UNM SCHOOL of ENGINEERING
Department of Computer Science

Virtualization-based Exascale Runtime System R&D

Patrick G. Bridges

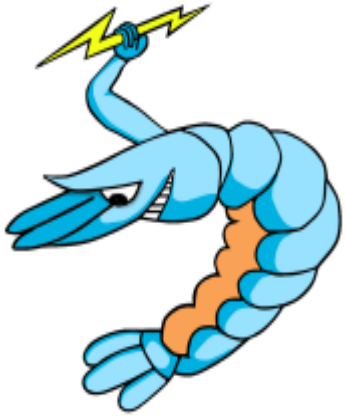
Exascale demands high quality runtimes and system software

- Bridge the architecture, programming model, and application
- Needs to understand what to bridge and how to do it well
- Stuck in the middle
  - Dictatorial operating systems on one side
  - Shifting architectural and programming models on the other

Virtualization-based Runtime Innovation

- "Why would you want to use virtualization in HPC?"
  - P. Bridges
- Ideal for Exascale runtime development
  - Overthrow (or co-opt) dictatorial OSes
  - Platform for OS/Runtime innovation
  - Compatibility
  - Essential tool for design space exploration

Palacios Virtual Machine Monitor



- Simple VMM for HPC systems
- Integrates into existing host OSes
- Funding
  - Initial: NSF, Oak Ridge LDRD, Sandia participation
  - Current: DOE ASCR X-Stack award
- Participants
  - UNM: Bridges, Arnold
  - Northwestern: Dinda, Joseph
  - Pittsburgh: Lange
  - Sandia: Pedretti, Brightwell
  - ORNL: Scott, Naughton
- Software, papers, available for download from http://www.v3vee.org

Sandia National Laboratories

Oak Ridge National Laboratory

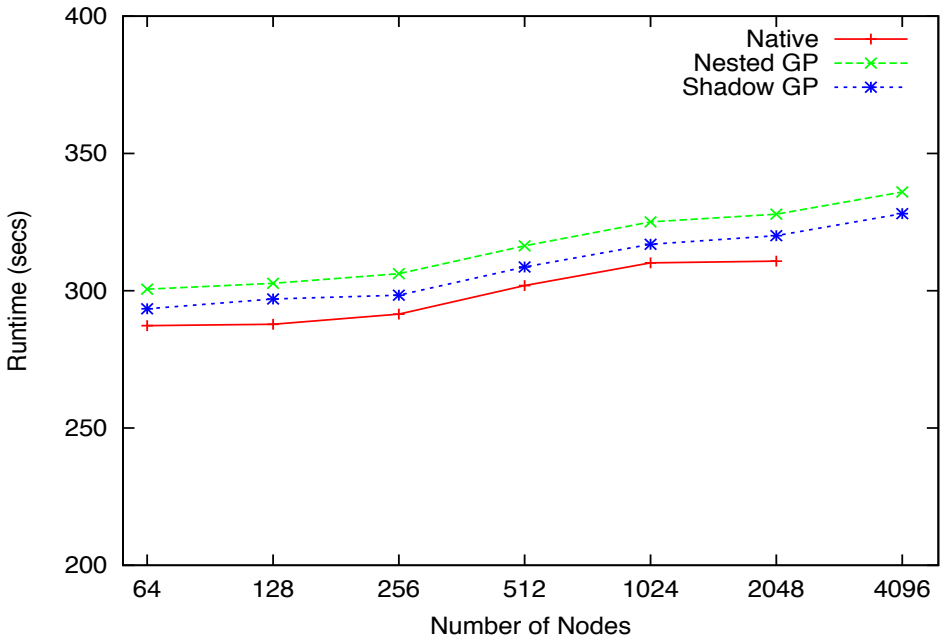
UNM | Scalable Systems Lab

Overthrowing the Dictator OS

- Modern full-feathered OSes do things for you
  - For your own good
  - Like it or not
- Runtimes want clean, predictable, easy to use interface to hardware
- Lightweight kernels can provide this but user and vendors love penguins

Overthrowing the Dictator OS

- Kitten lightweight kernel + Palacios VMM
- Run directly on the LWK w/o virtualization
- Run favorite OS in VM on top of Kitten/Palacios
- The LWK: your new benevolent dictator



CTH with virtualized Catamount guest on Palacios/Kitten running scaling up to 4096 single-core Red Storm nodes

How about just a Linux no-fly zone?

- Can we at least carve out a safe place on the system away from Linux ?
- Put Palacios into Linux – Palacios 1.3
  - Runs as a Linux kernel module
  - Segregates resources for itself from the Linux kernel
- A lightweight kernel or runtime in Palacios on Linux
  - Full control of its processors
  - Large contiguous physical memory
  - Can communicate with Linux for high-level services (networking, storage, checkpointing)

Simple VM-based Hardware Interfaces

- Present a simple hardware model to the runtime that still maps to the full power of the underlying system
- Examples
  - Portals 4.0 virtual NIC that the VMM maps to underlying high-performance zero copy fabric (in progress)
  - Ethernet virtual NIC that the VMM maps to storage/viz/etc. network (done)
  - Uniform virtual per-core power management interfaces
  - System-wide synchronized interrupt time sources
- Access could go straight through to underlying hardware in some cases

VM-based Design Space Exploration

- System Evaluation: Have to try to break things to see if they really work
- How do we test system software for big platforms that don't yet exist?
- Idea: Use VMs to make current machines act like future machines

Overall Design Space Exploration Vision

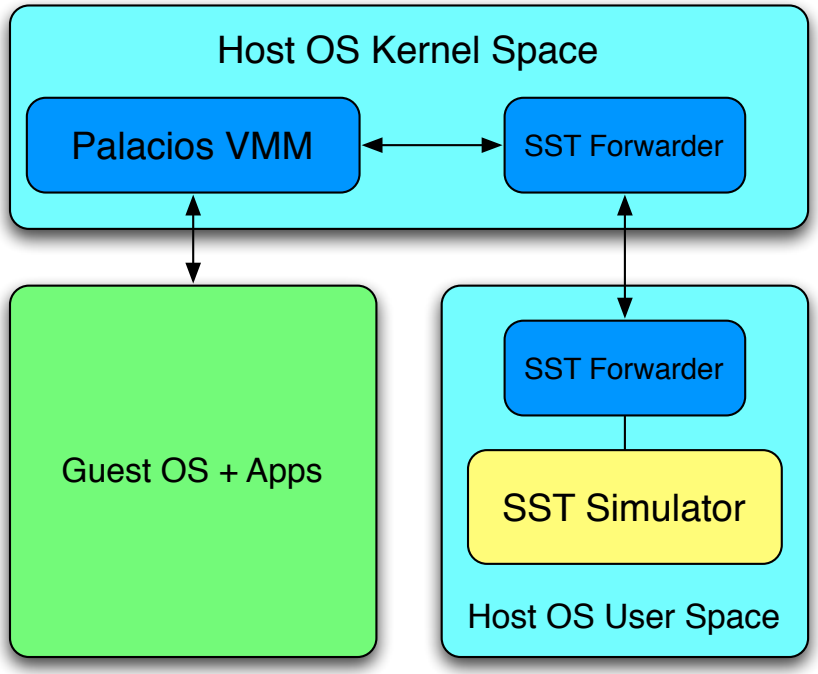
- Users build virtual versions of the machine on which they want to test their software stack
  - Run virtual system on current system using Palacios-based emulation and SST-based simulation
  - Study a range of hardware parameters over runs
  - Collect fine-grained performance information on system behavior during runs
  - Analyze collected data using models to identify bottlenecks, predict performance and large scales
- Enables flexible, large-scale co-design experiments
- Sharing results of experiments with community is critical

Palacios-based Design Space Exploration

- VMM can provide illusion of some "new" features to the OS and runtimes that run on them
  - Global memory addressing
  - New instructions (transactional memory, empty/full bits, etc.)
  - More or different speeds of processors
  - Different failure models
- Most of emulated processor runs natively, only new features needs to be handled by the VMM
- VMM controls the passage of guest time so that emulated instructions appear to take the appropriate amount of time
- Collect copious profiling information as you run

Interfacing with external simulators

- Want to leverage other design space exploration tools
- Integrate with existing simulator infrastructure
- Basic devices: trap appropriate I/O interactions, forward to SST running on host OS
- Processors: checkpoint VM state between processor simulator and GEM5



Where are we with this?

- Status:
  - Time virtualization/dilation support partially done
  - Palacios host device interfaces with SST's cycle-accurate Seastar network model
  - Can boot a VM in Palacios, move to GEM5
- Short-term goal: Provide a simple "enhanced" version of a current cluster
  - More processors with asymmetric speeds (turn 1 3GHz core into e.g. 1 1Ghz core and 16 250Mhz cores)
  - New instructions, e.g. empty/full bits
  - Different system failure models for memory, processor, etc.

Thanks

- Virtualization collaborators across the country
- Students who do all of the actual work
- This work was supported in part by the DOE Office of Science, Advanced Scientific Computing research, under award number DE-SC0005050, program manager Sonia Sachs, and a faculty sabbatical appointment from Sandia National Laboratories