Proposed Projects

From CSL WIKI

Long Term

This is a long term projects that can easily cover MSc thesis and more. For undergrads each such project must be split to many short term projects, each one with achievable objectives that fits in single semester time frame (about 150 hours) with measurable outcome.

1. QoS for NVMe Over Fabrics.
   - In long term, we want to show that there is a problem with QoS in NVMe over Fabrics, propose a solution, implement it in Linux kernel, show that it is working, and write an academic paper about it.
   - In short term, start with understanding basic terms of NVMe, SSD, block level QoS, RDMA, NVMe over Fabrics, building a setup, applying patches from experimental NVMe over fabrics Linux kernel tree, get familiar with environment and tools, understand concepts of QoS, etc.

2. SRIOV Security
   - This project also easily fits in MSc thesis. It is more focused, we have good understanding of it, targets are clear and we have higher confidence in it. I think this one is a "clear cut" for master thesis.

3. CPU NUMA effects
   - Currently just a place holder. Empirically, we see the phenomenon of CPUs NUMA effect on performance but not sure about what is the problem and how to solve it. I have some thoughts about it, not well defined yet.

Short Term

Short term projects, fits for a single semester, self contained, no long term commitments required, just fun, mostly programming effort.

1. NAPI-like mechanism for high speed storage NVMe SSD
2. Scalability of VT-d and CPU posted interrupts in Linux
3. Evaluate performance of VT-d and CPU posted interrupts in Linux
4. Operating system performance monitoring.
   - Brendan D. Gregg page (http://www.brendangregg.com/index.html) is a huge source of performance analysis thoughts and it would be extremely useful to bring this knowledge into our research group: study one specific topic and make a lecture with hands-ons.
   - It doesn't have to be just writing benchmark tools wrappers or performance results parser in python but also get familiar with
different performance monitoring techniques for different sections of Linux kernel, and deliver a lecture, make a hands-on in Dan's group, write useful script examples in wiki.

5. JOS on ARM: virtual memory initialization.
   - It is half-baked currently.

6. JOS: complete porting network driver for modern NICs (40Gbe Mellanox and Intel)

7. JOS 64bit, make it multi-threaded and port user-level high speed TCP/IP stack as mTCP

**QEMU/KVM**

Want to play with a real hypervisor? Do a project for QEMU/KVM: TODO (http://www.linux-kvm.org/page/TODO)


- This page was last modified on 3 November 2016, at 12:10.