

# Hobbes: OS and Runtime Support for Application Composition

Ron Brightwell  
Coordinating PI

OS/R Kickoff Meeting  
Chicago, IL  
August 15, 2013



*Exceptional  
service  
in the  
national  
interest*



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.

# Outline

- Team
- History
- Goals
- Motivation
- Activity areas
- Alignment with existing DOE OS/R R&D investments

# Hobbes Team

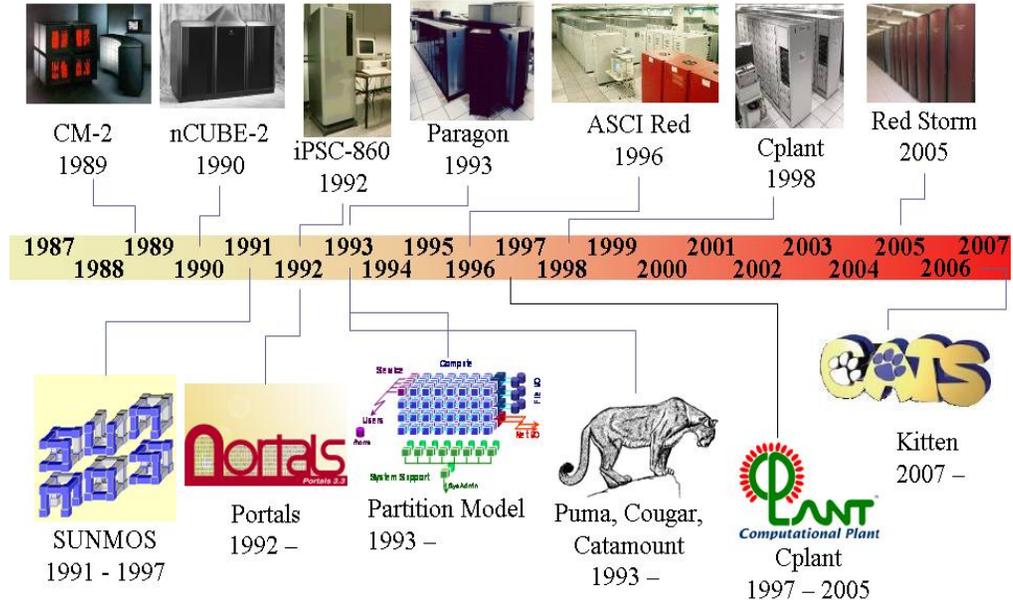
Institution	Person	Role
Georgia Institute of Technology	Karsten Schwan	PI
Indiana University	Thomas Sterling	PI
Los Alamos National Lab	Mike Lang	PI
Lawrence Berkeley National Lab	Costin Iancu	PI
North Carolina State University	Frank Mueller	PI
Northwestern University	Peter Dinda	PI
Oak Ridge National Laboratory	David Bernholdt	PI
Oak Ridge National Laboratory	Arthur B. Maccabe	Chief Scientist
Sandia National Laboratories	Ron Brightwell	Coordinating PI
University of Arizona	David Lowenthal	PI
University of California – Berkeley	Eric Brewer	PI
University of New Mexico	Patrick Bridges	PI
University of Pittsburgh	Jack Lange	PI

# System Software@Sandia

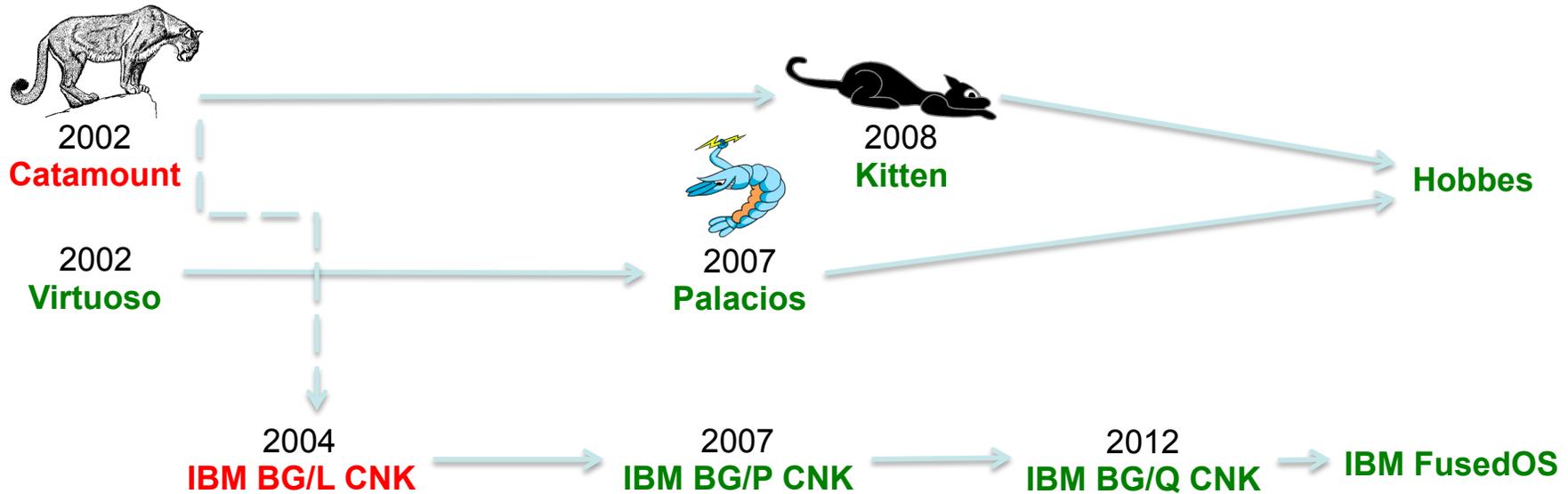
- Established the functional partition model for HPC systems
  - Tailor system software to function (compute, I/O, user services, etc.)
- Pioneered the research, development, and use of lightweight kernel operating systems for HPC
  - Only DOE lab to deploy OS-level software on large-scale production machines
  - Provided blueprint for IBM BlueGene OS
- Set the standard for scalable parallel runtime systems for HPC
  - Fast application launch on tens of thousands of processors
- Significant impact in the design and of scalable HPC interconnect APIs
  - Only DOE lab to deploy low-level interconnect API on large-scale production machines

## AWARDS:

- 1998** Sandia Meritorious Achievement Award, TeraFLOP Computer Installation Team
- 2006** Sandia Meritorious Achievement Award, Red Storm Design, Development and Deployment Team
- 2006** NOVA Award Red Storm Design and Development Team
- 2009** R&D 100 Award for Catamount Multi-Core Light Weight Kernel
- 2010** Excellence in Technology Transfer Award, Federal Laboratory Consortium for Technology Transfer
- 2010** National Nuclear Security Administration Defense Programs Award of Excellence



# Lightweight Kernel Timeline



**Green = Open Source**  
**Red = Closed Source**

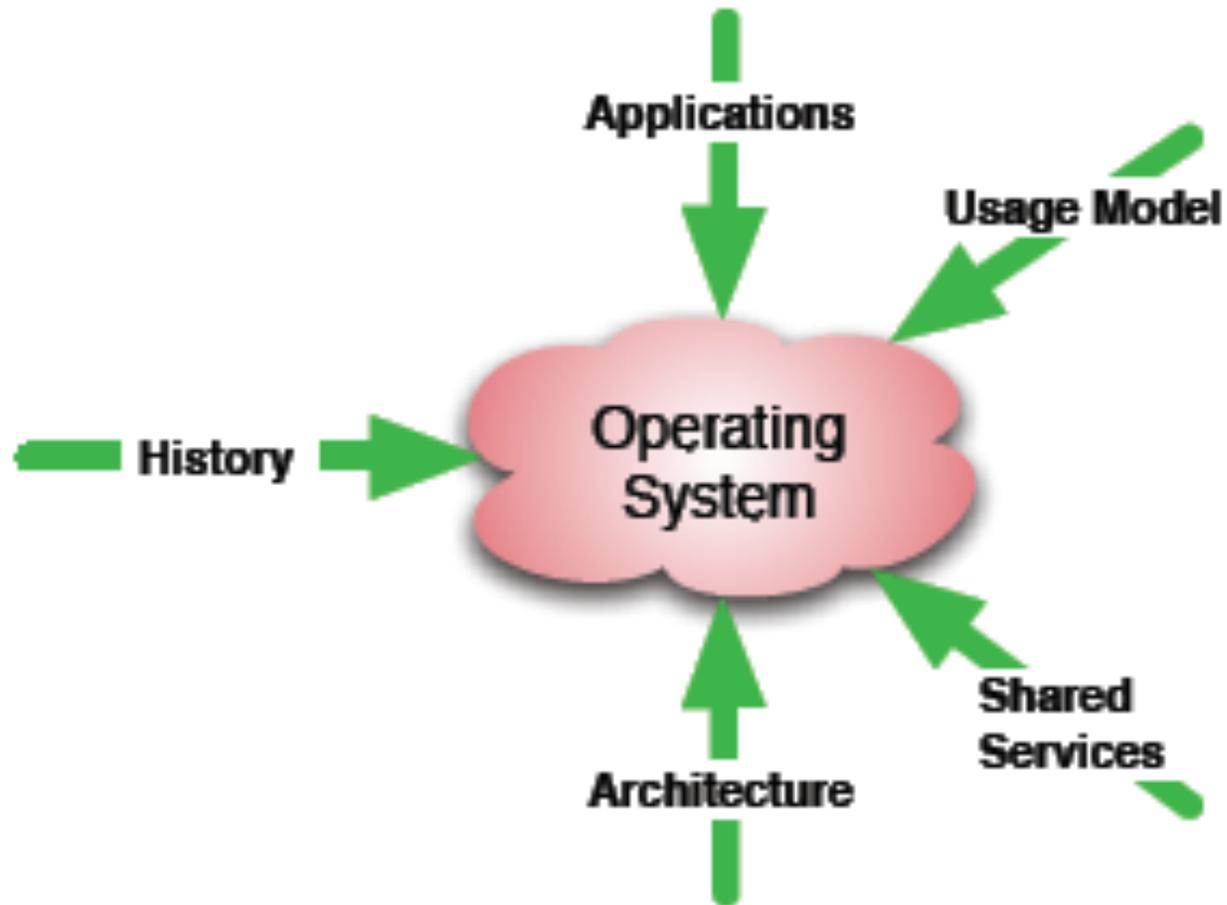
- Kitten and CNK similar in concept
  - Both support Linux API subset and ABI compatibility
  - Kitten targets x86 (ARM underway), CNK targets PowerPC only
  - Kitten leverages Linux source code, CNK uses no Linux source code

- Palacios and Xen are both hypervisors
  - Palacios designed to be embeddable in a host OS, Kitten or Linux
  - Palacios is designed for HPC, low overhead, predictable performance
  - Palacios targets x86, Xen targets x86 + other archs

# Project Goals

- Deliver 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
  - 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
- Leverage Kitten lightweight kernel and Palacios lightweight virtual machine monitor
  - Enable high-risk high-impact research in virtualization, energy/power, scheduling, and resilience

# Factors Influencing OS Design



# Exascale Focus on Hardware

- Reliability/Resilience
- Power/Energy
- Heterogeneity
- Memory hierarchy
- Cores, cores, and more cores
  
- Risk
  - Hardware advancements and investments can provide orders of magnitude improvement
  - OS/R advancements can provide double-digit percentage improvement

# Application Focus on Programming Models

- Dealing with effects of many-core
  - Advanced runtime systems
  - Node-level resource allocation and management
  - Managing locality
  - Extracting parallelism
  - Introspective, dynamic, adaptive capabilities
- Risk
  - Incremental approach (OpenMP) wins
    - Advanced runtime capabilities are overkill
  - No clear on-node parallel programming model winner
    - Difficult to optimize OS/R

# OS/R is Enabling Technology

- Need to support advanced run-time systems and approaches to resilience and power/energy, not necessarily provide solutions
- BASF mantra
  - We don't make it - we make it possible
- OS/R should focus on providing capability, not just overcoming limitations of current hardware
- Application composition is the responsibility of the OS/R
  - Capability will be required regardless of underlying hardware or overlying parallel programming model

# Application Composition Will Be Increasingly Important at Extreme-Scale

- More complex workflows are driving need for advanced OS services and capability
  - Exascale applications will continue to evolve beyond a space-shared batch scheduled approach
- HPC application developers are employing ad-hoc solutions
  - Interfaces and tools like mmap, ptrace, python for coupling codes and sharing data
- Tools stress OS functionality because of these legacy APIs and services
- More attention needed on how multiple applications are composed
- Several use cases
  - Ensemble calculations for uncertainty quantification
  - Multi-{material, physics, scale} simulations
  - In-situ analysis
  - Graph analytics
  - Performance and correctness tools
- Requirements are driven by applications
  - Not necessarily by parallel programming model
  - Somewhat insulated from hardware advancements

# Multiphysics Example



## Technical Discussion on CASL: Why is Multiphysics Coupling Difficult?

- The most complex software engineering project I have been involved with
  - Fortran, C, C++, Java, Python, Perl, ...
  - 21 git repositories
  - VERA is composed of 350+ software engineering packages, 12 TPLs
- Multiscale physics: Thermal hydraulics (CFD, Subchannel), Neutron transport (SN, MOC), materials models, crack propagation, multiphase boiling, ...
- Multiple discretizations and solution algorithms
  - Steady-state, transient (explicit, operator split, implicit), pseudo-transient, continuation, eigensolvers, etc...
  - CVFEM, FE, DGFEM, DAE network models, ...
  - Stability and Conservation are critical
- Code use different units, coordinate systems, dimensions, pin axis alignment
- **Software engineering quality of individual codes: app → library = disaster!**

Code integrations require a strong combination of skills in physics simulation, numerical algorithms and software engineering



# Multiphysics Example (cont'd)



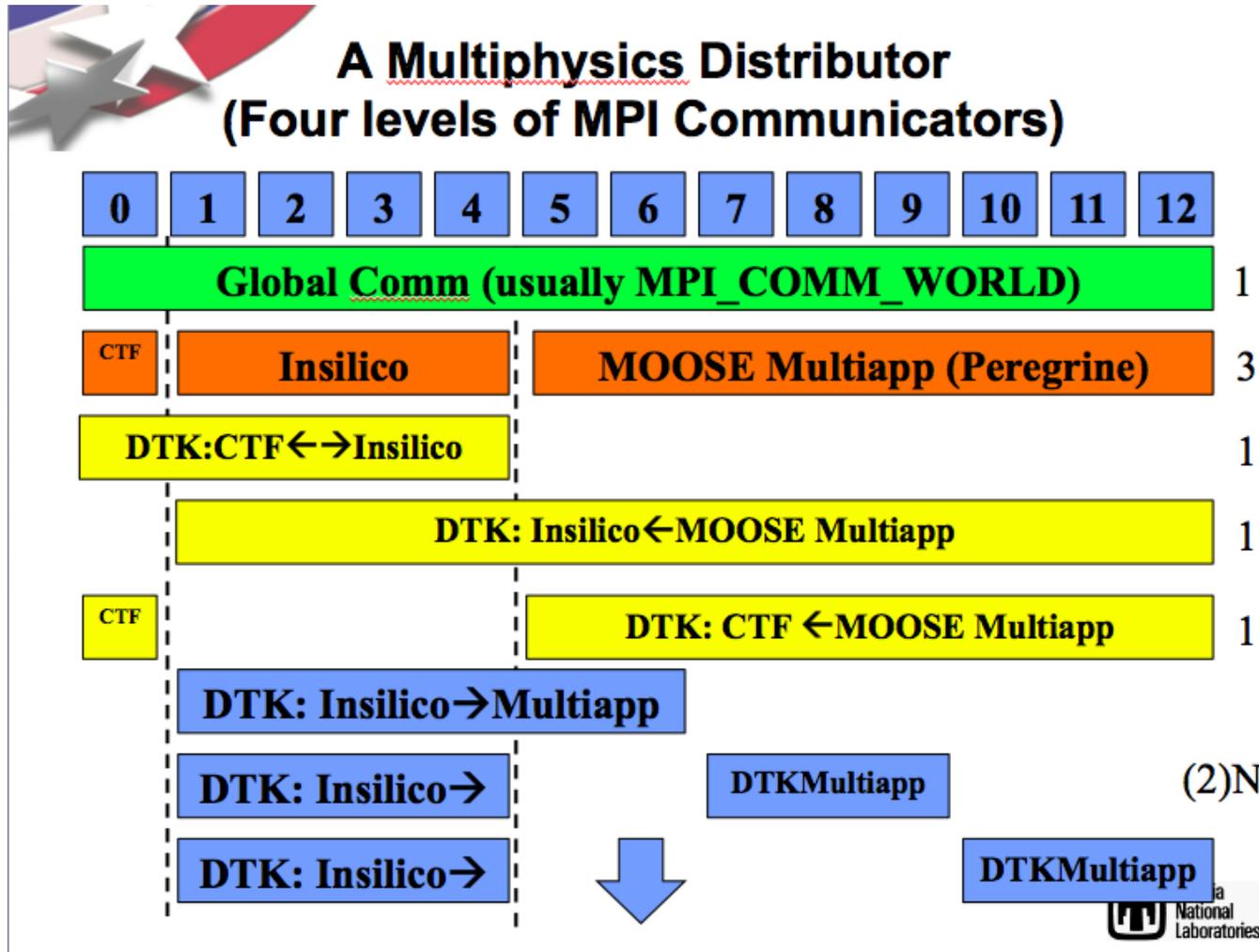
## Peregrine/Insilico/CTF Executable (Only ONE of many executables in VERA)

---

- VRIPSS
- COBRA-TF
- Exnihilio (Insilico, Denovo, nemesis)
- Drekar
- MOOSE/Peregrine
- Qt
- SCALE (200+ libraries, 30+ years of NRC codes)
- LIBMESH
- Data Transfer Kit
- LIME
- Trilinos (35+ libraries)
- PETSc
- HYPRE
- Netcdf
- HDF5
- Boost
- Many others...

**We are pulling in  
almost every  
general HPC  
library under one  
executable and  
dealing with  
massive collisions!**

# Multiphysics Example (concl'd)



# Project Activity Areas

<b>Activity (Section)</b>	<b>Coordinator</b>	<b>GaTech</b>	<b>IU</b>	<b>LANL</b>	<b>LBNL</b>	<b>NCSU</b>	<b>NWU</b>	<b>ORNL</b>	<b>Pitt</b>	<b>SNL</b>	<b>UA</b>	<b>UCB</b>	<b>UNM</b>
Enclave OS/R	D. E. Bernholdt	-	-	✓	✓	-	✓	<b>L</b>	✓	✓	-	✓	-
Node Virt. Layer	K. Pedretti	-	-	✓	-	-	✓	✓	✓	<b>L</b>	-	-	✓
Global Info. Bus	R. Oldfield	-	-	✓	✓	-	-	✓	-	<b>L</b>	-	-	✓
Energy/Power	J. Laros	✓	-	-	✓	-	✓	✓	-	<b>L</b>	✓	-	-
Scheduling	T. Jones	✓	-	-	-	-	-	<b>L</b>	-	-	✓	✓	✓
Resilience	C. Engelmann	-	-	-	-	✓	-	<b>L</b>	-	✓	✓	-	✓
Prog. Models	C. Iancu	-	✓	✓	<b>L</b>	-	✓	-	-	-	-	-	-

# Hobbes Complements Existing DOE OS/R R&D Investments

- DOE/ASCR
  - XPRESS
    - Deliver prototype system software stack that instantiates ParalleX model
    - HPX runtime system
    - Kitten enhancements for dynamic adaptive runtime system
  - Scalable System Virtualization
    - Exascale hardware/software design with Kitten/Palacios
- DOE/ASC CSSE
  - Scalable Interconnects project
    - Portals 4 for scalable application and system network services
  - Simulation Tools project
    - Kitten running on SST
  - Software and Tools for Scalability and Performance
    - Kitten port to ARM
- Sandia LDRD
  - Exascale Grand Challenge
    - Kitten+Qthreads+Portals 4 for unified simulation and analysis architecture

# Accomplishments So Far

- Ron Brightwell, et al. “Hobbes: Composition and Virtualization as the Foundations of an Extreme-Scale OS/R,” in Proceedings of the Workshop on Runtime and Operating Systems for Supercomputers, June 2013.
- Pete Beckman. “Argo: An Exascale Operating System and Runtime,” Invited Talk, Workshop on Runtime and Operating Systems for Supercomputers, Jun 2013.
- Ron Brightwell, Patrick Bridges, Terry Jones. “Hobbes: Operating System and Runtime Research for Extreme Scale,” INCITE Proposal for 36 million core hours on Titan.

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

Ron Brightwell, Ron Oldfield  
Sandia National Laboratories  
Center for Computing Research  
Albuquerque, NM  
(rbrighn,raoldfj@sandia.gov)

Arthur B. Maccabe, David E. Bernholdt  
Oak Ridge National Laboratory  
Computer Science and Mathematics Division  
Oak Ridge, TN  
(maccabe,bernholdtde@ornl.gov)

### Categories and Subject Descriptors

D.4.7 [Operating Systems]: Organization and Design; C.5.1 [Computer System Implementation]: Super (very large) computers; C.1.2 [Multiprocessors]: Parallel Processors

### Keywords

operating system, supercomputing, virtualization, application composition

### ABSTRACT

This paper describes our vision for Hobbes, an operating system and runtime (OS/R) framework for extreme-scale systems. The Hobbes design explicitly supports application composition, which is emerging as a key approach for applications to address scalability and power concerns anticipated with coming extreme-scale architectures. We make use of virtualization technologies to provide the flexibility to support requirements of application components for different node-level operating systems and runtimes, as well as different mappings of the components onto the hardware. We describe the architecture of the Hobbes OS/R, how we will address the cross-cutting concerns of power/energy, scheduling of massive levels of parallelism, and resilience. We also outline how the “users” of the OS/R (programming models, applications, and tools) influence the design.

### 1. INTRODUCTION

Application composition is a critical capability that will be the foundation of the way extreme-scale systems must be used in the future. The high-performance computing (HPC) community is already seeing the need for tighter integration of modeling and simulation with advanced analysis, and ad hoc solutions for coupling multiple simulations as well as integrating simulation and analysis are being developed and deployed. These ad hoc approaches are often hindered by system interfaces not designed to provide the full semantic

capability required, making it difficult to deliver scalable high-performance implementations.

A recent workshop report [4] published by the U. S. Department of Energy (DOE) describes the challenges facing OS/Rs for future extreme-scale scientific computing systems. Many of these challenges, such as the need for increased reliability and the desire to reduce power and energy use, are largely driven by limitations in hardware technology, and the computer architecture community is vigorously pursuing potential approaches and solutions. While the OS/R is an important component in exploiting hardware-based solutions, addressing near-term hardware limitations without considering application composition will lead to incomplete solutions. A more effective approach is to consider these challenges in the context of application composition and to allow for integration of the hardware or algorithmic solutions required to address them.

This paper outlines our vision for an OS/R that enables application composition, and leverages a lightweight virtualization capability to provide a system software infrastructure for future extreme-scale scientific computing platforms. The rest of this paper is organized as follows. Sec. 2 provides background on the important factors that influence OS/R design and how each of these factors influences our approach to application composition. Sec. 3 describes the scenarios that motivate the need for more advanced support for application composition on current and future extreme-scale systems. Sec. 4-7 describe our approach, based on lightweight virtualization, the architecture of the OS/R framework, the cross-cutting concerns we consider in the design, and how we incorporate the needs of the various “users” of the OS/R stack into our design. Finally, we summarize the key ideas and contributions of this paper in Sec. 8.

### 2. BACKGROUND

In this section, we provide our perspective of the key considerations that impact the design and development of the OS/R, briefly describe previous work on supporting application-specific OS/R functionality, and discuss the rationale for a development process appropriate for extreme-scale systems.

#### 2.1 Factors Influencing OS Design

Despite the rapidly changing landscape of HPC, OS/R design continues to be influenced by a small number of factors. We describe each of these factors in terms of extreme-scale parallel-computing platforms and the challenges they present, considering application composition and the perspective that the operating system needs to enable explo-

(c) 2013 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the United States government. As such, the United States Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only. ROSS '13, June 10, 2013, Eugene, Oregon, USA  
Copyright 2013 ACM 978-1-4503-2146-4/1306 ...\$15.00.