CS596 HW3

Chieh Wen Chang 4155100480

README:

pmd_m.c is the modified version of pmd.c

pmd_m.h is the header file I use for both pmd.c & pmd_m.c

I changed pmd.c to let it use pmd_m.h as its header and read pmd_m.in as input.

REPORT:

```
chiehwen@hpc-login2:~/work596
[chiehwen@hpc-login2 work596]$ cat pmd_m.out
----------------------------------------
Begin PBS Prologue Tue Oct  7 00:54:46 PDT 2014
Job ID:            9766660.hpc-pbs.hpcc.usc.edu
Username:          chiehwen
Group:             csci-ar
Name:              pmd_m
Queue:             quick
Shared Access:     no
All Cores:         no
Nodes:             hpc1118 hpc1120 hpc1121 hpc1122
PVFS:              /scratch (3.4T), /staging (328T)
TMPDIR:            /tmp/9766660.hpc-pbs.hpcc.usc.edu
End PBS Prologue Tue Oct  7 00:55:02 PDT 2014
----------------------------------------
*****Asynchronous*****
al = 5.129928e+00 5.129928e+00 5.129928e+00
lc = 2 2 2
rc = 2.564964e+00 2.564964e+00 2.564964e+00
nglob = 1728
CPU & COMT = 5.551132e+00 3.246959e+00
******Synchronous******
al = 5.129928e+00 5.129928e+00 5.129928e+00
lc = 2 2 2
rc = 2.564964e+00 2.564964e+00 2.564964e+00
nglob = 1728
CPU & COMT = 6.208243e+00 3.929975e+00
----------------------------------------
Begin PBS Epilogue Tue Oct  7 00:55:17 PDT 2014
Job ID:           9766660.hpc-pbs.hpcc.usc.edu
Username:         chiehwen
Group:            csci-ar
Job Name:         pmd_m
Session:          4057
Limits:           neednodes=4:ppn=4,nodes=4:ppn=4,walltime=00:02:00
Resources:        cput=00:00:16,mem=160512kb,vmem=2437912kb,walltime=00:00:15
Queue:            quick
Shared Access:    no
Account:          lc_an2
End PBS Epilogue Tue Oct  7 00:55:23 PDT 2014
----------------------------------------
[chiehwen@hpc-login2 work596]$
```

Asynchronous is faster.
Synchronous version needs to wait to recv then send. Even though we implemented it to avoid deadlock, the recv will still wait a bit time. But Asynchronous version use Irecv, which uses MPI_request. It automatically avoids deadlock and keeps waiting to receive message then send. So it is faster.