Implementing Bondage in FreeBSD

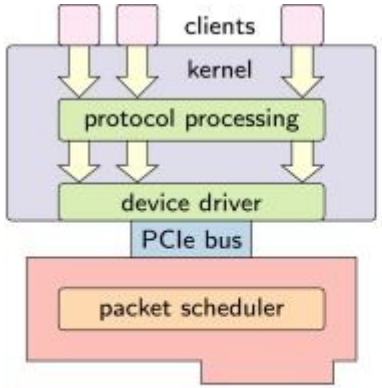
Alex Stanescu
Cal Lavicka
9/16/19
15-412 F’19

Implementing the PSPAT Subsystem in FreeBSD

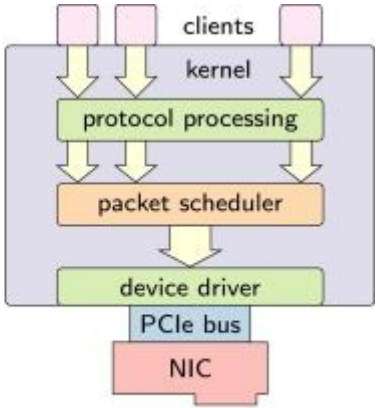
Alex Stanescu
Cal Lavicka
9/16/19
15-412 F’19

The Problem

- Virtual Machines in data centers can easily generate millions of packets/second
- Current systems have a single point of contention that causes major slow-downs, or introduces bugs.
  - 1(a): Problems here arise from hardware being designed with the average case in mind
  - Transmissions are completely serialized, and cannot sustain more than 1-2Mpps



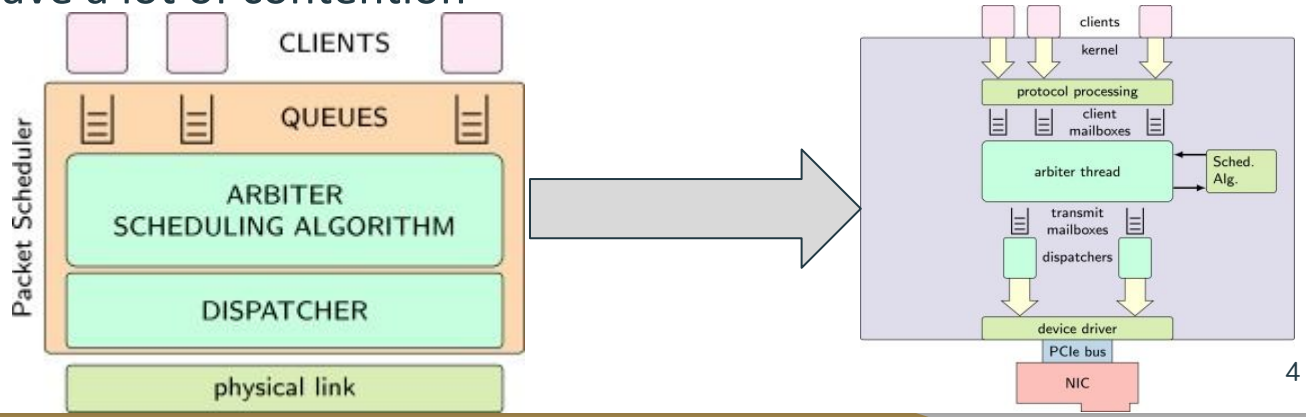
(a) Hardware Packet Scheduler using a multiqueue NIC.



(b) Software Packet Scheduler.

PSPAT Subsystem

- PSPAT = **P**arallel **S**cheduling **PA**rallel **T**ransmission
- PSPAT uses two sets of mailboxes to decouple clients, the scheduling algorithm and the actual delivery of packets to the NIC
  - Mailboxes are implemented as lock-free queues otherwise you have a lot of contention



Giuseppe

- Professor at Università di Pisa, Italy
- Chess Wizard
- Works on FreeBSD and Linux source
- Co-author on PSPAT paper
- Implementer of Linux Fork

Lines of code analysis

- In sys/net folder of FreeBSD: ~90000 lines of code
- Linux implementation: ~2000 lines of code
- Failed implementation: ~1500 lines of code
- Expected lines for a good implementation: ~2000 lines
  - Mailbox ~ 500
  - Dispatching ~1500

Licensing

- Existing PSPAT implementation (made by Giuseppe) in Linux, but not merged, so licensing is up in the air (?)
- FreeBSD has a BSD license
  - Redistributions retain license, but not modifications
  - Binaries also have modifications
  - So, better than GPL
  - https://www.freebsd.org/internal/software-license.html

Resources

- https://github.com/giuseppelettieri/linux-pspat/tree/pspat-4.13/net/pspat <- Linux Implementation
- https://github.com/theGodlessLakra/freebsd-pspat/tree/projects/pspat/sys/net/pspat <- Failed FreeBSD Implementation
- https://www.sciencedirect.com/science/article/pii/S0140366417300804 <- PSPAT paper
- Giuseppe

std::acceptance_process

- Standard process is
  - Either submit a change via the FreeBSD bug submission form
  - Or submit a patch to the appropriate mailing list and have senior members approve it
- However, this project is going to go through Giuseppe, who will review code and help us decide how to merge once we're done.