Better than Native: Using Virtualization to Improve Compute Node Performance

Brian Kocoloski

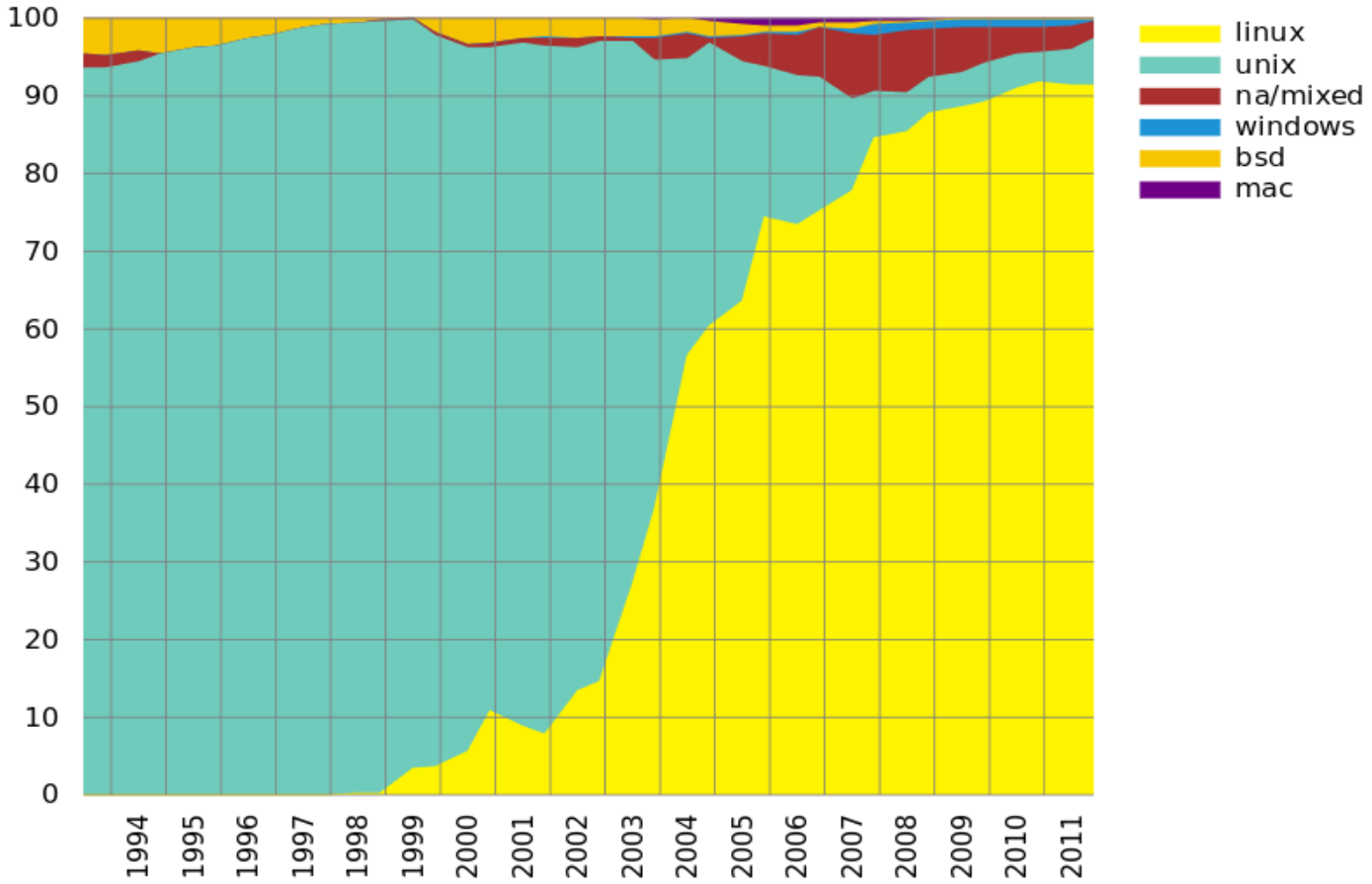
Jack Lange

Department of Computer Science

University of Pittsburgh

6/29/2012

Linux is becoming the dominant supercomputing OS …



Source: http://en.wikipedia.org/wiki/File:Operating_systems_used_on_top_500_supercomputers.svg

… but some applications need less overhead

- Lightweight Kernels (LWKs) provide low overhead access to hardware
- Q: How do we provide LWKs to applications that need them, but not to those that don't?
- A: **Virtualization**
- Applications running in a **virtual environment** can outperform the same applications running **natively**

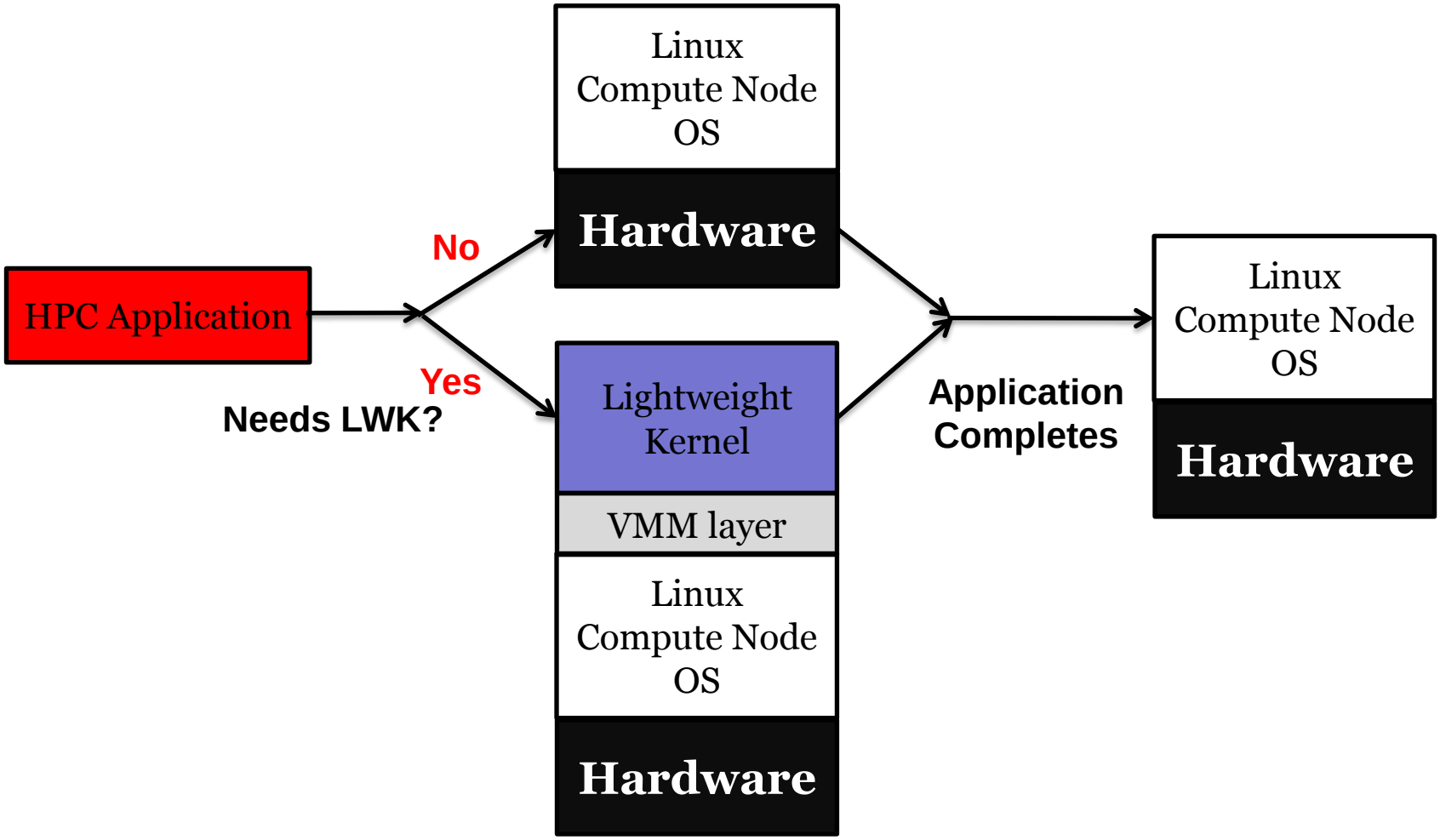
Drawbacks of Linux

- **Memory Management**
  - Biggest problem
  - Widely recognized as a source of overhead
- **OS Noise**
  - HPC apps are tightly synchronized
  - Timing is a big deal
- **Non-technical**

Disadvantages of Current Schemes

- **ZeptoOS**
  - "Big Memory"
  - Memory is **statically** sized, allocated at **boot** time
  - Compatibility
- **Cray's CNL**
  - HugeTLBfs
  - Maximum of **2MB-sized memory regions** available

Our Approach



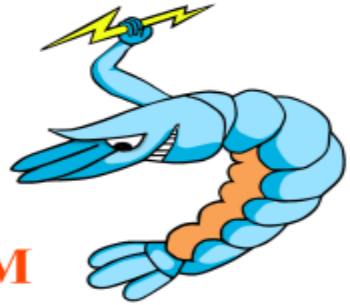
Palacios

- OS-independent embeddable virtual machine monitor
- Strip resources away from host OS
- Low noise, low overhead memory management
- Developed at Northwestern University, University of New Mexico, and University of Pittsburgh
- Open source and freely available

Palacios

An OS Independent Embeddable VMM

http://www.v3vee.org/palacios

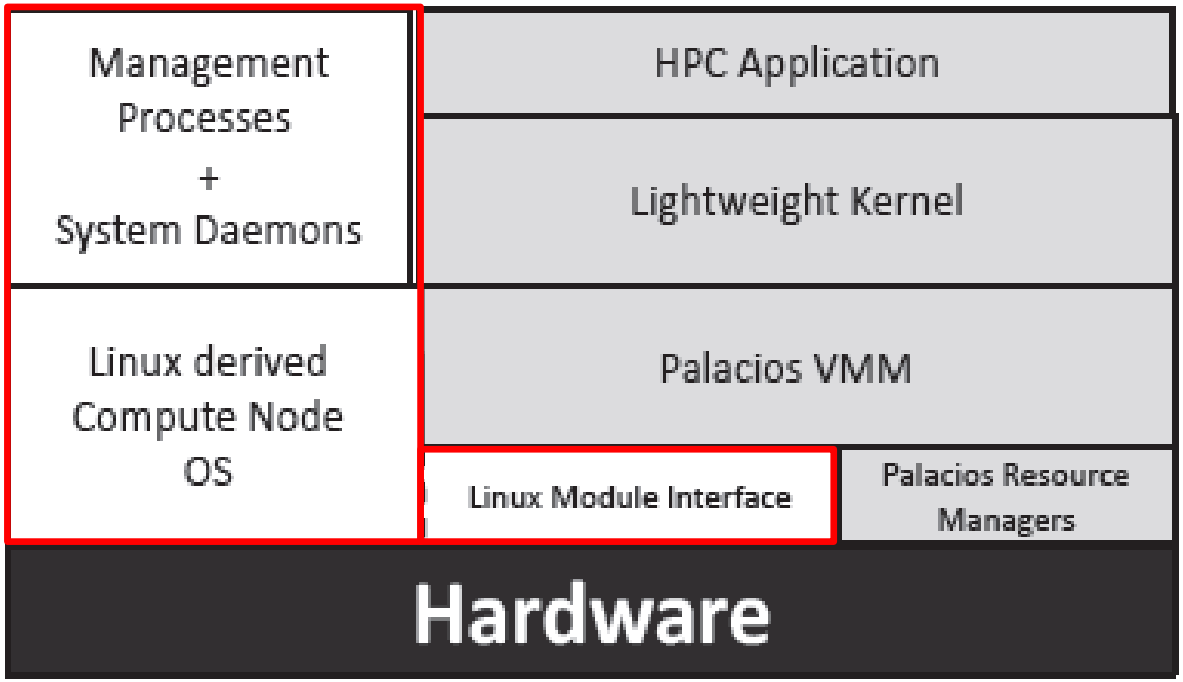


Kitten

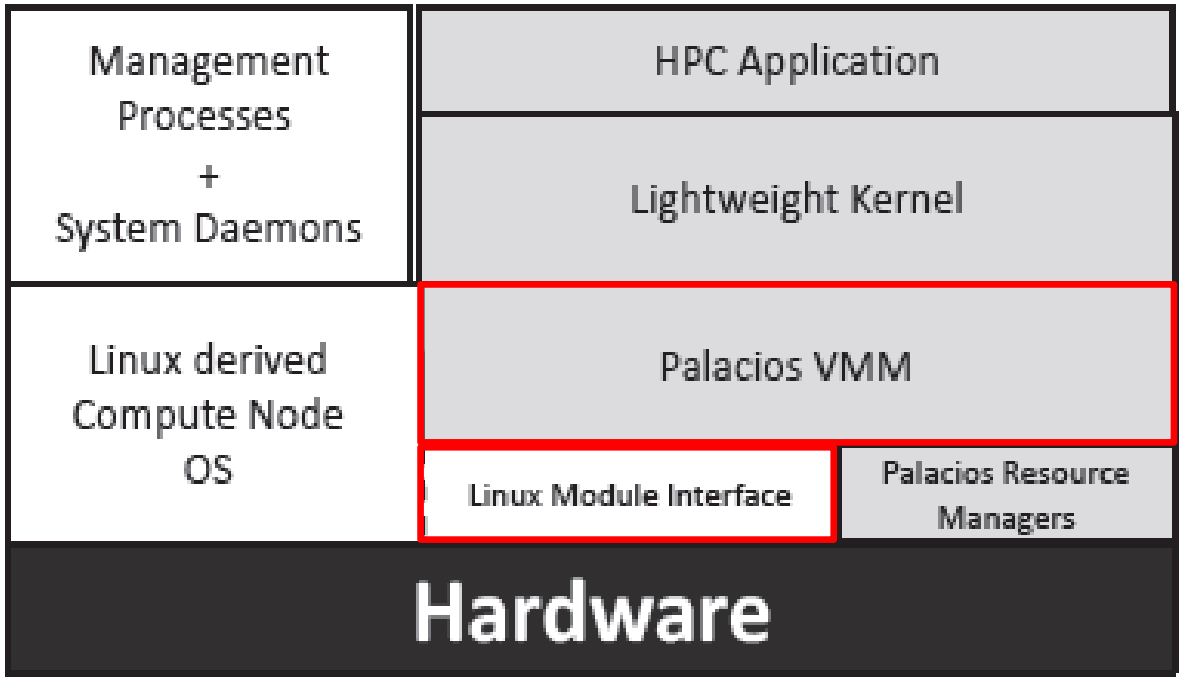
- Lightweight Kernel from Sandia National Labs
- Moves resource management as close to application as possible
- Mostly Linux-compatible user environment
- Modern code-base with Linux-like organization
- Open source and freely available

http://software.sandia.gov/trac/kitten

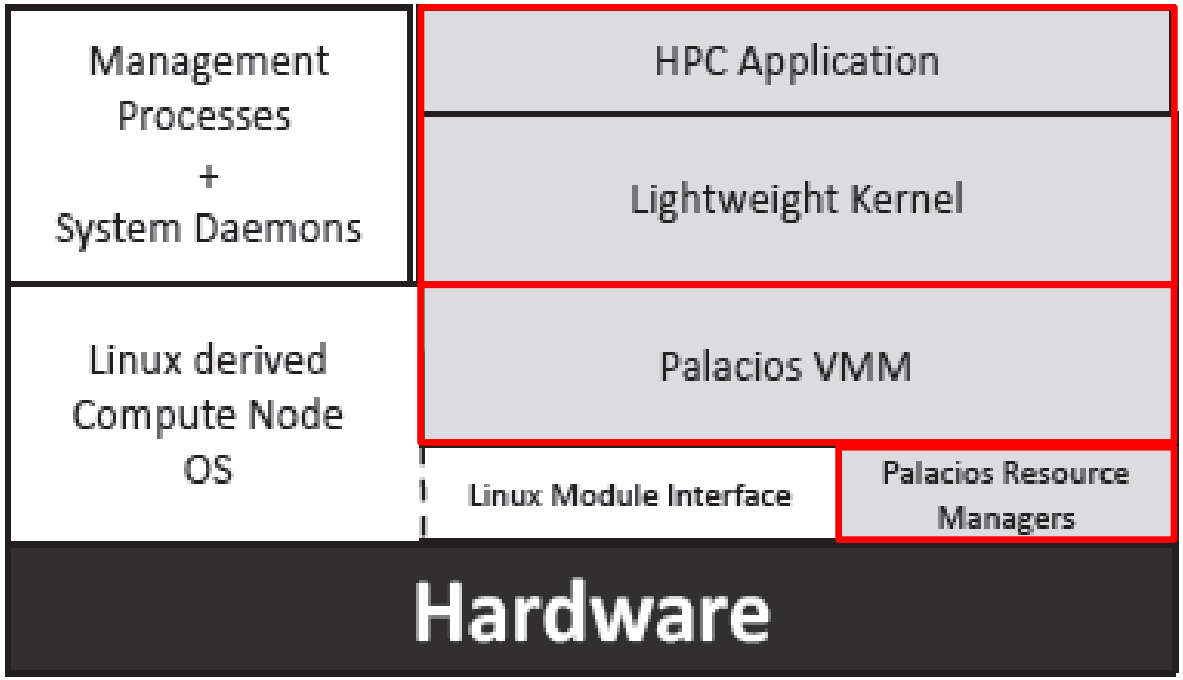
System Architecture



System Architecture

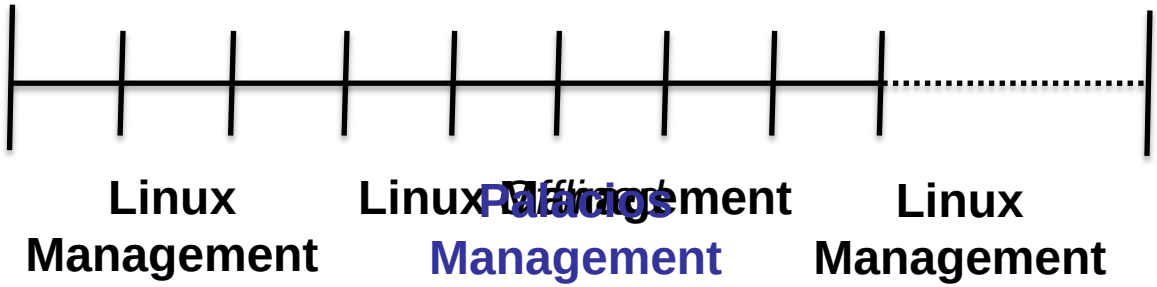


System Architecture



Palacios' Approach

- **Memory Management**



  - Bypass the Linux memory management strategies completely, at run time

- **OS Noise**
  - Control when the Linux scheduler is able to run
  - Take advantage of tickless host kernel

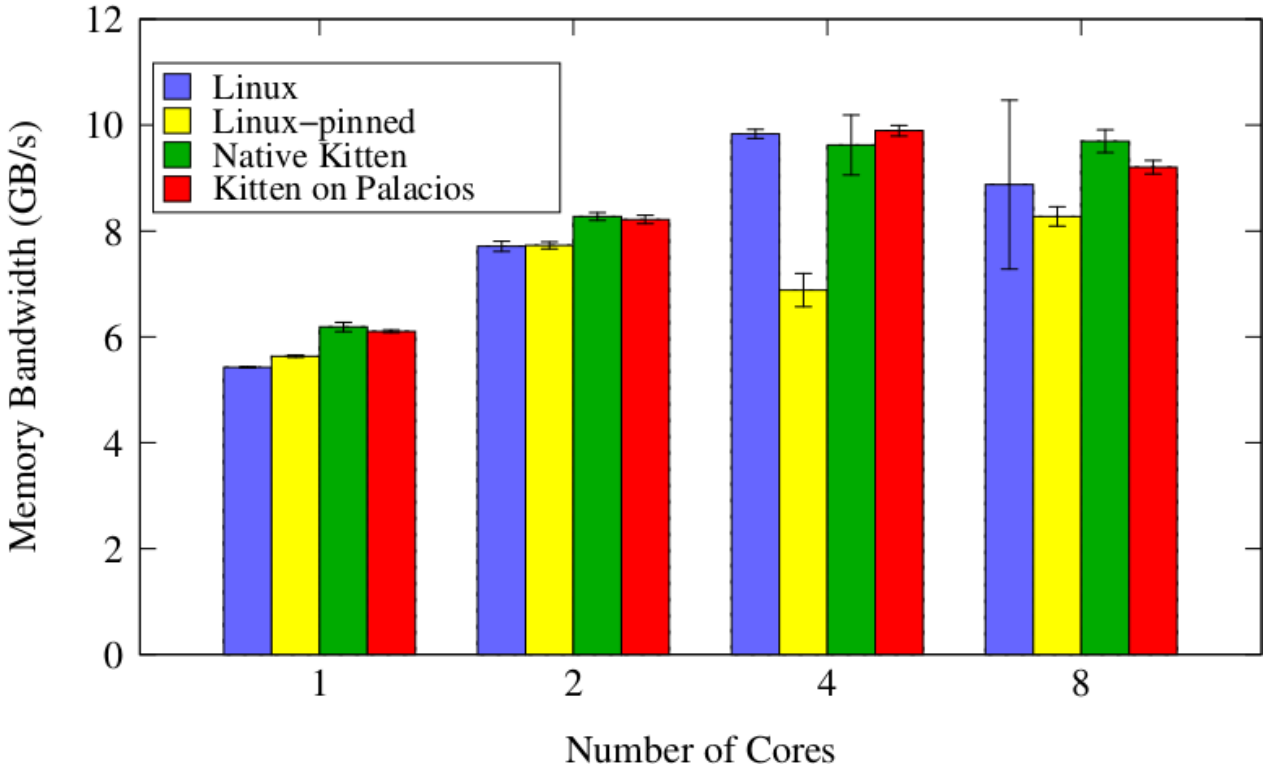
Evaluation

- Two part evaluation:
  1. Microbenchmarks – Stream, Selfish
  2. Miniapplications – HPCCG, pHPCCG
- Evaluation is preliminary
  1. Currently limited to a single node running a commodity Fedora 15 kernel
  2. Environments are not fully optimized

Environment

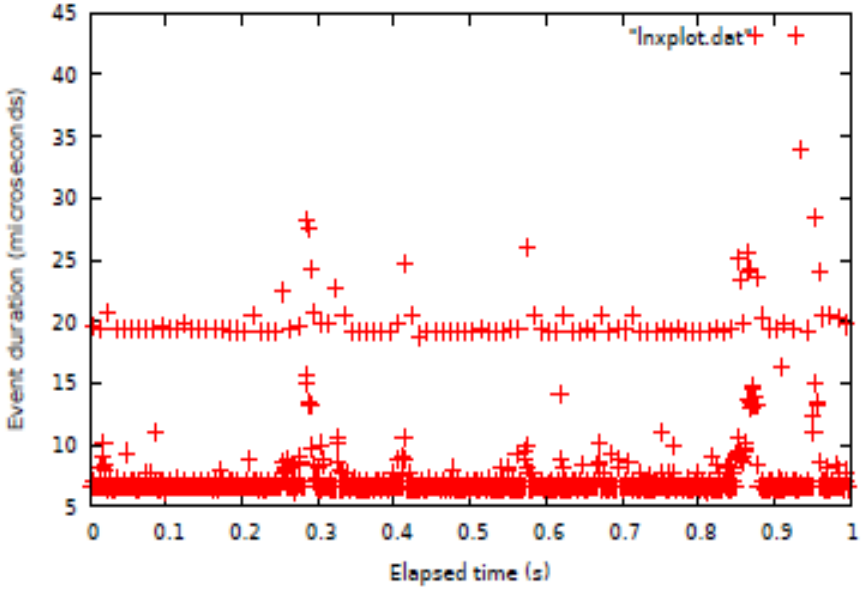
- Two 6-core processors and 16 GB memory
  - **NUMA** design
- Kitten VM was configured with 1 GB of memory
- Stream, HPCCG used OpenMP for shared memory and ran 10 times

Stream

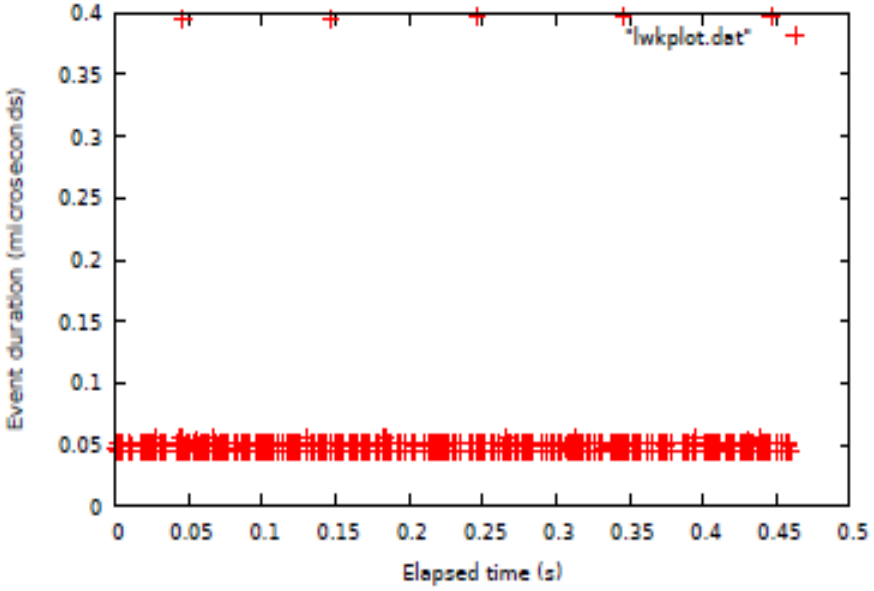


- Palacios provides ~400 MB/s better memory performance on average than Linux (4.74%)
- 0.34 GB/s lower standard deviation on average

Selfish Detour

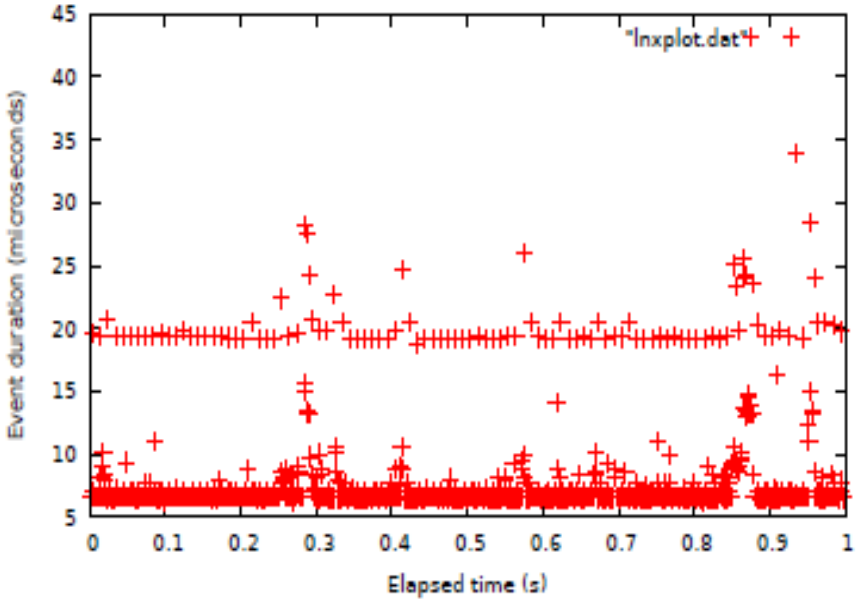


Linux

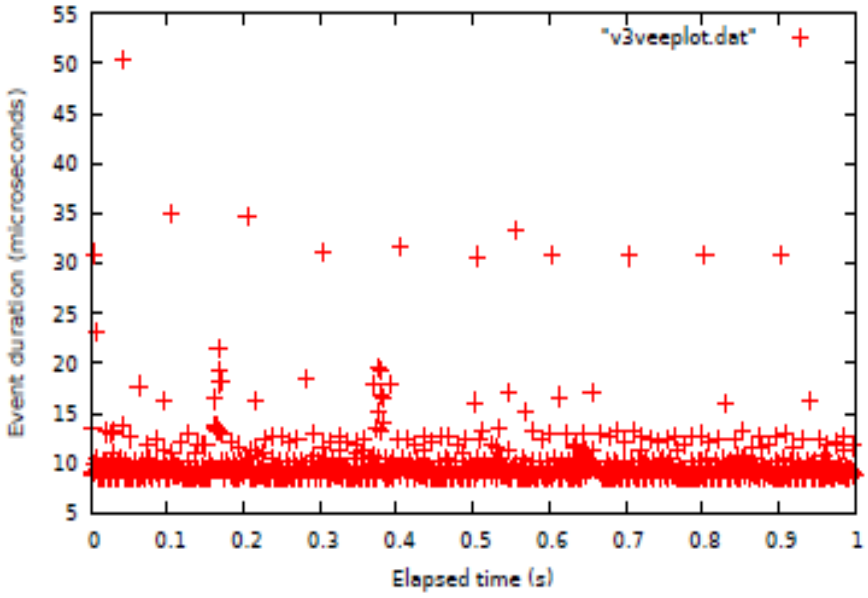


Kitten

Selfish Detour



Linux

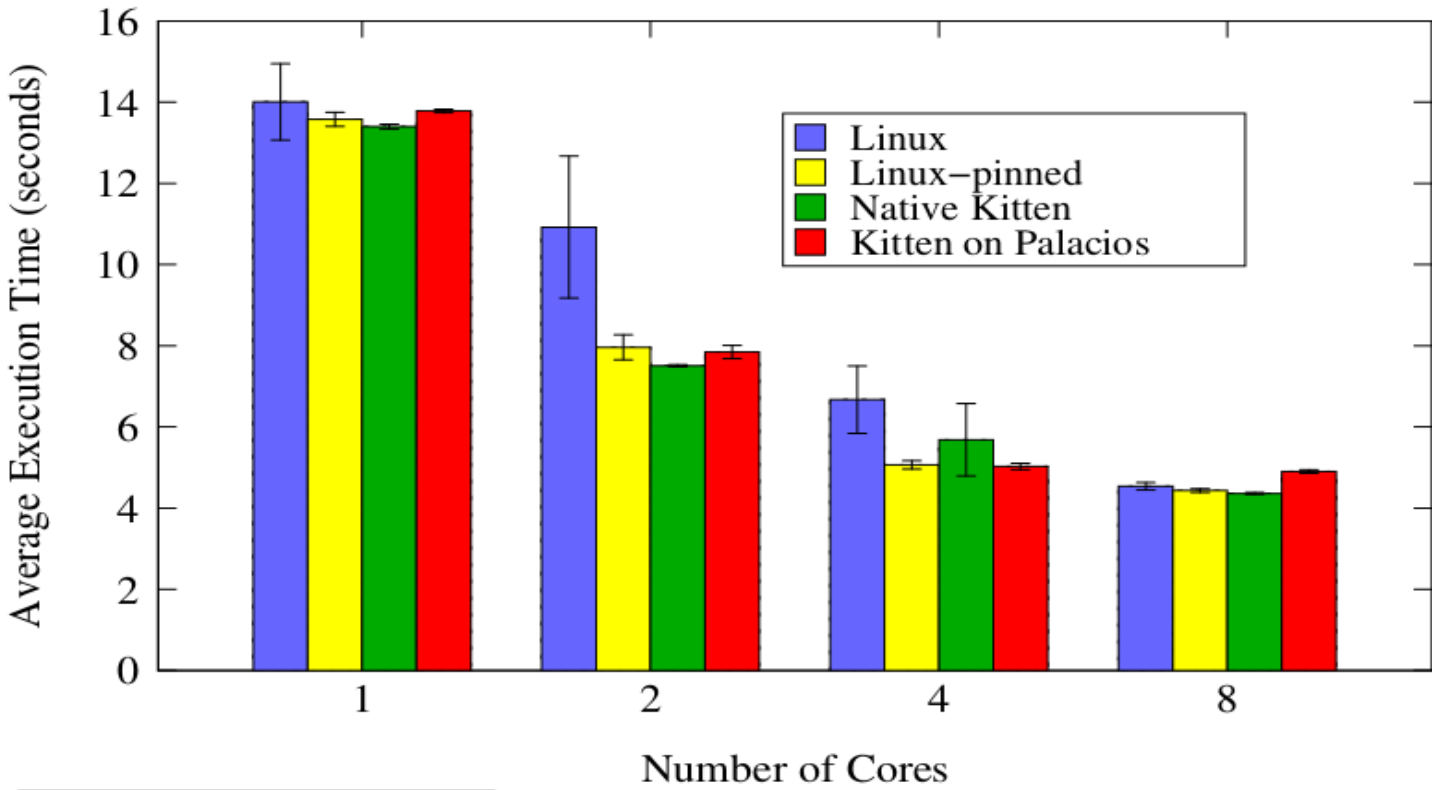


Virtualized Kitten

HPCCG

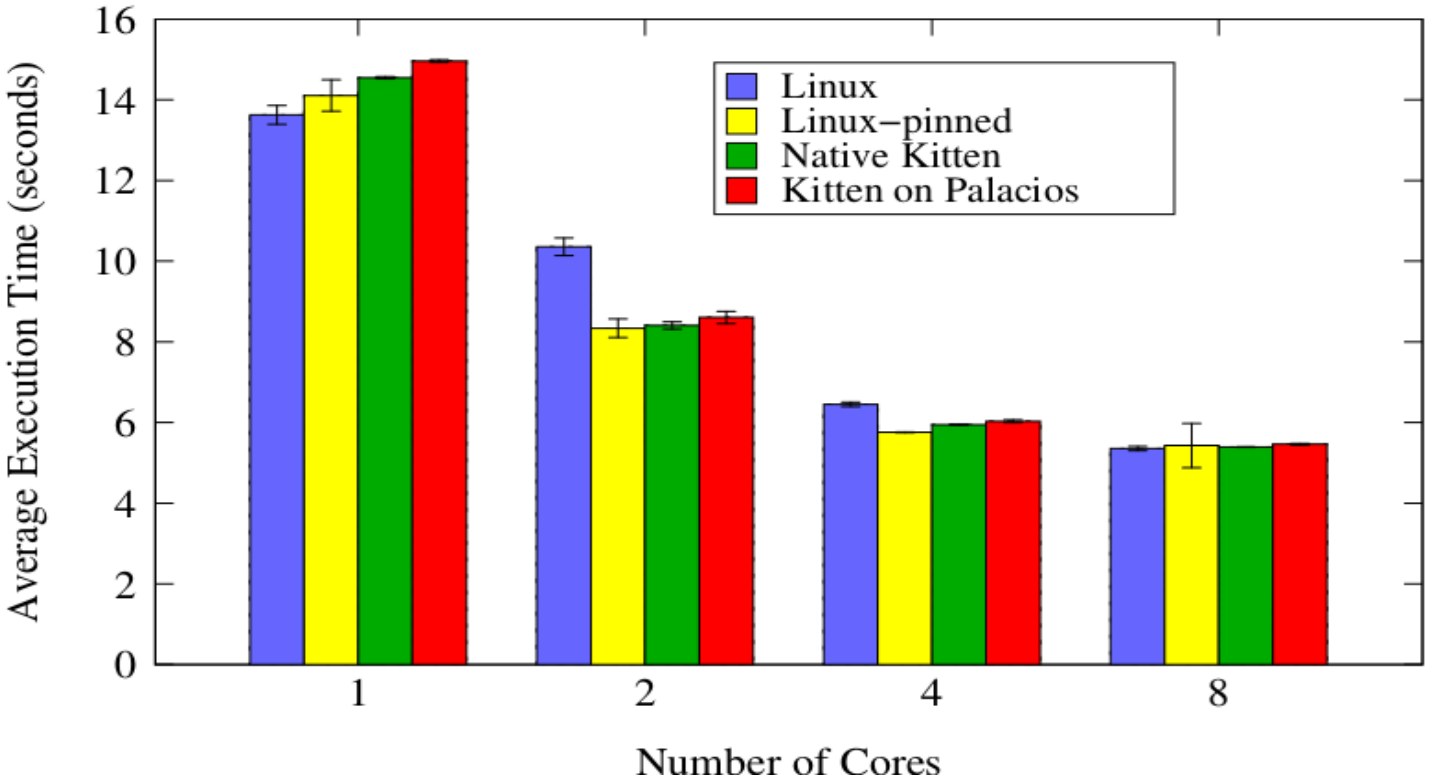
- Performs the conjugate gradient method to solve a system of linear equations represented by a sparse matrix
- Workload similar to that of many HPC applications
- Separate experiments to represent both CPU and memory intensive workloads

HPCCG – CPU intensive



Average standard deviations	
lnx	0.90
lnx-opt	0.16
lwk	0.25
v3vee	0.08

HPCCG – memory intensive



Average standard deviations	
lnx	0.14
lnx-opt	0.30
lwk	0.03
v3vee	0.06

Future Work

- Extend to actual Cray hardware with a CNL host
  - Show definitively if this approach can work
- Explore the possibility that this approach can be deployed in a cloud setting to provide virtual HPC environments on commodity clouds
  - Previously infeasible, due to the contention, noise, etc.
  - Problems we think can be solved by the same techniques used in this work

Conclusions

- Palacios is ***capable*** of providing superior performance to native Linux
- Palacios can provide a low noise environment, even when running on a noisy Linux host
- While results are preliminary, they show that this approach is feasible at small scales

Acknowledgments

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

- **Email**: briankoco@cs.pitt.edu
  jacklange@cs.pitt.edu