http://ssrg.ece.vt.edu

Systems Software Research Group

ece Bradley Department of Electrical & Computer Engineering

VirginiaTech Invent the Future®

Towards Operating System Support for Heterogeneous-ISA Platforms

Ben Shelton*, Antonio Barbalace, David Katz**, Alastair Murray, Binoy Ravindran

Department of Electrical and Computer Engineering, Virginia Tech, Virginia, USA

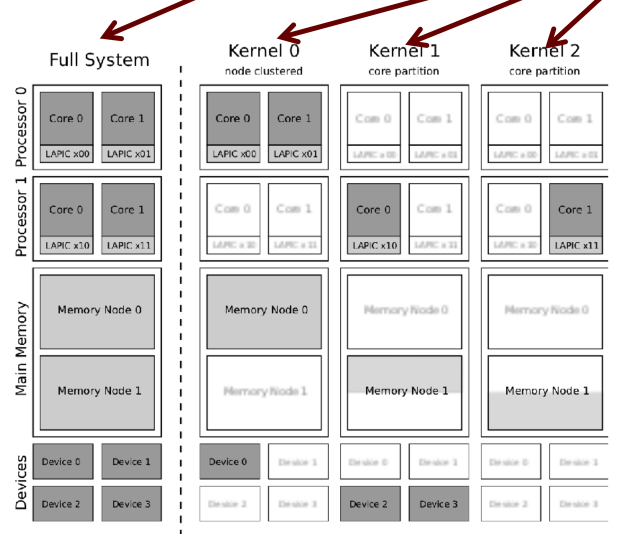
{beshelto, antoniob, davidk, alastair, binoy@vt.edu}

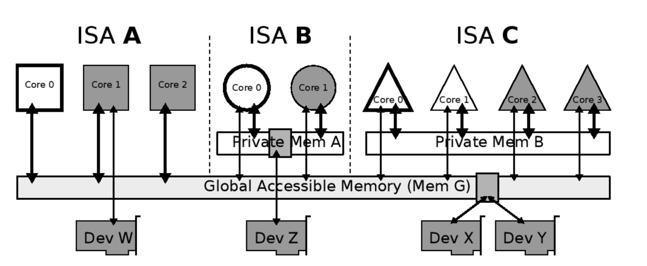
*Ben Shelton is now with National Instruments, **David Katz is with John Hopkins University Applied Physics Laboratory

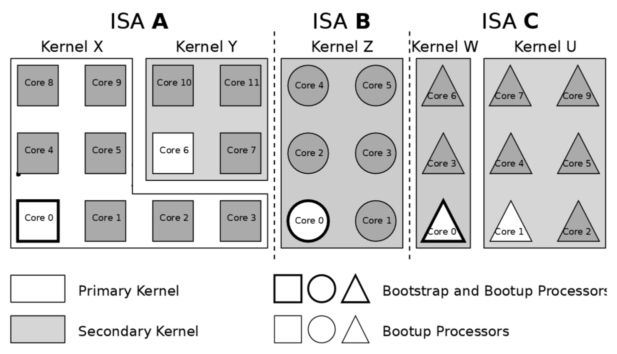
Future computing platforms are increasingly ISA-diverse and parallel

Are current OS designs ready for emerging hardware?

From Single-image to Replicated-Kernel Operating System





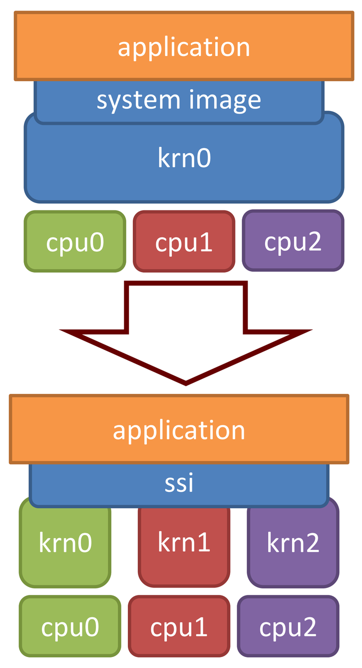


- Traditional SMP single-image OSs (Linux, UNIX, Windows, etc.) cannot run on heterogeneous-ISA platforms as-is
- ISA diversity stops the single-image per machine paradigm, one kernel (or more) per ISA group is needed

- Kernels communicate in order to maintain a common OS state that is (partially) replicated over every individual kernel instance
- Hardware resources (i.e. devices) are fully shared 'virtually' amongst the kernels (and applications)

Application Transparency

- In a replicated-kernel, different kernel instances are running in parallel on the same hardware
- Applications run transparently, like in a single-image operating system
- All kernels collaborate in order to show a single system image and filesystem namespace



OS Design Principles: Peer Kernels, Resource Sharing, Load Sharing

We implemented this design in Linux, and tested on x86

We developed a set of features that must be implemented in a traditional SMP OS in order to support ISA heterogeneity

Boot Process Mod

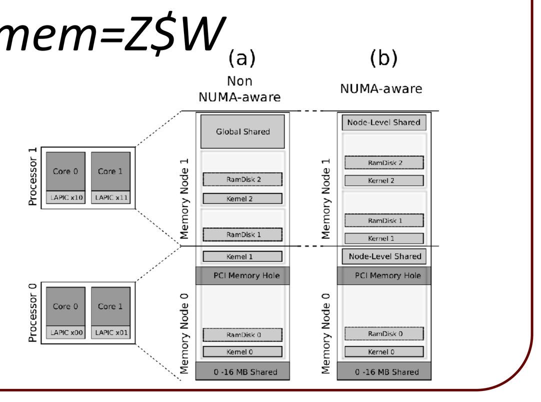
- Every kernel boots from a different binary image
- Binaries are compiled for a specific ISA
- We introduced primary and secondary kernels
- The primary kernel boots after the BIOS
- We extended *kexec* to boot secondary kernels
- A kernel should boot from anywhere in the physical address space

CPU Grouping

- In order to provide a single system image we developed a single CPU ID namespace
- On x86 this enum is compatible with Linux
- To boot a kernel on a specific subset of CPUs we added a kernel command line argument *present_mask= …*
- If not present Popcorn behaves like Linux

Memory Mngmnt

- We studied different memory partitions
- Static and dynamic memory re-partitions
- Linux already supports memory specific cmd line args *mem=X@Y* and *mem=Z$W*

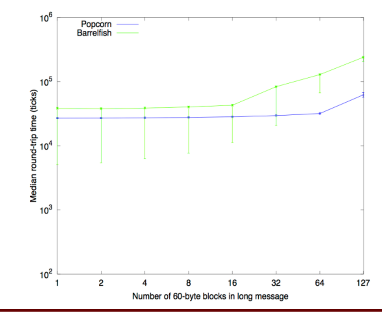


Device Drivers

- Only a subset of device drivers must be loaded at startup
- Every device in the system must be known by every kernel
- Only one kernel can use a specific device at a certain time
- The current user of a device must provide proxy access for all other kernels

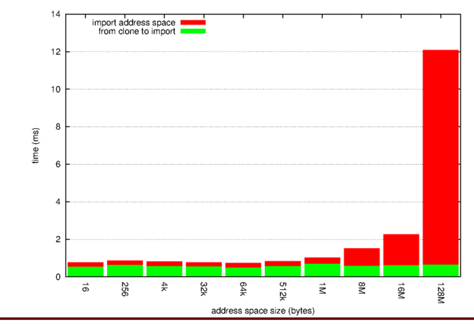
Inter-Kernel Comm

- Basic kernel to kernel communication is provided by a messaging layer (shared memory and IPIs on x86)
- OS state is replicated and kept coherent using the messaging layer



Process Migration

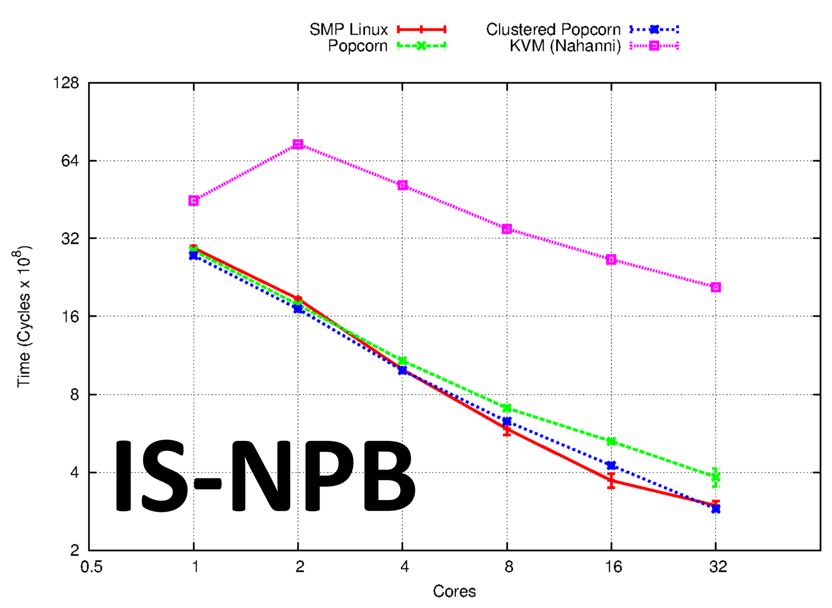
- A process can migrate to/from any kernel
- Based on the topology and heterogeneity of the system, different levels of replication of the address space are currently being studied

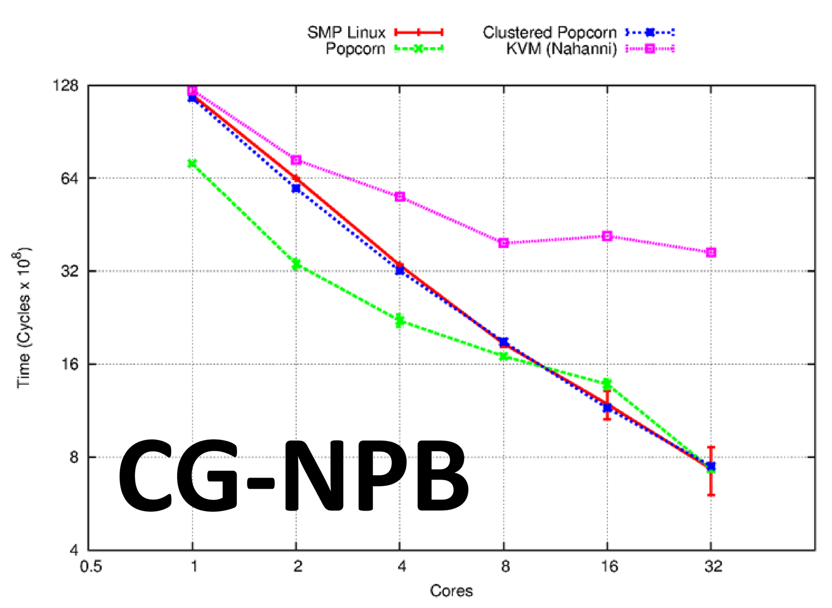


Initial Evaluation of Popcorn Linux

Compute-bound Tests

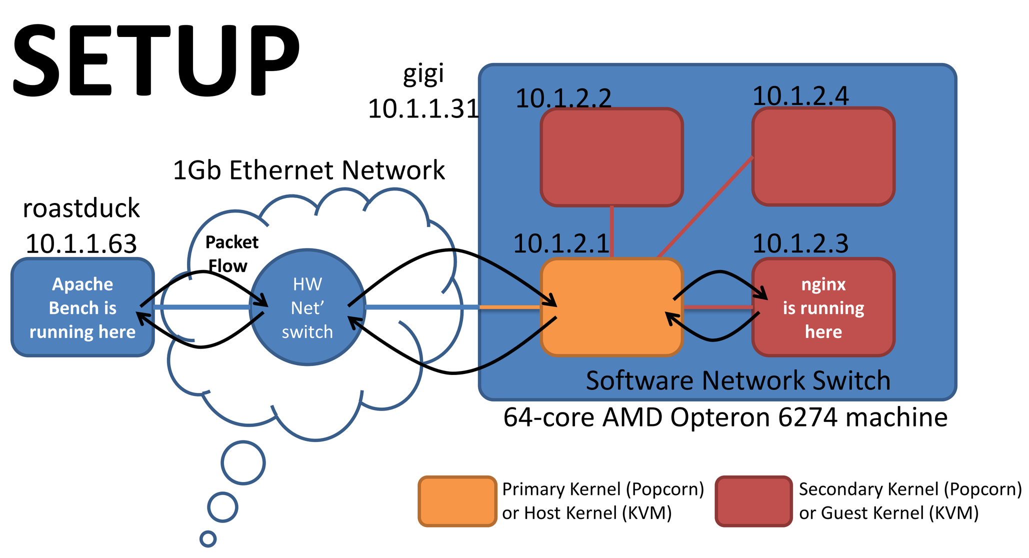
- We ran the NASA Parallel Benchmarks (NPB) MPI version to verify that Popcorn maintains the same compute-bound performance as Linux
  - We compared it with KVM and SMP Linux
  - We tested one kernel per core configuration (partitioned Popcorn)
  - We tested one kernel per NUMA node configuration (clustered)
- We used MPICH2 in all configurations (KVM uses inter-VM shared memory)

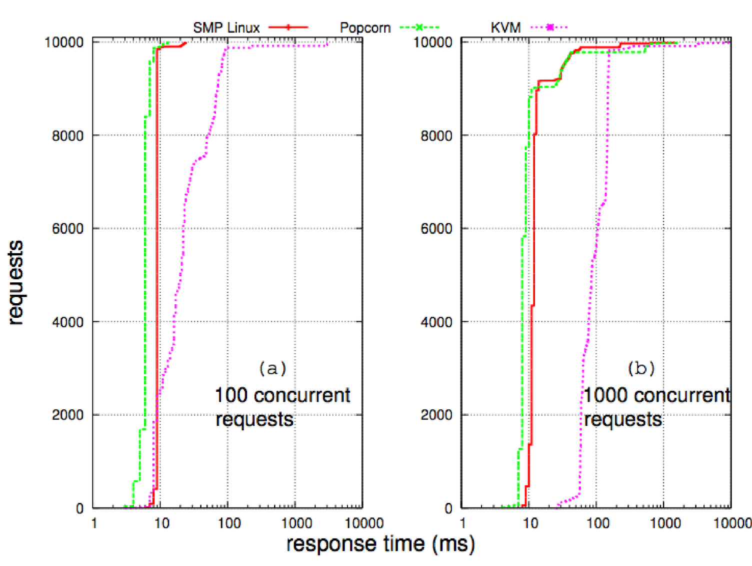




Network Tests

- We ran the Apache Bench test tool to evaluate *nginx* (web serv') response time
  - We compared it with KVM and SMP Linux
- Tested version is hybrid interrupt and polling mechanism, based on Linux NAPI
  - Fast ring buffers in shared memory between client and server
  - Under high traffic, using polling reduces the overhead of servicing the thousands of interrupts per second that would be generated





Conclusions

We began to address the problem of how to re-design a traditional SMP OS to support Het-ISA.

A traditional SMP OS, like Linux, can be used as the base to implement an OS for a Heterogeneous-ISA Platform.

Compute-bound and network tests show that the replicated-kernel design doesn't degrade the performance of Linux.

www.popcornlinux.org

ACKNOWLEDGMENTS

This work is supported by US Office of Naval Research under Contract N00014-12-1-0880.

Arijit Chattopadhyay implemented the virtual TTY driver. Shawn Furrow ran the Nahanni experiments.

Department of the Navy ONR Science & Technology