B₂₁,B₁

Non-spec.

...
Two Level Atomicity

Execution

Reg.ckpt. to B1 (0x8)
- r3 ← ld [r2]

Localckpt
- B2 (0x8)
  - r4 ← ld [r2 + r0*4 + 4]
  - r6 ← r3 + r4
  - r0 ← r0 + 1
  - if (r6 < r1) goto B2

Localckpt
- B3
  - st [r2+4] ← r6

Reg.ckpt.
- B2-Fixup
  - st [r2+4] ← r6

Reg.ckpt.
- Non-spec.
  - 1st Level
    - B21000
  - 2nd Level
    - B2999, ..., B21, B1
  - ...
Two Level Atomicity

Execution

Exception

Reg. cmit

B2-Fixup

Eip: 0x8 (B2_{1000})

B1

B2_{1000}

B2_{999}

B1

B2_{1001}

B2_{1002}

Non-spec.

0x8, B2-Fix, B2_{999}, ..., B1

Reg. cmit

B2-Fixup

Reg. cmit

st [r2+4] ← r6

Local cmit

r3 ← ld [r2]

Local ckpt

B2 (0x8)

r4 ← ld [r2 + r0*4 + 4]

r6 ← r3 + r4

r0 ← r0 + 1

if (r6 < r1) goto B2

Reg. ckpt.

B1 (0x8)

r3 ← ld [r2]

Reg. cmit

st [r2+4] ← r6

Reg. cmit

Reg. cmit

Reg. cmit

Reg. cmit
Two Level Atomicity

Reg.ckpt.
B1 (0x8)
r3 ← ld [r2]

Local ckpt
B2 (0x8)
r4 ← ld [r2 + r0*4 + 4]
r6 ← r3 + r4
r0 ← r0 + 1
if (r6 < r1) goto B2

Local cmit
B3
st [r2+4] ← r6

Reg. cmit
st [r2+4] ← r6

Amount of work discarded by rollback = 1 loop iterations

Execution
B1
B2₁
B2₂
...
B2₉₉₉
B2₁₀₀₀
B2-Fixup
Eip: 0x8 (B2₁₀₀₀)
B1
B2₁₀₀₁
B2₁₀₀₂

Exception
Reg. cmit
B2₁₀₀₀

Binary Compilation Technology
Preliminary Results

- Implemented TAO in a cycle accurate simulator
- 1\textsuperscript{st} level atomicity is modeled with frame
- 2\textsuperscript{nd} level atomicity is modeled assuming unlimited speculative cache
- Regions consist of frames
- Global partial redundancy elimination (PRE) and dead code elimination (PDE) are implemented
  - PDSE not measured due to simulator issue
  - Global optimization overhead is not measured, although frame level HW optimizations are modeled
Region and Frame Dynamic Sizes

![Bar Chart for Frames Only](chart1.png)

- FSPEC06: 40 frames
- ISPEC06: 30 frames
- multimedia: 25 frames
- office: 20 frames
- productivity: 15 frames
- server: 10 frames
- workstation: 5 frames
- All: 0 frames

![Bar Chart for TAO Regions](chart2.png)

- FSPEC06: 18000 regions
- ISPEC06: 14000 regions
- multimedia: 10000 regions
- office: 8000 regions
- productivity: 6000 regions
- server: 4000 regions
- workstation: 2000 regions
- All: 0 regions
Performance Potential

- Frames Only
- Additional from TAO Regions

<table>
<thead>
<tr>
<th>Category</th>
<th>Frames Only</th>
<th>Additional from TAO Regions</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSPEC06</td>
<td>15.8%</td>
<td>4.4%</td>
</tr>
<tr>
<td>ISPEC06</td>
<td>6.6%</td>
<td>4.2%</td>
</tr>
<tr>
<td>games</td>
<td>12.7%</td>
<td>3.5%</td>
</tr>
<tr>
<td>multimedia</td>
<td>24.9%</td>
<td>3.1%</td>
</tr>
<tr>
<td>office</td>
<td>12.5%</td>
<td>5.4%</td>
</tr>
<tr>
<td>productivity</td>
<td>15.3%</td>
<td>6.6%</td>
</tr>
<tr>
<td>server</td>
<td>12.9%</td>
<td>4.2%</td>
</tr>
<tr>
<td>workstation</td>
<td>17.8%</td>
<td>8.2%</td>
</tr>
<tr>
<td>All</td>
<td>16.3%</td>
<td>5.3%</td>
</tr>
</tbody>
</table>
Conclusions & Future Work

- Two Level Atomicity enlarges the Optimization Scope with limited rollback costs
- Promising performance gains over frame level atomicity alone
- More optimizations can boost the performance gain
  - Global fusion, etc
- May improve in-order co-designed processor by enabling more global scheduling
- Hardware needs more investigation
  - Pipeline Buffers + Speculative Cache
  - 2-level speculative cache
Questions?