Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 1**

по дисциплине

‘Операционные системы’

*Выполнил:*

Студент группы P33111

Павлов Александр Сергеевич

*Преподаватель:*

Осипов С. В.

Изображение выглядит как Шрифт, логотип, Графика, белый

Автоматически созданное описание

Санкт-Петербург, 2023

# Описание задания

Подбирая числовые значения для выданных параметров, и используя средства мониторинга, добиться **максимальной** производительности системы (BOGOPS, FLOPS, Read/Write Speed, Network Speed).

cpu: [int128double, ackermann]

cache: [cache-level, cache-fence]

io: [io-uring, ioport]

memory: [mmaphuge-mmaps, shm]

network: [sockdiag, netlink-task]

pipe: [sigpipe, pipe-ops]

sched: [sched-prio, yield]

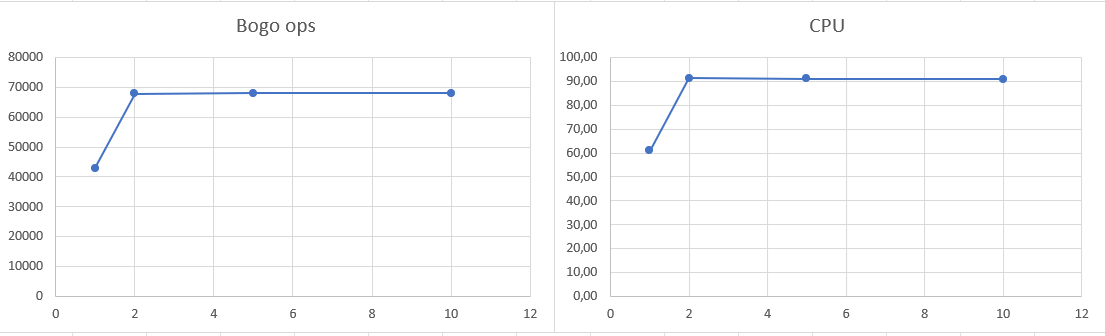
# CPU

Int128double

Тестирование процессора будет выполняться с использованием операций над 128-битными целыми числами и операций с двойной точностью с плавающей запятой.

В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --cpu $i --cpu-method int128double --metrics --timeout 10 | sar 1 10 -u, где i ∈ {1, 2, 5, 10}



Ackermann

Тестирование процессора будет выполняться с использованием функции Аккермана для оценки производительности и нагрузки на процессор, особенно на рекурсивные алгоритмы.

В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --cpu $i --cpu-method ackermann --metrics --timeout 10 | sar 1 10 -u, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, число, диаграмма

Автоматически созданное описание

# Cache

Cache-level

Тестирование кэша процессора. Будет протестирован первый уровень кэша, так как он самый быстрый.

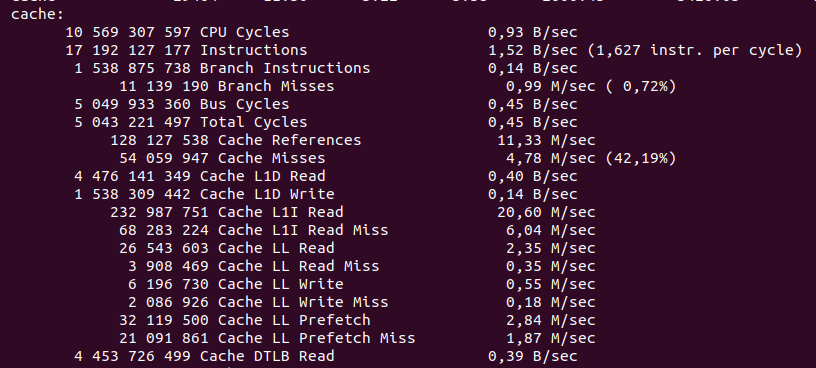
В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --cache $i --cache-level 1 --metrics --timeout 10 | sar 1 10 -u, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, диаграмма

Автоматически созданное описание

Посмотрим на поведение системы с помощью флага –perf для 1, 2, 5, 10 воркеров:

Изображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

Cache-fence

Принудительная сериализация записи для каждой операции хранения.

В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --cache $i --cache-fence --metrics --timeout 10 | sar 1 10 -u| sar 1 10 -u, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, диаграмма, текст

Автоматически созданное описание

Посмотрим на поведение системы с помощью флага –perf для 1, 2, 5, 10 воркеров:

Изображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

# IO

IOuring

Тестирование системы ввода-вывода с помощью механизма io\_uring. Нагрузка в виде операций чтения, записи, открытия и закрытия файлов. На каждый цикл 1024x512 байт пишутся и читаются во временный файл.

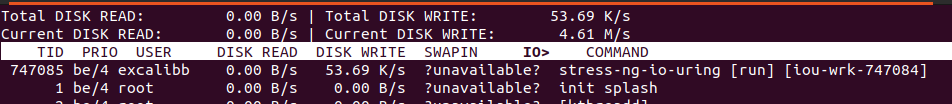
В качестве изменяемого числового параметра будем использовать количество воркеров.

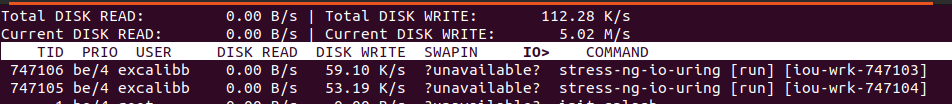
stress-ng --io-uring $i --metrics --timeout 10 | sar 1 10 -b, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, диаграмма, число

Автоматически созданное описание

Посмотрим на поведение системы с помощью команды iostat для 1, 2, 5, 10 воркеров:





Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, линия, График, снимок экрана

Автоматически созданное описание

IOport

Выполняет пакеты из 16 операций записи и чтения через порт 0х80. Нагрузка приведёт к задержкам в центральном процессоре.

В качестве изменяемого числового параметра будем использовать количество воркеров.

sudo stress-ng --ioport $i --metrics --timeout 10 | sar 1 10 -u, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, число, диаграмма

Автоматически созданное описание

# Memory

Mmaphuge-mmaps

Нагрузка на систему путём создания большого количества отображений памяти большого размера.

В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --mmaphuge $i --mmaphuge-mmaps 8192 --metrics --timeout 10 | sar 1 10 -r, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, диаграмма, число

Автоматически созданное описание

Посмотрим на поведение системы с помощью команды dstat -m для 1, 2, 5, 10 воркеров:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Shm

Нагрузка на систему путём использования разделяемой памяти, а именно: разделение и выделение объектов общей памяти, используя POSIX-интерфейсы.

В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --shm $i --metrics --timeout 10 | sar 1 10 -r, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, диаграмма, число

Автоматически созданное описание

Посмотрим на поведение системы с помощью команды dstat -m для 1, 2, 5, 10 воркеров:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

# Network

Sockdiag

Нагрузка на систему путём генерации сетевого трафика и выполнения диагностики сокетов.

В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --sockdiag $i --metrics --timeout 10 | sar 1 10 -n SOCK, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, чек, диаграмма

Автоматически созданное описание

Netlink-task

Нагрузка на сетевой стек ядра с помощью механизма Netlink. Сбор статистики задач с помощью Netlink.

В качестве изменяемого числового параметра будем использовать количество воркеров.

sudo stress-ng --netlink-task $i --metrics --timeout 10 | sar 1 10 -n SOCK, где i ∈ {1, 2, 5, 10}

Изображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание

# Pipe

Sigpipe

Нагрузка на систему путём неоднократного порождения дочернего процесса, который завершается до того, как родительский процесс завершает запись в канал, вызывая сигнал SIGPIPE.

В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --sigpipe $i --metrics --timeout 10 | sar 1 10 -u, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, текст, График, число

Автоматически созданное описание

Pipe-ops

Останавливает pipe-воркеров после достижения определенного количества операций записи в pipe.

В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --pipe $i --pipe-ops 1000000 --metrics --timeout 10 | sar 1 10 -u, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, текст, диаграмма

Автоматически созданное описание

# Sched

Sched-prio

Позволяет установить определенный приоритет планировщику процессов.

В качестве изменяемого числового параметра будем использовать количество воркеров.

stress-ng --cpu $i --sched-prio 1 --metrics --timeout 10 | sar 1 10 -u, где i ∈ {1, 2, 5, 10}

Изображение выглядит как линия, График, число, диаграмма

Автоматически созданное описание

Yield

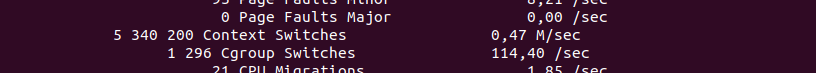
Нагрузка на систему путём системного вызова sched\_yield(2). Как минимум 2 дочерних процесса на каждый процессор будут использовать schield\_yield(2), независимо от количества воркеров. Это обеспечит быстрое переключение контекста.

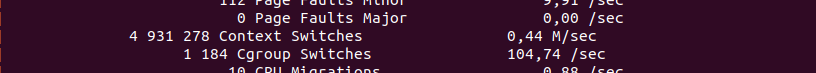
В качестве изменяемого числового параметра будем использовать количество воркеров.

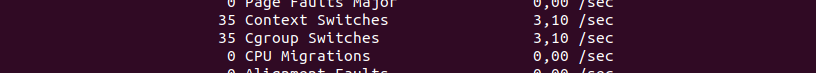
stress-ng --yield $i --metrics --timeout 10 | sar 1 10 -u, где i ∈ {1, 2, 5, 10}

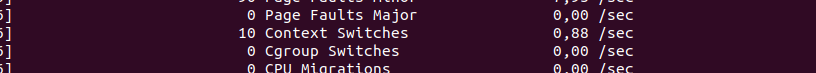
Изображение выглядит как линия, График, число, диаграмма

Автоматически созданное описание









Изображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание

# Вывод

Во время выполнения лабораторной работы я научился проводить нагрузочное тестирование Linux и профилирование при помощи различных средств мониторинга.