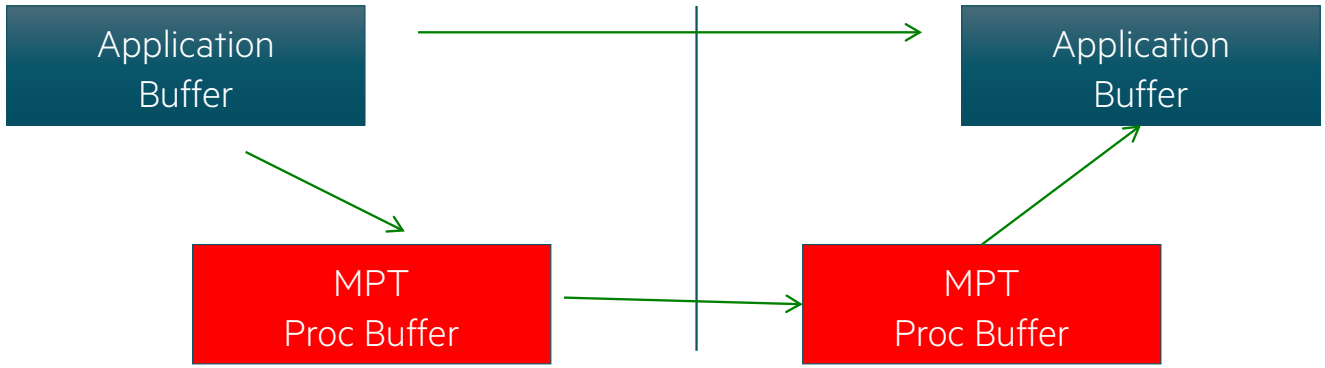
Hewlett Packard
Enterprise

HPE MPI
TUNING OF MPT

BUFFERING VS ZERO-COPY

- Buffering: Sender can keep going
- Zero-copy: Less memory and faster movement
- Control the threshold for blocking messages with MPI_BUFFER_MAX
  - If set, above this value messages are zero-copy
  - MPI_BUFFER_MAX=0 gives the best performance for the osu_latency microbenchmark



AMOUNT OF INTERNAL BUFFERS

- MPT keeps some number of internal 16k buffers
- Used for buffering and (un)packing non-contiguous datatypes
- MPI_BUFS_PER_PROC=128 by default
- Some users set to 2048
- Flexibility vs memory consumption and caching effects

BUFFERS WARNING MESSAGE

- You may see this warning:

MPT Warning: Could not allocate an internal send buffer in the last 30 seconds on rank 31 at r1i3n2. Try increasing MPI_BUFS_PER_PROC. Alternatively, destination rank 175 on host r1i2n13 may be running slowly.

- Try increasing MPI_BUFS_PER_PROC next time
- It may indicate fabric / hardware problems though

COLLECTIVES

HIGHLY OPTIMIZED COLLECTIVES

- MPT uses heuristics to determine which optimized version of a collective to use
- Sometimes the heuristics choose poorly
- Set MPI_COLL_OPT=false to disable all optimizations

CONTROLLING THE HEURISTICS

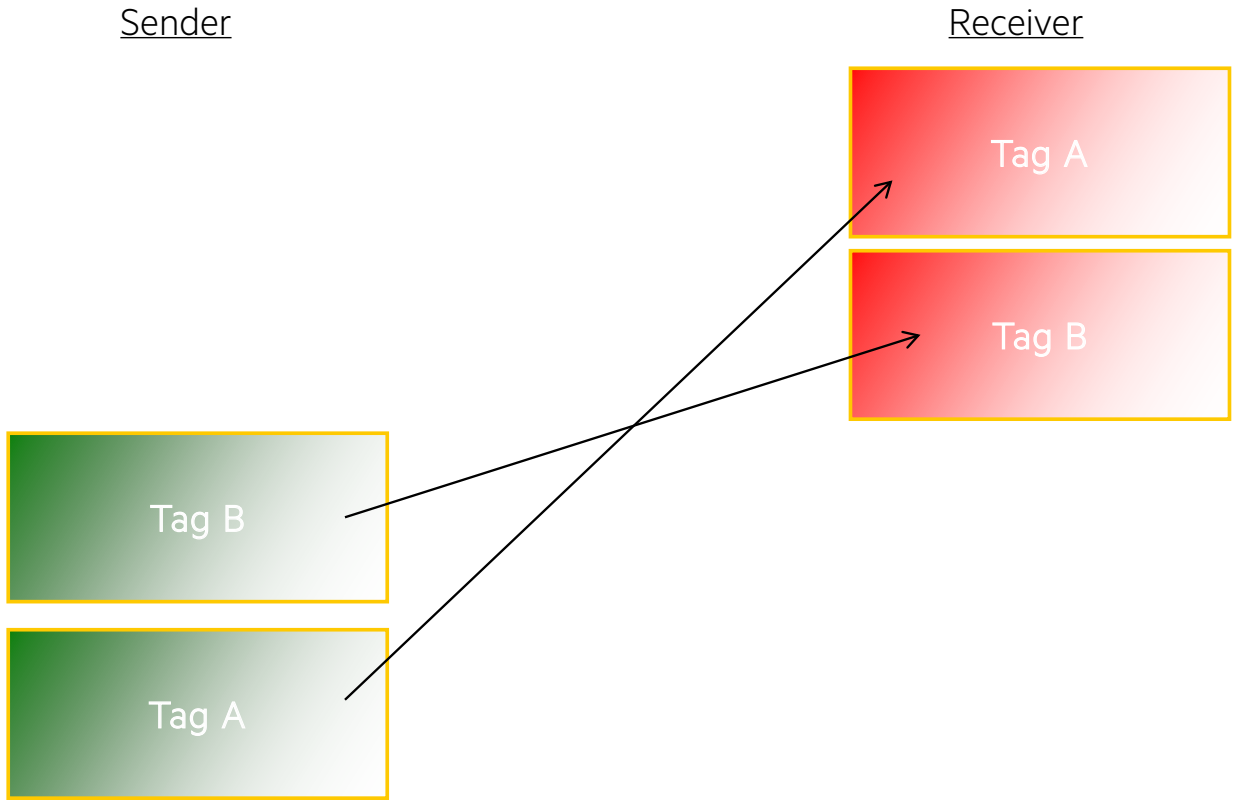
- You can force MPT to use specific implementations of each collective
- The MPI_ADJUST_* variables specify a specific optimization for all data and rank sizes
- E.g. MPI_ADJUST_ALLREDUCE=1 forces the use of recursive doubling
- See MPI(1) for more details

FLOATING POINT REPRODUCIBILITY

- The order of floating point operations can affect the results
- Don’t count on the same internal actions for reductions across different versions or even node allocations
- Set MPI_COLL_REPRODUCIBLE=true to force the same method to be used every time
- A small performance hit

INFINIBAND

TAG MATCHING IN HARDWARE



MELLANOX TAG MATCHING

- Implemented mostly in hardware whereas tag matching on OPA is all in software
- Has limits on the number of receive buffers that can be posted to the hardware

- Enabled with MPI_IB_TM=true
- Performance
  - Slightly slows down some microbenchmarks
  - Shows slightly better overlap with IMB-NBC Ialltoall

MELLANOX OFFLOAD - CONFIGURATION

$ export HPCX_HOME=/store/hpcx/latest/hpcx-v1.4.355-gcc-MLNX_OFED_LINUX-3.1-1.0.5-suse11.3-x86_64

- Or whatever your path is

$ module use $HPCX_HOME/modulefiles

$ module load hpcx

$ module load mpt

- After loading hpcx

$ export MPI_COLL_HCOLL=true

INFINIBAND - LARGE RUNS

- IB has several static settings that should be tweaked for large jobs
- MPT has individual variables for them
- The MPI_IB_CONGESTED toggle changes all of them to the best known values for large jobs
- Some sites are setting this for all their jobs
- You may also want to route your IB traffic on different QoS service levels
  - Configure this with MPI_IB_SERVICE_LEVEL
  - Put your interactive/shell traffic at the highest level
  - Put your Lustre traffic on the middle level
  - Put your MPI traffic on the lowest level so it does not starve Lustre

DEFAULT @ HLRS

DEFAULT ENVIRONMENT

- MPI_COLL_HCOLL=true
- MPI_CHECK_ARGS=true
- MPI_BUFFER_MAX=2048
- MPI_REQUEST_DEBUG=true
- MPI_IB_TM=true
- MPI_XPMEM_ENABLED=true

- MPI_SYSLOG_COPY=2
- MPI_IB_TIMEOUT=22
- MPI_IB_SERVICE_LEVEL=1

REFERENCES

- HPE Performance Software – Message Passing Interface Guide
- MPI(1)
- mpirun(1)
- intro_shmem(1)
- HPE Customer Service