

**Национальный исследовательский университет ИТМО**

**Факультет программной инженерии и компьютерной техники**

Отчёт по практическому заданию №1  
по дисциплине «Проектирование вычислительных систем»

Выполнил: Ларочкин Г.И  
Группа: Р3400  
Преподаватель: Пенской А.В.

Санкт-Петербург  
2020 г.

12:33 Thu 17 Sep

End Zoom

Mute Start Video Stop Video Participants More

Share Screen

Pen Color

Пользователи не имеют возможности обмениваться солюшенами. Это позволило бы сильно сократить время на прогон бенчмарка, если такие кейсы уже были проведены другими пользователями.

Пользователь не сможет получить солюшены, точно с теми кейсами, которые он задал. После процесса *benchmark common parameters* результат идёт в обязательном порядке в процесс *fork parameters* (процесс ищет возможные дополнительные оптимизации). Следует пользователю дать возможность запуск бенчмарка без дополнительных оптимизаций.

**Вывод**

В ходе данной лабораторной работы я приобрел навыки анализа вычислительных систем с архитектурной точки зрения, построил архитектурное описание проекта Tensile, ознакомился с инструментом документирования архитектуры Dataflow Diagram и произвёл анализ архитектурных проблем проекта.

— Ху.  $\sqrt{150 \cdot 1280}$

1) Гор. масштабы.

2) Воспроизводимость / достоверность.

— 11

5

Whiteboard Savepoint Pen Highlight Erase Line Rectangle Oval Text Color Erase Undo Redo Clear

## Цель

Приобрести навыки анализа вычислительных систем с архитектурной точки зрения, ознакомиться с инструментами документирования архитектуры.

## Описание проекта

В данной работе описан проект Tensile компании AMD. Это инструмент для создания benchmark-driven backend библиотеки для GEMM подобных алгоритмов, которые применяют операцию *multiplication* для многомерных объектов на видеокартах AMD. Пользователь может собрать таргетированную backend библиотеку для машинного обучения, основанную на наборе необходимых ему операций, архитектуры GPU, на которой планируется запуск, а также на наборе задаваемых пользователем ограничений (больше загрузить память, не кэшировать данные и т. д.)

## Жизненный цикл проекта

№	Этап	Роль для системы	Методы и средства
1	Задумка	Определение области применимости. Определение потенциальных пользователей.	Анализ потребностей рынка.
2	Анализ и постановка требований	Постановка требований к системе. Определение пользовательской базы.	Анализ потребностей внутренних проектов компании. Написание технического задания.
3	Проектирование Tensile benchmark	Проектирование архитектуры Tensile benchmark. Решение алгоритма поиска наилучшего solution.	Описание архитектуры с помощью диаграмм. Описание алгоритма, activity диаграммы.
4	Проектирование Tensile client	Проектирование интерфейса между библиотекой и пользовательским приложением.	Описание с помощью component & interface диаграммы.
5	Реализация Tensile client, Tensile benchmark	Реализация двух компонент системы. Модульное тестирование.	Написание кода, модульное тестирование
6	Интеграция client и benchmark	Реализация взаимодействия двух компонент системы. Тестирование работоспособности системы.	Написание кода, интеграционное тестирование
7	Внедрение и эксплуатация	Интеграция системы в высокоуровневые библиотеки (rocBLAS, MIOpen).	Совместная работа команд нескольких проектов
8	Поддержка системы	Поиск и исправление дефектов. Поддержка новых видеокарт.	Написание кода, тестирование
9	Вывод из эксплуатации	Вывод из эксплуатации	Вывод из эксплуатации

## **Анализ архитектурных проблем**

### **1. Масштабируемость**

Система должна адекватно работать вслед за выпуском и поддержкой новых архитектур и видеокарт.

### **2. Параметры оптимизации**

Система должна оперировать лишь с параметрами, которые могут существенно влиять на качество исполнения операции. Должен быть баланс между количеством параметров и скоростью исполнения benchmark.

### **3. Надежность исполнения**

Система должна работать с большим количеством параметров оптимизации, при этом результат операций должен быть воспроизводимым.

### **4. Надежность измерений**

Система должна качественно производить измерения скорости исполнения операции и занимаемых ресурсов устройства, результат должен быть воспроизводим.

### **5. Простота использования**

Система должна иметь простой и достаточный интерфейс для легкости внедрения библиотеки как во внутренние проекты компании, так и в пользовательские проекты.

## **Вывод**

В ходе данной лабораторной работы я приобрел навыки проектирования вычислительных систем и поиска архитектурных проблем.