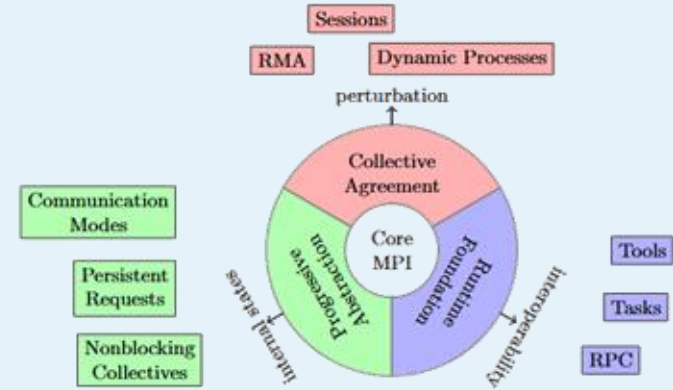


CORE MPI



EuroMPI 2026

Oct 7-9, 2026, Vienna, Austria

Hui Zhou
Argonne National Laboratory

OUTLINE

- ❖ State of MPI
- ❖ Principles of Core MPI
 - Principle 1
 - Principle 2
 - Principle 3
- ❖ MPI Core Set
- ❖ Case Studies
- ❖ Summary and Q&A

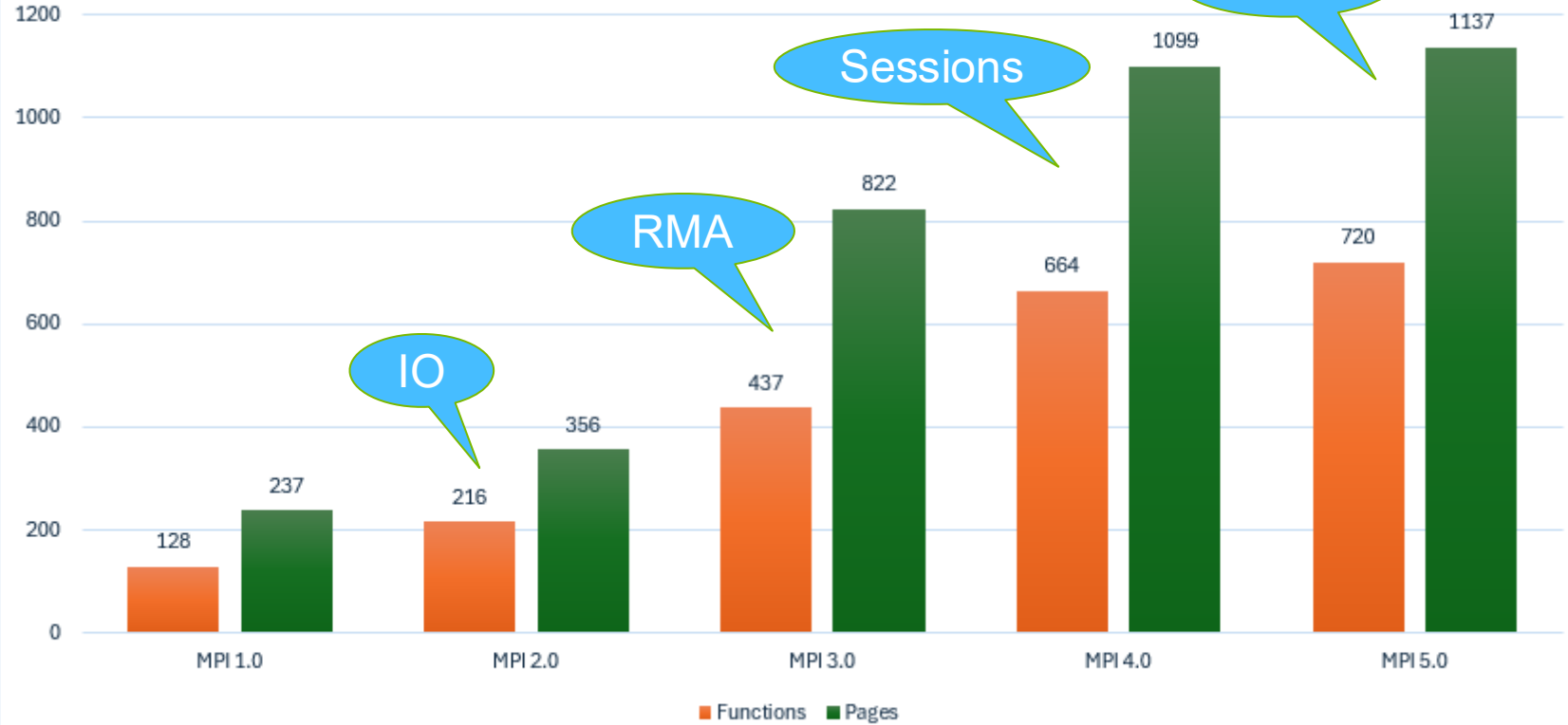
INTRODUCTION: STATE OF MPI



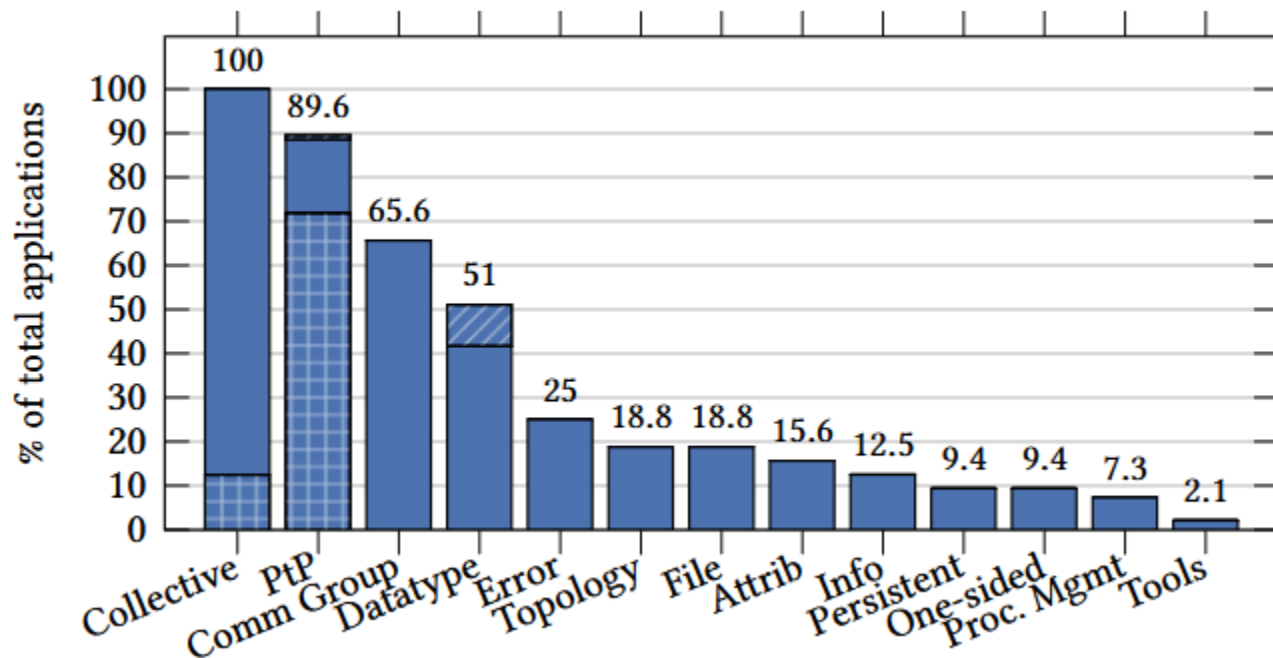
Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.



The Expansion of the MPI Standards



User Adoptions

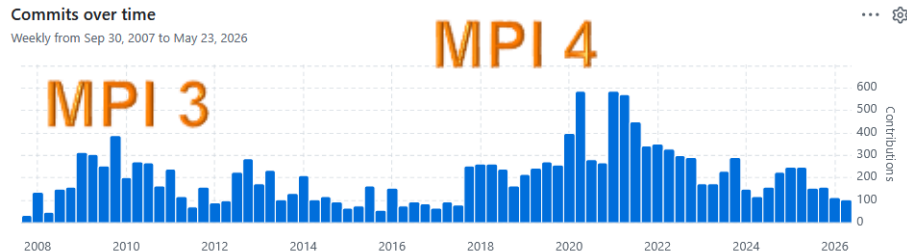


Hück, Alexander, Tim Jammer, Joachim Jenke, and Christian Bischof. "Investigating the usage of MPI at argument-granularity in HPC codes." In *Proceedings of the 30th European MPI Users' Group Meeting*, pp. 1-11. 2023.

Figure 2: MPI feature usage. The squared areas represent contributions of non-blocking functions. For PtP and Datatypes, the cross-hatched areas are contributions defined by the standard that we do not consider as *core* functionality for our analysis. For PtP, this includes request handling, which is also used for collectives. For Datatypes, this includes functionality that does not construct or duplicate a type.

Implementations

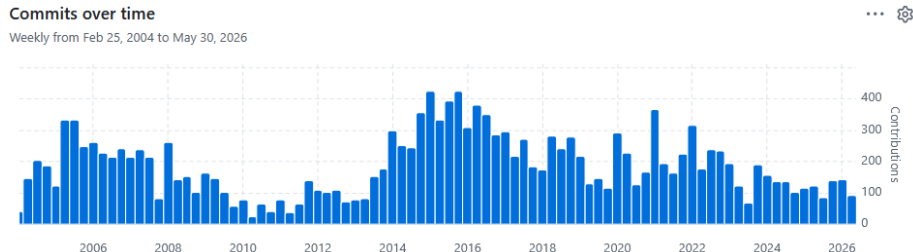
MPICH



- Limited by DOE fundings.
- Burdened with feature implementation.
- Depended upon by vendors and users.
- Monolithic MPI – user cannot easily vibe-code around.



OPEN MPI





'Logically concurrent' isn't #117

Closed



jsquyres opened on Dec 8, 2018

417 remaining items

Load more



wesland moved this from Had Reading to Passed
1st Vote in MPI 4.1 on Feb 8, 2023

```
1516 + \begin{users}
1517 + The MPI/ Forum believes the following paragraph is ambiguous and
      + may clarify the meaning in a future version of the MPI/ Standard.
```

User-Level Failure Mitigation #20

Open



wesland opened on Nov 11, 2015

Last edited by abouteiller

ULFM Fault Tolerance (slice 3: shrink)
mpi-next scheduled 1st vote scheduled no-no vote
scheduled reading wg-ft 1

#583 · abouteiller opened on Sep 12, 2022 · June 2026

ULFM: fault model is not completely defined
had reading mpi-next scheduled 1st vote
scheduled reading wg-ft 1 1

#816 · abouteiller opened on Nov 13, 2023 · June 2026

ULFM Fault Tolerance (slice 2: agree)
had reading mpi-next scheduled reading wg-ft 1 2

#582 · abouteiller opened on Sep 12, 2022

Understanding GPU Triggering APIs for MPI+X Communication

Patrick G. Bridges¹[0000-0003-4801-03903], Anthony Skjellum²[0000-0001-5252-6600], Evan D. Suggs²[0000-0002-8210-8992], Derek Schafer¹[0000-0001-8438-5144], and Purushotham V. Bangalore³[0000-0002-1098-9997]

¹ University of New Mexico, Albuquerque NM, 87131
{patrickb,dschafer}@unm.edu

² Tennessee Technological University, Cookeville, TN 38505
{askjellum,esuggs}@tttech.edu

³ The University of Alabama, Tuscaloosa, AL 35401
pvbangalore@ua.edu

Table 1. Taxonomy of MPI GPU-Triggered Communication Proposals

Proposal	Area 1 Control Path Used	Area 2: API Features				
		Reuses Existing APIs	Changes API Semantics	Separate Initialize and Start	GPU Completion Support	Collective Support
MPI-GDS	Stream	Yes	Weaker	No	Full	Full ¹
MPI-ACX Enqueued	Stream	Yes ¹	No	Yes ¹	Full	No
MPICH Triggering	Stream	Yes	No ²	No	Full	Partial
HPE Send-Recv	Stream	No	No	Yes	Full	No
Delorean	Stream	No ³	No	Yes	Full	Full
HPE One-sided	Stream	Yes	Stronger ⁴	Yes	Full	Group
Partitioned Comm.	Kernel	Yes	No	Yes	Partial ⁵	Full ¹
HPE Persistent	Kernel	Yes	No	Yes	Full	No
Intel GPU-Init	Kernel	Yes	No	No	Full	No

IS MPI TOO BIG TO MEET THE FUTURE?

Re-visiting the ideal of subsetting MPI

CORE MPI



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

CORE MPI

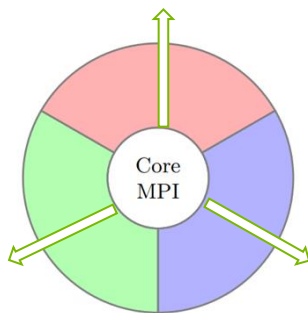
CORE MPI IS NOT:

- “MPI-1” -
which denies 30 years of progress
- “A mere subset” -
which can be subjective and arbitrary

RATHER:



- It's a **structure**
 - ❖ with principles and axes
 - ❖ helps users / developers / forum
 - ❖ to organize MPI
 - ❖ to extend MPI



We identify 3 core MPI principles

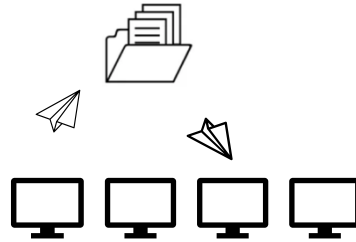
PRINCIPLE #1

MPI IS COLLECTIVE

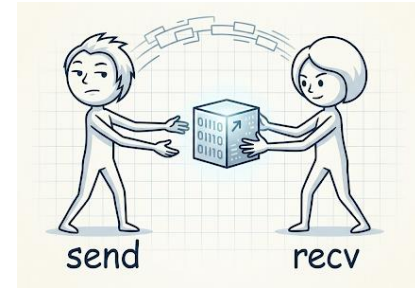
client / server



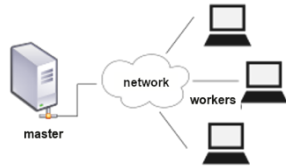
task dispatching



MPI



master / worker



event driven



dynamic discovery

assumed collective

THE ADVANTAGE OF ASSUMED COLLECTIVE

- Simpler to program
- Code is more explicit
- More efficient and performant

- Effective when the collective assumptions hold
- Suitable for dedicated and controlled facility (e.g. HPC)

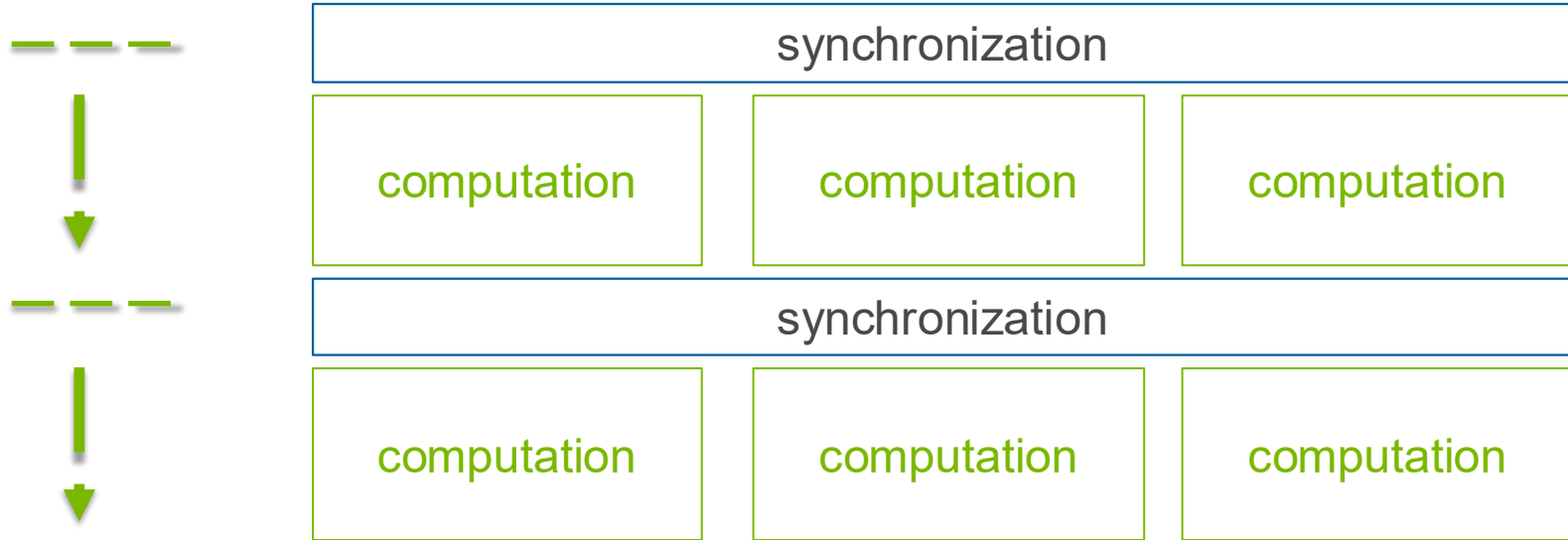


Message ordering and matching is the missing physics.



- Point-to-point messaging is dynamic in nature.
- It can be used to implement most dynamic designs.
- But it's key role in MPI is to support collective.
- Pt2Pt to Collective is like body to soul.

Bulk synchronous parallel (BSP) is the native MPI pattern.



Extending MPI by **Perturbation** to Assumed Collective Agreement

RMA

```
MPI_Win_create()
```

```
MPI_Win_fence()
```

```
Put/Get
```

```
MPI_Win_fence()
```

```
MPI_Win_free()
```

Dynamic Processes

```
MPI_Comm_spawn()
```

```
MPI_Intercomm_merge()
```

```
/*
```

```
 * Conventional
```

```
 * MPI Algorithms
```

```
*/
```

MPI_ANY_SOURCE

MPI_ANY_TAG

MPI_Probe

Inter communicators

MPI Sessions

...

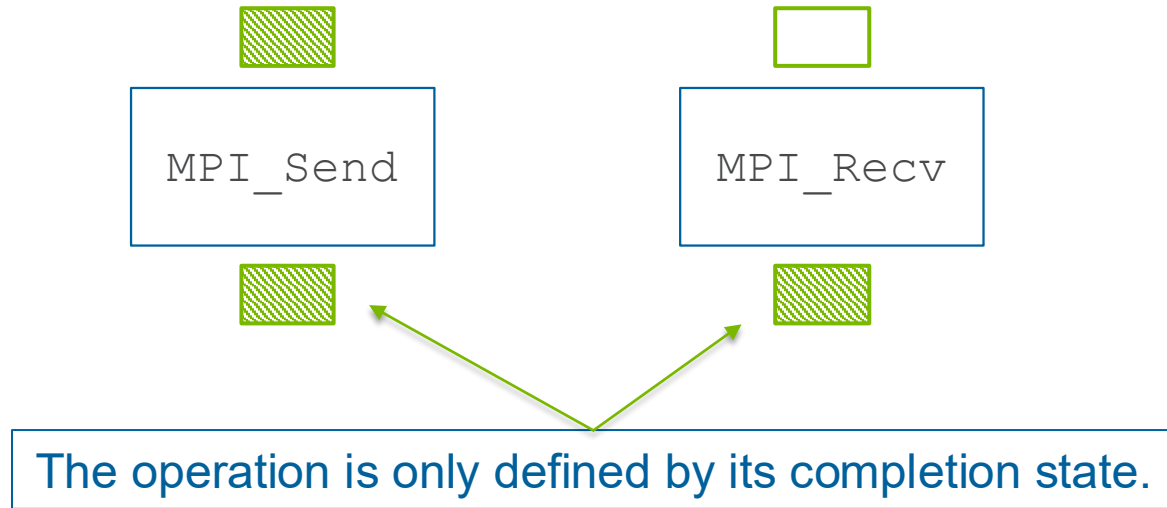
Caution: perturbation
may lead to state
jump



PRINCIPLE #2

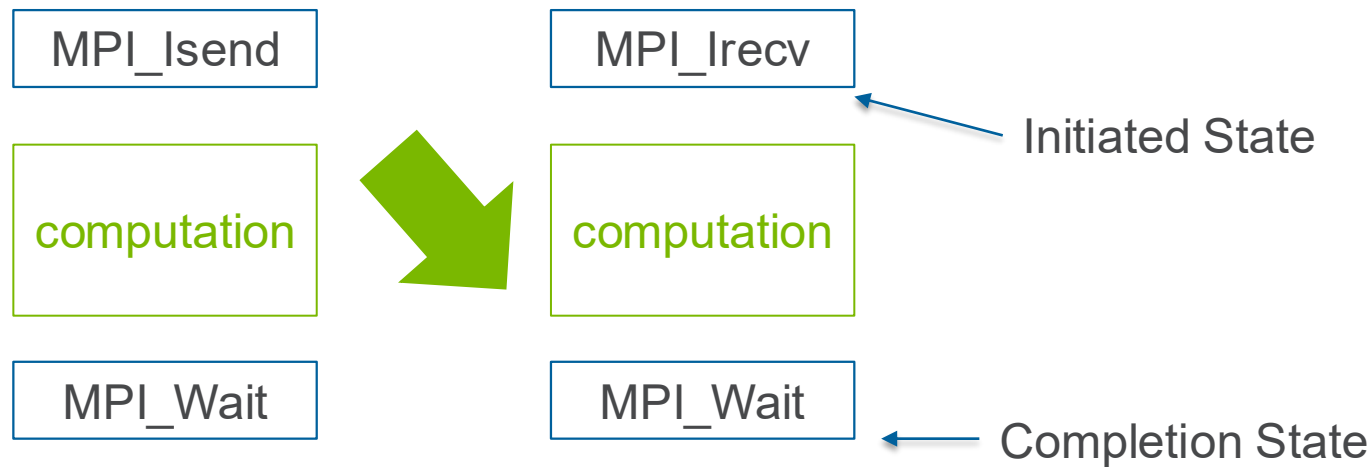
MPI IS ABSTRACT

- MPI's success is attributable to its abstraction and portability
- The blocking MPI_Send / MPI_Recv is a good example:



Extending MPI by Progressive **Exposure** of Internal States

- Blocking operation cannot express concurrency
- Nonblocking extension enables the expression of concurrency
- Communication can happen concurrently to processors and overlapping is critical for performance



The principle: Abstract by default, Extend to break abstraction by necessity

Message Mode



eager



rendezvous

MPI_Bsend
MPI_Rsend
MPI_Ssend

when: impl makes
wrong choice

Persistent Requests

preparations

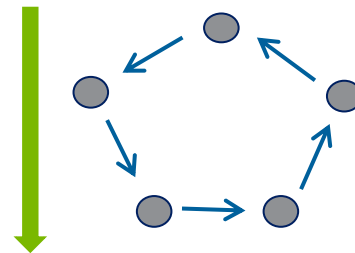
Initialized State

Active State

Completion State

when: request is
expensive

Nonblocking Collectives



when: BSP is
unbalanced

PRINCIPLE #3

MPI NEEDS TO BE COMPLEMENTARY

- MPI is the de facto runtime for HPC
- But alternative runtimes exist. MPI can either support them or be challenged by them.
- Supporting alt. runtime is conflicting to the prev. principles.
- To assume the role as a runtime foundation, MPI need to
 - Bootstrap alternative runtimes
 - Be interoperable, do not compromise
 - Be complementary
 - Overcome “*not invented here*” syndrome
- *Vice versa, advice for alternative runtimes:*
 - *MPI has a lot to offer, and it is mature. Leverage MPI. Don't always try to replace MPI and then end up reinventing MPI.*

MPI+Threads

- 🚧 Users unable to tell MPI about threads
- 🚧 Unnecessary synchronization compromises concurrency
- 🌟 Expose interoperable streams

Zhou, Hui, Ken Raffenetti, Yanfei Guo, and Rajeev Thakur. "MPIX stream: an explicit solution to hybrid MPI+ X programming." In *Proceedings of the 29th European MPI Users' Group Meeting*, pp. 1-10. 2022.

MPI Progress

- 🚧 Tasks depend on progress engine manage async operations
- 🚧 MPI progress is opaque
- 🚧 Managing MPI requests adds complexity and overhead
- 🚧 Unable to centralize progress affects performance
- 🌟 Expose interoperable MPI progress

Zhou, Hui, Robert Latham, Ken Raffenetti, Yanfei Guo, and Rajeev Thakur. "MPI progress for all." In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 425-435. IEEE, 2024.

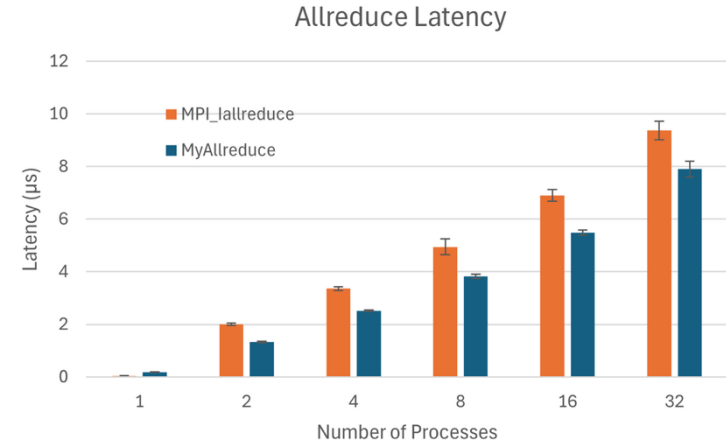
MPI Datatypes

- 🚧 MPI derived datatypes are an efficient high-level data layout abstraction but inaccessible to the user
- 🚧 The power is constrained by implementation
- 🚧 Users are unmotivated to use MPI datatypes
- 🌟 Expose interoperable datatype access via IOVEC

Zhou, Hui, Ken Raffenetti, Yanfei Guo, Thomas Gillis, Robert Latham, and Rajeev Thakur. "Designing and prototyping extensions to the Message Passing Interface in MPICH." *The International Journal of High Performance Computing Applications* 38, no. 5 (2024): 527-545.

- This last principle, interoperability, is the key to sustain MPI into the future.
- As the computing architecture and scientific applications shift, there will be more and more new design patterns and critical optimizations that MPI will struggle to support. It is unscalable if we depend on MPI to keep growing to support the future.
- With interoperability, MPI serve as a foundation runtime, we support more innovations, diversity, and competitions, and we spread the development burden of sustaining the future.

- Collective is the native expression for MPI operations
- But relying on MPI implementations is not scalable and sustainable
 - Limits on operation types
 - Limits on specific optimizations
 - Burdens on MPI developers
- By providing sufficient interoperability, MPI can support user-defined collective algorithms on top of MPI that matches native performance!



Zhou, Hui, Robert Latham, Ken Raffanetti, Yanfei Guo, and Rajeev Thakur. "MPI progress for all." In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 425-435. IEEE, 2024.

MPI CORE SET



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



A STRAWMAN CORE SET

3.1 Communicator Management

Init/Finalize

- MPI_Init, MPI_Finalize

Predefined Communicators

- MPI_COMM_WORLD, MPI_COMM_SELF

Constructors and Destructors

- MPI_Comm_dup
- MPI_Comm_split, MPI_Comm_split_type
- MPI_Comm_free

Queries

- MPI_Comm_size, MPI_Comm_rank

3.2 MPI Datatypes

Built-in Datatypes

- Integer: MPI_CHAR, MPI_SHORT, MPI_INT, MPI_LONG,
MPI_LONG_LONG_INT, MPI_SIGNED_CHAR,
MPI_UNSIGNED_CHAR,
- MPI_UNSIGNED_SHORT, MPI_UNSIGNED, MPI_UNSIGNED_LONG,
MPI_UNSIGNED_LONG_LONG
- Floating point: MPI_FLOAT, MPI_DOUBLE,
MPI_LONG_DOUBLE
- Byte: MPI_BYTE
- Auxiliary: MPI_PACKED

Derived Constructors

- MPI_Type_contiguous, MPI_Type_vector,
- MPI_Type_indexed, MPI_Type_create_struct,
- MPI_Type_create_resized
- MPI_Type_commit, MPI_Type_free

Queries

- MPI_Type_get_extent, MPI_Type_get_true_extent,
- MPI_Type_size

A STRAWMAN CORE SET – CONT.

3.3 Point-to-Point Communication

Blocking

- MPI_Send, MPI_Recv

Nonblocking

- MPI_Isend, MPI_Irecv

Completion

- MPI_Wait, MPI_Test, MPI_Waitall, MPI_Testall
- MPI_Request_free

3.4 Collective Operations

Synchronization and Data Movement

- MPI_Barrier
- MPI_Bcast, MPI_Reduce
- MPI_Gather, MPI_Scatter
- MPI_Allgather, MPI_Allreduce, MPI_Alltoall

Reduction Operators

- MPI_MAX, MPI_MIN, MPI_SUM, MPI_PROD, MPI_LAND, MPI_LOR, MPI_LXOR, MPI_BAND, MPI_BOR, MPI_BXOR
- MPI_Op_create, MPI_Op_free

3.5 Runtime Foundation Interfaces

Asynchronous Operations

- MPI_Request_start
- MPI_Wait
- MPI_Test
- MPI_Request_complete

Execution Context Interface

- MPIX_Stream_create
- MPIX_Stream_free

Progress Interface

- MPIX_Stream_progress

Datatype Exposure Interface

- MPI_Type_iov

MPI PROPOSAL CASE STUDIES

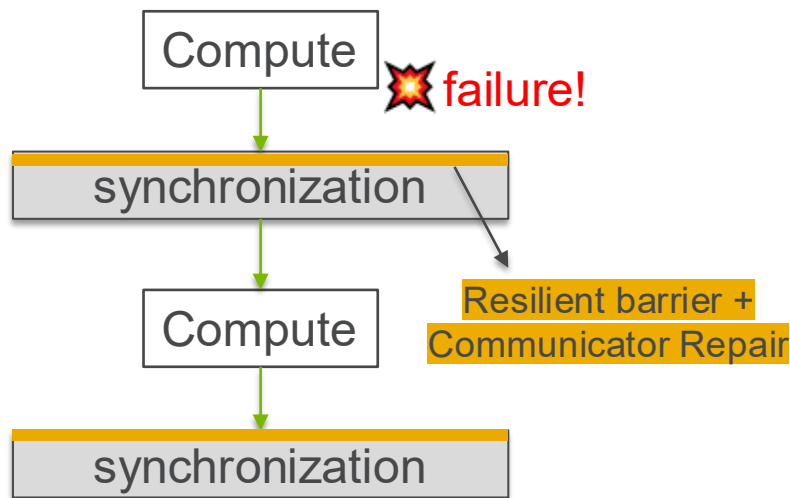


Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.



USER-LEVEL FAILURE MITIGATION (ULFM)

- Easily acceptable:
ULFM as a perturbation



- Difficult to impossible:
 - Universal failure model
 - General resilient operations

- Unexpected failures (🧑) break the assumed collective agreement
- Robust and performant resilient operations are impossible

Losada, Nuria, Patricia González, María J. Martín, George Bosilca, Aurélien Bouteiller, and Keita Teranishi. "Fault tolerance of MPI applications in exascale systems: The ULFM solution." *Future Generation Computer Systems* 106 (2020): 467-481.

STREAM-TRIGGERED MPI COMMUNICATION

- **Goal:** CPU-free GPU synchronization
- Interoperable extension: GPU stream
- Abstraction extension: Matching Stage

Difficulties:

- Too specific
- Too experimental

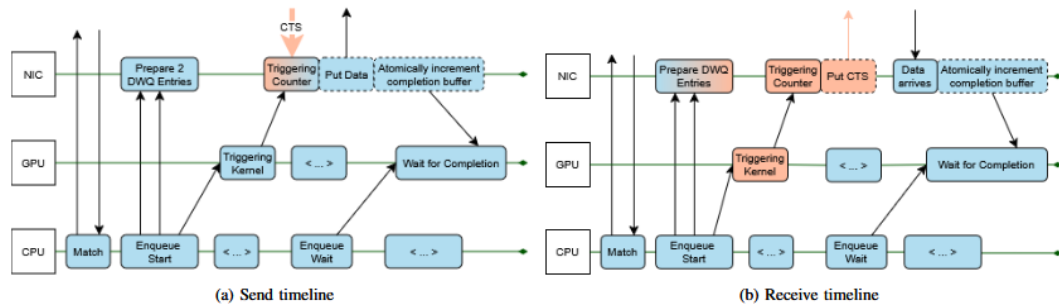


Fig. 4: Timeline for our stream-triggered implementation of send, ready send, and receive operations with respect to the CPU, GPU, and NIC. Dotted boxes represent deferred work queue operations being carried out; orange boxes represent actions specific to the implementation of regular (non-ready) send, as described in Section [V-B](#).

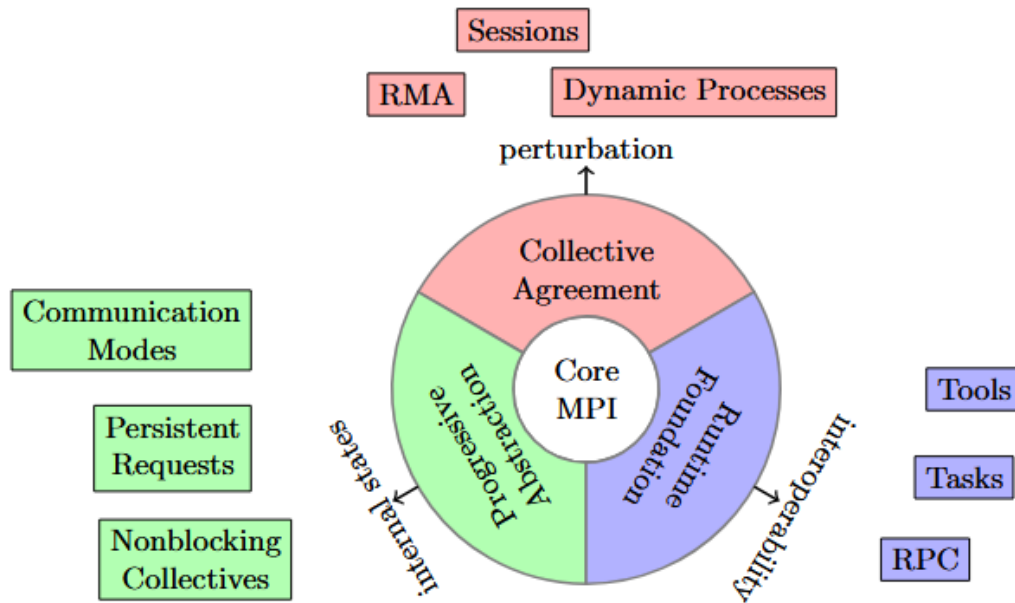
2nd thoughts:

- Where is the urgency?
- Why not MPI+NCCL?
- Need interop over stream and network

Bridges, Patrick G., Derek Schafer, Jack Lange, James B. White III, Anthony Skjellum, Evan Suggs, Thomas Hines, Purushotham Bangalore, Matthew GF Dosanjh, and Whit Schonbein. "Co-Design and Evaluation of a CPU-Free MPI GPU Communication Abstraction and Implementation." *arXiv preprint arXiv:2602.15356* (2026).

IN SUMMARY

The Core MPI Structure



Q & A



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