Using PVM on the T3E

Youngbae Kim

NERSC
Lawrence Berkeley National Laboratory
youngbae@nersc.gov

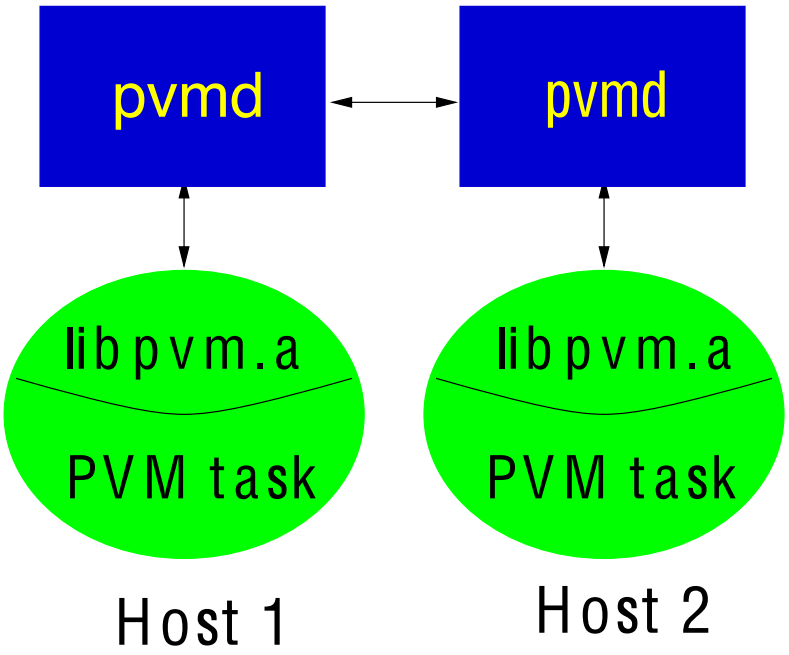
PVM (Parallel Virtual Machine)

- A software package to enable a collection of heterogeneous computer systems to be used as a virtual parallel computing resource
- A widely-used, de facto standard method of programming such a parallel virtual machine.
- Available as a public domain software
- Also available as a vendor-specific software
- widely available on various platforms

PVM Overview

- Configure a virtual machine (VM) with a collection of computers where you want to execute a parallel application
- An individual computer system is viewed as a host.
- Any combination of these hosts can be treated as a single VM to execute a parallel application.
- Write a parallel application using PVM library routines.
- Execute the parallel application on the virtual machine.
- Tasks are units of computation, possibly a Unix process.
- Tasks communicate with each other by explicit message passing.
- Supply the functions to automatically start up tasks on the VM and to allow the tasks to communicate.

PVM Overview (cont'd)



- PVM daemon (pvmd)
  - one daemon for each host
  - message router and controller
  - process control, fault tolerance
  - authentication, reconfiguration
- PVM libraries: subroutines
  - `libpvm.a`, `libfpvm3.a`, `libgpvm3.a`
  - task initiation, message passing
  - pack/unpack
  - synchronization, communication
  - dynamic configuration of tasks
  - data conversion (XDR)

T3E Implementation of PVM

- Cray own proprietary implementation based on PVM 3.3.10
- Support MPP architecture
- Operate in two different modes
  - Stand-alone mode
  - Distributed mode
- Interoperable with the generic PVM
- Available as a component of the MPT package

```
module load mpt
```

Stand-alone Mode of T3E PVM

- Used as another message passing library within a single executable like MPI
- No PVM daemons – no process management
- SPMD (Single Program Multiple Data)
- Simply execute a parallel program on a partition of application PEs.
- Communicate among PEs within the same partition.
- Optimized for the T3E.
- Use SHMEM for communication – fast
- A predefined group (called `global group`) of all PEs within the same partition
- PVMALL is used in Fortran for the global group
- Any group allowed within the same partition

A PVM Example


```
PROGRAM PP1
INCLUDE 'fpvm3.h'

INTEGER MY_TID, ME, INFO, NPROC

CALL PVMFMYTID (MY_TID)
!CALL PVMFGETPE (MY_TID, ME)
CALL PVMFJOINGROUP (PVMALL, ME)

CALL PVMFBARRIER(PVMALL,NPROC,INFO)
CALL PINGPONG(ME)
CALL PVMFEXIT(INFO)

END PROGRAM PP1
```

```
PROGRAM PP2
INCLUDE 'fpvm3.h'

INTEGER MY_TID, ME, TIDS(16), NPROC, INFO

CALL PVMFMYTID (MY_TID)
CALL PVMFJOINGROUP ('foo', ME)
IF (ME .EQ. 0) THEN
  READ *, NPROC
  CALL PVMFSPAWN ('pp2', PvmTaskArch, '*',
                  NPROC-1, TIDS, INFO)
ENDIF

CALL PVMFBARRIER('foo', NPROC, INFO)
CALL PINGPONG(ME)
CALL PVMFEXIT(INFO)

END PROGRAM PP2
```

A PVM Example (cont'd)


```
SUBROUTINE PINGPONG(ME)
INCLUDE 'fpvm3.h'

INTEGER ME, THE_OTHER, ISTAT
INTEGER, PARAMETER :: MSG_TAG = 99
IF (ME .EQ. 0) THEN
  CALL PVMFINITSEND (PVMDATARAW, ISTAT)
  CALL PVMFPACK (INTEGER8, ME, 1, 1, ISTAT)
  CALL PVMFSEND (1, MSG_TAG, ISTAT)
  CALL PVMFRECV(1, MSG_TAG, ISTAT)
  CALL PVMFUNPACK (INTEGER8, THE_OTHER, 1, 1, ISTAT)
  PRINT *, 'PE ', ME, ' received ', THE_OTHER
ELSE IF (ME .EQ. 1) THEN
  CALL PVMFINITSEND (PVMDATARAW, ISTAT)
  CALL PVMFPACK (INTEGER8, ME, 1, 1, ISTAT)
  CALL PVMFSEND (0, MSG_TAG, ISTAT)
  CALL PVMFRECV(0, MSG_TAG, ISTAT)
  CALL PVMFUNPACK (INTEGER8, THE_OTHER, 1, 1, ISTAT)
  PRINT *, 'PE ', ME, ' received ', THE_OTHER
ENDIF
RETURN
END SUBROUTINE PINGPONG
```

Distributed Mode of T3E PVM

- require a PVM daemon running
- Allow to spawn more than one executable within the T3E
- Use `pvm_spawn` calls

  ```
  call pvmfspawn(/u1/youngbae/pvm3/examples/xslave',
  PvmTaskArch, 'CRAY', NPROC, tids(0), numt)
  ```

- Allow to configure a VM that includes the T3E and other systems
- Use sockets for communication – slow

  between T3E processes that were not started at the same time

  between the T3E and other systems
- May have several sockets open at once
- Limits the number of open files per application and the number of open sockets in the system.

Distributed Mode (cont'd)

- By default only PE 0 can communicate w/ processes outside the T3E or between processes inside the T3E

  `setenv PVM_PE_LIST all` # all PEs communicate

  N.B. This setting should be done before a PVM daemon is started.

- Use SHMEM for communication between processes started in the same `pvm_spawn` call.

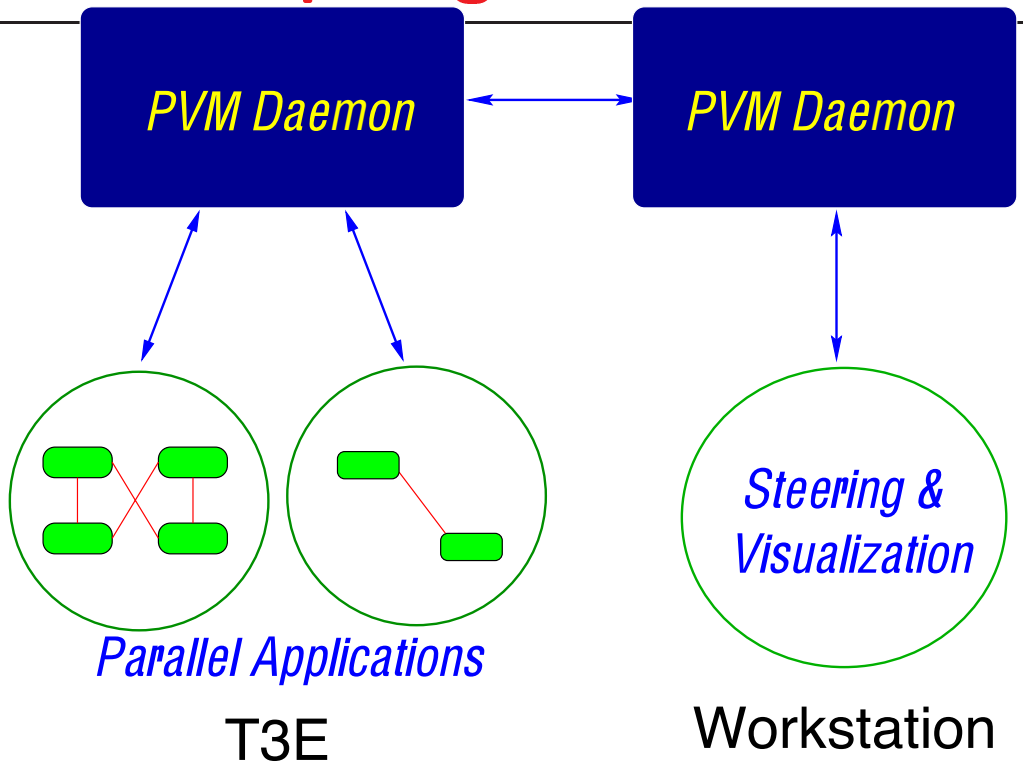
  fast, but slower than in stand-alone mode due to considerable overhead

- No dynamic group that spans different partitions and processes outside the T3E

- Any groups allowed within the same process

- Decide automatically in which mode a PVM program run by checking if pvmd is running.

A Computing Model with PVM



1. Setting up a VM including the T3E and other hosts
   - Starting up local (master) and remote (slave) PVM daemons
   - Use remote shell like `rsh` to start remote daemons
2. Starting local and remote processes

Virtual Machine Setup

1. Using the PVM console on any host and 'add *host*'

```
[pierre.55 ] pvm
3.3.10 (Cray PVM for UNICOS Version 3.1.x.6)
t40001
pvm> add dolly.lbl.gov louis.lbl.gov
2 successful
                  HOST     DTID
         dolly.lbl.gov    80000
         louis.lbl.gov    c0000
pvm> conf  # show the current configuration
3 hosts, 2 data formats
                HOST     DTID     ARCH   SPEED
              pierre    40000     CRAY    1000
       dolly.lbl.gov    80000    SUNMP    1000
       louis.lbl.gov    c0000    SUNMP    1000
pvm> quit               # quit the PVM console
pvmd still running.
[pierre.56 ]
```

Virtual Machine Setup (cont'd)

2. Using a host file w/ an entry per each host

```
[pierre.57 ] cat hostfile
dolly.lbl.gov dx=$PVM_ROOT/lib/pvmd
louis.lbl.gov dx=$PVM_ROOT/lib/pvmd
[pierre.58 ] pvmd hostfile &    # run pvmds
[1] 68001
socket address: /tmp/jtmp.008381a/aaa0000a68001
[pierre.58 ] pvm
pvmd already running.
3.3.10 (Cray PVM for UNICOS Version 3.1.x.6)
t40001
pvm> conf
3 hosts, 2 data formats
                HOST     DTID     ARCH   SPEED
              pierre    40000     CRAY    1000
       dolly.lbl.gov    80000    SUNMP    1000
       louis.lbl.gov    c0000    SUNMP    1000
                HOST      TID   FLAG 0x COMMAND
pvm> quit
pvmd still running.
```

3. Calling `pvm_addhosts()` in a PVM program

VM Setup including the T3E

- Normal VM setup doesn't work on the T3E
  - Incoming remote shell is not permitted for security reasons
  - Outgoing remote shell is permitted
- Other methods of VM setup on the T3E
  1. Using the T3E as a master host
     - start the master daemon on the T3E first
     - start a slave daemon on remote host
  2. Starting daemons by hand
     - use `so=ms` option in the host file
     - need an interactive login session for each host
     - work on any VM configuration
  3. Using `ssh`
     - ssh into the T3E is permitted within the LBL domain
     - ssh from the T3E is not supported yet.

A Manual Startup of PVM Daemons

- On a local workstation (dolly)

```
[dolly.122.~] cat hostfile
dolly.lbl.gov    so=ms
pierre.nersc.gov so=ms
[dolly.123.~] pvmd hostfile
7f000001:a467
*** Manual startup ***
Login to "pierre.nersc.gov" and type:
$PVM_ROOT/lib/pvmd -S -d0 -npierre.nersc.gov \
                  1 83f3f0e6:c467 4096 2 8037c86b:0000}
Type response: ddpro<2315> arch<CRAY> ip<8037c86b:057e> mtu<32768>
Thanks
```

- On the T3E (pierre)

```
[pierre.1.~ ] $PVM_ROOT/lib/pvmd -S -d0 -npierre.nersc.gov \
                  1 83f3f0e6:c467 4096 3 8037c82e:0000
ddpro<2315> arch<CRAY> ip<8037c82e:132f> mtu<32768>
[pierre.2.~ ]
```

Using ssh: currently not supported

- On the T3E

  `setenv PVM_RSH /usr/local/bin/ssh`

- On your local workstations

  Re-compile the PVM source code

  by setting `RSHCOMMAND` to the full path of ssh

  in the `$PVM_ARCH.def` file

- This also works if you want to use a different remote shell

Using PVM under NQS

- Works fine
  - in stand-alone mode (no pvmd)
  - in distributed mode (w/ pvmd running inside batch job)
    only when processes are spawned in the same batch job
    without adding any hosts
- PVM works differently in batch mode
  if you add hosts in the same batch job
  if you connect to the running pvmd from outside the batch job.
- There is a serious concern with T3E scheduling
  a batch job reserve # of PEs
  run `pvmd` inside the batch job
  do nothing until any PVM job started
- Recommend not to use PVM daemons under NQS

Why Not Use PVM?

- Moving target, overriden by MPI
- PVM vs. MPI
  - MPI is message passing standard
  - MPI has more functionality
  - MPI-2 specification released and being implemented on various architectures
    - MPI_SPAWN to start both MPI and non-MPI processes
    - One-sided communication such as put and get
    - Nonblocking collective communication
    - Parallel I/O
    - Language bindings
  - Invest the time and effort to write codes in MPI
  - Recommend MPI for communication within the T3E

Why Use PVM?

- Distributed computing in a heterogeneous environment
  - virtual machine concept
  - dynamic resource management and process control
  - support for heterogeneity
  - interoperability
- Recommend PVM for communication with processes outside the T3E

More Information on the T3E PVM

- Message Passing Toolkit: PVM Programmer's Manual
- CRAY T3E Fortran Optimization Guide
- On-line Documentations – `xhelp`, `man page`, `dynaweb`
- PVM: A Users' Guide and Tutorial for Networked Parallel Computing

  `http://www.epm.ornl.gov/pvm`