PVM & MPI program – rectangular integration

This is a brief description of a relatively basic and exceptionally useless introductory parallel application. The gist of things is to let you write an application yourself that makes use of these clusters you're building. You can use your personal clusters but probably you'll want to use XTORC because it has more than one compute node, (grin).

The chapters in [1] will be helpful on how to compile and execute PVM and MPI (MPICH or LAM/MPI) programs. Basically you will want to write a master/slave, similar to the example to sum numbers that I'm including. The slaves will calculate a single portion of the problem and then relay that back to the master who will sum the results and print them for the user.

Problem Details

- Calculate the following integral using rectangular integration.

$$\int_0^{\Pi} \sqrt{\sin^2(\frac{x}{2})}\ dx$$

  Ok, breathe. I'm not a math person but the idea is that you have this curve from $0 \rightarrow \Pi$, and you divide it up into N equal slices and calculate the area for each slice, (see also: Figure 1). So, basically you just have to calculate the area for a rectangle (height * width). The only trick is you have to determine what portion of the curve you "are". Then send this area back to the master (e.g. `pvm_parent()`), who will receive the results in any order and sum them for a total which is printed to the screen.

  The only input is the number of slices, which should probably correspond to the number of tasks spawned. So, if you used XTORC you might spawn 64 tasks, one per node and let that node calculate it's portion, all being done in parallel.

- Write this program in both PVM and MPI (either MPICH or LAM/MPI...your choice).
- The heart of the calculation will be something like this:

```
width  = M_PI / N;     /* N is number of slices/intervals */
mid    = (me + 0.5) * width;
height = sqrt( (3 * sin(mid/2)) * (3 * sin(mid/2)) );
area   = height * width;
```

PVM Account Setup

- I've not tested the MPI version yet so you'll probably want to start with the PVM version.
- If you are not using OSCAR-1.3, the case on XTORC, you'll have to add the following to your local BASH shell config file, `~/.bashrc`. Just append to the end, and either `source .bashrc` or logout/login.

```
export PVM_RSH=/usr/bin/ssh

if [ "$PVM_ROOT" = "" ]; then
        export PVM_ROOT=/opt/pvm3
        export PVM_ARCH=`$PVM_ROOT/lib/pvmgetarch` # use backticks

        export PATH=$PATH:$PVM_ROOT/lib            # generic
        export PATH=$PATH:$PVM_ROOT/lib/$PVM_ARCH  # arch-specific
        export PATH=$PATH:$PVM_ROOT/bin/$PVM_ARCH
fi
```

- Also, you will need to create the following directory (a defualt) where you will place the binaries that are to be run. (You might want to add the copy to this dir to the 'Makefile'.)

```
[tjn@xtorc] $ mkdir -p ~/pvm3/bin/LINUX
```

- I also find it helpful to use a hostfile (file w/ hostnames one per line) and feed that to `pvm` on the command-line as opposed to adding it to the src file or manually typing `add node1 node2`, etc.

```
[tjn@xtorc] $ cat mynodes
node1
node2
node3
[tjn@xtorc] $ pvm mynodes
pvm> conf
conf
4 hosts, 1 data format
                  HOST     DTID     ARCH   SPEED       DSIG
                 node0    40000    LINUX    1000 0x00408841
                 node1    80000    LINUX    1000 0x00408841
                 node2    c0000    LINUX    1000 0x00408841
                 node3   100000    LINUX    1000 0x00408841
pvm> quit
quit
Console: exit handler called
pvmd still running.
[tjn@xtorc] $ pushd ~/pvm3/bin/LINUX
[tjn@xtorc] $ ./master_rect
Enter the number of intervals (<=100): 55
Total area:  6.000204
[tjn@xtorc] $ pvm
pvmd already running.
pvm> halt
halt
Terminated
[tjn@xtorc] $ popd
[tjn@xtorc] $
```

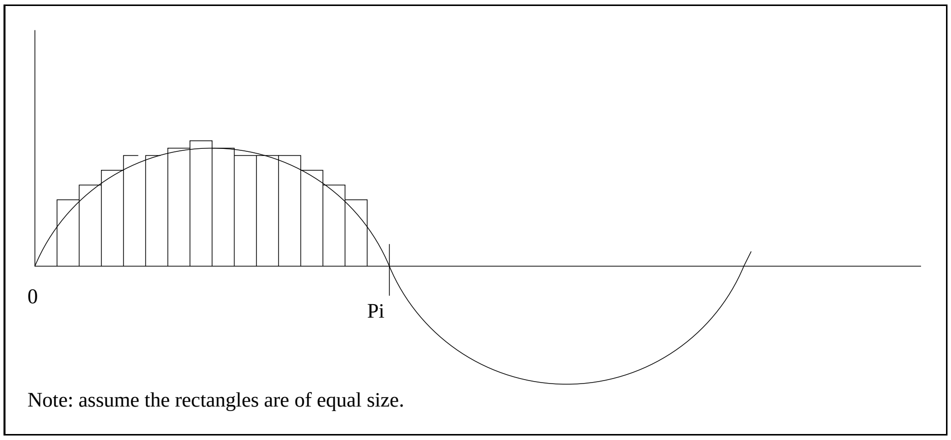

Figure 1: Basic idea of what the curve looks like, and the portion where rectangles will be calculated.

References

[1] Thomas Sterling (Editor) et al. *Beowulf Cluster Computing with Linux.* Scientific and Engineering Computation Series. MIT Press, 2002.

[2] Course notes from: *CSCI-533: Parallel Processing Concepts*, Chrisila Pettey, MTSU Spring 2000.