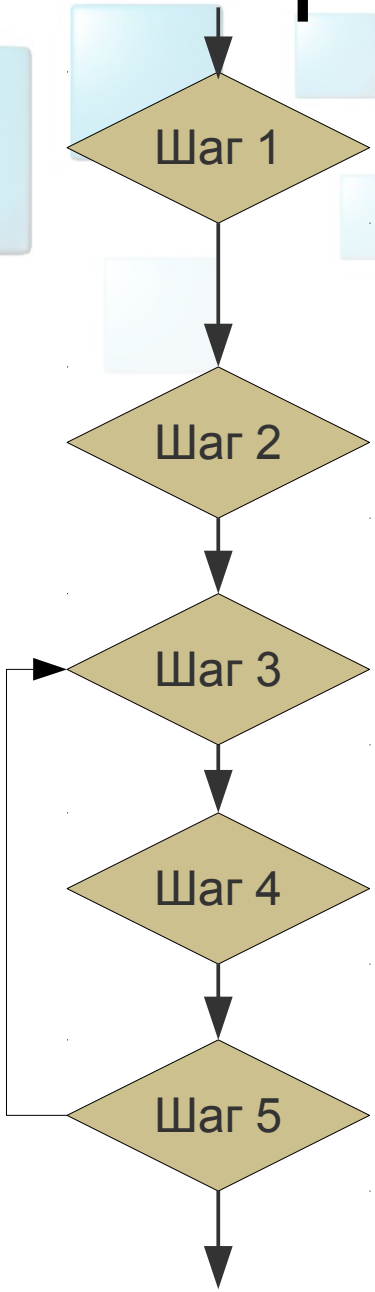
TTG LABORATORIES

TTG Apptimizer — почему оптимизировать GPGPU программы так просто

Докладчики:
Гризан С.А.

Ростов-На-Дону, 2013

Процесс GPGPU разработки



- Профилирование

  Поиск вычислительноемких участков CPU кода и оценка возможного ускорения

- Портирование

  Разработка CUDA/OpenCL версии

- Профилирование

  Анализ проблем производительности GPGPU кода

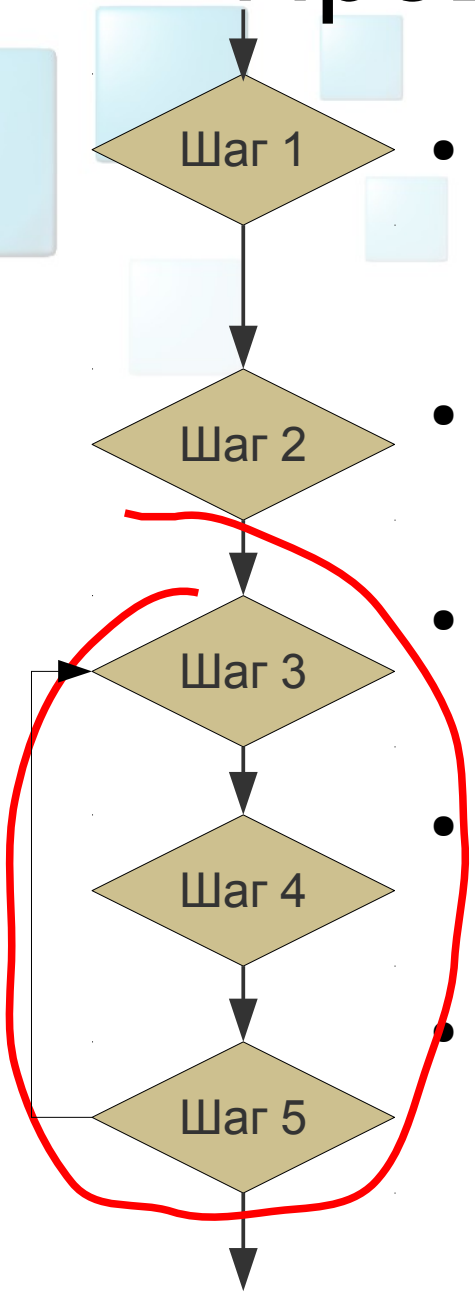
- Оптимизация

  Изменение алгоритмов, реализации GPGPU ядер

- Профилирование

  Определение достигнутого ускорения

Процесс GPGPU разработки



- Профилирование

  Поиск вычислительноемких участков CPU кода и оценка возможного ускорения

- Портирование

  Разработка CUDA/OpenCL версии

- Профилирование

  Анализ проблем производительности GPGPU кода

- Оптимизация

  Изменение алгоритмов, реализации GPGPU ядер

- Профилирование

  Определение достигнутого ускорения

Пути оптимизации

- Изменение алгоритма **x2 — x5 раза**
  - Перегруппировка циклов
  - Всевозможное кеширование

  ...
- Подстройка под «железо» **x1.5 — x2 раза**
  - Обеспечение выровненного доступа
  - Разбиение вычислений на блоки

  ...

Пути оптимизации

- Изменение алгоритма **x2 — x5 раза**
  - Перегруппировка циклов
  - Всевозможное кеширование
  - ...
- Подстройка под «железо» **x1.5 — x2 раза**
  - Обеспечение выровненного доступа
  - Разбиение вычислений на блоки
  - ...
- Подстройка под данные **x1.1 — x1.5 раза**
  - Использование разных вычислительных ядер
  - Изменение степени параллелизма
  - ...

Пути оптимизации

- Изменение алгоритма **x2 — x5 раза**
  - Перегруппировка циклов
  - Всевозможное кеширование
  - …
- Подстройка под «железо» **x1.5 — x2 раза**
  - Обеспечение выровненного доступа
  - Разбиение вычислений на блоки
  - …
- Подстройка под данные **x1.1 — x1.5 раза**
  - Использование разных вычислительных ядер
  - Изменение степени паралл
  - …

x2.5 + x1.75 = x4.375
x2.5 + x1.75 + x1.3 = x5.6875

Пути оптимизации

- Изменение алгоритма **x2 — x5 раза**
  - Перегруппировка циклов
  - Всевозможное кеширование
  - …
- Подстройка под «железо» **x1.5 — x2 раза**
  - Обеспечение выровненного доступа
  - Разбиение вычислений на блоки
  - …
- Подстройка под данные **x1.1 — x1.5**
  - Использование разных вы
  - Изменение степени парал
  - …

Случай с HDR:

1 алгоритм
+ 2 типа GPU
+ 3 диапазона данных

6 сборок программы

Autotuning

- Проблема: для хорошей оптимизации нужно слишком много «веток» / версий программы
- Идея: а давайте частично автоматизируем процесс подстройки программы под «железо» и данные
- Реализации идеи:
  - Компилятор «закладывает» в бинарник несколько версий вычислительных ядер
  - Система времени выполнения «подкручивает» одно универсальное ядро

TTG Apptimizer

- TTG Apptimizer
  - Минимальные изменения в исходном коде
  - Множество стратегий оптимизаций
  - Ускорение до 50% для одного GPU
  - Ускорение до 3 раз для GPU-кластера
- Стоимость
  - 500$ (Single workstation)
  - 5130$ (Cluster with 30 GPU)
  - 7930$ (Cluster with 60 GPU)

Что можно оптимизировать

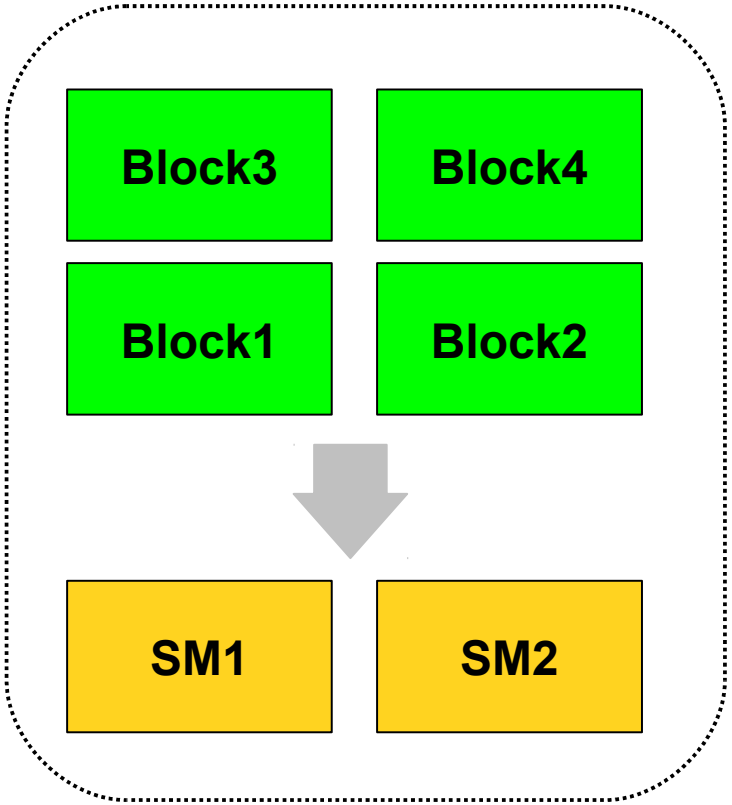
- Размеры всевозможных блоков
  - `cudaKernel<<<N / 256, 256>>>();`
- Тяжеловесность нитей вычислений
  - Одна нить на один пиксель, или сразу на четыре?
- Выбор вычислительного ядра / ветки
  - Использовать текстурную память, или положиться на кэши L1/L2?
- Выбор оптимального устройства
  - **CPU**, **iGPU** или **GPU** / **MIC**?
- Балансировка нагрузки между устройствами
  - **CPU** + 3x**GPU**, или просто 2x**GPU**?

Что можно оптимизировать

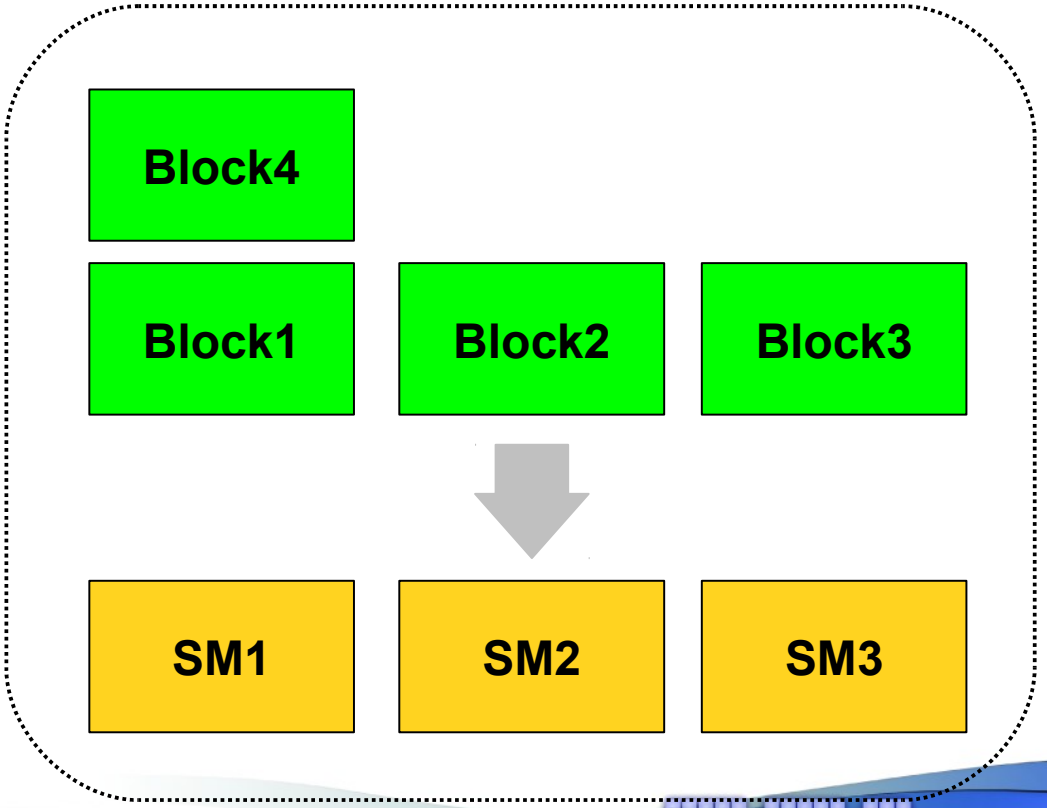
- Размеры всевозможных блоков
  - `cudaKernel<<<N / 256, 256>>>();`
- Тяжеловесность нитей вычислений
  - Одна нить на один пиксель, или сразу на четыре?
- Выбор вычислительного ядра / ветки
  - Использовать текстурную память, или положиться на кэши L1/L2?
- Выбор оптимального устройства
  - **CPU**, **iGPU** или **GPU** / **MIC**?
- Балансировка нагрузки между устройствами
  - **CPU** + 3x**GPU**, или просто 2x**GPU**?

Проблема 1. Как эффективно использовать мультипроцессоры ?

- Загрузка мультипроцессоров часто далека от оптимальной



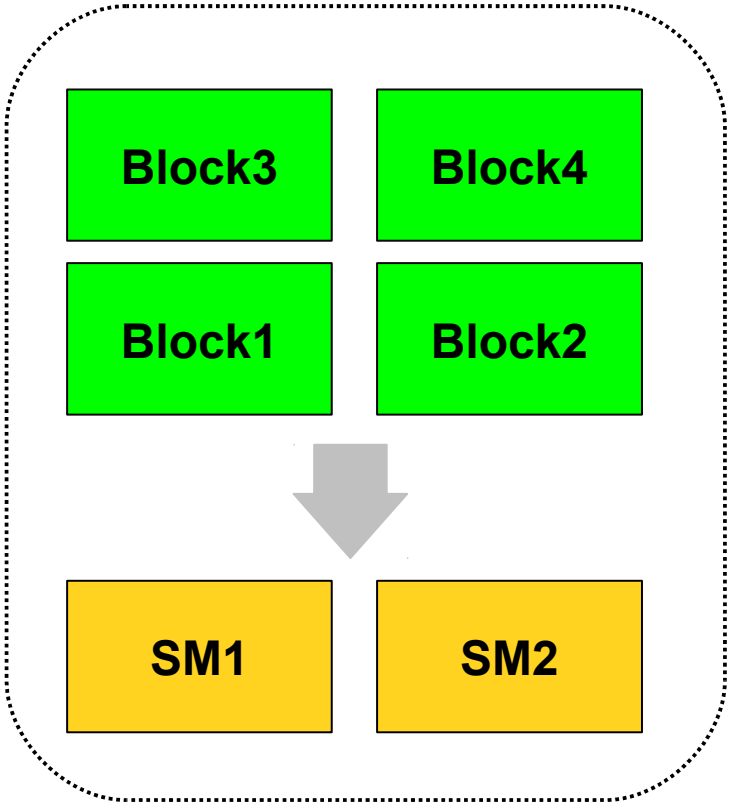
System 1



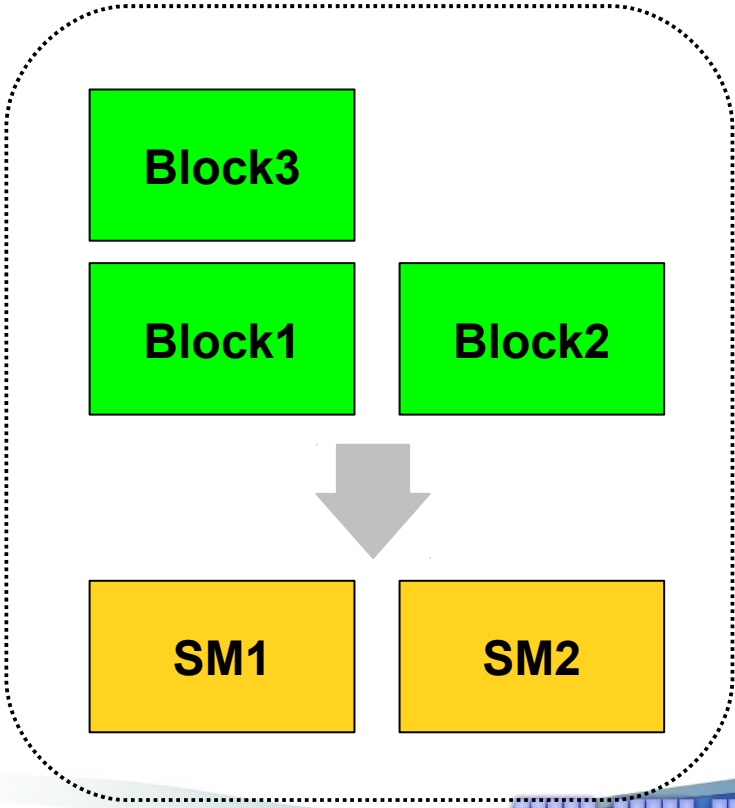
System 2

Проблема 1. Как эффективно использовать мультипроцессоры ?

- Загрузка мультипроцессоров часто далека от оптимальной



System 1



System 1

Проблема 1. Пример: решение уравнения теплопереноса

- Непрерывная задача
  - Найти функцию U(x, y, t) такую что

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \quad for\,(x, y) \in (0,1)\,x\,(0,1)$$

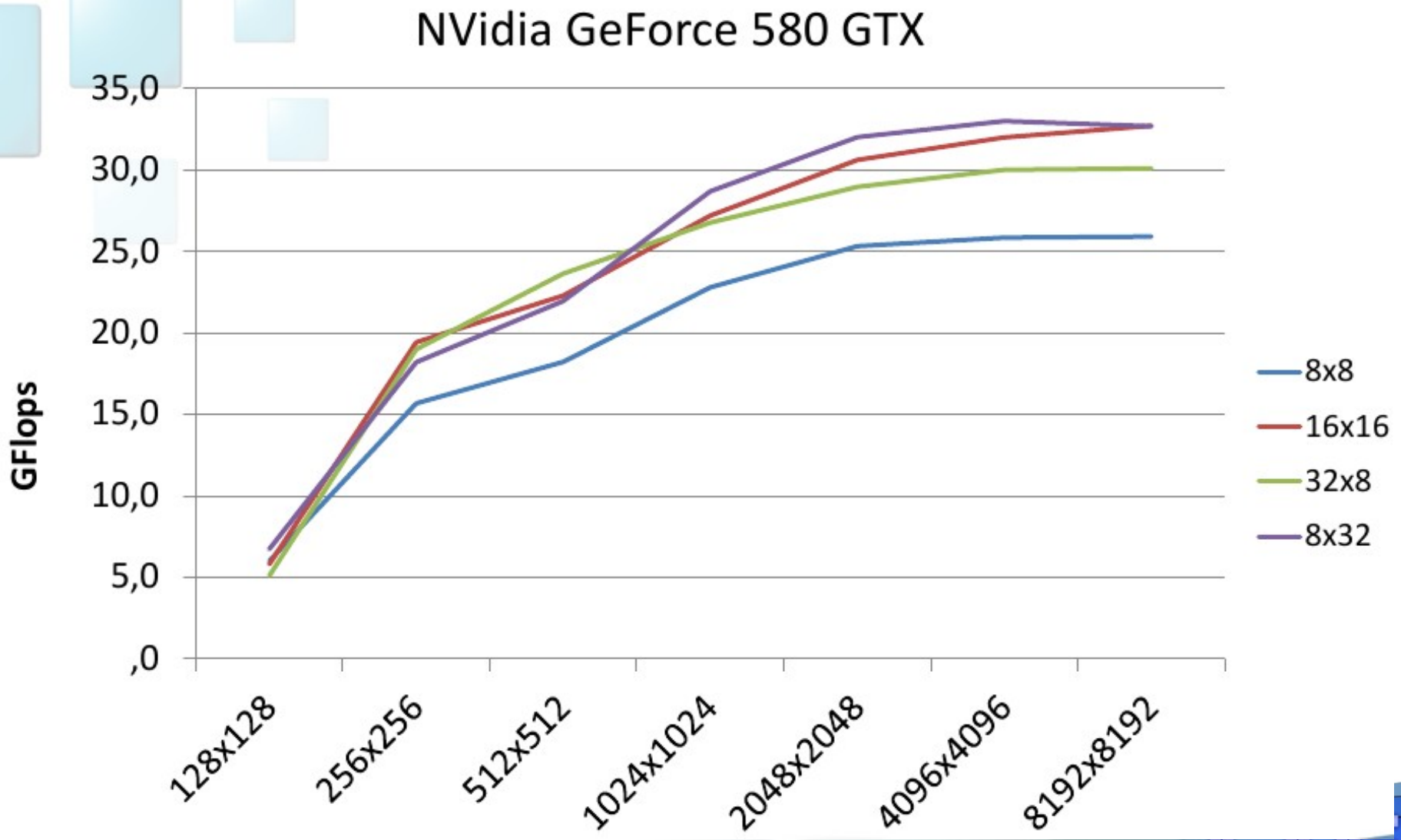
  - Граничные условия

$$\frac{\partial U}{\partial \vec{n}} = 0 \quad на\ границе$$

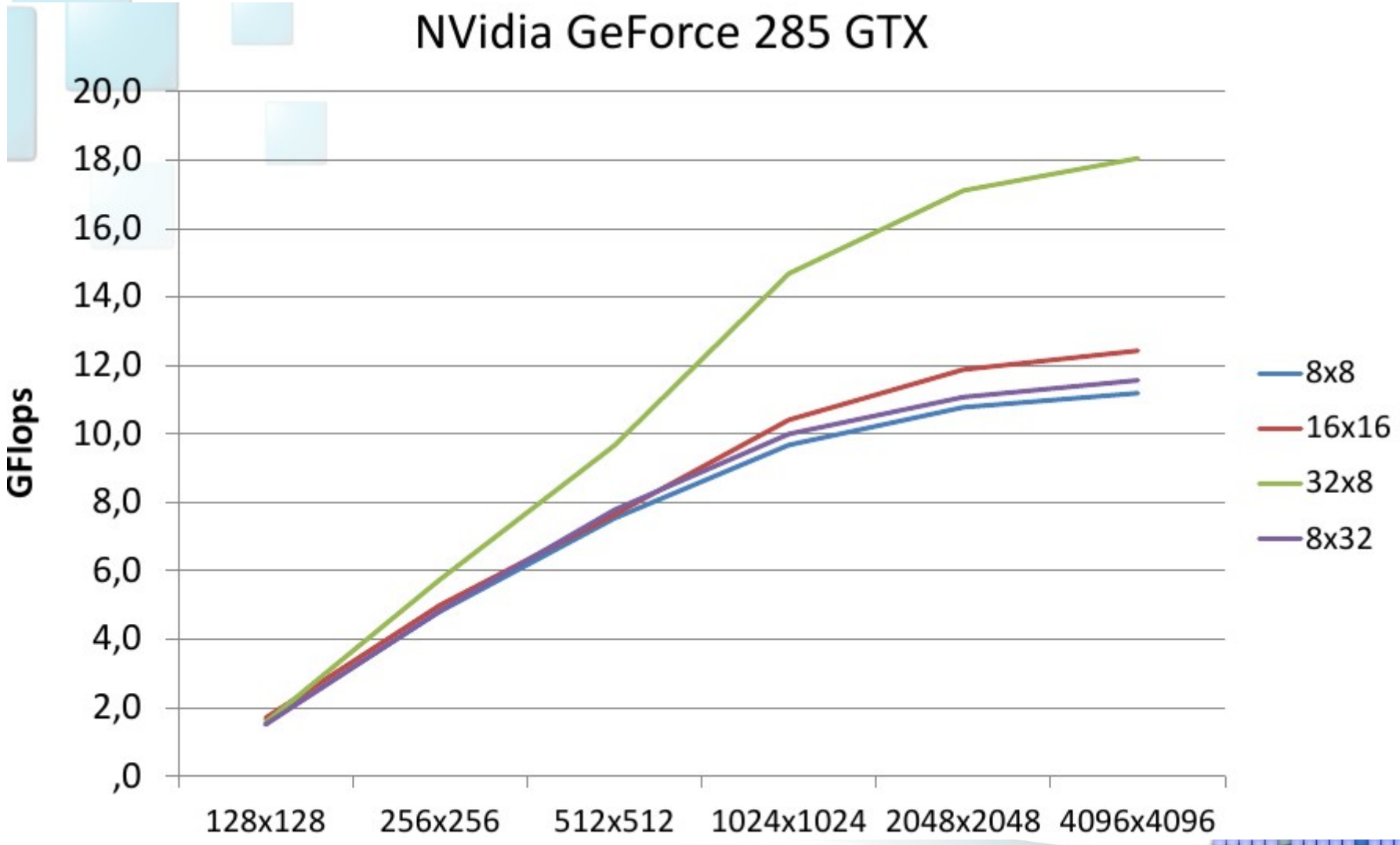
  - Начальные условия

$$U(x, y, t=0) = f(x)$$

Проблема 1. Пример: решение уравнения теплопереноса



Проблема 1. Пример: решение уравнения теплопереноса



Проблема 2. Определение тяжеловесных нитей

```
//Initialization
for (int i = 0;
     i < 1024;
     i++)
  //Calculations
```

Thread 1

OR

```
//Initialization
for (int i = 0;
     i < 512;
     i++)
  //Calculations
```

Thread 1

```
//Initialization
for (int i = 512;
     i < 1024;
     i++)
  //Calculations
```

Thread 2

Проблема 2. Пример: решение уравнения Пуассона

- Эллиптическое уравнение

$$\nabla \Phi(x, y, z) = f(x, y, z)$$

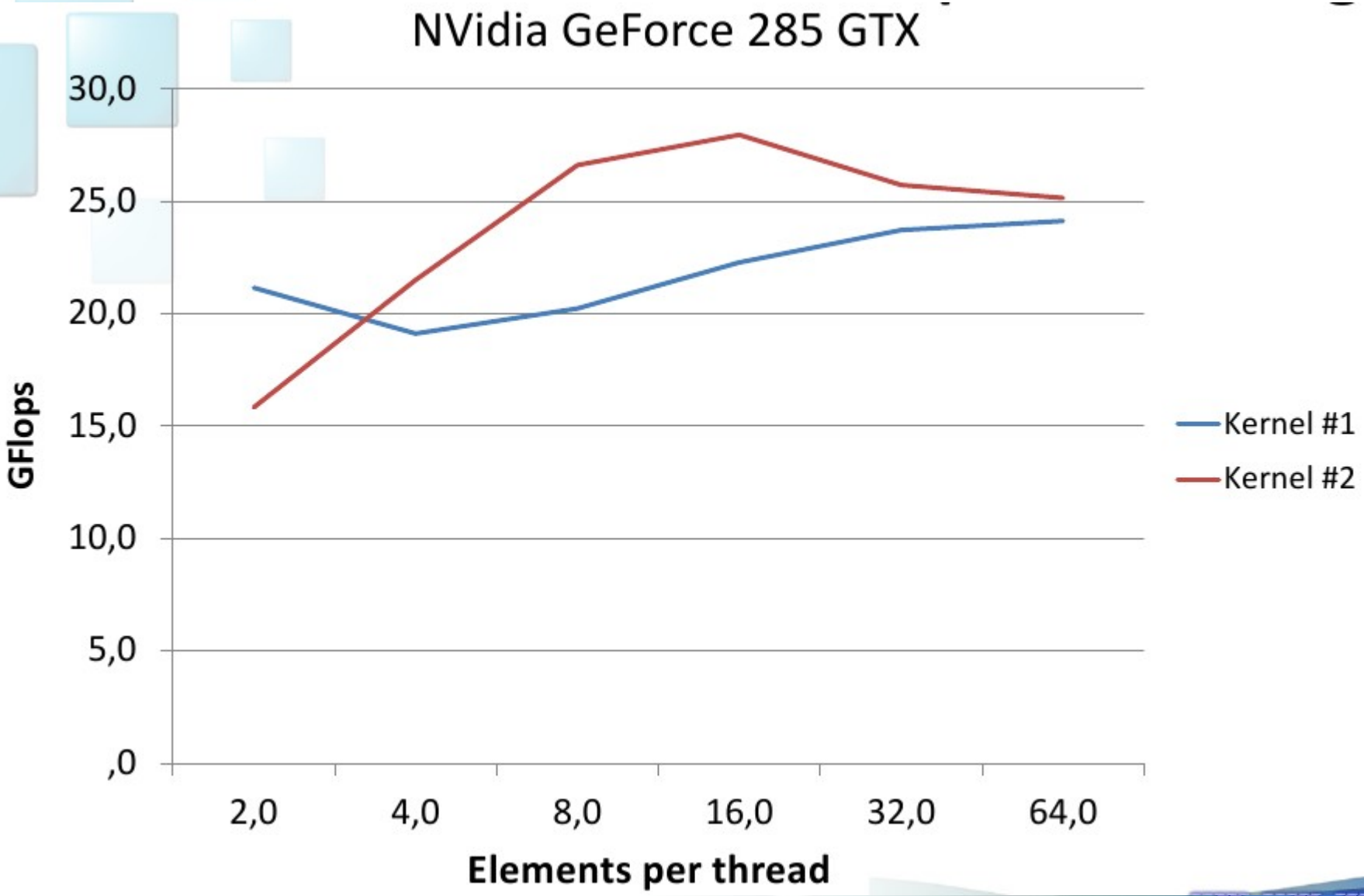
- Прямоугольная область *S*

$$S = [0.0, 1.0] \times [0.0, 1.0] \times [0.0, 1.0]$$

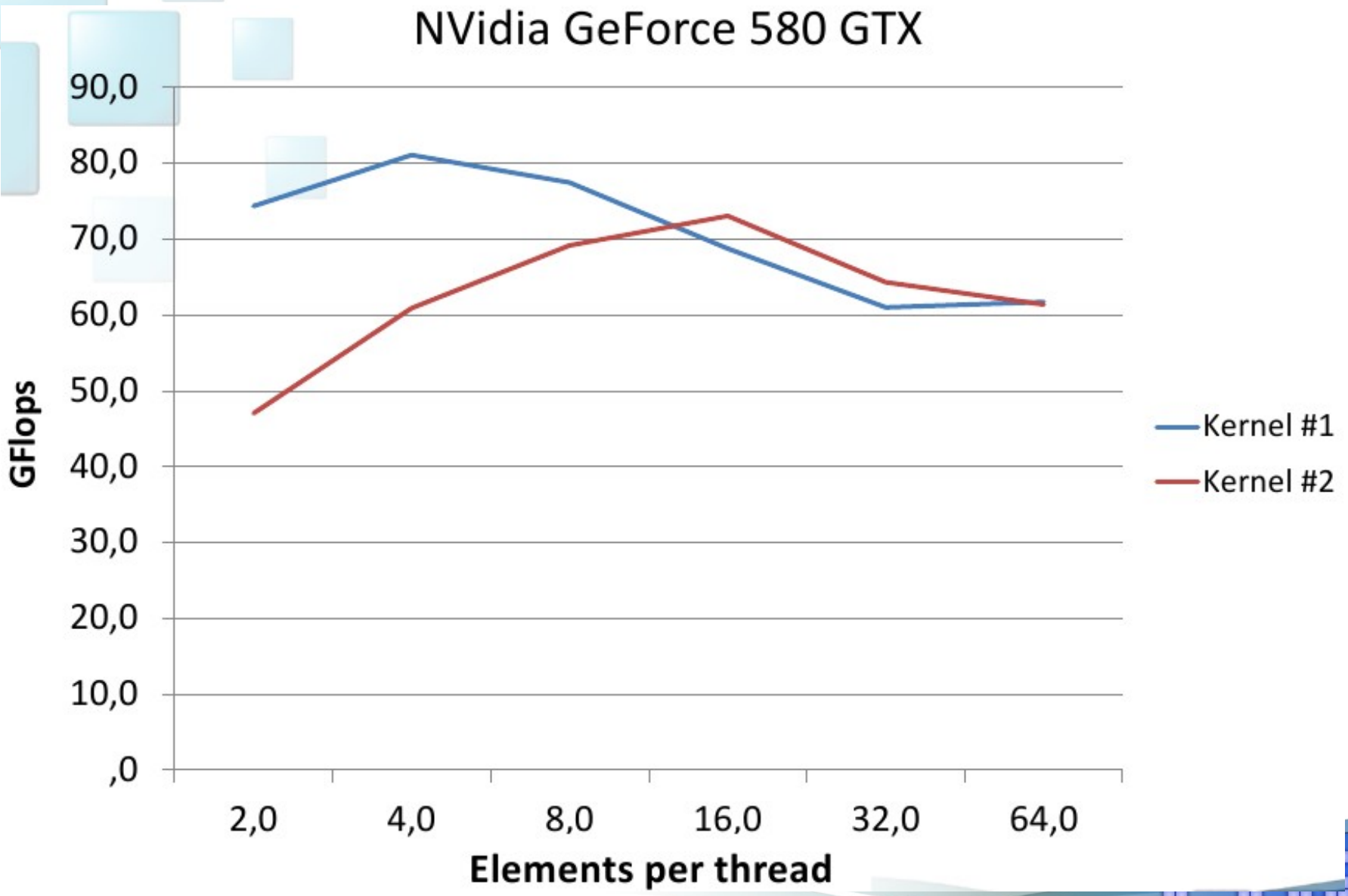
- Граничные условия Дирихле

$$\Phi(x, y, z)\big|_{(x,y,z)\in\bar{S}} = f(x, y, z)$$

Проблема 2. Пример: решение уравнения Пуассона



Проблема 2. Пример: решение уравнения Пуассона



Резюме

Благодаря правильному подбору параметров можно **«бесплатно»** получить ускорение порядка 20-50%.

Схема использования. Исходная программа

```
__global__ void myCudaKernel()
{ /*Computing something very important*/ }

//...

while (eps > 10e-9)
{
    myCudaKernel<<<N / 256, 256>>>();
    eps = GetResidual();
}
```

Схема использования. Делаем параметры динамическими.

```
__global__ void myCudaKernel()
{ /*Computing something very important*/ }

//...
EnumParameter<int> block(
                        «{128, 256, 512, 1024}»);

while (eps > 10e-9)
{
    myCudaKernel<<<N / block, block>>>();
    eps = GetResidual();
}
```

Схема использования. Включаем оптимизацию

```
__global__ void myCudaKernel()
{ /*Computing something very important*/ }

//...
EnumParameter<int> block(
                        «{128, 256, 512, 1024}»);
OptimizationSession os;
os.Attach(&block);

while (eps > 10e-9)
{
    os.StartIteration();
    myCudaKernel<<<N / block, block>>>();
    eps = GetResidual();
    os.FinishIteration();
}
```

Схема использования. Настраиваем оптимизацию

```
__global__ void myCudaKernel()
{ /*Computing something very important*/ }

//...
EnumParameter<int> block(
                    «{128, 256, 512, 1024}»);
OptimizationSession os;
os.Attach(&block);
os.AggressiveIteration() = 120;
os.StabilizationIteration() = 10;
os.Agregation() = 0.2;

while (eps > 10e-9)
{
    os.StartIteration();
    myCudaKernel<<<N / block, block>>>();
    eps = GetResidual();
    os.FinishIteration();
}
```

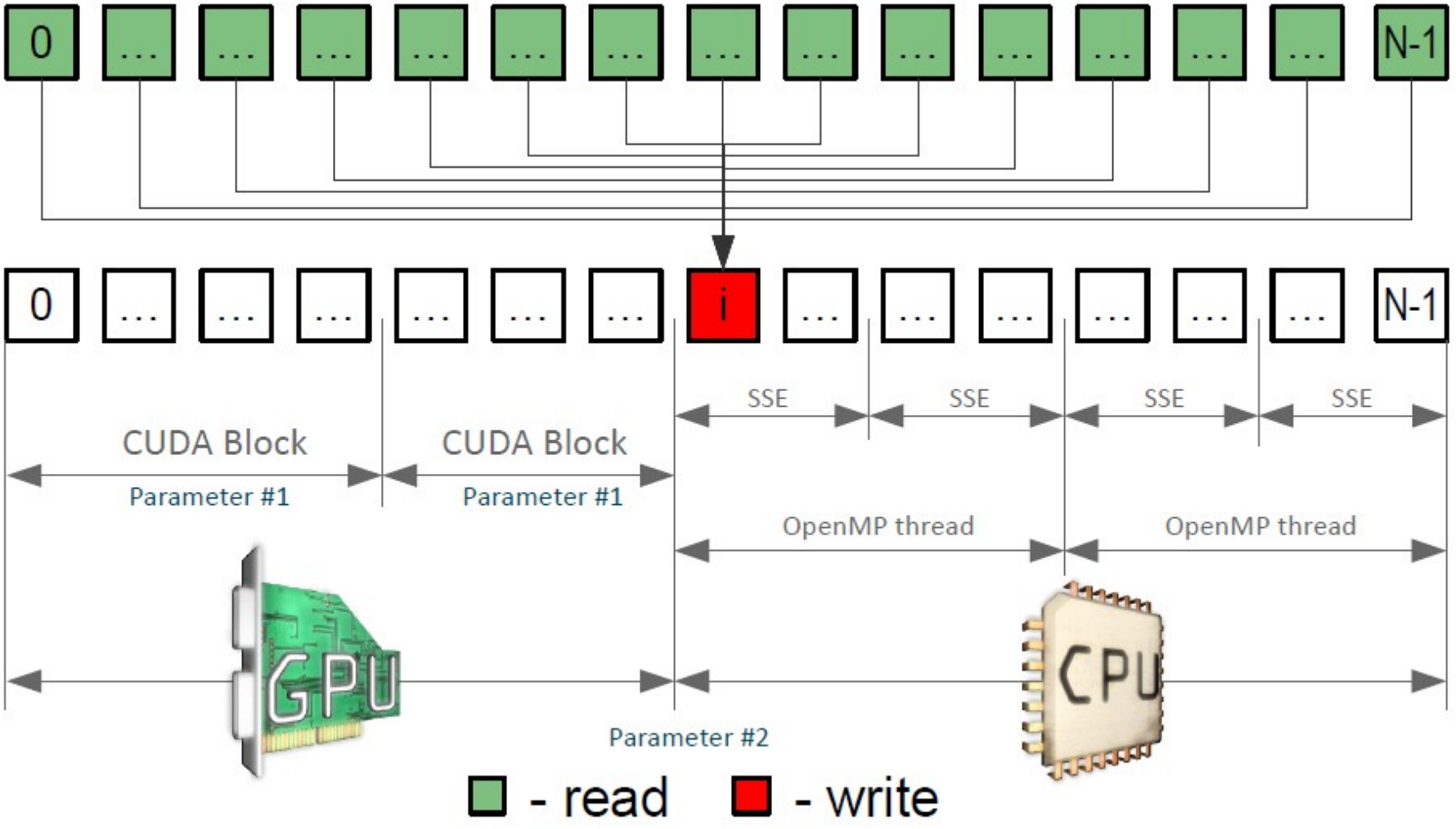
Схема использования. А можно не настраивать

```
__global__ void myCudaKernel()
{ /*Computing something very important*/ }

//...
EnumParameter<int> block(
                    «{128, 256, 512, 1024}»);
OptimizationSession os;
os.Attach(&block);
os.AggressiveIteration() = 120;
os.StabilizationIteration() = 10;
os.Agregation() = 0.2;

while (eps > 10e-9)
{
    os.StartIteration();
    myCudaKernel<<<N / block, block>>>();
    eps = GetResidual();
    os.FinishIteration();
}
```

Пример. Решение задачи N тел



Пример. Решение задачи N тел

Particles	GF 8800	GF 9800	GF 285	GF 480	GF 580	GF 680
0.5K	10.8 + 48.1%	13.1 + 56.8%	11.9 + 30.2%	15.3 + 2.8%	19.0 + 3.5%	18.1 + 0.5%
1.0K	22.4 + 48.1%	27.6 + 52.0%	24.8 + 28.3%	30.8 + 2.9%	38.4 + 3.4%	36.8 + 0.7%
2.0K	48.2 + 43.8%	60.5 + 45.7%	55.2 + 27.2%	65.3 + 1.3%	78.2 + 3.3%	75.2 - 0.0%
8.0K	90.9 + 0.0%	112.9 - 1%	103.8 + 55.3%	126.7 + 29.9%	229.9 - 0.1%	256.8 + 4.5%
16.0K	101 - 0.1%	127.7 - 0.0%	157.5 + 29.4%	125.3 +15.2%	252.7 + 0.7%	364.3 + 11.7%
32.0K	103.9 - 0.3%	131.7 - 0.1%	198.5 + 11.1%	160.6 +19.6%	255.3 + 0.7%	391.1 + 5.3%
Average	+23.3%	+25.6%	+30.3%	+12.0%	+1.9%	+3.7%

Пример. Решение задачи N тел

Particles	GF 8800	GF 9800	GF 285	GF 480	GF 580	GF 680
0.5K	10.8 + 48.1%	13.1 + 56.8%	11.9 + 30.2%	15.3 + 2.8%	19.0 + 3.5%	18.1 + 0.5%
1.0K	22.4 + 48.1%	27.6 + 52.0%	24.8 + 28.3%	30.8 + 2.9%	38.4 + 3.4%	36.8 + 0.7%
2.0K	48.2 + 43.8%	60.5 + 45.7%	55.2 + 27.2%	65.3 + 1.3%	78.2 + 3.3%	75.2 - 0.0%
8.0K	90.9 + 0.0%	112.9 - 1%	103.8 + 55.3%	126.7 + 29.9%	229.9 - 0.1%	256.8 + 4.5%
16.0K	101 - 0.1%	127.7 - 0.0%	157.5 + 29.4%	125.3 +15.2%	252.7 + 0.7%	364.3 + 11.7%
32.0K	103.9 - 0.3%	131.7 - 0.1%	198.5 + 11.1%	160.6 +19.6%	255.3 + 0.7%	391.1 + 5.3%
Average	+23.3%	+25.6%	+30.3%	+12.0%	+1.9%	+3.7%

Пример. Решение задачи N тел

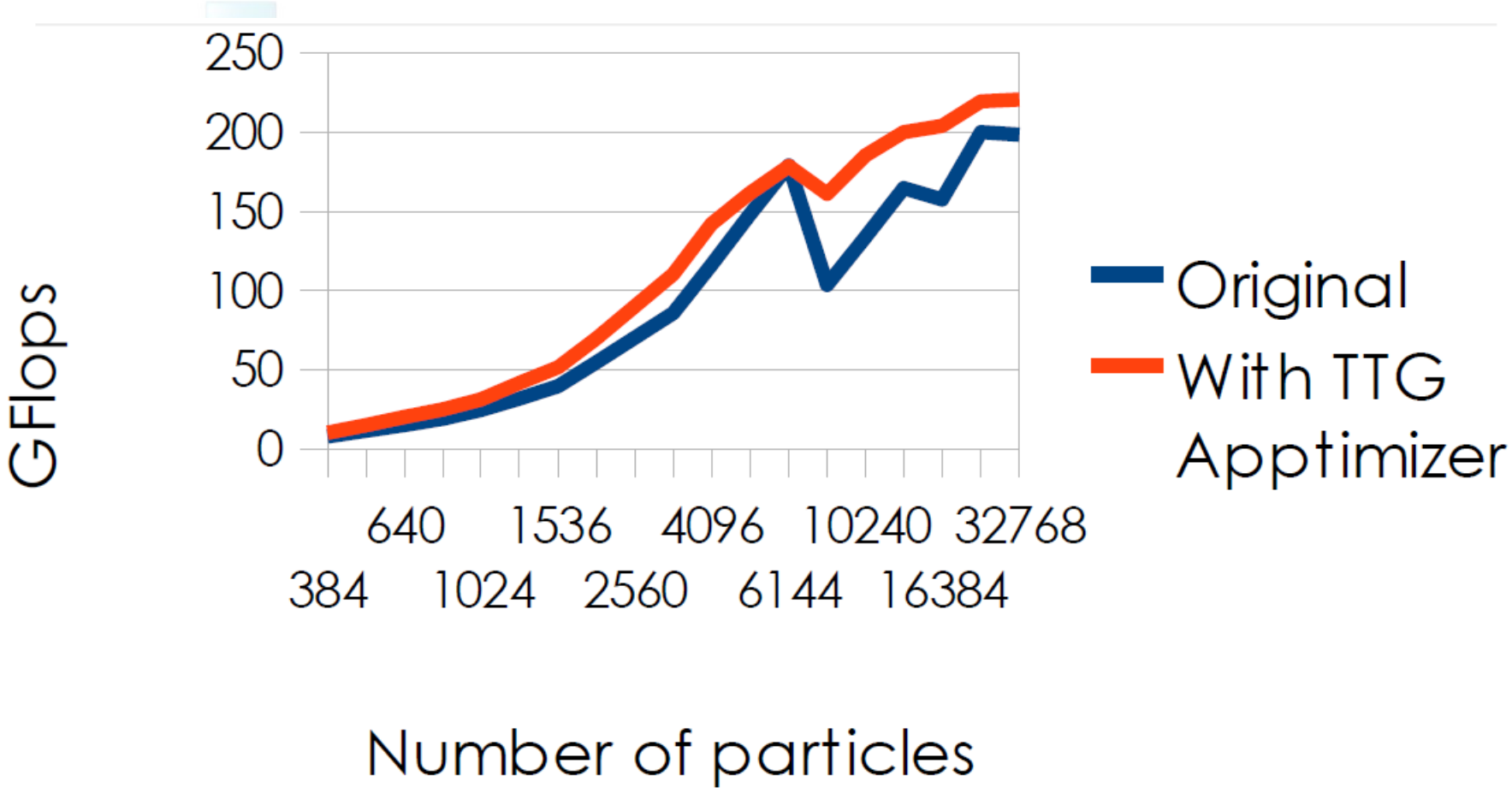


Схема использования. Исходная программа

```
void cpuKernel(void *, size_t, size_t);
void cudaKernel(void *, size_t, size_t);

//...

while (eps > 10e-9)
{
#ifdef USE_GPU
    cudaKernel(data, 0, 1024);
#else
    cpuKernel(data, 0, 1024);
#end
}
```

Схема использования. Включаем балансировку нагрузки

```
void cpuKernel(void *, size_t, size_t);
void cudaKernel(void *, size_t, size_t);

//...
HybridFor hFor;
hFor.Serial() += cpuKernel;
hFor.CUDA() += cudaKernel;
OptimizationSession os;
os.Attach(&block);

while (eps > 10e-9)
{
  os.StartIteration();
  hFor.Process(data, 0, 1024);
  os.FinishIteration();
}
```

Пример. Решение системы уравнений Эйлера

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho V_x)}{\partial x}+\frac{\partial(\rho V_y)}{\partial y}=0.0$$

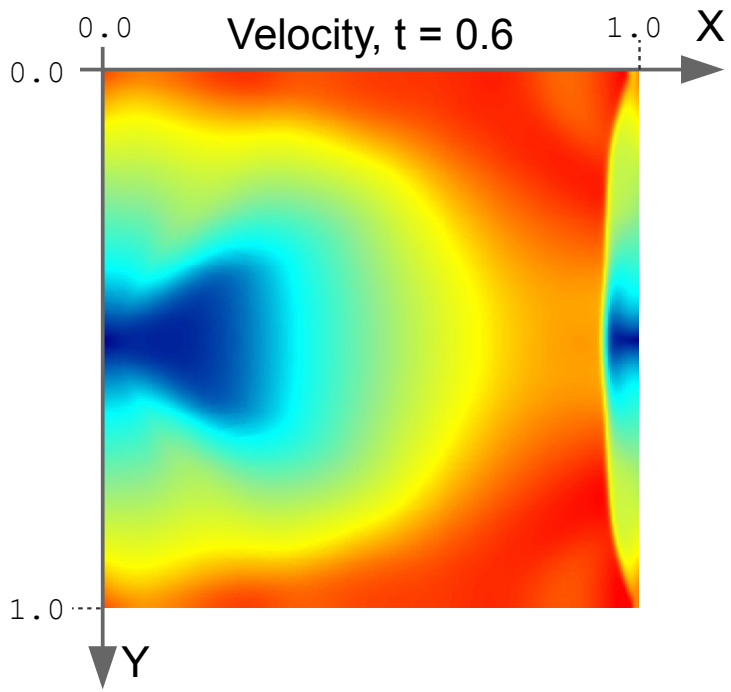
$$\frac{\partial \rho V_x}{\partial t}+\frac{\partial(\rho V_x^2+p)}{\partial x}+\frac{\partial(\rho V_x V_y)}{\partial y}=0.0$$

$$\frac{\partial \rho V_y}{\partial t}+\frac{\partial(\rho V_x V_y)}{\partial x}+\frac{\partial(\rho V_y^2+p)}{\partial y}=0.0$$

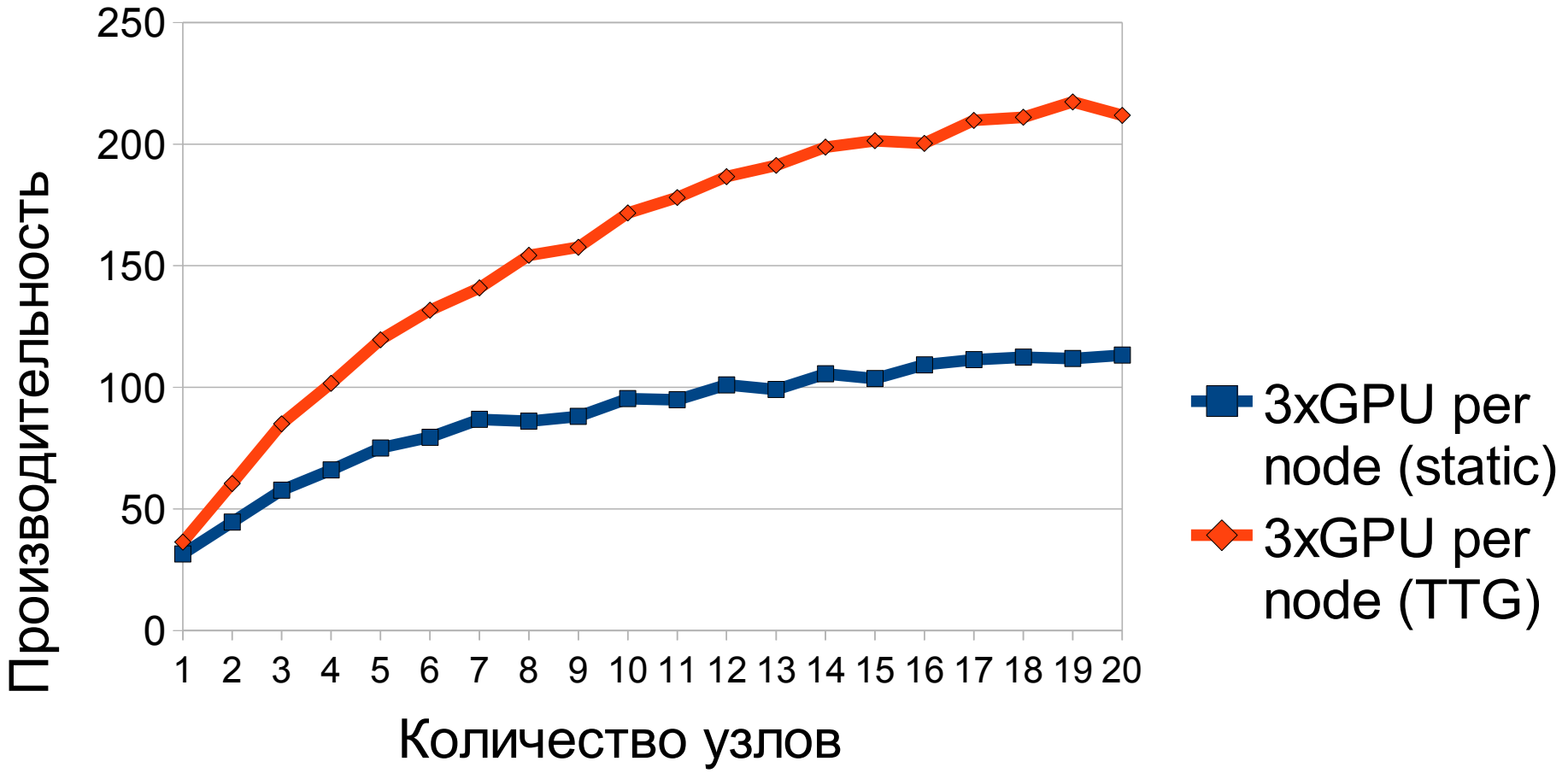
$$\frac{\partial \rho E}{\partial t}+\frac{\partial(\rho V_x H)}{\partial x}+\frac{\partial(\rho V_y H)}{\partial y}=0.0$$

$$E=\frac{1.0}{\gamma-1.0}\frac{p}{\rho}+\frac{V_x^2+V_y^2}{2}$$

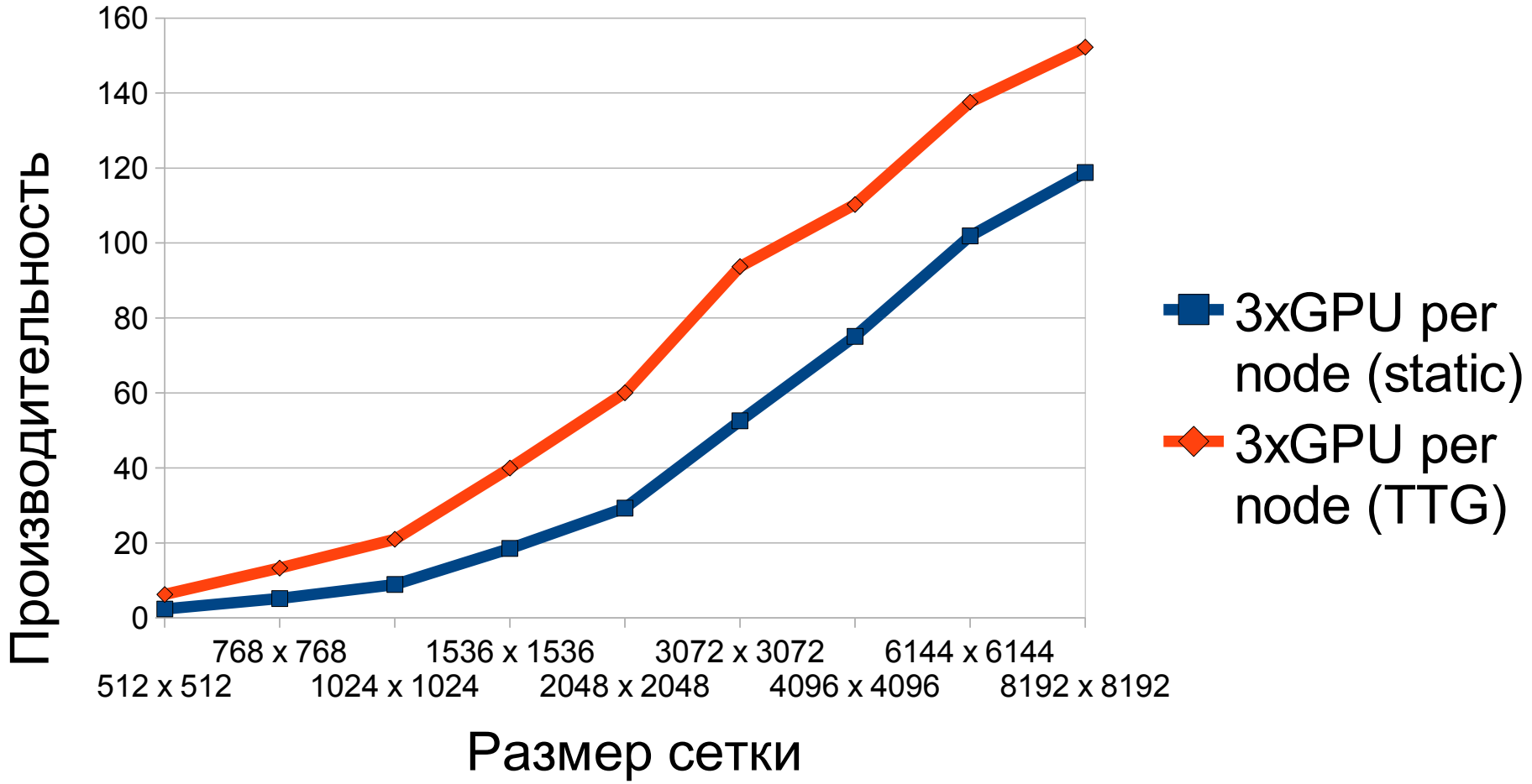
$$H=\frac{\gamma}{\gamma-1.0}\frac{p}{\rho}+\frac{V_x^2+V_y^2}{2}$$



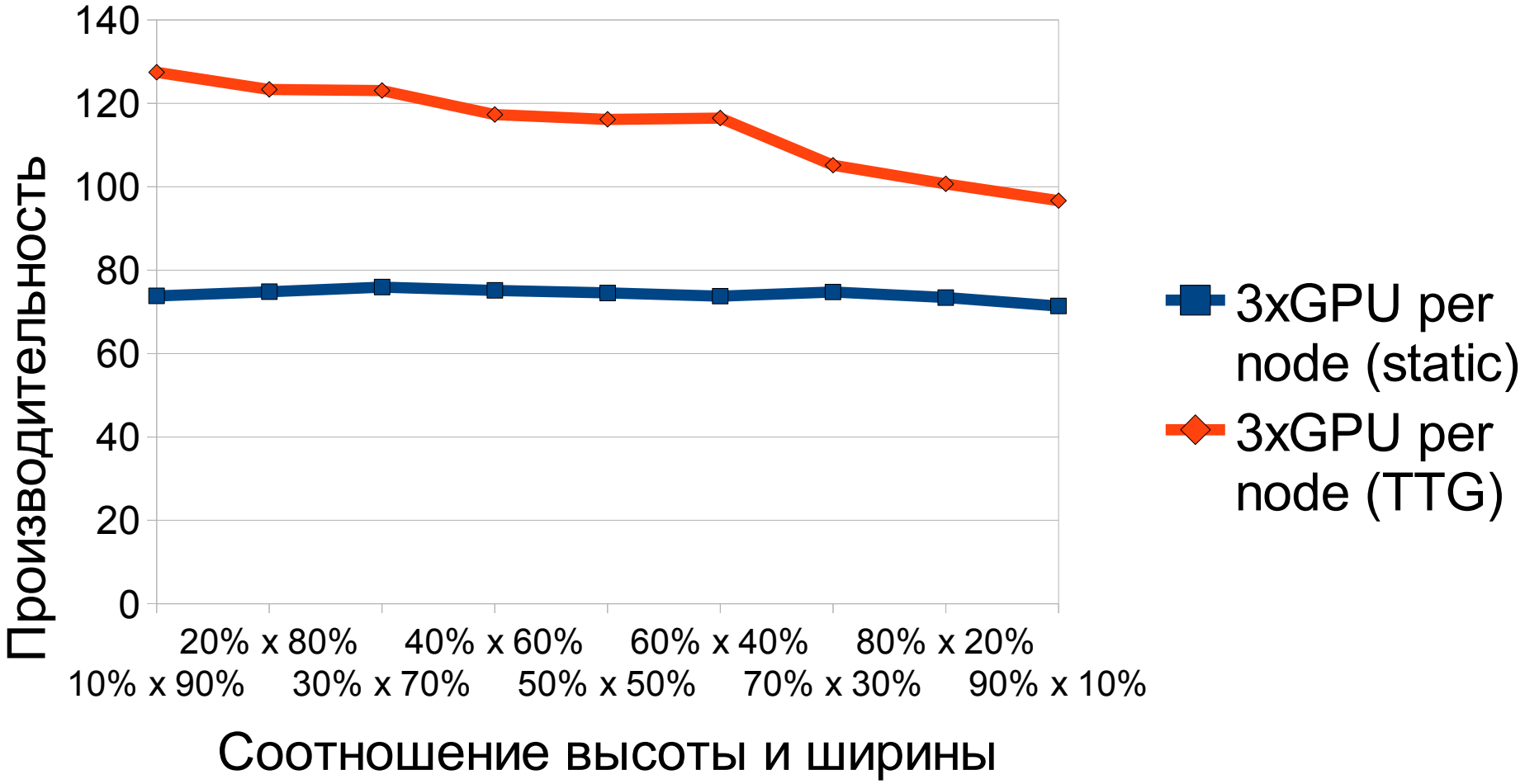
Пример. Решение системы уравнений Эйлера



Пример. Решение системы уравнений Эйлера



Пример. Решение системы уравнений Эйлера



TTG Apptimizer

TTG LABS

Вопросы?

(s_grizan@ttgLabs.com)