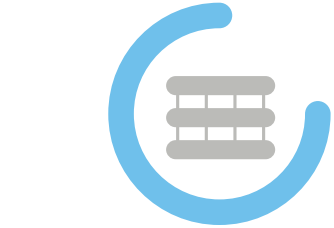


Meeting of the Technical Steering Committee (TSC) Board

Wednesday, January 30th, 2019
11:00am ET

Meeting Logistics

- https://zoom.us/j/556149142

- United States : +1 (646) 558-8656
  - Meeting ID: 556 149 142

Antitrust Policy Notice

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

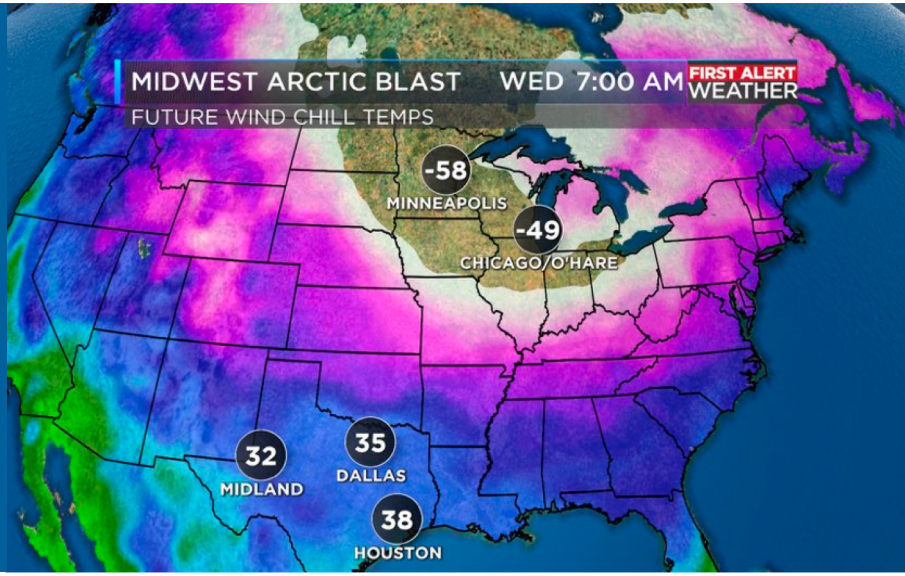
THE LINUX FOUNDATION

Agenda

- TSC Calendar meeting update
  - apologies for scheduling confusion last week
  - Neal has updated Calendar entries on the TSC groups.io list to be reflective of bi-weekly meetings
- CI update (from last time):
  - ~~centos7.6 image for x86 is now available (tested with previous ohpc 1.3.6 release)~~
  - ✓ centos7.6 image for aarch64 is now also live
- Reminder on upcoming submission deadlines:
  - PEARC'19 tutorial (Feb20)
  - ISC'19 BoF (Feb 20)
- MPICH 3.3
  - first stable release in 3.3 series
  - introduces new CH4 device layer implementation
  - Guest Presenter: **Ken Raffenetti**, Argonne National Laboratory
- Continued our discussion on next major distro versions:
  - SLE12
  - RHEL8/CentOs8

Argonne NATIONAL LABORATORY

MPICH CH4 DEVICE



KEN RAFFENETTI
Principal Software Development Specialist
Argonne National Laboratory

January 30th, 2019
OpenHPC Technical Steering Committee Meeting

THE MPICH PROJECT

- MPICH and its derivatives are the world's most widely used MPI implementations
  - Supports all versions of the MPI standard including the recent MPI-3.1
- Funded by DOE for 26 years
- Has been a key influencer in the adoption of MPI
- Award winning project
  - DOE R&D100 award in 2005

MPICH and it derivatives in the Top 10

1.Summit (US): Spectrum MPI

2.TaihuLight (China): Sunway MPI

3.Sierra (US): Spectrum MPI

4.Tianhe-2A (China): MPICH-TH2

5.ABCI (Japan): Intel MPI and MVAPICH

6.Piz Daint (Germany): Cray MPI

7.Titan (US): Cray MPI

8.Sequoia (US): IBM PE MPI

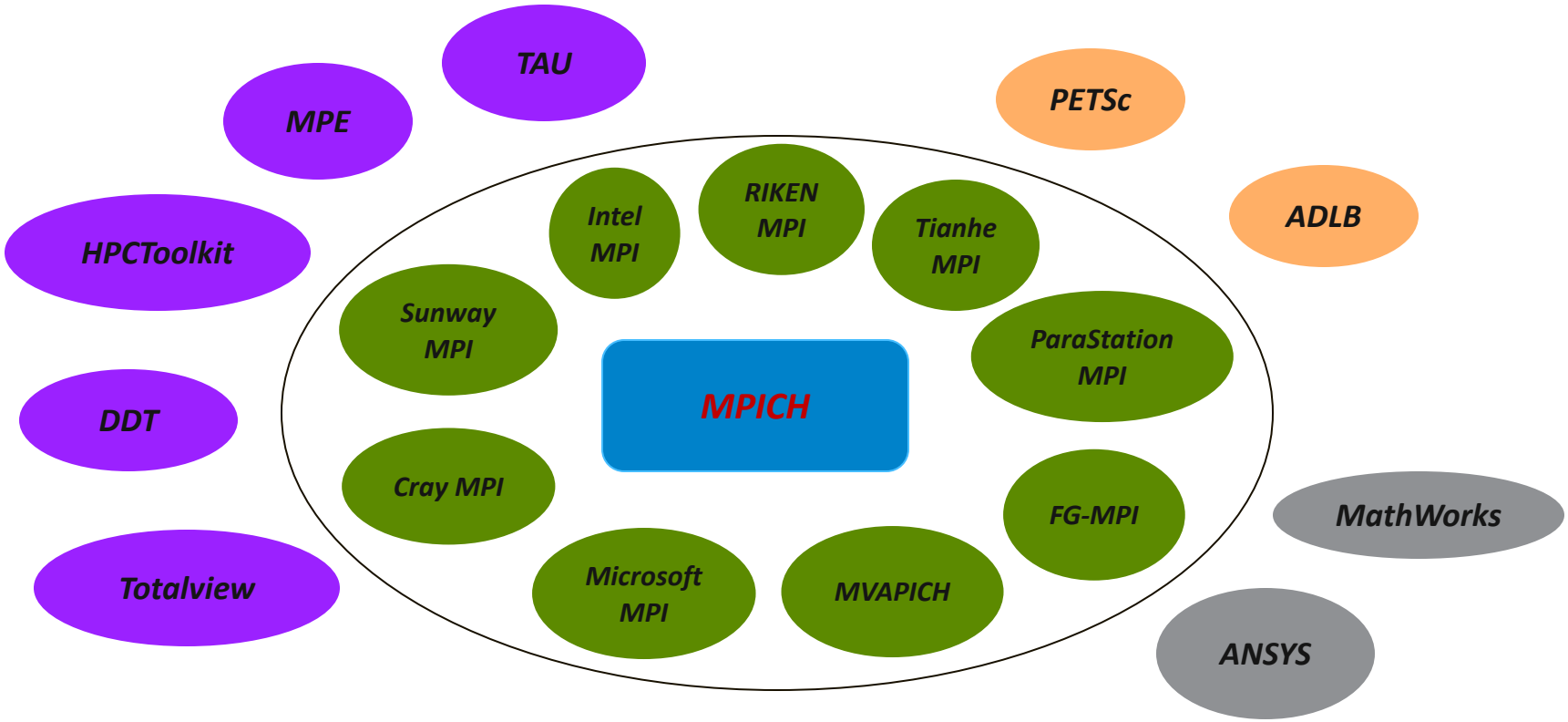
9.Trinity (US): Cray MPI

10.Cori (US): Cray MPI

MPICH and its derivatives power 8 of the top 10 supercomputers (Jun. 2018 Top500 rankings)

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"



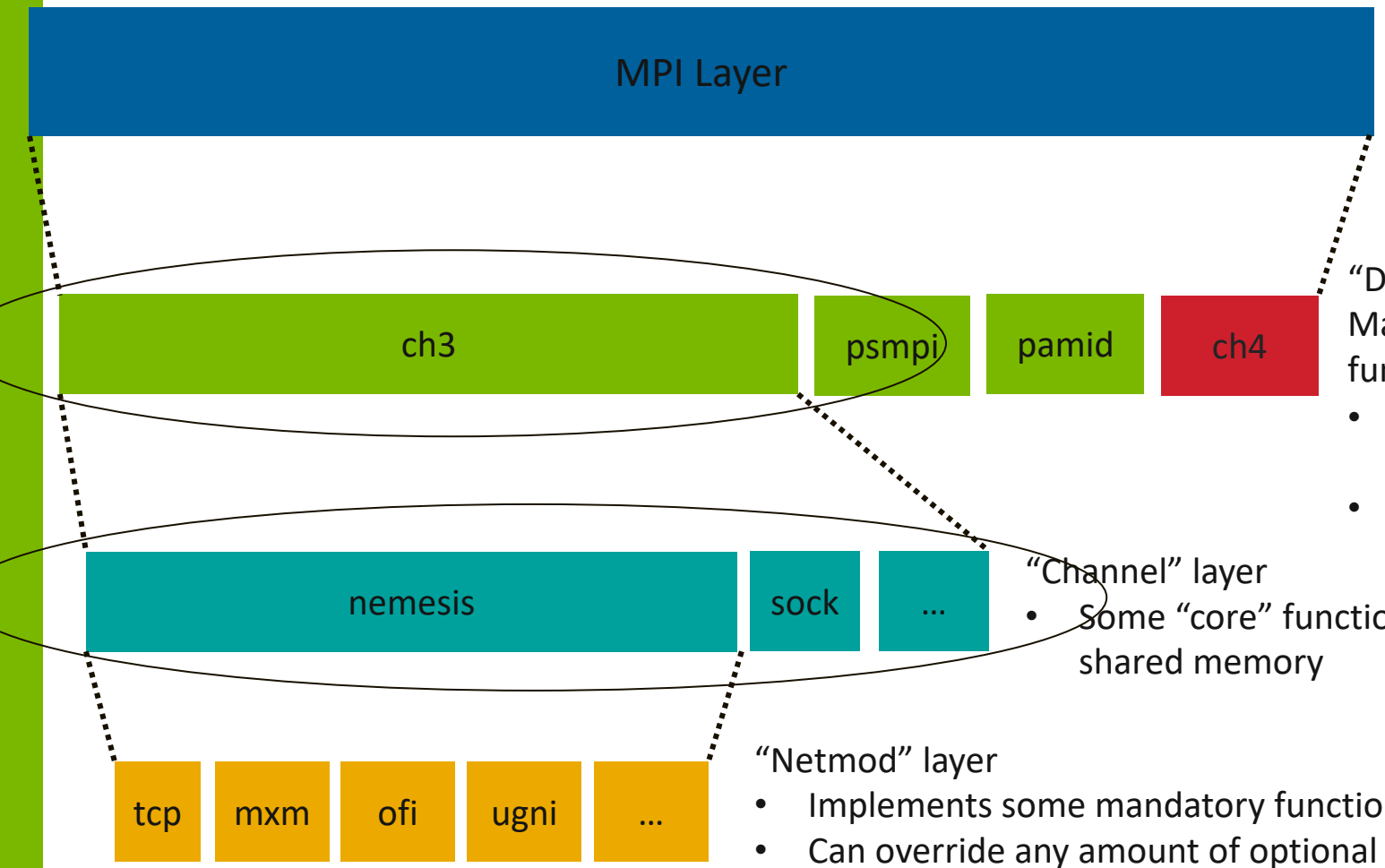
MPICH RELEASES

- MPICH typically follows an 18-month cycle for major releases (3.x), barring some significant releases
  - Minor bug fix releases for the current stable release happen every few months
  - Preview releases for the next major release happen every few months
- Current stable release is in the 3.3.x series
  - mpich-3.3 was released in November 2019
  - Bug-fix releases will follow
- Next major release, mpich-3.4, will be at SC 2019

MPICH-3.3 FEATURES

1. New device ch4: Low-instruction count communication
   - ***Thanks to Intel, Mellanox, and RIKEN for their significant contributions!***
   - Very lightweight communication
2. Support for very high thread concurrency (***partnership with Intel***)
   - Improvements to message rates in highly threaded environments
   - Move from the LOCK/WORK/UNLOCK model to a scalable ENQUEUE/DEQUEUE model
3. Memory Scalability Optimizations
4. Hardware Acceleration for MPI Atomics
5. Scalable job startup
6. New Collective Infrastructure (***partnership with Intel***)
   - New collective algorithms, more comprehensive algorithm selection
7. Topology-awareness improvements
8. Fault tolerance improvements
   - Non-catastrophic errors and limited abort scope
9. CUDA-awareness for contiguous GPU buffers (***partnership with Mellanox and NVIDIA***)

MPICH LAYERED STRUCTURE: CURRENT AND FUTURE



Platform independent code
- Collectives
- Communicator management

"Device" layer: Machine-specific functionality
- Derived datatype management
- RMA

"Channel" layer
- Some "core" functionality, such as shared memory

"Netmod" layer
- Implements some mandatory functionality
- Can override any amount of optional functionality (better bcast, better barrier)

CH4 DESIGN GOALS

High-Level Netmod API

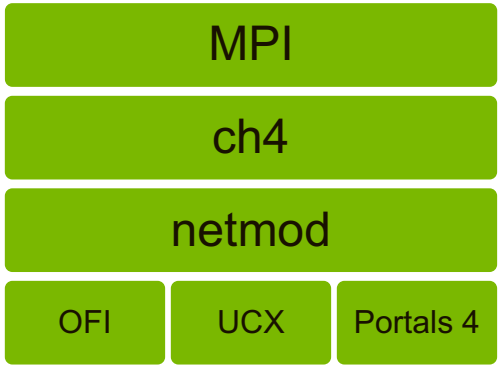
- Give more control to the network
  - `netmod_isend`
  - `netmod_irecv`
  - `netmod_put`
  - `netmod_get`
- Fallback to Active Message based communication when necessary
  - Operations not supported by the network

Provide default shared memory implementation in CH4

- Disable when desirable
  - Eliminate branch in the critical path
  - 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 overhead

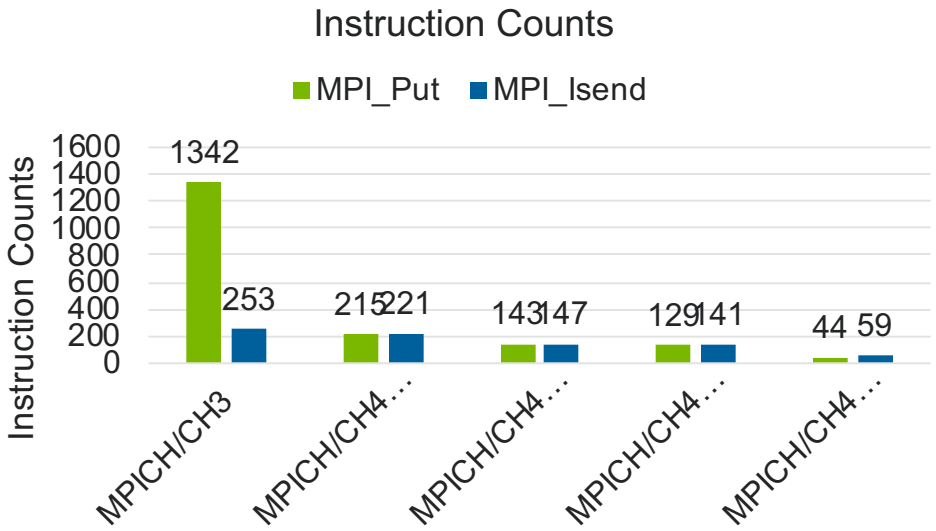


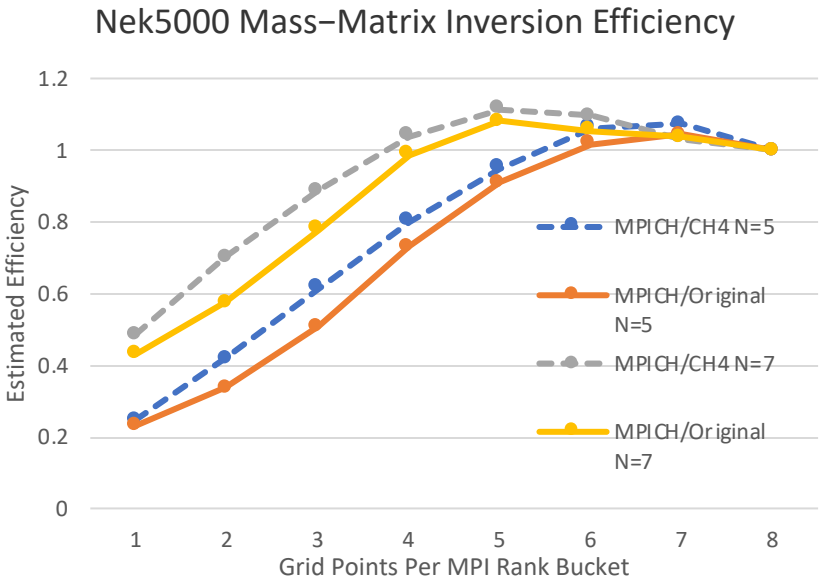
Minimal Per Process Data

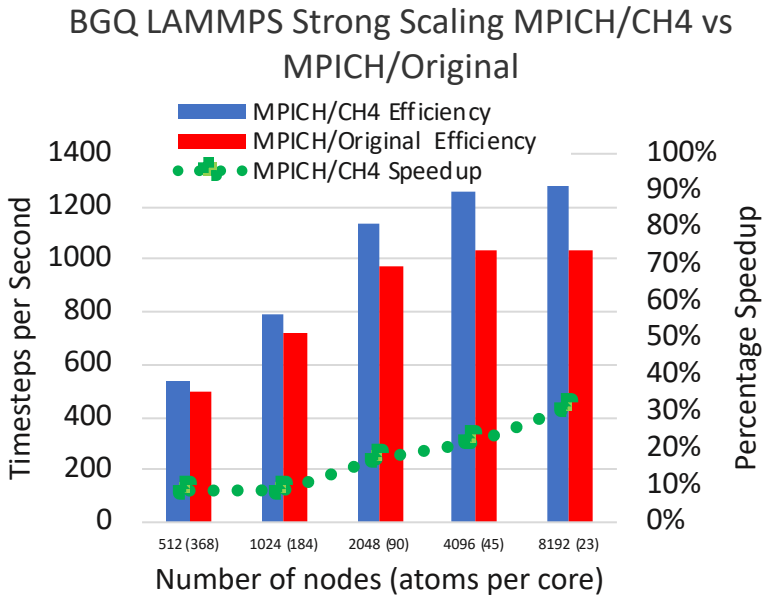
- Global address table
  - Contains all process addresses
  - Index into global table by translating `(rank+comm)`

Partnership with Intel, Mellanox, RIKEN

LOWER OVERHEADS = BETTER STRONG SCALING



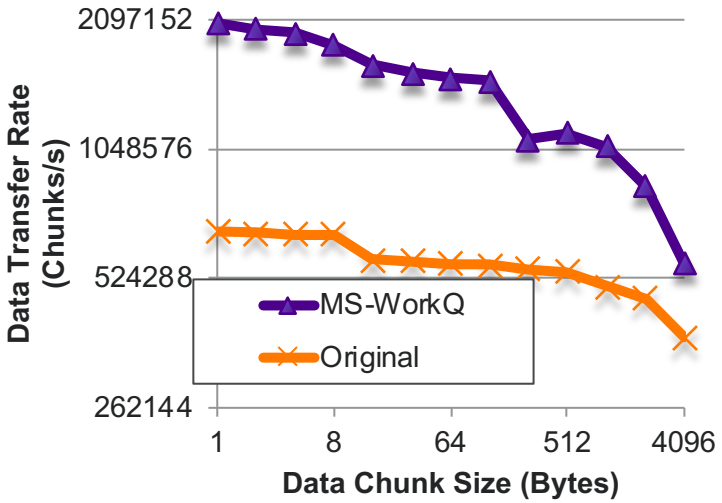


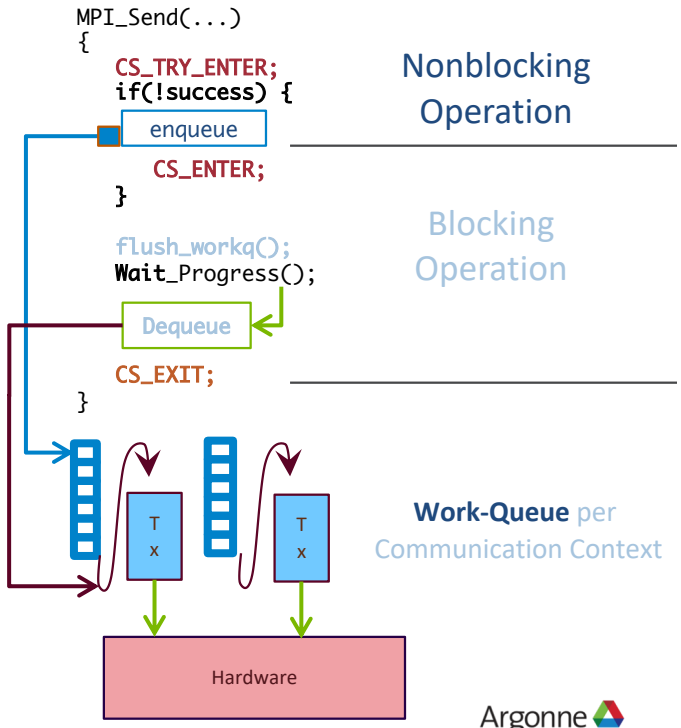


MULTITHREADED MPI WORK-QUEUE MODEL

- **Context**
  - Existing lock-based MPI implementations **unconditionally** acquire locks
  - **Nonblocking** operations may **block** for a lock acquisition
    - **Not** truly nonblocking!
- **Consequences**
  - Nonblocking operations may be slowed by blocking ones from other threads
  - Pipeline stalls: higher latencies, lower throughput, and less communication-computation overlapping
- **Work-Queue Model**
  - One or multiple **work-queues per endpoint**
  - Decouple blocking and nonblocking operations
  - Nonblocking operations enqueue **work descriptors** and leave if critical section held
  - Threads issue work on behalf of other threads when acquiring a critical section
  - Nonblocking operations **are truly** nonblocking

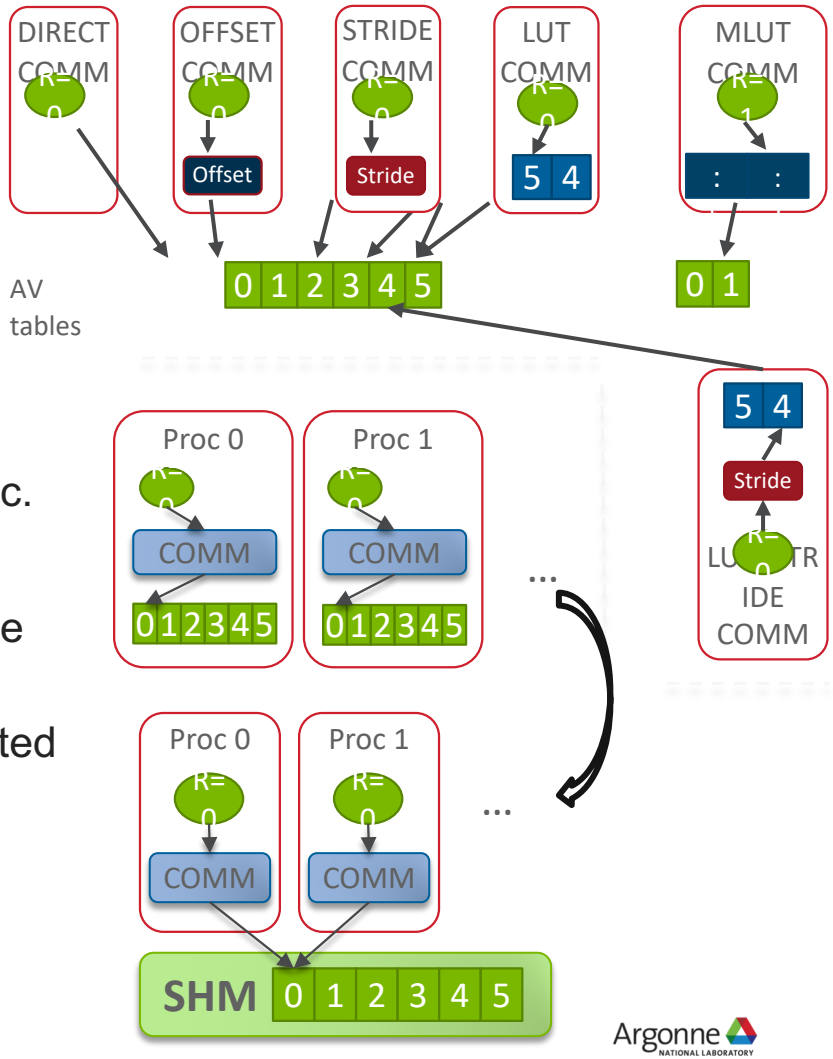
Partnership with Intel



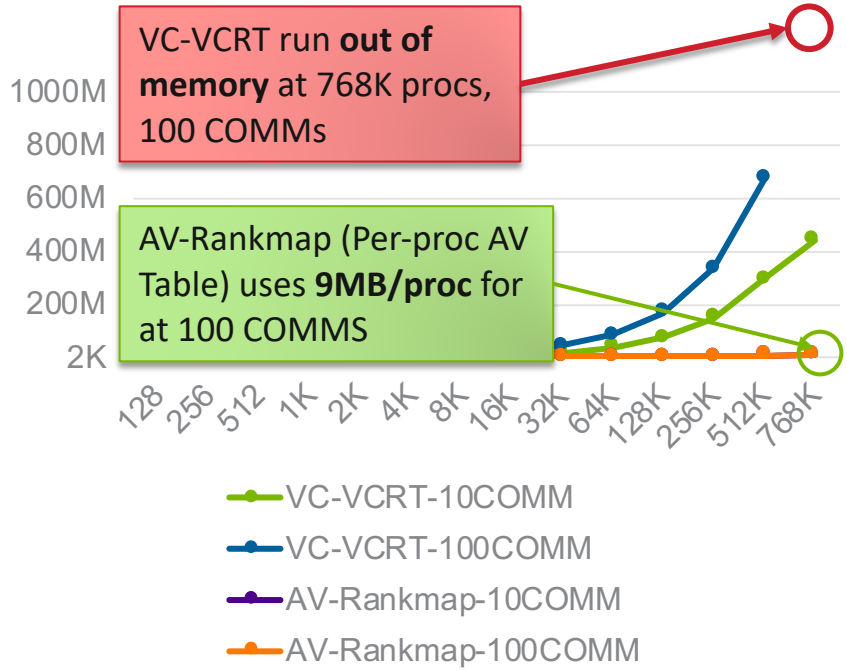


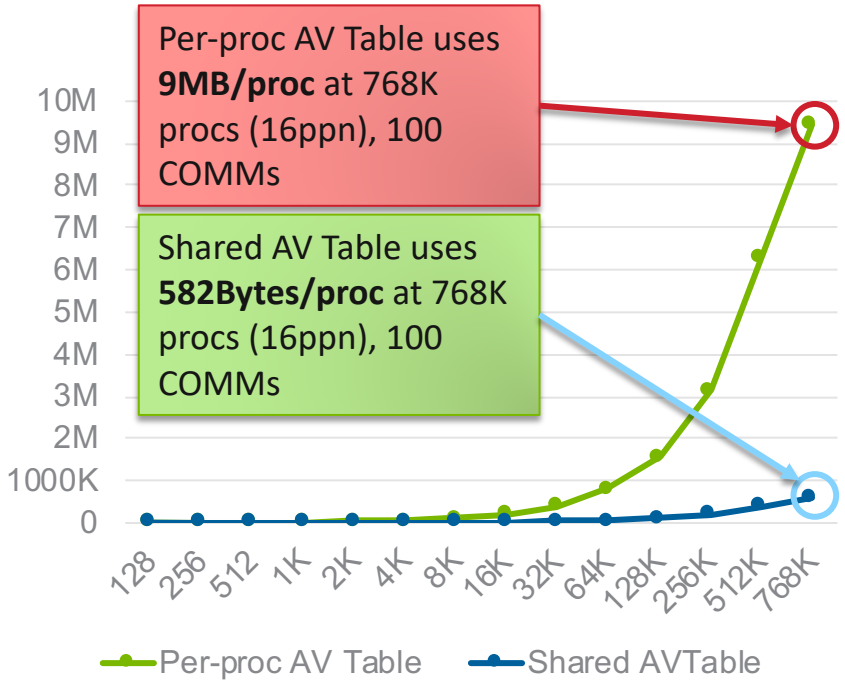
MEMORY SCALABLE NETWORK ADDRESS MANAGEMENT

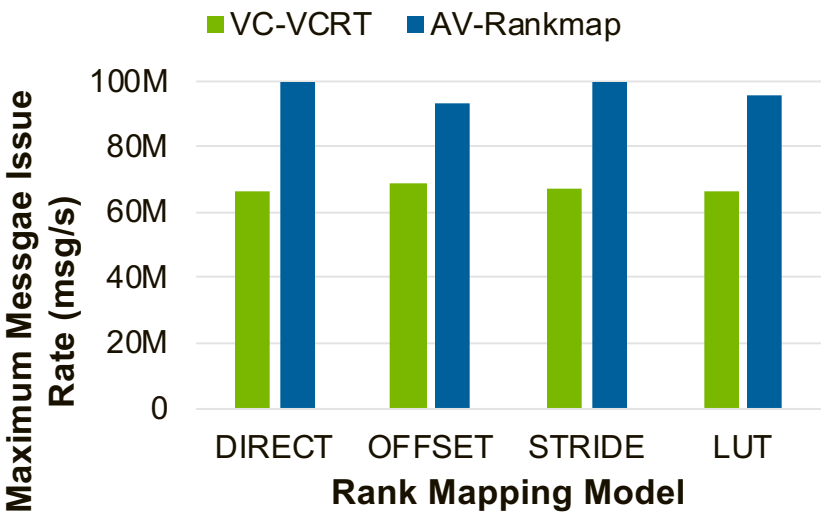
- AV Table: Compressing VC (480Bytes -> 12Bytes)
  - Compressing Multitransport Functionality
    - Function pointers are moved to a separate array
  - Deprioritizing Dynamic Processes
    - Process group information moved to COMM
- Rank Mapping Models
  - **Regular**: DIRECT, OFFSET, STRIDE, STRIDE_BLOCK
  - **Irregular**: LUT, MLUT
  - **Mixed**: LUT_STRIDE, LUT_STRIDE_BLOCK, etc.
- Shared AV Tables
  - AV Tables in shared memory for processes on the same node
  - Shared AV Table 0 (MPI_COMM_WORLD): created at init time, read-only, lock-free
  - Per-proc AV Tables (dynamic processes): avoid locking

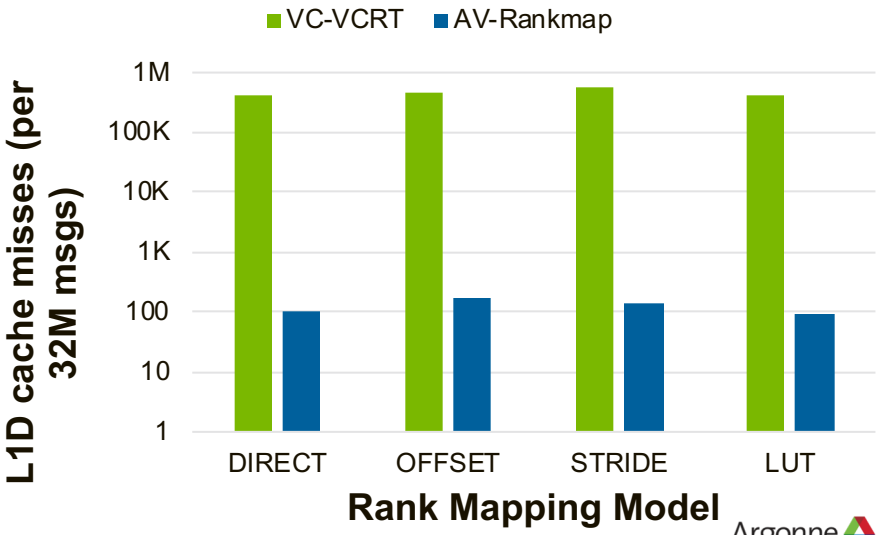


MEMORY SAVING AND PERFORMANCE IMPACT





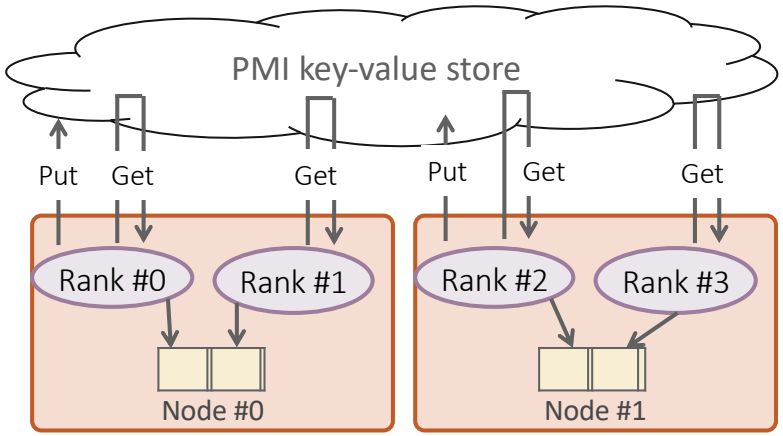


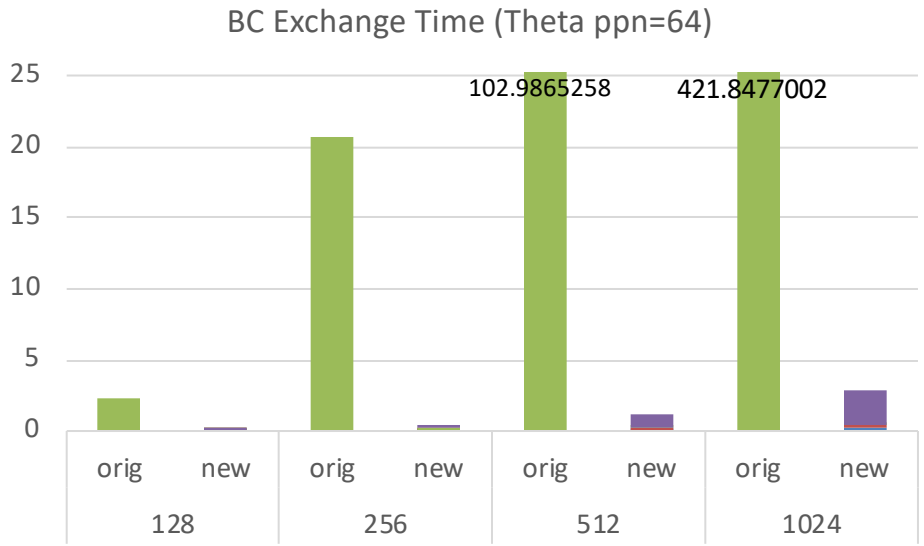


SCALABLE JOB STARTUP

Using shared memory and MPI collectives to reduce startup time

- Only node-root processes do PMI_KVS_Put
  - All processes on the node do PMI_KVS_Get into shared memory segment
  - No redundant lookups at the node level
- Remaining business cards are exchanged with MPIR_Allgather using the node-roots communicator, again into shared memory.
- BC exchange time reduced from 421 -> ~3 seconds on 1024 KNL nodes, 64 ppn on ALCF Theta (MPICH + OFI/gni)
- Launch time of ~8 seconds on the full OFP machine (8192 KNL nodes/64 ppn)





SUMMARY

- ch4 device features
  - Lower overheads
    - Instructions
    - Memory footprint
  - Better thread scalability
  - Support today fabric libraries
    - libfabric
    - UCX (libucp)

- Development and release plans
  - *all* active development happening in ch4
    - ch3 effectively moved to maintenance mode
  - mpich-3.4 release will make ch4 default
  - mpich-3.5 (4.0?) release will remove ch3

MPICH ABI COMPATIBILITY INITIATIVE

- Binary compatibility for MPI implementations
  - Started in 2013
  - Explicit goal of maintaining ABI compatibility between multiple MPICH derivatives
  - Collaborators:
    - MPICH (since v3.1, 2013)
    - Intel MPI Library (since v5.0, 2014)
    - Cray MPT (starting v7.0, 2014)
    - MVAPICH2 (starting v2.0, 2017)
    - Parastation MPI (starting v5.1.7-1, 2017)
    - RIKEN MPI (starting v1.0, 2016)

Argonne NATIONAL LABORATORY

intel

MVAPICH

CRAY THE SUPERCOMPUTER COMPANY

ParaStation MPI

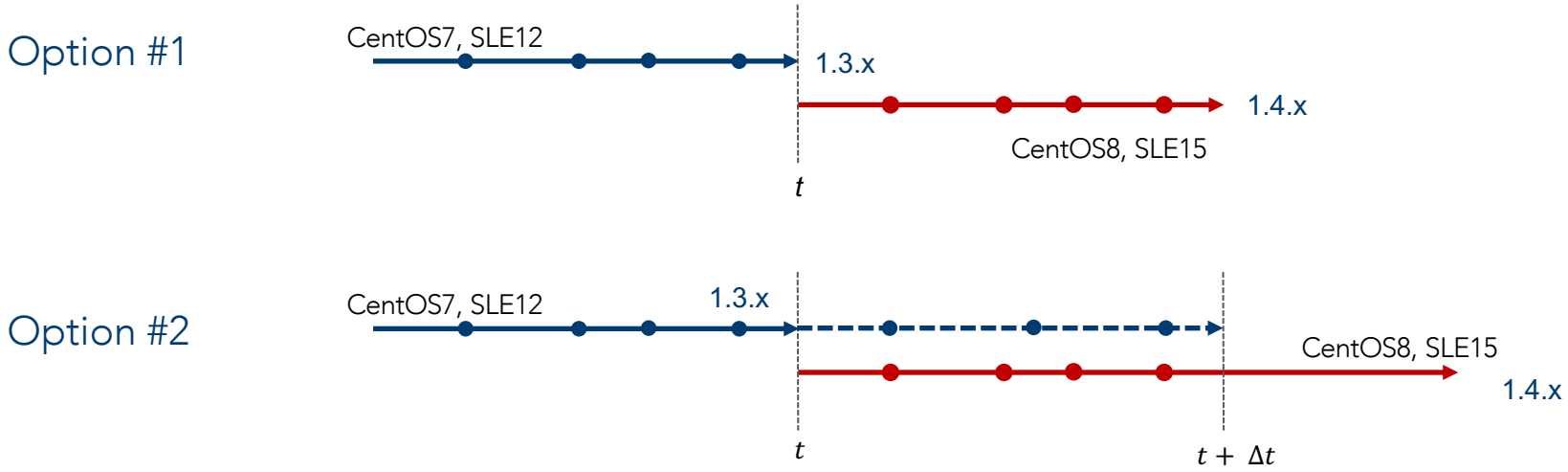
RIKEN AICS RIKEN Advanced Institute for Computational Science

- Open initiative: other MPI implementations are welcome to join
- http://www.mpich.org/abi

THANK YOU
QUESTIONS?

Next major distro versions

(*continued from last time*)



- Reminder: SLE15 and RHEL8 are coming (previously said we would target SLE 15SP1)
- High level considerations: Option #1 *vs* Option #2
  - Option #1 is easiest, we simply switch to 1.4.x branch and no more updates for 1.3.x
  - Option #2 continues to have some potential release in 1.3.x branch, **multiple degrees of freedom to consider:**
    - after time *t*, releases could potentially be synchronized or not between 1.3.x and 1.4.x
    - after time *t*, what types of updates would be eligible for 1.3.x?
      - security patches only?
      - significant bug fixes for existing component versions?
      - version updates to match changes in 1.4.x branch?
      - updates and testing for minor distro updates (e.g. what happens when CentOS 7.8 is release after we have CentOS8.x out)?
      - new compiler and MPI variants?
      - new component additions?
    - what is practical value for Δt?
      - 6 months
      - 1 year
      - ???