OPENFABRICS ALLIANCE

12th ANNUAL WORKSHOP 2016

STATUS OF OFI IN MPICH

Ken Raffenetti, Software Development Specialist

Argonne National Laboratory

[April 5th, 2016]

Argonne NATIONAL LABORATORY

OUTLINE

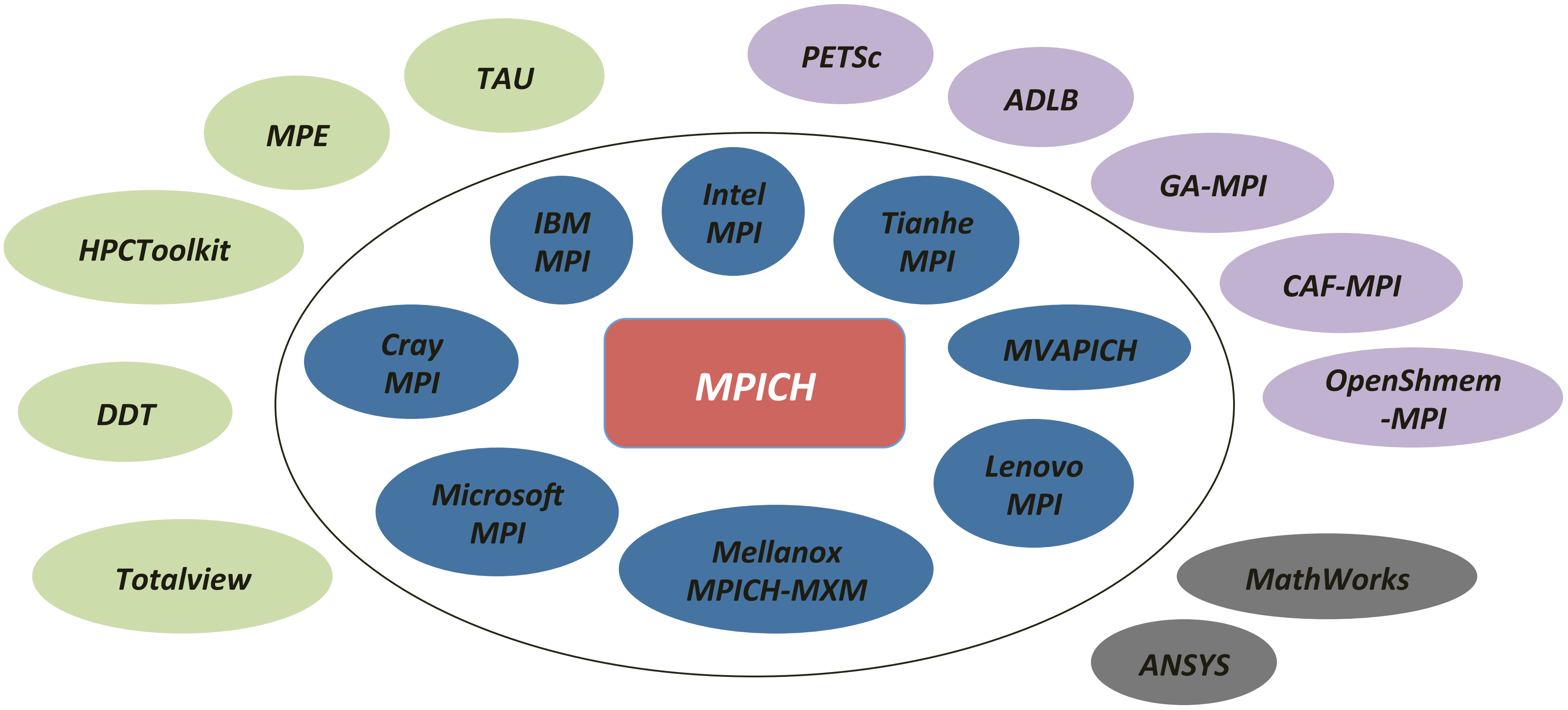
- **What is MPICH?**
- **Why OFI?**
- **OFI support in MPICH-3.2 (stable)**
- **MPICH-3.3**
  - Design focus
  - Development update
  - Roadmap
- **Conclusions**

WHAT IS MPICH?

- **MPICH is a high-performance and widely portable open-source implementation of MPI**
- **It provides all features of MPI that have been defined so far (up to and include MPI-3.1)**
- **Active development lead by Argonne National Laboratory and University of Illinois at Urbana-Champaign**
  - Several close collaborators who contribute features, bug fixes, testing for quality assurance, etc.
    - IBM, Microsoft, Cray, Intel, Ohio State University, Queen's University, Mellanox, RIKEN AICS and others
- **Current stable release is MPICH-3.2**
- **www.mpich.org**

MPICH: GOALS AND PHILOSOPHY

- MPICH aims to be the preferred MPI implementation on the top machines in the world
- Our philosophy is to create an "MPICH Ecosystem"



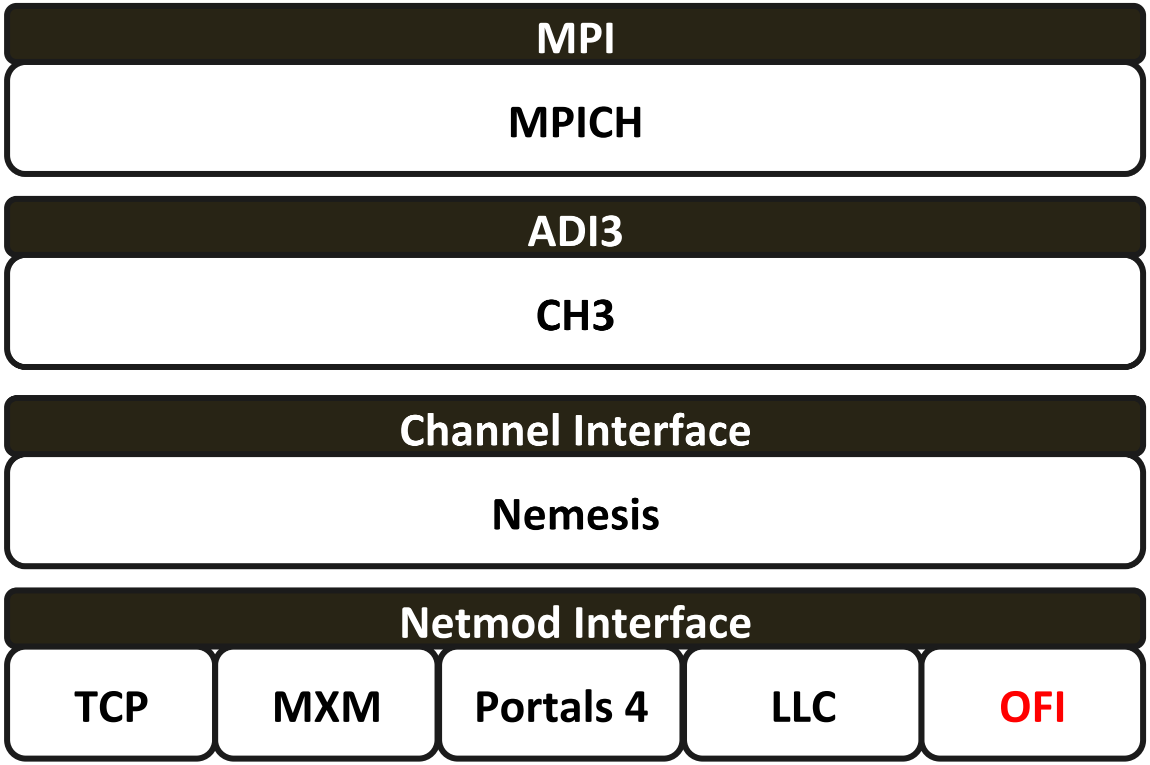
MOTIVATION

- **Why OFI/OFIWG?**
  - Support for diverse hardware through a common API
  - Actively, openly developed
    - Bi-weekly calls
    - Hosted on Github
  - Close abstraction for MPI
    - Less nitty-gritty network code
  - Fully functional sockets provider
    - Prototype code on a laptop

MPICH-3.2

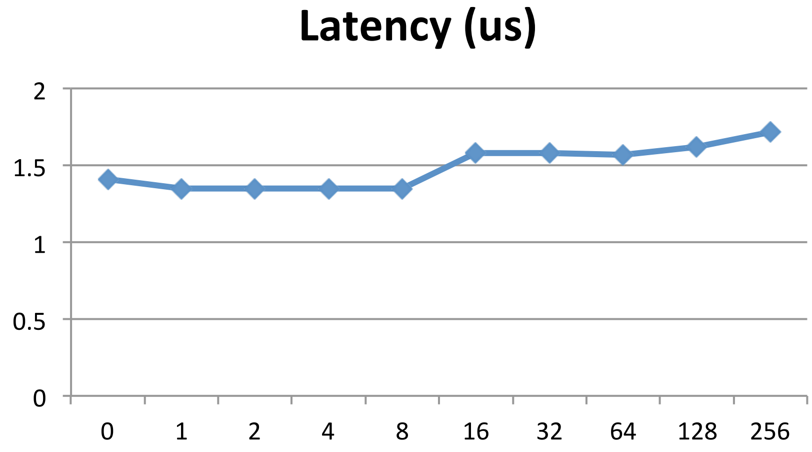
- **MPICH-3.2 is the latest major release series of MPICH**
  - Released MPICH-3.2 November 2015
- **Primary focus areas for MPICH-3.2**
  - Support for MPI-3.1 functionality (nonblocking collective I/O and others)
  - Fortran 2008 bindings
  - Support for the Mellanox MXM interface (thanks to Mellanox)
  - Support for the Mellanox HCOLL interface (thanks to Mellanox)
  - Support for the LLC interface for IB and Tofu (thanks to RIKEN)
  - Support for the OFI interface (thanks to Intel)
  - Improvements to MPICH/Portals 4
  - MPI-4 Fault Tolerance (ULFM)
  - Major improvements to the RMA infrastructure

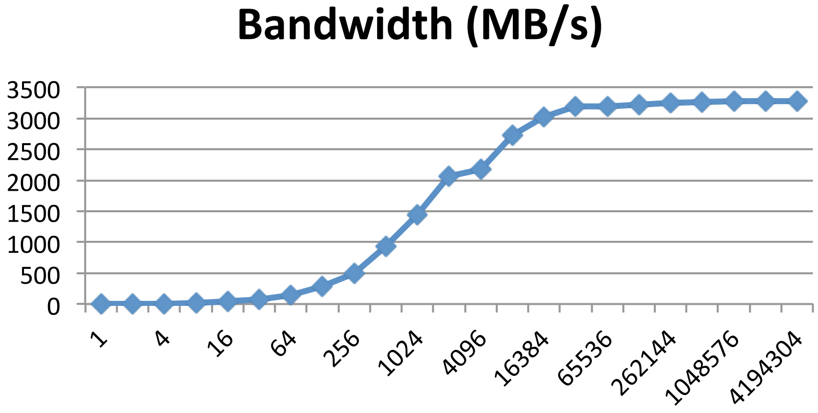
MPICH-3.2


MPI
MPICH
ADI3
CH3
Channel Interface
Nemesis
Netmod Interface
TCP
MXM
Portals 4
LLC
OFI

OFI NETMOD

- **CH3 netmod**
  - Send/Recv over `fi_tagged` interface
  - Control messages and RMA over `fi_msg`
- **Test machines**
  - QLogic QDR Infiniband
  - Infinipath PSM 3.3





OFI NETMOD

- **Where to improve?**
  - MPI RMA with `fi_rma, fi_atomic`
  - Collectives with `fi_trigger`
  - Would require major infrastructure changes to CH3
    - Step back and look at CH3 as a whole…

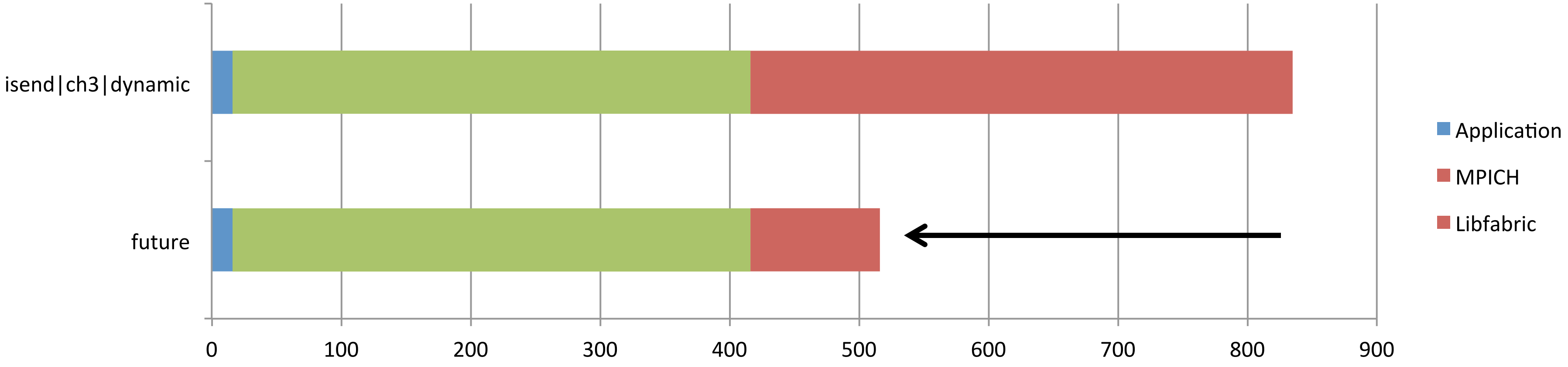
MPI ON OFI

- **Point-to-point data movement**
  - Closely maps to fi_tsend/trecv functionality

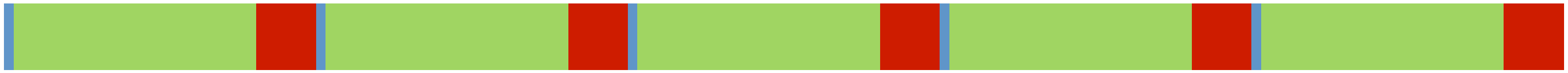
```
MPI_Isend(buf, count, datatype, dest, tag, comm, &req)

fi_tsend(gl_data.endpoint,       /* Local endpoint */
         send_buffer,            /* Packed or user */
         data_sz,                /* Size of the send */
         gl_data.mr,             /* Dynamic memory region */
         dest_addr,              /* Destination fabric address */
         match_bits,             /* Match bits */
         context);               /* Context */
```

OFI NETMOD

- **With MPI features baked into next-generation hardware, we anticipate network library overheads will dramatically reduce.**



- Message rate will come to be dominated by MPICH overheads



CH3 SHORTCOMINGS

Netmod API

- Passes down limited information and functionality to the network layer
  - `SendContig`
  - `SendNoncontig`
  - `iSendContig`
  - `iStartContigMsg`
  - `...`

Active Message Design

- **All communication involves a packet header + message payload**
  - Requires a non-contiguous memory access for all messages
- **Workaround for Send/Recv override exists, but was somewhat clunky add-in**

Singular Shared Memory Support

- **Performant shared memory communication centrally managed by Nemesis**
- **Network library shared memory implementations are not well supported**
  - Inhibits collective offload

Function Pointers Not Optimized By Compiler

```
if (vc->comm_ops && vc->comm_ops->isend){
     mpi_errno =
          vc->comm_ops->isend(vc, buf, count, ...)
     goto fn_exit;
}
```

Non-scalable "Virtual Connections"

- **480 bytes * 1 million procs = 480MB(!) of VCs per process**
- **Connection-less networks emerging**
  - VC and associated fields are overkill

MPICH-3.3 – CH4 DEVICE

- **Introducing the CH4 device**
  - Replacement for CH3, but we will maintain CH3 till all of our partners have moved to CH4
  - Co-design effort
    - Weekly telecons with partners to discuss design and development issues
  - Two primary objectives:
    - Low-instruction count communication
      - Ability to support high-level network APIs (OFI, UCX, Portals 4)
      - E.g., tag-matching in hardware, direct PUT/GET communication
    - Support for very high thread concurrency
      - Improvements to message rates in highly threaded environments (MPI_THREAD_MULTIPLE)
      - Support for multiple network endpoints (THREAD_MULTIPLE or not)

CH4 DESIGN GOALS

High-Level Netmod API

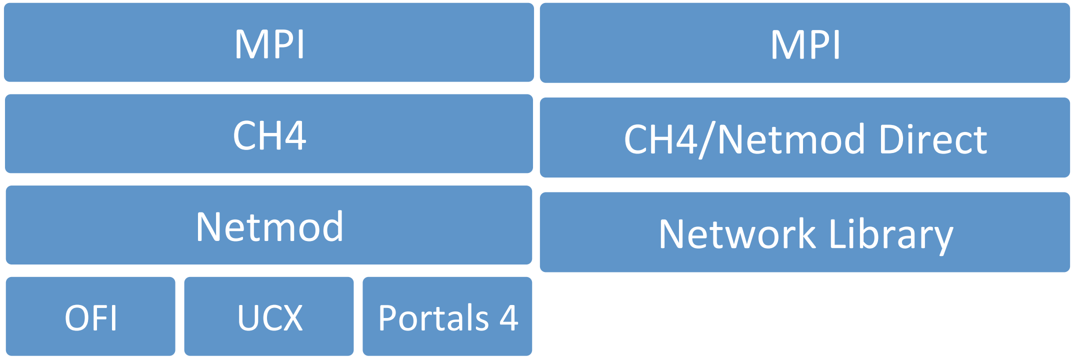
- Give more control to the network
  - `netmod_send`
  - `netmod_recv`
  - `netmod_put`
  - `netmod_get`
- Fallback to Active Message based communication when necessary
  - Operations not supported by the network

Configurable shared memory communication in CH4

- **Let the netmod decide**
  - Enable better tuned shared memory implementations
  - Collective offload

"Netmod Direct"

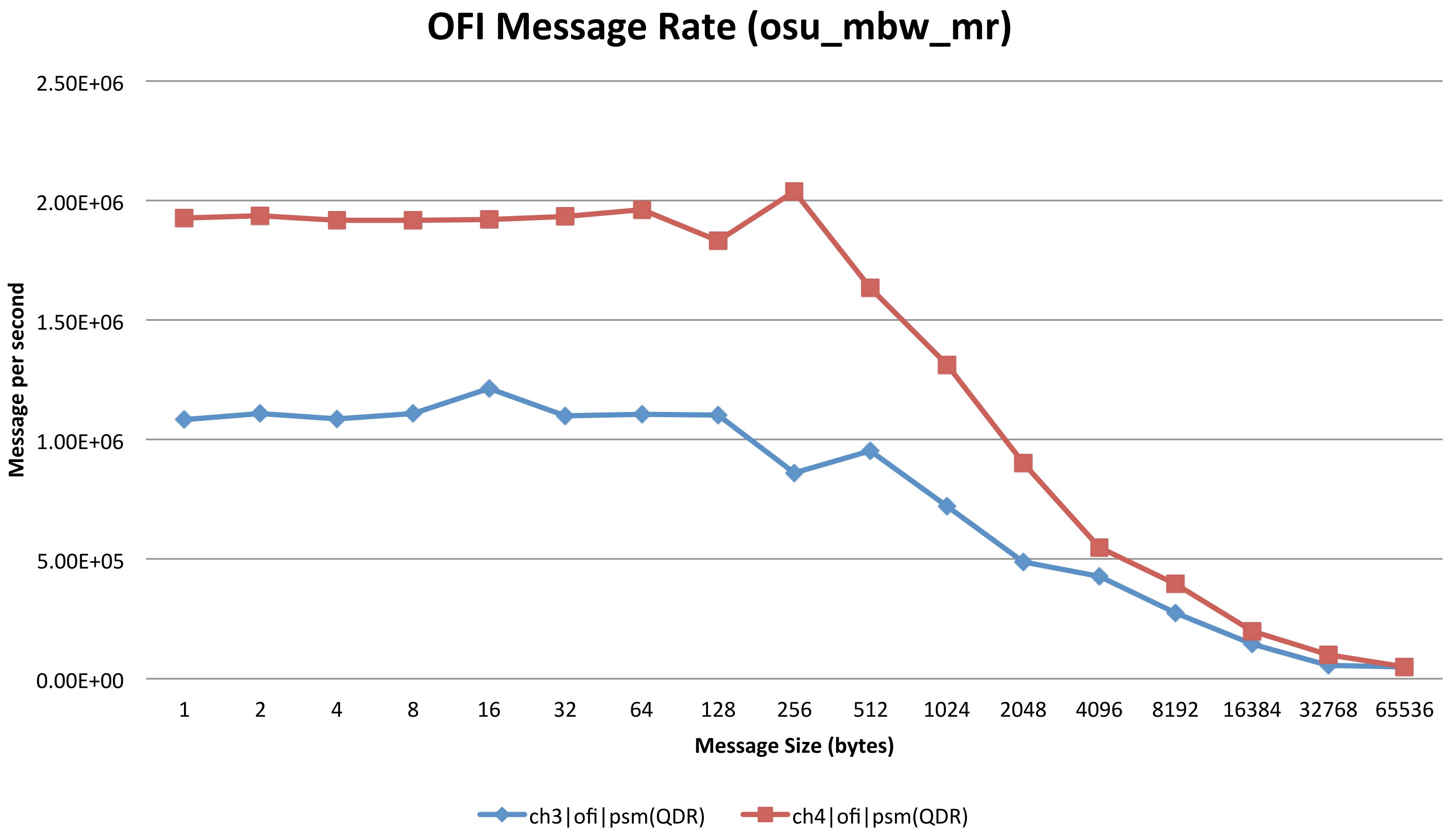
- Support two modes
  - Multiple netmods
    - Retains function pointer for flexibility
  - Single netmod with inlining into device layer
    - No function pointer



No Device Virtual Connections

- Global address table
  - Contains all process addresses
  - Index into global table by translating `(rank+comm)`

PRELIMINARY IMPROVEMENTS



MPICH-3.3 ROADMAP

- **CH4 code at http://git.mpich.org/mpich-dev**
  - Will land in main MPICH repo soon
- **MPICH-3.3a1 release out this spring**
  - Subsequent preview releases over the coming months
- **GA Release mid-2017**
- **Remaining work for OFI**
  - `fi_trigger` for collectives
  - Support for different OFI “capability sets”
  - Threading strategy
- **What about TCP?**
  - Leverage OFI sockets provider
  - Provide integration testing for FreeBSD and Solaris platforms

CONCLUSIONS

- **OFI will be well supported in MPICH**
  - CH3 support available now
    - MPICH-3.2.1 bugfix release this spring
  - CH4 available 2017
- **High-level APIs are driving us to:**
  - Reduce overheads
  - Give more control to the network layer

OPENFABRICS ALLIANCE

12th ANNUAL WORKSHOP 2016

THANK YOU

Ken Raffenetti, Software Development Specialist

Argonne National Laboratory

Argonne NATIONAL LABORATORY