UCF VIRTUAL WORKSHOP

Argonne NATIONAL LABORATORY

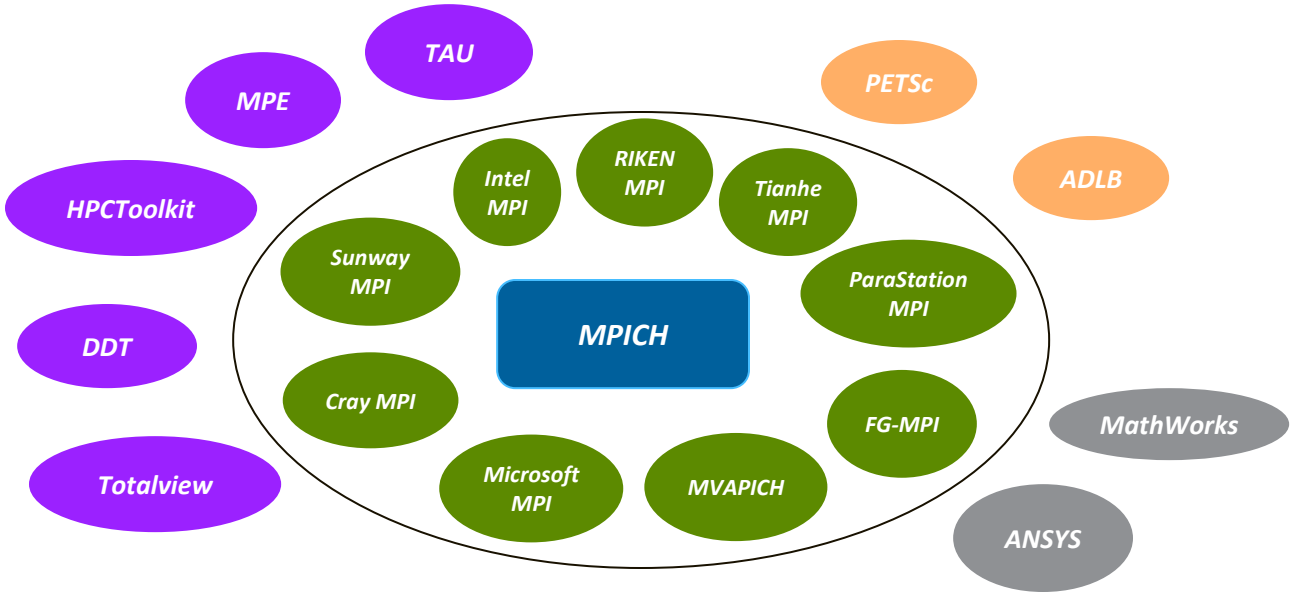
MPICH/UCX UPDATE

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

November 30, 2020

MPICH: GOALS AND PHILOSOPHY

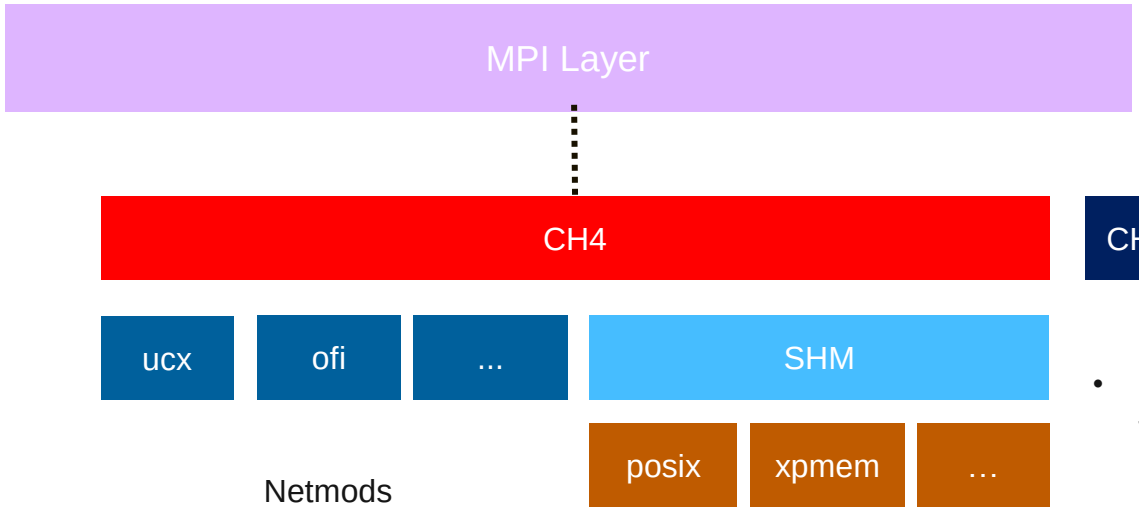
- MPICH continues to aim to be the preferred MPI implementations on the top machines in the world
- Our philosophy is to create an “MPICH Ecosystem”



AGENDA

- UCX Support in MPICH
- Pain Points (Past and Present)
- Active Messages
- Multi-VCI
- Other Odds and Ends

MPICH LAYERED STRUCTURE: CH4



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
  - MPICH Team
  - Tommy Janjusic (Mellanox)
- MPICH 3.4 just released
  - Includes an embedded UCX 1.9.0
- Native path
  - pt2pt
  - put/get for win_create/win_allocate windows
  - atomics pull request
    - https://github.com/minsii/mpich/pull/1
- Emulation path is ch4 active messages
  - Layered over UCX tagged API
  - Prototype over UCP active messages (details later)
- Not supported
  - MPI dynamic processes

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

PAIN POINT ☹

- Requests
  - MPICH allocates requests and assigns C integer handle values
    - Used as hash value to lookup struct
    - Other information can be encoded in the handle value
    - Part of our ABI and unlikely to change
  - ucp_tag_{send|recv}_nb allocates a ucp request
    - MPICH does a second allocation
  - ucp_tag_{send|recv}_nbr allows caller to provide a request
    - Unnecessary allocation when inline send is possible
    - Need to track/complete nbr requests separately

TAGGED NBX INTERFACES

- 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 🎉

PAIN POINT ☹

- Datatypes
  - UCX netmod passes contig or fully-generic pack/unpack function pointers
  - No intermediate support

PAIN POINT ☹

- Datatypes
  - UCX netmod passes contig or fully-generic pack/unpack function pointers
  - No intermediate support
- Yaksa datatype library
- Datatype working group
  - Pavan will provide more info

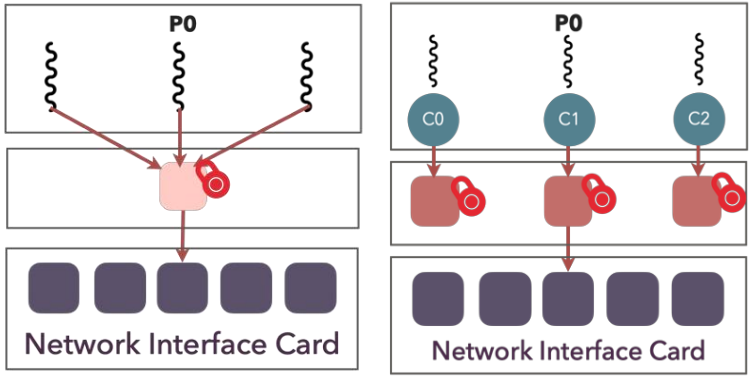
UCP ACTIVE MESSAGES

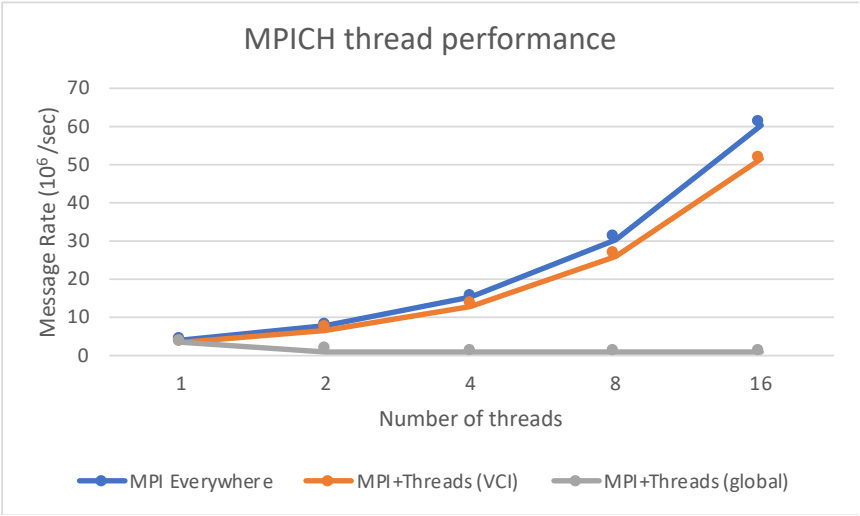
- Prototype with ucp_am_send_nb (1.9.0)
  - https://github.com/pmodels/mpich/pull/4934
  - Uses whole message flag
  - Good ☺
    - Porting from tagged API was straightforward
    - Eliminates matching overhead for native tagged messages
  - Not so good
    - Data needs to be copied for alignment purposes
      - Need to investigate ucp_am_send_nbx and rndv capability
      - Will rndv support device buffers?
    - Seems to be a bug with self transport
      - Working on minimal reproducer

PAIN POINT ☹

- How to integrate new interfaces into MPICH?
  - How far back in version should we go?
  - send/recv NBX added in 1.7.0
  - CentOS 7 provides UCX 1.5.2

VIRTUAL COMMUNICATION INTERFACE (VCI)





Multiple VCIs to preserve parallelism and enable strong scaling.

MULTIPLE VCI OVER UCX

- VCI mapped UCX worker
- Threading model

```
ucp_params.mt_workers_shared = 1;
ucp_params.field_mask |= UCP_PARAM_FIELD_MT_WORKERS_SHARED;
ucp_init(&ucp_params, config, &context);
```

- 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];
```

- Need to flush every worker to ensure RMA progress

OTHER ODDS AND ENDS

- MPICH adopting C99 features
  - Plus compiler atomics (C11 or other available)

- MPICH testing added support for sanitizers
  - AddressSanitizer
    - Faster and easier Valgrind
  - UndefinedBehaviorSanitizer
    - Good for uncovering bugs on non-x86_64
    - E.g. alignment

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