Department of Computer Science

VM-based Slack Emulation of Large-Scale Systems

Patrick G. Bridges, Kevin Pedretti, and Dorian Arnold

Systems Design: We break things!

How will exascale systems break?

- If it doesn’t exist, how can we break it?
- What will break that we don’t yet know about?

Is Simulation Sufficient?

- Accuracy vs. Time-to-solution tradeoffs
- Detailed: exascale-class machine to simulate an exaflop machine
- Fast: probably only see effects we already expected to see

Using Emulation to Accelerate Simulation

- New machines evolving from current architectures
- But some key features will be very different
  - Memory, storage architecture
  - Network interfaces
- Leverage current machines to scale large simulations
  - Emulate features similar to those on existing systems
  - Completely simulate radically new features
- Understand impact of new features across entire system
- Tradeoff some accuracy for scale and time to solution

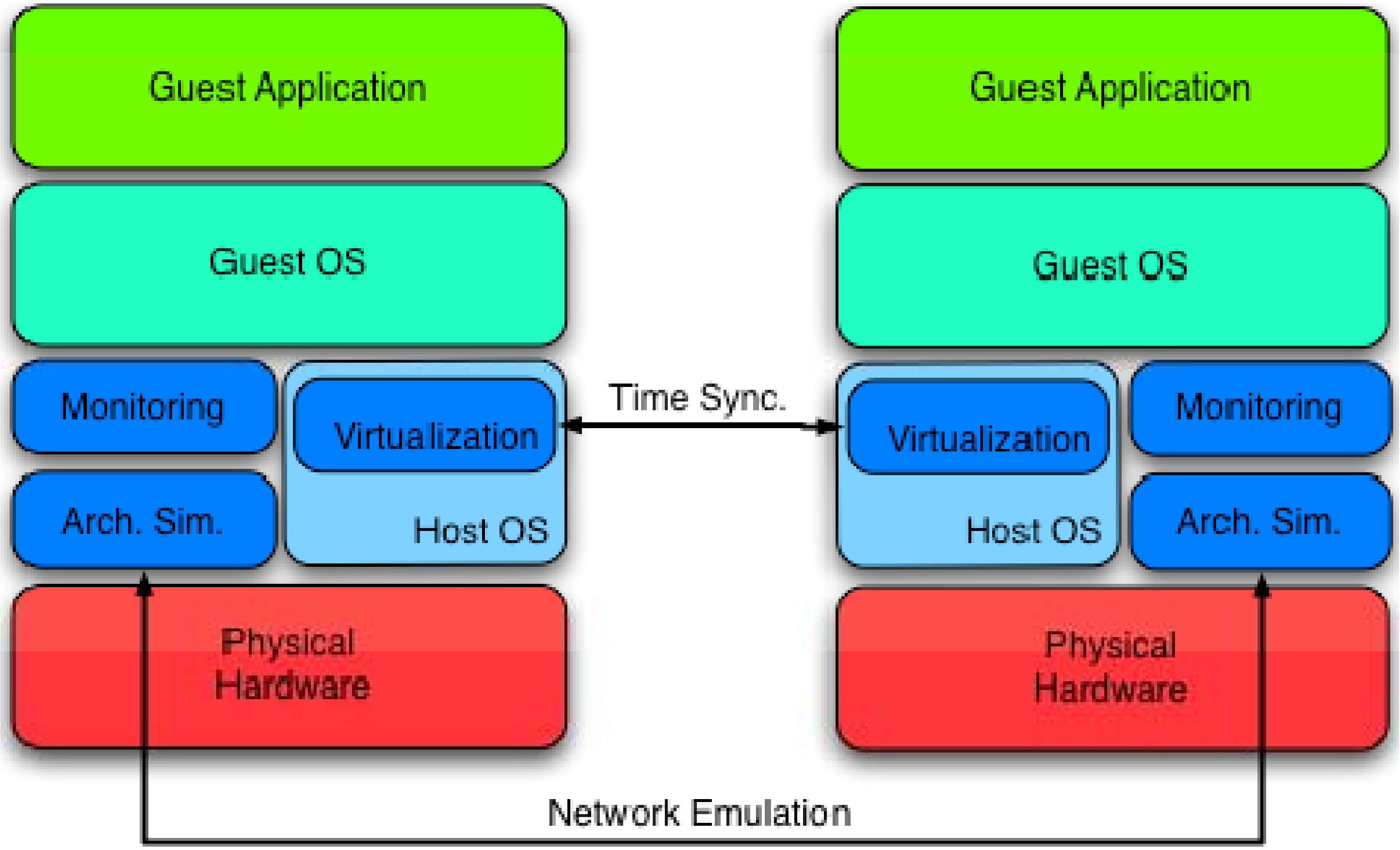
Example Uses

- Understand the impact of modified core performance
  - Many more or faster cores
  - Cores with heterogeneous performance
- Global Non-coherent Addressing
  - Impacts programming model
  - May impact OS structure
- Persistent memory systems
- Active messaging network interfaces
- Impact of different kinds and rates of failures

Basic Approach

- Goal: Large-scale, fast emulation of exascale systems
- Leverage large-scale virtualization technology
- **Dilate** time in the virtual machine to make minor changes to CPU/network performance
- **Simulate** new features using attached SST simulator
  - VMM calls into simulator to handle new devices
  - Simulator runs at user level on OS that hosts VMM
- **Loosely synchronize** per-node simulations

Architectural Diagram



Time Dilation

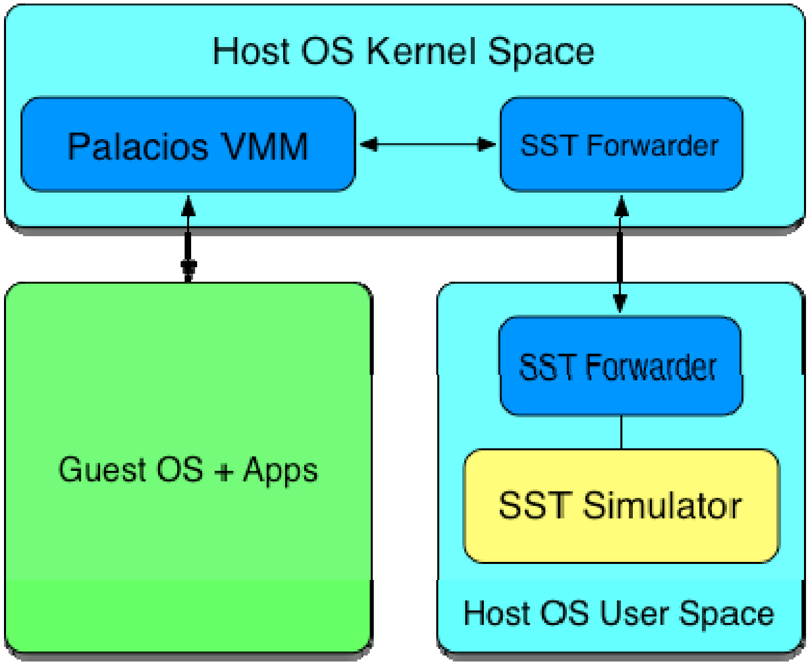
- Run the virtual machine slower than real time
  - Gives time to emulate more or faster CPUs
  - Also changes behavior/speed of underlying devices (e.g. NICs)
- Previously researched for loosely-coupled clusters
  - Emulating faster NICs (DieCast)
  - Uses **fixed** slowdown from real time
- Requires careful management of virtual time in the virtual machine monitor
- Not previously used for integration of simulator

Interfacing with Arch. Simulator

- Simulate behavior of devices that do not yet exist
  - Network interfaces
  - New memory and storage devices
  - Interesting processor features
- VMM/Simulator interface
  - VMM hooks physical interfaces to new device
  - Invoke simulator when physical device is touched
  - Pause passage of time in the local VMM when simulating
- Using Sandia Structural Simulation Toolkit

Simulator/VMM Interaction

- Simulator runs at user level parallel to machine being simulated
- VMM redirects calls between the VMM and the simulator
- Causes time to pass at uneven rates in different simulations!



Issue: Synchronizing Node Emulations

- Complete accuracy requires synchronizing actions across multiple machines
  - Preserve causality between actions on multiple machines
  - Make sure time passes consistently across entire system
  - Potentially very expensive
- Fixed time dilation avoids this by synchronizing systems to a uniform clock dilated from real time
- Not sufficient for us: uncertain simulation slowdowns!

Don’t Worry, Be Happy!

- **Slack Emulation –** keep simulations roughly in check and assume inaccuracies are minor
- Already been used in multicore CPU simulators
- Extend to large-scale system simulation
- Nodes periodically agree on slowdown factor
  - Natural interface with time dilation simulation
  - Low slowdown with possible, high slowdown when needed
- Assumes highly-accurate small-scale simulations also being used to validate the simulation

Performance Monitoring and Analysis

- Need tools to understand system behavior
- Integrate performance monitoring tools at base level of simulation/emulation system
- Understand App/OS/Hardware Interactions
- Monitor distributed interactions
- Estimate potential inaccuracy in simulations

Implementing VM-based Slack Emulation

- Leveraging Palacios HPC-oriented VMM
  - Low-overhead virtualization on HPC systems
  - < 5% overhead on Cray XT systems @ 4000 nodes
  - Open source
- Enhanced Palacios time virtualization features
  - Can fully virtualize time
  - Pause, resume, slow down guest time passage
  - Adding complete time dilation support
- Implemented interface for host-level devices to tie to simulators

Next steps

- Dynamic time dilation rates
- Simulation of simple devices
  - Basic persistent memory devices
  - Existing NIC simulation (Cray SeaStar functional simulation)
  - Global addressing
- Basic performance monitoring device integration

Acknowledgements

- DOE Office of Science, Advanced Scientific Computing research, award number DE-SC0005050, program manager Sonia Sachs
- Faculty sabbatical appointment from Sandia
- Ron Brightwell for giving this presentation
- Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04- 94AL85000