

Для разработки вычислительных процедур был использован программный комплекс VisualStudio .NET 2003. В качестве функциональной базы для организации и управления процессом обмена данными в сети, запуска локальных вычислительных процессов подчиненных компьютеров, а также контроля состояния и анализа параметров сети была использована библиотека MPI 2.0.

Для оценки эффективности разработанного комплекса программ была проведена серия экспериментов, состоящая из следующих этапов: сравнение рассчитанных теоретически параметров алгоритмов, параметров сети и требуемых вычислительных ресурсов с результатами, полученными экспериментальным путем; анализ полученной топологии сети и распределения вычислительных процессов для двух уровней вейвлет-разложения, анализ параметров полученной системы для трех и более уровней вейвлет-декомпозиции, а также анализ действий хост-компьютера и изменение параметров вычислительного кластера при выходе из строя одной из машин.

Таким образом, в ходе работы были получены практические результаты, доказывающие эффективность разработанной системы динамического распределения вычислительных ресурсов локальных сетей в задаче цифровой обработки сигналов.

Были получены достоверные данные, позволяющие сделать вывод, что системы динамического распределения вычислительного процесса на параллельных структурах является доступным средством повышения эффективности вычислительных процедур любого характера.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Воеводин В. В., Воеводин Вл. В.* Параллельные вычисления. – СПб.: БХВ – Петербург, 2002.
2. *Антонов А.С.* Параллельное программирование с использованием технологии MPI: учебное пособие. – М.: Изд-во МГУ, 2004.
3. *Шпаковский Г. И., Серикова Н. В.* Программирование для многопроцессорных систем в стандарте MPI. – Минск.: БГУ, 2002.
4. MPI: A Message-Passing Interface Standard (Version 1.1) (<http://parallel.ru/docs/Parallel/mpi1.1/mpi-report.html>).
5. MPI-2: Расширение стандарта на интерфейс передачи сообщений. МИНСК, 2001.
6. *Камерон Хьюз, Трейси Хьюз.* Параллельное и распределенное программирование и использованием C++. – М.: СПб, Киев, 2004.
7. *Новиков Л.В.* Основы вейвлет-анализа сигналов. – СПб: ИАиП РАН, 1999.

Д.А. Беспалов, В.Ф. Гузик

ЕСТЕСТВЕННЫЙ ПАРАЛЛЕЛИЗМ БЫСТРЫХ АЛГОРИТМОВ ВЕЙВЛЕТ-АНАЛИЗА

На современном этапе развития методов цифровой обработки сигналов одним из самых перспективных направлений являются методы вейвлет-анализа [1]. Однако к некоторому ограничению их повсеместного практического применения приводит невысокая скорость выполнения отдельных алгоритмов. В то же самое время одним из самых эффективных методов оптимизации вычислительного процесса является параллельная реализация программ на основе процессорных кластеров, многомашинных комплексов и локальных вычислительных сетей [7]. Соединение этих двух направлений (вейвлет-анализ и параллельные вычисления)

позволило бы перейти на новый качественный уровень обработки сигналов в самых разных областях науки и техники.

Данная работа посвящена решению именно этой проблемы, то есть теоретическому обоснованию необходимости, экспериментальному доказательству эффективности и целесообразности создания быстрых параллельных алгоритмов вейвлет-анализа. Так как большинство алгоритмов цифровой обработки сигналов применяется в специализированных вычислительных устройствах, было рассмотрено именно это приложение. Однако все теоретические выводы и большинство практических результатов также верно для отдельных ПЭВМ и многомашинных комплексов.

Подробный анализ основных алгоритмов вейвлет-анализа позволил сделать вывод о возможности и целесообразности параллельной реализации алгоритмов вейвлет-разложения и восстановления сигналов по полной или частной лифтинговой схеме [2] (разложения/восстановления каскадами фильтров) с использованием двух- или четырехпроцессорной схемы. Однако распараллеливание на уровне простого распределения операций по узлам вычислительной системы в большинстве случаев дает лишь небольшое повышение производительности. Более эффективным является использование внутреннего параллелизма алгоритмов и адаптация вычислительного процесса к особенностям архитектуры аппаратных средств поддержки вычислений. В соответствии с этим был проведен анализ алгоритмов с точки зрения внутреннего параллелизма отдельных операций и возможности дополнительной оптимизации уже на этапе их реализации на аппаратном уровне. Был выделен ряд операций, характерных для рассмотренных алгоритмов вейвлет-разложения и восстановления, а также для их производных алгоритмов сжатия сегментации и шумоподавления. Основу их составляют операции дискретной свертки и операции с симметричными потоками данных. Были подсчитаны численно и выражены аналитически вычислительные затраты на каждом уровне представления сигналов.

Самым оптимальным в настоящее время аппаратным решением является использование процессоров со специфической архитектурой, ориентированных на выполнение операций с параллельными потоками данных и выполнение циклических сверток. Такими аппаратными средствами являются высокопроизводительные процессоры цифровой обработки сигналов (ПЦОС) с супергарвардской архитектурой семейства SHARC ADSP 21000 [4]. Состав данных аппаратных средств ПЦОС позволил реализовать быструю схему вычислений, основанную на операции свертки, то есть на комплексах циклически повторяющихся операций умножения/накопления в сочетании с чтением данных и записью результатов по параллельным каналам.

Данная совокупность операций идеально подходит для аппаратной реализации алгоритмов вейвлет-анализа, тем более что выполнение каждой операции умножения с накоплением производится за один такт. Благодаря внутреннему счетчику цикла и стеку цикла процессор выполняет программу цикла с нулевыми потерями. Не требуется команд перехода для организации циклов или декремента и проверки счетчика. А наличие независимых шин памяти программ и памяти данных позволяет ядру процессора одновременно обращаться к командам и данным в общих блоках памяти. Коэффициенты фильтров декомпозиции/реконструкции в таком случае могут храниться в том же самом блоке памяти, который содержит команды, в то время как выборки данных хранятся в другом блоке. Такая организация позволяет достичь минимальных затрат на вычисление одной точки результирующего сигнала.

В ходе экспериментов все команды удалось распределить на L тактов, где L – длина фильтра, в то время как число бинарных операций в формуле быстрого вейвлет-преобразования на классической архитектуре подразумевает $L(L-1)$ команд на отсчет, помноженное на количество тактов выполнения команд, не считая операций обмена с памятью.

Дополнительным преимуществом использования ПЦОС такого класса является возможность обмена с общей памятью многопроцессорной системы и с другими процессами в фоновом режиме, то есть параллельно вычислениям. Такая особенность архитектуры позволила сократить до минимума потери времени на межпроцессорный обмен.

Для практической реализации и проведения экспериментов был использован многопроцессорный кластер ADP160QPCI v 1.1, выполненный на основе четырех процессоров SHARC ADSP-21160 фирмы Analog Devices. В ходе экспериментов синтезирован ряд конфигураций, реализующих разные варианты выполнения алгоритмов вейвлет-анализа: двухпроцессорный вариант, реализующий макрораспределение операций по ветвям полного дерева алгоритма Малла, четырехпроцессорный вариант, реализующий микрораспределение вычислений лифтинговой схемы разложения неполного алгоритма Малла по двум процессорам первого каскада с последующей параллельной обработкой на оставшихся двух процессорах.

Экспериментальная часть работы позволила сравнить полученные теоретические значения вычислительных преимуществ параллельной реализации алгоритмов на разных конфигурациях с практическими результатами. Средние значения вычислительных преимуществ (затрат на выполнение полного алгоритма) составили 48.8% – 50.2% для двухпроцессорного варианта и 25.1% – 25.4% для четырехпроцессорной схемы, где за 100% было принято значение временных затрат при реализации на базе одного процессора.

Практическая реализация быстрых параллельных алгоритмов вейвлет-анализа проводилась с использованием программного пакета DEASY TOOLS, предназначенного для разработки, отладки и выполнения программ цифровой обработки сигналов на процессорных модулях фирмы InSys Corporation, построенных с использованием процессоров ADSP-2116x фирмы Analog Devices, а также с использованием программного комплекса Visual DSP++™ 3.0 – системы разработки программного обеспечения для приложений ЦОС [5]. Все оптимизируемые алгоритмы до аппаратной реализации были протестированы в составе программного комплекса цифровой обработки сигналов методами вейвлет-анализа, разработанного для ПЭВМ при помощи программного пакета VisualStudio .NET 2003 [6].

Таким образом, в результате проведенной работы была доказана целесообразность разработки и существенное повышение производительности параллельной реализации быстрых алгоритмов вейвлет-анализа, получены подтвержденные серийой экспериментов теоретические результаты, показывающие высокую эффективность реализации комплекса алгоритмов на базе многопроцессорных кластеров ADSP-2116x фирмы Analog Devices.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Добеши И. Десять лекций по вейвлетам. - М.: РХД, 2001.
2. Воробьев В., Грибунин В. Теория и практика вейвлет-преобразования. – НИИ В.Г. ВУС, 1999.
3. Новиков Л.В. Основы вейвлет-анализа сигналов. – СПб.: ИАиП РАН, 1999.
4. ADSP 2106x SHARC User's manual. – Analog Devices Inc., 1996.
5. Engineer To Engineer Note EE-148 Introduction to SHARC® Multiprocessor Systems Using VisualDSP++™ Contributed by Maikel Kokaly-Bannourah April 01, 2003.

6. *Камерон Хьюз, Трейси Хьюз.* Параллельное и распределенное программирование и использованием C++. –М, СПб, Киев: 2004.
7. *Воеводин В. В., Воеводин Вл. В.* Параллельные вычисления. – СПб.: БХВ – Петербург, 2002.

В.Е.Золотовский, П.В.Савельев

ФОРМАТЫ ДАННЫХ ДЛЯ СИСТЕМЫ СИМВОЛЬНЫХ ВЫЧИСЛЕНИЙ

Базовым слоем любого вычислительного комплекса являются форматы представляемых данных. Для символьной математики вопрос об организации данных, пожалуй, наиболее важен, так как используемые форматы закладывают основу производительности системы, полученной по завершении разработки. Далее будут рассмотрены формы представления чисел произвольной размерности, а также символьных выражений, которые содержат буквенно-числовые обозначения переменных величин.

Обычно длинные числа представляются в виде коэффициентов при различных степенях основания заданной системы счисления. Более формально это показывает

$$X = x_0 + x_1B + x_2B^2 + \dots + x_nB^n,$$

если X является целым числом. В случае, когда X – дробное, формула преобразуется к виду

$$X = x_0 + \frac{x_1}{B} + \frac{x_2}{B^2} + \dots + \frac{x_n}{B^n};$$

$$X = Y \cdot B^e.$$

$$X = x,$$

где B – основание системы счисления, в котором записывается число, x_i – коэффициенты.

Знак числа обычно либо хранится отдельно, либо все $x_i < 0$.

Основание B зависит от максимального размера базового типа данных на вычислительной машине и выбирается исходя из следующих соображений:

- основание B подходит под один из базовых типов данных;
- B должно быть как можно больше, чтобы уменьшить размер представления длинного числа и увеличить скорость выполнения операций с ними, но достаточно малого размера, чтобы все операции с коэффициентами использовали базовый тип данных;
- для удобства можно выбрать степень B равную 10. Это упростит отладку модуля автоматического выполнения арифметических операций с длинными числами, а также позволит воспользоваться очень примитивными средствами вывода информации и отображения операндов и результатов;
- при использовании B равного степени двойки появляется возможность реализовывать быстрые операции на низком уровне. Однако сложной проблемой является переход от основания $B=2m$ к основанию $B=10n$, особенно для больших чисел;
- некоторые задачи, например определение простоты числа, могут использовать реализацию при $B=2m$, так как в качестве вывода используются логические значения ДА или НЕТ.