MPI - Message Passing Interface

The **Message Passing Interface** (MPI) is a standardized and portable message-passing system that defines the syntax and semantics of a core of library routines useful to a wide range of users writing portable message-passing programs in `C`, `C++`, and `Fortran`.

History

Before the 1990's, writing parallel applications for different computing architectures was a difficult and tedious task. At that time, many libraries could facilitate building parallel applications, but there was not a standard accepted way of doing it. A small group of researchers started discussions in Austria and came out with a Workshop on 'standards' for **Message Passing in a Distributed Memory Environment**. Attendees discussed the basic features essential to a standard message-passing interface and established a working group to continue the standardization process and put forward a preliminary draft proposal, **"MPI1"**, in November 1992. After its first implementations were created, MPI was widely adopted and still continues to be the de-facto method of writing **message-passing** applications.

MPI Chronology:

- 1994: MPI-1 (specification, not strictly a library)
- 1996: MPI-2 (addresses some extensions)
- 2012: MPI-3 (extensions, remove `C++` bindings)

Programming Model

- Distributed programming model. Also data parallel

- Hardware platforms: **distributed, shared and hybrid**

- Parallelism is explicit. The programmer is responsible for implementing all parallel constructs all parallel constructs.

The number of tasks dedicated to run a parallel program is static. New tasks can not be dynamically spawned during run time. However, MPI-2 addressed this issue.

Benefits of MPI

- **Portability:** There is no need to modify your source code when you port your application to a different platform that supports (and is compliant with) the MPI standard.

- **Standardization:** MPI is the only message passing library which can be considered a standard. It is supported on virtually all HPC platforms. Practically, it has replaced all previous message passing libraries.

- **Functionality:** Over 115 routines are defined in the MPI-1 alone.

- **Availability:** A variety of implementations are available, both vendor and public domain (see below).

How to get the MPI?

Your institutions HPC cluster should have some kinds of MPI installed already, which may be (but not limited to):

- Open MPI
- Intel MPI
- MPICH2
- SGI’s MPT
- and so on.

You need to load the appropriate module (e.g., `module load openmpi`) prior to running code containing MPI stuffs. If your load was successful, you should be able to type `mpiexec --version` and see something similar to this:

```
$ mpiexec --version
mpiexec (OpenRTE) 1.8.8

Report bugs to http://www.open-mpi.org/community/help/
```

MPI Header files:

MPI Header files:

- C/C++: `#include <mpi.h>`
- ForTran: `include 'mpif.h'`

Compiler wrappers:

- Intel: (`icc -lmpi`) and (`ifort -lmpi`)
- GNU: mpicc, `mpif77`, `mpif90`, `mpicxx`