**OFI WG Data Storage / Data Access Subteam Weekly telecom – 03/24/2015**

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

**Agenda**

* roll call, agenda bashing
* Summarize F-2-F
* How the SNIA work fits in, including an introduction
* Continue review of key objects and methods

**SNIA NVM Programming Model Discussion – Doug Voigt**

- four sessions at the Workshop last week, two on NVMe, one from Bernard, and one from Doug (SNIA)

- the NVMe ones are NOT part of SNIA

- not yet completely clear with which part of OFI SNIA should engage – DS/DA seems to make sense because of its storage orientation. Note that the SNIA NVM Prog Model has some user level components, which is consistent with other parts of the DS/DA work.

- Quick review of Doug’s slides from the workshop

- SNIA NVM Prog Model TWG (TWG) is suggesting the use of Memory Mapped Files. Essentially, open a file and give it space in the application virtual memory space. The buffers in the page cache are mapped and through a page fault mechanism, they appear in the users’ virtual memory space.

- Version 1 of SNIA NVM – similar in concept to verbs, but chose not to call it an API because different kernels will implement the API differently. This is virtually identical to ‘Verbs’ as described in the IB spec as a semantic description.

-NVDIMM – co-locate RAM and NVM on a single DIMM along with enough power to allow the RAM to be dumped to NVRAM.

- Focusing on NVM.PM

 -NVM.PM.VOLUME mode

 -NVM.PM.FILE mode

- Significant changes in NVMP Ver 1.1: atomicity and high availability:

 - High availability becomes important for remote access.

 - SNIA has published a white paper on this topic, including a set of requirements:

assurance of remote durability, efficient byte range transfers, efficient large transfers, atomicity of fundamental data types, resource recovery and hardware fencing after failure.

Want to make sure that these requirements can be boiled down and are included in our requirements document.

- NVM.PM.File.Map, Sync, Remote Optimized Flush: flushes CPU cache for indicated range.

Optimized flush: multiple ranges from user space (analogous to a disk write)

Optimized flush and verify: similar to write and verify read back with a disk drive.

Both assure that data is written to persistent media, but without the verify you don’t’ have assurance that it was actually written.

- How this relates to OFI WG:

 The semantic operations described above would be implemented as an API.

 That API would be a consumer of APIs produced by OFI / DS/DA.

- Could be modeled as a local and remote filesystem in kernel space, such that when a write or flush occurs, it is reflected to the remote filesystem automagically.

**Reviewing results of last week’s F2F**

- Completion semantics: OFI came up with three levels of completion semantics:

 1. local completion

2. remote completion (reliable service only) – downstream entity has acknowledged receiving the data.

3. remote commit (reliable service only) – data has been received at the remote end and is committed to memory.

- Add a type 4, where data is not only committed to memory, but is within the persistence domain at the far end.

- Need a use case to describe this.

**Agenda for next meeting**

- Continue with Doug Voigt’s discussion of the SNIA NVM Programming model

- Review completion use cases, discuss feedback from OFI WG discussion.

- Continue with EP setup for the client side (Stan)

**Next regular telecom**

Next meeting: Tuesday, 3/31/15.

8am-9am Pacific daylight time

**NOTE:** We have switched over to using Webex (courtesy of Cisco). The URL for joining meetings is:

<https://cisco.webex.com/cisco/j.php?J=200935598&PW=67935ad6df07030d5f05044a5b0f>