Pipelined Transpose

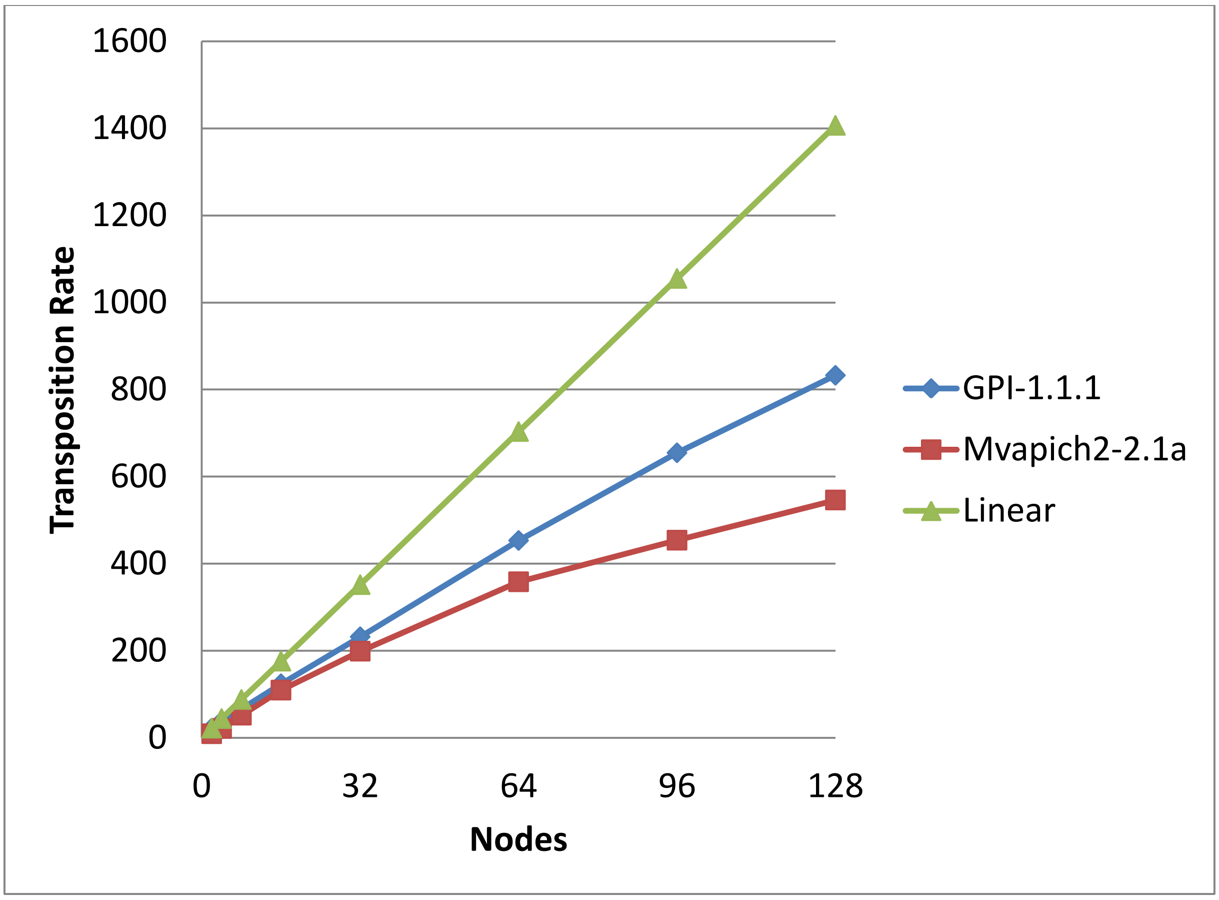
This is the initial release of a pipelined transpose proxy. This kernel is an attempt at establishing a small but meaningful proxy/benchmark for notification based 1-sided communication.

The kernel calculates a global transpose of a matrix for a column-based matrix distribution. This is a hybrid implementation where a global transpose is followed by a local transpose.

While the MPI implementation uses an MPI alltoall for global communication, the GASPI implementation uses a single communication step, in which all required communication to all target ranks is issued in a single loop. The notification based one sided communication requests are scheduled (target rank sequence) such that we minimize network congestion, i.e. we always aim for bidirectional communication even though the actual communication is entirely one-sided.

GASPI here can (and does) leverage pipelining, namely the overlap of a local transpose with the communication of the global transpose.

We note that this (fairly naïve) implementation appears to scale higher than a corresponding existing MPI implementation which make use of MPI_Alltoall.



Strong scaling for Pipelined Transpose (12288x12288 matrix), 24 Threads/Cores per node.

We note that MPI allows for pipelining via testing/waiting for non-blocking collectives and a corresponding tiling of the matrix. For pipelining a simple local transpose and for a strong scaling scenario however, this form of pipelining appears as suboptimal. The reason is that in strong scaling the tile size is shrinking with N^2 (with N == num procs). All communication then becomes latency bound rather than bandwidth bound. Tiling accelerates this effect. In addition the potential overlap of communication with computation also shrinks with N^2 (tile size).

Community involvement

We encourage the HPC community to provide new patterns or improve existing ones. No restrictions apply for either communication API or programming languages used in these improved patterns. Bonus points are granted for highly scalable implementations which also feature better programmability.

We are happy to include all those patterns/implementations in this Pipelined_transpose Release.

Related Documentation

MPI
http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf (MPI API)
http://www.open-mpi.org/ (MPI Release)
http://mvapich.cse.ohio-state.edu/news/https://computing.llnl.gov/tutorials/mpi/ (Tutorial)

GASPI
http://www.gpi-site.com/gpi2/gaspi/ (GASPI API)
https://github.com/cc-hpc-itwm/GPI-2 (GPI2 release, implements GASPI)
https://github.com/cc-hpc-itwm/GPI-2/tree/next/tutorial (Tutorial)