Porting the Kitten Lightweight Kernel Operating System to RISC-V

Nick Gordon [1,2], Kevin Pedretti[2], John R. Lange[3]
nick.gordon@cs.pitt.edu, ktpedre@sandia.gov, langejr@ornl.gov
[1]University of Pittsburgh, [2]Sandia National Laboratories, [3]Oak Ridge National Laboratories

Introduction

Hardware design in HPC is highly experimental and exploring new designs is difficult and time-consuming, requiring close vendor cooperation. RISC-V is an open standard processor ISA that we can use for hardware/software codesign:

- RISC-V democratizes chip/hardware design by being open
- Open ISAs ease hardware/software co-design by *shortening* the design cycle
- Shortened design cycles *reduce cost and time* by trying ideas sooner

This work aims to leverage RISC-V for hardware/software co-design by developing a suitable OS with which to conduct co-design.

Co-Design and the "Design Barrier"

Co-design is hindered by the uncoupled development cycles of hardware and software. Conventionally, design is done *hardware first* and software accommodates. Software design creates feedback, too, e.g. hardware virtualization.

We call this separation the "design barrier," allowing each to evolve independently. This is suitable for commercial/production, but not ideal for research.

To do effective co-design we need open, extensible hardware and open, extensible software.

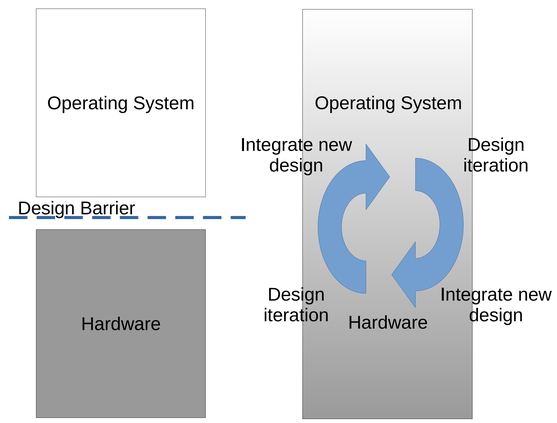


Figure 1. Co-design allows design decisions to flow across the "design barrier."

RISC-V

RISC-V is an open-standard hardware ISA developed by Berkeley, with an active community of researchers and contributors. First released in 2010, it has quickly gained traction for hardware research due to its openness and simplicity.

The ecosystem is now quite mature, with compiler toolchains, hardware simulators, and hardware design languages (HDLs).

Lightweight Kernels (vs Linux)

In HPC and elsewhere, Linux is now the dominant OS. This provides portability advantages, but creates massive design inertia. Using Linux for co-design requires many and potentially extensive modifications, which is non-trivial.

Lightweight kernels (LWKs) are an OS design approach that emphasizes simplicity and performance with modern design and minimum hardware abstraction, making them *ideal* as a co-design platform.

Methodology

We ported the Kitten to RISC-V, on both the SiFive Unmatched, an early development single-board computer, and the hardware emulator QEMU.

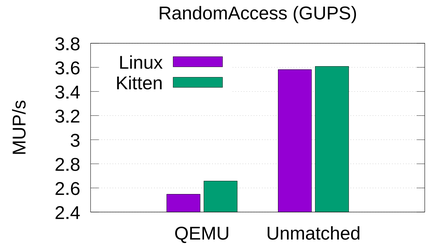
Kitten is a well-known lightweight kernel used in HPC applications, providing simplified resource allocation, simple kernel design, and a Linux-compatible ABI.

Performance Results

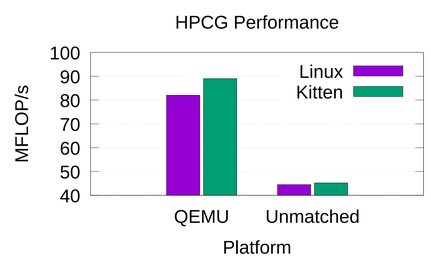
To evaluate the feasibility of our port we compared performance to Linux, which has existing support for RISC-V.

We evaluated performance using three well-known, benchmarks which highlight memory performance. Our benchmarks ran on a single core, as multicore support was not finished.

- *HPCG* – Conjugate gradient benchmark that is part of the Top500 benchmark suite
- *RandomAccess* – Random access benchmark measuring memory updates per second
- *STREAM* – Sustained memory bandwidth benchmark



(a) RandomAccess Performance



(b) HPCG performance across several systems. Result variance was 2% or less in all tests.

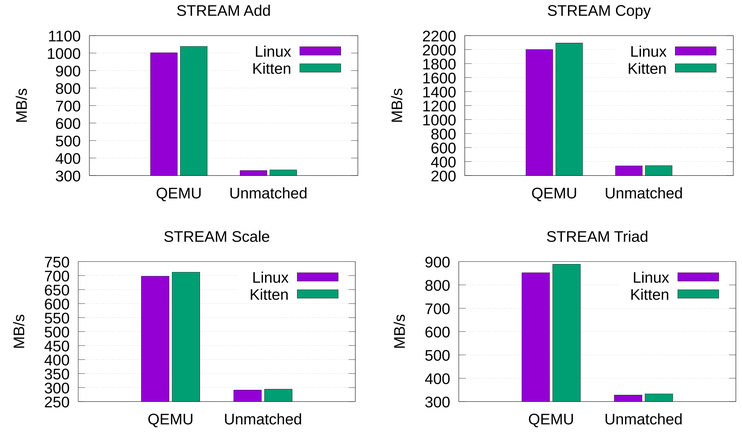


Figure 3. STREAM Benchmark Results

Conclusion

Our experiments show that LWKs can provide a simplified, extensible platform for co-design that has performance parity and even slight advantages compared to Linux, while retaining Linux ABI compatibility.

References

[1] Jack Dongarra, Michael A. Heroux, and Piotr Luszczek. HPCG Benchmark: a New Metric for Ranking High Performance Computing Systems. Technical Report ut-eecs-15-736, November 2015.

[2] Argonne National Laboratories. Argonne national laboratories benchmarks repo.

[3] John D. McCalpin. Memory Bandwidth and Machine Balance in Current High Performance Computers. *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pages 19–25, December 1995.

[4] Kevin Pedretti. Kitten: A Lightweight Operating System for Ultrascale Supercomputers. *Sandia Lab*, 2011.