

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности

Отчёт
по дисциплине «Системное программное обеспечение GNU/Linux»
на-тему
«Оценка производительности системы»

Выполнил: Кузнецов Виталий Витальевич

Группа: 5130904/30004

Преподаватель: Петров Александр
Владимирович

Оглавление

1	Введение.....	3
1.1	Цели и задачи.....	3
1.2	Платформа.....	3
2	Основная часть.....	3
2.1	Установка и запуск Phoronix test suite.....	3
2.2	Установка бенчмарков	4
2.3	Test-fio.....	4
2.4	Test-proc.....	6
2.5	Test-stream.....	6
2.6	Test-iperf	7
2.7	Test-gputest.....	8
3	Вывод.....	10

1 Введение

1.1 Цели и задачи

Цель – оценка производительности системы.

Задачи:

1. Установить и запустить Phoronix test suite
2. Установить и запустить бенчмарки.
3. Проанализировать полученные данные и подвести итог.

1.2 Платформа

Данные получены с помощью команды: `phoronix-test-suite system-info`

PROCESSOR: AMD Ryzen 5 5500U @ 2.10GHz

Core Count: 6

Thread Count: 12

Extensions: SSE 4.2 + AVX2 + AVX + RDRAND + FSGSBASE

Cache Size: 8 MB

Microcode: 0x8608103

Core Family: Zen 2

Scaling Driver: `acpi-cpufreq schedutil` (Boost: Enabled)

GRAPHICS: AMD Lucienne 512MB

Frequency: 1800/400MHz

BAR1 / Visible vRAM: 512 MB

Screen: 1920x1080

MOTHERBOARD: HUAWEI BOM-WXX9-PCB-B2

BIOS Version: 2.12

Chipset: AMD Renoir/Cezanne

Audio: AMD Renoir Radeon HD Audio

Network: Realtek RTL8822CE 802.11ac PCIe

MEMORY: 2 x 8GB DDR4-2667MT/s Micron 4ATF1G64HZ-3G2E1

DISK: PCIe-8 SSD 512GB

File-System: ext4

Mount Options: `errors=remount-ro relatime rw`

Disk Details: Block Size: 4096

OPERATING SYSTEM: Ubuntu 22.04

Kernel: 6.2.1 (x86_64)

Desktop: GNOME Shell 42.9

2 Основная часть

2.1 Установка и запуск Phoronix test suite

Устанавливал с сайта <https://ubunlog.com/en/phoronix-test-suite-a-multiplatform-benchmark-tool/>.

Набор команд для установки, которые указаны на сайте:

```
sudo apt-get install phoronix-test-suite
```

```
wget http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite\_8.6.0\_all.deb
```

```
sudo dpkg -i phoronix-test-suite_8.6.0_all.deb
```

```
sudo apt -f install
```

2.2 Установка бенчмарков

Чтобы запустить тестирование нужно использовать команду:

```
phoronix-test-suite benchmark < Название теста >
```

Список тестов:

1. pts/fio
2. pts/build-linux-kernel
3. pts/stream
4. pts/iperf
5. pts/gputest

Тесты были выбраны по популярности в интернете.

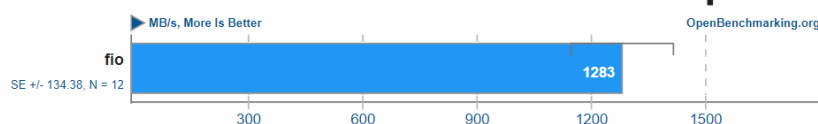
После того, как программа прошла тест, нам предложат перейти на сайт, где будут видны результаты (оттуда я их и брал).

2.3 Test-fio

Данный тест тестирует диск. Для теста использовал IO_uring, размер блока 8 мб, 12 потоков. Для тестировки выбрал случайное чтение, последовательное чтение, случайную запись и последовательную запись.

Flexible IO Tester 3.36

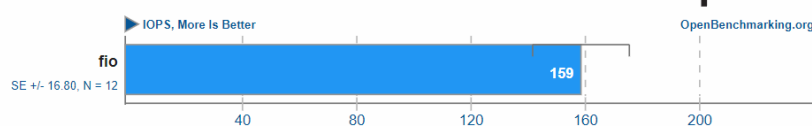
Type: Random Write - Engine: IO_uring - Direct: Yes - Block Size: 8MB - Job Count: 12 - Disk Target: Default Test Directory



1. (CC) gcc options: -rdynamic -ll -lnuma -lz -lm -laio -lpthread -ldl -std=gnu99 -ffast-math -include -O3 -fcommon -march=native

Flexible IO Tester 3.36

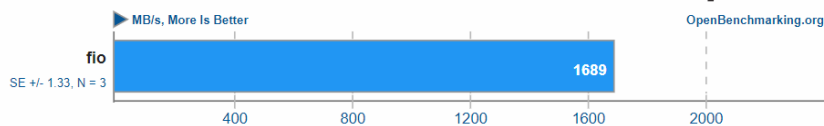
Type: Random Write - Engine: IO_uring - Direct: Yes - Block Size: 8MB - Job Count: 12 - Disk Target: Default Test Directory



1. (CC) gcc options: -rdynamic -ll -lnuma -lz -lm -laio -lpthread -ldl -std=gnu99 -ffast-math -include -O3 -fcommon -march=native

Flexible IO Tester 3.36

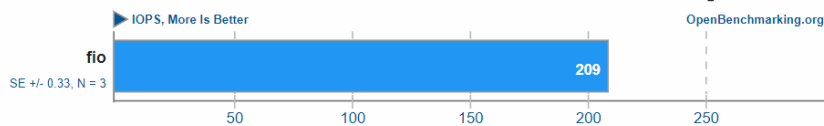
Type: Random Read - Engine: IO_uring - Direct: Yes - Block Size: 8MB - Job Count: 12 - Disk Target: Default Test Directory



1. (CC) gcc options: -rdynamic -l -lnuma -lz -lm -laio -lpthread -ldl -std=gnu99 -ffast-math -include -O3 -fcommon -march=native

Flexible IO Tester 3.36

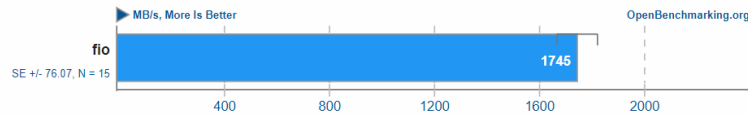
Type: Random Read - Engine: IO_uring - Direct: Yes - Block Size: 8MB - Job Count: 12 - Disk Target: Default Test Directory



1. (CC) gcc options: -rdynamic -l -lnuma -lz -lm -laio -lpthread -ldl -std=gnu99 -ffast-math -include -O3 -fcommon -march=native

Flexible IO Tester 3.36

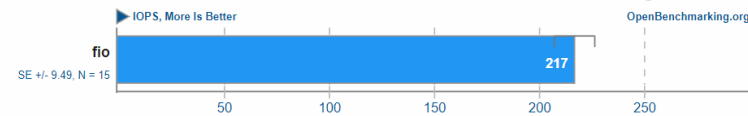
Type: Sequential Write - Engine: IO_uring - Direct: Yes - Block Size: 8MB - Job Count: 12 - Disk Target: Default Test Directory



1. (CC) gcc options: -rdynamic -l -lnuma -lz -lm -laio -lpthread -ldl -std=gnu99 -ffast-math -include -O3 -fcommon -march=native

Flexible IO Tester 3.36

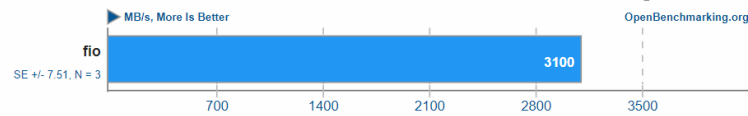
Type: Sequential Write - Engine: IO_uring - Direct: Yes - Block Size: 8MB - Job Count: 12 - Disk Target: Default Test Directory



1. (CC) gcc options: -rdynamic -l -lnuma -lz -lm -laio -lpthread -ldl -std=gnu99 -ffast-math -include -O3 -fcommon -march=native

Flexible IO Tester 3.36

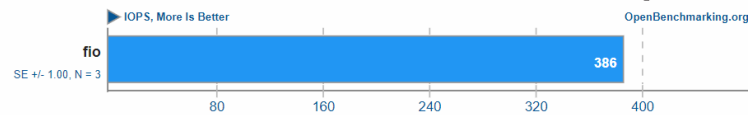
Type: Sequential Read - Engine: IO_uring - Direct: Yes - Block Size: 8MB - Job Count: 12 - Disk Target: Default Test Directory



1. (CC) gcc options: -rdynamic -l -lnuma -lz -lm -laio -lpthread -ldl -std=gnu99 -ffast-math -include -O3 -fcommon -march=native

Flexible IO Tester 3.36

Type: Sequential Read - Engine: IO_uring - Direct: Yes - Block Size: 8MB - Job Count: 12 - Disk Target: Default Test Directory



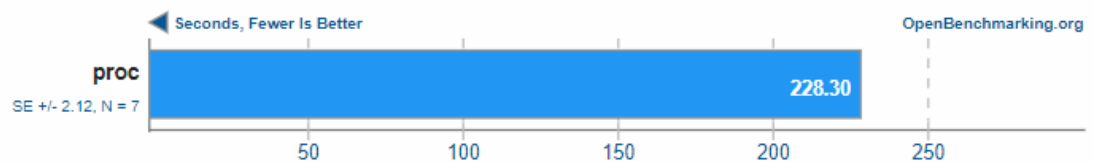
1. (CC) gcc options: -rdynamic -l -lnuma -lz -lm -laio -lpthread -ldl -std=gnu99 -ffast-math -include -O3 -fcommon -march=native

2.4 Test-proc

Данный тест собирает ядро со стандартным конфигом и фиксирует время его сборки.

Timed Linux Kernel Compilation 6.8

Build: defconfig



2.5 Test-stream

Для теста на оперативную память я выбрал Copy, Scale, Add и Triad.

Copy - проверяет скорость передачи данных внутри RAM. Он включает в себя копирование больших блоков данных из одного места в памяти в другое и измерение времени, необходимого для выполнения этой операции.

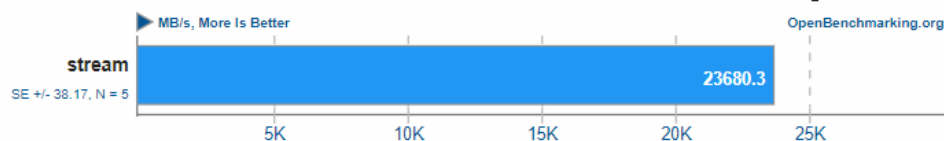
Scale - похож на тест "add", но с дополнительным множителем. Это позволяет проверять производительность RAM при работе с более сложными вычислениями.

Add - проверяет способность RAM выполнять базовые арифметические операции сложения. Обычно он включает в себя сложение больших массивов чисел, хранящихся в памяти, и проверку результатов на корректность.

Triad - скорее всего относится к комбинации трех вышеупомянутых тестов (add, copy, scale) в одном бенчмарке. Это позволяет получить более полное представление о производительности RAM.

Stream 2013-01-17

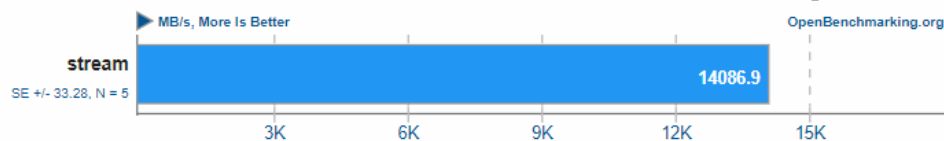
Type: Copy



1. (CC) gcc options: -O3 -march=native -fopenmp

Stream 2013-01-17

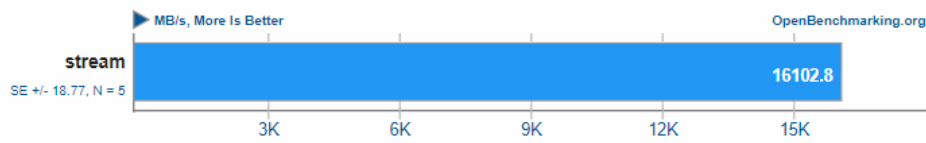
Type: Scale



1. (CC) gcc options: -O3 -march=native -fopenmp

Stream 2013-01-17

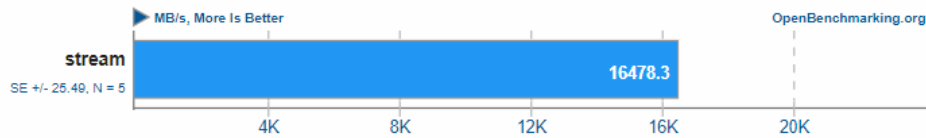
Type: Triad



1. (CC) gcc options: -O3 -march=native -fopenmp

Stream 2013-01-17

Type: Add



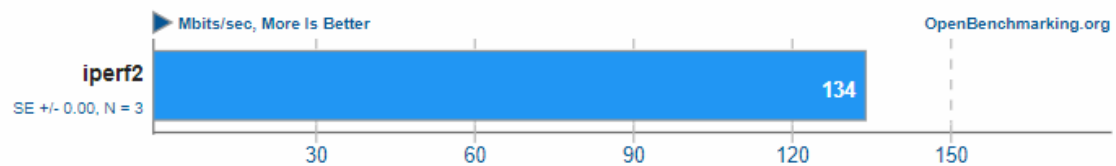
1. (CC) gcc options: -O3 -march=native -fopenmp

2.6 Test-iperf

Тест на сеть:

iPerf 3.14

Server Address: localhost - Server Port: 5201 - Duration: 360 Seconds - Test: UDP - Parallel: 128

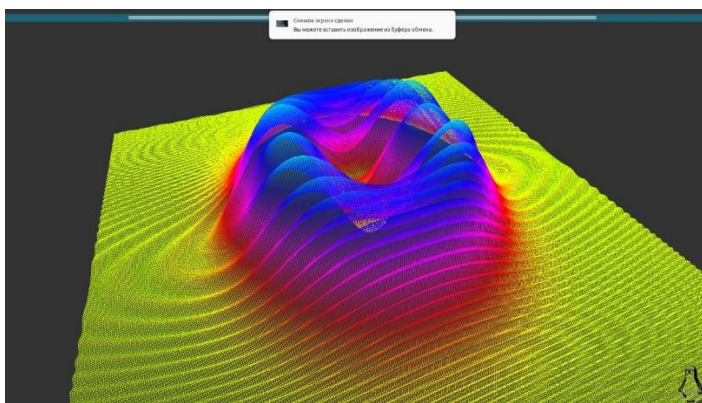
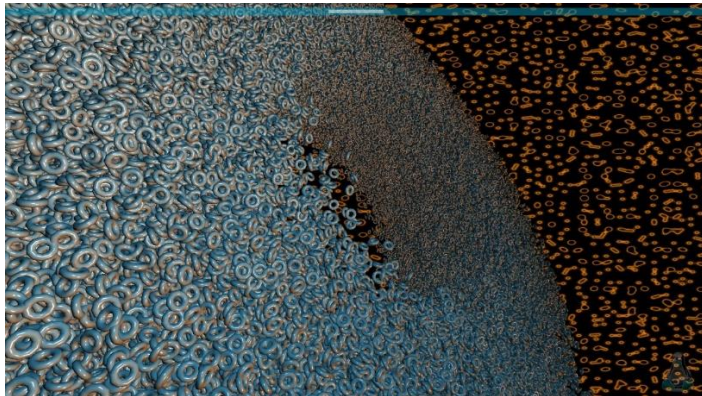


1. (CC) gcc options: -O3 -march=native -lssl -lcrypto -lm

2.7 Test-gputest

Тест на видеокарту:

Фотографии с теста видеокарты:

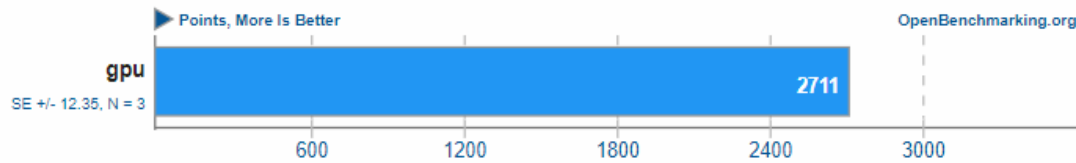


GpuTest 0.7.0

Test: GMark - Resolution: 1280 x 1024 - Mode: Fullscreen



OpenBenchmarking.org

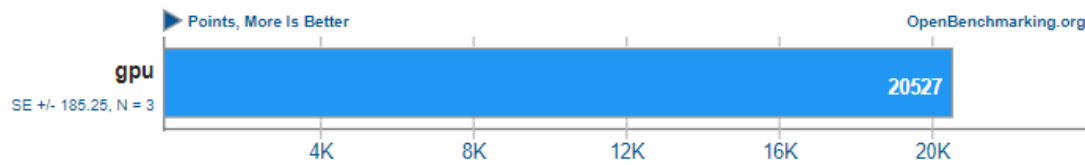


GpuTest 0.7.0

Test: Plot3D - Resolution: 1280 x 1024 - Mode: Fullscreen



OpenBenchmarking.org

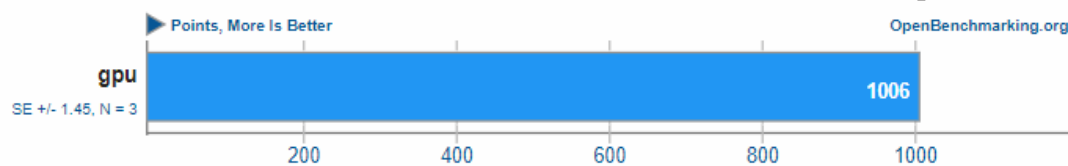


GpuTest 0.7.0

Test: Furmark - Resolution: 1280 x 1024 - Mode: Fullscreen



OpenBenchmarking.org

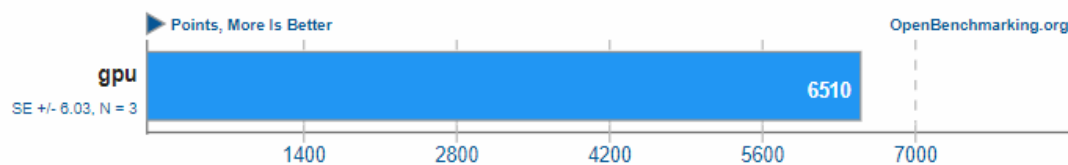


GpuTest 0.7.0

Test: TessMark - Resolution: 1280 x 1024 - Mode: Fullscreen

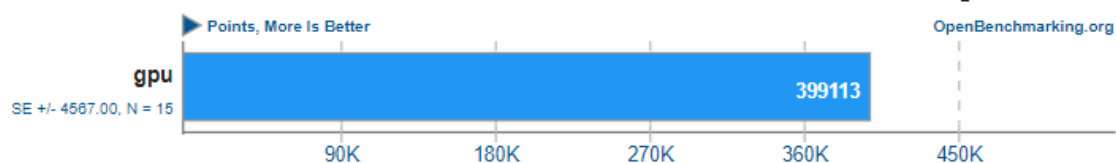


OpenBenchmarking.org



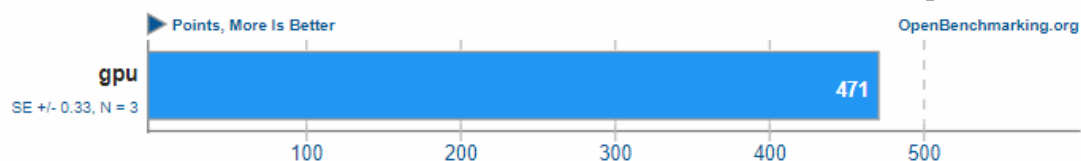
GpuTest 0.7.0

Test: Triangle - Resolution: 1280 x 1024 - Mode: Fullscreen



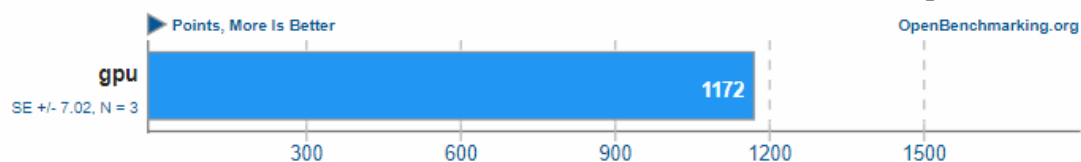
GpuTest 0.7.0

Test: Pixmark Piano - Resolution: 1280 x 1024 - Mode: Fullscreen



GpuTest 0.7.0

Test: Pixmark Volplosion - Resolution: 1280 x 1024 - Mode: Fullscreen



3 Вывод

Phroronix-test-suite для меня показался очень удобным. Было интересно протестировать свой компьютер и сверить данные с другими пользователями. Проблем для себя я не обнаружил.