



MPI编译环境的使用

李会民

hmli@ustc.edu.cn

中国科学技术大学 超级计算中心

2014年4月10日



1 主流MPI环境介绍

2 联系信息



- 各种MPI编译环境实际上为MPI标准的不同实现
- 利用在普通编译器（比如Intel编译器）基础上添加必要的MPI参数以指定MPI库的路径等链接MPI库进行编译
- 除具体MPI实现的参数之外，其调用的普通编译器的参数继续有效
- 优化等不仅需参考此MPI编译环境也要参考调用的普通编译器



- Intel MPI、Open MPI: 既支持InfiniBand, 也支持以太网
- MPICH和MPICH2: 支持以太网, 不支持InfiniBand网络
- MVAPICH、MVAPICH2: 基于MPICH和MPICH2, 支持InfiniBand网络
- 编译命令基本一致
- 一些编译参数有些不同
- MPI作业提交的参数也有所不同, 当前LSF作业调度系统提供的MPI作业运行脚本mpijob, 主要针对系统默认设置的MPI实现, 其它版本的MPI实现未必能直接使用, 需要根据具体MPI实现对提交作业的要求进行改编



当前科大超算系统部署的类型

- ChinaGrid高性能计算集群群(InfiniBand网络): Intel MPI和Open MPI
- 刀片及胖节点超级计算系统(千兆以太网): Intel MPI和Open MPI
- 联想深腾7000G GPU集群(InfiniBand网络): Open MPI、MVAPICH、MVAPICH2、QLogic MPI、LAM

Intel MPI和Open MPI为今后系统主要部署的MPI环境



设置MPI编译环境

- Intel MPI和Open MPI，并可与不同编译器相互配合使用，安装目录分别在 `/opt/intel/impi`和 `/opt/openmpi-*1`
- 用户可以运行 `mpi-selector-menu`命令按照提示选择自己使用的MPI环境（注意数字后需要加u），设置完成后最好重新登录以便设置生效：

```
Current system default: intel-mpi-4.1.0.030_intel-compiler-13.1.0.146
```

```
Current user default: <none>
```

```
"u" and "s" modifiers can be added to numeric and "U"  
commands to specify "user" or "system-wide".
```

1. intel-mpi-4.0.3.008_intel-compiler-13.0.1.117
2. intel-mpi-4.1.0.030_intel-compiler-13.0.1.117
3. intel-mpi-4.1.0.030_intel-compiler-13.1.0.146
4. openmpi-1.6.3_intel-compiler-13.0.1.117
5. openmpi-1.6.3_intel-compiler-13.1.0.146
6. openmpi-1.6.4_gcc-4.4.7
7. openmpi-1.6.4_intel-compiler-13.1.0.146
8. openmpi-1.6.4_pgi-10.6

```
U. Unset default
```

```
Q. Quit
```

```
Selection (1-8[us], U[us], Q):
```

¹具有不同版本的Open MPI与编译器的组合。



- which mpif90*

```
/opt/intel/impi/4.1.0.030/intel64/bin/mpif90
```

- mpif90 -v*

```
mpif90 for the Intel(R) MPI Library 4.1 for Linux*
Copyright(C) 2003-2013, Intel Corporation. All rights reserved.
Using built-in specs.
Target: x86_64-redhat-linux
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info
--with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --enable-shared
--enable-threads=posix --enable-checking=release --with-system-zlib --enable-__cxa_atexit
--disable-libunwind-exceptions --enable-gnu-unique-object
--enable-languages=c,c++,objc,obj-c++,java,fortran,ada --enable-java-awt=gtk --disable-dssi
--with-java-home=/usr/lib/jvm/java-1.5.0-gcj-1.5.0.0/jre --enable-libgcj-multifile
--enable-java-maintainer-mode --with-ecj-jar=/usr/share/java/eclipse-ecj.jar
--disable-libjava-multilib --with-ppl --with-cloog --with-tune=generic --with-arch_32=i686
--build=x86_64-redhat-linux
Thread model: posix
gcc version 4.4.7 20120313 (Red Hat 4.4.7-4) (GCC)
```



- 是一种多模消息传递接口(MPI)库
- 4.1版本实现了**MPI V2.2标准**
- 可以使开发者采用新技术改变或升级其处理器和互联网络而无需改编软件或操作环境成为可能
- 主要包含以下内容：
 - 运行时环境(RTO): 具有运行程序所需要的工具, 包含多功能守护进程(MPD)、Hydra及支持的工具、共享库(.so)和文档
 - 开发套件(SDK): 包含所有运行时环境组件和编译工具, 含编译器命令, 如***mpiicc***、头文件和模块、静态库(.a)、调试库、追踪库和测试代码
- 主页: <http://software.intel.com/en-us/intel-mpi-library/>

Intel MPI与Open MPI、MPICH等MPI实现不同:

- *mpiicc*、*mpiicpc*和*mpiifort*命令: 使用Intel编译器
- *mpicc*、*mpif90*和*mpifc*命令: 默认使用GNU编译器
- Intel MPI编译命令及其对应关系

| 编译命令 | 调用的默认编译器命令 | 支持的语言 | 支持的应用二进制接口 |
|--|------------|------------------------|------------|
| 通用编译器 | | | |
| <i>mpicc</i> | gcc, cc | C | 32/64 bit |
| <i>mpicxx</i> | g++ | C/C++ | 32/64 bit |
| <i>mpifc</i> | gfortran | Fortran77*/Fortran 95* | 32/64 bit |
| GNU* Compilers Versions 3 and Higher | | | |
| <i>mpigcc</i> | gcc | C | 32/64 bit |
| <i>mpigxx</i> | g++ | C/C++ | 32/64 bit |
| <i>mpif77</i> | g77 | Fortran 77 | 32/64 bit |
| <i>mpif90</i> | gfortran | Fortran 95 | 32/64 bit |
| Intel Fortran, C++ Compilers Versions 11.1 and Higher | | | |
| <i>mpiicc</i> | icc | C | 32/64 bit |
| <i>mpiicpc</i> | icpc | C++ | 32/64 bit |
| <i>mpiifort</i> | ifort | Fortran77/Fortran 95 | 32/64 bit |

其中:

- ia32: IA-32架构
- intel64: Intel 64(x86_64, amd64)架构
- 移植现有的MPI程序到Intel MPI库时, 请重新编译所有源代码
- 如需显示某命令的简要帮助, 可以不带任何参数直接运行该命令



- `-mt_mpi`: 采用以下级别链接线程安全的MPI库:
MPI_THREAD_FUNNELED、MPI_THREAD_SERIALIZED或
MPI_THREAD_MULTIPLE
默认使用MPI_THREAD_FUNNELED级别线程安全库
注意:
 - 如用Intel C编译器编译时添加了`-openmp`或`-parallel`参数, 则使用线程安全库
 - 如用Intel Fortran编译器编译时添加如下参数, 则使用线程安全库:
 - `-openmp`
 - `-parallel`
 - `-threads`
 - `-reentrancy`
 - `-reentrancy threaded`
- `-static_mpi`: 静态链接Intel MPI库, 并不影响其它库的链接方式
- `-static`: 静态链接Intel MPI库, 将其传递给编译器, 作为编译器参数



编译命令参数 II

- `-config=<name>`: 使用的配置文件
- `-profile=<profile_name>`: 使用的MPI分析库文件
- `-t`或`-trace`: 链接Intel Trace Collector库
- `-check_mpi`: 链接Intel Trace Collector正确性检查库
- `-ilp64`: 启用ILP64支持。对于Fortran程序编译时如果使用`-i8`选项，那么也需要此ILP64选项
- `-dynamic_log`: 与`-t`组合使用链接Intel Trace Collector库。不影响其它库链接方式
- `-g`: 采用调试模式编译程序，并针对Intel MPI调试版本生成可执行程序。可查看官方手册Environment variables部分`I_MPI_DEBUG`变量查看`-g`参数添加的调试信息。采用调试模式时不对程序进行优化，可查看`I_MPI_LINK`获取Intel MPI调试版本信息



- `-link_mpi=<arg>`: 指定链接MPI的具体版本, 具体请查看 `I_MPI_LINK` 获取Intel MPI版本信息。此参数将覆盖掉其它参数, 如 `-mt_mpi`、`-t=log`、`-trace=log` 和 `-g`
- `-O`: 启用编译优化
- `-fast`: 对整个程序进行最大化速度优化。此参数强制使用静态方法链接Intel MPI库。 *mpicc*、*mpiicpc* 和 *mpiifort* 编译命令支持此参数
- `-echo`: 显示所有编译命令脚本做的信息
- `-show`: 仅显示编译器如何链接, 但不实际执行
- `-{cc,cxx,fc,f77,f90}=<compiler>`: 选择使用的编译器。如:
mpicc -cc=icc -c test.c
- `-compchk`: 启用编译器设置检查, 以保证调用的编译器配置正确
- `-v`: 显示版本信息



- `-gcc-version=<nnn>`: 设置 *mpicxx* 和 *mpiicpc* 命令编译时采用GNU C++环境的版本, 如<nnn>的值为340, 表示对应GNU C++ 3.4.x

| <nnn>值 | GNU* C++版本 |
|--------|------------|
| 320 | 3.2.x |
| 330 | 3.3.x |
| 340 | 3.4.x |
| 400 | 4.0.x |
| 410 | 4.1.x |
| 420 | 4.2.x |
| 430 | 4.3.x |
| 440 | 4.4.x |
| 450 | 4.5.x |
| 460 | 4.6.x |
| 470 | 4.7.x |



- $I_MPI_{\{CC,CXX,FC,F77,F90\}}_PROFILE$ 和 $MPI_{\{CC,CXX,FC,F77,F90\}}_PROFILE$:
 - 默认分析库
 - 语法: $I_MPI_{\{CC,CXX,FC,F77,F90\}}_PROFILE=<profile_name>$
 - 过时语法: $MPI_{\{CC,CXX,FC,F77,F90\}}_PROFILE=<profile_name>$
- $I_MPI_TRACE_PROFILE$:
 - 设定-trace参数使用的默认分析文件
 - 语法: $I_MPI_TRACE_PROFILE=<profile_name>$
 - $I_MPI_{\{CC,CXX,F77,F90\}}_PROFILE$ 环境变量将覆盖掉 $I_MPI_TRACE_PROFILE$
- $I_MPI_CHECK_PROFILE$:
 - 设定-check_mpi参数使用的默认分析
 - 语法: $I_MPI_CHECK_PROFILE=<profile_name>$
- $I_MPI_CHECK_COMPILER$:
 - 设定启用或禁用编译器兼容性检查



- 语法: `I_MPI_CHECK_COMPILER=<arg>`
 - `<arg>`为enable | yes | on | 1时打开兼容性检查
 - `<arg>`为disable | no | off | 0时, 关闭编译器兼容性检查, 为默认值
- `I_MPI_{CC,CXX,FC,F77,F90}`和`MPICH_{CC,CXX,FC,F77,F90}`:
 - 语法: `I_MPI_{CC,CXX,FC,F77,F90}=<compiler>`
 - 过时语法: `MPICH_{CC,CXX,FC,F77,F90}=<compiler>`
 - `<compiler>`为编译器的编译命令名或路径
- `I_MPI_ROOT`:
 - 设置Intel MPI库的安装目录路径
 - 语法: `I_MPI_ROOT=<path>`
 - `<path>`为Intel MPI库的安装后的目录
- `VT_ROOT`:
 - 设置Intel Trace Collector的安装目录路径
 - 语法: `VT_ROOT=<path>`
 - `<path>`为Intel Trace Collector的安装后的目录



● *I_MPI_COMPILER_CONFIG_DIR*:

- 设置编译器配置目录路径
- 语法: *I_MPI_COMPILER_CONFIG_DIR*=<path>
- <path>为编译器安装后的配置目录, 默认值为< installdir >/<arch>/etc

● *I_MPI_LINK*:

- 设置链接MPI库版本
- 语法: *I_MPI_LINK*=<arg>, <arg>可为:
 - opt: 优化的单线程版本Intel MPI库
 - opt_mt: 优化的多线程版本Intel MPI库
 - dbg: 调试的单线程版本Intel MPI库
 - dbg_mt: 调试的多线程版本Intel MPI库
 - log: 日志的单线程版本Intel MPI库
 - log_mt: 日志的多线程版本Intel MPI库



编译举例

对于并行程序，对应不同类型源文件的编译命令如下：

- ***mpicc -o yourprog-mpi yourprog-mpi.c***
调用默认C编译器将C语言的MPI并行程序yourprog-mpi.c编译为可执行文件yourprog-mpi
- ***mpiicxx -o yourprog-mpi yourprog-mpi.cpp***
调用Intel C++编译器将C++语言的MPI并行程序yourprog-mpi.cpp编译为可执行文件yourprog-mpi
- ***mpif90 -o yourprog-mpi yourprog-mpi.f***
调用GNU Fortran编译器将Fortran 77语言的MPI并行程序yourprog-mpi.f编译为可执行文件yourprog-mpi
- ***mpiifort -o yourprog-mpi yourprog-mpi.f90***
调用Intel Fortran编译器将Fortran 90语言的MPI并行程序yourprog-mpi.f90编译为可执行文件yourprog-mpi



与编译器相关的编译选项

- MPI编译命令实际上是调用Intel、PGI或GCC编译器进行编译
- 具体优化选项等，请参看相关编译器手册



Open MPI等主流MPI编译环境编译举例

- Open MPI、MPICH、MVAPICH2和MVAPICH MPI编译命令主要为：
mpicc、mpic++、mpicxx、mpiCC、mpif77和mpif90
- 不同类型程序的编译命令如下：
 - 将C语言的MPI并行程序yourprog-mpi.c编译为可执行文件yourprog-mpi:
mpicc -o yourprog-mpi yourprog-mpi.c
 - 将C++语言的MPI并行程序yourprog-mpi.cpp编译为可执行文件yourprog-mpi，mpicxx也可换为mpic++或mpiCC:
mpicxx -o yourprog-mpi yourprog-mpi.cpp
 - 将Fortran 77语言的MPI并行程序yourprog-mpi.f编译为可执行文件yourprog-mpi:
mpif77 -o yourprog-mpi yourprog-mpi.f
 - 将Fortran 90语言的MPI并行程序yourprog-mpi.f90编译为可执行文件yourprog-mpi:
mpif90 -o yourprog-mpi yourprog-mpi.f90
- 编译优化等，主要结合所使用的编译器的编译选项与具体MPI实现的编译选项共同设置



- **-show**: 仅显示命令信息，但不进行编译
- **-help**: 给出简单帮助
- 用指定的编译器编译命令代替默认的编译命令，只有在编译器与MPICH库兼容时才可使用
 - **-cc=name**: mpicc的参数，指定C编译器
 - **-CC=name**: mpiCC和mpicxx的参数，指定C++编译器
 - **-fc=name**: mpif77的参数，指定Fortran 77编译器
 - **-f77=name**: mpif77的参数，指定Fortran 77编译器
 - **-f90=name**: mpif90的参数，指定Fortran 90之后的编译器
- **-compile-info**: 显示程序编译的过程
- **-link-info**: 显示链接过程



Open MPI主要编译选项

- -showme: 仅显示命令信息，但不进行编译
- -showme:compile: 仅显示编译器编译参数信息，但不进行编译
- -showme:link: 仅显示编译器链接时的参数信息，但不进行链接
- 用**OMPI_value**变量控制使用的编译命令、编译参数等，**value**可为：
 - CPPFLAGS: 预处理选项
 - LDFLAGS: 链接选项
 - LIBS: 链接库选项
 - CC: C编译命令
 - CFLAGS: C编译选项
 - CXX: C++编译命令
 - CXXFLAGS: C++编译选项
 - F77: Fortran 77编译命令
 - FFLAGS: Fortran 77编译选项
 - FC: Fortran 9x编译命令
 - FCFLAGS: Fortran 9x编译选项

如使用gfortan作为Fortran 90编译命令，并显示编译信息：

OMPI_FC=gfortan mpif90 -showme



Open MPI并行实现下的并程序调试

- 编译时添加-g参数，如

```
mpif90 -g yourmpi-prog.f90 -o yourmpi-prog
```

- 几种运行方式:

- 使用GNU调试命令 *gdb*，不调用初始调试命令调试:

```
mpiexec -n 4 xterm -e gdb -q -tui ./yourmpi-prog
```

- 使用GNU调试命令 *gdb* 调试，并调用调试命令文件 *dbg.txt*:

```
mpiexec -n 4 xterm -e gdb -q -tui -x dbg.txt ./yourmpi-prog
```

- 使用Intel调试命令 *idbc*，不调用初始调试命令调试:

```
mpiexec -n 4 xterm -e idbc ./yourmpi-prog
```

- 使用Intel调试命令 *idbc* 调试，并调用调试命令文件 *dbg.txt*:

```
mpiexec -n 4 xterm -e idbc -command dbg.txt ./yourmpi-prog
```

- 调试命令文件 *dbg.txt* 内容格式，每行一条命令，比如:

```
break 13  
condition 1 k==2  
run
```

注意: xterm为Linux下的一种图形终端命令，也可使用其它的。上述调试需要图形界面，如远程连接Linux系统调试，需打开X11转发



MPI程序出错时常用调试方式

以Intel调试器和Open MPI的配合为例:

- 添加-g参数编译
- 设置dbg.txt文件内容为

```
run
```

- 开始调试:

```
mpiexec -n 4 xterm -e idbc -command dbg.txt ./yourm
```

程序将会自动停止在出错的位置, 并显示对行的源代码

相关文档: <http://www.open-mpi.org/faq/?category=debugging>



- 中国科大超算中心:
 - 电话: 0551-63602248
 - 信箱: sccadmin@ustc.edu.cn
 - 主页: <http://scc.ustc.edu.cn>
 - 办公室: 中国科大东区新图书馆一楼东侧126室
- 李会民:
 - 电话: 0551-63600316
 - 信箱: hml@ustc.edu.cn
 - 主页: <http://hml.ustc.edu.cn>