



Знакомство с CUDA и системой LuNA



Выполнил: Веретенников А.А.

РП от организации: Перепёлкин В.А.

РП от НГУ: Малышкин В.Э.





Цель и задачи

Цель: получение практических навыков работы с системами автоматического конструирования параллельных программ и технологиями параллельных решений.

Задачи

1 Теоретические основы	Изучить теоретические основы фрагментированного программирования и системы LuNA
2 LuNA	Установить и настроить систему LuNA
3 Практическое задание N2	Разработать последовательные программы на C++ для скалярного произведения и умножения подматрицы на вектор
4 Пр. задание на LuNA	Разработать эквивалентные программы на языке LuNA
5 Сравнение	Сравнить результаты работы параллельных и последовательных программ
6 CUDA, Hello, World!	Установить CUDA Toolkit и выполнить тестовую программу



1 Теоретические основы

**ФВ — Фрагмент
Вычислений**

Последовательный
вычислительный
процесс,
задаваемый
последовательной
процедурой на
языке C++

**ФД — Фрагмент
Данных**

Входные и
выходные
аргументы ФВ

- ФД может быть
вычисленным и
невывисленным
- Если вычислен, то
его значение уже
не меняется

**ФК — Фрагмент
Кода**

C++ процедура,
реализующая
вычислительную
логику

Описываем множество фрагментов, но не предписываем
порядок их выполнения



2 LuNA

```
anvea@apc:~/luna$ luna exam
ples/tests_luna/basic1/test
.f
luna: generating recommenda
luna: generating cpp plugin
luna: constructing makefile
Hello world!
anvea@apc:~/luna$
```

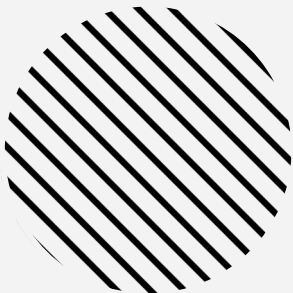
Проблема: в инструкции по установке указывалась необходимость установки Python 2.7. Однако эта версия не поддерживается в Ubuntu 24.04.

Решение: система LuNA была успешно собрана с использованием Python 3. Это не повлияло на процесс сборки и работоспособности системы.


Тестовый пример подтвердил корректность установки

3 и _4_ и _5_

**Практическое
задание N2**



Скалярное произведение



```
anvea@apc:~/practice/practical-task-2/s  
calar-product/cpp$ make  
g++ csp.cpp ucodes.cpp -o csp -std=c++1  
1  
anvea@apc:~/practice/practical-task-2/s  
calar-product/cpp$ ./csp  
Result: 182560.492714  
anvea@apc:~/practice/practical-task-2/s  
calar-product/cpp$ cd ../luna/  
anvea@apc:~/practice/practical-task-2/s  
calar-product/luna$ luna sp.fa  
Result: 182560.492714  
anvea@apc:~/practice/practical-task-2/s  
calar-product/luna$
```

Проблема: в директории с .fa лежал csp.cpp, из-за которого не удавалось корректно запустить .fa.

Решение: переместить csp.cpp в другую директорию.

Аналогично для умножения подматрицы на вектор



6 Установка CUDA для NVIDIA GeForce GTX 670

Проблема N1

Изначально была установлена CUDA 11.8. Не поддерживает архитектуру sm_30.

Решение:

Поддержка sm_30 была удалена начиная с CUDA 11.0. Поэтому была установлена максимальная совместимая версия — CUDA 10.2.

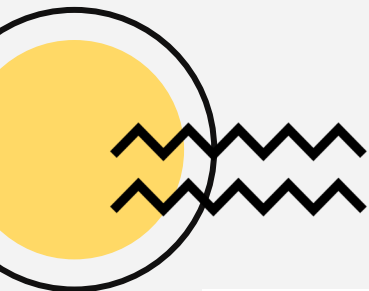
Проблема N2

CUDA 10.2 требует gcc версии не выше 8. В Ubuntu 24.04 установлен gcc 13.3.0.

Решение:

Установить gcc-8 (g++-8) и компилировать с флагом -cbin g++-8.





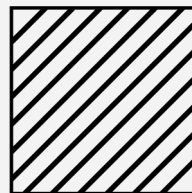
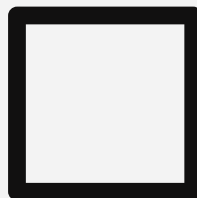
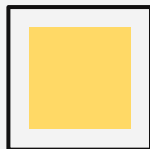
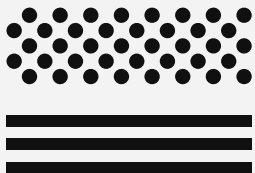
```
anvea@apc:~/cuda_test$ ls
Makefile hello2 hello2.cu
anvea@apc:~/cuda_test$ make
nvcc -arch=sm_30 -ccbin g++-8 hello.cu -o hello
anvea@apc:~/cuda_test$ ./hello
Hello from CPU!
Hello from GPU thread 0!
Hello from GPU thread 1!
Hello from GPU thread 2!
Hello from GPU thread 3!
Hello from GPU thread 4!
Hello from GPU thread 5!
Hello from GPU thread 6!
Hello from GPU thread 7!
Hello from GPU thread 8!
Hello from GPU thread 9!
anvea@apc:~/cuda_test$
```


Hello, World! на CUDA




Планы на будущее

Реализовать плагин для поддержки распараллеливания на видеокарте с помощью CUDA в системе LuNA





Спасибо за внимание!



Есть ли вопросы?

