MPI Overlap: Benchmark and Analysis

Alexandre Denis
Inria Bordeaux, France
Alexandre.Denis@inria.fr

François Trahay
CNRS Samovar, Télécom SudParis, Université Paris-Saclay, France
Francois.Trahay@telecom-sudparis.eu
MPI Overlap Benchmark
MPI Overlap

- **Amortize communication cost** through overlap
- Non-blocking MPI communications
  - **Post** a non-blocking communication request
  - Do some **computation**
  - **Wait** for request completion
  - Communication progressed in background
- **Application programmers assume communications progress** automagically
  - But **not guaranted** by MPI spec!
Communication Progression

- Communication may progress:
  - Only **inside MPI_* function** calls i.e. implementation is synchronous
    - Permitted by the MPI spec
    - No background progression
  - Using **NIC offloading**
    - Setup DMA, then let the hardware run the transfer
    - Limited to what the hardware can do
  - Using **threads**
    - pthreads, kernel tasklets, dedicated core, etc.
Features to benchmark

• **Exploration space**: 2D space (message size x computation time)
  - Local behavior may not represent global behavior
    • e.g. thresholds on message size (PIO/DMA/rendez-vous, cache effects),
      thresholds on time (fixed cost overhead, TCP timers, etc.)
  - **Sender side, receiver side, both sides**
    - Internal protocol (rendez-vous, RDMA write, etc.) is likely to exhibit non-symmetrical properties

• **Contiguous, non-contiguous datatype**
  - NIC offloading is ok for contiguous data blocks, but what about non-contiguous derived datatypes?

• **CPU overhead**
  - Measure CPU consumption used for communication progression

• **Multi-threaded computation**
  - Blocking communication in one thread, computation in other threads
Measure overlap

- Clock start/stop options:
  - $T_2 - T_0$: transfer time with overlap
    - Need global clock
  - Round-trip with overlap on both ways
    - Only for overlap on sender side
    - No way to synchronize computation start and receive
  - $T_1 - T_0$: send time
    - No guarantee data arrived – may have been buffered
    - Only sender side
  - $T_3 - T_0$: uncertainty on $T_3 - T_2$
Benchmarks

- Sender-side overlap
  - MPI_Isend
  - MPI_Wait
  - MPI_Recv

- Receiver-side overlap
  - MPI_Irecv
  - MPI_Send
  - MPI_Wait

- Both-sides overlap
  - MPI_Isend
  - MPI_Wait
  - MPI_Irecv
  - MPI_Wait

- Non-contiguous datatype, sender-side
  - MPI_Isend
  - MPI_Wait
  - MPI_Recv

- CPU overhead, sender-side
  - MPI_Isend
  - MPI_Wait
  - MPI_Recv

- N computing threads (funneled)
  - MPI_Send
  - MPI_Wait
  - MPI_Recv
Metrics and visualization

- **2D parameters** space
  - X axis: data size
  - Y axis: computation time
- **Overhead** ratio relative to perfect overlap
  - 0 : perfect overlap
  - 1 : no overlap, computation & communication serialized
  - >1 : slower than serialized

**Expected time if perfect overlap**

\[
\text{ratio} = \frac{\text{measured} - \max(\text{computation, comm})}{\min(\text{computation, comm})}
\]

Normalizing such as ratio=1 for no overlap

- Computation longer than communication
- Communication longer than computation

Communication time = computation time
Benchmark results
OpenMPI 1.10, ibverbs – sender-side overlap
OpenMPI 1.10, ibverbs – receiver-side overlap
OpenMPI 1.10, ibverbs – sender-side, non-contig overlap
MPICH 3.2 – send overlap, shm
OpenMPI Mosaic

<table>
<thead>
<tr>
<th>overlap_sender</th>
<th>overlap_recv</th>
<th>overlap_both</th>
<th>sender noncontig</th>
<th>send overhead</th>
<th>overlap_Nthread</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenMPI 1.8 ibverbs</td>
<td><img src="openmpi-1_8-ibverbs-inti-haswell.dat" alt="Image" /></td>
<td><img src="openmpi-1_8-ibverbs-inti-haswell.dat" alt="Image" /></td>
<td><img src="openmpi-1_8-ibverbs-inti-haswell.dat" alt="Image" /></td>
<td><img src="openmpi-1_8-ibverbs-inti-haswell.dat" alt="Image" /></td>
<td><img src="openmpi-1_8-ibverbs-inti-haswell.dat" alt="Image" /></td>
</tr>
<tr>
<td>OpenMPI 1.10 ibverbs</td>
<td><img src="openmpi-1_10-ibverbs-william.dat" alt="Image" /></td>
<td><img src="openmpi-1_10-ibverbs-william.dat" alt="Image" /></td>
<td><img src="openmpi-1_10-ibverbs-william.dat" alt="Image" /></td>
<td><img src="openmpi-1_10-ibverbs-william.dat" alt="Image" /></td>
<td><img src="openmpi-1_10-ibverbs-william.dat" alt="Image" /></td>
</tr>
<tr>
<td>OpenMPI 1.10 TCP/Ethernet</td>
<td><img src="openmpi-1_10-tcp-william.dat" alt="Image" /></td>
<td><img src="openmpi-1_10-tcp-william.dat" alt="Image" /></td>
<td><img src="openmpi-1_10-tcp-william.dat" alt="Image" /></td>
<td><img src="openmpi-1_10-tcp-william.dat" alt="Image" /></td>
<td><img src="openmpi-1_10-tcp-william.dat" alt="Image" /></td>
</tr>
<tr>
<td>OpenMPI 2.0 ibverbs</td>
<td><img src="openmpi-2_x-ibverbs-william.dat" alt="Image" /></td>
<td><img src="openmpi-2_x-ibverbs-william.dat" alt="Image" /></td>
<td><img src="openmpi-2_x-ibverbs-william.dat" alt="Image" /></td>
<td><img src="openmpi-2_x-ibverbs-william.dat" alt="Image" /></td>
<td><img src="openmpi-2_x-ibverbs-william.dat" alt="Image" /></td>
</tr>
<tr>
<td>OpenMPI 2.0 shm</td>
<td><img src="openmpi-2_x-shm-jack.dat" alt="Image" /></td>
<td><img src="openmpi-2_x-shm-jack.dat" alt="Image" /></td>
<td><img src="openmpi-2_x-shm-jack.dat" alt="Image" /></td>
<td><img src="openmpi-2_x-shm-jack.dat" alt="Image" /></td>
<td><img src="openmpi-2_x-shm-jack.dat" alt="Image" /></td>
</tr>
</tbody>
</table>
## MPICH & MVAPICH

<table>
<thead>
<tr>
<th></th>
<th>overlap_sender</th>
<th>overlap_recv</th>
<th>overlap_both</th>
<th>sender noncontig</th>
<th>send overhead</th>
<th>overlap_Nthread</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVAPICH 2.0</td>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
<td><img src="image3.png" alt="Graph" /></td>
<td><img src="image4.png" alt="Graph" /></td>
<td><img src="image5.png" alt="Graph" /></td>
<td><img src="image6.png" alt="Graph" /></td>
</tr>
<tr>
<td>ibverbs</td>
<td><img src="image7.png" alt="Graph" /></td>
<td><img src="image8.png" alt="Graph" /></td>
<td><img src="image9.png" alt="Graph" /></td>
<td><img src="image10.png" alt="Graph" /></td>
<td><img src="image11.png" alt="Graph" /></td>
<td><img src="image12.png" alt="Graph" /></td>
</tr>
<tr>
<td>MVAPICH 2.2</td>
<td><img src="image13.png" alt="Graph" /></td>
<td><img src="image14.png" alt="Graph" /></td>
<td><img src="image15.png" alt="Graph" /></td>
<td><img src="image16.png" alt="Graph" /></td>
<td><img src="image17.png" alt="Graph" /></td>
<td><img src="image18.png" alt="Graph" /></td>
</tr>
<tr>
<td>ibverbs</td>
<td><img src="image19.png" alt="Graph" /></td>
<td><img src="image20.png" alt="Graph" /></td>
<td><img src="image21.png" alt="Graph" /></td>
<td><img src="image22.png" alt="Graph" /></td>
<td><img src="image23.png" alt="Graph" /></td>
<td><img src="image24.png" alt="Graph" /></td>
</tr>
<tr>
<td>MPICH 3.2</td>
<td><img src="image25.png" alt="Graph" /></td>
<td><img src="image26.png" alt="Graph" /></td>
<td><img src="image27.png" alt="Graph" /></td>
<td><img src="image28.png" alt="Graph" /></td>
<td><img src="image29.png" alt="Graph" /></td>
<td><img src="image30.png" alt="Graph" /></td>
</tr>
<tr>
<td>TCP/Ethernet</td>
<td><img src="image31.png" alt="Graph" /></td>
<td><img src="image32.png" alt="Graph" /></td>
<td><img src="image33.png" alt="Graph" /></td>
<td><img src="image34.png" alt="Graph" /></td>
<td><img src="image35.png" alt="Graph" /></td>
<td><img src="image36.png" alt="Graph" /></td>
</tr>
<tr>
<td>MPICH 3.2</td>
<td><img src="image37.png" alt="Graph" /></td>
<td><img src="image38.png" alt="Graph" /></td>
<td><img src="image39.png" alt="Graph" /></td>
<td><img src="image40.png" alt="Graph" /></td>
<td><img src="image41.png" alt="Graph" /></td>
<td><img src="image42.png" alt="Graph" /></td>
</tr>
<tr>
<td>shm</td>
<td><img src="image43.png" alt="Graph" /></td>
<td><img src="image44.png" alt="Graph" /></td>
<td><img src="image45.png" alt="Graph" /></td>
<td><img src="image46.png" alt="Graph" /></td>
<td><img src="image47.png" alt="Graph" /></td>
<td><img src="image48.png" alt="Graph" /></td>
</tr>
</tbody>
</table>
Supercomputers

overlap_sender  overlap_recv  overlap_both  sender noncontig  send overhead  overlap_Nthread

bluegene-juqueen-default.dat
IBM BluGene/Q (default)

bluegene-juqueen-threadmultiple.dat
IBM BluGene/Q (thread multiple)

bluewaters-gemini.dat
Cray Gemini Bluewaters

bluewaters-shm.dat
Cray shm Bluewaters

fujitsu-tofu-K.dat
Fujitsu Tofu K Computer
### Latest additions (not in the paper)

<table>
<thead>
<tr>
<th>Overlap</th>
<th>OpenMPI 2.0</th>
<th>IB EDR</th>
<th>ibverbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>overlap_sender</td>
<td><img src="overlap-openmpi-2-0-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_recv</td>
<td><img src="overlap-openmpi-2-0-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_both</td>
<td><img src="overlap-openmpi-2-0-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sender noncontig</td>
<td><img src="overlap-openmpi-2-0-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>send overhead</td>
<td><img src="overlap-openmpi-2-0-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_Nthread</td>
<td><img src="overlap-openmpi-2-0-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overlap</th>
<th>MVAPICH 2.2</th>
<th>IB EDR</th>
<th>ibverbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>overlap_sender</td>
<td><img src="overlap-mvapich-2-2-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_recv</td>
<td><img src="overlap-mvapich-2-2-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_both</td>
<td><img src="overlap-mvapich-2-2-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sender noncontig</td>
<td><img src="overlap-mvapich-2-2-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>send overhead</td>
<td><img src="overlap-mvapich-2-2-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_Nthread</td>
<td><img src="overlap-mvapich-2-2-ibverbs-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overlap</th>
<th>OpenMPI 1.10</th>
<th>IB FDR</th>
<th>Mellanox MXM</th>
</tr>
</thead>
<tbody>
<tr>
<td>overlap_sender</td>
<td><img src="overlap-openmpi-1-10-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_recv</td>
<td><img src="overlap-openmpi-1-10-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_both</td>
<td><img src="overlap-openmpi-1-10-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sender noncontig</td>
<td><img src="overlap-openmpi-1-10-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>send overhead</td>
<td><img src="overlap-openmpi-1-10-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_Nthread</td>
<td><img src="overlap-openmpi-1-10-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overlap</th>
<th>MPICH 3.2</th>
<th>IB FDR</th>
<th>Mellanox MXM</th>
</tr>
</thead>
<tbody>
<tr>
<td>overlap_sender</td>
<td><img src="overlap-mpich-3-2-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_recv</td>
<td><img src="overlap-mpich-3-2-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_both</td>
<td><img src="overlap-mpich-3-2-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sender noncontig</td>
<td><img src="overlap-mpich-3-2-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>send overhead</td>
<td><img src="overlap-mpich-3-2-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overlap_Nthread</td>
<td><img src="overlap-mpich-3-2-mxm-newpie.dat" alt="" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overlap</th>
<th>Intel MPI 5.1</th>
<th>Omni-Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>overlap_sender</td>
<td><img src="overlap-intelmpi-5-1-newpie.dat" alt="" /></td>
<td></td>
</tr>
<tr>
<td>overlap_recv</td>
<td><img src="overlap-intelmpi-5-1-newpie.dat" alt="" /></td>
<td></td>
</tr>
<tr>
<td>overlap_both</td>
<td><img src="overlap-intelmpi-5-1-newpie.dat" alt="" /></td>
<td></td>
</tr>
<tr>
<td>sender noncontig</td>
<td><img src="overlap-intelmpi-5-1-newpie.dat" alt="" /></td>
<td></td>
</tr>
<tr>
<td>send overhead</td>
<td><img src="overlap-intelmpi-5-1-newpie.dat" alt="" /></td>
<td></td>
</tr>
<tr>
<td>overlap_Nthread</td>
<td><img src="overlap-intelmpi-5-1-newpie.dat" alt="" /></td>
<td></td>
</tr>
</tbody>
</table>
3 Results analysis
Classification of results

- Threaded/non-threaded library
  - **Threaded libraries** (MadMPI, IBM MPI BluGene/Q): good overlap
  - Threaded computation + blocking communication: (mostly) good overlap
- Rendez-vous threshold
  - **Large messages** need CPU to process rendez-vous request; hard to overlap
- TCP
  - Kernel threads makes communication progress
- Shared memory
  - CPU-hungry, hard to overlap, often slower than serialized
- Non-contiguous datatypes
  - Needs CPU to gather/scatter data; hard to overlap
Interleaved MPI and InfiniBand traces

- Trace functions calls to MPI and ibverbs (low-level network driver)
  - Observe how the MPI library makes communication progress
- **EZTrace** trace generator & **ViTE** trace visualizer
  - Use builtin `mpi` plugin and specially developed `ibverbs` plugin
  - **Interactive exploration** needed to get details (zoom, event properties)
    - Hard to read on screenshots, and not enough time for a demo

![Diagram showing MPI and InfiniBand traces]

- Black bullet: `ibv_post`
- Pink bullet: `ibv_poll`
- White block: computation
Pathological case traces

- Sender-side overlap: sender notices RTR (rendez-vous reply) after computation

- Receiver-side overlap: receiver notices rendez-vous request after computation

- Non-contiguous, sender-side overlap: data sent in 64KB chunks, only first chunk is overlaped

- Sender-side CPU overhead: MPI_Isend has a fixed, non-overlaped cost (2 usec.)
Conclusion

- Amortize communication cost through MPI overlap
- Contributions:
  - **Benchmark suite** to assess actual MPI overlap
  - **Results** on large panel of MPI libraries & machines
  - Trace framework to **understand pathological cases**
- Future works: RMA, multi-threaded/OpenMP benchmark, collectives
- **MPI application programmers assume overlap happens**
  - In real life: it's more complicated!
- Release and results:
  - [http://pm2.gforge.inria.fr/mpibenchmark/](http://pm2.gforge.inria.fr/mpibenchmark/)
Thank you

Get the overlap benchmark from:

http://pm2.gforge.inria.fr/mpibenchmark/