Assignment 5

CPSC424

Jake Brawer

April 25, 2017

1 Software and Development Environment

All the programming for this assignment was done in vim. This document was made using emacs. The only modules used in this assignment were Langs/Intel/15 and MPI/OpenMPI/1.8.6-intel15.

1.1 How to run the code

To compile the code and load the appropriate modules do the following:

```
cd jnb37_ps5_cpsc424
sh setup.sh
```

In order to run code, run

```
qsub run_task_<n>.sh
```

where n is the the task number

2 Data

2.1 Task 0

Flag	Avg GFLOPS
-O0	0.4 +- 0.0
-O3	3.7 +- 0.0

2.2 Task 1

I was able to achieve an average performance of 9.5 GFLOPS. I was able to increase performance mainly by 1) cutting down on divisions (by using multiplication by inverse) 2) taking an inverse squareroot. A smart compiler would use the inverse squareroot algorithm, which uses very clever bitshifting to approximate an inverse squareroot.

2.3 Task 2

Threads	Avg GFLOPS
1	11.5 +- 0.0
2	22.8 +- 0.0
4	45.3 +- 0.2
8	89.1 +- 0.2

I found this task to be the most challenging. I originally only parallelized the inner loop of moveBodies() (using reductions), which gave correct results, but was not very fast. It took me many hours to realize that I should try parallelizing the outer loop. In the former case, each thread is doing a smaller task, but there are many such tasks. In the latter, there are fewer tasks but they are larger (each thread has to loop through all N bodies). Clearly the latter case is preferable.

2.4 Task 3

With 1 Core

N	Avg. GFLOPS
2048	52.6 +- 13.3
4096	82.1 +- 3.1
8192	94.0 +-0.0
16384	92.8 +- 4.2
32768	90.6 +- 0.8

With 8 Cores

N	Avg. GFLOPS
2048	54.2 +- 13.3
4096	80.7 +- 3.1
8192	94.1 +-.2
16384	94.7 +- 0.1
32768	90.6 +-1

It's fairly surprising that the 1 core and 8 core performances are so close. This could be in part because each thread is accessing fairly contiguous memory, which means that there would be some benefit to sharing L1 and L2 caches.

2.5 Task 4

With 2 tiles

N	Avg. GFLOPS
2048	39.7 +- 1.6
4096	43.8 +- 0.1
8192	44.3 +-0.1
16384	44.3 +- 0.1
32768	39.1 +- 1.7

With 4 tiles

N	Avg. GFLOPS
2048	39.1 +- 1.7
4096	43.9 +- 0.1
8192	44.3 +-0.1
16384	43.9 +- 1.0
32768	44.2 +- 0.2

With 8 tiles

N	Avg. GFLOPS
2048	39.7 +- 1.6
4096	43.6 +- 0.2
8192	44.2 +-0.1
16384	44.3 +- 1.0
32768	44.2 +- 0.2

With 16 tiles

N	Avg. GFLOPS
2048	39.9 +- 2.7
4096	38.5 +- 7.0
8192	44.3 +-0.1
16384	44.3 +- .8
32768	44.2 +- 0.2

I'm not sure why this code is not giving me the expected performance. However, I found it interesting that the performance was the worst for $N = 2048$ Im assuming this is because there is a lot of overhead associated with parallelizing the code, and therefore bigger N's are required to justify that initial price.

3 Environment

```
MKLROOT=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/mkl
MANPATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/share/man:/home/apps/fas/Langs
GDB_HOST=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugger/gdb/in
HOSTNAME=compute-33-1.local
IPPROOT=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/ipp
INTEL_LICENSE_FILE=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/licen
TERM=xterm
SHELL=/bin/bash
HISTSIZE=1000
GDBSERVER_MIC=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugger/g
SSH_CLIENT=10.191.63.252 36982 22
LIBRARY_PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/lib:/home/apps/fas/Langs/
PERL5LIB=/opt/moab/lib/perl5
FPATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/include:/home/apps/fas/Langs/Int
QTDIR=/usr/lib64/qt-3.3
OLDPWD=/home/fas/cpsc424/jnb37/scratch
QTINC=/usr/lib64/qt-3.3/include
MIC_LD_LIBRARY_PATH=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/mpir
SSH_TTY=/dev/pts/1
ANT_HOME=/opt/rocks
USER=jnb37
LD_LIBRARY_PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/lib:/home/apps/fas/Lan
5_update2/composer_xe_2015.2.164/compiler/lib/intel64:/home/apps/fas/Langs/Intel/2015_u
MIC_LIBRARY_PATH=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/compile
ROCKS_ROOT=/opt/rocks
CPATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/include:/home/apps/fas/Langs/Int
YHPC_COMPILER=Intel
OMPI_MCA_orte_precondition_transports=f20cd2d28f432704-15e3f8c3bb8e89d6
NLSPATH=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/compiler/lib/int
MAIL=/var/spool/mail/jnb37
PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/bin:/home/apps/fas/Langs/Intel/20
```

```
YHPC_COMPILER_MINOR=164
TBBROOT=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/tbb
C_INCLUDE_PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/include
F90=ifort
PWD=/home/fas/cpsc424/jnb37/scratch/jnb37_ps5_cpsc424_NEW
_LMFILES_=/home/apps/fas/Modules/Base/yale_hpc:/home/apps/fas/Modules/Langs/Intel/15:/h
YHPC_COMPILER_MAJOR=2
JAVA_HOME=/usr/java/latest
GDB_CROSS=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugger/gdb/i
DOMAIN=omega
LANG=en_US.iso885915
MODULEPATH=/home/apps/fas/Modules
MOABHOMEDIR=/opt/moab
TBBROOT=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/tbb
C_INCLUDE_PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/include
F90=ifort
PWD=/home/fas/cpsc424/jnb37/scratch/jnb37_ps5_cpsc424_NEW
_LMFILES_=/home/apps/fas/Modules/Base/yale_hpc:/home/apps/fas/Modules/Langs/Intel/15:/h
YHPC_COMPILER_MAJOR=2
JAVA_HOME=/usr/java/latest
GDB_CROSS=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugger/gdb/i
DOMAIN=omega
LANG=en_US.iso885915
MODULEPATH=/home/apps/fas/Modules
MOABHOMEDIR=/opt/moab
YHPC_COMPILER_RELEASE=2015
LOADEDMODULES=Base/yale_hpc:Langs/Intel/15:MPI/OpenMPI/1.8.6-intel15
KDEDIRS=/usr
F77=ifort
MPM_LAUNCHER=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugger/mp
CXX=icpc
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
HISTCONTROL=ignoredups
INTEL_PYTHONHOME=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugge
SHLVL=1
HOME=/home/fas/cpsc424/jnb37
FC=ifort
LOGNAME=jnb37
QTLIB=/usr/lib64/qt-3.3/lib
CVS_RSH=ssh
```

```
SSH_CONNECTION=10.191.63.252 36982 10.191.12.33 22
MODULESHOME=/usr/share/Modules
LESSOPEN=||/usr/bin/lesspipe.sh %s
arch=intel64
INFOPATH=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugger/gdb/i
CC=icc
INCLUDE=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/mkl/include
MPI_PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15
G_BROKEN_FILENAMES=1
BASH_FUNC_module()=() {  eval ‘/usr/bin/modulecmd bash $*‘
}
_=/bin/env
```