System-Level Support for Composition of Applications

Brian Kocoloski Hasan Abbasi David Bernholdt

Jack Lange

University of Pittsburgh

Terry Jones

Oak Ridge National Laboratory

Jai Dayal

Georgia Tech

Noah Evans

Michael Lang

Los Alamos National Laboratory EST. 1943

Jay Lofstead

Kevin Pedretti

Sandia National Laboratories

Patrick Bridges

UNM

The Hobbes Exascale Operating System and Runtime

- Hobbes: Composition and Virtualization as Foundations of an Extreme-Scale OS/R **[Brightwell et al., ROSS '13]**
  - **Hardware challenges of exascale are systemic**
  - Energy efficiency, resilience, management of heterogeneity - cannot be solved by the OS alone.
  - OS needs to provide infrastructure for exploring solutions
  - **Composition and lightweight virtualization help enable systemic research**
- Composition today performed at full system level, not node level
  - Decoupled applications (simulation, post-processing, analytics, etc.) add nontrivial performance overhead and consume power
  - **Node level composition: move computation to data on same node**
- **This talk: Hobbes OS/R with support for composition of real DOE applications**

Talk Roadmap

- Hobbes and the case for Composition at Extreme Scale
- Components of the Hobbes OS/R
- Evaluation of Real DOE Applications
- Conclusion

Composition Use Case: Crack Detection in Molecular Dynamics

- LAMMPS (Large Scale Atomic/Molecular Massively Parallel Simulator)
  - **Used across a variety of domains relevant to DOE interests**
  - Effectively, applies stress to a particular modeled material until it “cracks”
  - Periodically, outputs simulation data referring to various material characteristics (particle positions, atomic makeup, etc.)
- Bonds crack detection
  - **Ingest and analyze LAMMPS output to detect and explore crack genesis**
- Composition details
  - **LAMMPS and Bonds built as separate binary applications**
  - Data transfer accomplished via abstract communication channels. Underlying transport varies based on system capabilities

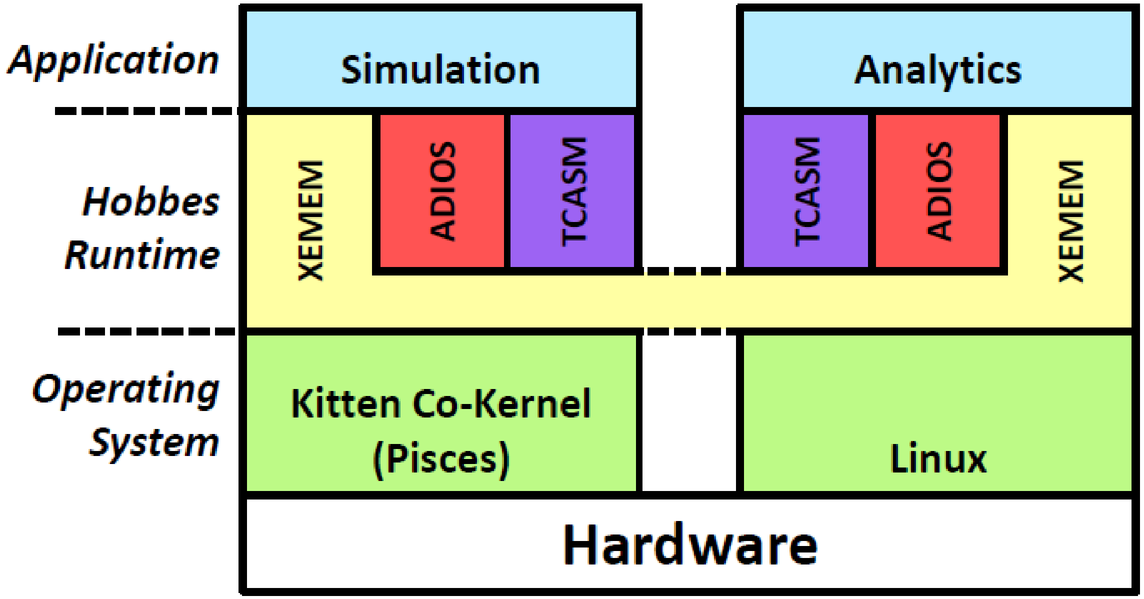
Hobbes in the Broader Exascale OS/R Spectrum

- Recent exascale OS/R efforts: Argo, McKernel, mOS, FusedOS, …
  - Common ground: multi-enclave systems provide customized environments

- **Hobbes: application composition a key capability for exascale systems**
  - Data movement a bottleneck for performance and power consumption
  - Key example: tight coupling of simulation and analysis applications
  - Others: multi-materials simulation, debugging + introspection

- **Performance isolation is a major requirement**
  - This is a systemic problem – hardware is not the only shared resource
  - **Coupling cannot come at the cost of reduced performance isolation**

Hobbes Supporting a Composed Application



- Explicit support for composition via ***enclaves***
- Each *enclave* customized for a particular application component
- **Performance isolation in hardware and system software**
- Consistent shared memory interface to user-level applications

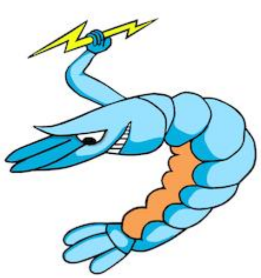
Components of the Hobbes OS/R

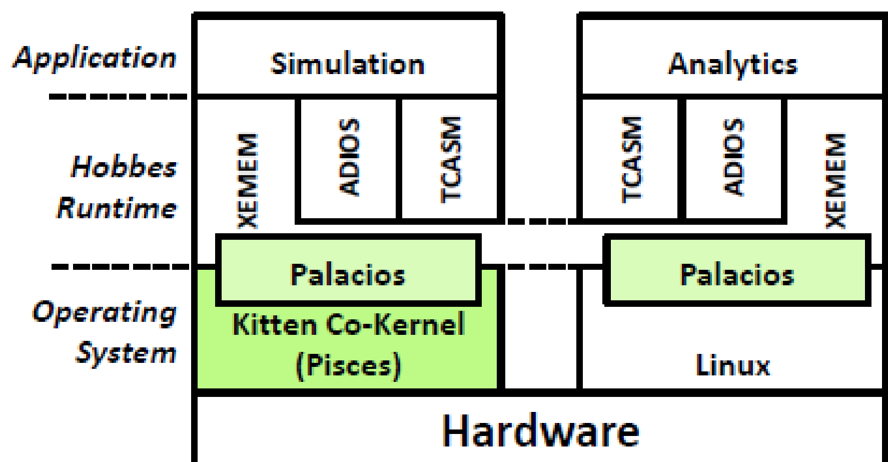
- Operating System Components
  - Palacios and Kitten
  - Pisces lightweight co-kernel architecture

- Runtime Components
  - XEMEM: cross enclave shared memory
  - HPC library support
    - Cray/SGI XPMEM
    - ADIOS (Adaptable I/O System)
    - TCASM (Transparently Consistent Asynchronous Shared Memory) **[Akkan et al., ROSS '13]**

OS Level: Palacios VMM and Kitten LWK

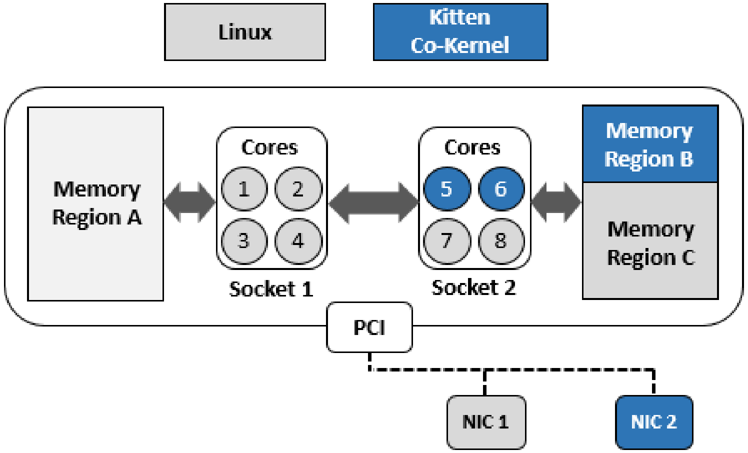
- Palacios: OS-independent, embeddable virtual machine monitor
- Kitten: lightweight kernel from Sandia National Laboratories
- Established history providing scalable environments for HPC
  - **Near native performance at 4096 nodes of a Cray XT3 [Lange et al., IPDPS '10, VEE '11]**
  - **Better than native at small scale [Kocoloski and Lange, ROSS '12]**
- Emphasis on repeatability and consistency
  - **Lack of "enterprise" features**
- Allow application to get "close" to hardware

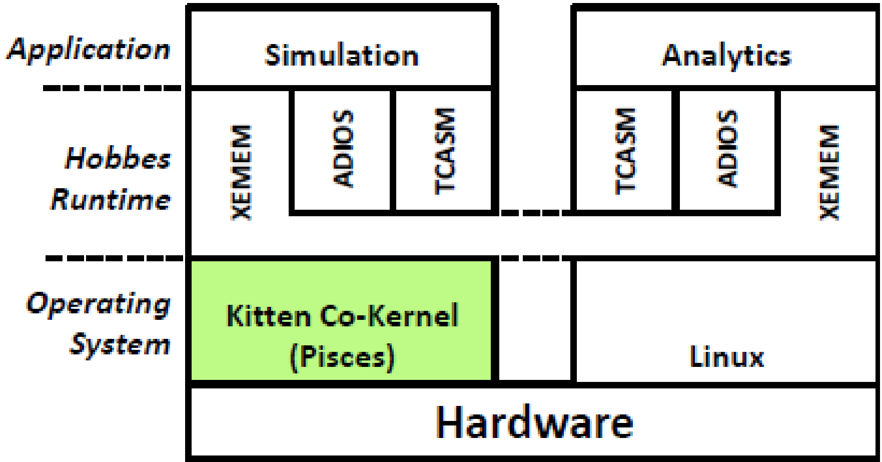




OS Level: Pisces Lightweight Co-Kernel Architecture

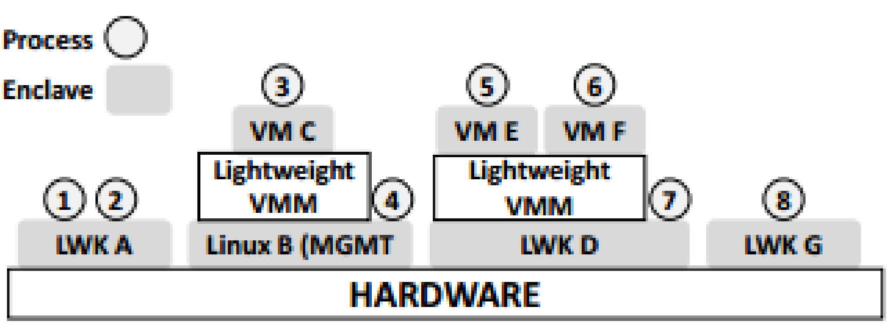
- Supports the decomposition of a node's hardware into independent system software environments **[Ouyang et al., HPDC '15]** **(Talk Thursday!)**
- **Primary design goal: performance isolation between enclaves**
- Decomposed hardware
  - **CPU cores, memory blocks, I/O devices**
- Internalized system software
  - **Operating system, device drivers, I/O + network subsystems**

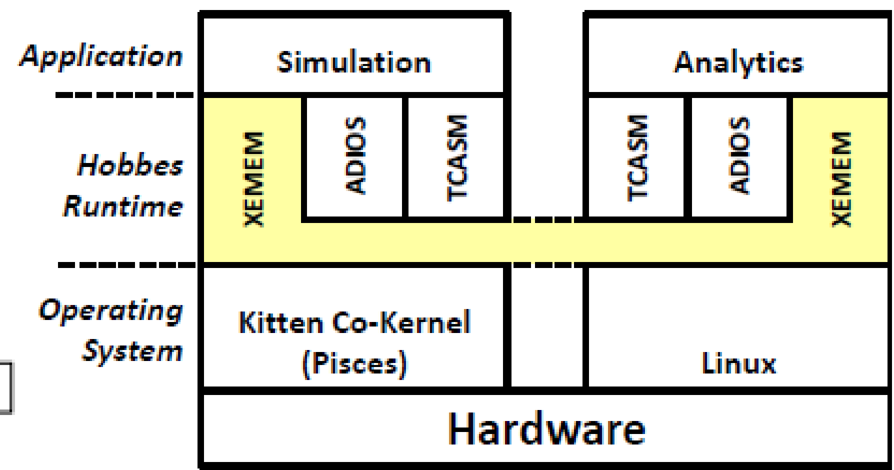




Runtime Foundation: XEMEM (Cross Enclave Memory)

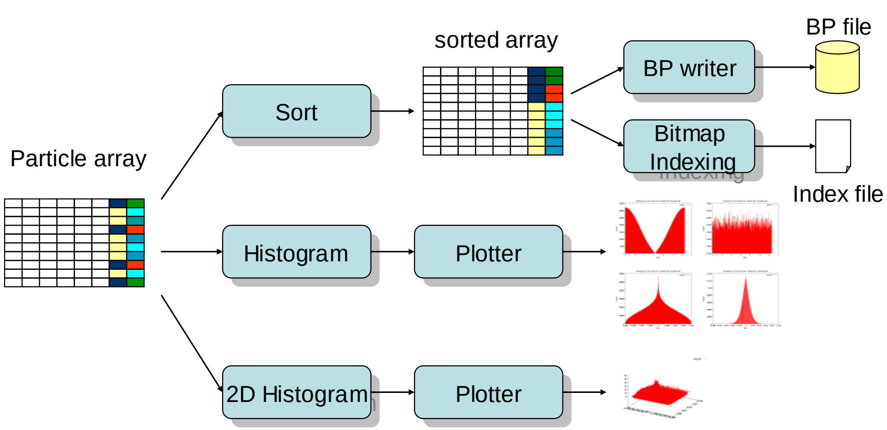
- Shared memory architecture supporting user-level shared memory across enclaves **[Kocoloski and Lange, HPDC '15]** **(Talk tomorrow!)**
- Supports shared memory between Linux host enclave, Kitten co-kernels, and guest OSes in Palacios VMs
- **Arbitrary enclave topologies**
  - Common namespace for exported memory regions
  - Protocol based on cross enclave page frame shipping
- **Backwards compatible API with Cray/SGI XPMEM**

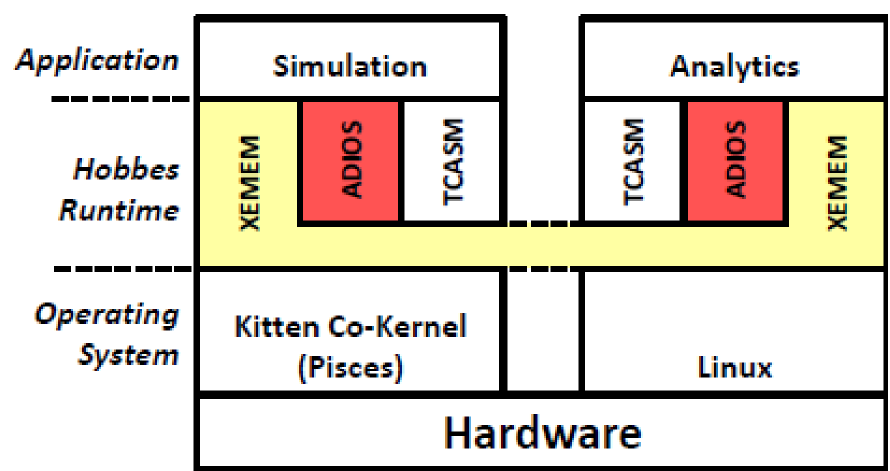




ADIOS (Adaptable I/O System)

- High performance middleware enabling flexible data movement
  - Abstracts Data-at-Rest to Data-in-Motion
- **Established history enabling composition**
  - **Multi-physics [F. Zheng et. al., IPDPS ’10]**
  - **Interactive visualization [Dayal et al., CCGrid ’14]**
- Multiple transport methods which leverage a common API
- **Novel memory / network transports can be integrated quickly**





Evaluation Methodology

- Main goal: proof of concept experimental demonstration
  - **Support of real DOE applications**
  - **Demonstration of functionality and flexibility in underlying enclave configurations**
- Two applications, both highly relevant to DOE
  - **LAMMPS coupled via ADIOS with Bonds**
    - From the SmartPointer analytics toolkit
  - **GTC (Gyrokinetic Toroidal Code) coupled via ADIOS with PreData**
    - Performs sorting of particle data and histogram generation for visualization
- Main performance goal: effective performance isolation through **low application variance** across multiple runs

Evaluation Details

- Evaluation Environment
  - Single compute node of Sandia's "Curie" Cray XK7 testbed
  - Node consists of 16-core 2.1 GHz AMD Opteron CPU, 32 GB DDR3
  - Baseline environment: Compute Node Linux (CNL)

- Enclave configurations
  - Single Linux OS running Cray CNL
  - Multi-enclave environments utilizing Pisces co-kernels running Kitten LWK.
  - Some configurations utilize Palacios embedded with Kitten to provide Linux in VMs.
  - Coupling performed via ADIOS' XEMEM and POSIX file-based transports

Results

- Collected average and standard deviation of at least 5 runs in each enclave configuration

LAMMPS + Bonds

LAMMPS Enclave	Bonds Enclave	Average	StdDev
Cray-Linux (POSIX)		165.02	0.20
Cray-Linux (XEMEM)		153.48	0.08
Cray-Linux	Kitten	153.50	0.15
Kitten	Cray-Linux	153.91	0.04
Cray-Linux	Linux-VM	153.35	0.17
Linux-VM	Cray-Linux	156.10	0.15
Kitten-Enclave1	Kitten-Enclave2	153.83	0.03

GTC + PreData

GTC Enclave	Analysis Enclave	Average	StdDev
Cray-Linux (POSIX)		148.42	0.12
Cray-Linux (XEMEM)		147.40	0.09
Cray-Linux	Linux-VM	147.52	0.09

Results

- Collected average and standard deviation of at least 5 runs in each enclave configuration

LAMMPS + Bonds

LAMMPS Enclave	Bonds Enclave	Average	StdDev
Cray-Linux (POSIX)		165.02	0.20
Cray-Linux (XEMEM)		$\boxed{153.48}$	0.08
Cray-Linux	Kitten	153.50	0.15
Kitten	Cray-Linux	153.91	0.04
Cray-Linux	Linux-VM	$\boxed{153.35}$	0.17
Linux-VM	Cray-Linux	156.10	0.15
Kitten-Enclave1	Kitten-Enclave2	153.83	0.03

GTC + PreData

GTC Enclave	Analysis Enclave	Average	StdDev
Cray-Linux (POSIX)		148.42	0.12
Cray-Linux (XEMEM)		147.40	0.09
Cray-Linux	Linux-VM	147.52	0.09

- No performance loss in most cases, even with running a component virtualized – **performance gain even possible!**

Results

- Collected average and standard deviation of at least 5 runs in each enclave configuration

LAMMPS + Bonds

LAMMPS Enclave	Bonds Enclave	Average	StdDev
Cray-Linux (POSIX)		165.02	0.20
Cray-Linux (XEMEM)		153.48	$\boxed{0.08}$
Cray-Linux	Kitten	153.50	0.15
Kitten	Cray-Linux	153.91	$\boxed{0.04}$
Cray-Linux	Linux-VM	153.35	0.17
Linux-VM	Cray-Linux	156.10	0.15
Kitten-Enclave1	Kitten-Enclave2	153.83	$\boxed{0.03}$

- No performance loss in most cases, even with running a component virtualized – **performance gain even possible!**
- **LAMMPS in Kitten reduces standard deviation**

GTC + PreData

GTC Enclave	Analysis Enclave	Average	StdDev
Cray-Linux (POSIX)		148.42	0.12
Cray-Linux (XEMEM)		147.40	0.09
Cray-Linux	Linux-VM	147.52	0.09

Results

- Collected average and standard deviation of at least 5 runs in each enclave configuration

LAMMPS + Bonds

LAMMPS Enclave	Bonds Enclave	Average	StdDev
Cray-Linux (POSIX)		165.02	0.20
Cray-Linux (XEMEM)		153.48	0.08
Cray-Linux	Kitten	153.50	0.15
Kitten	Cray-Linux	153.91	0.04
Cray-Linux	Linux-VM	153.35	0.17
Linux-VM	Cray-Linux	156.10	0.15
Kitten-Enclave1	Kitten-Enclave2	153.83	0.03

- No performance loss in most cases, even with running a component virtualized – **performance gain even possible!**
- **LAMMPS in Kitten reduces standard deviation**

GTC + PreData

GTC Enclave	Analysis Enclave	Average	StdDev
Cray-Linux (POSIX)		148.42	0.12
Cray-Linux (XEMEM)		147.40	0.09
Cray-Linux	Linux-VM	147.52	0.09

- **GTC performance comparable with analysis in VM**

Conclusion

- **Application composition is an emerging requirement for extreme scale applications**
- The Hobbes OS/R provides explicit support for application composition
  - Multi-enclave OS/R supports heterogeneous application components
  - Performance isolation a key design requirement
  - High performance I/O/middleware libraries support higher-level behavior of unmodified application components
- **The Hobbes OS/R adds no overhead to applications on single node and limits application variance through effective performance isolation**

Upcoming Talks from the Hobbes Team

- **From the Hobbes team:**
  - **Oscar Mondragon: Quantifying Scheduling Challenges for Exascale System Software** (ROSS , right now!)

  - **Kyle Hale: A Case for Transforming Parallel Run-times into Operating System Kernels** (HPDC, Wednesday 10:50 AM)

- **XEMEM, Pisces talks:**
  - **Brian Kocoloski: XEMEM: Efficient Shared Memory for Composed Applications on Multi-OS/R Exascale Systems** (HPDC, Wednesday 4:35 PM)

  - **Jiannan Ouyang: Achieving Performance Isolation with Lightweight Co-kernels** (HPDC, Thursday 2:00 PM)

Thank You

- Brian Kocoloski
  - briankoco@cs.pitt.edu
  - http://people.cs.pitt.edu/~briankoco
- Source available
  - The Prognostic Lab @ U. Pittsburgh
  - http://www.prognosticlab.org

- The Hobbes project
  - http://xstack.sandia.gov/hobbes/

University of Pittsburgh
Oak Ridge National Laboratory
Georgia Tech
Los Alamos National Laboratory EST. 1943
Sandia National Laboratories
UNM

TCASM (Transparently Consistent Asynchronous Shared Memory)

- **Producer consumer model, designed for coupled applications (simulation + analytics, debugging)** **[Akkan et al., ROSS '13]**
  - Simulation + analytics/visualization
  - Debugging
- Leverages copy-on-write (COW) semantics to create multiple views of shared memory pages
  - No wasted memory – copies only made when needed
  - No application-level synchronization
- In Hobbes, XEMEM shares read only snapshots across enclave boundaries