

Re: LAM/MPI MPICH-2 Compatibility

Source: <http://newsgroups.derkeiler.com/Archive/Comp/comp.parallel.mpi/2007-04/msg00016.html>

- *From:* blmblm@xxxxxxxxxxxxxx <blmblm@xxxxxxxxxxxxxx>
 - *Date:* 14 Apr 2007 10:16:21 GMT
-

In article <1176525944.853140.172830@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>, <ananghudaya@xxxxxxxxxx> wrote:

Hi Massingill,

Here is the error code that I have obtained:

LAM 7.0.6/MPI 2 C++/ROMIO – Indiana University

```
MPI_Recv: process in local group is dead (rank 1, MPI_COMM_WORLD)
MPI_Recv: process in local group is dead (rank 2, MPI_COMM_WORLD)
MPI_Recv: process in local group is dead (rank 4, MPI_COMM_WORLD)
Rank (2, MPI_COMM_WORLD): Call stack within LAM:
Rank (2, MPI_COMM_WORLD): - MPI_Recv()
Rank (2, MPI_COMM_WORLD): - MPI_Bcast()
Rank (2, MPI_COMM_WORLD): - main()
Rank (1, MPI_COMM_WORLD): Call stack within LAM:
Rank (1, MPI_COMM_WORLD): - MPI_Recv()
Rank (1, MPI_COMM_WORLD): - MPI_Bcast()
Rank (1, MPI_COMM_WORLD): - main()
Rank (4, MPI_COMM_WORLD): Call stack within LAM:
Rank (4, MPI_COMM_WORLD): - MPI_Recv()
Rank (4, MPI_COMM_WORLD): - MPI_Bcast()
Rank (4, MPI_COMM_WORLD): - main()
MPI_Recv: process in local group is dead (rank 16, MPI_COMM_WORLD)
Rank (16, MPI_COMM_WORLD): Call stack within LAM:
Rank (16, MPI_COMM_WORLD): - MPI_Recv()
Rank (16, MPI_COMM_WORLD): - MPI_Bcast()
Rank (16, MPI_COMM_WORLD): - main()
```

Hope that it could help...

Not as much as I had hoped -- apparently I'm not as good at interpreting these messages as I might have thought -- but I Googled, and GWMF. Here's a useful-looking FAQ:

<http://lam-mpi.miscellaneousmirror.org/faq/category6.php3>

Re: LAM/MPI MPICH-2 Compatibility

which says that, for example, the first "process is dead" message means that some process tried to receive from process 1 and couldn't, because process 1 had ended already. I'm not sure that's entirely consistent with the messages about call stack, which seem to indicate that process 1 was trying to do an MPI_Recv and failed because the process to be received from had ended. But surely it's one or the other, and maybe this will help a bit in narrowing down the problem?

Is it something to do with my usage of
MPI_ANY_SOURCE?

I don't spot anything obviously wrong in the calls that use MPI_ANY_SOURCE. At first I thought maybe it didn't make sense to use this when your program logic appears to need to be able to distinguish between messages, but then I noticed that you're using tags for that. So okay.

I guess my suggestion at this point would be to try, as best you can, to be sure that all the sends and receives match up — i.e., for every MPI_Send there's exactly one corresponding MPI_Recv, and vice versa. If it were my code, and a bit of rethinking about matching sends/receives didn't find the problem, I'd start putting in debug print statements to try to trace all sends and receives.

Hope this helps, and good luck.

--

B. L. Massingill

ObDisclaimer: I don't speak for my employers; they return the favor.

.