

Министерство науки и высшего образования Российской Федерации  
**Муромский институт**  
федерального государственного бюджетного образовательного учреждения  
высшего образования  
**«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(МИВлГУ)**

Факультет \_\_\_\_\_ ИТ \_\_\_\_\_

Кафедра \_\_\_\_\_ ИС \_\_\_\_\_

# ОТЧЕТ

По преддипломной практике

Руководитель

Щаников С.А.

(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Студент

ИС-117

(группа)

Минеев Р.Р.

(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Муром 2021

## Оглавление

Введение .....	3
1. Анализ технического задания .....	4
1.1. Зачем нейросетям краш-тест .....	4
1.2. Обзор методов организации параллельных вычислений .....	4
1.3. Выбор средства.....	6
1.4. Пара слов о ... ..	6
2. Проектирование .....	8
3. Тестирование .....	9
Заключение.....	10
Список использованной литературы.....	11

					МИВУ 09.03.02-00.000			
Из	Лист	№ докум.	Подп.	Дата				
Студент		Минеев Р.Р..			Отчет по преддипломной практике	Литера	Лист	Листов
Руков.		Щаников С.А..				У	2	-
Конс						МИ ВлГУ ИС-117		
Н.контр.								
Утв.								

Введение

Значимость работы, как она может улучшить мир

## 1. Анализ технического задания

### 1.1. Зачем нейросетям краш-тест

### 1.2. Обзор методов организации параллельных вычислений

Необходимость разделять вычислительные задачи и выполнять их одновременно (параллельно) возникла задолго до появления первых вычислительных машин.

Существует 3 метода распараллеливания расчетов:

- распараллеливание по задачам (такое распараллеливание актуально для сетевых серверов и других вычислительных систем, выполняющих одновременно несколько функций либо обслуживающих многих пользователей);

- распараллеливание по инструкциям (аппаратно реализовано в современных центральных процессорах общего назначения, поскольку оно эффективно при исполнении программ, интенсивно обменивающихся разнородной информацией с другими программами и с пользователем компьютера);

- распараллеливание по данным.

Потоковая обработка данных особенно эффективна для алгоритмов, обладающих следующими свойствами, характерными для задач физического и математического моделирования:

- большая плотность вычислений — велико число арифметических операций, приходящихся на одну операцию ввода-вывода (например, обращение к памяти). Во многих современных приложениях обработки сигналов она достигает 50:1, причем со сложностью алгоритмов увеличивается;

- локальность данных по времени - каждый элемент загружается и обрабатывается за время, малое по отношению к общему времени обработки, после чего он больше не нужен. В результате в памяти потокового процессора для

					МИВУ 09.03.02-00.000	Лист 4
Из	Лист	№ докум.	Подп.	Дата		

каждого «вычислителя» можно хранить только данные, необходимые для обработки одного элемента, в отличие от центральных процессоров с моделью произвольно зависимых данных.

Со времени своего появления в начале 1980-х годов, персональные компьютеры развивались в основном как машины для выполнения программ, сложных по внутренней структуре, содержащих большое количество ветвлений, интенсивно взаимодействующих с пользователем, но редко связанных с потоковой обработкой большого количества однотипных данных. Центральные процессоры ПК оптимизировались для решения именно таких задач, поэтому характеризовались следующим:

- большим количеством блоков для управления исполнением программы (кеширование данных, предсказание ветвлений и т.п.) и сравнительно малым количеством блоков для вычислений;
- архитектурой, оптимальной для программ со сложным потоком управления (обработка разнородных команд и данных, организация взаимодействия программ между собой и с пользователем);
- памятью с максимальной скоростью произвольного доступа к данным.

Увеличение производительности CPU в основном было связано с увеличением тактовой частоты и размеров высокоскоростной кешпамяти (память, расположенная прямо на процессоре). Программирование CPU для ресурсоемких научных вычислений подразумевает тщательное структурирование данных и порядка инструкций для эффективного использования всех уровней кеш-памяти.

Ядра современных центральных процессоров являются суперскалярными, поддерживая векторную обработку (расширения SSE и 3DNow!), сами же CPU обычно содержат несколько ядер. Таким образом, в совокупности центральные процессоры могут реализовывать десятки параллельных вычислительных потоков. Однако графические процессоры включают в себя тысячи параллельных «вычислителей». Кроме того, при поточно-параллельных расчетах графические процессоры имеют преимущество благодаря следующим особенностям архитектуры:

- память GPU оптимизирована на максимальную пропускную способность (а не на скорость произвольного доступа, как у CPU), что ускоряет загрузку потока данных;

- бóльшая часть транзисторов графического процессора предназначена для вычислений, а не для управления исполнением программы;

- при запросах к памяти, за счет конвейерной обработки данных, не происходит приостановки вычислений.

Однако обработка ветвлений (исполнение операций условного перехода) на GPU менее эффективна, поскольку каждый управляющий блок обслуживает не один, а несколько вычислителей.

Таким образом, производительность одного GPU при хорошо распараллеливаемых вычислениях аналогична кластеру из сотен обычных вычислительных машин, причем графические процессоры сейчас поддерживают практически все операции, используемые в алгоритмах общего назначения:

- распространенные математические операции и функции вещественного аргумента. В рамках SM 4.0 поддерживаются целые числа и логические операции, а в SM 4.1 и CUDA — также и вещественные числа двойной (64-битной) точности;

- организацию циклов. В SM 3 длина циклов ограничена 255 итерациями, в SM 4 длина циклов не ограничена;

- операции условного перехода (которые исполняются сравнительно медленно, поскольку в составе GPU блоков управления меньше, чем вычислительных блоков).

### 1.3. Выбор средства

### 1.4. Пара слов о ...

[https://elibrary.ru/download/elibrary\\_21502913\\_61591470.pdf](https://elibrary.ru/download/elibrary_21502913_61591470.pdf)

					МИВУ 09.03.02-00.000	Лист
						6
Из	Лист	№ докум.	Подп.	Дата		

[https://elibrary.ru/download/elibrary\\_23636603\\_72998025.pdf](https://elibrary.ru/download/elibrary_23636603_72998025.pdf)

[https://elibrary.ru/download/elibrary\\_13919877\\_96111886.pdf](https://elibrary.ru/download/elibrary_13919877_96111886.pdf)

<https://cs.petrus.ru/~kulakov/courses/parallel/lect/arch.pdf>

<https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B8%D0%B4>

<https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%8F%D0%BF%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D1%85%D0%B8%D1%80%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D1%8B%D1%85%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9>

<https://elib.spbstu.ru/dl/2/5187.pdf/info>

[https://kpfu.ru/portal/docs/F\\_1709436329/mcc18a.pdf](https://kpfu.ru/portal/docs/F_1709436329/mcc18a.pdf)

<http://python-3.ru/page/multiprocessing>

<https://habr.com/ru/company/otus/blog/501056/>

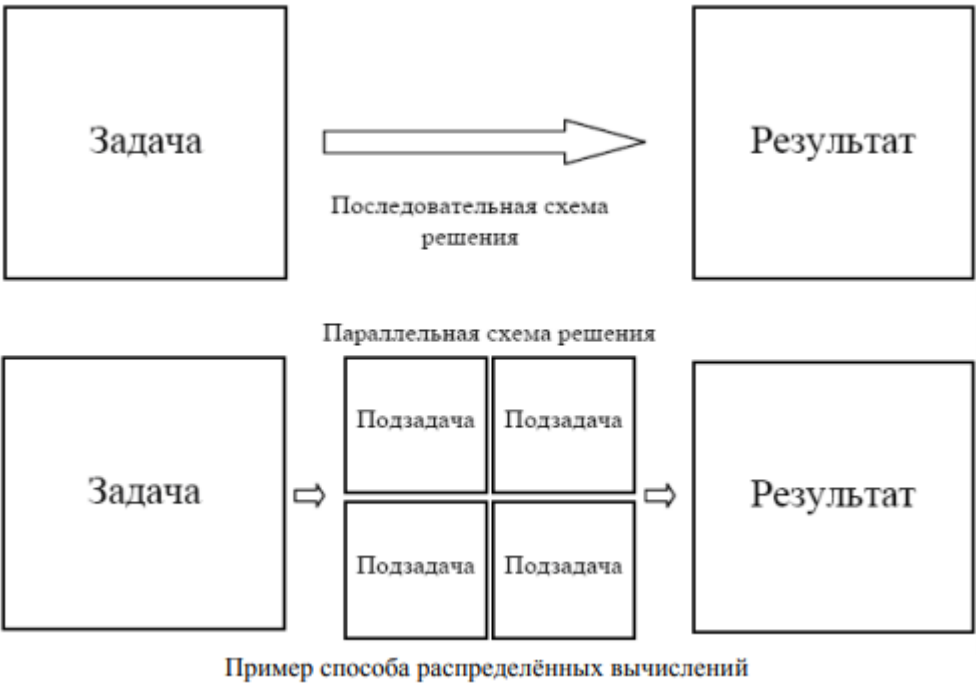
<http://window.edu.ru/resource/944/76944/files/book9.pdf>

[https://elibrary.ru/download/elibrary\\_13919877\\_96111886.pdf](https://elibrary.ru/download/elibrary_13919877_96111886.pdf)

					МИВУ 09.03.02-00.000	Лист
						7
Из	Лист	№ докум.	Подп.	Дата		

2. Проектирование

Тут про классы и архитектуру разрабатываемой системы





3. Тестирование

Заключение

					МИВУ 09.03.02-00.000	Лист
						10
Из	Лист	№ докум.	Подп.	Дата		

Список использованной литературы