

Системный подход к созданию специального программного обеспечения для обработки больших массивов ЦРГ, получаемых космическим РСА

М.А. Семенов¹, С.Э. Зайцев², О.А. Карпов³, Т.А. Лепёхина¹, В.И. Николаев¹

¹АО «Концерн «Вега»: 121170, Москва, Кутузовский проспект, 34; e-mail: tatonika@inbox.ru

²АО «ВПК «НПО Машиностроения», г.Реутов, МО, ул. Гагарина, 33, e-mail: zaisergeri@mail.ru

³ЗАО «АЭРОКОН», г. Жуковский, МО, ул. Гагарина, 1, e-mail: karp.smtp@mail.ru

Разработано программно-алгоритмическое обеспечение для реализации различных вычислений над матрицами цифровых радиоголограмм, позволяющее оптимизировать ряд вычислительных процессов. Программное обеспечение выполняет произвольный набор стандартных операций над матрицами, с возможностью дальнейшего расширения набора доступных операций. Создана методика оценки времени работы алгоритма в зависимости от выбранного набора операций.

Software for different computations execution upon digital radioholograms has been developed to optimize most calculation processes. This software performs desired group of elementary operations upon matrixes with ability of further operation set extension. Algorithm processing time estimation technique based on used operation sequence is created.

Цифровые радиоголограммы (ЦРГ), поставляемые космическими радиолокаторами с синтезированием апертуры антенны (РСА), представляют собой комплексные матрицы больших размеров. Для получения радиолокационного изображения требуется выполнить значительные объемы вычислений, что влечет за собой ощутимые временные затраты [1]. Разработка специального программного обеспечения (ПО) с позиций системного подхода предполагает анализ имеющихся аппаратных средств для построения оптимального вычислительного процесса с учетом операций, требующих значительных временных затрат, а также систематизация вычислительного процесса путем динамического описания последовательности выполнения операций над исходной матрицей, и оптимизации всего механизма обработки данных операций. При таком подходе любое изменение алгоритма не требует разработки нового ПО или отдельного процесса оптимизации, основанного на практической оценке времени выполнения каждого варианта, поскольку оценка узких мест архитектуры позволяет предсказать наиболее оптимальное решение до практической реализации.

Проанализируем основные операции, используемые при обработке ЦРГ для оптимизации временных затрат на каждую из них. Для этого рассмотрим вычислительную задачу обработки ЦРГ с точки зрения скорости выполнения. К числу наиболее затратных операций можно отнести выполнение преобразования Фурье по быстрым алгоритмам (БПФ) и транспонирование матрицы. Несмотря на то, что БПФ состоит полностью из простейших с точки зрения вычислительной техники операций умножения и сложения, а транспонирование вовсе не требует математических операций, затрачиваемое на их выполнение время существенно и формируется за счет скорости обмена данными между ОЗУ и процессором, а также накладными расходами контроллера ОЗУ при непоследовательном доступе к данным. Дело в том, что при нынешнем состоянии вычислительных средств становится неактуальным учет вычислительной сложности в количестве требуемых вычислительных операций – рядовой процессор общего назначения (Intel Core i5 2500K) способен выполнять в секунду по 48 млрд. операций умножения и еще столько же операций сложения, в то

время как непосредственный доступ к ОЗУ позволяет обеспечить его данными только для 2,5 млрд. операций.

Таким образом, возникает необходимость учета не только вычислительных операций, но также и процессов информационного обмена данными между физическими структурами памяти, различающимися по объёму и скорости доступа: (регистровая память, кэш-память различных уровней, ОЗУ, Solid-State Disk и Hard Disk Drive), с последующей оптимизацией данного обмена, в том числе переносом данных из одной структуры в другую.

Для решения данных задач требуется организовать процесс вычислений таким образом, чтобы минимизировать непрофильные временные затраты, обусловленные особенностями вычислительной архитектуры, обеспечив выполнение следующих требований к вычислительным алгоритмам:

- Блочный доступ к памяти. Использование ОЗУ блоками максимального возможного размера, со стремлением к размеру блока в 4 Кбайт, соответствующего странице памяти. Данное требование служит для снижения накладных расходов на доступ к памяти, вызванных обращением к дескрипторам страниц памяти, и латентность доступа к строкам ОЗУ.

- Последовательный доступ к памяти. Запись в ОЗУ особыми командами, работающими в обход кэша, во избежание считывания излишних данных из ОЗУ в кэш. Считывание ОЗУ в пределах блока в порядке монотонного возрастания адресов, в целях корректной работы предвыборки данных из ОЗУ менеджером памяти [2].

- Максимальное использование кэша разных уровней. Это позволяет увеличивать количество вычислительных операций, приходящихся на одно обращение к физическому ОЗУ.

- Использование SIMD-инструкций (Single Instruction – Multiple Data) для выполнения вычислений, в целях реализации вычислительного потенциала процессора.

- Независимое распределение задач и кэш-памяти 3-го уровня между вычислительными потоками, с целью минимизации нагрузки на внутреннюю шину обмена данными между ядрами процессора [3].

При этом требуется сохранить возможность перенастройки вычислительного процесса с учетом разных параметров ЦРГ и гибкость в добавлении дополнительных вычислительных операций без необходимости внедрения в основной алгоритм организации вычислений.

В рамках данной работы был реализован алгоритм, обеспечивающий хранение основного массива ЦРГ в ОЗУ группами по 4 или 8 строк (в зависимости от точности представления чисел), и позволяющий последовательно выполнять над данным массивом группы операций. Каждая группа предполагает копирование столбцов (с транспонированием при копировании) или строк в отдельную область памяти, где реализуется распараллеленная обработка в несколько потоков, по числу ядер процессора. Обработка допускает предопределенные операции, реализованные на ассемблере, которые могут выполнять над группой строк преобразование Фурье или простейшие математические операции, а также преобразовывать группу в отдельные строки и наоборот, для возможности проведения над ними не предопределенных ранее операций, реализованных на языке высокого уровня.

Все реализованные предопределенные операции обеспечивают требования по обращениям к памяти и использованию кэша. Скорость выполнения данных операций, определенная опытным путем, близка к верхней оценке времени выполнения, вычисленной по количеству фактических обращений к ОЗУ в процессе вычисления: по одному дуплексному обращению к памяти на простейшие математические операции,

операции копирования / транспонирования / группирования / разгруппирования; три обращения – на операцию БПФ.

Для обработки массива по строкам или столбцам используются 2 буфера по 64 строки максимального размера, что позволяет оптимизировать процедуру транспонирования и обеспечить пространство для изменения длины обрабатываемых строк (рис. 1).

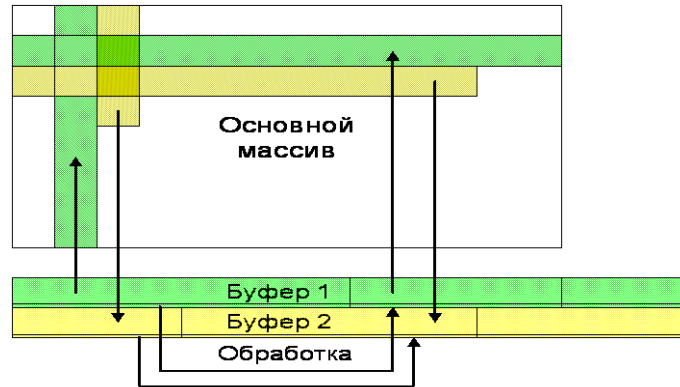


Рис. 1. Обмен между основным массивом и буферами

Использование двух промежуточных буферов обработки позволяет разделить во времени процедуры обработки и копирования между основным массивом и буфером, обеспечивая многопоточную обработку данных без задержек на ожидание копирования. Пример распределения данных операций во времени с использованием четырех потоков обработки, а также потоков чтения и записи, представлен на рис. 2.

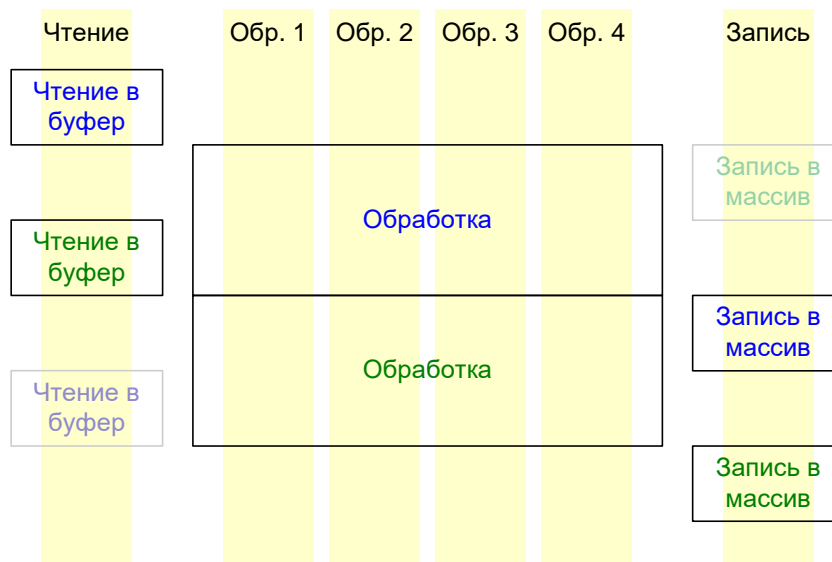


Рис. 2. Распределение операций по потокам

Исследование узких мест архитектуры и оптимизация обработки позволили, кроме увеличения скорости выполнения вычислений, также определить, чем именно ограничивается эта скорость, и, следовательно, проводить предварительную оценку скорости вычислений по объёму требуемых операций.

Для проверки на практике оценки скорости работы алгоритма в качестве примера была взята обработка голограммы путем свертки с опорной функцией по дальности и гармонического анализа по азимуту. Оценка проводилась аналитическим путем, через

расчет необходимого количества дуплексных обращений к памяти, которые даже после проведенной оптимизации остаются наиболее узким местом всего процесса вычислений. Объем считался для каждой группы операций, при этом учитывались операции чтения и записи в соответствии с объемом модифицируемой области голограммы, простейшие операции в соответствии со своей длиной и количеством строк, а для операций БПФ учёт вёлся в тройном размере (два прохода и двоично-инверсная перестановка) [4].

При оценке использовались голограммы размером 2048×140800 отсчетов (объем голограммы №1 – 2,15 Гбайт, общий объем учтённых обращений к памяти 61 Гбайт) и 19860×15064 отсчетов (объем голограммы №2 – 2,23 Гбайт, общий объем учтённых обращений к памяти – 184 Гбайт, за счет вычисления преобразования Фурье длины, не равной степени 2, через свертку большего размера).

Для полученного объема информационного обмена между ОЗУ и процессором были вычислены верхняя и нижняя границы оценки, из расчета 8,7 Гбайт/с для верхней (измеренная на практике скорость прямого последовательного копирования, при фактическом потолке в 10 Гбайт/с, определенном по тактовой частоте без учета латентности) и 6,6 Гбайт/с для нижней (утроенная измеренная на практике скорость выполнения БПФ для строк длины 2 млн. элементов).

Результаты экспериментального определения времени вычислений представлены на рис. 3. Данные результаты показали высокий уровень соответствия оценки времени выполнения фактическим значениям. Кроме того, можно отметить, что использование третьего ядра процессора для обработки дает лишь небольшой прирост скорости, а четвертого – и вовсе несущественный, что говорит о недостаточной загруженности вычислительных возможностей самого процессора.

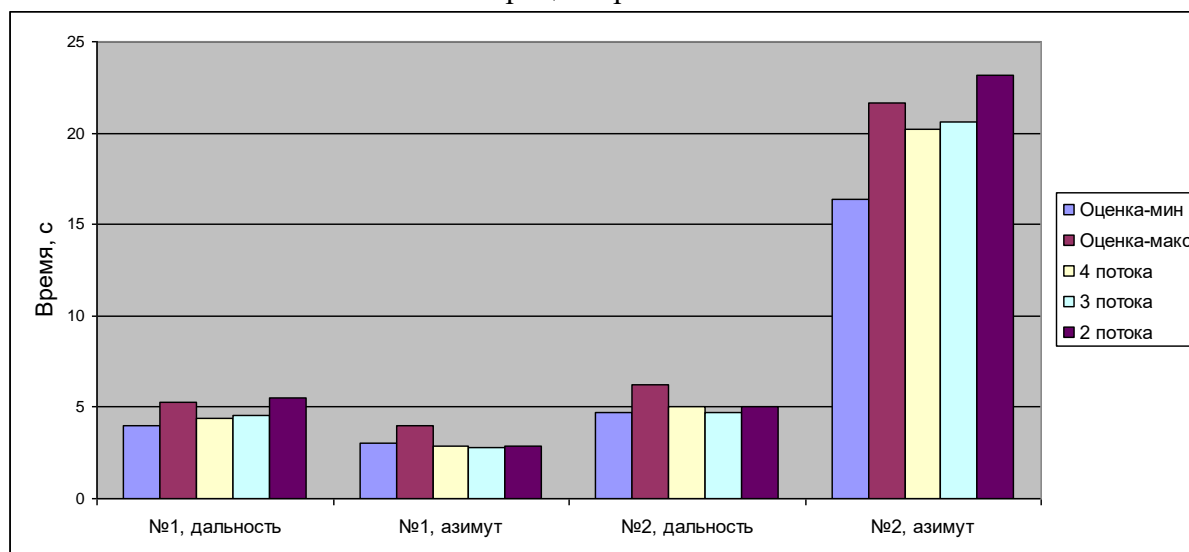


Рис. 3. Время выполнения алгоритма

Это может иметь значение при добавлении операций, не поддерживающих выполнение в режиме Single Instruction – Multiple Data (SIMD) – тогда дополнительная нагрузка вычислительной мощности ядер окажет не столь существенное влияние на скорость обработки в целом, поскольку будет скомпенсирована операциями, критичными к пропускной способности памяти.

Предложенный подход обеспечивает возможность обработки массивов ЦРГ требуемого объема в течение определенного промежутка времени благодаря снижению нагрузки на наиболее проблемных участках вычислительной архитектуры

Таким образом, еще на этапе разработки ПО для реализации определенного алгоритма обработки ЦРГ можно количественно оценить временные затраты и выработать определенные требования к специально разрабатываемым вычислительным машинам, в том числе машинам бортовой обработки.

Литература

1. Радиолокационные станции воздушной разведки, дешифрование радиолокационных изображений / Л.А.Школьный, Е.Ф.Толстов, А.Н.Детков и др. Под ред. Л.А.Школьного. – М.: Изд. ВВИА им. проф. Н.Е. Жуковского, 2008. – 530 с.
2. Intel® 64 and IA-32 Architectures. Optimization Reference Manual. URL: <http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html> (дата обращения: 15.01.2017).
3. Романченко В. Микроархитектура Intel Sandy Bridge, часть II. URL: http://www.3dnews.ru/guide/intel_sandy_bridge_2/print (дата обращения: 12.01.2017).
4. Лепёхина Т.А., Николаев В.И., Семёнов М.А., Толстов Е.Ф. Проблемы обработки больших массивов цифровых радиоголограмм космических РСА высокого разрешения // Информационно-измерительные и управляющие системы. – 2013. – Вып. №1. – С.37-41.