Mercury Updates
Mercury

- Base low-level RPC component used for communication between Mochi services
  - Always consider higher-level components first before directly using the mercury API

- In-depth documentation:
  - https://mercury-hpc.github.io

- Two main data transfer methods
  - Point-to-point RPC through eager messages
    - Connection-less semantics
  - Bulk data through RDMA
    - No memory copy
    - Requires memory registration internally
Status and Roadmap

- **2.1.0 version released**
  - Added initial support for UCX
  - Bug fixes

- **2.2.0rc1 version released**
  - OFI / UCX:
    - Better handling of addressing formats and support for IPv6
    - Support device (CUDA, ROCm) to host transfers
  - OFI:
    - Support HPE Slingshot 11 through cxi provider
    - Support NIC locality through hwloc
  - UCX:
    - Switch to active messages for RPC requests

- **PSM/PSM2 (new plugin to support OmniPath)**
  - Hopeful to have psm2 supported through OFI opx provider in the future

- **Improved diagnostics through `diag` log subsystem and improved OFI provider selection information**

- **Checksums disabled by default**
  - Introduced checksum levels

- **3.0.0 version**
  - Extend addressing capabilities to address contexts (enhanced multithreading support and composability)
    - Improved support to OFI scalable endpoints
## Supported Transports

<table>
<thead>
<tr>
<th></th>
<th>tcp</th>
<th>verbs</th>
<th>shm</th>
<th>psm</th>
<th>psm2</th>
<th>gni</th>
<th>cxi</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFI</td>
<td>✓</td>
<td>✓</td>
<td>✗*</td>
<td>✗*</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>UCX</td>
<td>✓</td>
<td>✓</td>
<td>✗*</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>SM</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>PSM</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>BMI</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

* Not explicitly supported by mercury but may be supported by underlying library
Slingshot Support and Locality Awareness

- Slingshot 11 supported w/ OFI cxi provider
  - Support only native addressing (i.e., no IP)
    - “ofi+cxi://cxi[0-9]:[0-510]”
  - All Mercury features supported by cxi provider except blocking progress
    - Busy spinning progress at the moment but will be resolved in a future libfabric update

- Locality awareness
  - Enabled when no interface is explicitly selected
    - “ofi+cxi://[0-510]” or “ofi+cxi”
  - Uses PCI NIC information from libfabric and hwloc output to match closest NIC

- As for Cray GNI, communication between separate jobs may require key exchange (still under evaluation)

Credit: https://docs.olcf.ornl.gov/systems/crusher_quick_start_guide.html#system-overview
UCX

- Relies on UCP API of UCX
  - Combines both active and tagged messages
  - Supports native RDMA for bulk data
- All features of Mercury now supported
  - Only tested using tcp and verbs (in general ~1us faster than OFI on verbs)
- Supports only IP type of addressing
  - `ucx+all://<hostname, IP, iface>:port`
    - Recommended to always use “all” and let UCX decide on best protocol to use
- Thread safety mode can be relaxed w/ init info
  - Default is thread-safe
- Additional options passed through UCX environment variables
Host to Device RDMA Transfers

- New routine to provide memory type information on bulk handle creation
  - `HG_Bulk_create_attr()` with `HG_MEM_TYPE_CUDA`, `HG_MEM_TYPE_ROCM`, etc (default is `HG_MEM_TYPE_HOST`)
  - Supported by both OFI and UCX plugins
    - Only verbs and cxi for OFI
    - Transparent for UCX
- RPC requests and response are always sent between CPUs
  - Eager bulk transfers disabled when using device memory to prevent extra copy from device to CPU
- More testing remains to be done
Logging and Diagnostics

```bash
export HG_LOG_LEVEL=debug
export HG_LOG_SUBSYS=diag
```

Diagnostics counters can tell you about the type of RPCs that were sent / received.

Debug output for OFI info give information about OFI provider.
Compare with `fi_info` output.

```bash
export HG_LOG_LEVEL=debug
export HG_LOG_SUBSYS=cls
```

(Similar debug output for UCX)