Get Started with Intel® MPI Library

for Intel® oneAPI on Linux* OS

The Intel® MPI Library is a multi-fabric message passing library that implements the Message Passing Interface, version 3.1 (MPI-3.1) specification. Use the library to develop applications that can run on multiple cluster interconnects.

Intel MPI Library has the following features:

- Scalability up to 340k processes
- Low overhead enables analysis of large amounts of data
- MPI tuning utility for accelerating your applications
- Interconnect independence and flexible runtime fabric selection

Intel MPI Library is available as a standalone product, as part of the Intel® Parallel Studio XE Cluster Edition, and as part of the Intel® oneAPI HPC Toolkit.

Product Contents

The product consists of the following main components:

- Compilation tools, including compiler drivers such as `mpiicc` and `mpifort`
- Include files and modules
- Shared (`.so`) and static (`.a`) libraries, debug libraries, and interface libraries
- Process Manager and tools to run programs
- Test code
- Documentation provided as a separate package or available from the Intel Developer Zone

Intel MPI Library also includes Intel® MPI Benchmarks, which enable you to measure MPI operations on various cluster architectures and MPI implementations. For details, see the *Intel MPI Benchmarks User Guide*. Source code is available in the GitHub repository.

Key Features

Intel MPI Library has the following major features:

- MPI-1, MPI-2.2 and MPI-3.1 specification conformance
- Interconnect independence
- C, C++, Fortran* 77, Fortran 90, and Fortran 2008 language bindings

June 2020

Prerequisites

Before you start using Intel MPI Library, make sure to complete the following steps:

1. Source the `vars.sh` script to establish the proper environment settings for the Intel MPI Library. The script is located at `<install-dir>/mpi/<version>.<update>/env`, where `<install-dir>` is the Intel oneAPI HPC Toolkit installation directory (by default, `/opt/intel/oneapi`).
2. Create a `hostfile` text file that lists the nodes in the cluster using one host name per line. For example:

   ```
   clusternode1
   clusternode2
   ```

3. Make sure the passwordless SSH connection is established among all nodes of the cluster. It ensures the proper communication of MPI processes among the nodes.

After completing these steps, you are ready to use Intel MPI Library.

For detailed system requirements, see the "System Requirements" section in Release Notes.

Building and Running MPI Programs

Compiling an MPI Program

1. Make sure you have a compiler in your `PATH`. To check this, run the `which` command on the desired compiler. For example:

   ```
   $ which icc

   /opt/intel/inteloneapi/compiler/<version>.<update>/linux/bin/intel64/icc
   ```

2. Compile a test program using the appropriate compiler driver. For example:

   ```
   $ mpiicc -o myprog <install-dir>/test/test.c
   ```

Running an MPI Program

Use the previously created hostfile and run your program with the mpirun command as follows:

```
$ mpirun -n <# of processes> -ppn <# of processes per node> -f ./hostfile ./myprog
```

For example:

```
$ mpirun -n 2 -ppn 1 -f ./hostfile ./myprog
```

The test program above produces output in the following format:

```
Hello world: rank 0 of 2 running on clusternode1

Hello world: rank 1 of 2 running on clusternode2
```

This output indicates that you properly configured your environment and Intel MPI Library successfully ran the test MPI program on the cluster.

Troubleshooting

If you encounter problems when using Intel MPI Library, go through the following general procedures to troubleshoot them:

- Check system requirements, known issues and limitations in the *Release Notes*.
- Check hosts accessibility. Run a simple non-MPI application (for example, `hostname` utility) on the problem hosts with `mpirun`. This check helps you reveal the environmental problem (for example, SSH is not configured properly), or connectivity problem (for example, unreachable hosts).
- Run the MPI application with debug information enabled. To enable the debug information, set the environment variable `I_MPI_DEBUG=6`. You can also set a different debug level to get more detailed information. This action helps you find out the problem component.

See more details in the “Troubleshooting” section of the Developer Guide.

Training and Documentation

- Online Training: An excellent resource for learning the Intel MPI Library capabilities through various guides, videos, webinars, and more.
- Release Notes: Up-to-date information about the product, including: what's new, key features, system requirements, and known limitations.
- Online Documentation: Links to all available Intel MPI Library documents.

Notices and Disclaimers