Performance Isolation for Multicore within Labeled RISC-V

Zihao Yu, Bowen Huang, Jiuyue Ma, Ninghui Sun, Yungang Bao

Nov 28th, 2017 @ Milpitas

Institute of Computing Technology (ICT),
Chinese Academy of Sciences (CAS)
Control in Computer Architecture

1950s - microprogramming
Focus on control unit

1980s - RISC
Pay more attention to datapath
Weak Control -> Uncertainty

- Latency-critical vs. Throughput-oriented – under multicore environment

Which one is more critical?
Uncertainty in reality

• Datacenter - QoS violation
  

• Aviation - unpredictable WCET


Performance isolation is important!
Inspiration - Computer as a Network

• Labeled network
  – MPLS header carries user demands

• Hardware components communicate via packets
  – PCIe packets, NoC packets, QPI packets...

Apply labeling mechanism to arch? Yes!
Labeled von Neumann Architecture (LvNA)

1. Fine-grained object
2. Semantic association
3. Propagation
4. Software-defined control logic

Programmable Architecture for Resourcing-on-Demand

1. Add label Reg
2. Allocate label to each VM
3. Attach label to each Req
4. Add label-based, programmable control logic
5. Abstract label Regs and CLs into files

Ma et. al, Supporting Differentiated Services in Computers via Programmable Architecture for Resourcing-on-Demand (PARD), ASPLOS, 2015
LvNA + RISC-V = Labeled RISC-V

- Features to add
  - Labels registers after tiles
  - Cache CL
  - Label Converter in TLtoAXI
  - Memory CL
  - PRM

- Open sourced
  [github.com/fsg-ict/labeled-RISC-V](https://github.com/fsg-ict/labeled-RISC-V)

Address mapping

<table>
<thead>
<tr>
<th>DS-id</th>
<th>Base</th>
<th>Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x0000</td>
<td>0x4000</td>
</tr>
<tr>
<td>2</td>
<td>0x8000</td>
<td>0x8000</td>
</tr>
</tbody>
</table>

Labeled token bucket

https://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.pdf
Overheads

• < 3% extra codes
  – 16 lines of chisel code to add labels into RocketChip

• < 5% resource overheads
  – Much less with complex cores, e.g. BOOM

• No performance overheads for critical apps according to the timing report
Demo 1 - LabelHype

**PARD Server**

Isolate resources (address space, device) by CLs

Push the software hypervisor down to LvNA

**Traditional Server**

```
root@ldom:~# while true; do date; root@ldom:~# sleep 1; done
```
Demo 2 - Memory Bandwidth Isolation

- Use labeled token buckets to protect the bandwidth from attacker

<table>
<thead>
<tr>
<th>Function</th>
<th>Best Rate MB/s</th>
<th>Avg time</th>
<th>Min time</th>
<th>Max time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy:</td>
<td>53.8</td>
<td>0.074376</td>
<td>0.074305</td>
<td>0.074430</td>
</tr>
<tr>
<td>Scale:</td>
<td>46.7</td>
<td>0.086195</td>
<td>0.085691</td>
<td>0.087494</td>
</tr>
<tr>
<td>Add:</td>
<td>50.1</td>
<td>0.119921</td>
<td>0.119842</td>
<td>0.120034</td>
</tr>
<tr>
<td>Triad:</td>
<td>48.2</td>
<td>0.124578</td>
<td>0.124473</td>
<td>0.124679</td>
</tr>
</tbody>
</table>

solo

<table>
<thead>
<tr>
<th>Function</th>
<th>Best Rate MB/s</th>
<th>Avg time</th>
<th>Min time</th>
<th>Max time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy:</td>
<td>11.7</td>
<td>0.344165</td>
<td>0.340619</td>
<td>0.347746</td>
</tr>
<tr>
<td>Scale:</td>
<td>12.1</td>
<td>0.341771</td>
<td>0.331738</td>
<td>0.356621</td>
</tr>
<tr>
<td>Add:</td>
<td>8.1</td>
<td>0.748021</td>
<td>0.737555</td>
<td>0.755204</td>
</tr>
<tr>
<td>Triad:</td>
<td>8.1</td>
<td>0.751058</td>
<td>0.740938</td>
<td>0.760358</td>
</tr>
</tbody>
</table>

interfered

<table>
<thead>
<tr>
<th>Function</th>
<th>Best Rate MB/s</th>
<th>Avg time</th>
<th>Min time</th>
<th>Max time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy:</td>
<td>53.7</td>
<td>0.074583</td>
<td>0.074555</td>
<td>0.074617</td>
</tr>
<tr>
<td>Scale:</td>
<td>46.6</td>
<td>0.085922</td>
<td>0.085853</td>
<td>0.086026</td>
</tr>
<tr>
<td>Add:</td>
<td>50.0</td>
<td>0.120866</td>
<td>0.119926</td>
<td>0.122406</td>
</tr>
<tr>
<td>Triad:</td>
<td>48.1</td>
<td>0.125066</td>
<td>0.124837</td>
<td>0.125717</td>
</tr>
</tbody>
</table>

isolated
Future works & Summary

Plan to tape out with TSMC’s 40nm next year!

Labeled RISC-V - an open-sourced impl. of LvNA

PARD - a case of LvNA for performance isolation

??? - a case of LvNA for security

LvNA - a model of labeled architecture