Ratchet

Intermittent Computation without Hardware Support or Programmer Intervention

University of Michigan

Joel Van Der Woude
Sandia

Matthew Hicks
MIT/LL
Batteries are bulky and inconvenient
Energy harvesting
Hardware monitoring vs. programmer insight

or
Goals of Ratchet

- Burden the compiler not the programmer
- Without hardware to measure voltage
- Ensure forward progress under frequent power failures (~100ms)
Checkpointing

• Save volatile state to non-volatile memory
• After a power fail
  – restart
  – restore
  – continue
Re-execution causes incorrectness

Initial | Correct | At Fail | Error
---|---|---|---
mem[a] | x | y | 42
mem[b] | y | 42 | 42

Write after Read (WAR)
What is idempotence?

- Re-execution generates semantically correct result
- A section of code that does not overwrite/update inputs
Idempotence enabling re-execution

\[
\begin{align*}
\text{mem}[a] &= \text{mem}[b] \\
\text{mem}[b] &= 42
\end{align*}
\]

\[
\text{return a}
\]

<table>
<thead>
<tr>
<th></th>
<th>Initial</th>
<th>Correct</th>
<th>At Fail</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>x</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>b</td>
<td>y</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
</tbody>
</table>
Ratchet system overview

```
Ratchet
Idempotence Analysis
Function Entry
Insert Checkpoints

Instruction Selection & Register Allocation

Ratchet
Fix-up
Combine Redundant Checkpoints
Emit Minimum Checkpoint

Finalize
```

Frontend
IR
.c file

Ratchet
IR
.o file

ASM

_asm

_function

_instrumentation

_entry

_checkpoints

_allocated

_register

_allocation

_selection

_instruction

_reduction

_redundancy

_optimization

_compilation

_intermediate

_representation

_translation

_codegen

_output

_file

Idempotence Analysis

- .c file
- Ratchet
  - Idempotence Analysis
  - Function Entry
  - Insert Checkpoints
- Instruction Selection & Register Allocation
- Ratchet
  - Fix-up
  - Combine Redundant Checkpoints
  - Emit Minimum Checkpoint
- .o file
- Finalize

Frontend

12/5/16 jvdw@umich.edu
Identifying WARs

```
mem[a] = mem[b]
mem[b] = 42
...
return a
```
Insert checkpoints

Frontend → IR → Ratchet → IR → Instruction Selection & Register Allocation → ASM → ASM → Finalize

- .c file
- Ratchet
  - Idempotence Analysis
  - Function Entry
  - Insert Checkpoints
- .o file
- Ratchet
  - Fix-up
  - Combine Redundant Checkpoints
  - Emit Minimum Checkpoint
Seperating WARs with checkpoints

• Weighted with
  – Loop depth
  – # of WAR cut

12/5/16
jvdw@umich.edu
Fix-up

Frontend → IR → IR → ASM → Finalize

.c file

Ratchet
Idempotence Analysis
Function Entry
Insert Checkpoints

Instruction Selection & Register Allocation

Ratchet
Fix-up
Combine Redundant Checkpoints

.E file

Emit Minimum Checkpoint

IR
ASM
The backend may create WARs

• Stack Teardown

• Register Spilling
Redundant checkpoints

Frontend
- .c file
- IR

Ratchet
- Idempotence Analysis
- Function Entry
- Insert Checkpoints
- IR
- IR

Instruction Selection & Register Allocation

Ratchet
- Fix-up
- Combine Redundant Checkpoints
- Emit Minimum Checkpoint
- ASM

Finalize
- .o file
- ASM

12/5/16
jvdw@umich.edu
Relocating checkpoints to combine

- Redundant checkpoints caused by...
  - Optimizations
  - Scheduling Decisions
  - Register Spills

```plaintext
r0 = mem[a]  
r1 = mem[b]  
r3 = r0+r1    
r4 = r3+1
checkpoint()  
r5 = r3+r4
mem[a] = r3  
mem[b] = r5
```
Minimum checkpoint

Frontend

.c file

IR

frontend

IR

Function Entry

Instruction Selection & Register Allocation

IR

Idempotence Analysis

ASM

Insert Checkpoints

Combine Redundant Checkpoints

ASM

Ratchet

Finalize

Ratchet

.o file

Fix-up

Emit Minimum Checkpoint

ASM

Function Entry

Idempotence Analysis

Insert Checkpoints

Checkpoints

Redundant Checkpoints
Optimizing Ratchet

- Ratchet
- RatchetFE
- RatchetFE+RD
- RatchetFE+RD+LR
- Ideal

- Unoptimized
- Function Entry
- +Remove Duplicates
- +Live Registers
- Single Checkpoint
How we evaluated Ratchet

• Simulator
  – Cycle accurate
  – Random power failures
  – Dynamically track idempotence
    – https://github.com/impedimentToProgress/thumbulator

• Benchmarks
  – MI Bench
  – newlib
    – https://github.com/impedimentToProgress/MiBench2
Conclusion
Questions?