SOS Panel on Compilers and Tools Software Scaling Challenges for the Exascale

March 10, 2009

Kevin Pedretti
Scalable System Software Department
Sandia National Laboratories
ktpedre@sandia.gov

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Panel Questions

- What is a/the major challenge to reaching productive exascale computing?
  - Being more resilient to faults
  - Dealing effectively with massive parallelism and data/thread placement issues within a node
  - Simply understanding what is going on in the machine
  - Innovating with and alongside commodity system software
- How can tools and compilers help in addressing these issues?
  - Tools and libraries for easing or hiding (partially) redundant computation
    - Sandia Redundant MPI library
  - Smarter intra-node auto-parallelizing compilers
    - Hopeful general-purpose computing market will push this
    - But not sold yet on the “negotiate with the compiler” model
  - Real-time performance data gathering and display/replay
    - OS and runtime driven, rather than individual app process-driven
    - Identify hot spots and mitigate by migrating threads/data
  - Research tools for exploring system software options
  - Auto-tuners to explore (ever-expanding) option spaces

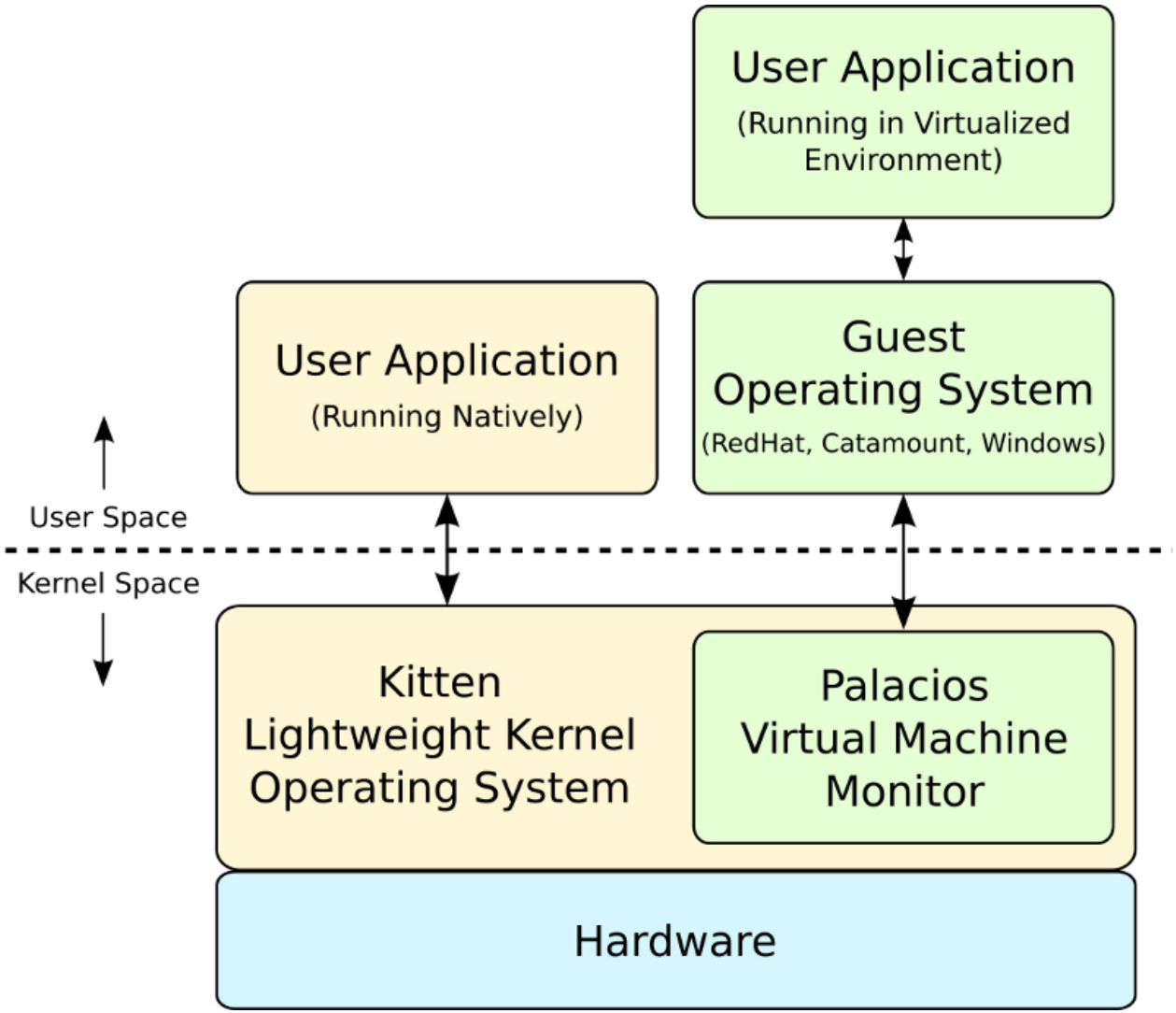
Google adds ~300K LOC to Linux, ~75% core changes (Source: lwn.net)

Sandia National Laboratories

Panel Questions (continued)

- What won't compilers and tools be able to help with?
  - Making exascale programming easy
- Tools and compilers for petascale were incremental changes from the gigascale. Why is this not the case for petscale -> exascale, or will it be?
  - It will be incremental until it can't be
  - Will take time and experience to figure out what non-incremental tools would be helpful
- What is the one piece of current software and tools that you would totally scrap and either do without or replace? Why?
  - Honestly, couldn't come up with anything serious
  - Not big fan of the tracing+profiling tools I've used, haven't tried Tau or Vampir but plan to

Using Virtualization as a Tool

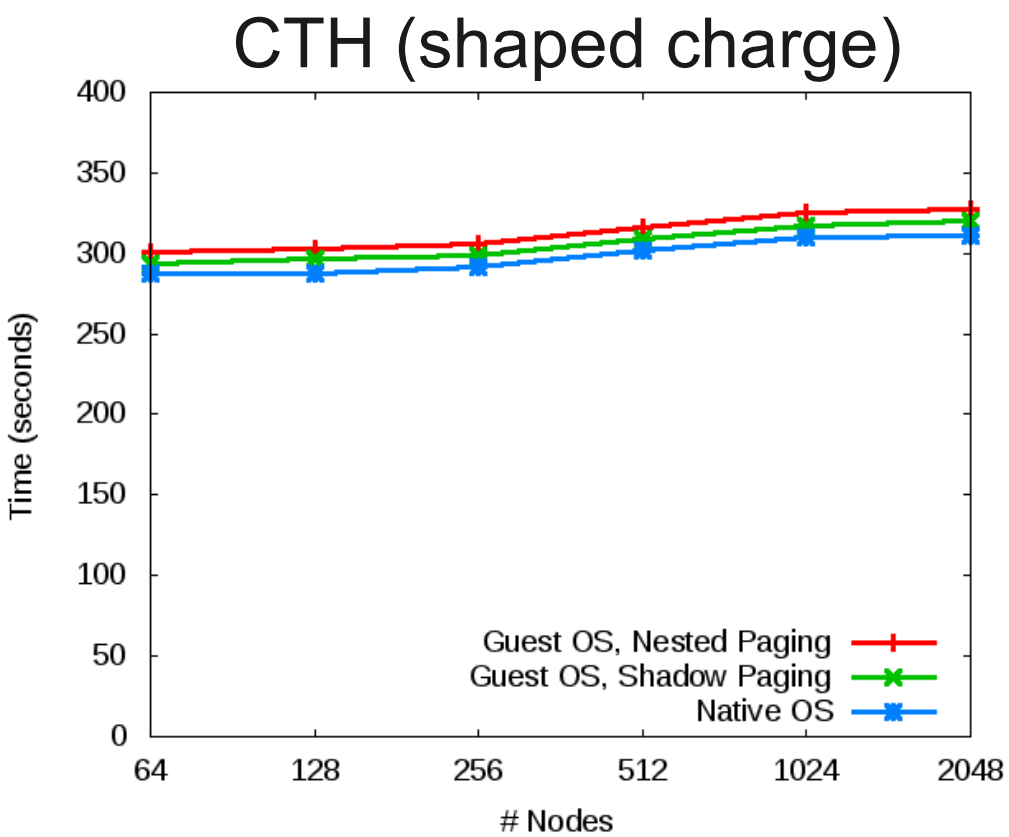


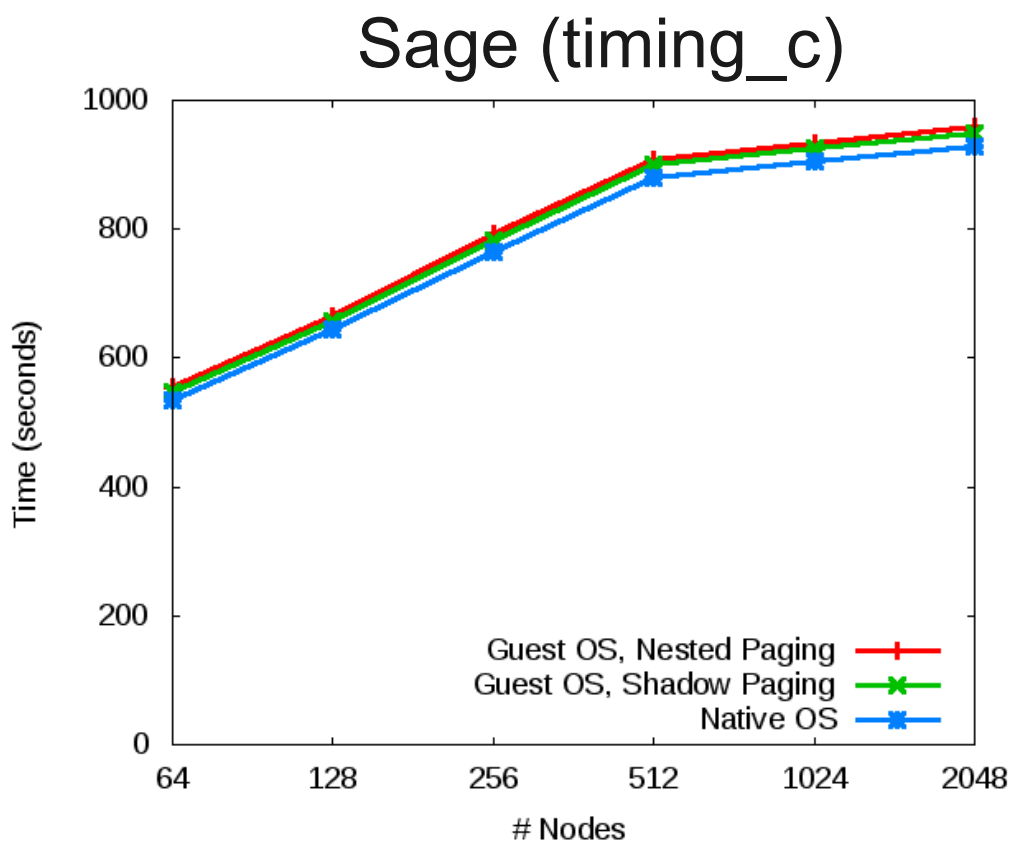
- For end-user flexibility
  - Provide full functionality OS option in LWK environment
  - Run commodity OSes
  - Convenient packaging
- For research
  - X-Stack development and large-scale test
  - Add capabilities to guest OS without modifying it
  - VM migration/resilience
  - Instrumentation and debugging

Kitten homepage: https://software.sandia.gov/trac/kitten
Palacios homepage: http://www.v3vee.org/palacios/

Sandia National Laboratories

Large-scale Virtualization Experiments on Red Storm

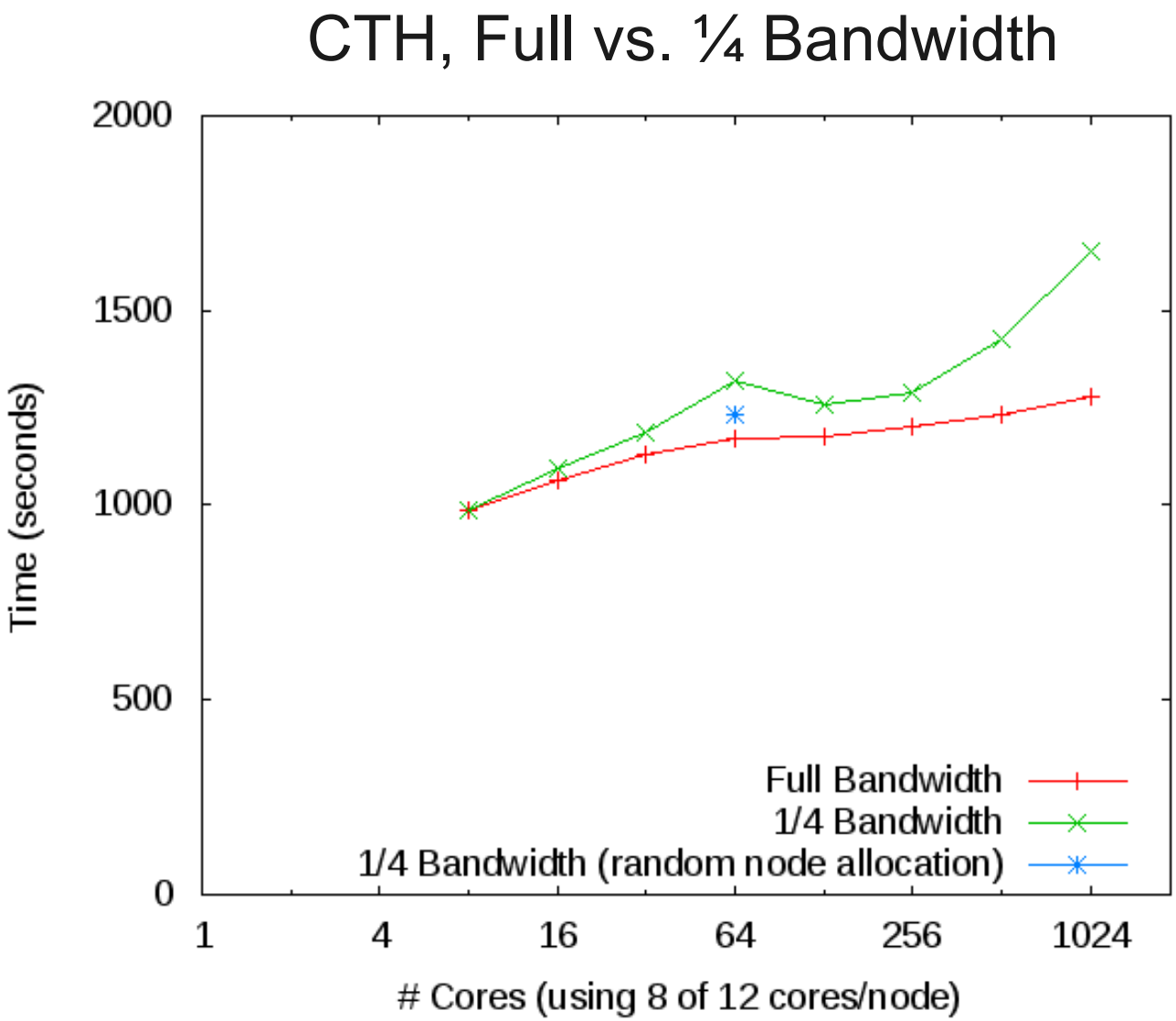




< 5% virtualization overhead for all cases tested

Sandia National Laboratories

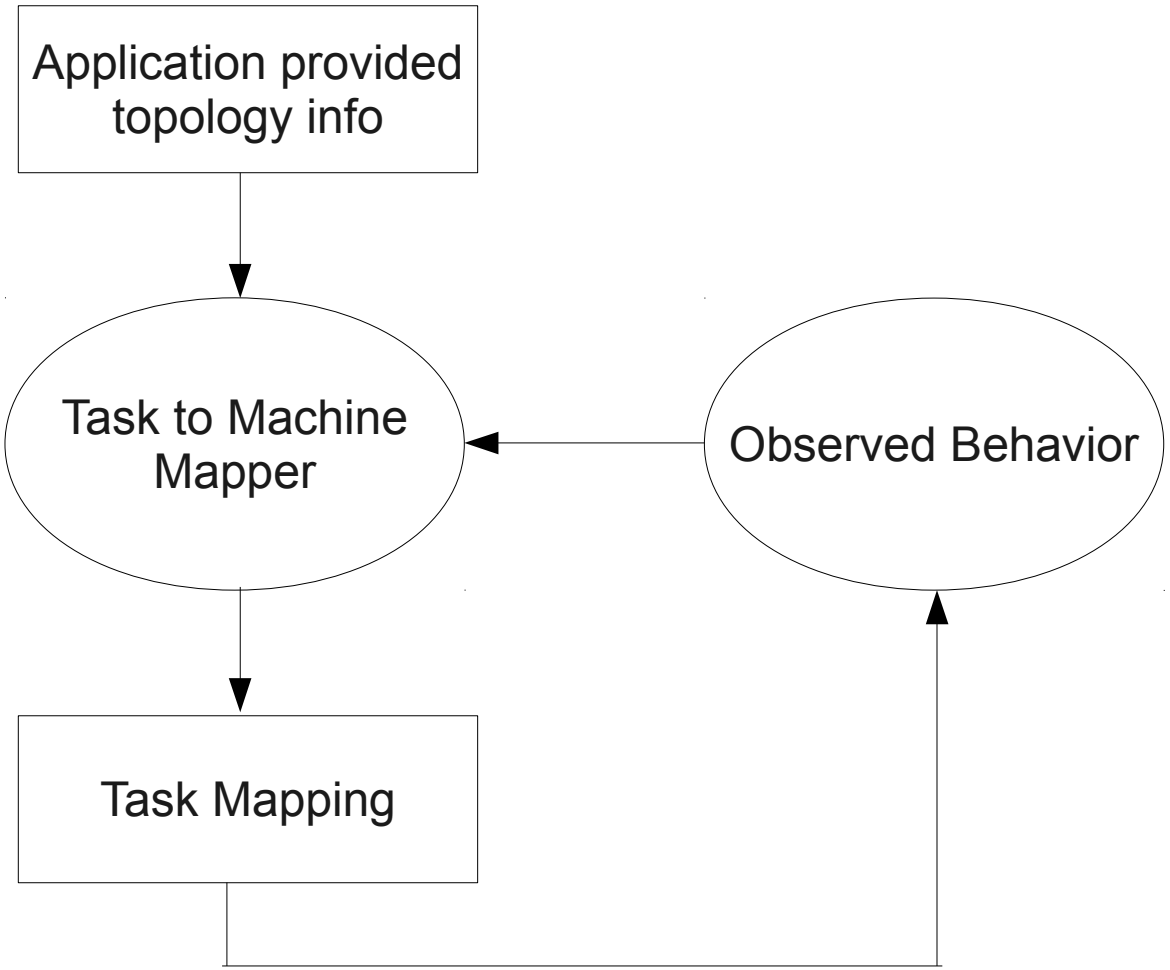
With Reduced Network/Compute Balance, Node Allocation (Might?) Become More Important



- 29% worse at 1024 cores (128 nodes)
- Default vs. random node allocation 7.4% worse at 64 cores (8 nodes)
- Reasons to care
  - Predictability
  - Power

Sandia National Laboratories

Why Don't We Do This?



- Over provisioning of network largely hides impact of bad placement today, tomorrow?
- Requires inter-node task migration
- Requires cooperation of multiple software components
- Model applies to both inter and intra node placement

Sandia National Laboratories