

Задачи по курсу “Эффективность загрузки ВС”

Вахитов Александр Тимурович

12 декабря 2014 г.

1 Необходимость синхронизации

1. Установить Visual Studio 2008 и boost 1_47.
2. Написать функцию вычисления числа Пи. Её нужно n ($n > 10000$) раз вызывать в цикле. Есть внешняя глобальная переменная counter, которая должна инкрементироваться при каждом вызове функции вычисления Пи.
3. Обернуть этот цикл в функцию, которую вызывать в отдельных потоках. Таких потоков нужно столько, чтобы была видна потеря значений счётчика. Не забыть join'ить потоки.
4. Замерить время работы с мьютексами и без них.

Цель: убедиться, что такая потеря имеет место.

Срок выполнения: 12 ноября 2014.

2 Читатель и писатель

1. Написать эмулятор устройства(камеры), который работает в два потока. Первый поток генерирует кадры 1000×1000 с белым шумом(заполняет случайными числами) и кладёт их в очередь. Второй достаёт данные из очереди и отправляет их следующему модулю.
2. Второй модуль состоит также из двух потоков. Первый поток кладёт в очередь обработки кадр при его получении. Второй поток выполняет downsample картинки из очереди обработки и сохраняет.

Нужно использовать Concurrent Queue из Threading Building Blocks(TBB), Boost::thread, cv::Mat (OpenCV 2.6) для работы с картинками.

Срок выполнения: 12 ноября 2014.

3 Число Пи (task parallelism)

Вычисление числа π методом иглы Бюффона. Алгоритм заключается в генерации точек внутри квадрата со сторонами 1 и подсчете числа точек, попавших в круг диаметра 1, вписанный в этот квадрат. Необходимо реализовать этот алгоритм, используя совместимую с многопоточным выполнением генерацию случайных чисел, поместить его внутрь TBB Task, и создать 1, 2, 4, 8, 16 задач, которые считают число пи, при этом всего генерируется одинаковое число точек (скажем, 10 миллионов). Для получения финального ответа, если задач было создано более одной, выполнить редукцию результата. Оценить качество результатов и время работы на каждом запуске.

Срок выполнения: 31 декабря 2014.

4 Уменьшение изображения (downsampling) на OpenCL

Задача заключается в уменьшении изображения размером 1000 на 1000 методом билинейной интерполяции в 2,5 раза. Алгоритм следующий:

1. смазать изображение гауссовским фильтром, у которого дисперсия равна 0,5 пикселя
2. сгенерировать точки, в которых будут находиться центры пикселей в уменьшенном изображении, в координатах исходного изображения
3. выполнить билинейную интерполяцию по ближайшим соседям и вычислить значения в центрах пикселей, сохранить их в изображение разрешения 250 на 250

Для вычисления с помощью OpenCL, необходимо написать ядро (kernel) для билинейной интерполяции. Остальные шаги выполнять на хост-компьютере, а этот шаг - на графической карте.

Срок выполнения: 31 декабря 2014.