

## **Лабораторная работа №6 (вариант повышенной сложности). Экспериментальная оценка параметров производительности операционной системы**

### **Цель лабораторной работы**

Оценить реальные накладные расходы на параллельное выполнение задач в условиях преимущественного использования ресурса процессора или ресурса дисковой подсистемы.

Для выполнения лабораторной работы необходимо организовать и провести две группы экспериментов.

### **1. Эксперименты с последовательным и параллельным выполнением вычислительно сложных задач**

Для первой группы экспериментов необходимо найти или придумать самим вычислительно сложный алгоритм. Алгоритм должен минимально требовать входные данные и выгружать результаты (в идеале, получает на вход несколько числовых параметров и на выходе формирует одно или несколько числовых значений), но требовать для завершения работы значительного времени, порядка 2-3 секунд или более. Например, алгоритм может вычислять с высокой точностью значение некоторой сложной функции в заданной точке. Реализацию алгоритма можно сделать как на C/C++, так и прямо на bash.

Идея эксперимента. Необходимо получить N результатов выполнения созданного алгоритма с разными комбинациями входных параметров (например, результаты вычисления значения функции в N точках). Важно, что исполнение алгоритма с каждой комбинацией входных параметров должно требовать приблизительно равного времени на выполнение. Эксперимент будет проходить в два этапа. На первом этапе вычисления каждого из N результатов будут осуществляться последовательно друг за другом, а во втором случае параллельно. При этом будет замеряться суммарное время, затрачиваемое на получение всех N результатов.

Для нивелирования влияния случайных процессов в ОС на результаты эксперимента, каждый эксперимент должен проводиться 5-10 раз с усреднением результатов.

Также необходимо оценить различия в результатах в системе с одним или с несколькими процессорами.

В результате можно предложить такой план проведения экспериментов (его можно корректировать, желательно, обсуждая с преподавателем).

#### **1. Первый эксперимент.**

- a. В виртуальной машине настроить использование одного процессора.
- b. Написать скрипт, который будет, получив параметр N, запускать последовательно друг за другом N вычислений для разных значений входных параметров. Запуск вычисления для следующего параметра должен происходить сразу после завершения предыдущего вычисления.
- c. С помощью другого скрипта для каждого N в диапазоне от 1 до 20 запускать 10 раз запускающий скрипт из пункта 2 через утилиту time и фиксировать время, затраченное на полное выполнение запускающего скрипта. На выходе должно получиться 20 серий по 10 значений.
- d. В каждой серии посчитать среднее арифметическое значений и построить график зависимости этих усредненных оценок от N.

#### **2. Второй эксперимент.**

- a. Написать скрипт, который, получив параметр N, параллельно (не дожидаясь завершения предыдущего) запустит N вычислений для разных входных параметров.
- b. Повторить пункты "c" и "d" из первого эксперимента с использованием написанного в предыдущем пункте запускающего скрипта.

#### **3. Третий и четвертый эксперимент – повторить первый и второй эксперимент, выделив в настройках виртуальной машины 2 процессора.**

В результате выполнения этой группы экспериментов должны получиться 4 графика.

### **2. Эксперимент с параллельным и последовательным выполнением задач с большими объемами считываемых и сохраняемых данных**

Для второй группы экспериментов необходимо реализовать алгоритм, который будет обрабатывать большие массивы данных, но с простыми вычислениями. Например, создать 20 файлов, размером по 10 МБайт (можно меньше, нужно смотреть по реальной производительности тестовой системы), представляющими

последовательность целых чисел. Задачей алгоритма будет в одном файле умножить все значения на два и дописать значения в конце файла. Алгоритм должен считывать очередное значение из файла, умножать его на 2 и сохранять результат, дописывая в конце файла получившееся значение, а затем переходить к следующему значению и т.д. Важно реализовать именно такое взаимодействие с файлом: чтение отдельного значения – изменение – сохранение измененного значения в файл. Нельзя формировать сразу все значения в памяти и потом записывать разом в файл – это будет уже другой эксперимент. Размер исходного файла нужно подобрать так, чтобы обработка одного файла занимала 2-3 секунды.

Далее нужно повторить эксперименты по плану из первой группы. Важно, чтобы каждый запущенный алгоритм работал со своим отдельным файлом.

Результатом выполнения этой группы экспериментов будет еще 4 графика.

Можно и приветствуется, если Вы дополните эксперименты так, чтобы кроме оценки времени выполнения наборов заданий, оценивались и другие параметры ОС: память, потребляемые отдельными процессами ресурсы и т.д. Выбор таких параметров и методов их оценки на усмотрение экспериментаторов.

### **Оформление результатов**

После получения всех результатов, необходимо оформить отчет, в котором описать:

1. Детали реализации каждого алгоритма.
2. Параметры виртуальной машины и хостового компьютера, на которых проводился эксперимент.
3. Уточненный (возможно) план эксперимента.
4. Графики с описанием, в рамках какого эксперимента каждый из них получен.
5. Выводы. Попытаться теоретически объяснить читаемые на графиках явления.