OPENFABRICS ALLIANCE

2023 OFA Virtual Workshop

Libfabric OPX Provider

Tim Thompson, Senior Software Engineer (Libfabric)

Cornelis Networks

CORNELIS NETWORKS

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, Cornelis Networks, 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.

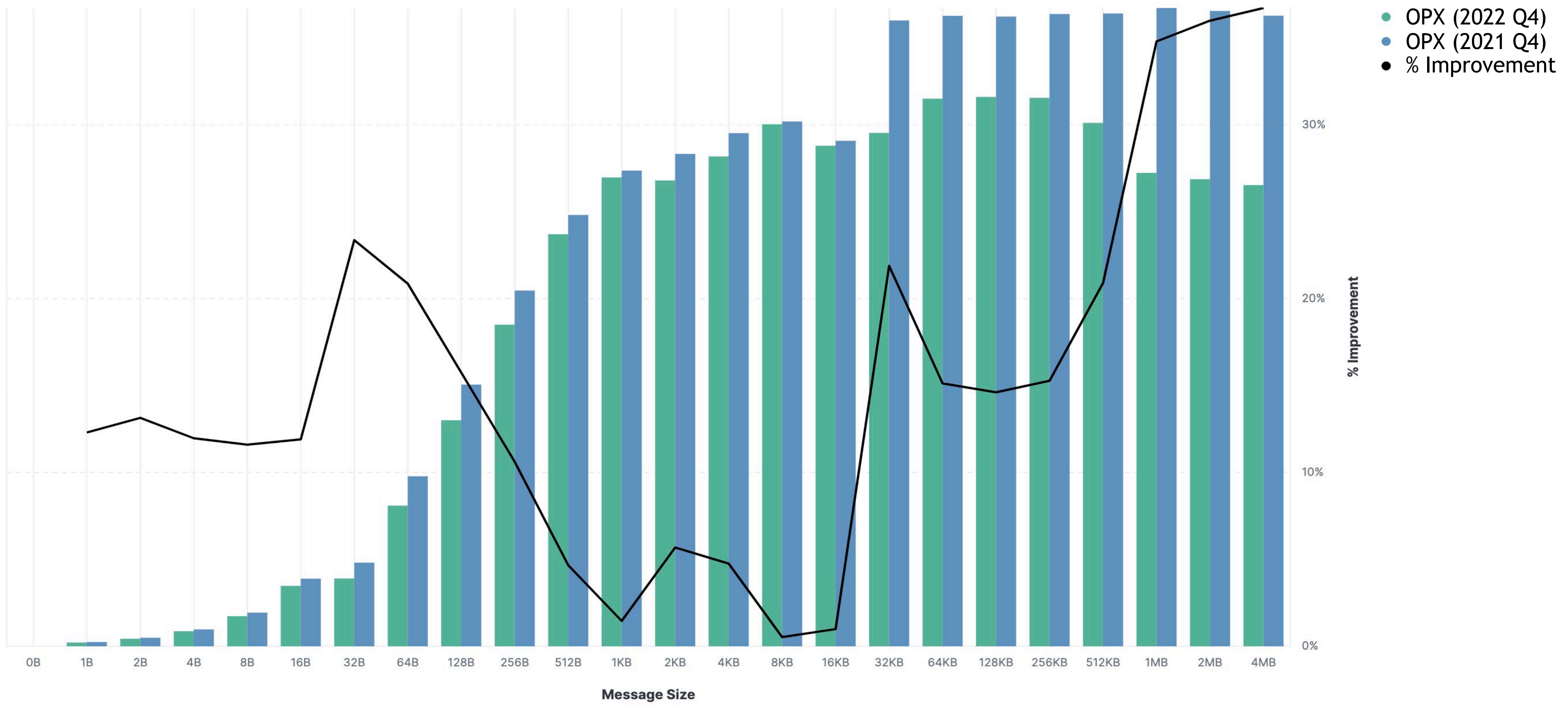
Introduction

- Who is Cornelis Networks?
  - Omni-Path Architecture (OPA)
  - Third year of talks in this workshop
  - Spun out of Intel
- What is OPX?
  - Labfabric provider for Cornelis and Omni-Path fabrics
  - User-space part of hfi1 device driver/hardware interfaces
  - Started as a clone of the BGQ provider
  - Supports 100Gbps and 400Gbps (upcoming) fabrics
- Who am I?
  - User-space Senior Software Engineer Cornelis Networks

Past Year's Progress

- Bulk Transfer Tx (offloads Tx PIO interface)
- Additional feature: Auto progress
- Additional feature: AV_TABLE
- Enhanced feature: Tag matching at scale
- DAOS progress (coming in version 2.4)
- Reliability enhancements for HPC apps at scale
- One-sided MPI and OpenSHMEM support
- Observability
- Performance has increased for vast majority of workloads and message sizes

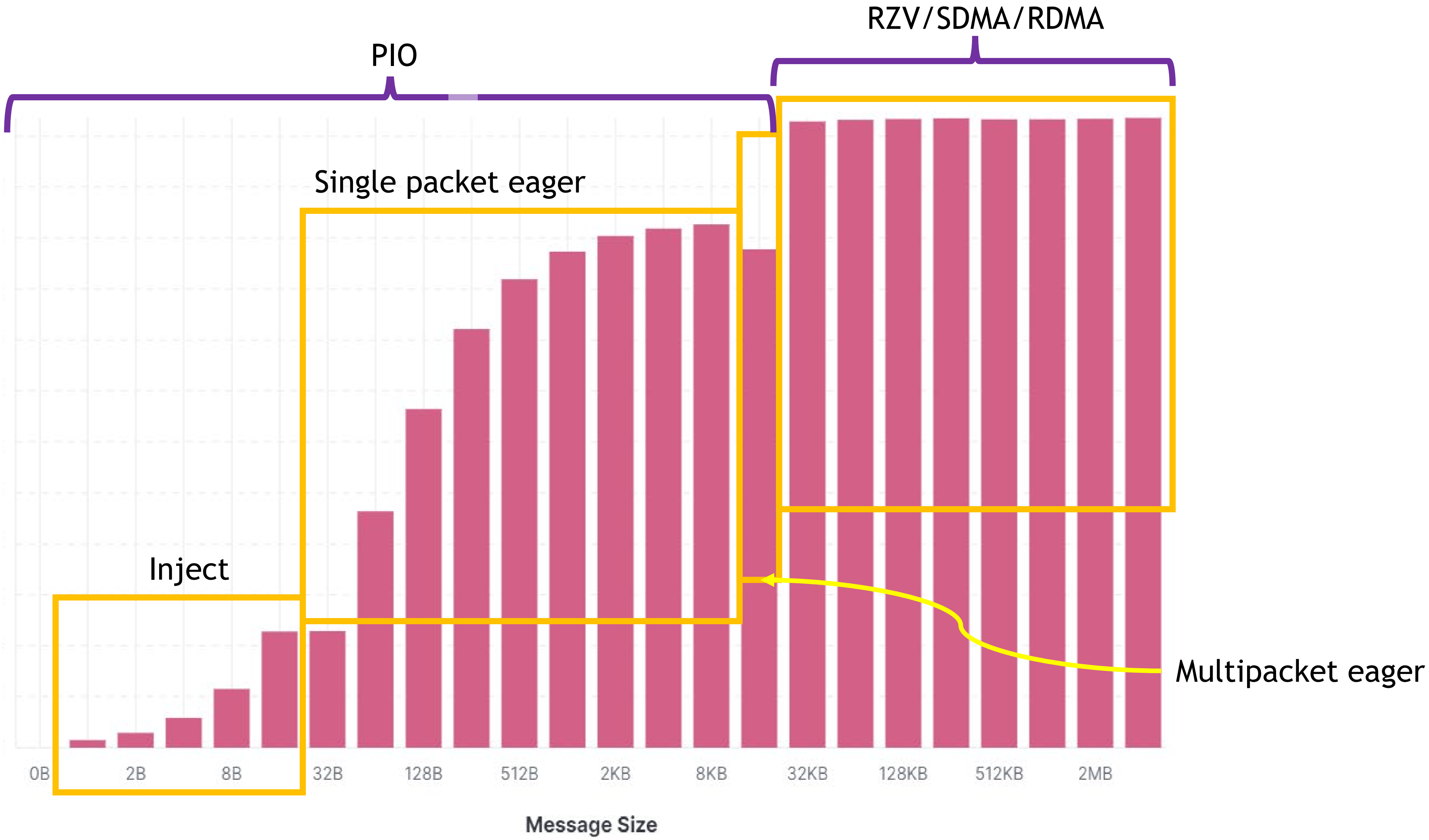
2022 Performance improvement



In Progress

- RDMA - Expected Receive
  - Eliminates Rx bounce buffers and offloads Rx Eager Ring
  - Requires hardware mapping/pinning of HPC application memory pages
  - Extra overhead makes this non-performant for ‘small’ messages
  - Buffer fragmentation adds to headaches
  - HPC app behavior affects performance (re-map buffers vs re-use)
- DAOS support
  - Internode and Intranode durable connection resume support
  - Eventually will need Scalable Endpoints
  - Alternative to IB_verbs for HPC storage
- GPU support
  - GPUs have their own non-standard APIs, need to use these for best performance
  - Testing base support for CUDA, intranode and internode traffic
  - Avoiding vendor lock-in creates more testing and more design requirements
- CN5000 – 400 Gbps adapters, more fabric features for scalability
- Programmer creature comforts: Observability and Debuggability

Message length thresholds



Upstream First

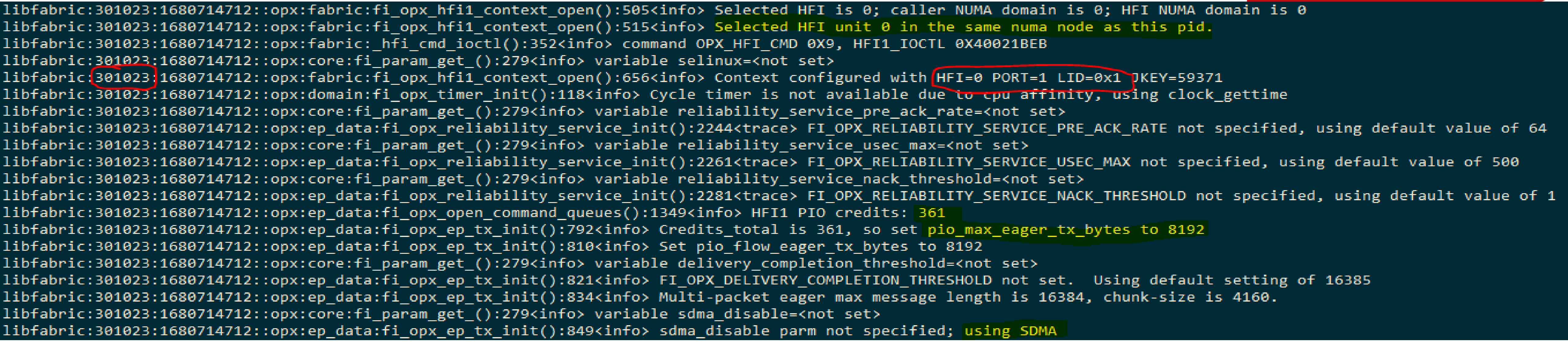
- Similar points Dennis mentioned for the hfi1 device driver
- User space HPC community easier to work with than Linux Kernel
- Testing before upstream, SO MANY VARIATIONS, CI not 100% coverage
- There’s also test case and debug code that is #ifdef out
- Plan Cornelis software releases around upstream project releases whenever possible
- Official support is still Cornelis Networks software releases of ‘OPXS’ (used to be called IFS)
- Libfabric CI on upstream PRs

Extended testing

- How does a developer know if their changes affected performance?
    - Microbenchmarks don’t tell the whole story
    - Testing at scale, what is ‘big’ in HPC?
- Developers need tooling, emulators, and hardware
    - Hardware counters, processor, PCIe, and hfi1 driver
    - Tools like Intel SDE and vTune
    - Instrumented testing with asserts, debug-builds, and testing code
    - Hard to automate this type of testing
- Amount of extended testing is limited by Developer’s time

Observability

- Means that anyone (a dev or a user) can see granular details about how the hardware under their job is configured
- With HPC performance constraints, a re-compile of Libfabric/Opx may be needed for extra logs and counters. Debug builds are prohibitively verbose currently.
- User can enable/configure more logging with #define and re-compile…
- Set ENV FI_LOG_LEVEL=info with any build of Libfabric to see SOME things, especially useful on HPC job startup.



```
libfabric:301023:1680714712::opx:fabric:fi_opx_hfi1_context_open():505<info> Selected HFI is 0; caller NUMA domain is 0; HFI NUMA domain is 0
libfabric:301023:1680714712::opx:fabric:fi_opx_hfi1_context_open():515<info> Selected HFI unit 0 in the same numa node as this pid.
libfabric:301023:1680714712::opx:fabric:_hfi_cmd_ioctl():352<info> command OPX_HFI_CMD 0X9, HFI1_IOCTL 0X40021BEB
libfabric:301023:1680714712::opx:core:fi_param_get_():279<info> variable selinux=<not set>
libfabric:301023:1680714712::opx:fabric:fi_opx_hfi1_context_open():656<info> Context configured with HFI=0 PORT=1 LID=0x1 PKEY=59371
libfabric:301023:1680714712::opx:domain:fi_opx_timer_init():118<info> Cycle timer is not available due to cpu affinity, using clock_gettime
libfabric:301023:1680714712::opx:core:fi_param_get_():279<info> variable reliability_service_pre_ack_rate=<not set>
libfabric:301023:1680714712::opx:ep_data:fi_opx_reliability_service_init():2244<trace> FI_OPX_RELIABILITY_SERVICE_PRE_ACK_RATE not specified, using default value of 64
libfabric:301023:1680714712::opx:core:fi_param_get_():279<info> variable reliability_service_usec_max=<not set>
libfabric:301023:1680714712::opx:ep_data:fi_opx_reliability_service_init():2261<trace> FI_OPX_RELIABILITY_SERVICE_USEC_MAX not specified, using default value of 500
libfabric:301023:1680714712::opx:core:fi_param_get_():279<info> variable reliability_service_nack_threshold=<not set>
libfabric:301023:1680714712::opx:ep_data:fi_opx_reliability_service_init():2281<trace> FI_OPX_RELIABILITY_SERVICE_NACK_THRESHOLD not specified, using default value of 1
libfabric:301023:1680714712::opx:ep_data:fi_opx_open_command_queues():1349<info> HFI1 PIO credits: 361
libfabric:301023:1680714712::opx:ep_data:fi_opx_ep_tx_init():792<info> Credits_total is 361, so set pio_max_eager_tx_bytes to 8192
libfabric:301023:1680714712::opx:ep_data:fi_opx_ep_tx_init():810<info> Set pio_flow_eager_tx_bytes to 8192
libfabric:301023:1680714712::opx:core:fi_param_get_():279<info> variable delivery_completion_threshold=<not set>
libfabric:301023:1680714712::opx:ep_data:fi_opx_ep_tx_init():821<info> FI_OPX_DELIVERY_COMPLETION_THRESHOLD not set.  Using default setting of 16385
libfabric:301023:1680714712::opx:ep_data:fi_opx_ep_tx_init():834<info> Multi-packet eager max message length is 16384, chunk-size is 4160.
libfabric:301023:1680714712::opx:core:fi_param_get_():279<info> variable sdma_disable=<not set>
libfabric:301023:1680714712::opx:ep_data:fi_opx_ep_tx_init():849<info> sdma_disable parm not specified; using SDMA
```

Debuggability

- OPX observability and debug logs COULD sometimes help users debug their own code…but probably not much help

- Users need help/hints about their own bugs like hangs and performance issues

- OPX can provide counters and 'current status', like how many unmatched messages are sitting in the match queues

- **What information from the provider/fabric do users want to help their own debug?**

Thank You

www.cornelisnetworks.com

CORNELIS™ NETWORKS