

УДК 519.6

Б.Ж.Медетов, А.Е.Ыскак, К.А.Ниязалиев, А.С.Койшигарин

Казахский Национальный университет им. Аль-Фараби, г. Алматы, Казахстан

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНОГО ВЫЧИСЛЕНИЯ ПРИМЕНИТЕЛЬНО К СПЕКТРАЛЬНОМУ АНАЛИЗУ СИГНАЛОВ

Аннотация. В работе был проведен сравнительный анализ двух широкораспространенных технологии параллельного вычисления применительно к спектральному анализу. Экспериментально были оценены вычислительные производительности обеих технологий.

Ключевые слова: спектральный анализ, преобразование Фурье, параллельное вычисление, метод наименьших квадратов, квазистационарные участки.

Различают два варианта представления сигнала: временной и частотный. Для перехода из временной области в частотную используют метод спектрального анализа. Спектральный анализ незаменимый инструмент в обработке сигналов, так как частотные составляющие сигнала дают более полную информацию о сигнале. В случае когда сигнал имеет только одну устойчивую частоту, не имеет большого смысла применять этот метод. Однако, в реальных экспериментах, в реальной жизни сигналы такого рода встречаются крайне редко. Зачастую реальные сигналы имеют сложную структуру, т.е. имеют несколько частотных составляющих (например, хаос, квазипериодические сигналы и т.д.). Преобразование Фурье одно из основных методов разложения сигнала на частотные составляющие.

Анализ Фурье — это метод математического описания сигналов, позволяющий переходить от их рассмотрения во временной области в частотную область и обратно. Для этого используется ряд Фурье, который представляет собой разложение несинусоидальной периодической функции — сигнала на синусоидальные компоненты, т. е. заменяет временное представление сигнала частотным. Вообще говоря, любая периодическая функция может быть представлена в виде суммы

членов ряда Фурье следующим образом [1]:

$$f(t) = A_0 + A_1 \cos \alpha + A_2 \cos 2\alpha + A_3 \cos 3\alpha + \dots + A_n \cos n\alpha + A_0 + B_1 \cos \beta + B_2 \cos 2\beta + B_3 \cos 3\beta + \dots + B_n \cos n\beta + \dots \quad (1)$$

Фурье преобразование требует довольно большой объем вычислений, в связи с этим, при обработке сигналов, содержащих огромное количество значений, необходимо повысить скорость вычислений данных, так как в ходе исследований бывают моменты, когда время выполнения крайне важны.

Как правило, для увеличения скорости выполнения операции на современных компьютерах используются методы параллельного вычисления. На данный момент широко распространены два вида технологии параллельного вычисления: OpenMP и Nvidia CUDA. Технология OpenMP работает на центральном процессоре, а технология Nvidia CUDA использует графический процессор.

Технология OpenMP (Open Multiprocessing) — это прикладной интерфейс программирования (API), предназначенный для параллельного программирования с применением разделяемой памяти. Поддерживаются языки программирования C, C++ и Fortran, операционные системы Solaris, AIX, HP-UX, Linux, Mac OS X, Windows. Разработка OpenMP ведётся с участием крупных ИТ-

компаний, таких как AMD, Intel, IBM, Cray и др.[2]

CUDA (Compute Unified Device Architecture) – это программно-аппаратная платформа для параллельных вычислений, задействующая ресурсы графического процессора nVidia для неграфических вычислений.[3]

Каждый из технологий имеют сильные и слабые стороны:

При использовании OpenMP модификации свелись к добавлению нескольких директив компилятору, все необходимые библиотеки уже распространяются с компилятором GCC;

При использовании CUDA обязательно наличие графического процессора от nVidia не младше серии 8000, а также требуется изучение расширений языка и установки средств разработки;[4]

Если говорить именно о спектральном анализе (в частности о преобразовании Фурье), то в технологии CUDA имеется встроенная библиотека CUFFT, которая позволяет получать быстрое преобразование Фурье.

В данной работе был проведен эксперимент, с целью выяснения уровня производительности вышеупомянутых технологий. Суть исследования заключалась в следующем: заданный сигнал разбивался на множество квазистационарных участков (сегменты). Это делается потому, что проводить спектральный анализ всего сигнала в большинстве случаев не имеет смысла. Необходимые характеристики для описания сигнала находятся на определенных сегментах, с сравнительно устойчивыми параметрами. Затем измерялись время вычисления Фурье преобразования на центральном процессоре и графическом процессоре, который поддерживает технологию Nvidia CUDA.

Эксперимент

Эксперимент выполняется на платформе с характеристиками:

Процессор: Intel(R) Core(TM) i5-3317U, с тактовой частотой 1.70GHz, количество ядер/потоков - 2/4;

Графический процессор: NVIDIA GeForce GT 740M, объем памяти/разрядность шины памяти - 1Гб/128 бит;

Оперативная память: DDR3, 6144 Mb;

На основе измерений были получены следующие результаты:

На графике, который находится в приложении, изображены зависимости времени вычисления от количества сегментов. График, на котором точки изображены кружками, производительность центрального процессора, а «звездочками» графического процессора. Из рисунка 1 можно заметить, что функции обеих вычисления являются линейными. Чтобы убедиться в этом и найти параметры, соответствующие функциям использовался метод наименьших квадратов. С помощью нижеследующих формул были найдены параметры функции и были оценены ошибки, т.е. отклонения от действительных функций $y=kx$ и $y=a+bx$.

Рассмотрим зависимость $y = kx$ (прямая, проходящая через начало координат). Составим величину φ – сумму квадратов отклонений наших точек от прямой

$$\varphi = \sum_{i=1}^n (y_i - kx_i)^2 \quad (2)$$

Величина φ всегда положительна и оказывается тем меньше, чем ближе к прямой лежат наши точки. Метод наименьших квадратов утверждает, что для k следует выбирать такое значение, при котором φ имеет минимум

$$\frac{d\varphi}{dk} = -2 \sum x_i (y_i - kx_i) = 0 \quad (3)$$

или

$$k = \frac{\sum x_i y_i}{\sum x_i^2} \quad (4)$$

Вычисление показывает, что среднеквадратичная ошибка определения величины k равна при этом

$$S_k = \sqrt{\frac{1}{(n-1)} \cdot \left(\frac{\sum (y_i - kx_i)^2}{\sum x_i^2} \right)} \quad (5)$$

где n – число измерений.

Рассмотрим теперь несколько более трудный случай, когда точки должны удовлетворить формуле $y = a + bx$ (прямая, не проходящая через начало координат).

$$b = \frac{\sum[(x_i - \bar{x})y_i]}{\sum(x_i - \bar{x})^2} \quad (6)$$

$$a = \bar{y} - b\bar{x} \quad (7)$$

Среднеквадратичные ошибки определения a и b равны

$$S_b = \sqrt{\frac{\sum(y_i - bx_i - a)^2}{(n-2)\sum(x_i - \bar{x})^2}} \quad (8)$$

$$S_a = \sqrt{\left(\frac{\sum(y_i - bx_i - a)^2}{n-2}\right)\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum(x_i - \bar{x})^2}\right)} \quad (9)$$

В результате исследования функции методом наименьших квадратов максимальное значение ошибки не превышал 4.2%. Этот факт дает нам возможность с уверенностью сказать, что функции являются линейно-зависимыми. Надо отметить то, что функция измерения на GPU имеет вид $y = a + bx$ потому что при вычислении преобразования Фурье в библиотеке CUFFT создается план преобразования текущего сигнала и на это всегда расходуется определенное количество времени.

В результате исследования были получены следующие данные: при 500 сегментах центральный процессор выполнил задачу за 0,034228 секунд, а графический за 0,17 секунд. При максимальном количестве сегментов (104000) CPU и GPU потратили на вычисления 7,071002 и 0,3 секунд соответственно. Из этого следует, что в поставленной задаче данного исследования

графический процессор, поддерживающий технологию Nvidia CUDA, справился с задачей примерно в 23 раза быстрее.

Вывод.

Данный анализ показывает, что при относительно не большом количестве сегментов в сигнале, с точки зрения производительности целесообразнее использовать вычисление на центральном процессоре. Но с ростом количества сегментов более выигрышным оказывается использование технологии Nvidia CUDA.

Необходимо отметить тот факт, что вычисления на CPU производились на одном ядре. Но при очень больших объемах данных центральный процессор физический не может конкурировать с графическим процессором в плане вычислительной производительности (если рассматривать одинаковый ценовой сегмент).

Литература

1. Томаси У. Электронные системы связи // Москва: Техносфера, 2007, 1360 с.;
2. OpenMP Application Program Interface // Version 3.1 July 2011. – www.openmp.org/mp-documents/OpenMP3.1.pdf
3. What is CUDA. – <http://developer.nvidia.com/what-cuda>
4. Ханкин К.М. Сравнение эффективности технологий OPENMP, NVIDIA CUDA и STARPU на примере задачи умножения матриц. Вестник ЮУрГУ // Серия: Компьютерные технологии, управление, радиоэлектроника, № 1 / т. 13 / 2013

Принято в печать 20.10. 14

Приложение

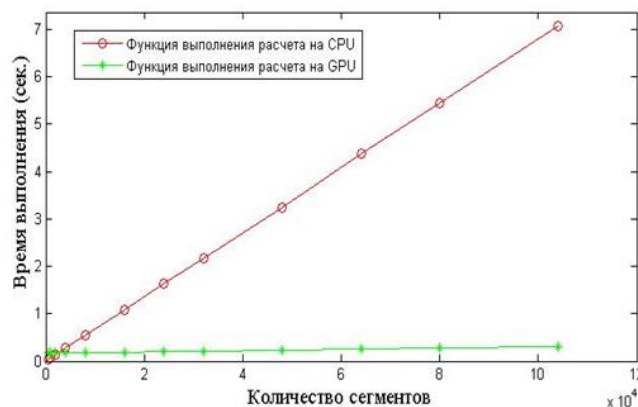


Рисунок 1 - Зависимость времени вычисления от количества сегментов

Б.Ж.Медетов, А.Е.Ыскак, К.А.Ниязалиев, А.С.Койшигарин
Казахский Национальный университет им. Аль-Фараби, г. Алматы, Казахстан

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНОГО ВЫЧИСЛЕНИЯ ПРИМЕНИТЕЛЬНО К СПЕКТРАЛЬНОМУ АНАЛИЗУ СИГНАЛОВ

Аннотация. В работе был проведен сравнительный анализ двух широкораспространенных технологии параллельного вычисления применительно к спектральному анализу. Экспериментально были оценены вычислительные производительности обеих технологий.

Ключевые слова: спектральный анализ, преобразование Фурье, параллельное вычисление, метод наименьших квадратов, квазистационарные участки.

B.Zh.Medetov, A.E.Yskak, K.A.Niyazaliev, A.S.Koyshigarin
Kazakh National University. Al-Farabi Kazakh National University, Almaty, Kazakhstan

COMPARATIVE ANALYSIS OF PARALLEL COMPUTATION WITH APPLIED TO THE SPECTRAL ANALYSIS OF SIGNALS

Abstract. In this paper was compared two common parallel computing technologies applied to spectral analysis and experimentally was evaluated the performance of both computing technology.

Б.Ж.Медетов, А.Е.Ыскак, К.А.Ниязалиев, А.С.Койшигарин
Әл-Фараби атындағы Қазақ Ұлттық университеті, Алматы қаласы, Қазақстан.

СИГНАЛДАРҒА СПЕКТРАЛДЫ ТАЛДАУДЫ ҚОЛДАНУҒА ҚАТЫСТЫ ПАРАЛЛЕЛДІ ЕСЕПТЕУ ТЕХНОЛОГИЯЛАРЫНЫҢ САЛЫСТЫРМАЛЫ АНАЛИЗІ

Андатпа. Берілген жұмыста спектральды талдауды қолдануға қатысты кеңтараған параллелді есептеу технологияларының салыстырмалы анализі жасалды. Тәжірибелік түрде екі технологияның да есептеу өнімділігі бағаланды.