**OFI WG telecom – 08/15/2017**

**Agenda:**

* Continue Persistent Memory Discussion
* Slides from Sean on NVML

Minutes of 8/1 and 8/8 meetings are included here in order to give a complete picture of the proceedings. Scroll down to find the 8/15/17 minutes.

**PM over Fabrics – NVML introduction – Chet Douglas (Intel) - 8/1/17 minutes**

* ‘non-allocating’ means that data is written direct to the persistence domain, bypassing L3 cache
* DDIO means a direct write to L3 (as opposed to non-allocating).
* **Reqmt: include persistence as part of memory registration, such that the NIC will know if inbound data is bound for a persistence domain or not. Today, it’s on a per RNIC basis…all writes to the same RNIC are handled the same. Need to discuss/debate the granularity of this.**
* CLWB + fence / CLFLUSHOPT + fence / CLFLUSH / NR store + fence / + WBINVD (kernel only) are all CPU specific cache flushing mechanisms and are all abstracted by NVML.
* NVML is user space, open source <http://pmem.io/> librpmem, librmemd are the relevant libraries.
* Use case – data replication: Application opens a file and does R/Ws to the file, then does ‘make persist’ (addr, len) to make a certain range persistent. This same region is then replicated to remote memory
* NVML began as a solution around local NVDIMM. Its main use case is for replication of data between the local and remote nodes.
* **Agreed to continue the discussion during a special off-cadence meeting next week – 8/8/17.**
* **LOGISTICS WILL BE POSTED TO THE OFA CALENDAR:** <https://openfabrics.org/index.php/ofa-calendar.html>
* **Please check the calendar.**

**Resuming the Discussion - 8/8/17**

**Slides are found at:**

**http://downloads.openfabrics.org/WorkGroups/ofiwg/dsda\_nvm/PMoF-%20Architecture%20Overview%20&%20NVML%20Implementation.pptx**

* Last time we got through the architecture of PMoF and how it might be realized in NVML
* Reminder – NVML is indeed built around a replication paradigm.
* The pmemAware file system is used to create the memory mapped file which is then used by the application to do reads and writes to memory.
* For replication purposes, librpmem on the local node communicates directly with librpmemd on the remote node. A secure socket is used to exchange authentication items between themselves to set up the operation.
* NVML can be thought of as a collection of libraries – libpmem, libpmemobj, libpmemblk, libpmemlog, etc. See <http://pmem.io>
* Could take the discussion at this point in one or two directions:
  + Use cases
  + Talk about the API we support
  + We’ve chosen to go down the Use Cases path
* **Reqmt – a discovery mechanism that makes a consumer aware of the persistence characteristics of remote memory.**

**Resuming the discussion – 8/15/17**

**Sean’s slides – based on his review of NVML (see pmem.io)**

**See the slides here:**

**http://downloads.openfabrics.org/WorkGroups/ofiwg/dsda\_nvm/ofi-pmem.pptx**

* Two libraries: librpmem
* Librpmem
  + Assumes that there is an SSH connection used for establishing the RDMA connection.
  + Main focus is on Replication – assumption is that there is local pmem
  + rpmem\_create – creates a file at the target
  + rpmem\_open – opens remote file
  + rpmem\_persist – blocking call that forces remote data to be written to persistent memory. Takes an address followed by a length.
  + rpmem\_read –
  + Not clear if there is ordering between multiple connections
  + Would be helpful to have a discussion with someone from SNIA to answer questions about connection ordering, etc.
* libpmem
  + a peer to repmem
  + used for accessing local memory. An application would use both of these in tandem.
  + Based on pmem\_map\_file – which is similar to mmap
  + pmem\_persist – local version of rpmem\_persist
  + pmem\_msync – an optimization of pmem\_persist, allows mixing pmem w/ normal mem
  + there are a bunch more of these
* Intel-specific CPU instructions
  + CLFLUSHOPT
  + CLWB
* From a libfabric perspective - options
  + Persistent writes – RMA write to peer pmem MR
  + Persistent reads – RMA reads from remote memory (either persistent or not) into local PM
  + Atomic operations
  + Commit semantics – what is the granularity of commits?
* Proposal – FI\_GETINFO
  + New capability bit: FI\_RMA\_PMEM. Provider supports RMA to/fr PM. Allows us to add atomics later as a new capability flag.
* Proposal – Memory registration
  + New flag that indicates if the remote region is persistent or not. May not be needed?
  + How do the new mr\_mode instructions (rel 1.5) interact with PM?

**Recordings:**

|  |  |  |
| --- | --- | --- |
| **OFIWG/libfabric meeting-20170801 1608-1** | | |
| Tuesday, August 1, 2017 | | |
| 12:08 pm  |  Eastern Daylight Time (New York, GMT-04:00) | | |
| [**Play recording**](https://cisco.webex.com/ciscosales/lsr.php?RCID=2b980800b3a34440b65b1e232625efcf) (54 min 47 sec) | |
| Recording password: UvzWRwE7 |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **OFIWG one-off meeting: The Buttery Goodness of NVMoF-20170808 1609-1** | | |
| Tuesday, August 8, 2017 | | |
| 12:09 pm  |  Eastern Daylight Time (New York, GMT-04:00) | | |
| [**Play recording**](https://cisco.webex.com/ciscosales/lsr.php?RCID=bc990c7104c84abab3db23bff135cb4c) (1 hr 0 min 35 sec) | |
| Recording password: VsWbPtT7 |  |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **OFIWG/libfabric meeting-20170815 1605-1** | | |
| Tuesday, August 15, 2017 | | |
| 12:05 pm  |  Eastern Daylight Time (New York, GMT-04:00) | | |
| [**Play recording**](https://cisco.webex.com/ciscosales/lsr.php?RCID=8149cbde77d94a28ac9cab9fe34de655) (56 min 44 sec) | |
| Recording password: WmfRzPD7 |  |

|  |
| --- |
|  |

**Webex link:** See the OFA central calendar for meeting logistics. <https://openfabrics.org/index.php/ofa-calendar.html>

**OFIWG Download Site:** [www.openfabrics.org/downloads/OFIWG](http://www.openfabrics.org/downloads/OFIWG)

**Github:** <https://github.com/ofiwg/libfabric>

**OFI Software Download Site:** [www.openfabrics.org/downloads/OFI](http://www.openfabrics.org/downloads/OFIWG)

**Next regular telecon**

Next meeting: Tuesday, 8/29/17

9am – 10am Pacific daylight time