Hobbes

Composition and Virtualization as the Foundations of an Extreme-scale OS/R

Ron Brightwell, Ron Oldfield, Arthur B. Maccabe, David E. Bernholdt

ROSS '13
June 2014

Motivation

Exascale challenges (U.S. Department of Energy)

- Massive (dynamic) parallelism
- Heterogeneity
- Deep memory hierarchies
- Power/Energy limitations
- Fault-proneness

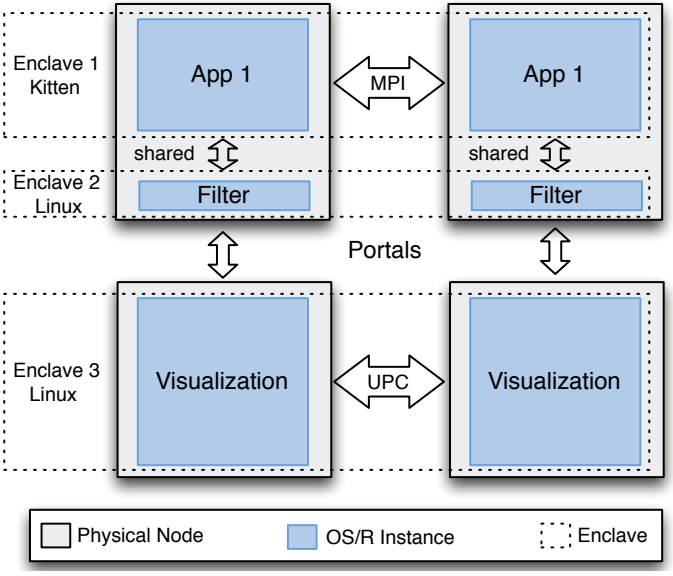
$\Rightarrow$ Largely driven by limitations in hardware technology

Motivation II

Application composition

- Exploratory Analytics
- Streaming Analytics
- Graph Analytics
- Code Coupling
- Application Frameworks

Application composition for “Exploratory Analytics”



Design Goals/Influences

- Hardware (expected ~2020)
  - Variety of processors: load/store, streaming, near-/in-memory
  - Variety of memory: bandwidth, latency, energy, ...
  - High-speed inter-node network
  - Customizable monitor and control

Design Goals/Influences

- Hardware (expected ~2020)
  - Variety of processors: load/store, streaming, near-/in-memory
  - Variety of memory: bandwidth, latency, energy, ...
  - High-speed inter-node network
  - Customizable monitor and control
- Novel usage models
  - Co-location
  - Global addressing + massive multithreading
  - Event-based processing

Design Goals/Influences

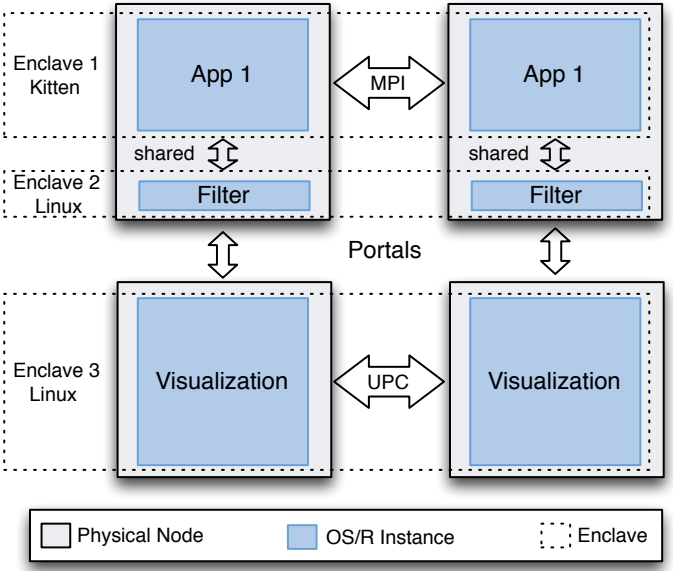
- Hardware (expected ~2020)
  - Variety of processors: load/store, streaming, near-/in-memory
  - Variety of memory: bandwidth, latency, energy, ...
  - High-speed inter-node network
  - Customizable monitor and control
- Novel usage models
  - Co-location
  - Global addressing + massive multithreading
  - Event-based processing
- Programming environments and tools
- Legacy applications
- Scalability (down)

Enclaves

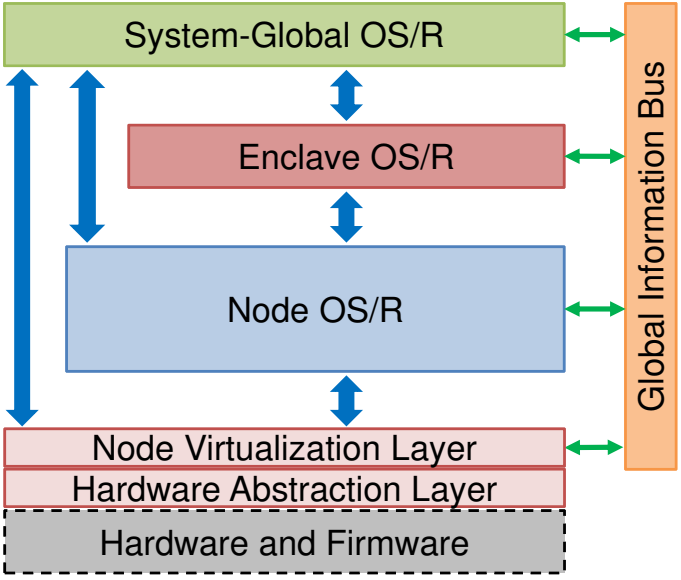
- Different runtime requirements: minimal . . . Linux
- Application composition usually enforces common runtime environment

> "An enclave (i.e., partition) is a set of resources dedicated to one application or one service. To the greatest possible extent, system functionality is encapsulated within enclaves. In particular, the failure of an enclave should not cause global system failure, and different enclaves can provide different implementations of the same function. "

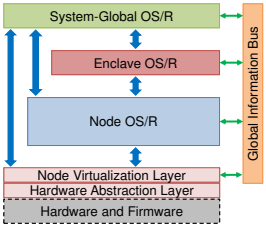
Application composition for “Exploratory Analytics”



Main Components

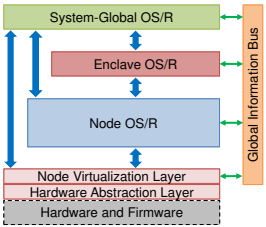


Main Components



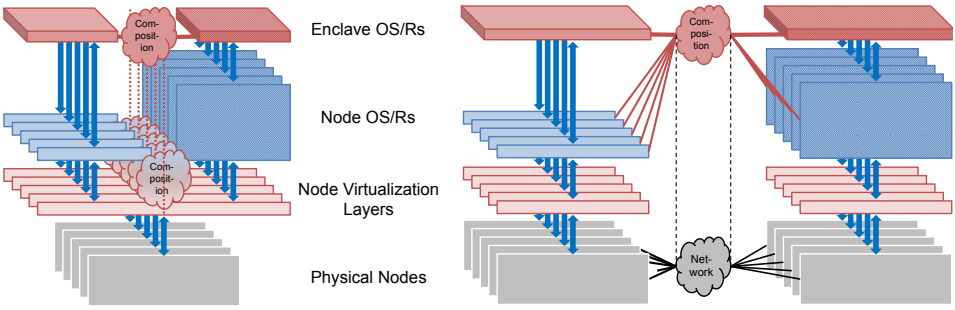
- System-Global OS
  - Scheduling, monitoring, resource management
  - Mapping of enclaves on nodes
  - Requires richer job specification (OS/R selection, mapping hints, ...)
- Enclave OS/R
  - Control (launch, terminate, pause, resume) an enclave and handle dynamic resource addition/removal
  - Composition of enclaves: "selective breaking of isolation"

Main Components



- Node OS/R
  - Abstract interface to underlying compute, memory, and network resources
  - Support SGOS and EOS in higher-level resource management
- Node Virtualization Layer: support for...
  1. Bare-metal applications
  2. Full-feature OSes
  3. "Virtual nodes" to host multiple enclaves

Basic Enclave Composition



(a) Intra-node composition

(b) Inter-node composition

“Crosscutting Areas”

- Power and energy: measurement and management
- Scheduling: coordination between layers
- Resilience: mask/handle faults, resilient OS/R data structures

Conclusion

Summary

- Hobbes: extreme-scale OS/R and "playground"
- Design based on anticipated trends in hard- and software
- Explicit support for application composition
- Virtualization for flexibility and co-location

Conclusion

Summary

- Hobbes: extreme-scale OS/R and "playground"
- Design based on anticipated trends in hard- and software
- Explicit support for application composition
- Virtualization for flexibility and co-location

Discussion

- Similarities to DoE's "Reference Architecture"
- Hardware expectations vs. current HPC systems
- Vague in some points (EOS vs. NOS vs. NVL)
- Implications for FFMK?