Sandia National Laboratories

Exceptional service in the national interest

CCR
Center for Computing Research

Embracing Diversity: OS Support for Integrating High-Performance Computing and Data Analytics

Ron Brightwell

Scalable System Software Department

Workshop on Clusters, Clouds, and Data for Scientific Computing

October 3-6, 2016

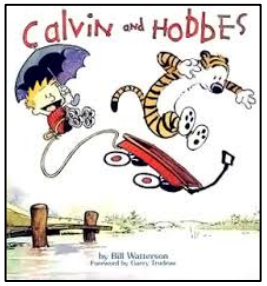
U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

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

Hobbes Project

- US DOE/ASCR project in OS/R Program started in 2013
- Develop prototype OS/R environment for R&D in extreme-scale scientific computing
- Focus on application composition as a fundamental driver
  - Develop necessary OS/R interfaces and system services required to support resource isolation and sharing
  - Evaluate performance and resource management issues for supporting multiple software stacks simultaneously
  - Support complex simulation and analysis workflows
- Provide a lightweight OS/R environment with flexibility to build custom runtimes
  - Compose applications from a collection of enclaves (partitions)
- Leverage Kitten lightweight kernel and Palacios lightweight virtual machine monitor
- 11 partner institutions – 4 DOE labs, 7 universities



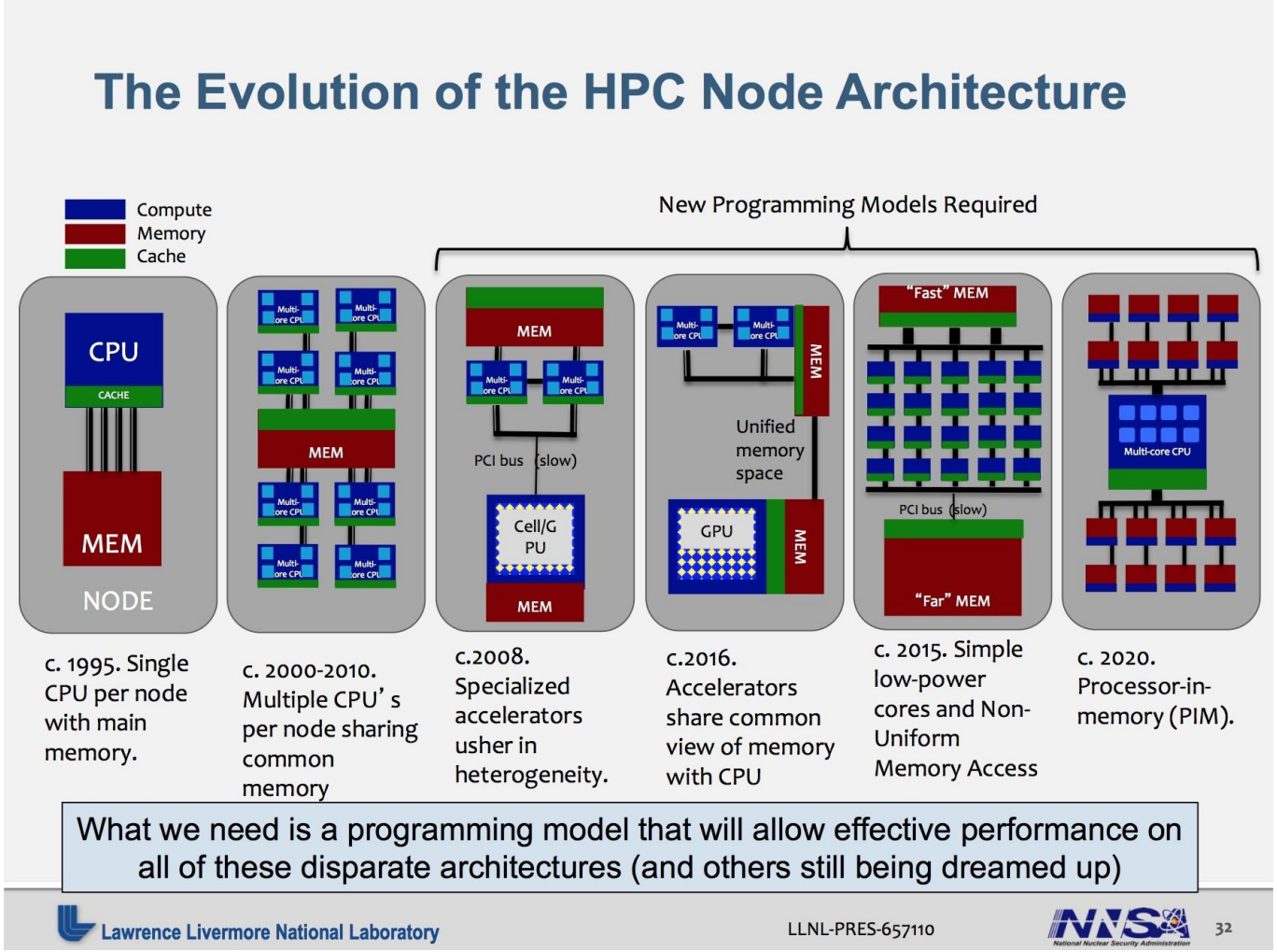
Applications and Usage Models are Diverging

- Application composition becoming more important
  - Ensemble calculations for uncertainty quantification
  - Multi-{material, physics, scale} simulations
  - In-situ analysis and graph analytics
  - Performance and correctness analysis tools
- Applications may be composed of multiple programming models
- More complex workflows are driving need for advanced OS services and capability
  - "Workflow" overtaken "Co-Design" as top US/DOE buzzword
- Support for more interactive workloads
  - Facilities need to find a new charging model
- Desire to support "Big Data" applications
  - Significant software stack comes along with this

Applications Workflows are Evolving

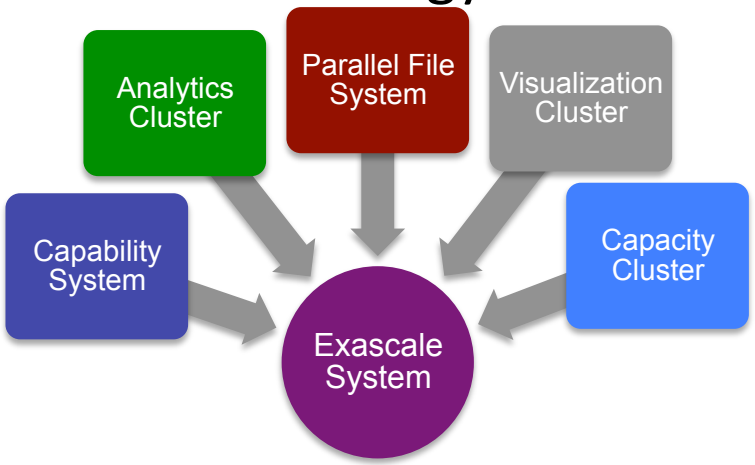
- More compositional approach, where overall application is a composition of coupled simulation, analysis, and tool components
- Each component may have different OS and Runtime (OS/R) requirements, in general there is no "one-size-fits-all" solution
- Co-locating application components can be used to reduce data movement, but may introduce cross component performance interference
  - Need system software infrastructure for application composition
  - Need to maintain performance isolation
  - Need to provide cross-component data sharing capabilities
  - Need to fit into vendor's production system software stack

Node Architecture is Diverging

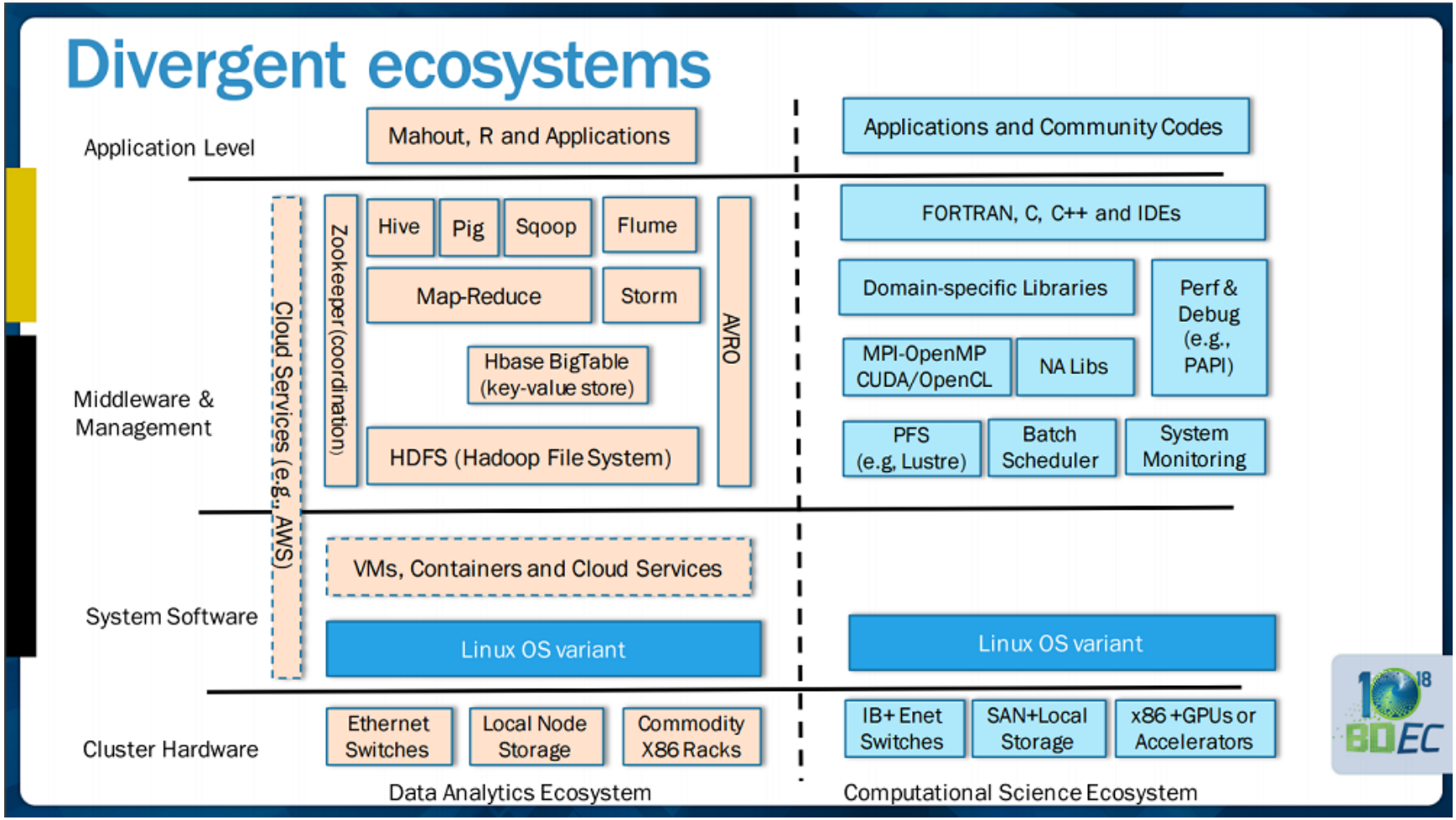


Systems Are Converging to Reduce Data Movement

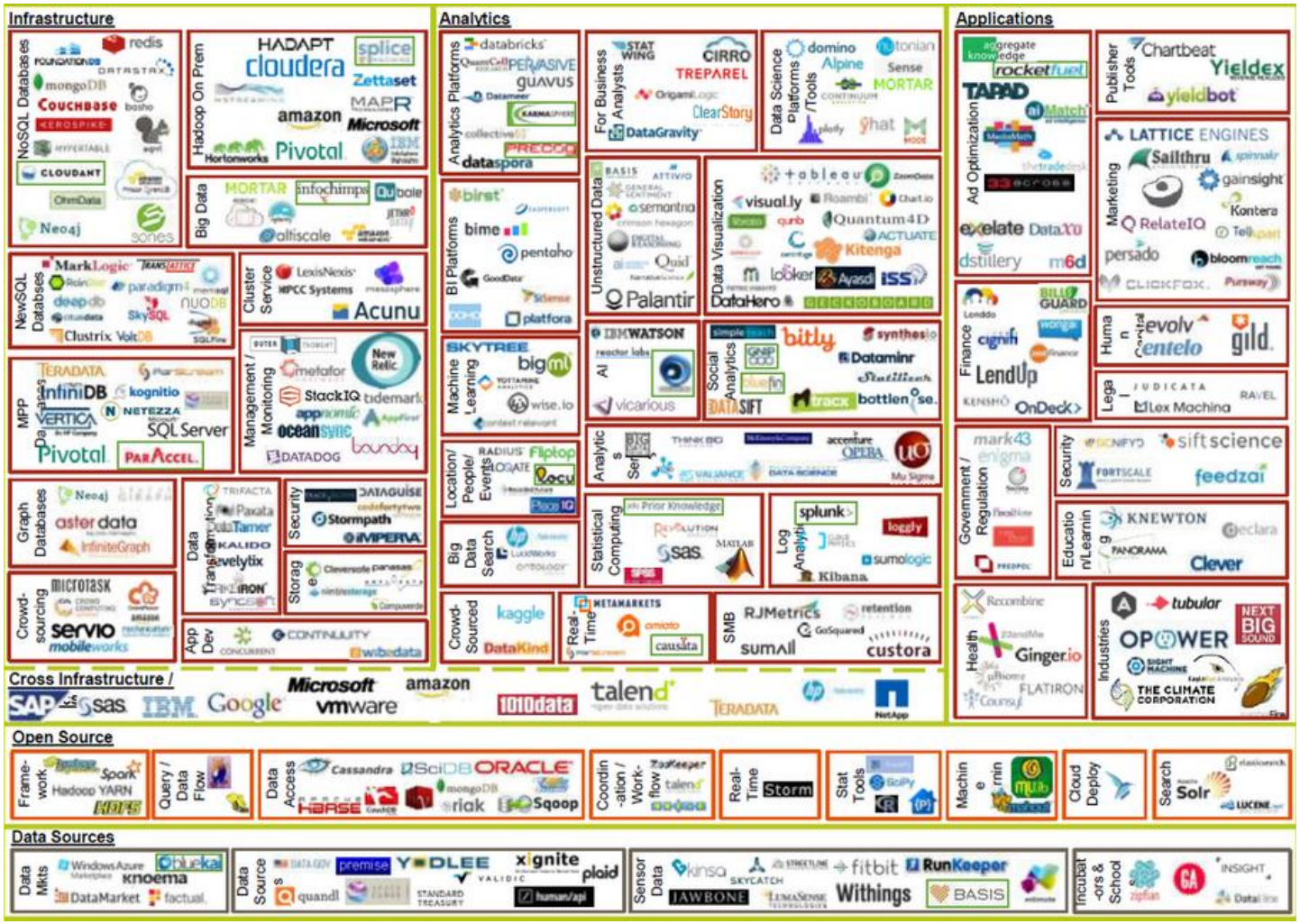
- External parallel file system is being subsumed
  - Near-term capability systems using NVRAM-based burst buffer
  - Future extreme-scale systems will continue to exploit persistent memory technologies
- In-situ and in-transit approaches for visualization and analysis
  - Can't afford to move data to separate systems for processing
  - GPUs and many-core processors are ideal for visualization and some analysis functions
- Less differentiation between advanced technology and commodity technology systems
  - On-chip integration of processing, memory, and network
  - Summit/Sierra using InfiniBand



HPC and Big Data Software Stacks



“Big Data” Environment

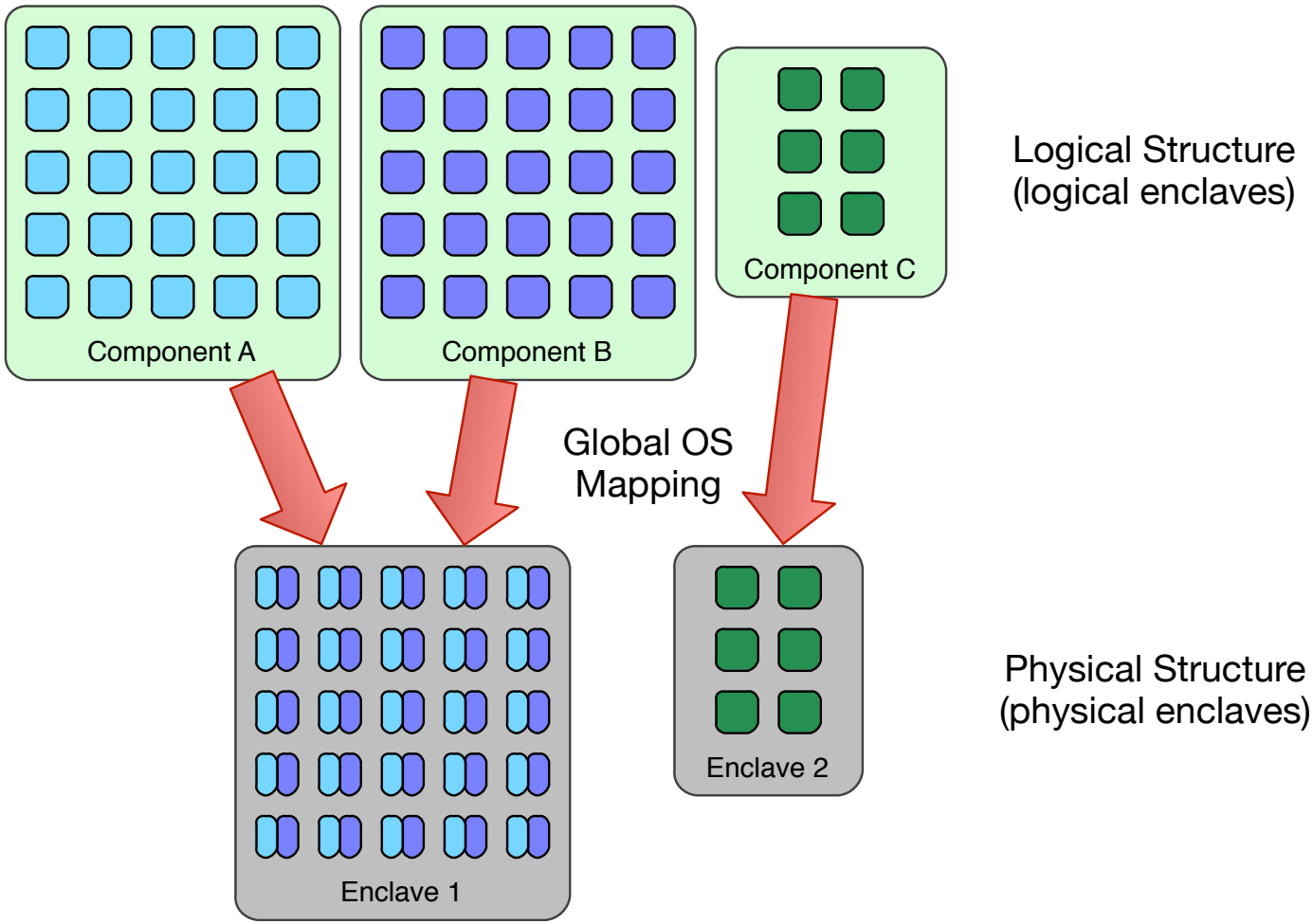


We Should Embrace Divergence

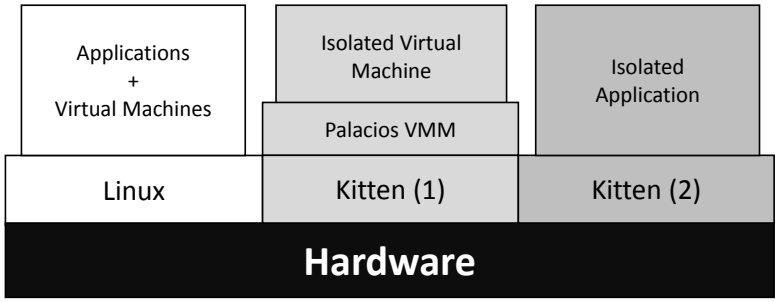
CCR
Center for Computing Research

- Functional partitioning based on software stack will continue
  - Service nodes, I/O nodes, network nodes, compute nodes, etc.
  - Nodes are becoming too big to be smallest unit of allocation
- Provide infrastructure to manage diverse software stacks
  - Node-level partitioning of resources with different stacks
  - Support for improved resource isolation
  - Mechanisms that provide sharing to reduce data movement
- Enable applications and workflows to define their own software environment

Application Composition in Hobbes



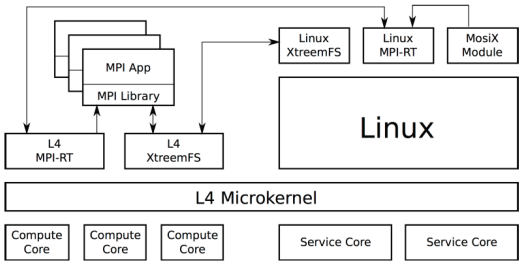
Hobbes Using a Co-Kernel Architecture



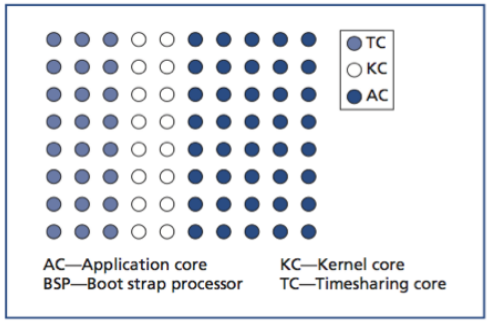
- Multi-stack architecture tools implemented and functional
  - Host boots Linux
  - Cores and memory can be taken from Linux, forming one or more containers
  - Kitten can be launched in each container
  - Each Kitten instance operates cooperatively with Linux as a co-kernel
  - Each co-kernel can run a different application
    - Or guest OS via Palacios VMM
  - Containers can be dynamically resized without rebooting
    - Number of cores and size of memory can grow and shrink
  - Shared memory communication between any OS using XEMEM
- Ported to Cray Linux Environment
- Multi-enclave launch working on XK7 testbed at Sandia

“Combined OS” Approach is Not New

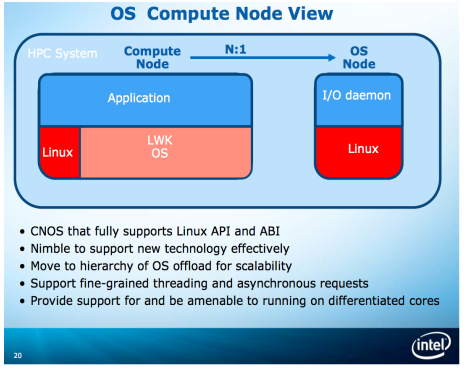
TU Dresden L4Linux (2010)



IBM/Bell Labs NIX (2012)



Intel mOS (2013)



IBM FusedOS (2011)

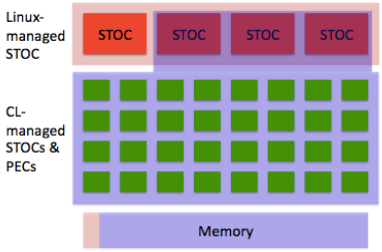


Fig. 3. Partitioning of cores and memory for HPC applications.

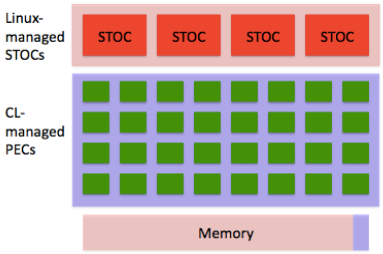
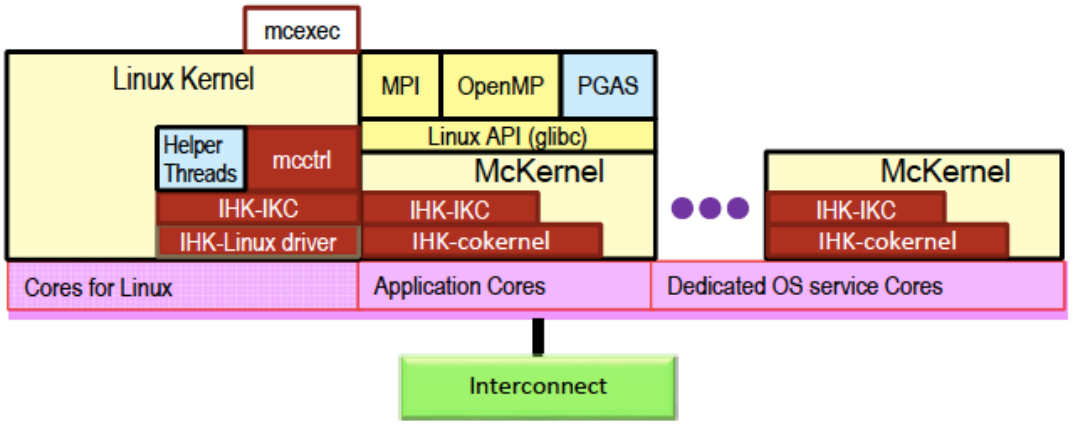


Fig. 4. Partitioning of cores and memory for Linux applications.

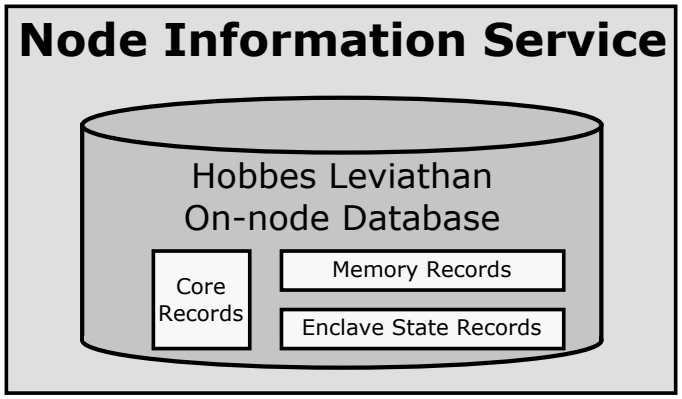
MAHOS (2013)



Leviathan Node Manager

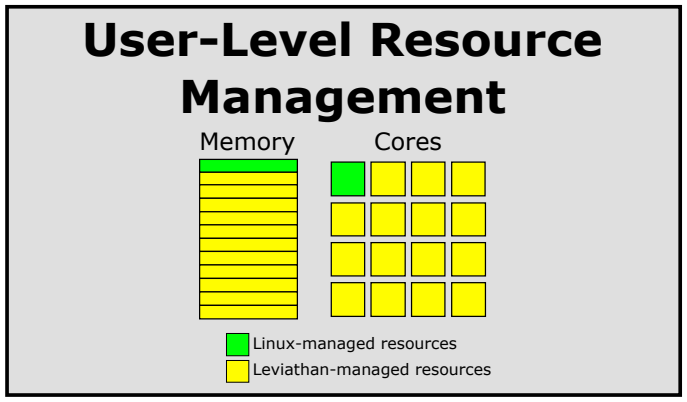
- Compute node resources tracked via an in-memory NoSQL database (WhiteDB)
  - Records for cores, memory, NICs, NUMA info, …
  - Records for system service state, name services, enclave state, ...
- User-level given explicit control of physical resources
  - Resources space partitioned into multiple enclaves
  - Libhobbes.a provides C API, translates under covers to DB operations
  - Provides flexibility vs. traditional OS “one-size-fits-all” approach
- Mechanisms for inter-enclave composition
  - XEMEM for cross-enclave memory sharing (extended version of XPMEM)
  - XASM provides memory snapshot sharing via COW (extends XEMEM)
  - Libhobbes.a provides global ID allocation, name services, command queues, and generic RPC mechanisms
  - Host I/O layer allows flexible routing of system calls between enclaves (e.g., Kitten app routing its system calls to a Linux driver VM)

Leviathan On-Node Manager Ties Things Together

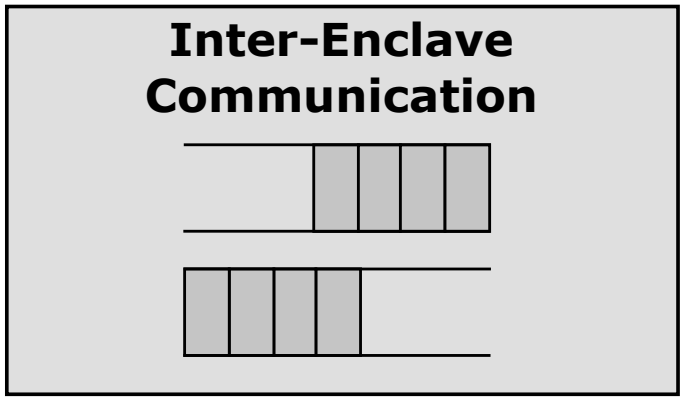


State of all resources tracked in in-memory NoSQL database

The Leviathan shell provides commands to form enclaves and launch applications

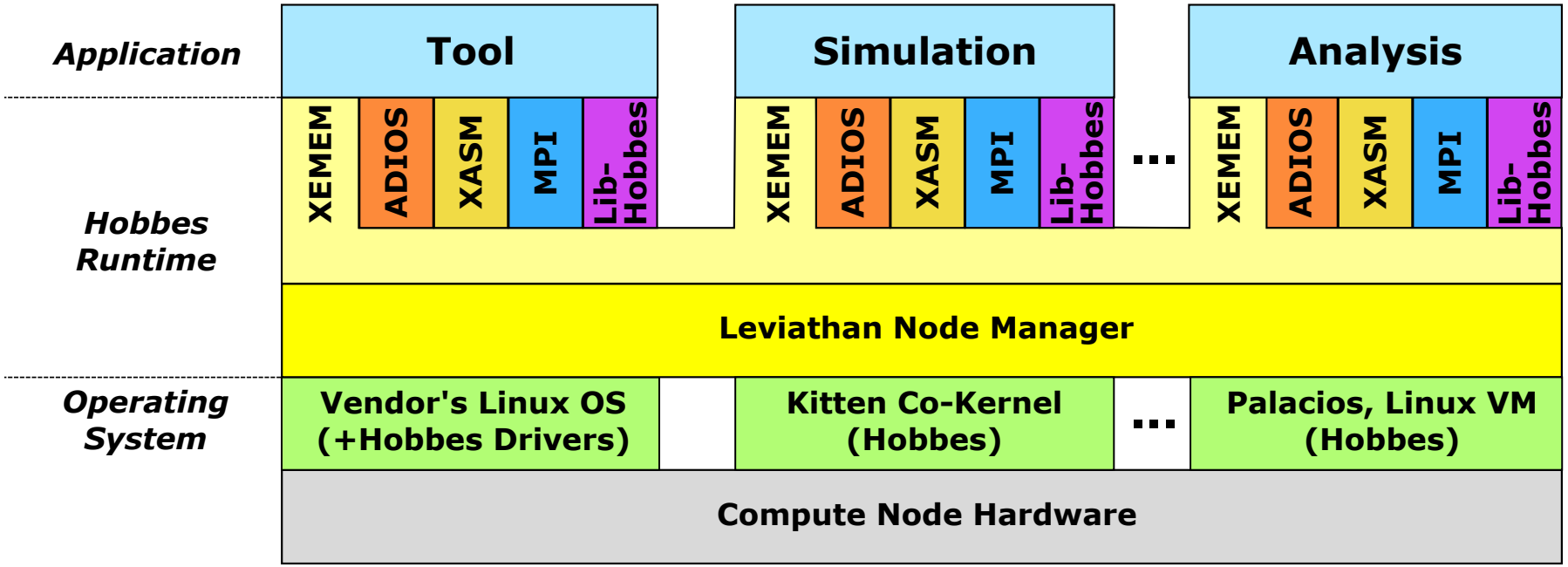


User-level has explicit control of physical resources managed by Leviathan

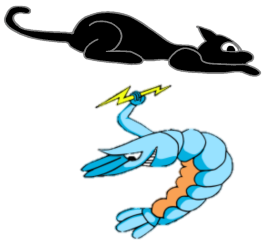


Built-in services for command queues, discovery, global IDs, and generic RPC

The Hobbes Node Virtualization Layer



- Leverages experience with Kitten Lightweight Kernel and Palacios Virtual Machine Monitor
- Provides Infrastructure for application composition
  - Complement's vendor's Linux stacks, adds capability to it
  - Enables OS/R stack functionality through enclaves
  - Provides low-level mechanisms for cross-enclave composition
- Team: U. Pittsburgh, Northwestern, UNM, LANL, ORNL



http://xstack.sandia.gov/hobbes