15th ANNUAL WORKSHOP 2019

EXPERIENCES WITH LIBFABRIC

Harold E. Cook, Director of Engineering

Lightfleet Corporation

March 20, 2019

BACKGROUND

- **Open Fabrics Alliance developed and supports the libfabric interface to provide a high-performance, scalable, application centric, extensible interface for the OFI stack that has as a goal to be hardware agnostic.**
  - For more information see: https://ofiwg.github.io/libfabric/

- **By no means is this presentation:**
  - a condemnation of libfabric
  - its developers
  - its design.
- **Rather, it is to share experiences with the intent of improving libfabric.**

- **In fact, this community should be very grateful for the time and effort that has gone into the development of libfabric**
  - Thank you to OFIWG and the contributors!

LIBFABRIC OBJECTIVES

- **From the 1.7.0 README file:**

* High-performance: provide optimized software paths to hardware
  - Independent of hardware implementations
* Scalable: targets support for millions of processes
  - Designed to reduce cache and memory footprint
  - Scalable address resolution and storage
  - Tight data structures
* Application-centric
  - Interfaces co-designed with application developers and hardware vendors
* Extensible
  - Easily adaptable to support future application needs

PERSPECTIVE OF THIS PRESENTATION

- **Lightfleet is a hardware vendor bringing low latency, high throughput interconnects that deliver:**
  - True multicast with zero jitter and skew
  - User space to user space transfers (RDMA)
  - Zero lost packets
  - Determinism
  - Hardware packet routing without software overhead
- **Our focus is on API level network abstractions of which there are many**
  - Both open source as well as commercial
- **Our libfabric effort is about 3 months along and we do not yet have a releasable provider**
  - Cannot yet comment on the support or integration issues, may be a topic for the next years workshop?
- **Yes, we will contribute to the libfabric effort as we are able.**

WITH ALL THIS IN MIND, OUR EXPERIENCES THUS FAR…

OVERALL IMPRESSIONS

- **libfabric exhibits hallmarks typical of Open Source development projects:**

  - There is Documentation but…
    - We have found it to be inadequate for development of a provider
    - What is there is not always clear or is inaccurate and in some cases very “dated”
      - Ex: differentiation between “domain” and “fabric” is not clear to us at this point..
  - Roadmap is not always clear
    - Ex: our first approach was sockets based, only to discover in an e-mail conversation with Sean H that socket support is being deprecated.
  - Support is primarily available by github, mail reflectors or interaction with contributors. Also OFIWG attendance
  - Design is predominantly point-to-point with support for a reliable multipoint datagram protocol
    - Multicast support is TBD
  - Code is not always clear and requires reverse engineering
    - Example later.

WHAT WE ARE IMPRESSED WITH THUS FAR…

- **The design:**
  - Support for multiple interfaces and subnets.
  - In some cases, if a feature is requested by the application and it is not supported by the provider, a layer is added that emulates the support in software.
  - Verification tools (aka fabtests) included in the releases
    - But… (more later)
  - Support for various address schemes – part of the network agnosticism

FABTESTS

- **Good tools for initial verification**

- **Caution is necessary:**
  - Appears that some of the tests report passing conditions when in reality they did nothing because the necessary support from the provider was not present

  - Appear to rely primarily on TCP/IP addressing
    - fabtests need to be modified for other addressing schemes?

CODE/DOCUMENTATION ISSUE EXAMPLE

fabric.h:

```
struct fi_tx_attr {
    uint64_t caps;
    uint64_t mode;
    uint64_t op_flags;
    uint64_t msg_order;
    uint64_t comp_order;
    size_t inject_size;
    size_t size;
    size_t iov_limit;
    size_t rma_iov_limit;
}
```

- **What does size specify?**
- **From fi_endpoint man page:**
  - The size of the context. The size is specified as the minimum number of transmit operations that may be posted to the endpoint without the operation returning -FI_EAGAIN.
- **However…**
  - We found the value of size is silently raised to a power of two, so it isn't really the minimum
    - The power of 2 adjustment is not described in the documentation
  - The term “context” appears to be used in different ways in other locations in the documentation
    - Could lead to confusion

AND SO IT GOES…

- **We are working to get basic point-to-point functionality**
- **Anticipated challenges:**
  - True multicast support
    - Area where we will likely contribute in the future
  - Integration and Support issues

15th ANNUAL WORKSHOP 2019

THANK YOU

Harold E. Cook, Director of Engineering

Lightfleet Corporation

hcook@Lightfleet.com