Assignment 6

CPSC424

Jake Brawer

May 7, 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 GPU/CUDA/8.0.

1.1 How to run the code

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

```
cd jnb37_ps6_cpsc424
sh setup.sh
```

In order to run code, run

```
qsub runTask<N><subtask>.sh
```

Where N is the task (1-3) and *subtask* is a subtask (a-c).

2 Task 1

2.1 A

	Run 1	Run 2	Run 3	Run 4
Time	37 ms	18589 ms	305 ms	2292 ms
GPU	M2070	M2070	M2070	M2070
Block Dim	32	32	32	32
Grid X	32	256	32	256
Grid Y	32	256	32	256

Unfortunately my code segfaulted for run 5 so I could not get timings. Here I chose my Grid Xs and Grid Ys as a function Block Dim (e.g. Grid X = n / Block Dim.

2.2 B

GPU:M2070

Task B took ~28645 ms, a little more than twice as long than the single precision run. This is not surprising given that double is twice as long as a float and therefore takes up more cache space.

2.3 C

The largest matrices I was able to multiple together without any clear sort of error had the following dimension: $n = 32768, m = 32768, p = 8192$

3 Task 2

3.1 A

	Run 1	Run 2	Run 3	Run 4
Time	16 ms	8301 ms	129 ms	1042 ms
GPU	M2070	M2070	M2070	M2070
Block Dim	32	32	32	32
Grid X	32	256	32	256
Grid Y	32	256	32	256

Unfortunately my code segfaulted for run 5 so I could not get timings. Unsurprisingly, The tiling algorithm is much quicker than the untiled algorithm; the tiling algorithm is doing far fewer reads from main memory which causes considerable speed up.

3.2 B

GPU:M2070

TaskB took ~11221 ms, slower than floating point version of the tiling algorithm, but considerably faster than the floating point untiled version. Clearly, less shared memory is still better than no shared memory.

3.3 C

The largest matrices I was able to multiple together without any clear sort of error had the following dimension: $n = 16384, m = 16384, p = 8192$

4 Task 3

I was not able to get my Task 3 code to give me the correct answers, but conceptually I think I was on the right track. Basically the idea was to add another loop in the tiling code. This loop would loop through

all adjacent B tiles and multiply them with the corresponding A tile. Afterwards, you'd read in the new a tile and repeat the process. This way your doing multiple computations with a single memory read in of your A tile.

5 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/Intel/2015_
5_update2/composer_xe_2015.2.164/debugger/gdb/intel64/share/man/:/home/apps/fas/Langs/Intel/2015_up
hare/man:/opt/moab/share/man:
GDB_HOST=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugger/gdb/intel64_mic/bi
HOSTNAME=login-0-0.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/licenses:/opt/int
TERM=xterm
SHELL=/bin/bash
HISTSIZE=1000
GDBSERVER_MIC=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugger/gdb/target/mi
SSH_CLIENT=172.27.42.118 46332 22
LIBRARY_PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/lib:/home/apps/fas/Langs/Intel/2015_u
s/Langs/Intel/2015_update2/composer_xe_2015.2.164/ipp/lib/intel64:/home/apps/fas/Langs/Intel/2015_u
/Intel/2015_update2/composer_xe_2015.2.164/mkl/lib/intel64:/home/apps/fas/Langs/Intel/2015_update2/
PERL5LIB=/opt/moab/lib/perl5
FPATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/include:/home/apps/fas/Langs/Intel/2015_upda
QTDIR=/usr/lib64/qt-3.3
QTINC=/usr/lib64/qt-3.3/include
MIC_LD_LIBRARY_PATH=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/mpirt/lib/mic:/h
c:/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/compiler/lib/mic:/home/apps/fas/La
c/coi/device-linux-release/lib:/opt/intel/mic/myo/lib:/home/apps/fas/Langs/Intel/2015_update2/compo
SSH_TTY=/dev/pts/34
ANT_HOME=/opt/rocks
USER=jnb37
LD_LIBRARY_PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/lib:/home/apps/fas/Langs/Intel/201
/Intel/2015_update2/composer_xe_2015.2.164/ipp/../compiler/lib/intel64:/home/apps/fas/Langs/Intel/2
/Intel/2015_update2/composer_xe_2015.2.164/ipp/tools/intel64/perfsys:/opt/intel/mic/coi/host-linux-
te2/composer_xe_2015.2.164/compiler/lib/intel64:/home/apps/fas/Langs/Intel/2015_update2/composer_xe
poser_xe_2015.2.164/tbb/lib/intel64/gcc4.4:/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015
MIC_LIBRARY_PATH=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/compiler/lib/mic:/h
mic:/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/tbb/lib/mic
```

```
ROCKS_ROOT=/opt/rocks
CPATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/include:/home/apps/fas/Langs/Intel/2015_upda
update2/composer_xe_2015.2.164/mkl/include:/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015
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/intel64/locale/
/ipp/lib/intel64/locale/%l_%t/%N:/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/mkl
mposer_xe_2015.2.164/debugger/gdb/intel64_mic/share/locale/%l_%t/%N:/home/apps/fas/Langs/Intel/2015
/%N
MAIL=/var/spool/mail/jnb37
PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/bin:/home/apps/fas/Langs/Intel/2015_update2/c
MAIL=/var/spool/mail/jnb37
PATH=/usr/local/cluster/hpc/MPI/OpenMPI/1.8.6-intel15/bin:/home/apps/fas/Langs/Intel/2015_update2/c
e2/composer_xe_2015.2.164/mpirt/bin/intel64:/home/apps/fas/Langs/Intel/2015_update2/composer_xe_201
date2/composer_xe_2015.2.164/debugger/gdb/intel64/bin:/home/apps/fas/Modules:/usr/lib64/qt-3.3/bin:
bin:/usr/java/latest/bin:/opt/rocks/bin:/opt/rocks/sbin:/home/apps/bin:/home/fas/cpsc424/jnb37/bin
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
_LMFILES_=/home/apps/fas/Modules/Base/yale_hpc:/home/apps/fas/Modules/Langs/Intel/15:/home/apps/fas
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/intel64_mic/b
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
F77=ifort
MPM_LAUNCHER=/home/apps/fas/Langs/Intel/2015_update2/composer_xe_2015.2.164/debugger/mpm/bin/start_
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/debugger/python/int
SHLVL=1
HOME=/home/fas/cpsc424/jnb37
FC=ifort
```

```
LOGNAME=jnb37
QTLIB=/usr/lib64/qt-3.3/lib
CVS_RSH=ssh
SSH_CONNECTION=172.27.42.118 46332 172.18.89.8 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/intel64/share/
CC=icc
DISPLAY=localhost:14.0
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
```