Message Passing Interface (MPI) Issues

The DiFX software correlator uses MPI for organizing the spawning of processes on different computers and for transmitting data from one process to another. MPI is used only for communication between mpifxcorr processes. Messaging to other software is primarily done using multicast XML documents; eVLBI uses direct TCP or UDP connections to datastream nodes, and real-time fringe searching again uses a socket-based approach; none of these comminication paths use MPI.

For the most part, the user of DiFX should not need to worry about the version of DiFX installed. The user can, however, change the way MPI runs DiFX through various command line parameters to the MPI launcher program, mpirun. These options are in general MPI vendor and version dependent. Some hints are in vendor specific sections below. Within the DiFX software tree there is only one place where mpirun is embedded in a script. That is within startdifx, a python script included in the calcif2 source tree. Recent versions factor mpirun options to the top of this file for ease of tuning. Long term a more appropriate configuration location should be established. By default startdifx assumes OpenMPI and also assumes that Infiniband is not being used – an assumption with bad performance impacts if you are indeed hoping to make use of Infiniband!

MPI rank

MPI assigns each process that it starts an integer identifier (called rank) that is used to refer back to that process. The first process has rank 0, the next rank 1, and so on. mpifxcorr makes use of the "rank" to assign to each process its purpose. Rank 0 is always the manager node. In a correlation with N datastreams, ranks 1 through N are datastream nodes, ordered by their occurrence in the .input file. The remainder (rank > N) are processing nodes.

Supported MPI implementations

DiFX uses a relatively small subset of the MPI specification. (Fix Me! – I think MPI specification version 1 is all that is actually needed). MPICH and OpenMPI have been shown to work.

OpenMPI

A vast majority of known DiFX installations use OpenMPI (available at www.openmpi.org). OpenMPI is in active development by many major corporations, research institutes, and universities. It rather simply compiles and installs, making use of the GNU configuration tools. OpenMPI makes use of a "modular component architecture" where various subsystems (schedulers, transport layer, …) can each be tuned at run-time for a particular application. Some mpirun options that have been employed by users of DiFX include

- `–mca btl ^udapl,openib` This option turns off support for Infiniband and hence prevents a warning message from appearing in cases where infiniband is not available. This should be excluded from systems that actually want to use Infiniband.
- `–mca mpi_yield_when_idle 1` This option turns off overly greedy polling of a network socket to reduce CPU usage. For DiFX, latency is not a problem due to the generous buffering at every stage so this should only improve performance.

- `–mca rmaps seq` This option tells mpirun to assign monotonicly increasing MPI rank strictly according to the ordering of entries in the machines file. Without this, OpenMPI takes liberties and will change the ordering if the same machine is listed more than once. This option can be useful if it is necessary to run more than one mpifxcorr process (which itself can have multiple threads) on one machine.