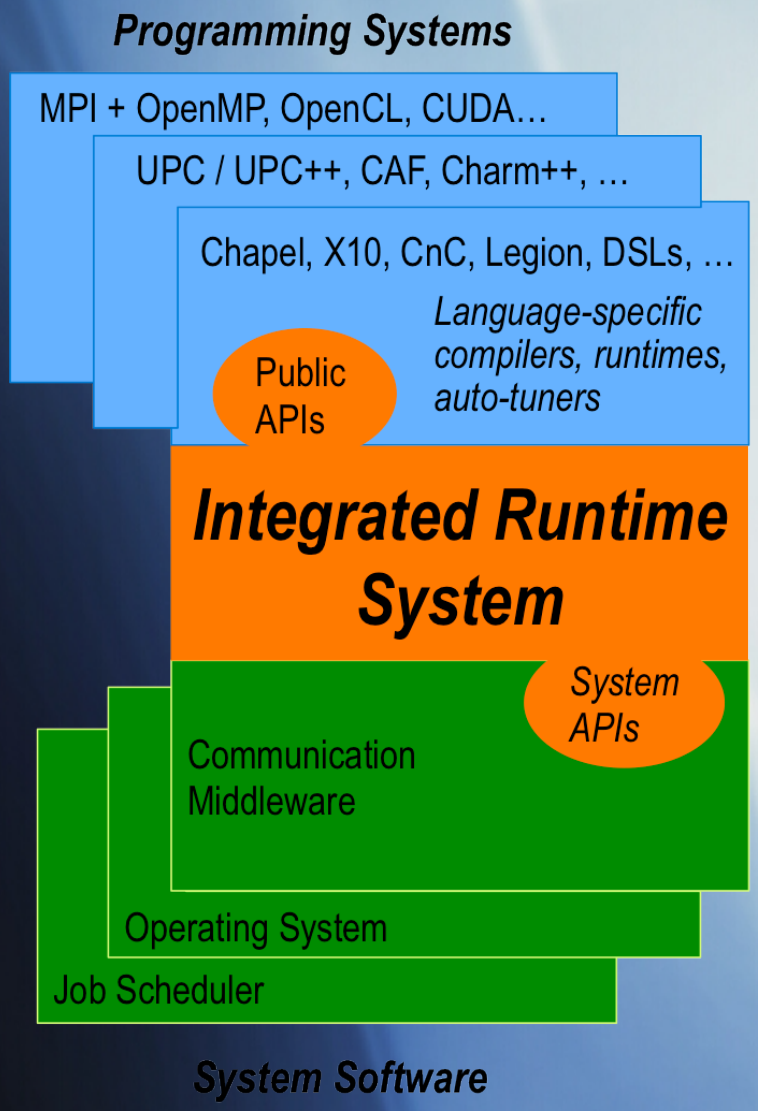
RICE George R. Brown School of Engineering
Computer Science



X-Stack Runtime Summit

Vivek Sarkar
E.D. Butcher Chair in Engineering
Professor of Computer Science
Rice University
vsarkar@rice.edu

Outline

- Runtime Summit
- My take-away
- Today's runtime-related events
  - Runtime Systems Panel
  - Runtime demonstrations in Technology Marketplace

Runtime Summit, April 9th, 2014, Washington DC

- Attendees
  - Ron Brightwell, Andrew Chien, Maya Gokhale, Sanjay Kale, Milind Kulkarni, Rich Lethin, Wilf Pinfold, Sonia Sachs, Vivek Sarkar, Vijay Saraswat, Thomas Sterling, Kathy Yelick
- Meeting Summary
  - https://xstackwiki.modelado.org/Runtime_Systems

Summit Goals

- **Generate a roadmap for achieving a unified runtime systems *architecture* for Exascale systems**
  - Reach consensus on the top six challenges and solutions for them.
  - Agree on a comprehensive set of questions that must be answered in order to achieve such architecture
  - Current known answers to posed questions
- **Generate a roadmap for a research program on runtime systems**
  - Consistent with achieving a unified runtime systems architecture
- **Discuss future workshop**
- **Prepare for writing a report**

Summit rule: Participants should not promote their current research agenda

Discussion themes

- Identification of runtime challenges
- Runtime abstractions ➔ service-oriented views
- Identification of challenge problems for runtime systems
  - e.g., multi-physics applications
  - Full generality of runtime systems may not be needed in DOE applications?
- Community discussion
  - Identifying the community
  - Ecosystem creation
  - Strong voices for shared implementations of well-understood runtime components
  - Strong voices that it is too early to standardize runtime components

RICE

Outline

- Runtime Summit
- My take-away
- Today's runtime-related events
  - Runtime Systems Panel
  - Runtime demonstrations in Technology Marketplace

Runtime Systems are critical for Exascale Computing

- Exascale challenges necessitate *dynamic* creation and mapping of parallel computations and data accesses, while leveraging static approaches where applicable
  - Billion-way parallelism --- overdecomposition, point-to-point synchronization, dag parallelism, asynchronous collectives, …
  - Heterogeneity --- thin cores, fat cores, accelerators, scratchpad, fast DRAM, slow DRAM, NVRAM, …
  - Energy optimization --- minimization of data movement, communication avoidance, energy-aware resource allocation, …
  - Resilience --- fault tolerance, hierarchical decomposition, …
  - Programmability --- forward-scalable expression of parallelism and locality in face of extreme heterogeneity & variability, …

We have accumulated extensive experience with multiple Runtime Systems in the Community …

- Charm++
- GASNet
- UPC
- Chapel
- X10
- Habanero
- HPX
- OCR
- . . .

Lots of documented results that show benefits of new runtimes over state of the art

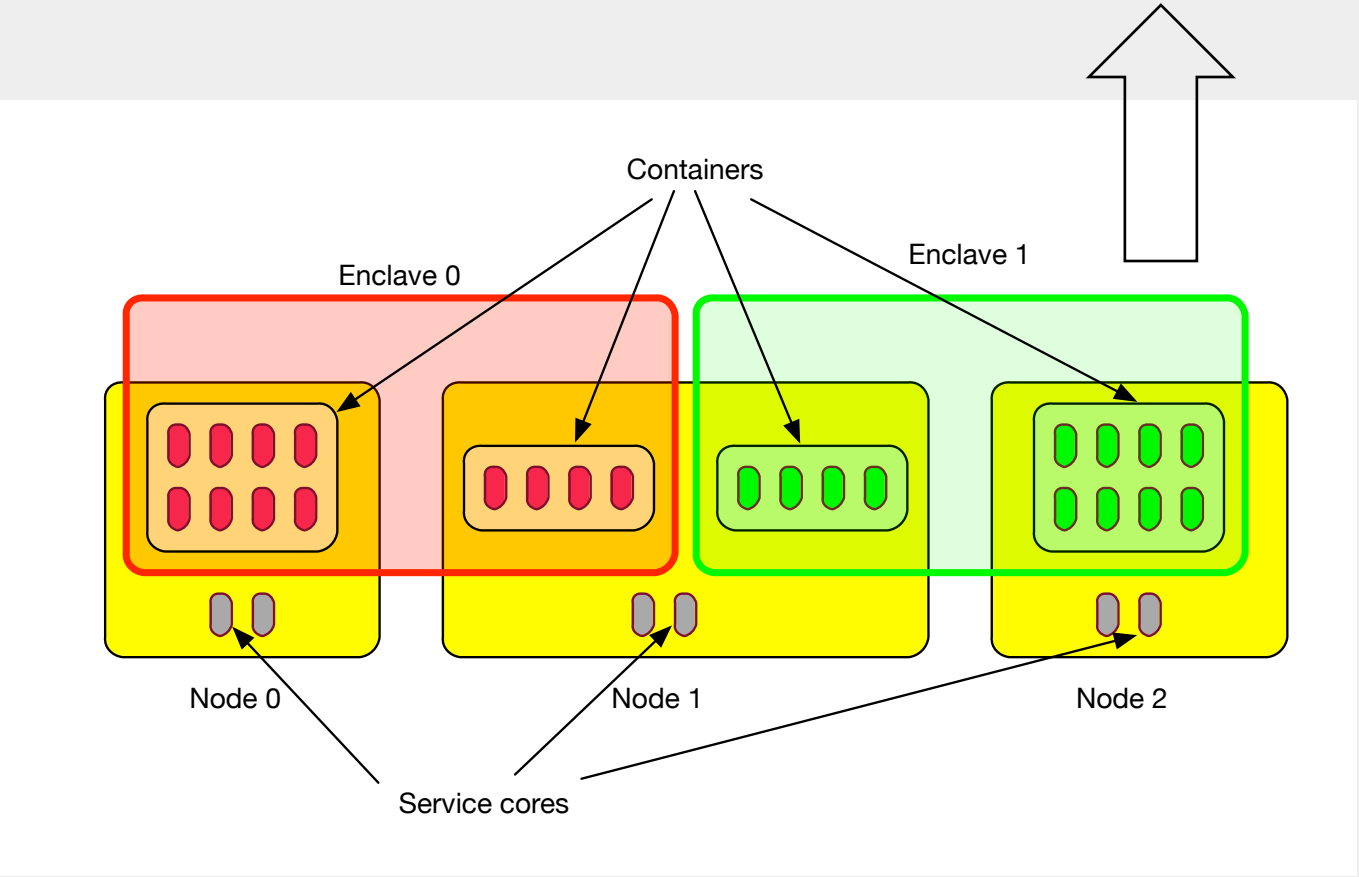
RICE

… with many ideas in common, but differences in what we call them

- Dag parallelism
  - async, finish, spawn, sync, futures, codelets, event-driven tasks, …
- Distributed data structures
  - distributed arrays. …
- Point-to-point data transfers, reductions, synchronization
  - put, get, send, reduce, accumulators, …
- Computation and data hierarchies
  - places, enclaves, containers, containment domains, …
- Mutual exclusion
  - actors, isolation, transactions, …
- Resilience
  - idempotent tasks, data/compute migration, …

Analogy: OS/R view of hardware

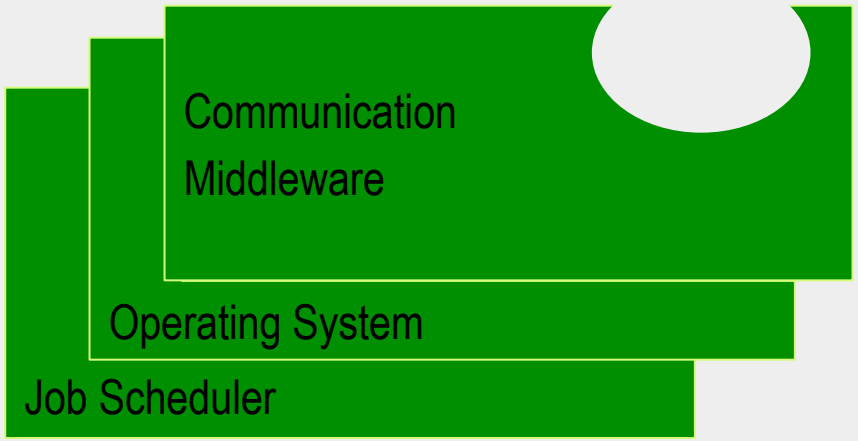
X-Stack hardware abstraction



➔ X-Stack programming systems need a similar abstraction of the X-Stack runtime

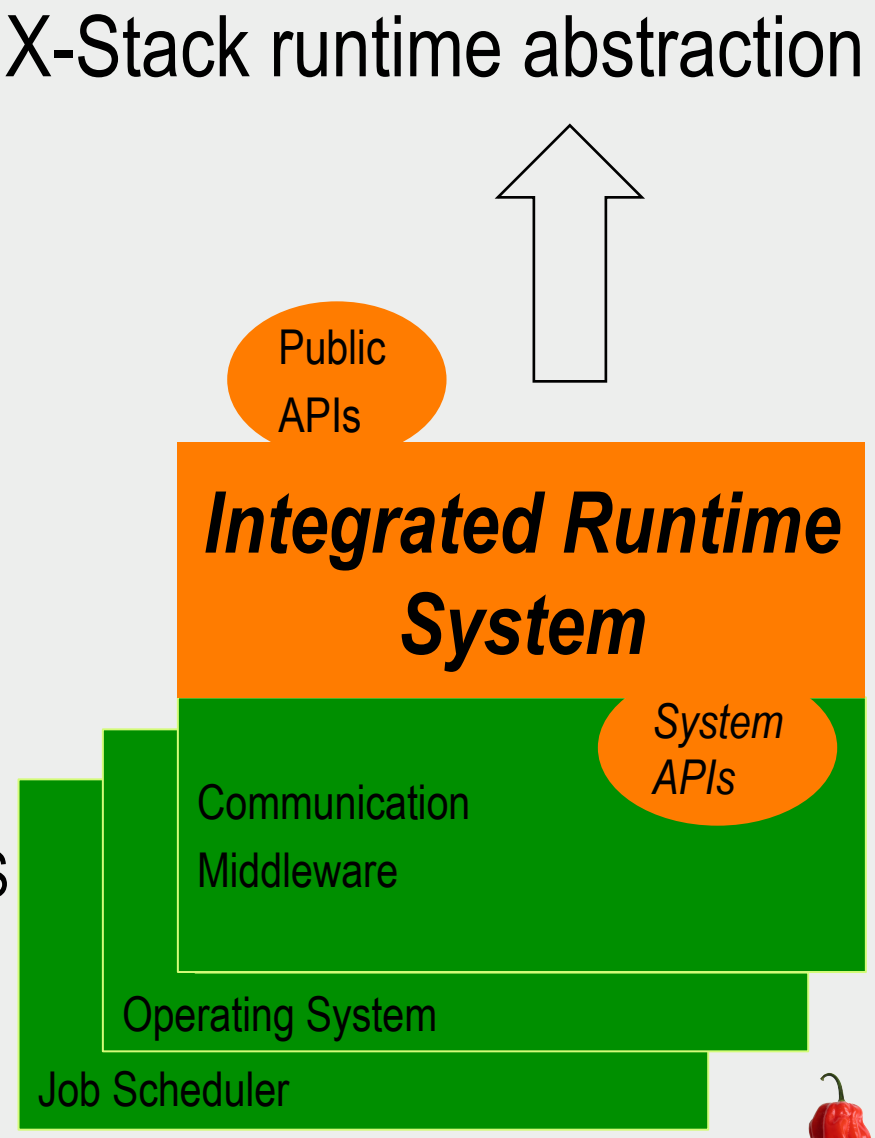
Runtime Vision --- Setup

- Application is submitted to job scheduler with desired resource ranges (nodes, cores, memory, bandwidth, disk, …)
- Job scheduler selects initial “enclave” of dedicated/shared resources, and initializes parallel environment with parallel OS and low-level communication middleware
- OS launches application across multiple containers in enclave



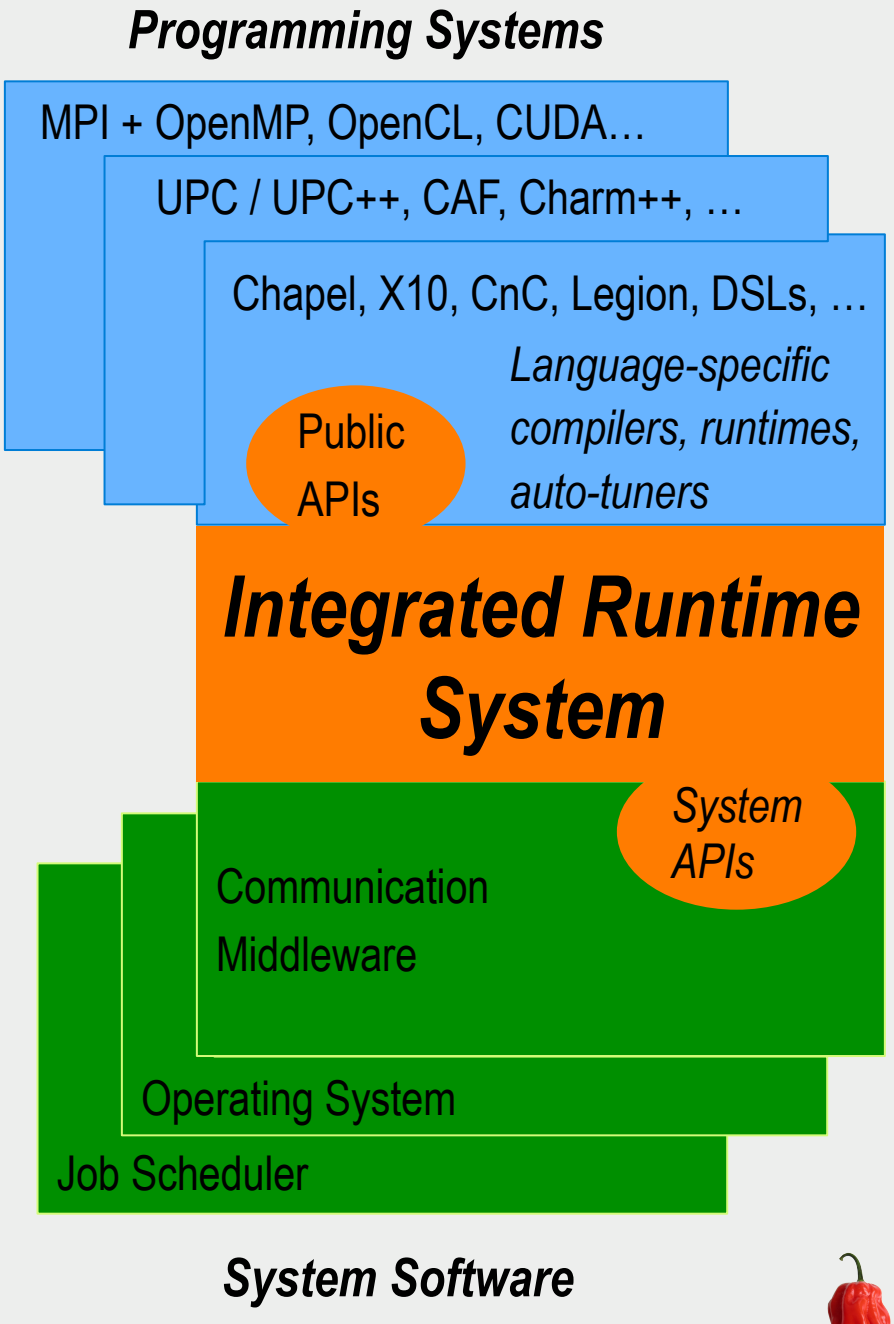
Runtime Vision --- Scope

- An integrated user-level runtime system launches worker threads on cores, and initializes all resources available in enclave (memory, bandwidth, disk)
- Integrated runtime system is both "application-facing" and "machine-facing"
- Runtime is adaptive --- adjusts computation and data mappings to address exascale challenges
- Runtime is reactive --- responds to OS requests to give up resources and informs OS when it can use more resources



Runtime Vision --- Impact

- Integrated runtime system enables a wide range of programming environments on a wide range of system software
- Some capabilities needed in integrated runtime are better understood …
  - User-level threads
  - Event-driven tasks
  - Scalable memory allocators
  - Accelerator offload
- … compared to others
  - Fine-grained resilience management
  - Tightly coupled OS-RT interactions



Outline

- Runtime Summit
- My take-away
- Today's runtime-related events
  - Runtime Systems Panel
  - Runtime demonstrations in Technology Marketplace

Runtime Systems Panel

- Panelists: David Grove, Sanjay Kale, Kevin Pedretti, Wilf Pinfold, Kathy Yelick
- Moderator: Vivek Sarkar
- Format:
  - Short position statements by panelists (at most 3 minutes, 1 slide)
  - Questions from audience
  - See https://xstackwiki.modelado.org/Runtime_Research_Questions for possible topics

Runtime Demonstrations in Technology Marketplace

- APGAS Runtime, David Grove (IBM)
- Charm++, Sanjay Kale (UIUC)
- Habanero-C++ and UPC++, Vivek Kumar (Rice)
- OCR and CnC, Vincent Cave (Rice) & Romain Cledat (Intel)
- SemCache, Milind Kulkarni (Purdue)