Национальный исследовательский ядерный университет «МИФИ» (Московский Инженерно–Физический Институт)

Кафедра №42 «Криптология и кибербезопасность»

**Лабораторная работа №1:**

**«Введение в Docker»**

Антон Гатченко Б22-525

2024 г.

*Используемая рабочая среда:*

* Процессор - AMD Ryzen 5 5600H (laptop), 6c/12t
* Оперативная память – DDR4 16 ГБ
* ОС - Windows 10 Pro 22H2 19045.4780, 64 bit

*Ход работы:*

В качестве программу на языке С++, способной полностью загрузить процессор на хост-машине, была выбрана программа с алгоритмом сортировки Шелла, распараллеленная с помощью OpenMP. Сортировался массив размером элементов. Время работы программы составило около 55 секунд.

*Dockerfile:*

FROM alpine – указывает на образ, на котором будет базироваться наш образ

RUN apk update && apk add g++ git – установка g++ и git.

WORKDIR /app – назначение рабочей директории, относительно которой будут выполняться последующие команды (RUN, CMD).

RUN git clone https://github.com/Arondy/PARVPO\_lab1 . – клонируем исходник программы в текущую папку (app в нашем случае)

RUN g++ -o lab1 -fopenmp main.cpp – компилируем программу

CMD ./lab1 > /output/output.txt – запуск процесса и перенаправление stdout в файл output.txt, который будет находиться в папке output.

Для билда использовалась команда **docker build -t lab1** **.**, а также **docker builder prune** для очистки слоев (кэшированных данных билда). Запуск осуществлялся командой **docker run -v ${pwd}\output:/output –name lab1 lab1**, таким образом была смонитрована директория *output*, чтобы в нее сохранялся результат выполнения программы.

*Заключение:*

В ходе работы был написан Dockerfile на основе alpine, который устанавливал необходимые инструменты для компиляции и запуска программы, код которой получался с github.

Время выполнения на хост-машине - 51.714 с; в контейнере - 56.620 с, на 9% медленнее, что можно считать незначительным в контексте использования контейнеров. Использование памяти – МБ (Приложение 2). Процессор использовался практически полностью.

Небольшое снижение производительности контейнера связано с дополнительными накладными расходами на виртуализацию и управление ресурсами, которые создаёт Docker.

*Приложение:*

1. git репозиторий с .cpp и Docker файлами:

<https://github.com/Arondy/PARVPO_lab1>

1. Скриншот docker stats:

