



Application Developer Days

CONFLUX: GPGPU ДЛЯ .NET

Евгений Бурмако, 2010

Видеокарты сегодня



Application Developer Days

- На борту – десятки/сотни ALU на частоте более 1 GHz.
- В пике – 1 TFLOPS (и >100 GFLOPS двойной точности)
- API – произвольный доступ к памяти, структуры данных, указатели, подпрограммы.
- Возраст API – почти четыре года, несколько поколений графических процессоров.

С точки зрения программиста



Application Developer Days

Современные модели программирования GPU (CUDA, AMD Stream, OpenCL, DirectCompute):

- Параллельный алгоритм задается парой: 1) ядро (итерация цикла), 2) границы итерации.
- Ядро компилируется драйвером.
- На основе границ итерации создается решетка вычислительных потоков.
- Входные данные копируются в видеопамять.
- Запускается выполнение задачи.
- Результат копируется в оперативную память.



Пример: SAXPY на CUDA

```
__global__ void Saxpy(float a, float* X, float* Y)
{
    int i = blockDim.x * blockIdx.x + threadIdx.x;
    Y[i] = a * X[i] + Y[i];
}
```

```
cudaMemcpy(X, hX, cudaMemcpyHostToDevice);
cudaMemcpy(Y, hY, cudaMemcpyHostToDevice);
Saxpy<<<256, (N + 255) / 256>>>(a, hX, hY);
cudaMemcpy(hY, Y, cudaMemcpyDeviceToHost);
```

Вопрос к знатокам



Application Developer Days

? Support for GPGPU in .Net 4.0? 📡



Lachlan01



Hi, with the advent of nVidia's CUDA etc, is there going to be support for programming against a GPU in .Net 4.0? There seems to be a large focus on parallelism as regards multicore CPUs, but haven't seen anything about GPUs.

[Sign In to Vote](#)

Cheers.

Отвечает Александр Друзь



Application Developer Days

GPGPU programming with C# and F#



dsyme 28 Jul 2010 4:54 PM



0

People often ask me about GPGPU (General Purpose GPU) programming with F#.

One project you might like to take a look at is [Brahma](#), which lets you compile C# expression trees to the GPU. When combined with the F# Power Pack support for converting F# quotations to LINQ expression trees, this should give a viable technique for using GPUs for computation.

Another route is to use [Microsoft Accelerator](#), an incubation project from Microsoft Research. New functionality includes...

По факту



Application Developer Days

Brahma:

- Структуры данных: data parallel array.
- Вычисления: выражения C#, LINQ-комбинаторы.

Accelerator v2:

- Структуры данных: data parallel array.
- Вычисления: арифметические операторы, набор предопределенных функций.

Это работает для многих алгоритмов. Но что, если есть ветвления или нерегулярный доступ к памяти?

А ВОТ ЧТО



Application Developer Days

```
saxpy = @"__global__ void Saxpy(float a, float* X, float* Y)
{
    int i = blockDim.x * blockIdx.x + threadIdx.x;
    Y[i] = a * X[i] + Y[i];
}";
```

```
nvcuda.cuModuleLoadDataEx(saxpy);
nvcuda.cuMemcpyHtoD(X, Y);
nvcuda.cuParamSeti(a, X, Y);
nvcuda.cuLaunchGrid(256, (N + 255) / 256);
nvcuda.cuMemcpyDtoH(Y);
```

Конфлакс



Application Developer Days

Ядра пишутся на C#: поддерживаются структуры данных, локальные переменные, ветвления, циклы.

```
float a;
```

```
float[] x;
```

```
[Result] float[] y;
```

```
var i = GlobalIdx.X;
```

```
y[i] = a * x[i] + y[i];
```

Конфлакс



Application Developer Days

Не требует явного общения с неуправляемым кодом, позволяет работать с родными типами данных .NET.

```
float[] x, y;  
var cfg = new CudaConfig();  
var kernel = cfg.Configure<Saxpy>();  
y = kernel.Execute(a, x, y);
```

Как это работает?



Application Developer Days

- Front end: декомпилятор C#.
- Преобразование AST: инлайн вызываемых методов, деструктуризация классов и массивов, отображение вычислительных операций.
- Back end: генераторы PTX и многоядерного IL.
- Привязка к драйверу nvcc, который умеет исполнять программы на ассемблере.

Успехи



Application Developer Days

<http://bitbucket.org/conflux/conflux>

- Альфа-версия.
- Умеет вычислять hello-world параллельных вычислений: умножение матриц.
- За вычетом [на текущий момент] высоких издержек на JIT-компиляцию идея оправдывает себя даже для наивного кодогенератора: $1x \text{ CPU} < 2x \text{ CPU} \ll \text{GPU}$.
- Тройная лицензия: AGPL, исключение для OSS-проектов, коммерческая.

Демонстрация



Application Developer Days

The screenshot displays the Visual Studio interface during a unit test session. On the left, the Solution Explorer shows a project named 'Conflux' with sub-projects 'Conflux.Playground', 'XenoGears', and 'XenoGears.Playground'. The main window, titled 'Unit Test Sessions - Session #1', shows a tree view of test results. All tests passed, including 'MatMulKernelTests' and its sub-tests 'BigTest_Cpu_1x', 'BigTest_Cpu_2x', and 'BigTest_Cuda'. The 'BigTest_Cuda' test is highlighted in blue. Below the tree view, a console window shows the execution details for the 'RunKernel' function, including grid configuration and memory usage in VRAM.

```
Launching function RunKernel (CUfunction 0x1e854ec0)...
Grid is configured as (416, 304, 1): blockdim is (16, 16, 1), gri
+0000 InOut System.Single[,] (524272 bytes in VRAM)
+0004 In 302 (0 bytes in VRAM)
+0008 In 434 (0 bytes in VRAM)
+0012 InOut System.Single[,] (708288 bytes in VRAM)
+0016 In 434 (0 bytes in VRAM)
+0020 In 408 (0 bytes in VRAM)
+0024 InOut System.Single[,] (492864 bytes in VRAM)
+0028 In 302 (0 bytes in VRAM)
+0032 In 408 (0 bytes in VRAM)
*****
0036 (1725424 bytes in VRAM)
Function execution succeeded in 00:00:00.003988 (ε = 0.5 μs).
```



Следующие шаги

- Оптимизации для графических процессоров (лесенка для оптимальной пропускной способности при транспозиции матриц).
- Полиэдральная модель оптимизации циклов (конфигурируется относительно иерархии и размеров кэшей, есть линейные эвристики, оптимизирующие локальность данных в вычислительной решетке).
- Исполнение на кластере (следующий шаг после полиэдральной модели: добавляется виртуальный уровень кэша – вычислительный узел).

Заключение



Application Developer Days

Conflux: GPGPU для .NET

<http://bitbucket.org/conflux/conflux>

eugene.burmakov@confluxrpc.net