



University of Stuttgart

Germany

IPVS, Scientific Computing



Hierarchical Collective Operations for HPX

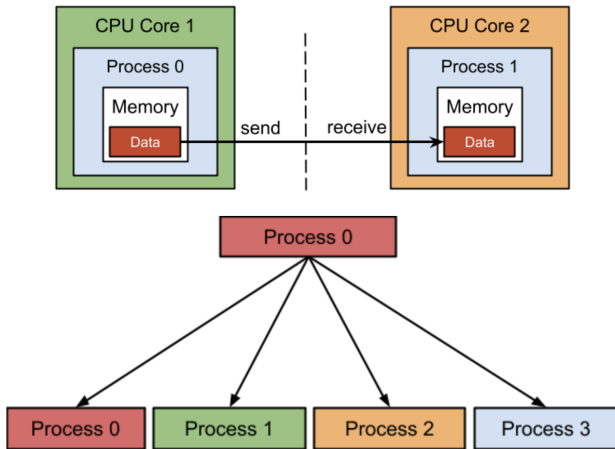
November 7th, 2025

Alexander Strack,
Lukas Zeil,
Hartmut Kaiser,
and Dirk Pflüger

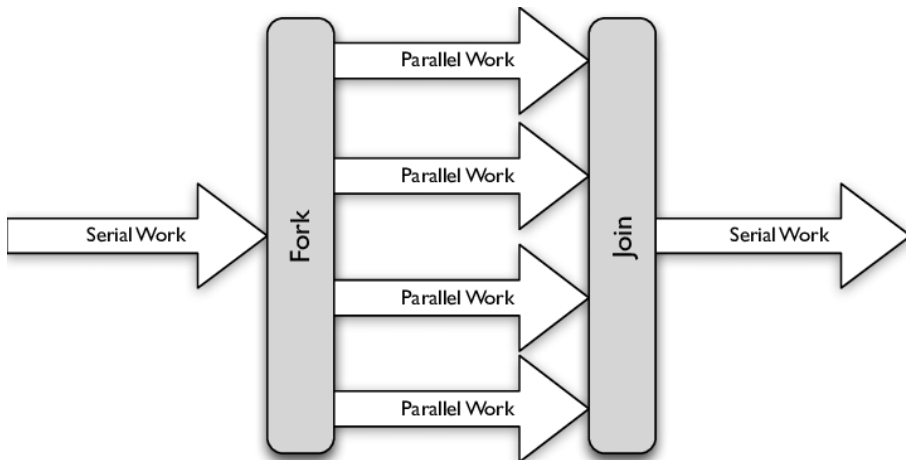


Motivation

MPI

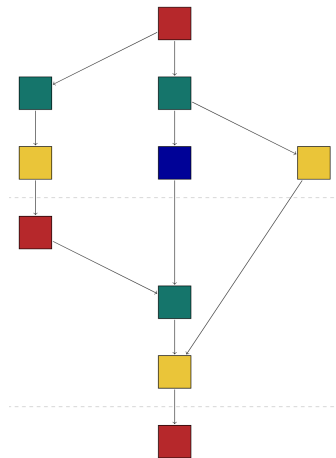


OpenMP

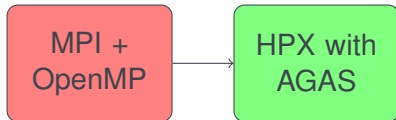


HPX

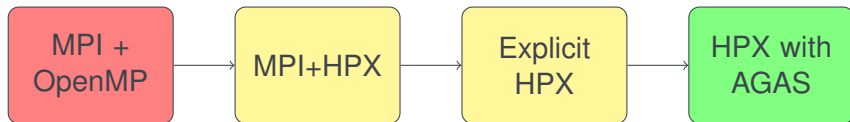
- C++ Standard Library for Concurrency and Parallelism
- Task futurization allows building asynchronous task graphs
- Lightweight threads, dataflow-based dependency management, and executors to target various hardware
- Implicit communication using an active global address space (AGAS)
- Communication abstraction with parcelports such as TCP, MPI, and LCI



From MPI+OpenMP to HPX



From MPI+OpenMP to HPX



From OpenMP to HPX

OpenMP parallel loop:

```
1 #pragma omp parallel for
2 for (int i = 0; i < n; ++i) {
3     // loop body
4 }
```


From OpenMP to HPX

OpenMP parallel loop:

```
1 #pragma omp parallel for
2 for (int i = 0; i < n; ++i) {
3     // loop body
4 }
```

HPX parallel loop:

```
1 hpx::experimental::for_loop(hpx::execution::par, 0, n, [&](int i) {
2     // loop body
3 });
```

From OpenMP to HPX

OpenMP private variables:

```
1 int x = 0;
2 #pragma omp parallel for private(x)
3 for (int i = 0; i < n; ++i) {
4     // loop body
5 }
```

From OpenMP to HPX

OpenMP private variables:

```
1 int x = 0;
2 #pragma omp parallel for private(x)
3 for (int i = 0; i < n; ++i) {
4     // loop body
5 }
```

HPX private variables:

```
1 hpx::experimental::for_loop(hpx::execution::par, 0, n, [&](int i) {
2     int x = 0; // Declare 'x' inside the loop body
3     // loop body
4 });
```

From OpenMP to HPX

OpenMP shared variables:

```
1 int x = 0;
2 #pragma omp parallel for shared(x)
3 for (int i = 0; i < n; ++i) {
4     // loop body
5 }
```

From OpenMP to HPX

OpenMP shared variables:

```
1 int x = 0;
2 #pragma omp parallel for shared(x)
3 for (int i = 0; i < n; ++i) {
4     // loop body
5 }
```

HPX shared variables:

```
1 std::atomic<int> x = 0; // Declare 'x' outside the loop
2 hpx::experimental::for_loop(hpx::execution::par, 0, n, [&](int i) {
3     // loop body
4 });
```

From MPI to HPX

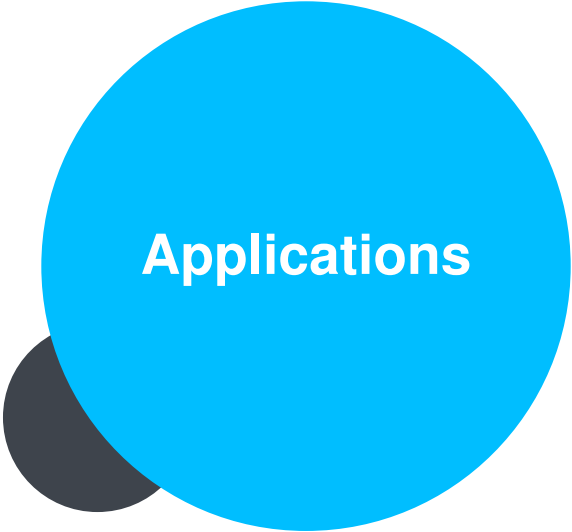
MPI Collectives:

```
1 // info
2 MPI_Comm_size(MPI_COMM_WORLD, &ranks);
3 MPI_Comm_rank(MPI_COMM_WORLD, &this_rank);
4 // scatter data from the root locality to all other localities
5 MPI_Scatter(&data[0], 1, MPI_INT, &local_data, 1, MPI_INT, 0,
             MPI_COMM_WORLD);
```

From MPI to HPX

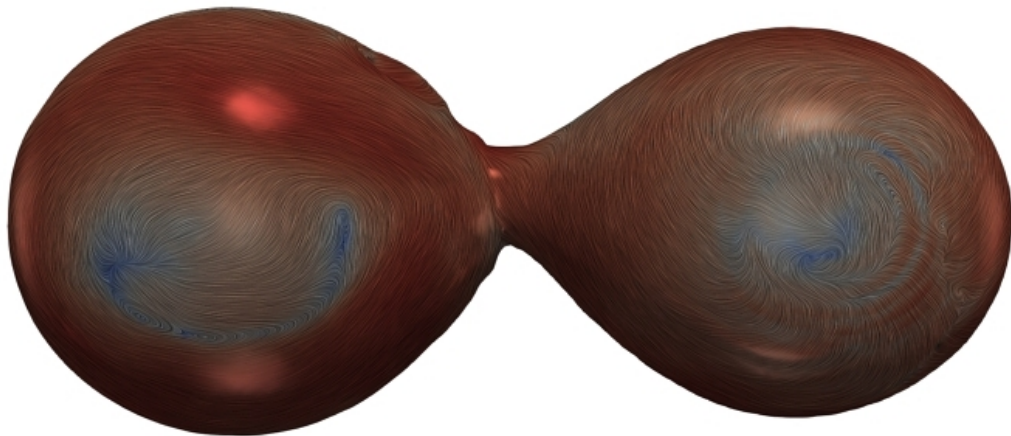
HPX Collectives:

```
1 // info
2 std::uint32_t localities = hpx::get_num_localities();
3 std::uint32_t this_locality = hpx::get_locality_id();
4 // setup communicator
5 auto scatter_direct_client = hpx::collectives::create_communicator(
    scatter_direct_basename, num_sites_arg(num_localities),
    this_site_arg(this_locality));
6 // scatter data from the root process to all other processes
7 if (this_locality == 0){
8     auto result = scatter_to(scatter_direct_client, std::move(data)).
        get();
9 }else{
10     auto result = scatter_from<T>(scatter_direct_client).get();
11 }
```

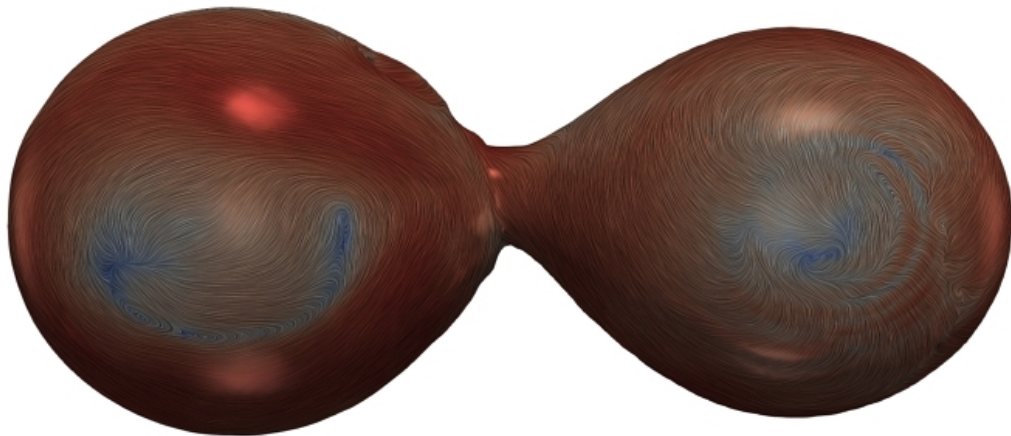


Applications

Octo-Tiger: Astrophysics

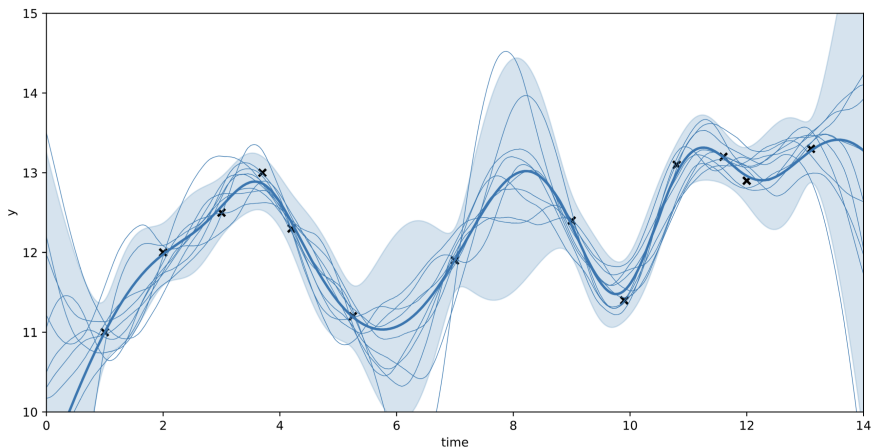


Octo-Tiger: Astrophysics

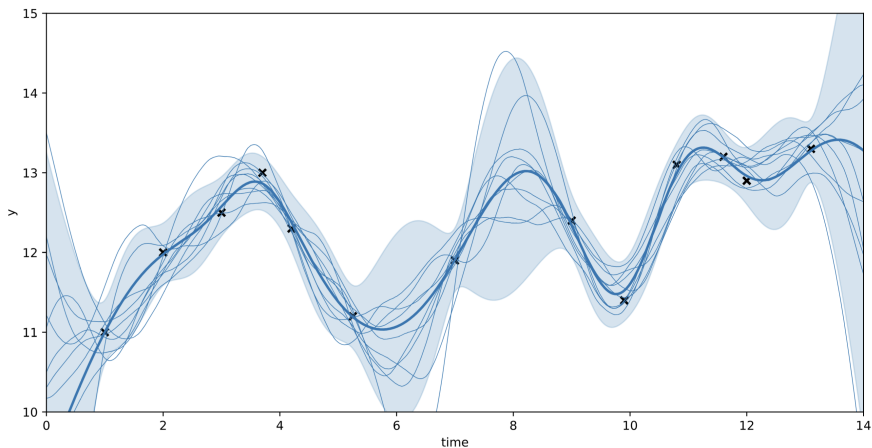


→ HPX with AGAS

GP_{Rat}: Gaussian Process Regression

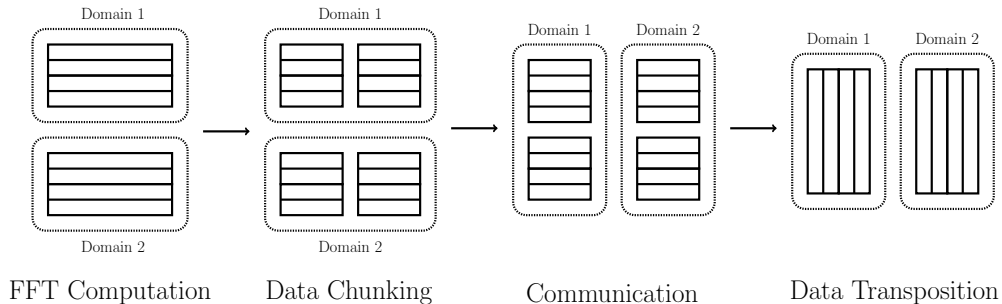


GP_{Rat}: Gaussian Process Regression

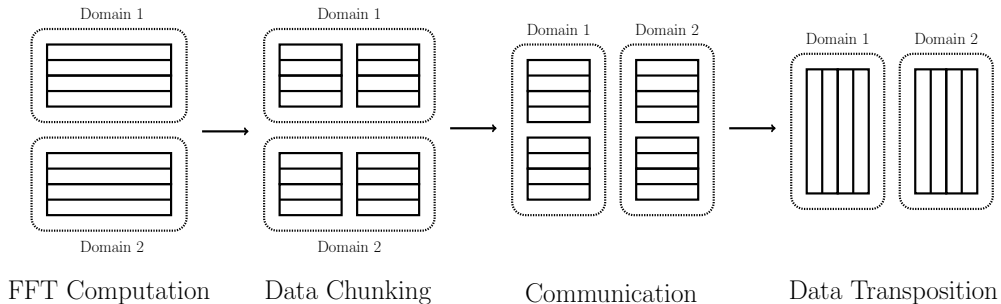


→ HPX with AGAS

HPX-FFT: Distributed multi-dimensional FFT

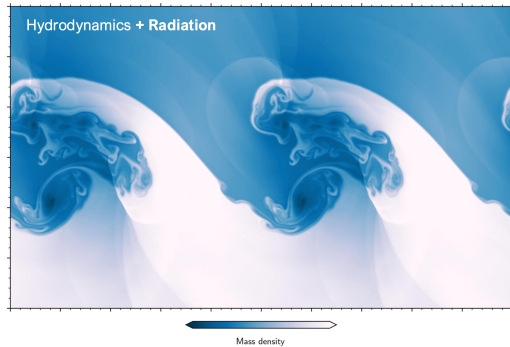
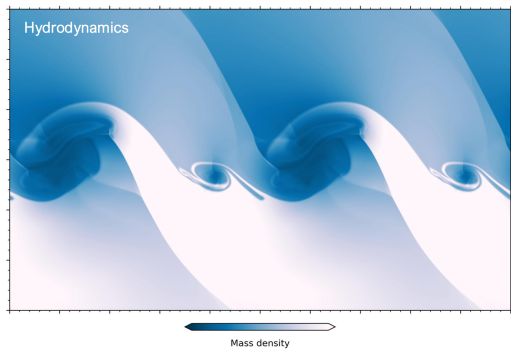


HPX-FFT: Distributed multi-dimensional FFT

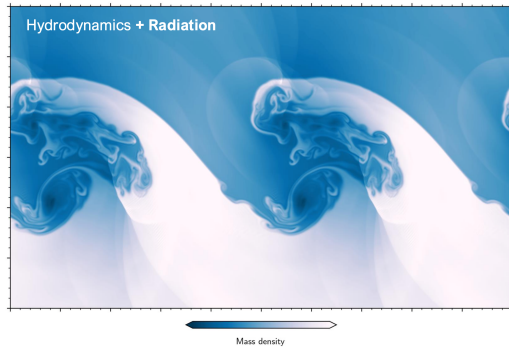
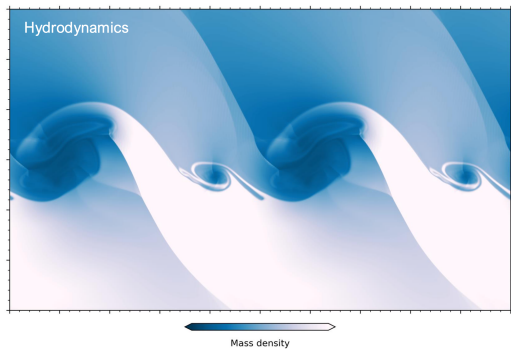


→ Explicit HPX

FleCSI-HARD: Radiation Hydrodynamics

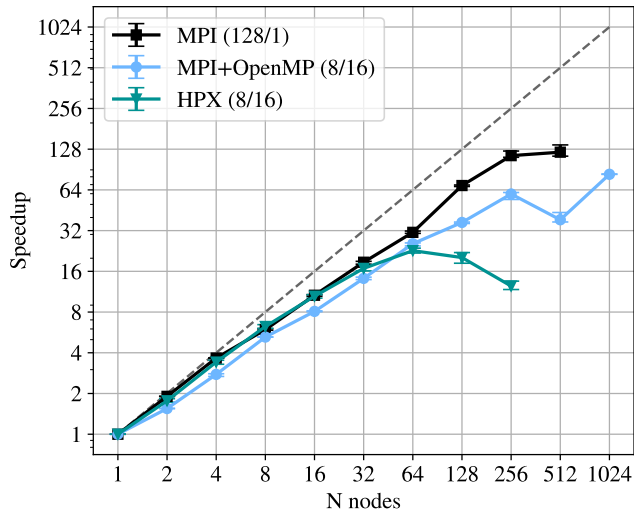


FleCSI-HARD: Radiation Hydrodynamics

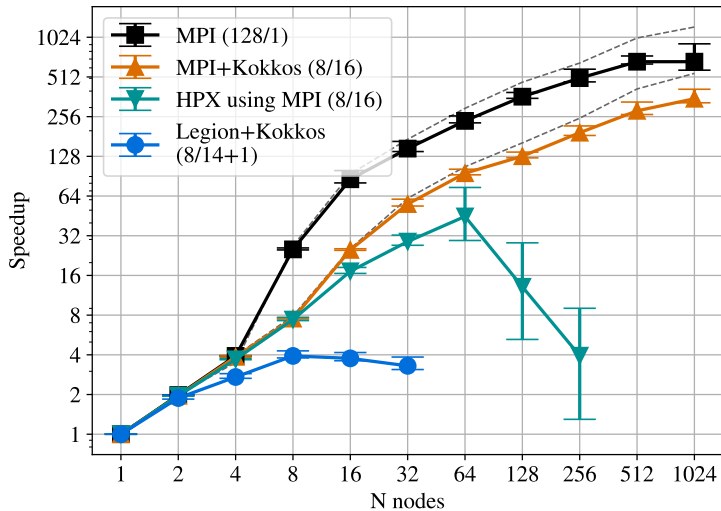


→ Explicit HPX

FleCSI-HARD Strong Scaling



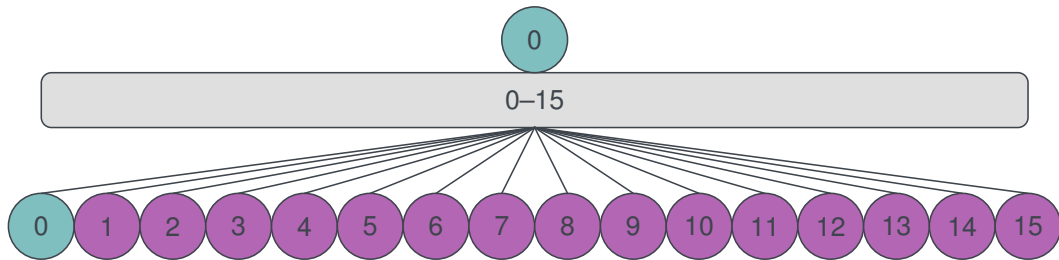
FleCSI-Poisson Strong Scaling



The logo consists of two overlapping circles. The larger circle is bright blue and contains the text 'HPX Collectives' in white. The smaller circle, positioned behind and to the left of the larger one, is a dark charcoal grey.

HPX Collectives

Existing HPX Communicator



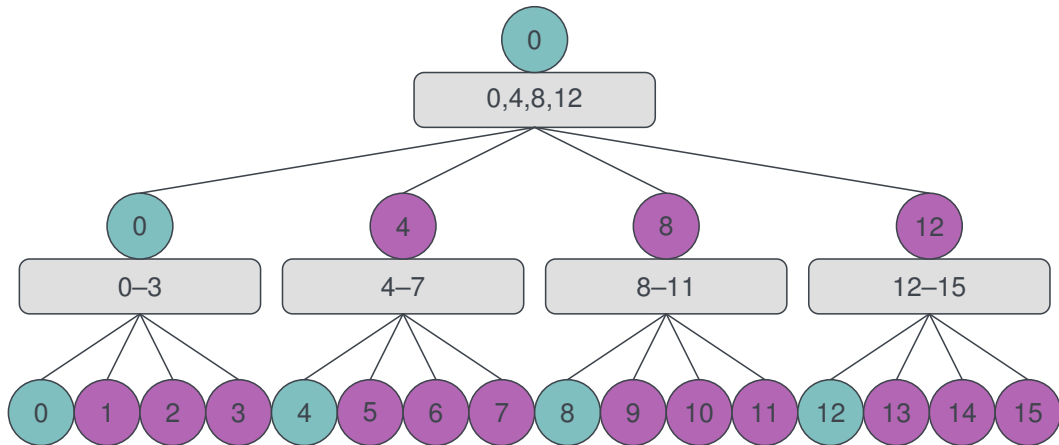
Approaches

- Wrap HPX collectives directly around MPI collectives
- Write a new communicator from scratch
- Write hierarchical communicator relying on existing communicator

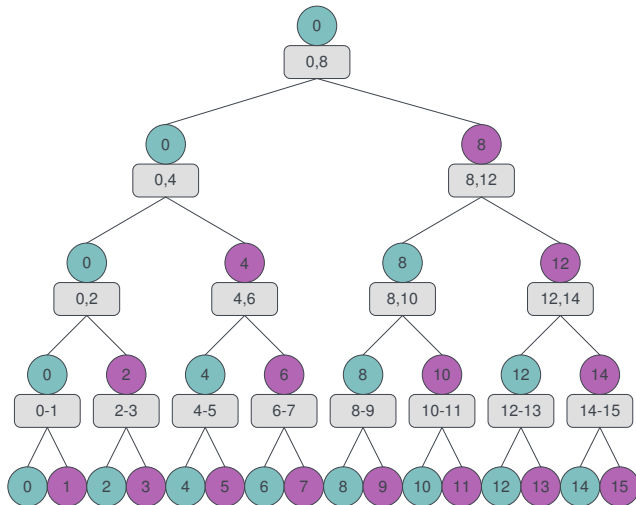
Approaches

- Wrap HPX collectives directly around MPI collectives
- Write a new communicator from scratch
- **Write hierarchical communicator relying on existing communicator**

Hierarchical HPX Communicator with Arity 4



Hierarchical HPX Communicator with Arity 2



HPX Collectives Overview

- Broadcast
- Reduce
- Scatter
- Gather
- All Reduce
- All Gather
- All to All

HPX Collectives Overview

- **Broadcast**
- **Reduce**
- **Scatter**
- **Gather**
- All Reduce
- All Gather
- All to All

HPX Collectives Overview

- **Broadcast**
- **Reduce**
- **Scatter**
- **Gather**
- All Reduce
- All Gather
- All to All



Results

Benchmark

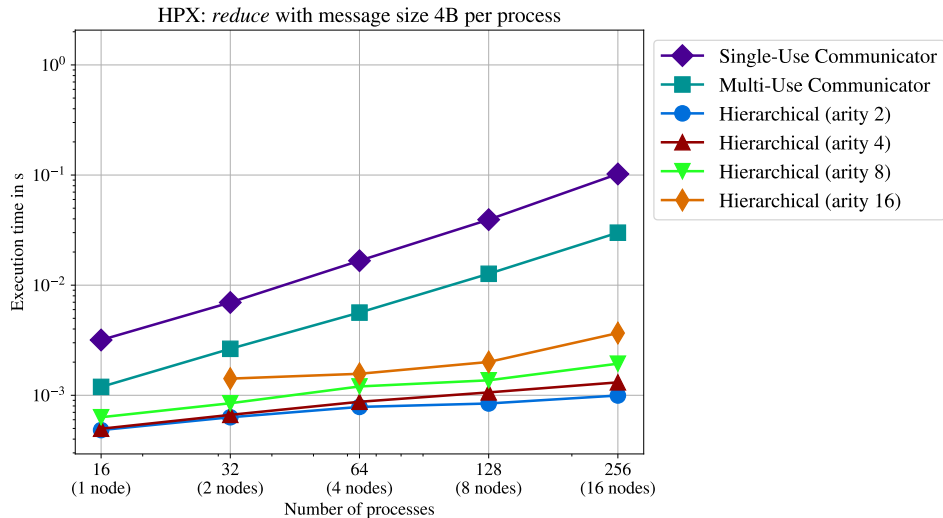
- OpenMPI with tuned MPI collectives
- MPI Parcelport for HPX
- Runtimes averaged over 100 collectives
- 16 processes per node

Cluster	medusa
Nodes	16
Connection	InfiniBand FDR
Speed	54Gb/s
Sockets	2
CPU	Intel Xeon Gold 6148
Cores	40

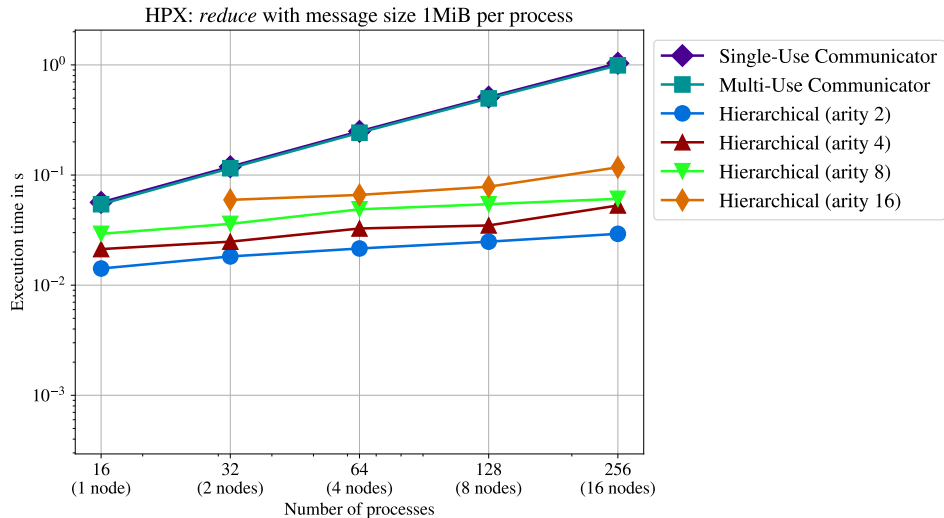


Arity

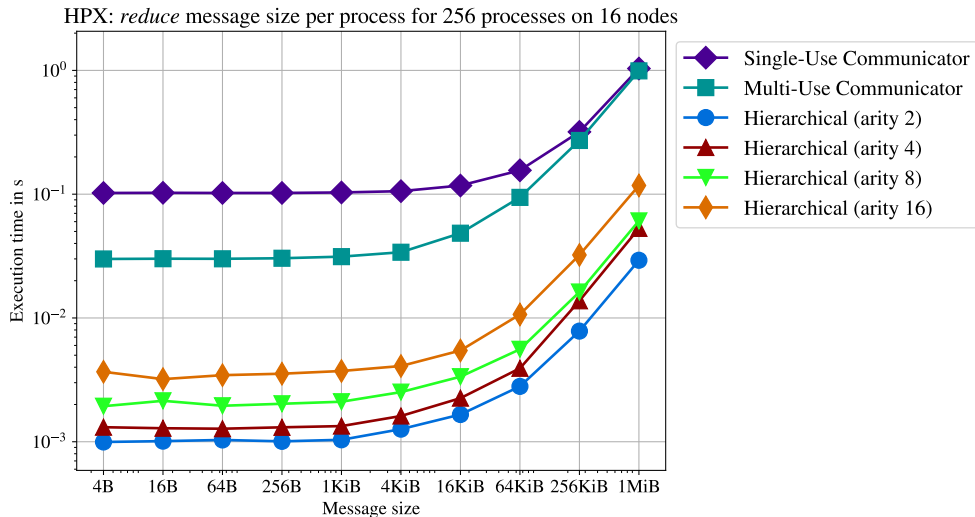
Reduce Process Scaling

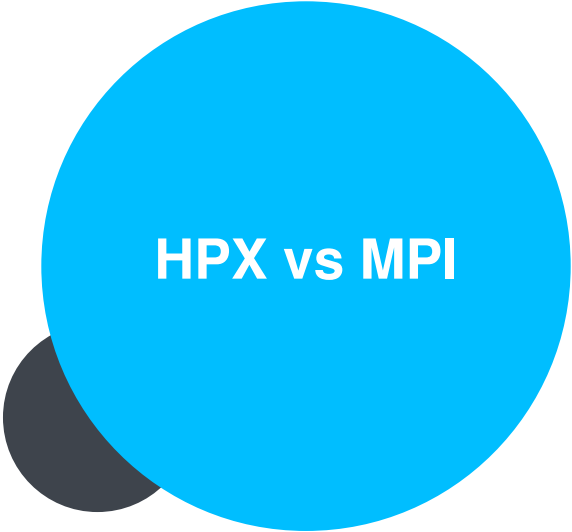


Reduce Process Scaling



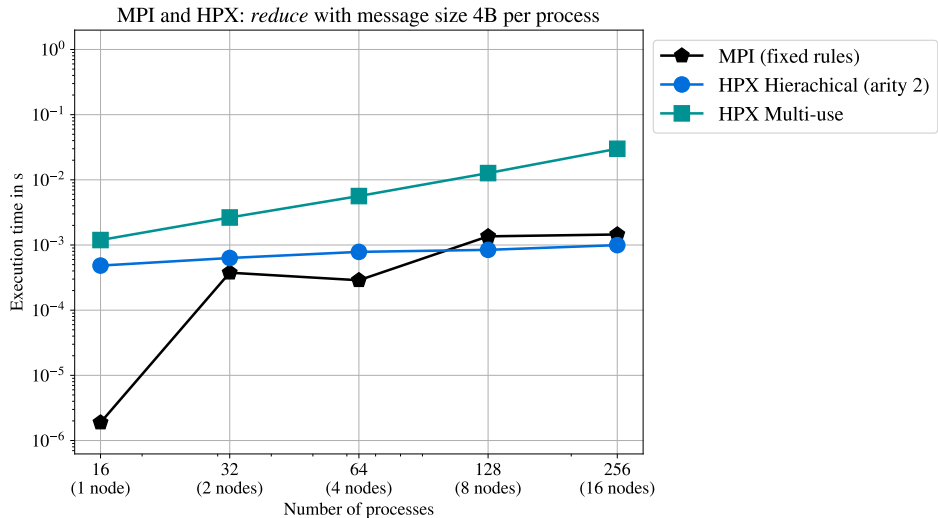
Reduce Message Size Scaling



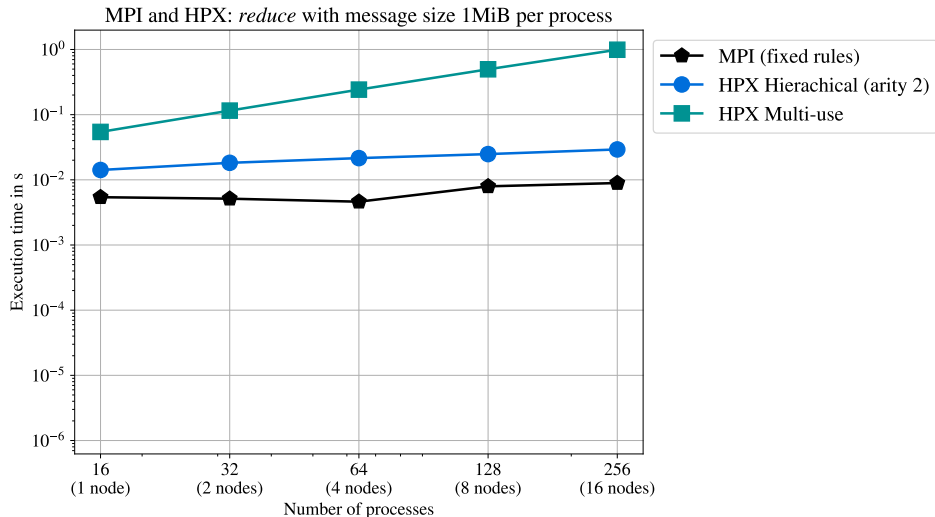


HPX vs MPI

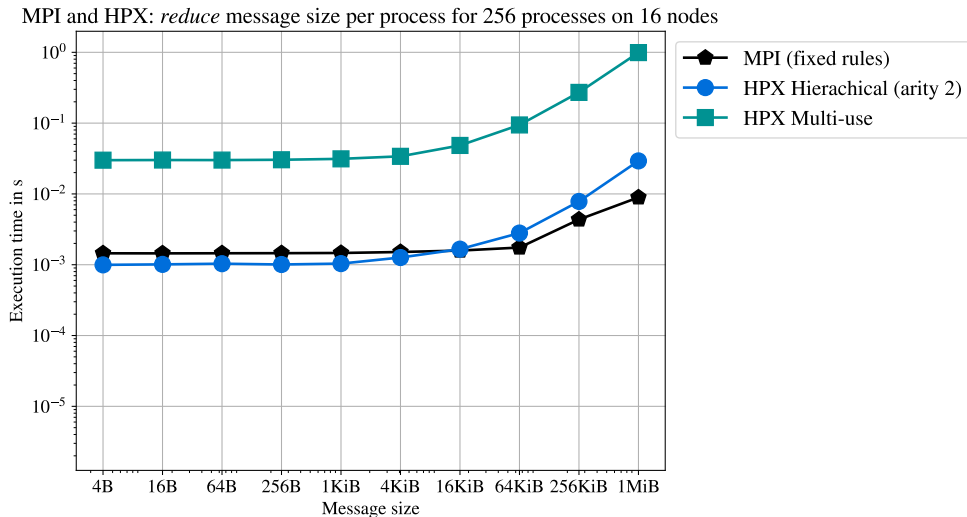
HPX Reduce vs MPI Reduce



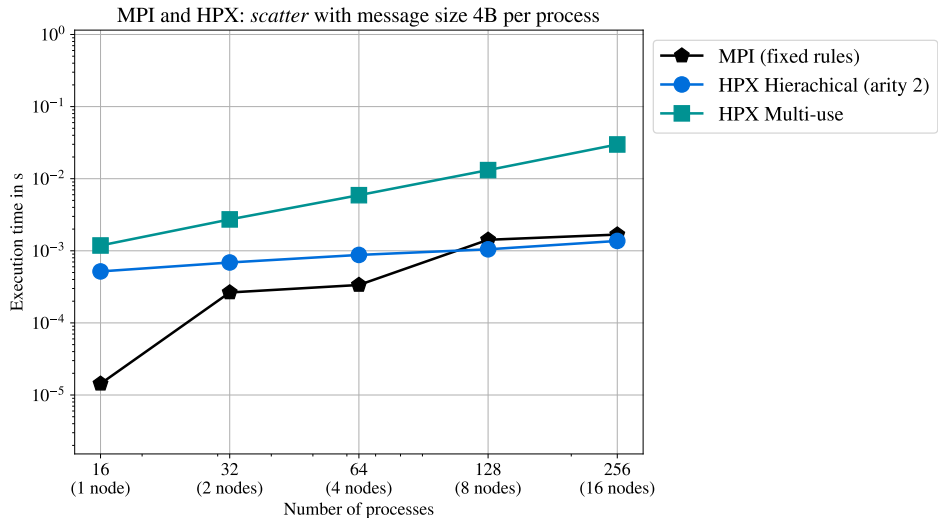
HPX Reduce vs MPI Reduce



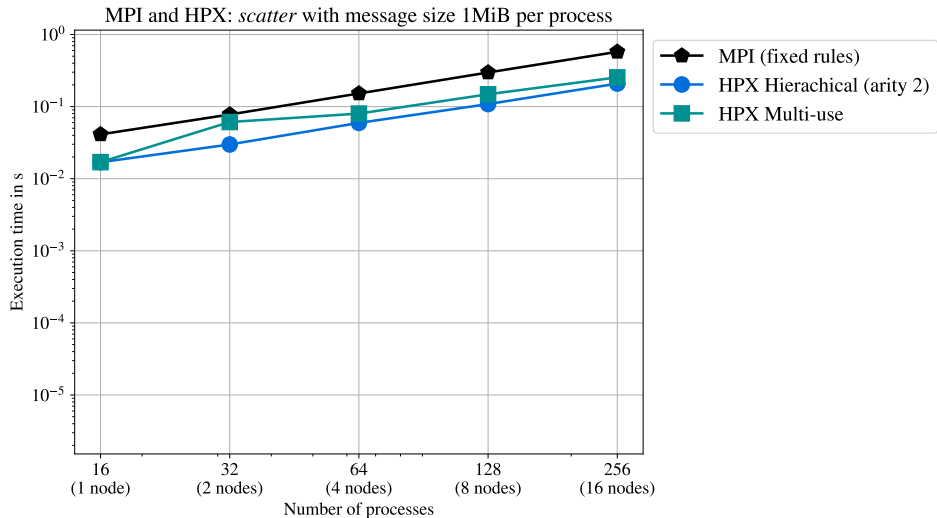
HPX Reduce vs MPI Reduce



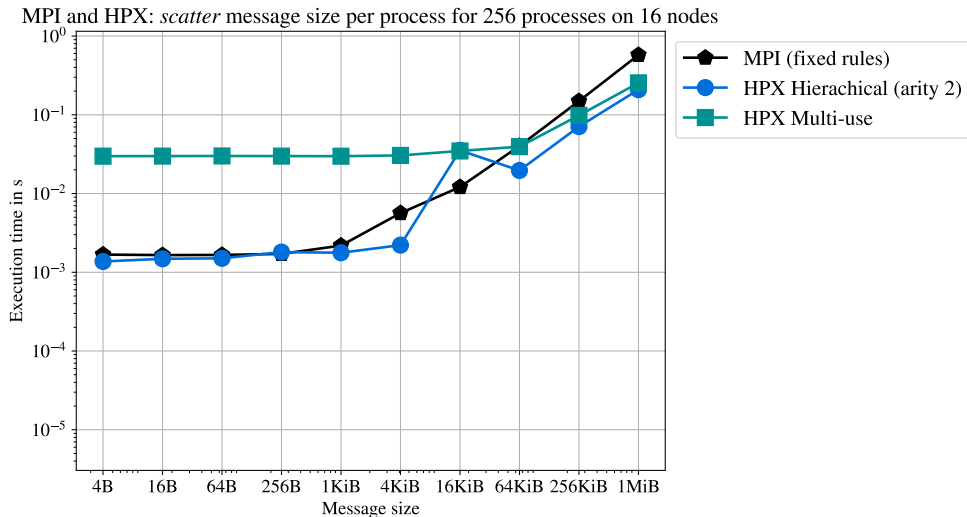
HPX Scatter vs MPI Scatter

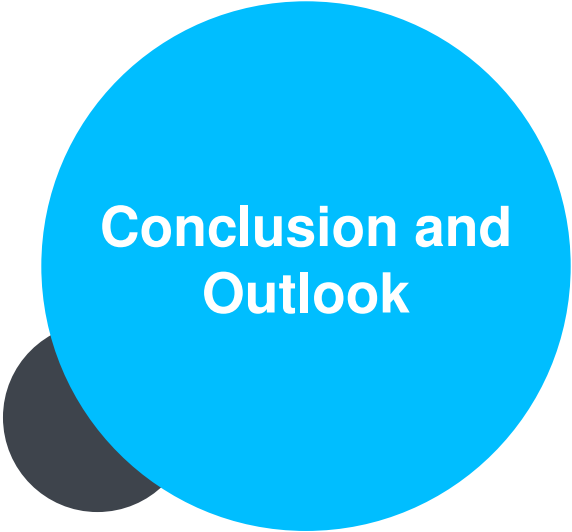


HPX Scatter vs MPI Scatter



HPX Scatter vs MPI Scatter





Conclusion and Outlook

Conclusion

- Existing HPX collectives can be a major bottleneck for applications
- Hierarchical HPX communicator significantly improves performance
- Preliminary results show similar scaling to MPI collectives
- Reduce and broadcast profit more than gather and scatter

Outlook

- Complete set of HPX collectives using the hierarchical communicator
- Performance evaluation using HPX-FFT and FleCSI-HARD
- Test the LCI parcelport of HPX
- Integration into upcoming HPX release



University of Stuttgart
Germany

Thank You!



Alexander Strack
IPVS

Mail	alexander.strack@ipvs.uni-stuttgart.de
Phone	+49 711 685 88308
Internet	https://www.ipvs.uni-stuttgart.de