# Workflow Builder для библиотеки Brahma.FSharp

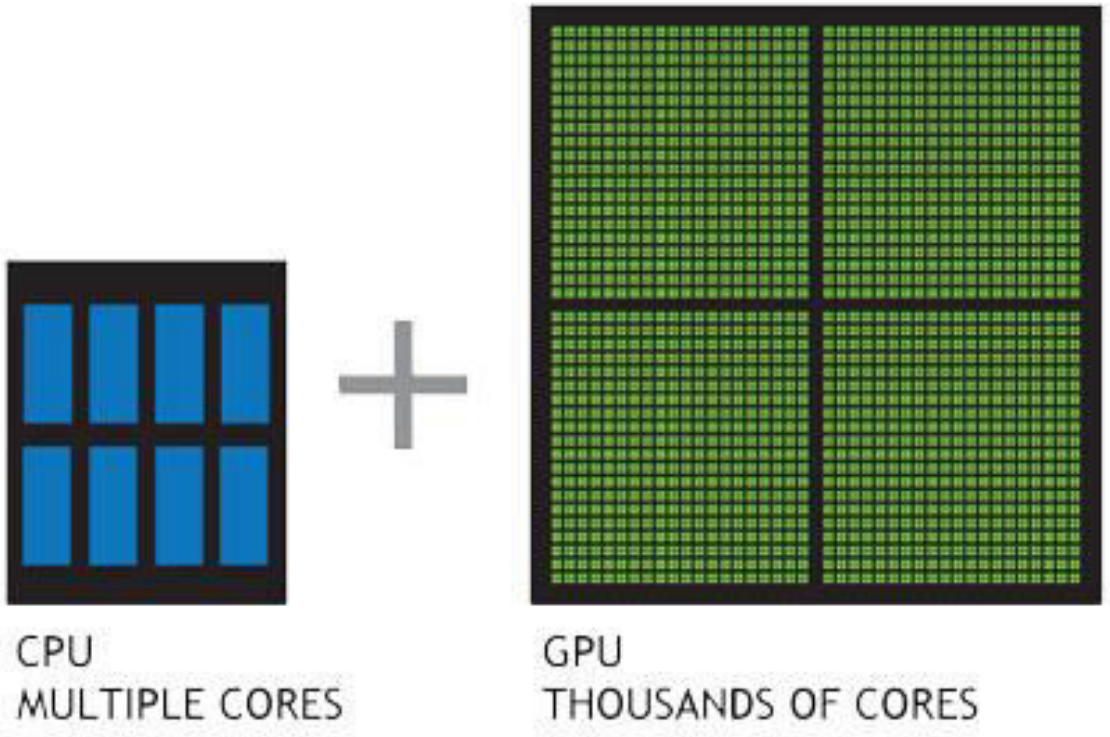
Автор: Васенина Анна Игоревна, 243 группа
Научный руководитель: Григорьев Семён Вячеславович

Санкт-Петербургский государственный университет
Кафедра системного программирования

17 октября 2017 года

# Введение

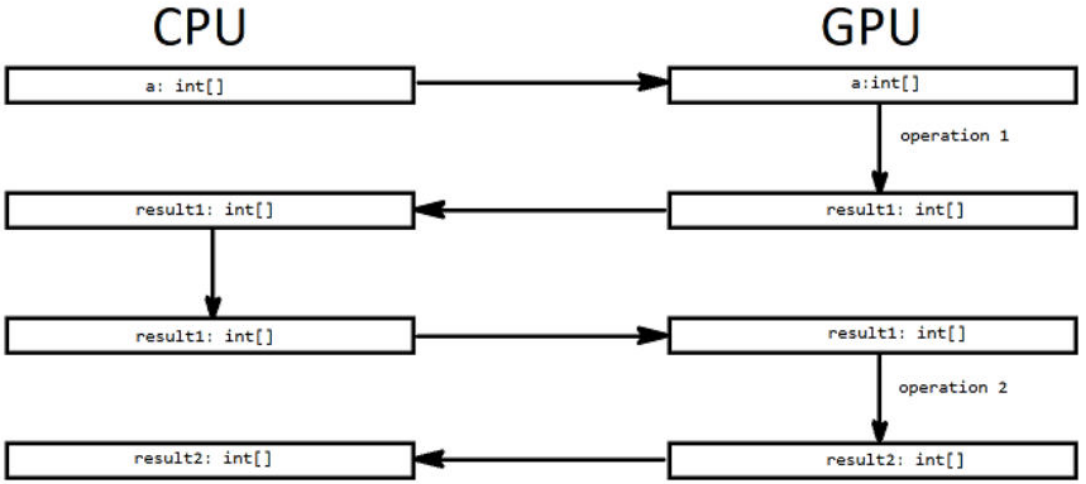
**GPGPU (General-purpose computing for graphics processing units, неспециализированные вычисления на графических процессорах) — вычисление неграфических данных на графических процессорах**




GPU и CPU (источник: http://www.nvidia.ru/object/gpu-computing-ru.html)

# Обзор

Brahma.Fsharp -библиотека на F# для интеграции GPGPU-вычислений по технологии OpenCL.




Используемые технологии:

- Computation expressions
- Монада reader

# Цель и задачи

Задачи:

- Спрятать передачу контекста
- Выполнять операции на GPU последовательно

Целью работы является создание модуля gpu{..} для библиотеки Brahma.FSharp, выполняющего поставленные задачи, и конструктора таких модулей для разных контекстов.

# Решение

type ReaderM<’context, ‘out> = ‘context -> ‘out

val run: ‘a -> ReaderM<’a,’b> -> ‘b

val constant: ‘c -> ‘a ->‘c

bind оборачивает в контекст, выполняет действие, разворачивает из контекста обратно

# Решение

```
type GPUBuilder (actcontext: context) =
    let provider = prov actcontext
    let mutable commandQueue = CQ actcontext
    let length = len actcontext
    let localWorkSize = WS actcontext

    member __.Bind(m, f)    = m >>= f
    member __.Yield (outArr: array<_>) =
        let _ = commandQueue.Add(outArr.ToHost provider).Finish()
        constant outArr
    member __.Zero = constant None
    member __.Return (outArr: array<_>) =
        let _ = commandQueue.Add(outArr.ToHost provider).Finish()
        commandQueue.Dispose()
        provider.Dispose()
        provider.CloseAllBuffers()
        constant outArr
    member __.Delay(f)      = f ()
```

# Результаты

1)Реализована возможность выполнения последовательных операций на GPU

2)Внутри модуля gpu{..} производится работа с заранее определенным контекстом вида (ComputeProvider * CommandQueue * int * int). Работать с функциями получающими такой контекст последним из аргументов можно без указания контекста

3)Проведено тестирование в системе NUnit