**OFI WG Data Storage / Data Access Subteam Monterey F-2-F – 03/15/2015**

**OFIWG Download Site:** [www.openfabrics.org](http://www.openfabrics.org) 🡪OFED/OFA Resources 🡪 OpenFabrics Interfaces WG

**Agenda**

- Sunday, 3/15, 8am to 12pm. DS/DA only, followed by joint OFI meeting Monday evening

- OFI WG will be meeting Sunday afternoon. All are encouraged to attend.

- There will be a dial-in bridge (number to be published), can Cisco provide a day-long webex session?

- Paul to send out a calendar invitation with relevant details.

- Everyone should plan to attend Doug Oucharek’s session at the OFA workshop.

-Paul to send a pointer to the current workshop agenda.

- f-2-f agenda:

 - ~~settle on repo strategy~~ (done)

 - ~~settle on the architecture proposal from Intel~~ (done)

 - prioritization of requirements – (mostly done, but needs a review)

 - use cases for each requirement

 - detailed walk through of key objects and methods from current OFI repo

**Requirements Discussion – See the prioritized requirements document https://www.openfabrics.org/downloads/ofiwg/dsda\_rqmts/**

Completions

- does completion include placement on the remote end? Is data within the persistent domain?

- today, we get completions from the NIC. Recognize that things change when you cross the domain into the application space.

- Proposal: begin by relying on the existing IB completion semantics.

 - add the ability to specify that the provider supports ‘enhanced completion’ (committed to memory).

 - or, add another operation, above and beyond the normal completion, indicating commitment to memory.

 - or, make it ‘application-signaled’ completions.

- changed the ‘must have’ requirement for 2.2 to ‘highly desirable’.

 2.2: Data placement committed – data have been completely placed into persistence domain.

Ordering – 1.2, 1.3

- lazy ordering intended to help achieve highest possible IOPS

- fencing proposal is similar to existing fence, but extended to the remote side. Not clear if this offers much in the way of performance improvement, although it allows requests to be queued at the responder side.

Reliable datagram, reliable multicast

- There is a lustre use case for reliable datagrams to allow the number of OSTs to be scaled to reach exascale. Using RD reduces the connection footprint as the number of OSTs.

NVM specific commands

- ‘Trim’ is similar/same as SCSI unmap

- a means of forcing the device to execute garbage collection

- releases resources in the device.

-it’s a flash thing, kind of like a flush.

- use case: make certain blocks of the remote (peer) memory inaccessible: ‘these blocks are still mine, but the data has been scrubbed.

- use case: security – scrub a block that is no longer in use. Alternative would be for the application to write zeros.

- does ‘trim’ require the memory to be obliterated, or simply release the resource for re-use

- how much of the ‘e’ part of NVMe do we need to include?

SanDisk – see Bart vA’s slides

- customers are saying, if you do not provide RDMA support, we do not want to hear from you.

- want protocol offload, minimize CPU utilization, etc.

- Bart walked through the multi-queue proposal for SCSI. See his slides posted to the OFI download site.

Connecting to the FC community

- FC-NVMe – working on NVMe over FC. Doesn’t include any enhanced mechanisms for data movement, relies solely on FC.

NVDimms in the Linux kernel – see the link sent by Bart

[http://thread.gmane.org/gmane.linux.kernel/1901204/focus=1901226](http://thread.gmane.org/gmane.linux.kernel/1901204/focus%3D1901226)

- this is a patch series that allows support for NVDimms from the kernel

- not clear if this patch supports byte addressable memory or not.

GIThub

- approach a kernel maintainer in person, to ascertain basic interest

 - google – who’s working on flash stuff?

 - jens Axboe - facebook

- implement the kernel driver itself

- establish a user base to ensure that it is well-tested

- need a kernel component that actually uses

 - NFSoRDMA

 - NVMf

I am also attaching a useful email from Bart from October 6, 2014. Although probably not yet critical, this will become very useful as we approach the point of having something to push upstream.

“Regarding what has been discussed during the conference call earlier

today: if you would like to reach out to the Linux kernel maintainers to discuss storage API changes further, I think one of the following two events would be appropriate:

\* The Linux Filesystem, Storage and Memory Management Summit.

\* The file systems and storage track of the Linux Plumbers Conference.

More information about these events can be found e.g. here:

\* The 2014 Linux Filesystem, Storage, and Memory Management Summit (<http://lwn.net/Articles/LSFMM2014/>).

\* Linux Plumbers Conference, File and Storage Systems, 15-17 Oct. 2014 track (<http://www.linuxplumbersconf.org/2014/ocw/events/LPC2014/tracks/303>).

Although the LSF is the most interesting event, the barrier for acceptance of proposals for the LSF is higher than that of the Linux Plumbers Conference. Last year I attended the LSF and presented a topic about reducing the latency of the SCSI stack (<http://lwn.net/Articles/548368/>). By the way, most of what has been proposed in April 2013 is now upstream in Linux kernel 3.17 (see e.g.

SCSI Multi-Queue Performance Appears Great For Linux 3.17 (<http://www.phoronix.com/scan.php?page=news_item&px=MTcyMjk)>).

I plan to attend the Linux Plumbers Conference next week in Düsseldorf.

Jens Axboe, the current Linux block layer maintainer promised to attend the Plumbers conference.

If you would like to talk during one of these events about reducing latency of one or more Linux storage API's I think that quite some people will be interested. Quite some hyperscale (=cloud computing) companies are currently researching how to reduce the latency of their storage stack. Jens Axboe is working for one of these hyperscale companies, namely Facebook.”