Hobbes Node Virtualization Layer

System Software Infrastructure for Application Composition and Performance Isolation

Noah Evans, Kevin Pedretti, Shyamali Mukherjee, Ron Brightwell
Sandia National Laboratories

Brian Kocoloski and John R. Lange
University of Pittsburgh

Patrick G. Bridges
University of New Mexico

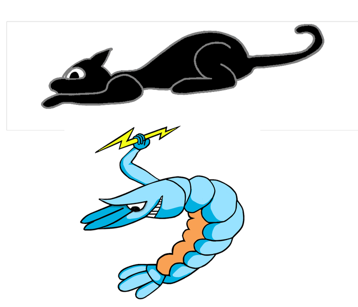
Problem

* 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**

  > Need infrastructure for **application composition**
  > Need to maintain **performance isolation**
  > Need to provide **data sharing capabilities**
  > Need to be deployable on **production systems**

Approach

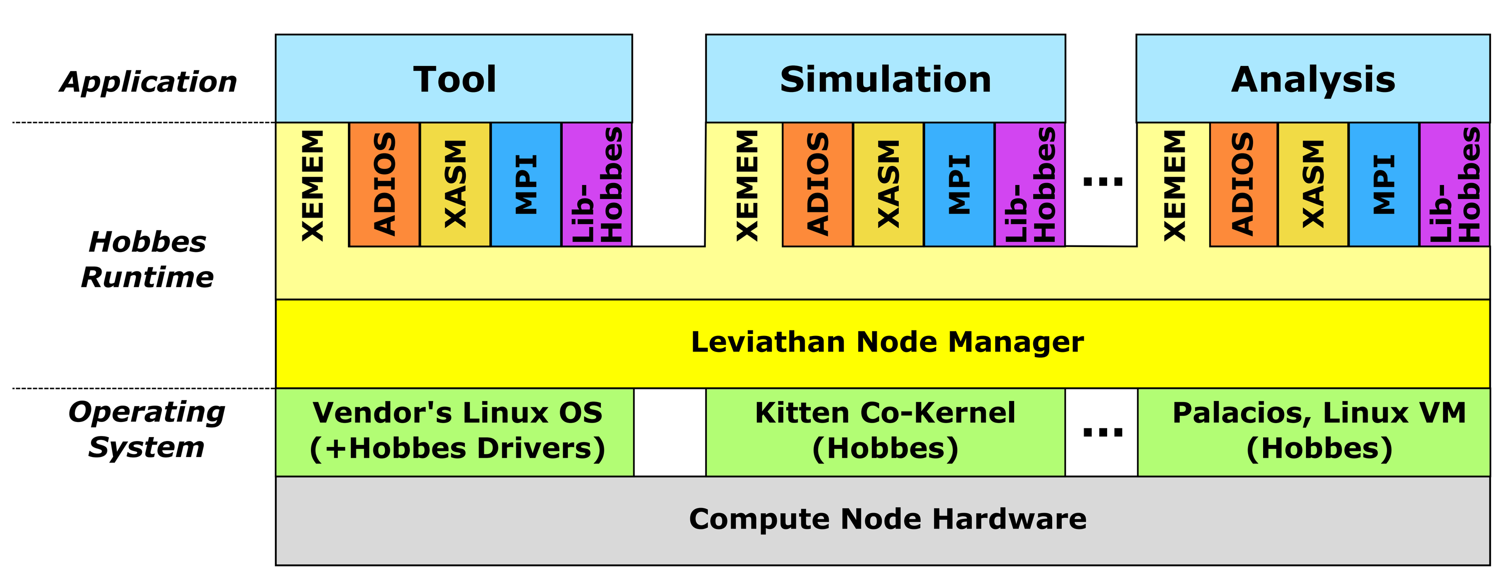
* Leverage experience with Kitten Lightweight Kernel and Palacios Virtual Machine Monitor [1]



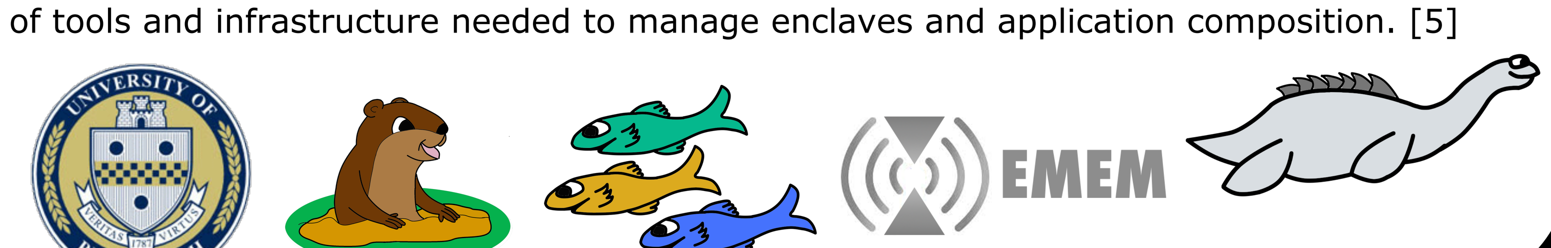
* Build infrastructure for application composition:

  > Complement vendor's Linux stack, add capability
  > Enable OS/R stack flexibiltiy through enclaves [2]
  > Create mechanisms for cross-enclave composition [3,4]

Hobbes Node Virtualization Layer

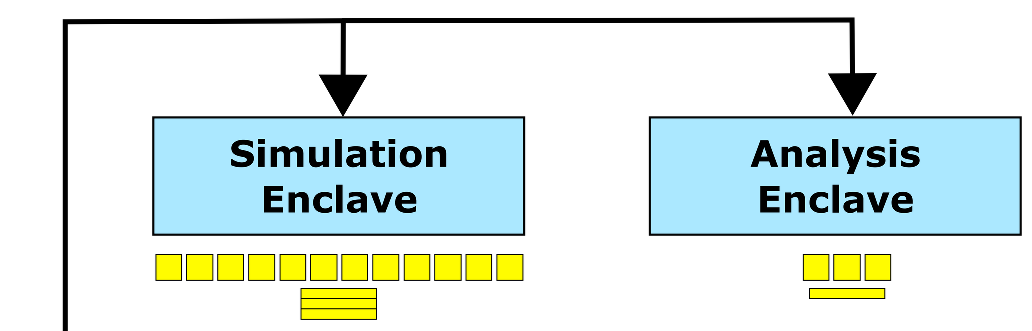


High-level view of the Hobbes Node Virtualization Layer compute node environment. Three enclaves are shown, each running a different application component. The components are composed together across enlaves using shared memory mappings provided by XEMEM [3], copy-on-write memory snapshots provided by 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]

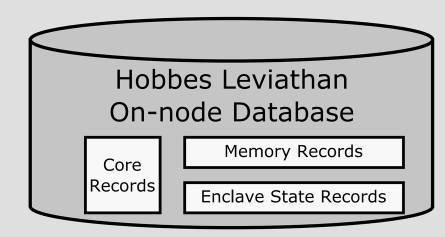


Leviathan Node Manager

Leviathan is an intranode information and 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.

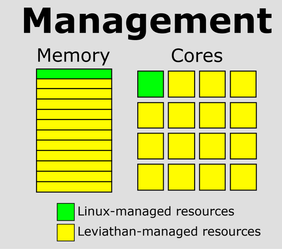


Node Information Service



The Node Information Service tracks the state of all resources managed by Leviathan. The service can be accessed directly by any enclave using the Leviathan client library, libhobbes.a. It is currently implemented using the WhiteDB NoSQL in-memory database and exported to all enclaves via an XEMEM shared memory mapping.

User-Level Resource Management



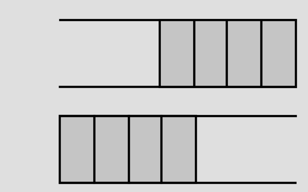
Physical hardware resources such as cores and memory are offlined from the Linux host and placed under the control of Leviathan. Clients can then manage these resources from any enclave. Information such as the NUMA topology can be inspected to determine how to intelligently allocate resources, all at user-level.

Enclave Lifecycle Management

```
Launch/Destroy Enclaves
Launch/Destroy Virtual Machines
Launch/Destroy Applications
```

The resources managed by Leviathan can be space partitioned into multiple enclaves. Each enclave runs its own OS/R stack, for example a native Kitten lightweight kernel instance or Linux virtual machine. The Leviathan shell provides commands for forming enclaves, loading virtual machines, and launching applications.

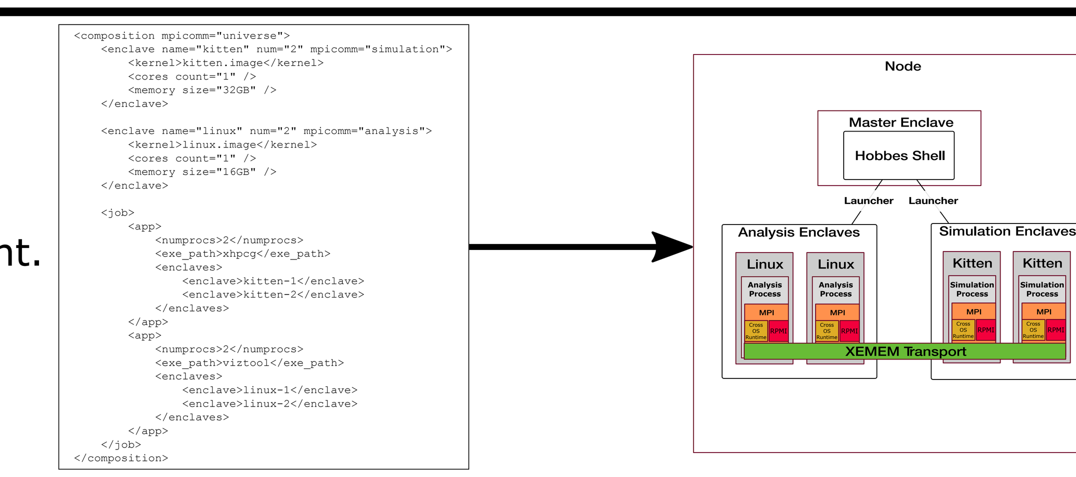
Inter-Enclave Communication



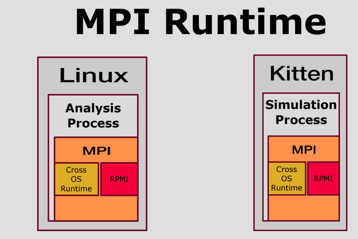
Leviathan offers a suite of APIs and services for communication between application components running in separate enclaves. These include command queues, naming services, generic RPC mechanisms, and flexible system call forwarding. Application-specific communication services may also be created.

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 seemlessly integrate into the Hobbes NVL environment.



Remora Cross Enclave MPI Runtime



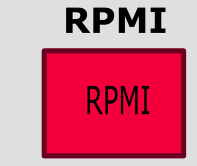
The Remora Cross OS MPI Runtime replaces ORTE in OpenMPI. It provides a set of services that minimize POSIX dependencies in order to run across a wide variety of OSes, especially lightweight kernels like Kitten. Process startup, resource allocation and teardown are done by per-enclave control processes that use Leviathan services and communicate via XEMEM. Remora utilizes the Vader BTL, which is built on the XPMEM API and thus is supported via XEMEM without modification.

Rank Spaces

```
MPI_COMM_UNIVERSE
MPI_COMM_SIMULATION
MPI_COMM_ANALYTICS
```

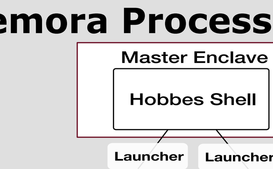
Rank Spaces are a mechanism that Remora uses to make composition of MPI applications possible via custom intracommunicators. Individual applications maintain their original MPI_COMM_WORLD communicator. However, Remora provides the entire composed set of application components with a new intracommunicator MPI_COMM_UNIVERSE, which is the union of all application communicators.

RPMI



The Remora Process Management Infrastructure (RPMI) is an implementation of the PMI client/ server infrastructure, built over Leviathan. The RPMI maintains a global database of mappings of processes to enclaves in an in-memory database. Individual MPI processes query the Leviathan information service in order to discover the global state of MPI applications.

The Remora Process Launcher



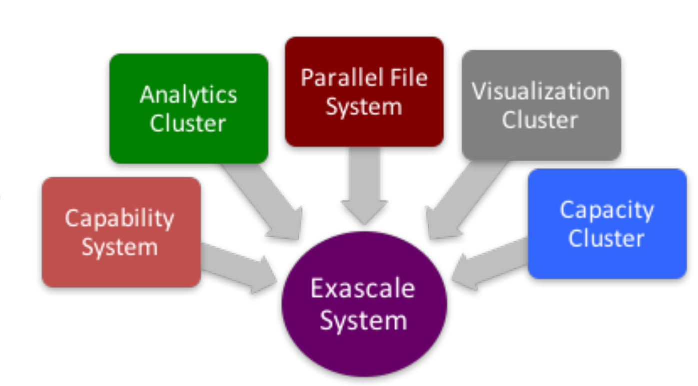
The Remora Process Launcher launches composed applications using an XML file which specifies a multilevel topology mapping applications to communicators and enclaves to applications. The process launcher extends the support provided by the Leviathan lifecycle management tools to allow global rank assignment and spawn jobs across enclaves.

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:

[1] Palacios and Kitten: New High Performance Operating Systems for Scalable Virtualized and Native Supercomputing, IPDPS'10
[2] Achieving Performance Isolation with Lightweight Co-Kernels, HPDC'15
[3] XEMEM: Efficient Shared Memory for Composed Applications on Multi-OS/R Exascale Systems, HPDC'15
[4] A Cross-Enclave Composition Mechanism for Exascale, ROSS'16
[5] Open-source software available at: http://www.prognosticlab.org and "git clone http://www.github.com/hobbesosr/nvl"

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.