GNI Provider Iovector Support for Libfabric

Evan Harvey
New Mexico Institute of Mining and Technology
916 Annette Ave. Socorro, NM 87801
10304 Dunbar St. NW Albuquerque, NM 87114
+1 (505) - 948 - 0697
evan.harvey@student.nmt.edu

ABSTRACT
Portable HPC middleware such as MPICH has typically targeted low-level network APIs that are vendor specific. Libfabric is a new portable, low-level API that aims to alleviate the burden of porting to new network APIs without sacrificing performance. The Libfabric GNI "provider" was developed to enable testing Libfabric-based middle-ware at scale using Cray XC(TM) systems with the Aries interconnect. This paper will describe the implementation of scatter-gather lists for point-to-point data transfers using the Aries Chip. [Choi, Tiffany personal communications]

CCS Concepts
• Network architectures→Programming interfaces • Software and its engineering→Contextual software domains→Software infrastructure→Middleware→Message oriented middleware.

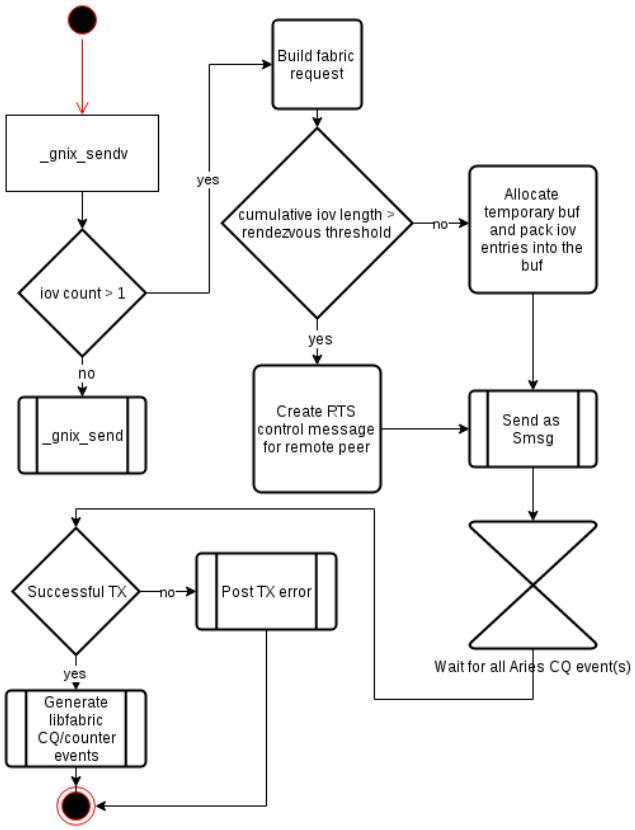
Keywords
Architecture and Networks; System Software

1.INTRODUCTION
Portable HPC middle-ware such as MPICH has typically targeted low-level network APIs that are vendor specific. Consequently, when a new network comes to market, MPICH developers must port the device specific "netmod" to the new interface. Libfabric is a new portable, low-level interface that aims to alleviate this burden without sacrificing performance. [Choi personal communications]

2.LIBFABRIC
Libfabric is a library that provides a network API which may span many networks. What distinguishes Libfabric from other portable network APIs is the 'fi_getinfo' function, a way to obtain information about available network services at runtime, making it much more flexible than current APIs. In this way, Libfabric provides a way for application programmers to port, develop, and test code using the Libfabric API for new systems on current generation systems. Libfabric developers implement "providers" that target the actual hardware. [1] [Choi, Tiffany personal communications]

3.MPICH
The next generation MPICH ch4 device targeting the Libfabric API is currently under development. MPICH sits on top of the Libfabric API and the ch4 "netmod" will be used by Intel, Cray and others to target next generation networks. Since any Libfabric provider may not fully support the Libfabric API on a given platform, there is a general active-message version of all the MPICH functions; MPICH active-message functions require "scatter-gather" lists for point-to-point data transfer operations in order to augment the provider's missing API functionality. Therefore, to enable testing of MPICH ch4 at scale using today's Aries-based systems, the GNI provider should support an iovector limit of at least 2. [2] [Choi personal communications]

4.GNI PROVIDER
The Libfabric GNI provider for the Aries interconnect uses the user-level Generic Network Interface (uGNI) API. The Aries Network Interface Card (NIC) is one of the few interfaces that supports true one-sided operations. The Aries interconnect is currently available in all Cray XC(TM) systems including large DOE installations such as NERSC Cori. As such, the GNI provider is an ideal vehicle for testing HPC middle-ware on existing systems at scale, to prepare for tomorrow's systems. [Choi personal communications]



5.GNI PROVIDER IOVECTOR SUPPORT

uGNI performance results for data transfers using the FMA and RDMA routines show that chain lengths greater than one and individual messages 4K or less in size perform better using chained FMA GETs rather than RDMA GETs. Additionally, the overhead associated with copying the sender's iovector entires into temporary storage for transfers sizes less than 16kbytes was determined to be less than the overhead associated with the rendezvous protocol. Consequently, the approach shown in Figure 1 and 2 was taken. [3] [Tiffany, Zinger personal communications]

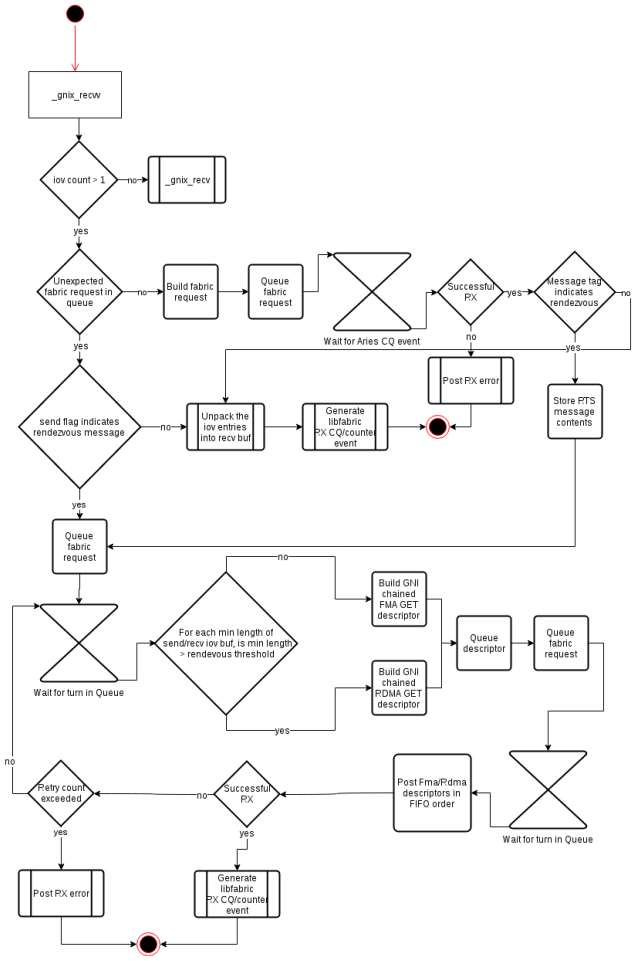


Figure 2.

5.1.RESULTS

The Aries NIC has a theoretical user data bandwidth of ~10GB/s. This implementation utilized the Aries RDMA, FMA, and SMSG mechanisms on two Broadwell nodes as follows. The RDMA mechanism realized bandwidths of ~7.50 GB/s to ~9.85 GB/s for iovector entries summing to 16KB to 4MB in length, showing the highest bandwidth of ~9.85 GB/s with an iovector count of 6 and an iovector entry length of 128KB using 'fi_sendv', 'fi_recv' and 'fi_sendv', 'fi_recvv'. The Aries FMA mechanism did not perform as efficiently as RDMA, producing bandwidth of ~1.0 GB/s to ~1.37 GB/s for iovector entry lengths summing to 2KB to 8KB. The Aries SMSG messages approximately doubled bandwidth for 'fi_sendv', 'fi_recvv' in comparison to iteratively calling 'fi_send', 'fi_recv', resulting in bandwidth of ~2.30 MB/s to ~1.5 GB/s for iovector entry lengths summing to 1B to 2KB. SMSG message rates increased by ~750% for messages less than 1KB in size, FMA message rates decreased by ~60%, and RDMA message rates increased by ~50% for 'fi_sendv'/'fi_recvv' in comparison to calling 'fi_send'/'fi_recv' iterativley for each iovector entry. The latency for all three mechanisms remains between ~1.75 microseconds to ~470 microseconds for messages of 8B to 4MB in size, respectively. Note that the data collected for individual 'fi_send' and 'fi_recv' calls does not include the overhead associated with the additional indexing information and memory required by the application in order to guarantee the arrival order of the iovector entries as defined by the sender.

Fi

6.CONCLUSION AND FUTURE WORK

These results show that a portable network API implementation such as Libfabric does deliver good network performance and that good performance can be obtained from the 'fi_sendv' and 'fi_recvv' interfaces which do not map directly to a hardware feature. The significance of this API addition to the Libfabric GNI provider will not be realized until MPICH ch4 has been developed with the Libfabric API and tested on Aries-based systems. Future work includes profiling the “scatter-gather” code paths in the GNI provider. Static code inspection and the results listed above suggest that increasing the rendezvous threshold may improve message rate and throughput, particularly for the chained FMA mechanism. Additionally, preallocating FMA GET descriptors should reduce the amount of time spent preparing to pull the data to the remote peer. Preallocating buffers that are as large as the rendezvous threshold should further improve the SMSG message rates and bandwidth by reducing the time spent dynamically allocating temporary storage for “scatter-gather” SMSG transfers.

7.ACKNOWLEDGMENTS

Special thanks to Howard Pritchard for all of his guidance and assistance with this project. I would also like to thank my mentors at Cray: Sung-Eun Choi, Zachary Tiffany, and James Swaro for all of their help.

8.REFERENCES

1. Group, OpenFabrics. "Libfabric". Ofiwg.github.io. N.p., 2016. Web. 3 Oct. 2016.
2. "AMMPI - Active Messages Over MPI". Gasnet.lbl.gov. N.p., 2016. Web. 3 Oct. 2016.
3. United States. Cray Inc. “Using the GNI and DMAPP APIs”. 2011. Print.