Logical Partitions on Many-core Processors

Ramya Masti, Claudio Marforio, Kari Kostiainen, Claudio Soriente, Srdjan Capkun

ETH Zurich

ACSAC 2015
Infrastructure as a Service (IaaS)

Personal platforms

Shared cloud platform (IaaS)
- Economies of scale
- Shared resources
- Security problems
Resource Partitioning

Cloud platform with dedicated resources
- Guaranteed performance
- Secure against shared memory/CPU attacks
- Example: IBM cloud
Resource Partitioning

Every logical partition needs dedicated resources

Many logical partitions need platforms that have lots of resources!
Many-core Processors for Logical Partitioning

Cores, caches

Many-core processor

Many simple cores

Many logical partitions

VM

Designed for workloads that share data (no isolation!)

Can many-core architectures support scalable logical partitioning?
Hypervisor Design Alternatives

**Traditional**

- VM
- Hypervisor (H)
- C₀, C₁, ..., Cₙ

**Distributed**

- VM
- Hypervisor (H)
- C₀, C₁, ..., Cₙ

**Centralized**

- Hypervisor (H)
- VM
- VM
- C₀, C₁, ..., Cₙ

Run-time VM-hypervisor interaction
- Increases hypervisor’s attack surface
- Performance overhead
Centralized Hypervisors Today

Every core supports virtualization
- Hypervisor and VM modes
- Privileged instructions and memory addressing
- Enables memory confinement

But every core runs only in one mode!

Unused functionality in every core

Solutions without processor virtualization
Intel Single-chip Cloud Computer (SCC)

Sources: Howard et al. IJSSC’10; The Future of Many-core Computing, Tim Mattson, Intel Labs
Intel SCC Architecture

**TILE**

<table>
<thead>
<tr>
<th>Core</th>
<th>Core</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NETWORK INTERFACE**

Look Up Tables

<table>
<thead>
<tr>
<th>Look Up Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Core**
- Asymmetric processor
- Each core can run an independent OS
- Simple cores: no virtualization support

**Look Up Tables (LUTs)**
- Determines the resources available to a core
- Resources: other tiles, RAM and I/O

**Network on Chip**

All system resources are memory mapped
Intel SCC Address Translation

Virtual address 0x87256199

CORE
Memory Management Unit

Physical address 0x72008127

NETWORK INTERFACE
Look Up Tables (LUTs)

System-wide address

On-tile
LUT
Registers
On-tile Memory

Off-tile
I/O
Memory at Another Tile

DRAM
Problem: Lack of Isolation

Every core can change its memory access configuration!
Master core configures memory confinement for all cores

Network on Chip

DRAM
Contributions

1. **Hardware change to Intel SCC that enables logical partitioning**
   - isolation on Network on Chip
   - emulate in software

2. **Custom hypervisor**
   - small TCB and attack surface

3. **Cloud architecture (IaaS)**
   - implementation and evaluation
Cloud Architecture (IaaS)

- Computing Node
  - Intel SCC
  - VM
  - Crypto engine (HSM)
  - HW-virtualized peripherals
- Management Service
  - User Management
  - VM Management
- Storage Service
Operation example: VM startup

1. Boot
2. Start
3. Fetch VM
4. Decrypt VM
5. Assign core and configure LUTs
6. Start

Encrypt and install VM
Recap of Main Properties

No processor virtualization
- Beneficial for high number of cores

Small TCB
- Hypervisor implementation size 3.4K LOC
- Tolerance to compromise of other cloud components

Reduced VM interaction
- Small attack surface
- No virtualization overhead
<table>
<thead>
<tr>
<th></th>
<th>No Processor Virtualization</th>
<th>Small TCB</th>
<th>Reduced VM Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xen (&gt;100K LOC)</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>HypeBIOS (4K LOC)</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>NoHype (&gt;100K LOC)</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Our solution (3.4K LOC)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Summary

Many-core processor

Yes!
(with minor modifications)

Intel SCC case study
- Isolation on Network on Chip (NoC)
Thank you!

Ramya Masti, Claudio Marforio, Kari Kostiainen, Claudio Soriente, Srdjan Capkun