Cornelis Networks Omni-path Express (OPX) Libfabric provider - Observability and Tuning

Tim Thompson - Cornelis Networks
August 2023
Notices and Disclaimers

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH CORNELIS NETWORKS PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN CORNELIS NETWORKS’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, CORNELIS NETWORKS ASSUMES NO LIABILITY WHATSOEVER, AND CORNELIS NETWORKS DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF CORNELIS NETWORKS PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. CORNELIS NETWORKS PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.

Cornelis Networks may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined”. Cornelis Networks reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice. Roadmap not reflective of exact launch granularity and timing. The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Any code names featured are used internally within Cornelis Networks to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Cornelis Networks to use code names in advertising, promotion or marketing of any product or services and any such use of Cornelis Networks’ internal code names is at the sole risk of the user.

All products, computer systems, dates and figures specified are preliminary based on current expectations and are subject to change without notice. Material in this presentation is intended as product positioning and not approved end user messaging.

Performance tests are measured using specific computer systems, components, software, operations, and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Cornelis Networks technologies’ features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration.

Cornelis, Omni-Path, Omni-Path Express, and the Cornelis Networks logo belong to Cornelis Networks, Inc. Other names and brands may be claimed as the property of others.

Copyright © 2023, Cornelis Networks, Inc. All rights reserved.
Agenda

• Who is Cornelis Networks?
• What is OPX and how to use it?
• OPX features and topology
• How to assess and tune Fabric performance
• Observability with OPX
Cornelis Networks

- From Startup -> Silverstorm -> QLogic (IB) -> Intel (OPA) -> Cornelis (OPX)
- Omni-Path Architecture (OPA) 100 Gbps Fabric
- Spun out of Intel 2+ years ago
  - Bring the customers and technology
- Carry OPA flag forward and advance the technology
- Next Generation is 400Gbps (CN 5000)
- New fabric topologies
What is OPX?

• OPX is a libfabric (OFI) provider
• Replacing PSM2
  • Not a re-write or refactor of PSM2
  • Origins in BGQ (Blue Gene) provider
• Highly optimized
  • Support for RDMA and GPU RDMA
  • MVAPICH3.0b w/libfabric
• Does not require changes to hfi1 driver or Fabric Manager
• Really is a drop in replacement for PSM2
## Software Stack

<table>
<thead>
<tr>
<th>File Systems</th>
<th>I/O ULps</th>
<th>IPoF, SRP, iSER, uDAPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVAPICH2</td>
<td>OpenMPI</td>
<td>NCCL</td>
</tr>
<tr>
<td>Intel MPI</td>
<td>OpenMPI</td>
<td>MVAPICH2</td>
</tr>
<tr>
<td>GASNet</td>
<td>Sandia SHMEM</td>
<td>Charm++</td>
</tr>
<tr>
<td>Chapel</td>
<td>DAOS</td>
<td>GPU Support [NVIDIA, AMD, Intel]</td>
</tr>
<tr>
<td>NCCL</td>
<td>PyTorch</td>
<td>TensorFlow</td>
</tr>
<tr>
<td>rsockets</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OMNI-Path Express Native OFI Provider**

**OFI Libfabric**

**OFA Verbs**

**OMNI-Path HFI Driver**

**OMNI-Path Adapter**
How to use OPX?

• Best-practice, install Cornelis OPXS package
• Optional: Download and compile Libfabric upstream and upstream Linux kernel with newer hfi1 driver
• Build Libfabric
• Make sure MVAPICH is not configured for internal-only provider
• Set LD_LIBRARY_PATH=/mylib/libfabric.so: $LD_LIBRARY_PATH
• Set FI_PROVIDER=opx
• First run check, set FI_LOG_LEVEL=info and look for “HFI1 PIO credits”. If you see this, you’re running the OPX provider
OPX provider topology

MPI Application

MVAPICH (Messaging Middleware)

OPX Endpoint

CQ (Completion Queues)  CNTR (Counters)  OPS (Bind, Close, Control, EQ)  MSG (Send, Recv, Inject, Tags)  RMA (Read/Write)  AV (Address Vectors)  MR (Memory Regions)

Dynamic Binding

Core messaging logic
(Create packet headers, frag data, hardware requirements, completion semantics)

Intranode (Lock free MPMC queues)  Reliability (Onload/Offload) (Bounce Buffer/Zero Copy)  Kernel Bypass (MMIO)  HFI1 Driver (ioctl, writev, /sys/class/ib)

RDMA  GPU
Many tuning knobs split across many levels

- Direct
  - Application tuning knobs and `#define`, `MPI_THREAD_MODEL`
  - MVAPICH ENVs (prefix `MV2_*`), `#define`, anything else?
  - Libfabric and OPX (prefix `FI_*`, `FI_OPX_*`), `#define`
  - Hfi1 device driver (`/sys/module/hfi1/parameters`)
  - Fabric Service Level, Virtual Lane, and FGAR

- Indirect
  - Linux kernel/distro tuning and de-jittering, hugepages
  - HPC job topology <-> Fabric topology
  - Compute Node BIOS settings (c-state, NUMA, PCIe, etc)
  - Package versions (Spack, Ansible, Docker, ...)
OPX Libfabric Endpoint config

• Locking vs Non-locking
  • `FI_THREAD_DOMAIN, FI_THREAD_ENDPOINT, FI_THREAD_COMPLETION` — most performant
  • OPX has high-level locking for thread safety.

• Manual vs Auto progress
  • Onload-type reliability makes all flows RX/TX. Polling for progress via `cq_read` allows OPX to do progress work. Better if MPI calls this (this is the manual progress model)
  • Auto progress works, will run threadsafe, and spawn an external thread to call `cq_read` to ‘progress’ the provider. Not as fast

• AV_TABLE vs AV_MAP
  • AV_MAP is more tested and faster (maybe). Libfabric2 might remove

• CQ_COMPLETION — When is an op ‘complete’?
  • `FI_INJECT_COMPLETE` vs `FI_TRANSMIT_COMPLETE` vs `FI_DELIVERY_COMPLETE`
  • `FI_SELECTIVE_COMPLETION`

• Onload vs Offload reliability
  • Onload model only supported right now. Offload model might be coming

• `FI_CONTEXT_2` — Only provider that requires this?
OPX ENVs

- Documentation in OPXS docs and Libfabric man page:  
  `fi_opx(7)` (ofiwg.github.io)
- Best-practice: Run `fi_info` with your target Libfabric to print the OPX provider ENVs: 
  `fi_info -g FI_OPX`
- `FI_OPX_HFI_SELECT` - How to deal with multiple HFIs and NUMA
- `FI_OPX_DELIVERY_COMPLETION_THRESHOLD` - Bounce buffers
- `FI_OPX_EXPECTED_RECEIVE_ENABLE` - RDMA
- `FI_OPX_RELIABILITY_SERVICE_USEC_MAX` - Reliability Ping
- `FI_OPX_RELIABILITY_SERVICE_PRE_ACK_RATE` - Send window
Hfi1 device driver has many tunable module parms

- List them all with this command (on a system that has OPXS installed and hfi1 loaded):
  - `grep . /sys/module/hfi1/parameters/*`

    - Many parms, psm2 tuning guide has advice on tuning these, and it works pretty well for OPX
    - `rcvhdr cnt`: Depth of the Eager Rx rings for each context. Default 2048. Recommend 8192 for larger HPC jobs. The larger value will make the memory footprint larger
    - `num_user_contexts`: Maximum number of ranks per hfi1 adapter in this compute node. Default is -1, which means assume max ranks. Setting this value to a number that is equal to the number of ranks you expect to launch (probably the number of cores?) MIGHT increase performance by allocating extra TX send buffer. The lower the number, the more credits.
OPX Observability - Logs

• `FI_LOG_LEVEL=WARN` or `TRACE` or `INFO` or `DEBUG` or `MAX`

• Optimized or Debug builds, these are valid for both. Optimized builds skip much logging

• OPX is a potential data firehose of text

```c
libfabric:34638:1680714411::opx::opx_hfi_wait_for_device():163::info:: Found /dev/hfi0 after 0.0 seconds
libfabric:34638:1680714411::opx::hfi_cmd_ioctl():352::info:: command: OPX_HIFI_CM_DX, HFI1_IOCTL 0x8006418f
libfabric:34638:1680714411::opx::hfi_cmd_ioctl():352::info:: command: OPX_HIFI_CM_DX, HFI1_IOCTL 0x80064181
libfabric:34638:1680714411::opx::hfi_cmd_ioctl():352::info:: command: OPX_HIFI_CM_DX, HFI1_IOCTL 0x80064182
libfabric:34638:1680714411::opx::hfi_userInit_internal():548::info:: CONTEXT INIT cxtinfo:: active1, unit 0, cxt 11, subcxt 0
libfabric:34638:1680714411::opx::hfi_userInit_internal():558::info:: CONTEXT INIT cxtinfo:: rcvtds 1400, credits 361
libfabric:34638:1680714411::opx::hfi_userInit_internal():562::info:: CONTEXT INIT cxtinfo:: numa 0, cpu 0, sendcxt 139
libfabric:34638:1680714411::opx::hfi_userInit_internal():562::info:: CONTEXT INIT cxtinfo:: rcvhdrcnt 284, rcvhdrcntsize 128
libfabric:34638:1680714411::opx::hfi_userInit_internal():556::info:: CONTEXT INIT cxtinfo:: tgtids 32, sdma_ring_size 128
libfabric:34638:1680714411::opx::hfi_userInit_internal():532::info:: CONTEXT INIT cxtinfo:: active1, unit 0, cxt 11, subcxt 0
libfabric:34638:1680714411::opx::hfi_userInit_internal():573::info:: CONTEXT INIT baselinfo:: hwver 303710, swver 60003, jkey 59371, gp 128
libfabric:34638:1680714411::opx::hfi_userInit_internal():573::info:: CONTEXT INIT baselinfo:: credit_addr dabb000000000008, sop dabb000000000008, pl0 dabb000000000008
libfabric:34638:1680714411::opx::hfi_userInit_internal():579::info:: CONTEXT INIT baselinfo:: hdpbase dabb000000000000, egpbase dabb000000000000, sdbase dabb000000000000
libfabric:34638:1680714411::opx::hfi_userInit_internal():579::info:: CONTEXT INIT baselinfo:: eventBase dabb000000000000, statbase dabb000000000000, tailaddr 0
libfabric:34638:1680714411::opx::hfi_userInit_internal():579::info:: CONTEXT INIT baselinfo:: eventBase dabb000000000000, statbase dabb000000000000, tailaddr 0
libfabric:34638:1680714411::opx::hfi_userInit_internal():579::info:: CONTEXT INIT baselinfo:: eventBase dabb000000000000, statbase dabb000000000000, tailaddr 0
```

© 2023 Cornelis Networks
OPX Observability – Debug Counters

- Requires debug build of Libfabric AND `#define OPX_DEBUG_COUNTERS`
- Will affecting timing/race conditions

```c
#define DEBUG_COUNTERS

// mp_eager_send_first_packets
// mp_eager_send_nth_packets
// mp_eager_send_first_force_cr
// mp_eager_send_nth_force_cr
// mp_eager_send_fall_back_to_rsv
// mp_eager_send_full_replay_buffer_rx_poll
// mp_eagerrecv_max_ue_queue_length
// mp_eagerrecv_max_sq_queue_length
// mp_eager_recv_first_packets
// mp_eager_recv_nth_packets
// mp_eager_recv_completed_process_context
// mp_eager_recv_completed_eager_first
// mp_eager_recv_completed_eager_nth
// mp_eager_recv truncation
// mp_eager_recv_nth_no_match
// mp_eager_recv nth_match
// mp_eager_recv_total_completed
// reliability_ping.acks_sent
// reliability_ping.acks_preamptive_sent
// reliability_ping.acks_received
// reliability_ping.acks ignored
// reliability_ping.nacks_sent
// reliability_ping.acks_preamptive_sent
// reliability_ping.nacks_received
// reliability_ping.nacks ignored
// reliability_ping.pings_sent
// reliability_ping.pings received
```
Current Status

• MVAPICH2 numbers look performant, good out of box experience
• OPX Code is upstream, recommend v1.19 of Libfabric.
• New features
  • Large message improvements – upstream but not default on (update hfi1 driver first)
  • GPU support – Almost up streamed
• Available on GitHub, Distro*, OPXS Software Suite
  • Checkout Libfabric ‘main’ branch
• Get involved
  • Happy to take patches via GitHub
Thank You

www.cornelisnetworks.com