**Problem**

- Applications are evolving to a more compositional approach, where an overall application workflow is a composition of coupled simulation, analysis, and tool components.
- Each component may have different Operating System and Runtime (OS/R) requirements, in general there is no "one-size-fits-all" solution.
- Co-locating application components can reduce data movement, but may introduce cross-component performance interference.
- Needs for infrastructure for application composition
- Need to maintain performance isolation
- Need to provide data sharing capabilities
- Need to be deployable on production systems

**Approach**

- Leverage experience with Kitten Lightweight Kernel and Palacios Virtual Machine Monitor [1]
- Build infrastructure for application composition:
  - Complement vendor’s Linux stack, add capability
  - Enable OS/R stack flexibility through enclaves [2]
  - Create mechanisms for cross-enclave composition [3,4]

**Hobbes Node Virtualization Layer**

High-level view of the Hobbes Node Virtualization Layer compute node environment. Three enclaves are shown, each running a different application component. The components are composed together across enclaves using shared memory mappings provided by XEMEM [3], copy-on-write memory snapshots provided by XASM [4], I/O mechanisms provided by ADIOS, or by MPI. The Leviathan Node Manager provides a set of tools and infrastructure needed to manage enclaves and application composition. [5]

**Leviathan Node Manager**

Leviathan is an intranode and inter-enclave control service to enable the management and configuration of multiple enclaves running on the same local compute node. In general Leviathan implements a portable interface that is accessible to each enclave instance. Leviathan provides features such as command queues, node level information such as enclave topologies and layouts, advertisements for global resources such as shared memory regions, and general management capabilities such as heartbeat monitors and global process IDs.

Leviathan offers a suite of APIs and services for communication between application components running in separate enclaves. These include command queues, naming services, generic RPC mechanisms, and flexible system call forwarding. Application-specific communication services may also be created.

**Remora**

As part of our work on the Node Virtualization Layer (NVL) of the Hobbes project we have provided a mechanism to compose MPI applications without requiring invasive source code modifications. Remora provides MPI components and support libraries which make it possible for MPI to seamlessly integrate into the Hobbes NVL environment.

**Significance**

Exascale systems are evolving to subsume the functionality of several currently separate systems. Hobbes provides:

- Infrastructure to enable custom system software environments for these different functions
- Support for application composition while maintaining performance isolation
- Interfaces and mechanisms for memory sharing to reduce data movement

**References**

git clone http://www.github.com/hobbesosr/nvl