

Задание №7 в рамках вычислительного практикума. Исследование характеристик программного обеспечения

Кострицкий А. С., Ломовской И. В.

Москва — 2023 — TS2304271005

Содержание

1 Задание №1	1
2 Задание №2	1

1 Задание №1

Используя функцию `nanosleep` и каждый из четырёх известных Вам способов замеров времени (`gettimeofday`, `clock_gettime`, `clock`, `__rdtsc`), исследовать среднее значение времени выполнения вызовов функции `nanosleep` для задержки в 1с, 100мс, 50мс, 10мс.

2 Задание №2

На основе задачи №4 ЛР№2 (сортировка) по курсу «Программирование на Си» проведите сравнение производительности работы программы по трём плоскостям:

1. Разные способы работы с элементами одномерного массива:
 - (a) использование операции индексации `a[i]`;
 - (b) формальная замена операции индексации на выражение `*(a + i)`;
 - (c) использование указателей для работы с массивом.
2. Разные уровни оптимизации: 00, 02.
3. Разные исходные массивы: наилучший случай (все элементы изначально отсортированы), элементы расположены в случайном порядке.

Иначе говоря, при компиляции должны получиться $3 \times 2 \times 2 = 12$ вариантов программы. Для сравнения производительности следует реализовать несколько скриптов:

1. `build_apps.sh`, вызвав который, можно получить весь набор необходимых исполняемых файлов.
2. `update_data.sh`, вызвав который, можно добавить некоторые данные в датасет данных исследования.

3. `make_preproc.sh|py`, вызвав который, можно подготовить данные из набора, провести первичный анализ: посчитать среднее арифметическое, медианное, найти максимум и минимум, вычислить нижний и верхний квартили, etc.
4. `make_postproc.sh|py`, вызвав который, можно получить указанные ниже графики.
5. `go.sh`, вызвав который, можно получить данные исследования (скрипт вызывает по очереди предыдущие четыре).

В отчёте привести следующие **графики**:

1. Обычный кусочно-линейный график зависимости времени выполнения в любых единицах измерения времени от числа элементов массива для всех 6 вариантов программы, обрабатывающей наилучший случай.
2. Обычный кусочно-линейный график зависимости времени выполнения в любых единицах измерения времени от числа элементов массива для всех 6 вариантов программы, обрабатывающей массив общего вида.
3. Кусочно-линейный график с ошибкой (среднее, максимум, минимум) для всех вариантов обработки массива при уровне оптимизации 02.
4. График с усами (среднее, максимум, минимум; нижний, средний и верхний квартили) для варианта обработки «через квадратные скобки» при уровне оптимизации 02.

В отчёте привести **таблицы**:

1. Таблицы ко всем указанным графикам. Напоминаем, что график, если не ставится цель лишь схематично изобразить что-либо, всегда строится по табличным данным. Столбцы: длина массива n , время выполнения t_n , величина *относительной стандартной ошибки среднего*.
2. Таблицу для результатов обработки «через квадратные скобки» с уровнем оптимизации 02 со столбцами: длина массива n , время выполнения t_n , величина $\frac{\ln(t_{i+1}) - \ln(t_i)}{\ln(n_{i+1}) - \ln(n_i)}$ для всех строк, кроме последней.

Если Вы проводили исследование с большим количеством вариантов длины массива (получилось много строк в таблицах), достаточно оставить в таблицах десять строк с данными.

Примечания:

1. Помните, что для уменьшения влияния побочных эффектов время замеряют только у целевого алгоритма.
2. Проводить эксперимент следует со массивами размером от минимального (в некоторых задачах это 1, в некоторых – несколько больше) до, как минимум, 10000. Если производительность Вашего ПК позволяет собрать результаты 20 тестов в час для массивов большего размера, можно поместить эти результаты в отчёт. Если производительность ПК не позволяет на 10000 элементов собирать хотя бы 20 результатов в час, то максимальный размер обсуждается отдельно.
3. Шаг измерений – 500. Можно плотнее.

4. Помните, что для чистоты эксперимента состояние системы не должно зависеть от того, какой конкретно алгоритм был использован. Финальное измерение времени всегда проводится тогда, когда система приведена к ожидаемому состоянию.
5. Для отрисовки графиков можно использовать `gnuplot`, `matplotlib` или любой другой «лёгкий» пакет. Два требования: возможность нарисовать всё, что требуется в задании, и наличие консольного интерфейса.
6. При реализации скриптов на Питоне дополнительные пакеты разрешается использовать только в случае, когда стандартной библиотеки недостаточно. Все зависимости в этом случае следует указать. **Базовые функции работы со статистикой в стандартной библиотеке Питона есть.**
7. При желании разрешается добавить другие плоскости исследования: наличие/отсутствие `restrict`, различные максимальный и текущий размеры статического массива, другие флаги оптимизации, etc. Сам смысл задания заключается не только в том, чтобы собрать данные исследования, но и чтобы создать удобный для внесения условий нового исследования интерфейс – поэтому Вы можете проверить, насколько быстро и удобно Вам удалось добавить новое исследование в набор. Результаты дополнительных исследований в отчёт не выносятся, но интересные вопросы могут обсуждаться отдельно.

В отчёте объясните полученные результаты и приведите ответы на следующие **вопросы**:

1. Какой способ обработки быстрее и почему?
2. В датасете обнаружена серия экспериментов с одним результатом. Можно ли заменить её одним экспериментом?
3. Если заполнение случайными числами массива (или любая другая инициализация) присутствует в каждом эксперименте, то почему Вы измеряете время только у целевого алгоритма?