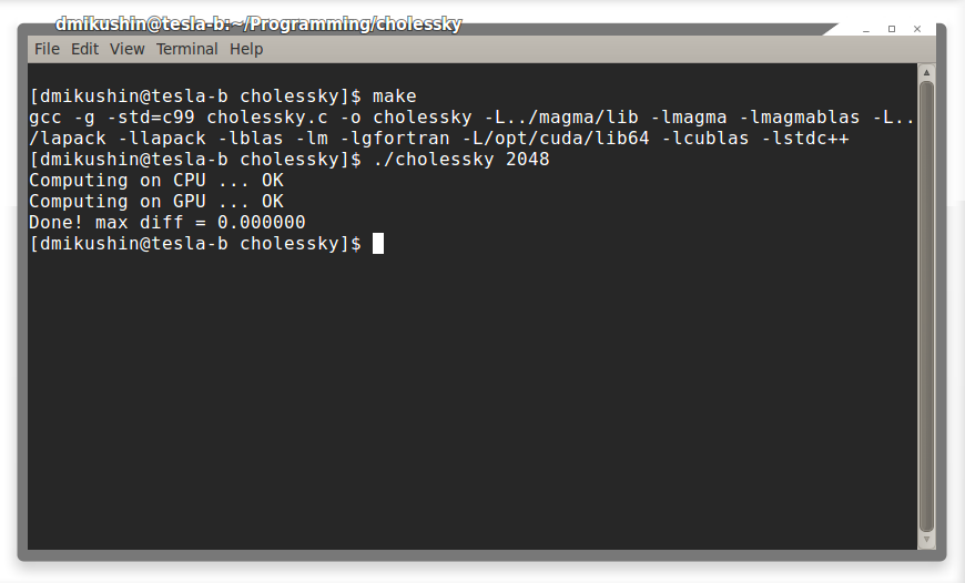
Пример разложения Холецкого на GPU



Элементы программы, использующей GPU

- Подготовка входных данных
- Загрузка входных массивов на GPU
- Запуск CUDA-ядер
- Выгрузка результатов из GPU

Пакеты со встроенной поддержкой GPU

- Запуск CUDA-ядер происходит *внутри* библиотечных функций, поэтому явного использования CUDA пользователем не требуется

- Управление загрузкой/выгрузкой данных *иногда* тоже происходит без помощи пользователя

Примеры библиотек с поддержкой GPU

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

...

Пример cholessky

- Генерация исходной матрицы
- Превращение матрицы в симметр. с диаг. преобл.
- Запуск разложения Холецкого на CPU (lapack)
- Запуск разложения Холецкого на GPU (magma)
- Сравнение результатов

Целевая система

Tesla A/B, Teslatron (тестовые узлы для практикума)

Core 2 Duo / Quad, 4 GB DDR3

Fedora 13 / Debian x86_64

cuda toolkit 3.2 (компилятор для gpu, runtime)

gcc 4.4.4 / 4.3.2 (хост-компилятор)

Сборка библиотек

Исходный код библиотек хранится на сервере и уже настроен для компиляции (код, загруженный из других мест возможно придётся настраивать)

LAPACK

Загрузка исходного кода из репозитория в локальную папку

marcusmae@teslatron:~$ svn co

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

Компиляция

marcusmae@teslatron:~$ cd lapack/

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

Сборка библиотек

Исходный код библиотек хранится на сервере и уже настроен для компиляции (код, загруженный из других мест возможно придётся настраивать)

MAGMA

Загрузка исходного кода из репозитория в локальную папку

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

Компиляция (**-j2** – в два параллельных процесса)

marcusmae@teslatron:~$ cd magma/

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

Сборка cholessky

marcusmae@teslatron:~$ cd cholessky/

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

Как работает сборка

GNU Make - http://www.gnu.org/software/make/

makefile состоит из правил сборки вида

<цель>: <цели, от которых зависит данная цель>

<табуляция><команды shell>

…

<табуляция><команды shell>

all – это правило, с которого начинается любая сборка

gcc -g -std=c99 cholessky.c -o cholessky -L../magma/lib -lmagma

-lmagmablas -L../lapack -llapack -lblas -lm -lgfortran -L/opt/cuda/lib64

-L/usr/local/cuda/lib64 -lcublas -lstdc++

Как работает сборка

gcc -g -std=c99 cholessky.c -o cholessky -L../magma/lib -lmagma -lmagmablas -L../lapack -llapack -lblas -lm -lgfortran -L/opt/cuda/lib64 -L/usr/local/cuda/lib64 -lcublas -lstdc++

gcc – обращение к компилятору GNU C

-g – включать отладочную информацию

cholessky.c – включаемые файлы исходного кода

-o <output> – выходной исполняемый файл

-L<path> – путь для поиска линкуемых библиотек

-l<name> – линкуемые библиотеки с заменой lib в имени на -l, например для линковки liblapack.so нужно указать -llapack или полный путь

Результаты

правильный результат (девайс-версия совпадает с контрольной)

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

Computing on CPU ... OK

Computing on GPU ... OK

Done! max diff = 0.000000

неправильный результат (CPU-версия и GPU-версия не совпадают)

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

Computing on CPU ... OK

Computing on GPU ... OK

Done! max diff = 4032.119629