Accelerating Computational Storage Over NVMe with RISC-V

Stephen Bates, CTO
Eideticom
Outline

- What is RISC-V (I’m joking)
- What is NVM Express?
- NVMe performance on a RISC-V SoC
- What is NVMe computation?
- What is p2pdma?
- p2pdma and NVMe Computational Storage for RISC-V
What is NVM Express?

- A high performance protocol designed for communicating with Non-Volatile Memory (NVM).
- Originally designed for PCIe but now also works over RDMA and TCP/IP.
- **Superb support in Linux.**
- Now over 50% of all new SSDs sold are NVMe (rather than SATA).
- A healthy roadmap of upcoming features:
  - Low-latency (e.g. Optane).
  - Computational Storage (compute on/near SSD).
  - IO Quality of Service

An open standard! Not unlike RISC-V ;-)!!
NVMe Performance on a RISC-V SoC

- NVM Express is designed to be very efficient wrt the host CPU.
  - Minimal PCI load/stores.
  - High performance DMA done by the SSD, not the CPU.
  - Very parallel (uses blk-mq in Linux) to scale across CPU cores.

- However on RISC-V SoC tested there are some issues:
  - DDR bandwidth can become a bottleneck.
  - PCIe bandwidth can become a bottleneck.
  - CPU cores can only issue so many IO per second.
  - The VMEMMAP for RISC-V is not great for this SoC (32 bit PCI host bridge).
  - No IOMMU so SWIOTLB has to get involved.

![Throughput vs Block Size for NVMe on RISC-V](attachment:image.png)

Performance saturates at 225MB/s due to limits in SoC IO.
What is p2pdma?

- p2pdma transfers bypass CPU memory
- p2pdma uses PCIe EP’s memory (e.g., NVMe CMB, PCIe BAR)
- A P2P capable Root Complex or PCIe switch is needed
- The p2pdma framework in Linux allows PCI EPs to donate and/or consume p2pmem for DMAs
- p2pdma framework upstreamed in 4.20 for x86_64 and is very arch specific.
- p2pdma ported to RISC-V by Eideticom. Upstreaming of this code is under discussion.
What is NVMe Computational Storage?

- NVMe is a great way for talking to PCIe devices.
- Those EPs don’t have to be SSDs, they can be devices that can do computation.
- We can leverage all the goodness of the NVMe eco-system to connect accelerators to a host CPU.
- This concept is called NVMe Computational Storage.
- Right now SNIA and the NVM Express standards body are discussing how to standardize this.
- In the meantime startups (like us) are shipping product in this space.

- NVM Express End Point.
- Performs compression, RAID, dedupe etc.
- Leverage P2PDMA via a Controller Memory Buffer
- Standard NVMe form-factor (2.5”).
- PCIe Gen4
What is NVMe Computational Storage?
Using P2PDMA and NVM Express based Computational Storage we can decouple the data-processing performance from the SoC performance.

For larger IO sizes we can saturate PCIe Gen3x4 (~3.4GB/s).

With PCIe Gen4 we can hit over 5GB/s!

A scalable approach. Add more NoLoads and NVMe and get more performance.

Can introduce (R)NICs to add network connectivity and continue to scale.

Demo achieves >3GB/s of zlib compression using Eideticom NoLoad and Samsung NVM Express SSD on SiFive RISC-V based system.
Come see the Demo!

**NoLoad™ NVMe Scale-Out Compression**

Freedom U540 SoC and External PCIe Switch

1500x Faster than Software (5GB/s vs 3MB/s/core)

- Zero load on CPU Memory
- IO Performance decoupled from CPU performance!
- IO at PCIe Gen4, CPU at PCIe Gen2!

Enabled with Eideticom’s p2pdma Linux framework

**Come See our Demo in the SiFive Booth to Learn More!**
P2PDMA and NVMe Computational Storage for RISC-V

- With p2dma IO is no longer limited by SoC IO constraints (225MB/s->2GB/s (or more))..
- In this case P2PDMA of 2GB/s were achieved. This is the limitation of the NVMe SSD.
- Adding more NVMe SSDs and NoLoads would allow for linear scaling.
- Using a PCIe Gen4 switch and PCIe Gen4 NoLoad is also an option.
Conclusions and Future Work

● Linux on RISC-V is a fantastic platform for NVM Express:
  ○ Open hardware and software allows innovation.
  ○ Linux has great NVMe support for any ARCH!
  ○ RISC-V allows for open development of IO devices like root-ports, host-bridges, PCIe switches and IOMMUs.

● Classic NVMe performance has some issues:
  ○ DDR bandwidth can be a bottleneck
  ○ PCIe bandwidth can be a bottleneck
  ○ The physical memory map is awkward when PCI host bridge is 32 bit.
  ○ No IOMMU means SWIOTLB needed

● P2P NVMe performance has no issues:
  ○ DDR bandwidth no longer on performance path
  ○ PCIe Gen4 capable with right hardware.

● Next Steps
  ○ Add p2pdma NVMe over Fabrics
Much of the code needed for this talk and demo are GitHub’ed and Docker-ized. Ping me for more details.