Unikernel Linux and the Secrecy Project

Presented by Ethan Klein

Special thanks to my Supervisor Professor Orran Krieger and my mentor PhD student Eric Munson

Boston University College of Engineering

Unikernel Linux and interrupts

What is Unikernel Linux?

Unikernel Linux(UKL) is a project designed to improve the performance of a wide range of programs. The Project aims to move the program from the user-space level, down to the kernel. By doing so the program can avoid many repetitive performance throttles, such as system calls and interrupts.

Motivation behind Unikernel Linux

People have made unikernel OSes before, but usually begin with a specialized OS.

By instead patching Linux, UKL can feature the major performance improvements of a unikernel with the hardware acceleration and all the development tools of Linux.

Interrupts

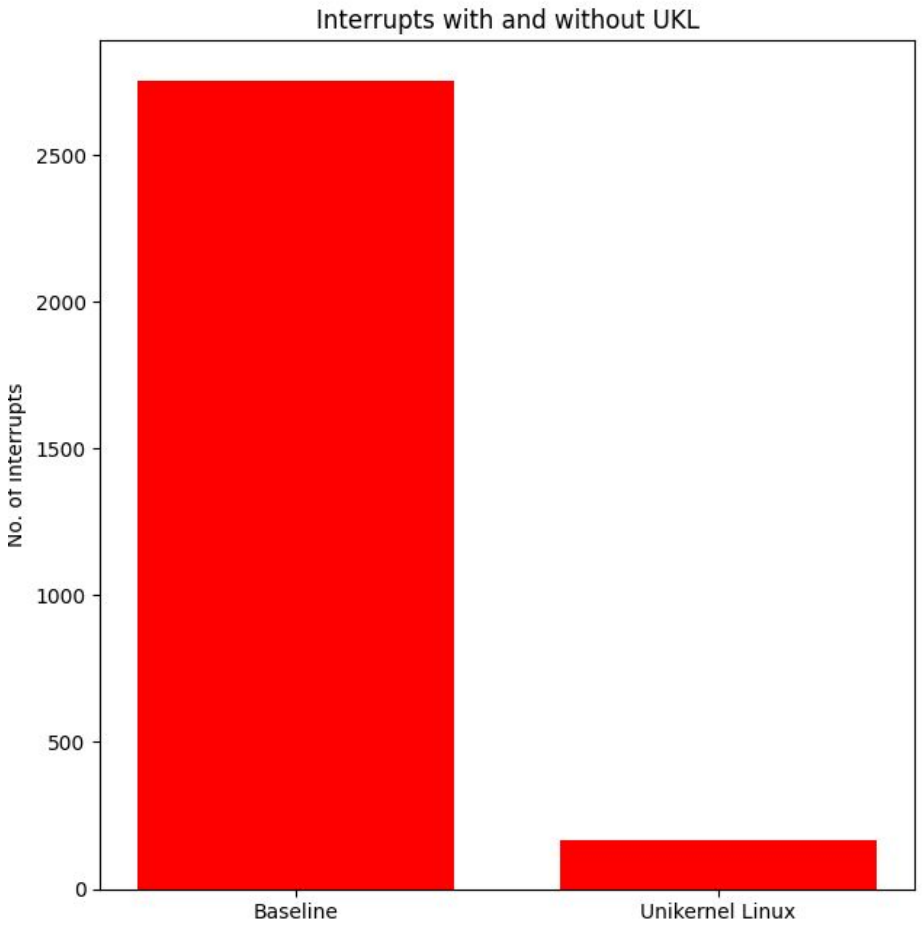
I compared the number of interrupts on a message passing program with and without UKL

My comparison revealed a significant decrease in the number of interrupts, greatly increased performance.

UKL vs. Baseline

The graph to the right demonstrates the difference made when running the program inside of UKL.

In fact, the program faced roughly 6% the number of interrupts that we saw in the baseline test.



Red Hat BOSTON UNIVERSITY

Interrupts and NAPI

NAPI(or “New API”) is an abstraction for interacting with hardware. NAPI includes an interrupt mitigation mode, called polling mode.

When run inside of UKL, we believe that the system is able to stay in this polling mode for longer.

Liburing

To ensure the comparison was as fair as possible, I edited the TCP code to use Liburing.

While the program did exhibit a major increase in performance, it was still significantly slower when compared to the same program run in UKL.

Secrecy

What is Secrecy?

Secrecy is a multi-party communication(MPC) program designed to allow many parties to combine data for analysis while preventing any sensitive data from being leaked between them.

This form of communication has many applications, like analysis of medical conditions without the sharing of patient information.

Motivation behind Secrecy

While not the first MPC program, the Secrecy team's logical and physical optimizations have allowed them to improve query performance by orders of magnitude.

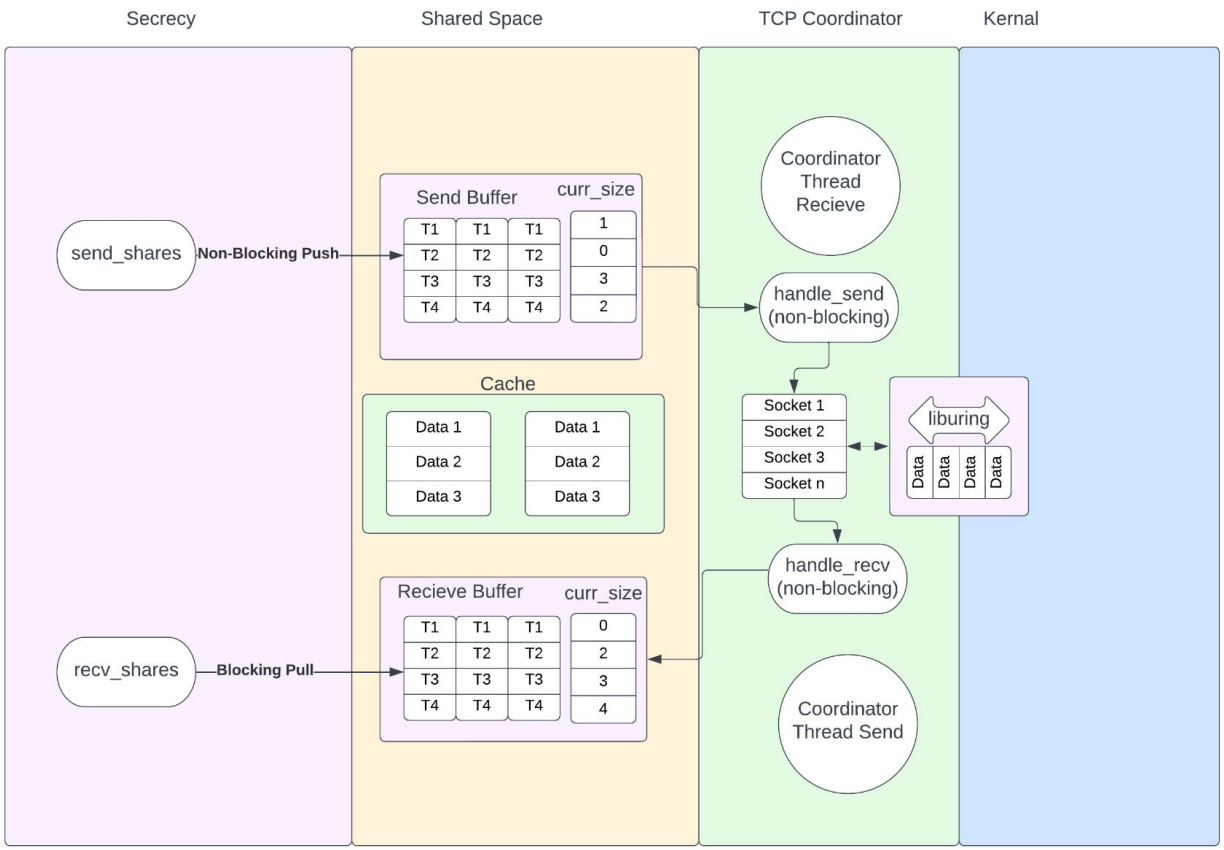
These optimizations also allow for reduced query execution cost when outsourcing the analysis to third-parties.

TCP Communication Layer

I worked with Master's student Ian Saucy on a TCP communication layer for Secrecy.

Our design had to multiplex messages from any given thread, through the network, and on to a corresponding thread on the next machine.

TCP Communication Diagram



TCP communication is called through *send_shares* and *recv_shares*.

Data is buffered before being handled by the coordinator thread or *recv_shares* is called.

Uses atomic variables to maintain a lockless design.

Why have a coordinator thread?

Without a coordinator, each thread would need the socket for each thread on every device in the communication ring.

Without a coordinator thread, this would come out to $sockets = threads^2 * (nodes - 1)$ per device.

By instead using a coordinator thread, we can limit this to just $sockets = threads * (nodes - 1)$ per device.

Zero Copy

One improvement we looked into was using zero copy data transfer.

By avoiding copying message data, and instead using the data inplace, we would be able to see faster reads and writes.

Unfortunately, this has not yet been implemented into our design.

Moving forward with UKL and Secrecy

Moving forward with Unikernel Linux

The paper for **Unikernel Linux has been accepted by EuroSys 2023**.

Trim latency by routing events to the UKL linked application directly.

Look into reproducing the dramatic decrease in interrupts more generically.

Moving forward with Secrecy

The implementation of the TCP communication is still a work in progress.

Ian and the team are working on making it more modular, so it is more testable and less bug prone. The fundamental design will remain the same.

Questions?

Editing this Template

- The info at the top of the slide can be changed. Simply go to the **View** drop-down menu, select **Header and Footer**, and make your edits in the box that appears.
  - The Header and Footer menu can also be used to enable/disable automatic slide numbering.

- To change the school name in the footer, go to the **View** drop-down menu and select **Master**, then **Slide Master**. Click on “School/college name here” and update accordingly.
  - Please note: To appear on all slides of your presentation, this change must be made on both of the first two slides of the master.