IBM Parallel Environment

Pidad D'Souza (pidsouza@in.ibm.com)
IBM, System & Technology Group

Parallel Operating Environment (POE)

- Software for developing and executing the parallel applications across multiple operating system images, called **nodes**. The node from which POE is launched is called **home node**.
- Transparently manages the allocation of remote nodes where the parallel application actually runs (With the help of Hostfile).
- Handles the various requests and communication between the home node and the remote nodes via the underlying network.
- Eases the transition from serial to parallel programming by hiding the differences, and allowing you to continue using standard Linux tools and techniques.
- The *processor node* is a physical entity or operating system image that is defined to the network. It can be a standalone machine, or a processor node within a cluster, or an SMP node. From POE's point of view, a node is a single copy of the operating system.

Before You Start POE

- **Access** - You must have the ***same*** **user ID and group ID on the home node and each remote node** on which you will be running the parallel application. (Does not allow running application as root).

- **User authorization -** You must have remote execution authority on all the nodes in the system that you will use for parallel execution. Either one of the following must be done:

  - **/etc/hosts.equiv file:** Authorize both the home node machine and the user name (or machine names) in this file on each remote node.

  - **~/.rhosts file:** Create this file in the home directory of the user ID for each node that you want to use, containing either the explicit IP address of the home node, or the home node name.

  - Test if you can login between nodes without password.

Before You Start POE

- **Hostlist File:** Helps POE to know over which nodes to run the program. **Default location is current working directory**, but can be pointed via **MP_HOSTFILE** environment variable or **"-hfile"** command line option. **Default name is host.list**.

  For running 4 tasks, a sample hostfile:

  $cat host.list
  nodeA.ibm.com
  nodeB.ibm.com
  nodeC.ibm.com
  nodeD.ibm.com

  Total entries >= Total Tasks

- Set environment variable "LANG" as "en_US".

- Set NLSPATH environment variable as "/opt/ibmhpc/ppe.poe/%L/%N" (assuming base dir as "/opt/ibmhpc/ppe.poe)

Running POE

The **poe** command enables you to load and execute programs on remote nodes. The syntax is:

poe [program] [options]

When you invoke **poe**:

- It allocates processor nodes for each task.
- Initializes the local environment.
- It then loads the program and reproduces the local shell environment on each processor node.
- POE also passes the user program arguments to each remote node.

Running POE – Some Examples

```
$ poe hostname -procs 4
nodeB.ibm.com
nodeC.ibm.com
nodeA.ibm.com
nodeD.ibm.com

$ poe hostname -procs 4 –labelio yes
1: nodeB.ibm.com
2: nodeC.ibm.com
0: nodeA.ibm.com
3: nodeD.ibm.com

$ poe hostname -procs 4 –labelio yes -stdoutmode ordered
0: nodeA.ibm.com
1: nodeB.ibm.com
2: nodeC.ibm.com
3: nodeD.ibm.com
```

POE Options

Description	POE Options	Allowed values	Environment variables
Control number of tasks	-procs	1→N	MP_PROCS
Specify host file	-hfile or -hostfile	<host file name>	MP_HOSTFILE
Label I/O with task numbers	-labelio	yes/no	MP_LABLEIO
Specify the level of messages needed from POE	-ilevel or -infolevel	1-6	MP_INFOLEVEL
Request diagnostic messages be logged to a file in /tmp on each node used	-pmdlog	No value	MP_MDLOG
Specify output data display	-stdoutmode	ordered	MP_STDOUTMODE

Note: POE options temporarily overwrite the corresponding ENV variables

Invoking the MPI Compiler

To compile the MPI programs, invoke the appropriate compiler script:

```
$ mpcc -o hello_world_c hello_world.c

$ mpfort -o hello_world_f hello_world.f
** main === End of Compilation 1 ===
1501-510 Compilation successful for file
 hello_world.f.
```

POE scripts **mpcc**, **mpCC**, and **mpfort** link the parallel libraries that allow programs to run in parallel.

Language	Compiler
Fortran 77	mpxlf
Fortran 90	mpxlf90
Fortran 95	mpxlf95
C	mpcc
	mpCC

Quick Reference Page – Cheat Sheet
Compile and Execute an MPI program

$ mpcc mpi.c –o mpi

< create host.list file similar to the following >
$cat host.list

```
r36n11.pbm.ihost.com
r36n11.pbm.ihost.com
r36n11.pbm.ihost.com
r36n11.pbm.ihost.com
```

$ set up access permission in /etc/hosts.equiv or .rhosts

$ poe mpi –procs 4 –hostfile host.list

Or

$ export MP_PROCS=4; MP_HOSTFILE=host.list; poe mpi

$ poekill mpi.exe

Factors Affecting MPI Performance

Some Important Environment Variables

Parameter	Values	Description
MP_BUFFER_MEM	0 – 64,000,000	Buffer for early arrivals
MP_EAGER_LIMIT	0 - 262144	Threshold for rendezvous protocol
MP_SHARED_MEMORY	{yes,no}	Use of shared memory on node
MP_USE_BULK_XFER	yes, no	Block Transfer: message striping
MP_EUILIB	{us,ip}	Communication Method

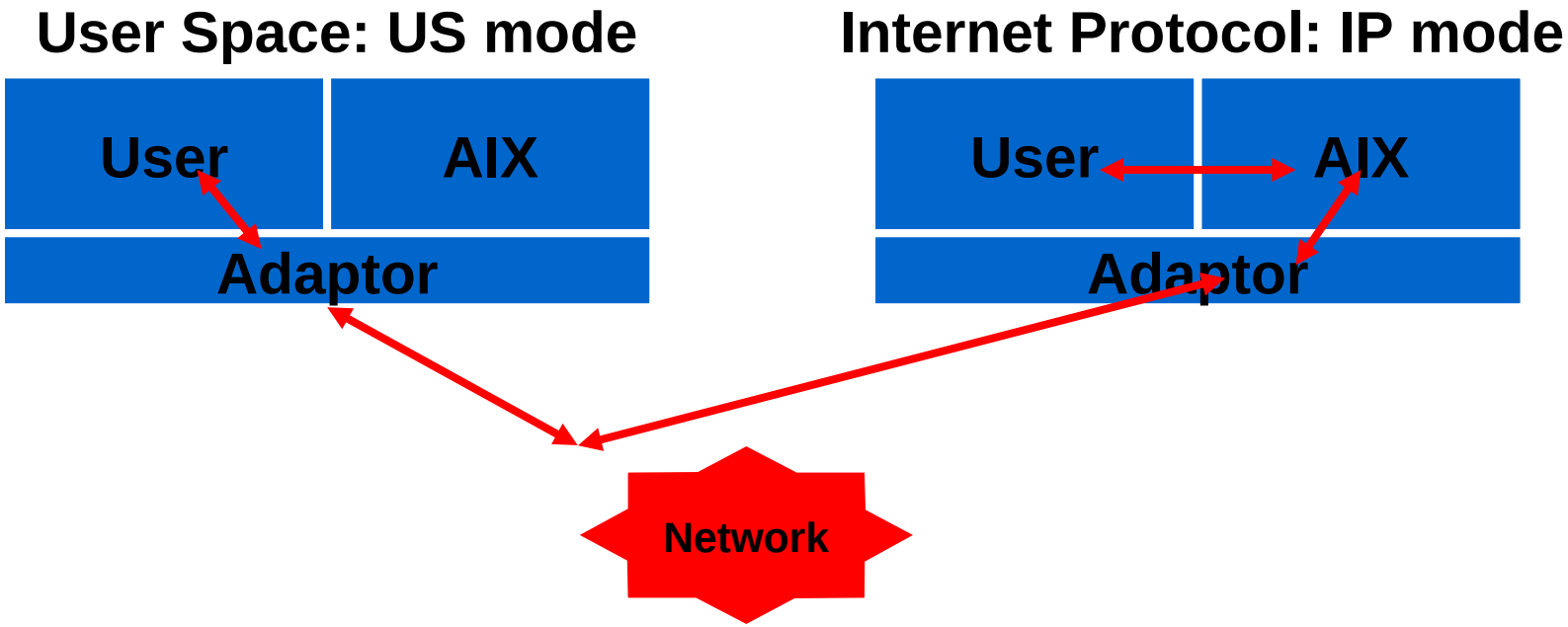
Number of Tasks (processors)

- MP_PROCS=MP_NODES* MP_TASKS_PER_NODE
  - MP_PROCS : Total number of processes
  - MP_NODES : Number of nodes to use
  - MP_TASKS_PER_NODE : number of proc. per node
- Any two variables can be specified
  - MP_TASKS_PER_NODE is (usually) the number of processors per node

Message Passing Library

- MP_EUILIB={us,ip}
  - us: user space
    - Much faster: 5 microseconds latency, 2000 Mbyte/s bandwidth
  - ip: useable with ethernet
    - Much slower: 50 microsecond latency
- US mode is usually default

MP_EUILIB


User Space: US mode
User
AIX
Adaptor
Internet Protocol: IP mode
User
AIX
Adaptor
Network

Other MPI Tuning

Env. Variable	Values	Comment
MP_SINGLE_THREAD	yes, no	For Non-threaded Application and those who do not use the nonstandard MPE_Ixxx non-blocking collectives, MPI-IO.

Setting	Values	Comment
Address Mode	-q32 -q64	Enhanced MPI collectives performance with 64-bit addresses

Generating Light Weight Core Files

- In large scale parallel processing a core file is generally very huge, which takes available disk space. In being written out, theses core files can take up an unacceptable amount of CPU time and network bandwidth.

- *Standardized Lightweight Corefile Format* (LCF) from Parallel Tools Consortium (a collaborative body of parallel-programming researchers, developers, and users from governmental, industrial, and academic sectors)

- export **MP_COREFILE_FORMAT=**my*_light_corefile_name*
  OR
  poe *program* ***-corefile_format*** my*_light_corefile_name*

- Lightweight core file only contains necessary thread stack traces.

- Change the default corefile directory name by **MP_COREDIR** env.

- *Don't Forget to run a binary created with "-g" option to MPI compiler*.

Use of Multiple Program Multiple Data (MPMD)

- Each task in MPI session can be a unique program:
  - export MP_PGMMODEL=<mpmd/spmd>
  - export MP_CMDFILE=cmdfile

cmdfile
a.out
b.out
c.out

Host File
node1
node2
node3

Execution command:
$ MP_PGMMODEL=mpmd
$ MP_CMDFILE=cmdfile
$ poe -procs 3

MDMP – Sample Program

- **$ cat a.c**
- #include<stdio.h>
- #include<mpi.h>
- int main(int argc, char **argv) {
- MPI_Init(&argc,&argv);
- printf("HI, My exe name = %s\n",argv[0]);
- MPI_Finalize();
- return 0;
- }

- **$ cat b.c**
- #include<stdio.h>
- #include<mpi.h>
- int main(int argc, char **argv) {
- MPI_Init(&argc,&argv);
- printf("HI, My exe name = %s\n",argv[0]);
- MPI_Finalize();
- return 0;
- }

MDMP - Sample

- **$ cat cmdfilename**

a.out
b.out

- Compiling:

$ mpcc a.c -o a.out
$ mpcc b.c -o b.out

- Running:

$ poe -pgmmodel mpmd -cmdfile cmdfilename
0:HI, My exe name = a.out
1:HI, My exe name = b.out

Environment, Statistics and Information

Env. Variable	Values	Comment
MP_PRINTENV	yes, no	Echo environment Variables
MP_STATISTICS	yes, no	Low level statistics
MP_INFOLEVEL	{0,1,2,3,4,5,6}	Information Warnings Errors

MPI Environment Variables – MP_PRINTENV

```
0:Task    0-   1:Library: 32bit(ip)  ppe_rorl MPCI_MSG: MPI/MPCI library
 was compiled on   Tue Jan 19 08:16:10 2010
0:Task    0-   1:Hostname: z25c4s3.ppd.pok.ibm.com
0:Task    0-   1:Job ID (MP_PARTITION): 1264719621
0:Task    0-   1:Number of Tasks (MP_PROCS): 2
0:Task    0-   1:Number of Nodes (MP_NODES): NOT SET
0:Task    0-   1:Number of Tasks per Node (MP_TASKS_PER_NODE): NOT
 SET
0:Task    0-   1:Library Specifier (MP_EUILIB): ip
0:Task    0-   1:Adapter Name: ib
0:Task    0-   1:IP Address: ::ffff:9.114.247.110
0:Task    0-   1:Window ID: NA
0:Task    0-   1:Device Name (MP_EUIDEVICE): not_spe
0:Task    0-   1:Window Instances (MP_INSTANCES * # of networks): 1
0:Task    0-   1:Striping Setup: Striping off
0:Task    0-   1:Protocols in Use (MP_MSG_API): mpi
0:Task    0-   1:Effective Libpath (LIBPATH):
 /u/vivekkum/debug:/usr/lpp/ppe.poe/lib
0:Task    0-   1:Current Directory: /u/vivekkum/mpi/mpi_wrkshp/cc
0:Task    0-   1:64 Bit Mode: NO
0:Task    0-   1:Threaded Library: YES
```