UCF 2021

Argonne NATIONAL LABORATORY

MPICH + UCX: STATE OF THE UNION

KEN RAFFENETTI
Principal Software Development Specialist
Mathematics and Computer Science Division
Argonne National Laboratory
Email: raffenet@anl.gov

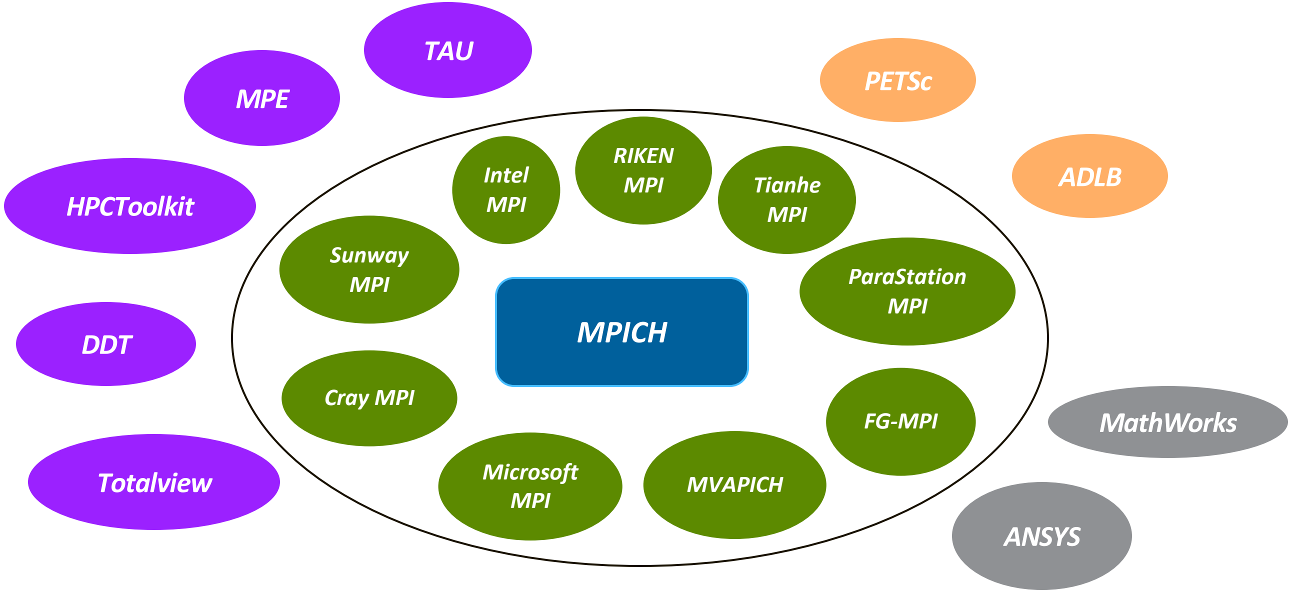
U.S. DEPARTMENT OF ENERGY

Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.

November 30, 2021

MPICH: GOALS AND PHILOSOPHY

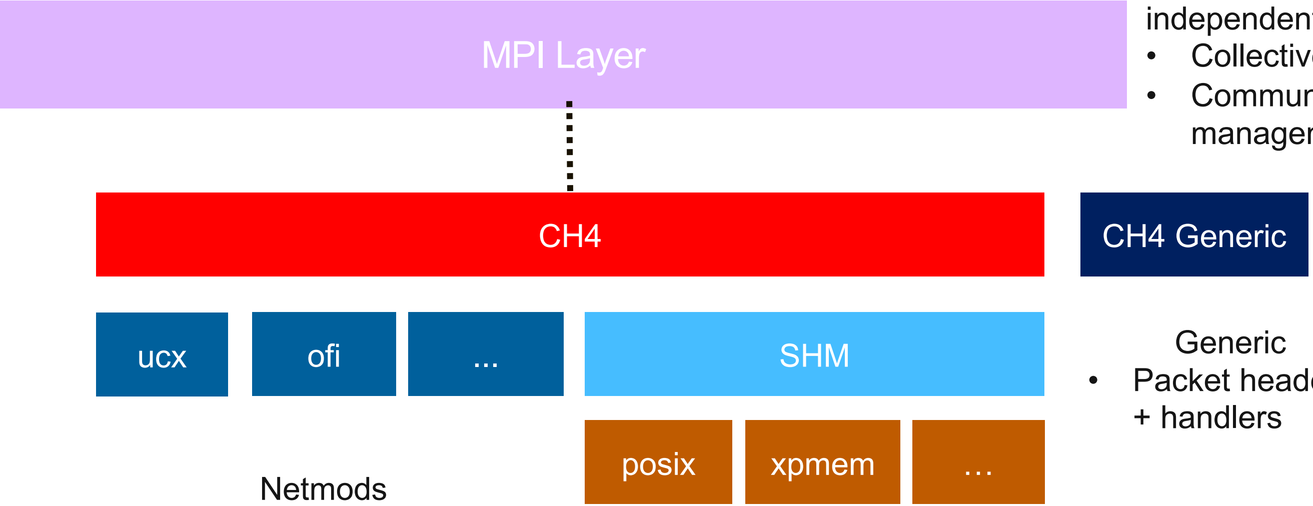
- MPICH continues to aim to be the preferred MPI implementations on the top machines in the world
- Our philosophy is to create an “MPICH Ecosystem”



AGENDA

- UCX Support in MPICH
- Request Handling
- Active Messages
- Multi-VCI
- Future Development
- Other Odds and Ends

MPICH LAYERED STRUCTURE: CH4



MPI Layer

CH4

CH4 Generic

ucx

ofi

...

SHM

posix

xpmem

…

Platform independent code

- Collectives
- Communicator management

Generic

- Packet headers + handlers

Netmods

- Provide all functionality either natively, or by calling back to “generic” implementation

Shmmods

- Implements some mandatory functionality
- Can override any amount of optional functionality (e.g. better bcast, better barrier)

UCX SUPPORT IN MPICH

- UCX “Netmod” Development
  - Argonne MPICH Team
  - Mellanox/NVIDIA
- MPICH 4.0b1 just released
  - Adds support for new MPI-4.0 functionality
  - Includes an embedded UCX 1.11.2
  - Tested with NVIDIA and AMD GPUs
- “Native” path
  - pt2pt over ucp tagged nbx interfaces (new)
  - contiguous put/get for win_create/win_allocate windows
  - atomics support
    - https://github.com/pmodels/mpich/issues/3514 with PR linked
- Generic path is ch4 active messages (new)
  - Migrated from tagged to UCP active messages
- Not supported
  - MPI dynamic processes
    - https://github.com/pmodels/mpich/pull/5467
    - Worker address size an issue. New format may help?

OSU Latency: 0.99us
OSU BW: 12064.12 MB/s
Argonne JLSE Gomez Cluster
- Intel Haswell-EX E7-8867v3 @ 2.5 GHz
- Connect-X 4 EDR
- HPC-X 2.2.0, OFED 4.4-2.0.7

REQUEST HANDLING

- Requests
  - MPICH allocates objects and assigns C integer handle values
    - typedef int MPI_Request;
    - Used as hash value to lookup underlying struct
    - Information can be encoded in the handle value
      - E.g. thread safety information
    - Part of our ABI and unlikely to change

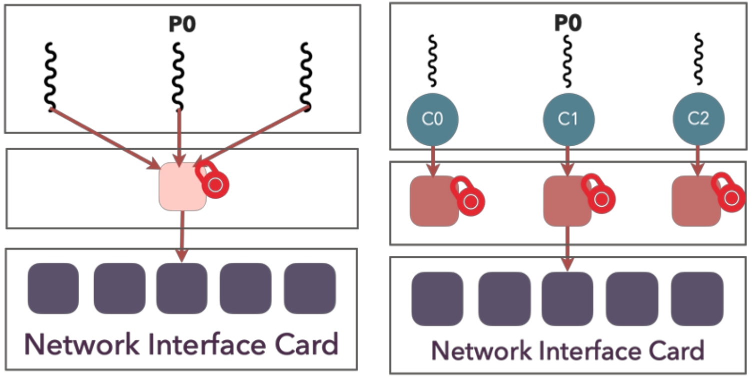
- Adoption of nbx interfaces in MPICH
  - ucp_tag_send_nbx 😐
    - Not using UCP_OP_ATTR_FIELD_REQUEST
    - Force immediate completion flag (my idea) does not work as expected
      - Second attempt might immediately complete!
      - Send request allocation not an issue since progress was removed
    - MPICH code remains largely the same
  - ucp_tag_recv_nbx 🙏
    - Not using UCP_OP_ATTR_FIELD_REQUEST
    - **Major code improvement** with user_data parameter
      - Solves completion function executing without access to MPICH request 🎉

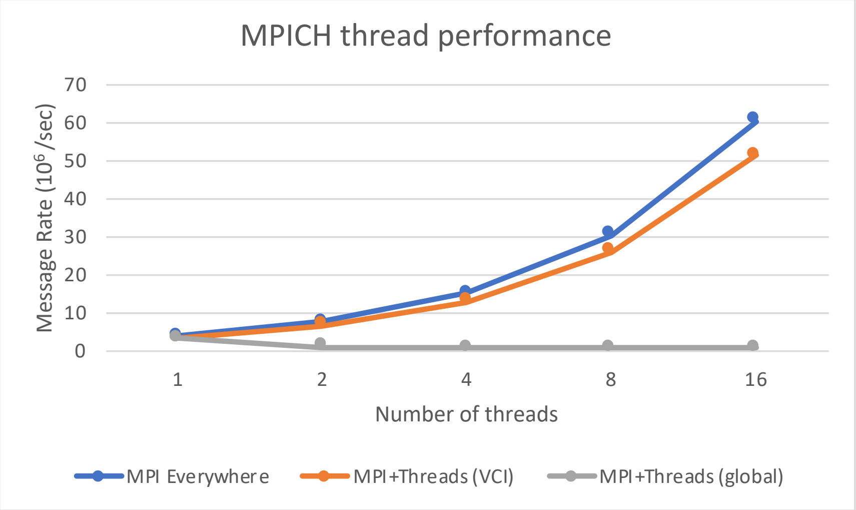
UCP ACTIVE MESSAGES

- MPICH now uses with ucp_am_send_nb
  - Uses whole message flag

  - Good ☺
    - Porting from tagged API was straightforward
    - Eliminated matching overhead for native tagged messages
  - Not so good
    - Data needs to be copied for alignment purposes
      - Fixed in https://github.com/openucx/ucx/pull/6791?
      - Need to test
    - Plan to move to ucp_am_send_nbx
      - Needs more testing
      - Does rndv support device buffers? In our tests no, but recent fixes may have gone in?

VIRTUAL COMMUNICATION INTERFACE (VCI)





Multiple VCIs to preserve parallelism and enable strong scaling.

How I learned to stop worrying about user-visible endpoints and love MPI (ICS '20)
Rohit Zambre, Aparna Chandramowlishwaran, Pavan Balaji

MULTIPLE VCI OVER UCX

- VCI mapped to UCX worker
- Threading model

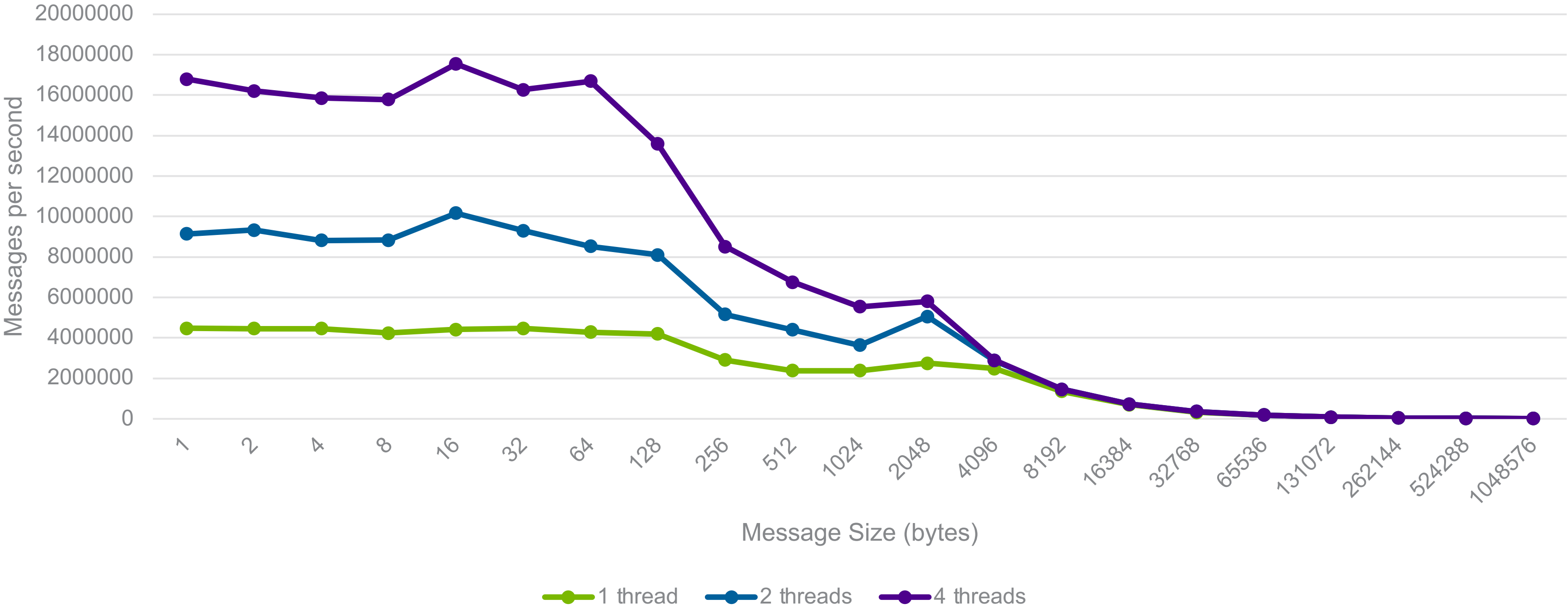
```
ucp_params.mt_workers_shared = 1;
ucp_params.field_mask |= UCP_PARAM_FIELD_MT_WORKERS_SHARED;
worker_params.field_mask = UCP_WORKER_PARAM_FIELD_THREAD_MODE;
worker_params.thread_mode = UCS_THREAD_MODE_SERIALIZED;
```

- Address exchange

```
for i_local=0:num_vnis
    for r=0:size
        for i_remote=0:num_vnis
             ucp_ep_create(ctx[i_local].worker, &ep_params,
&av[r][i_local][i_remote];
```

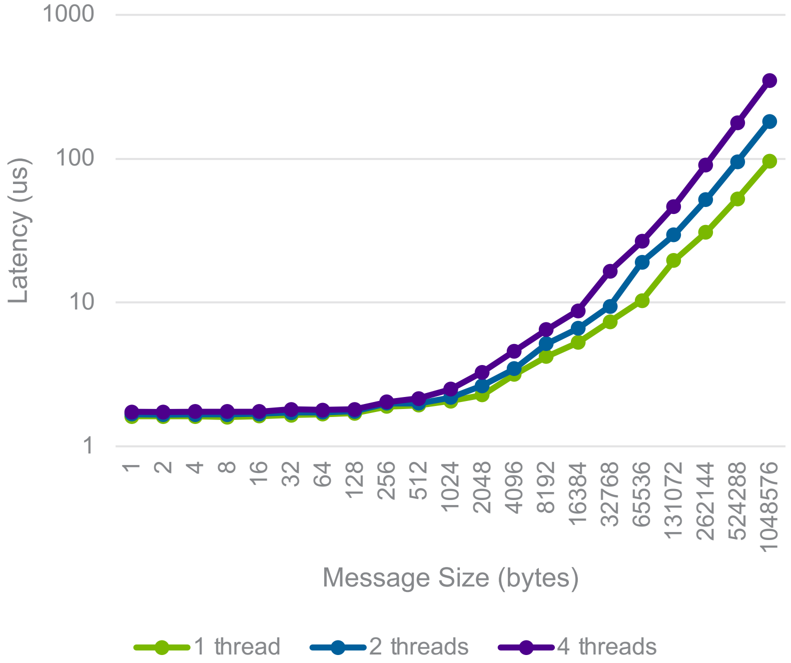
MT.COMB BENCHMARK (PT2PT MSG RATE)

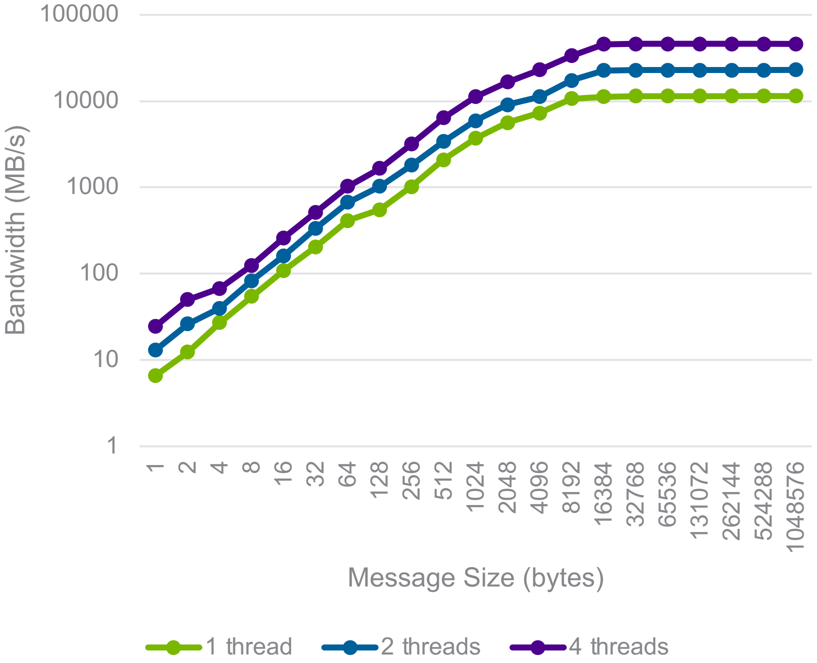
Intel Xeon Platinum 8180M, ConnectX-6, ppn=1, comm per thread



RMA-MT PERFORMANCE

-o put -s flush -w





FUTURE DEVELOPMENT

- Support for some non-contig RMA
  - Issue multiple operations for dense data
  - Packing for sparse data
- Dynamic Process Management
  - Test new address format
- Native atomics
  - https://github.com/pmodels/mpich/issues/3514
- UCX collectives
  - Prototype and evaluation
  - Hopefully straightforward port from HCOLL

OTHER ODDS AND ENDS

- MPICH+UCX Jenkins tests passing with sanitizers
  - AddressSanitizer
    - export UCX_MEM_MALLOC_HOOKS=n
    - export UCX_MEM_MMAP_HOOK_MODE=none
  - UndefinedBehaviorSanitizer
    - Good for uncovering bugs on non-x86_64
    - E.g. active message alignment issue

POINTERS

- Website
  - www.mpich.org
- Mailing Lists
  - lists.mpich.org
- Github
  - http://github.com/pmodels/mpich
  - Submit an issue or pull request!
- Slack (pmrs.slack.com)
  - Ping me an invite