Isolation on Many-core Architectures
Ramya Jayaram Masti, Devendra Rai, Claudio Marforio, Srdjan Čapkun

Abstract
The use of many-core platforms like Intel’s Single-chip Cloud Computer (Intel’s SCC) in cloud-like environments, requires them to support security guarantees found in common multi-core platforms. In this work we explore the problem of how to isolate execution of sensitive processes on many-core platforms. In particular, we identify the desirable properties of a security kernel that enables isolation on such platforms. We design a centralized security kernel that achieves isolation and assumes small hardware changes to Intel’s SCC. We prototype our design and report the time needed to setup and execute isolated Linux instances.

Desirable Properties
- Small Security Kernel
  Minimize interaction with co-resident (potentially malicious) software. Scheduling and resource management (disengaged).
- Restricted Security Kernel Capabilities
  Minimize the impact of its compromise. Must only be able to terminate a process and not schedule it (DoS).
- Context Awareness
  Mechanism to learn system configuration (e.g., sharing of resources). Preferably without interaction with the Security Kernel.

Experiments
- Linux Setup Time
  1. Lookup Table Setup
  2. Load executable (i.e., Linux image)
  3. Clear on-tile memory (i.e., MPB)
  4. Reset core to start execution

Future Work
- Explore other security properties enabled by many-core systems
- Implement and compare distributed and centralized solutions for Intel’s SCC
- Evaluate other commercially available architectures (e.g., Adapteva’s Epiphany, Tilera’s TilePro)

References

Design Alternatives
- Distributed Security Kernel
  - Avoid single point of failure
  - Implementable on current hardware
  - Requires coordination between components
- Centralized Security Kernel
  - Better disengagement
  - Less intrusive (e.g., for clouds)
  - Requires hardware support