FAQ: MPI Problems

My MPI program fails to compile

If for your C++ MPI compilation you receive error messages like "SEEK_SET is #defined but must not be for the C++ binding of MPI.", sometimes also "Include mpi.h before stdio.h", then please consider reworking the header ordering in your source code. As a workaround, it is also possible to set the macro `-DMPICH_IGNORE_CXX_SEEK`.

My MPI program crashes. What do I do?

The symptom will look somewhat like this (sgi MPT):

MPI: MPI_COMM_WORLD rank 0 has terminated without calling MPI_Finalize()
MPI: aborting job
MPI: Received signal x (x may e.g. be 11)

Even if your program appeared to run correctly on another machine/with a different number of CPUs, there still may be bugs in the program. There also may be bugs in the MPI implementation, but that is less probable. To find out where bad things are happening, please perform a traceback procedure as described below.

MPI crash due to incorrect header information (any MPI)

If debugging shows that MPI calls very obviously deliver incorrect results (especially administrative calls), please check whether you've got a file called mpi.h, mpi.f or mpi.mod somewhere in your private include path which interferes with the corresponding files in the system include path. This may lead to errors since different MPI implementations are not binary or even source compatible. Please either remove the spurious files or change your include path so these files are not referenced.

Traceback for parallel codes

The following recipe works for SGI's MPI implementation (MPT).

First, build your application as described in the section about FAQ: Compiler Problems (in the serial case), except that you should use mpif90, mpicc etc.

Then, perform the following command sequence inside a SLURM batch script or inside a salloc shell:

```
$ mpiexec -n 32 ./myparprog.exe
```

to trace back to the point in the code where the crash happens.