Практикум CUDA
Отчет No2

Плужников Иван, 423
2021 Г, МГУ, ВМК, СКИ

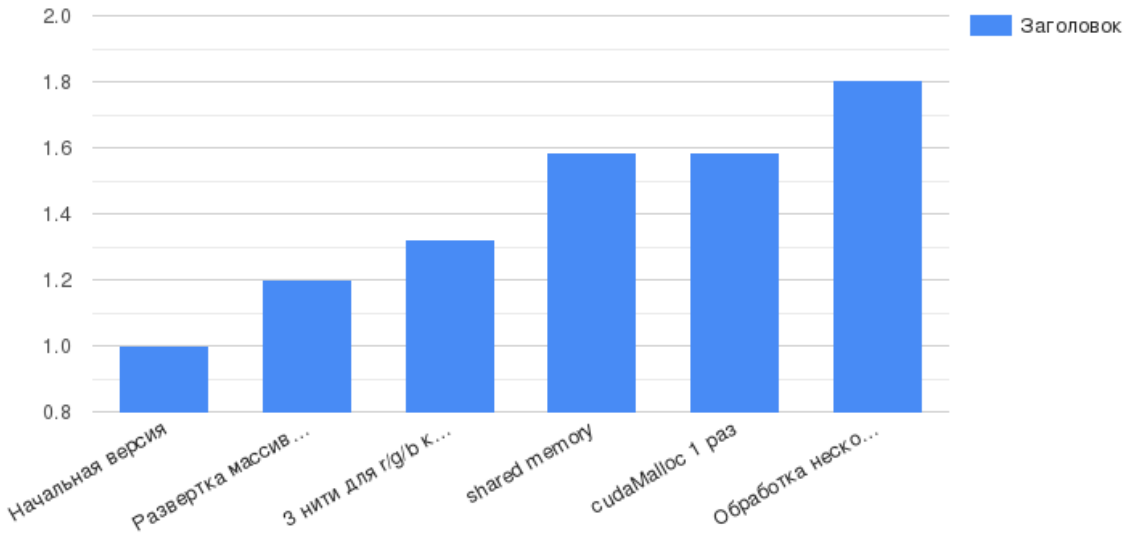
Задание:
Оптимизации для обработки больших и малых изображений:
- Развертка массива, где хранится изображение из массива структур в структуру массивов для улучшения шаблона доступа к глобальной памяти (Pixel * -> 3 массива unsigned char* для хранения 3 компонент изображения);
- Последовательный доступ к памяти от нитей варпа к массиву с изображением;
- Использование разделяемой (shared) памяти для применения фильтра (по аналогии со stencil) ;
- Использование 3х нитей для обработки r/g/b компонент;
- Различные походы к передаче фильтра в матрицу (full unroll, константная память);
- Развертка циклов, применяющих фильтров внутри каждой нити;
- Подбор оптимальных значений размера CUDA блока;
- Минимизация числа простаивающих нитей;
Дополнительные оптимизации для обработки набора из маленьких изображений:
- Выделение памяти (cudaMalloc) под обрабатываемые изображения 1 раз (а не каждый раз для каждого изображения заново);
- Обработка нескольких изображений за раз одним ядром или обработка нескольких изображений в конкурентном режиме при помощи CUDA-потоков;
- Одновременные копирования DtoH, HtoD и запуск ядер;
- Параллельная работа с файлами обработка изображений на GPU для групп из N - изображений: загружаем группу из N изображений с диска, пока их обрабатываем - грузим следующую. Сохранение на диск можно отключить (ifdef __NEED_TO_SAVE__).

Результат:

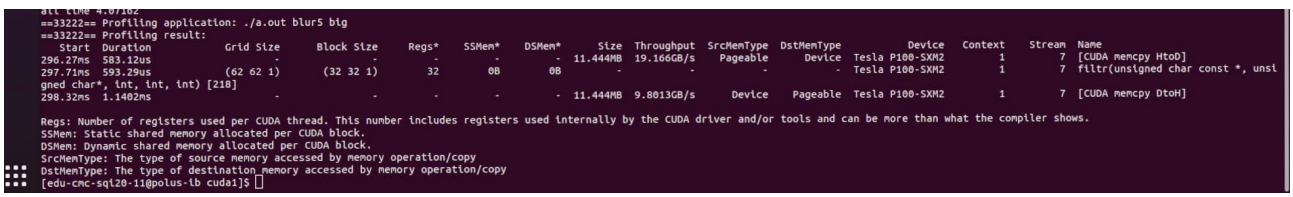
Оптимизация	3 маленьких изображения (вычисление, ms)	Большое изображение, (вычисление, ms)	Ускорение
Начальная версия	0.111072	0.414688	---
Развертка массива, где хранится изображение	0.09346	0.351216	~1.2
Использование 3х нитей для обработки r/g/b компонент	0.08496	0.31928727	~1.1
Использование разделяемой (shared) памяти для применения фильтра	0.07093	0.27436	~1.2
Выделение памяти (cudaMalloc) под обрабатываемые	0.0708134	---	~1

изображения 1 раз			
Обработка нескольких изображений за раз	0.062376	---	~1.14

Оптимизации 2, 6, 7 были реализованы в 1 версии программы, оптимальный размер блока и к-во нитей так же.
Пояснение оптимизации:
1) Улучшение кэширования.
2) Каждая нить выполняет меньше работы.
3) Копирование из глобальной памяти в глобальную работает значительно медленнее, чем связка глобальная память – разделяемая память – глобальная память.
4) Уменьшение затрат времени на выделение памяти.
5) Уменьшение накладных расходов на передачу данных.



Старая версия:
Большая картинка:



Маленькие картинки:

```
==33676== Profiling application: ./a.out blur5 small
==33676== Profiling result:
   Start  Duration           Grid Size      Block Size     Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context    Stream  Name
262.83ms  10.593us                    -               -         -         -         -  263.67KB  23.738GB/s    Pageable      Device  Tesla P100-SXM2         1         7  [CUDA memcpy HtoD]
263.35ms  18.272us              (9 9 1)       (32 32 1)        32        0B        0B         -           -           -           -  Tesla P100-SXM2         1         7  filtr(unsigned char const *, unsi
gned char*, int, int, int) [218]
263.38ms  8.5130us                    -               -         -         -         -  263.67KB  29.538GB/s      Device    Pageable  Tesla P100-SXM2         1         7  [CUDA memcpy DtoH]
466.84ms  9.3760us                    -               -         -         -         -  263.67KB  26.819GB/s    Pageable      Device  Tesla P100-SXM2         1         7  [CUDA memcpy HtoD]
467.33ms  17.665us              (9 9 1)       (32 32 1)        32        0B        0B         -           -           -           -  Tesla P100-SXM2         1         7  filtr(unsigned char const *, unsi
gned char*, int, int, int) [246]
467.37ms  8.0640us                    -               -         -         -         -  263.67KB  31.183GB/s      Device    Pageable  Tesla P100-SXM2         1         7  [CUDA memcpy DtoH]
588.97ms  9.7600us                    -               -         -         -         -  263.67KB  25.764GB/s    Pageable      Device  Tesla P100-SXM2         1         7  [CUDA memcpy HtoD]
589.46ms  17.472us              (9 9 1)       (32 32 1)        32        0B        0B         -           -           -           -  Tesla P100-SXM2         1         7  filtr(unsigned char const *, unsi
gned char*, int, int, int) [274]
589.50ms  8.0010us                    -               -         -         -         -  263.67KB  31.428GB/s      Device    Pageable  Tesla P100-SXM2         1         7  [CUDA memcpy DtoH]

Regs: Number of registers used per CUDA thread. This number includes registers used internally by the CUDA driver and/or tools and can be more than what the compiler shows.
SSMem: Static shared memory allocated per CUDA block.
DSMem: Dynamic shared memory allocated per CUDA block.
SrcMemType: The type of source memory accessed by memory operation/copy
DstMemType: The type of destination memory accessed by memory operation/copy
```

Новая версия:
Большая картинка:

```
==34034== Profiling application: ./a.out blur5 big
==34034== Profiling result:
   Start  Duration           Grid Size      Block Size     Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context    Stream  Name
373.99ms  193.76us                    -               -         -         -         -  3.8147MB  19.226GB/s    Pageable      Device  Tesla P100-SXM2         1         7  [CUDA memcpy HtoD]
374.23ms  180.48us                    -               -         -         -         -  3.8147MB  20.641GB/s    Pageable      Device  Tesla P100-SXM2         1         7  [CUDA memcpy HtoD]
374.46ms  195.56us                    -               -         -         -         -  3.8147MB  19.050GB/s    Pageable      Device  Tesla P100-SXM2         1         7  filtr(unsigned char const *, unsi
gned char const *, unsigned char const *, unsigned char*, unsigned char*, unsigned char*, int, int, int) [228]
375.19ms  817.58us           (186 62 1)       (32 32 1)        32  3.7969KB        0B         -           -           -           -  Tesla P100-SXM2         1         7  [CUDA memcpy DtoH]
376.03ms  728.37us                    -               -         -         -         -  3.8147MB  5.1146GB/s      Device    Pageable  Tesla P100-SXM2         1         7  [CUDA memcpy DtoH]
377.04ms  399.11us                    -               -         -         -         -  3.8147MB  9.3339GB/s      Device    Pageable  Tesla P100-SXM2         1         7  [CUDA memcpy DtoH]
377.73ms  404.55us                    -               -         -         -         -  3.8147MB  9.2084GB/s      Device    Pageable  Tesla P100-SXM2         1         7  [CUDA memcpy DtoH]

Regs: Number of registers used per CUDA thread. This number includes registers used internally by the CUDA driver and/or tools and can be more than what the compiler shows.
SSMem: Static shared memory allocated per CUDA block.
DSMem: Dynamic shared memory allocated per CUDA block.
SrcMemType: The type of source memory accessed by memory operation/copy
DstMemType: The type of destination memory accessed by memory operation/copy
```

Маленькие картинки:

```
==34147== Profiling result:
   Start  Duration           Grid Size      Block Size     Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context    Stream  Name
297.47ms  11.168us                    -               -         -         -         -  263.67KB  22.516GB/s    Pageable      Device  Tesla P100-SXM2         1        15  [CUDA memcpy HtoD]
297.49ms  10.977us                    -               -         -         -         -  263.67KB  22.908GB/s    Pageable      Device  Tesla P100-SXM2         1        15  [CUDA memcpy HtoD]
297.51ms  9.5370us                    -               -         -         -         -  263.67KB  26.366GB/s    Pageable      Device  Tesla P100-SXM2         1        15  [CUDA memcpy HtoD]
298.03ms  163.85us           (252 9 1)       (32 32 1)        32  3.7969KB        0B         -           -           -           -  Tesla P100-SXM2         1        15  filtr(unsigned char const *, unsi
gned char const *, unsigned char const *, unsigned char*, unsigned char*, unsigned char*, int, int, int) [229]
298.22ms  8.7370us                    -               -         -         -         -  263.67KB  28.781GB/s      Device    Pageable  Tesla P100-SXM2         1        15  [CUDA memcpy DtoH]
298.31ms  8.4800us                    -               -         -         -         -  263.67KB  29.653GB/s      Device    Pageable  Tesla P100-SXM2         1        15  [CUDA memcpy DtoH]
298.39ms  8.2890us                    -               -         -         -         -  263.67KB  30.336GB/s      Device    Pageable  Tesla P100-SXM2         1        15  [CUDA memcpy DtoH]

Regs: Number of registers used per CUDA thread. This number includes registers used internally by the CUDA driver and/or tools and can be more than what the compiler shows.
SSMem: Static shared memory allocated per CUDA block.
DSMem: Dynamic shared memory allocated per CUDA block.
SrcMemType: The type of source memory accessed by memory operation/copy
DstMemType: The type of destination memory accessed by memory operation/copy
[edu-cmc-sqi20-11@polus-ib cuda1]$ nvprof --print-gpu-trace ./a.out blur5 small
```

Вывод:
Удалось получить ускорение работы программы в 1.5 раза