Secure Enclaves: An Isolation-centric Approach for Creating Secure High Performance Computing Environments

Ferrol Aderholdt[1], Susan Hicks[1], Thomas Naughton[1], Lawrence Sorrillo[1], Blake Caldwell[2], James Pogge[3] and Stephen L. Scott[3]

[1] Oak Ridge National Laboratory
[2] University of Colorado Boulder
[3] Tennessee Technological University

Challenges

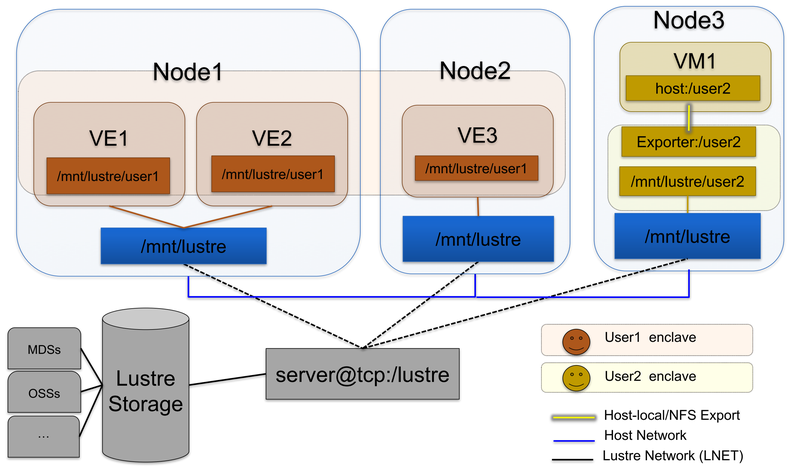
- **HPC environments process variety of workloads**
  - May process data at various levels of security
  - Often forced to enclave at highest security posture
- **Controlling access to shared HPC resources**
  - Gaps in security/protection mechanisms for HPC services
  - Must retain "P" (performance) for HPC workloads
- **Create configurable and secure enclaves**
  - Customizable to support different users
  - Properly isolated for multi-tenant environment

Project Overview

- **Compute: Use different methods of virtualization**
  - Hypervisor-based vs. Containers-based virtualization
  - Virtual Machine (VM): Customize guest kernel/system
  - Virtual Environment (VE): Customize system stack (above kernel)
- **Network: Create per-tenant enclaves**
  - Software Defined Networking (SDN) for on-demand enclaving
- **Storage: Marshal parallel file-system access**
  - File-system security + Network & Compute isolation mechanisms

SUMMARY

1. **Create customizable compute environments with appropriate protections to ensure control is maintained**
2. **Leverage isolation mechanisms at compute, network, file-system layers to create secure & high-performance enclaves**

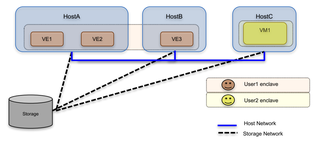


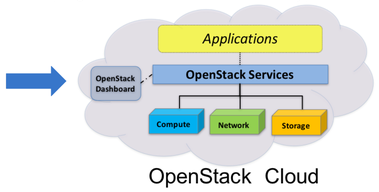
Objectives

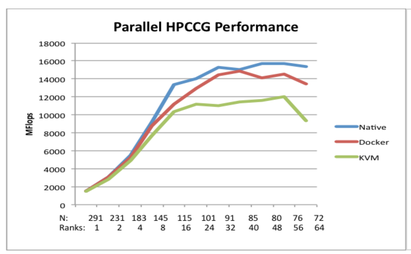
- **Secure Compute Node Customization**
  - Customizable environments and utilization of shared services
  - Low-latency, low-jitter, and high bandwidth
- **Network Enclaving**
  - Configure on demand and isolate network traffic
  - Low-latency, high-bandwidth communication
- **Secure Shared Storage**
  - Ability to isolate access to shared storage on per-enclave basis
  - High-bandwidth and high IOPS

Approach

- **OpenStack-based Prototype**
  - Component based platform for deployment, management & usage
  - Rich set of interfaces & components to map secure enclaves
    - Tenant abstraction, SDN Interfaces/plugins, Different virtualization types

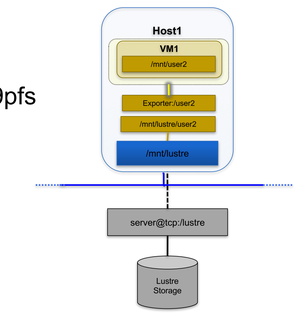






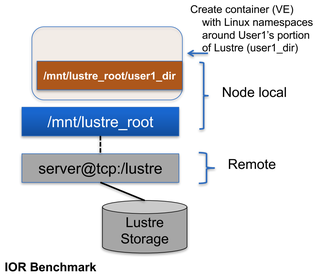
VM Access to Parallel Filesystem

- **Re-Exporting Host Filesystem**
  - RPC based "IO Forwarding"
  - Network File System (NFS) vs. VirtFS/9pfs
- **Setup**
  - Host mounts Lustre
  - Host exports VM's section of Lustre
  - VM (KVM) accesses Lustre via IOFwd



VE Access to Parallel Filesystem

- **Bind-mounted share of Filesystem**
  - Avoid need for IO-Forwarding
  - Single OS offers (near) native performance
- **Setup**
  - Host mounts Lustre
  - Host bind-mounts VE's section of Lustre
  - VE (LXC) access Lustre via mounted region



Status & Next Steps

- **Container-based virtualization shows great promise**
  - Efficient & secure high-performance compute enclaving
  - Near or slightly better IOR performance in VE vs. Native
- **Hypervisor-based very good performance with VirtFS**
  - IO-Forwarding of Lustre from host to VE ~15-20% < Native
- **Leverage OpenStack and SDN for on-demand enclaving**
  - Switch level enforcement with dynamic reconfigurable VLANs
  - User-creatable networks & user-customizable compute instances

Computational Research & Development Programs

This work was supported by the United States Department of Defense (DoD) and used resources of the Computational Research and Development Programs at Oak Ridge National Laboratory.

OAK RIDGE National Laboratory