

使用 gdb 调试 mpich 多进程程序

周永强

June 22, 2014

1 说明

1.1 目的

本文档讲述如何在 linux 下通过 gdb 对 mpich 程序进行调试，mpich 运行通常是多个进程同时运行，因此调试起来比较麻烦，而且具有一定的技巧。文中通过一个调试例子展示如何进行调试。

1.2 原理

调试的方式有两个要点：一是通过进程号来进入要调试的程序，二是通过在源代码中设置循环的技巧使程序阻塞。

2 调试过程

要调试的程序为 hellow，其源代码为 hello.c。

1. 编辑 hello.c 的源文件如下

```
#include <stdio.h>
#include "mpi.h"

#define MAX_STRING 100

int main( int argc, char *argv[] )
{
    char greeting[MAX_STRING];
    int my_rank;
    int comm_sz;
    int a = 1;

    MPI_Init(NULL, NULL);
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
    MPI_Comm_size(MPI_COMM_WORLD, &comm_sz);
```

```

if (my_rank == 0)
{
    printf("I am root.\n");
    sprintf(greeting, "Hello, my child", my_rank, comm_sz);

    while (a == 1);
    MPI_Send(greeting, strlen(greeting)+1,
             MPI_CHAR, 1, 0, MPI_COMM_WORLD);
}
else
{
    while (a == 1);
    MPI_Recv(greeting, MAX_STRING, MPI_CHAR,
             0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    printf("I am child: %d, greeting from root: %s\n",
           my_rank, greeting);
}

MPI_Finalize();
return 0;
}

```

2. 编译程序，注意加上参数 -g

```
mpicc -g hellow.c -o hellow
```

3. 运行 hellow，这里以两个进程为例

```
mpiexec -n 2 ./hellow
```

4. 由于设置了死循环，所以上述程序会阻塞执行，另外打开两个终端，分别对两个进程进行调试，调试前需要先获取进程的 pid 号，获取命令为

```
ps aux | grep hellow
```

5. 找到对应的进程号，我机器上 hellow 对应的两个进程分别为 9135 和 9136，可以对二者分别进行调试，调试 9135 号进程的方式为，首先运行 gdb，命令为

```
gdb
```

6. gdb 启动以后，对 9135 进程调试，9136 调试方式一样，在 gdb 中运行命令

```
attach 9135
```

7. 此时可以看到程序阻塞在循环处，通过设置循环的变量，使循环的条件变为假，跳出循环，进入下一步的执行，便可接着进行调试了

```
set var a = 0
```