OSMosis: Modeling OS Isolation

Sidhartha Agrawal, Reto Achermann, Margo Seltzer

Problem

- Modern systems provide myriad different **isolation mechanisms**, but it is **difficult to identify** precisely **what hardware and software state is shared** between two tasks.
- This **lack of transparency** leads to architecture-based side-channel attacks and opaque performance/security tradeoffs.

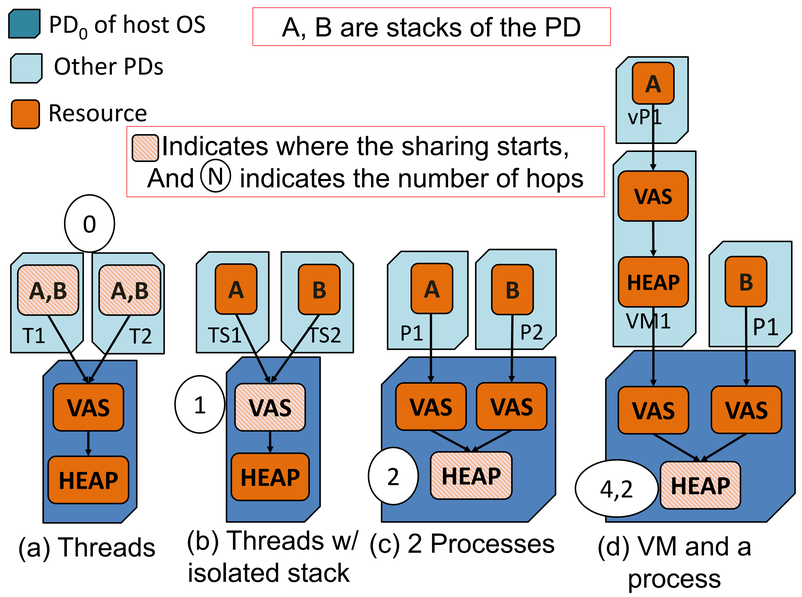
Our Approach

- **Develop a model** that formally describes state sharing.
- **Query the model** to get insights about the extent of sharing between different tasks.
- **Quantify** the degree of isolation.

Solution: Isolation Model

- Every task is a Protection Domain.
- Every Protection Domain has access to Resources.
  - Resource can be Virtual or Physical.
  - Resource Relation is the dependency relation between resources (→)

Using the Model



- The higher the number of hops at which sharing happens, the higher the isolation
- Gives us a concrete way to capture degrees of isolation

Querying the Model

Once the system is captured using a model it is easy to query.

- Find all the resources used by a PD
  - ✓ Transitive closure of the resource relation (→)
- Find the resources used by the PD at ***N*** hops
  - ✓ Traverse the Resource Relation for ***N*** hops
- Find the number of hops at which sharing begins
  - ✓ First common resource for the two PDs
- Find if a PD is sufficiently isolated
  - ✓ For a given number of hops, check that the set of common resources is empty

What else does this enable us to do?

- ✓ **Viewing isolation as a *spectrum***
- ✓ **Precisely state the extent of sharing**
- ✓ **Explore the design space of mechanisms**

What's next?

- Find a performant way to trace resource relations for all the resources
- Protype the model on two real systems viz. Linux and Genode

More Details