**OFI WG telecon – 07/16/2019**

**Agenda:**

1. Opens, Agenda bashing
2. Collective offloads
	1. Need to specify / allocate resources needed to complete a collective task?
	2. This is separate from group identification and creation.
3. Updates to the hooking provider
	1. Plea for help
4. NUMA attribute?
5. Peer-to-peer transfers for e.g. GPU or heterogenous memory

**Opens**

-none-

**Collectives Offloads**

Updated Collectives PR – moved from being an overlay on atomics onto being a distinct set of interfaces for collectives.

Currently, there is a way to define the members of a collective group, and a way to join the group, but there is nothing in the API today for allocating resources, (other than saying “go join the group”).

* No ability to specify queue depths (number of collectives in flight at once),
* No way to reserve switch buffer space,
* Provider knows only the members of the group.

Venkat currently working on an implementation of collectives with the TCP provider, which may provide some insights, but it doesn’t support switch-based offloads.

Suggestions:

* Survey vendors to see if there is a set of resources common to all providers
* Review presentations from the workshop on SHARP (e.g. Michael Aguilar’s session)

Static inline wrappers remain the same, three or four new collectives call which look very much like the original version based on atomics. Barrier has its own entry point, the rest map onto a write-read call (reverse of what happens with an atomic, which are read-write)

**Hooking Provider – seeking help**

* Arun has started work on the hooking provider, but need volunteers to do some work to implement it. Want an easy way to implement a new hooking provider, maybe even to the level of an individual function call, i.e. less boiler plate to implement a new hook.

**NUMA Attribute?**

NIC attributes currently report PCIe information, could the NUMA node the device is on also be reported? Getting the NUMA information given the PCIe attributes is straightforward; the NUMA node could be reported as part of the NIC attributes. MPIs today use hw\_loc to gather topology information and there is a mechanism to allow applications to share the resulting data, avoiding the scalability issue of having every instance of the application discover the topology independently. Considering expanding the API to allow an application to pass this blob of data, including NUMAness, down to the provider for its use, if it can. For example, the provider could access information about its NIC and cross reference that with the hwloc data to develop a view of the topology, including NUMAness This avoids the need for every provider to re—discover this information. The question is if there is a demand from the application for that NUMA info that the provider has developed.

**GPU peer-to-peer, heterogenous memory accesses, etc.**

More work is needed here, starting with the existing proposal. Sean will distribute the existing slides.

**Next meeting**

Tuesday, July 30, 2019

9:00 – 10:00AM PST

**Recording:**

|  |
| --- |
| **OFIWG every-two-week meeting (2019)-20190716 1604-1** |
| Tuesday, July 16, 2019 |
| 12:04 pm  |  Eastern Daylight Time (New York, GMT-04:00) |

|  |
| --- |
|   |

|  |
| --- |
| [**Play recording**](https://cisco.webex.com/cisco/lsr.php?RCID=f89f57e3c73c4b7998a5c9b26656501f) (55 min 3 sec) |
| Recording password: VujjcXT3 |  |

**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)