MPI Study Notes

Tao He

Computer Science

Medical Imaging High Performance Computing Lab

Introduce to MPI

- Message passing is a programming paradigm used widely on parallel computers.
- MPICH is a open-source implementation of MPI.
- Power for a distributed program on PC Cluster based on Beowulf structure
- MPICH website:

  http://www-unix.mcs.anl.gov/mpi/mpich/

  http://www-unix.mcs.anl.gov/mpi/mpich2/

LINUX Cluster

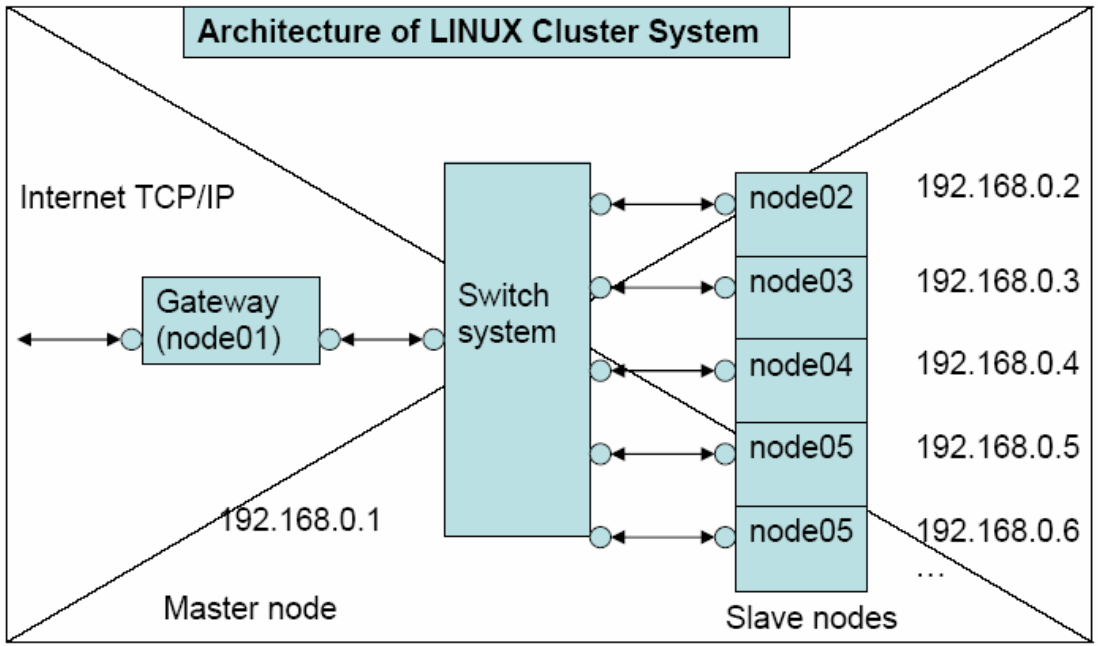

Figure 1 Architecture design of LINUX cluster

- http://www.cs.uiowa.edu/~jni/HowTo/HowToBuildAClusterG.pdf

Quick Start

1. Download SSH(Secure Shell is a program that provides secure connections to host machines and to use Secure FTP client.) from ITS help desk

   https://helpdesk.its.uiowa.edu/software/download/

2 Login on our Cluster system with your account and password with ssh.

   hostname: mihpclab.radiology.uiowa.edu

   testuser: testm

   password: test1234

3 Write your MPI program. MPICH binds C, C++, Fortran77, and Fortran90 compilers.

4 Compile the source files with the commands:

   $mpicc –o YourExe-File YourMPI-C-file

   $mpicxx –o YourExe-File YourMPI-C++-file

   $mpif77 –o YourExe-File YourMPI-F77-file

   $mpif90 –o YourExe-File YourMPI-F90-file

5. Run a MPI program:

   $mpirun –np YourProcessNum YourExe-File

MPI Program Structure

```
#include<mpi.h>
Main(){
  //share variables
  MPI_Init(&argc,&argv);
  MPI_Comm_rank(MPI_COMM_WORLD,&myid);
  if (myid==0){
   // root processor's operations
    MPI_Send(buffer, strlen(buffer)+1, MPI_CHAR, dest_id, 99, MPI_COMM_WORLD);
    MPI_Recv(buffer, BUFLEN, MPI_CHAR, MPI_ANY_SOURCE, 99,   MPI_COMM_WORLD, &status);
   }else{
     // children processors' operations
      MPI_Recv(buffer, BUFLEN, MPI_CHAR, MPI_ANY_SOURCE, 99, MPI_COMM_WORLD, &status);
     …
      MPI_Send(buffer, strlen(buffer)+1, MPI_CHAR, next, 99, MPI_COMM_WORLD);
      printf("%d sent '%s' \n",myid,buffer);
     }
      MPI_Barrier(MPI_COMM_WORLD);
      MPI_Finalize();
      return (0);
}
```

9 Common MPI routines

- **`MPI_Init`**
- **`MPI_Finalize`**
- **`MPI_Comm_size`**
- **`MPI_Comm_rank`**
- **`MPI_Send`**
- **`MPI_Recv`**
- **`MPI_Bcast`**
- **`MPI_Reduce`**
- **`MPI_Barrier`**

MPI Data Type

MPI_CHAR	char
MPI_BYTE	unsigned char
MPI_SHORT	short
MPI_INT	int
MPI_LONG	long
MPI_FLOAT	float
MPI_DOUBLE	double
MPI_UNSIGNED_CHAR	unsigned char
MPI_UNSIGNED_SHORT	unsigned short
MPI_UNSIGNED	unsigned int
MPI_UNSIGNED_LONG	unsigned long
MPI_LONG_DOUBLE	long double

MPE (Multi-Processing Environment)

- MPE provides programmers a complete suite of performance analysis tools for their MPI programs.
- **Logging Library ----** This is the most useful and widely used profiling libraries in MPE. (CLOG, SLOG and SLOG2: logging file formats)
- MPE's graphical tools include multiple display programs, such as **jumpshot-2** for CLOG, **jumpshot-3** for SLOG, **jumpshot-4** for SLOG2.

Jumpshot-4 for slog2

- Compile a MPICH program with MPE Logging Library:

```
taohe@master:~/examples/mpe> mpicc -c cpi.c
taohe@master:~/examples/mpe> mpicc -o cpi_log cpi.o -llmpe -lmpe –lm
```

- Convert log format from CLOG to SLOG:

taohe@master:~/examples/mpe> clogTOslog2 cpi_log.clog

- Start Jumpshot-4