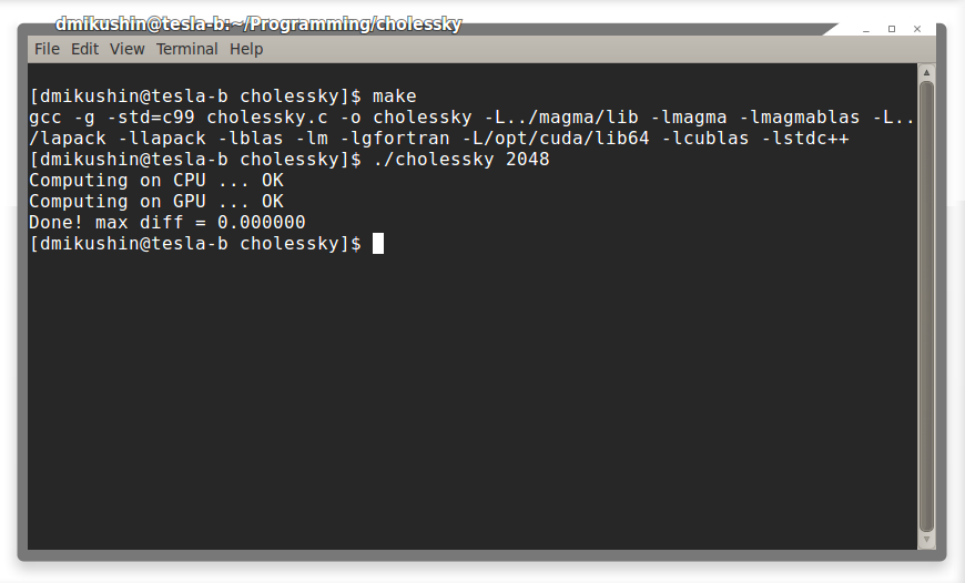
Using Cholessky decomposition on GPU



Libraries with internal support of GPUs

- CUDA-kernels invocation is performed *inside* library calls, so user does not need to use CUDA explicitly

- Data transfer between host and GPU is also performed automatically sometimes

Libraries with GPU support

- CUBLAS, CUFFT, CUSPARSE, CURAND
- PLASMA, MAGMA

...

Cholessky example pipeline

- Generate a random matrix
- Produce a matrix with diagonal domination
- Perform Cholessky decomposition on CPU (lapack)
- Perform Cholessky decomposition on GPU (magma)
- Compare the results

Compile the libraries

Source code of required libraries is already hosted on server

LAPACK

Load the source code of lapack from repository into your folder

marcusmae@teslatron:~$ svn co

http://tesla.parallel.ru/svn/lapack

Compile the library

marcusmae@teslatron:~$ cd lapack/

marcusmae@teslatron:~/lapack$ make clean && make

Compile the libraries

Source code of required libraries is already hosted on server

MAGMA

Load the source code of lapack from repository into your folder

svn co http://tesla.parallel.ru/svn/magma

Compile (**-j2** – use two compile processes in parallel)

marcusmae@teslatron:~$ cd magma/

marcusmae@teslatron:~/magma$ make clean && make -j2

Compile the sample itself

marcusmae@teslatron:~$ cd cholessky/

marcusmae@teslatron:~/cholessky$ make clean && make

Results

the correct result (GPU version and CPU version do match)

marcusmae@teslatron:~/cholessky$ ./cholessky 4096

Computing on CPU ... OK

Computing on GPU ... OK

Done! max diff = 0.000000

incorrect result (GPU version and CPU version are not matching)

marcusmae@teslatron:~/cholessky$ ./cholessky 4096

Computing on CPU ... OK

Computing on GPU ... OK

Done! max diff = 4032.119629

References

http://icl.cs.utk.edu/magma/