3. Макконнелл Дж. Основы современных алгоритмов. 2-е доп. изд. М:, Техносфера, 2004. 368 с.

4. Елкина В.Н. и Загоруйко Н.Г. Количественные критерии качества таксономии и их использование в процессе принятия решений. – "Вычислительные системы", 1969, вып. 36, с.29 – 46.

5. Жиляков Е.Г. и Барсук А.А. О компьютерной реализации автоматической вариационной классификации объектов на спутниковых фотографиях земной поверхности. – "Вопросы радиоэлектроники", сер. ЭВТ, 2010, вып. 1, с. 166-171.

6. Яне Б. Цифровая обработка изображений. М:, Техносфера, 2007. 584 с.

Статья поступила 12 10 2010

Д.т.н., проф. Е.Г. Жиляков, А.Ю. Лихошерстный (БелГУ)

E.G. Zhilyakov , A.J. Lihosherstnyj

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА АЭРОКОСМИЧЕСКИХ ИЗОБРАЖЕНИЙ НА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

PARALLEL PROCESSING OF SPACE IMAGES FOR HIGH-PERFOMANCE COMPUTING SYSTEMS

Проведено распараллеливание алгоритма фильтрации изображений на основе частотных представлений с помощью технологии MPI на суперкомпьютере IBM Blue Gene/P

Key words image, parallel processing, filtering, acceleration

Развитие суперкомпьютерных технологий определяет прогресс в области обработки информации. Такие технологии являются базовыми критическими технологиями, поскольку сегодня они важнейшие в спектре тех, которыми владеет человечество. Именно на их основе решаются наиболее трудные и ресурсоемкие междисциплинарные задачи современной науки, техники, промышленности и бизнеса.

Одной из важнейших задач, решаемых при помощи суперкомпьютерных технологий, является исследование поверхности Земли, а именно: мониторинг влияния изменения климата, оценка загрязнения атмосферы, моделирование наводнений и оползней,

картографирование, определение структуры земного покрова и землепользования, а также поддержка мероприятий по обороне и безопасности (в настоящее время этими проблемами серьезно занимаются компании Intel и NVIDIA).

В качестве инструмента исследования поверхности Земли часто применяется космофотосъемка. При этом информация извлекается на основе анализа полученных со спутников космофотоизображений (изображений, полученных в результате съемки, выполненной специальной аппаратурой из космоса).

Поскольку анализ и обработка аэрокосмических изображений требуют использования всех ресурсов доступных аппаратно-программных средств решение этих задач на высокопроизводительных вычислительных комплексах является актуальным и включает в себя:

1. Получение данных дистанционного зондирования от центров приема спутниковой информации;
2. Распределенную и параллельную обработку аэрокосмической информации в локальных сетях суперкомпьютерных центров;
3. Реализацию параллельных алгоритмов обработки аэрокосмических изображений.

Одной из основных процедур обработки космофотоснимков является фильтрация. Под фильтрацией изображений понимают операцию, имеющую результатом изображение того же размера, полученное из исходного по некоторым правилам [2]. Эта процедура позволяет очистить изображение от ненужных или вредных, с точки зрения восприятия, компонент, что улучшает визуальное качество снимков.

Целью работы является программная реализация алгоритма фильтрации аэрокосмических изображений на основе частотных представлений [5] на высокопроизводительной вычислительной системе IBM Blue Gene/P с помощью технологии MPI и проведение вычислительных экспериментов с реальными космическими изображениями.

Изображение можно определить как двумерную функцию fik, где i=1,2,...M и k=1,2,...,N – координаты в пространстве (на плоскости), и значение f которой в любой точке, задаваемой парой координат (i,k), называется интенсивностью изображения в этой

точке [2].

Выражение для нахождения результата фильтрации Ф* изображения Ф в частотной области Ω выглядит следующим образом [3]:

$$\Phi^{*}_{\Omega} = A^{T}\Phi B \qquad (1)$$

где элементы матриц A=(a_{i1i2}) и B=(b_{k1k2}) вычисляются по формулам:

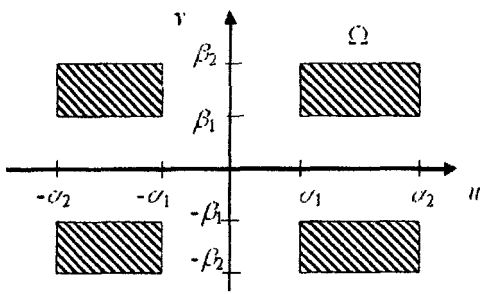
$$a_{i_1 i_2} = \begin{cases} \dfrac{Sin(\alpha_2(i_1 - i_2)) - Sin(\alpha_1(i_1 - i_2))}{\pi(i_1 - i_2)}, & i_1 \neq i_2, \\ \dfrac{\alpha_2 - \alpha_1}{\pi}, & i_1 = i_2, \end{cases}$$

$$b_{k_1 k_2} = \begin{cases} \dfrac{Sin(\beta_2(k_1 - k_2)) - Sin(\beta_1(k_1 - k_2))}{\pi(k_1 - k_2)}, & k_1 \neq k_2, \\ \dfrac{\beta_2 - \beta_1}{\pi}, & k_1 = k_2. \end{cases}$$

$$0 \le \alpha_1, \alpha_2, \beta_1, \beta_2 \le \pi$$

В качестве области Ωik рассматривается следующая центрально-симметричная область частотной плоскости [4]:

$$\Omega ik : \{\Omega ik(u,v) \mid (u \in [\alpha_1, \alpha_2], v \in [\beta_1, \beta_2]) \cup (u \in [\alpha_1, \alpha_2], v \in [-\beta_2, -\beta_1]) \cup (u \in [-\alpha_2, -\alpha_1], v \in [-\beta_2, -\beta_1]) \cup (u \in [-\alpha_2, -\alpha_1], v \in [\beta_1, \beta_2])\}$$



$$-\pi \le u, v \le \pi$$

Рис.1

Вычисление выражения (1) является очень трудоемким. Сложность алгоритма составляет O(n^3). В то же время, процедуры

перемножения матриц легко распараллелить. Для вычисления выражения (1) была выбрана наиболее распространенная технология программирования для параллельных компьютеров с распределенной памятью - MPI (Message Passing Interface). MPI имеет следующие особенности [6]:

- MPI — библиотека, а не язык. Она определяет имена, вызовы процедур и результаты их работы. Программы, которые пишутся на FORTRAN, C и C++ компилируются обычными компиляторами и связаны с MPI-библиотекой;
- MPI — описание, а не реализация. Все поставщики параллельных компьютерных систем предлагают реализации MPI для своих машин как бесплатные, они могут быть получены из Интернета. Правильная MPI-программа должна выполняться на всех реализациях без изменения;
- MPI соответствует модели многопроцессорной ЭВМ с передачей сообщений.

В модели передачи сообщений процессы, выполняющиеся параллельно, имеют раздельные адресные пространства. Связь происходит, когда часть адресного пространства одного процесса скопирована в адресное пространство другого процесса. Эта операция совместная и возможна только, когда первый процесс выполняет операцию передачи сообщения, а второй процесс — операцию его получения.

Принципиальным моментом при разработке параллельных алгоритмов является анализ эффективности использования параллелизма. Основной задачей является оценка максимально возможного ускорения процесса решения рассматриваемой задачи (анализ всех возможных способов выполнения вычислений).

Ускорение (speedup), получаемое при использовании параллельного алгоритма для p процессоров, по сравнению с последовательным вариантом выполнения вычислений, определяется величиной [1]:

$$S_p(n) = T_1(n) / T_p(n) \qquad (2)$$

где величина n используется для параметризации вычислительной сложности решаемой задачи и может пониматься, например, как количество входных данных задачи; T — время выполнения программы.

Множество операций, выполняемых в исследуемом алгоритме решения вычислительной задачи, и существующие между операциями информационные зависимости могут быть представлены в виде ациклического ориентированного графа:

$$G=(V,R) \qquad (3)$$

где V - множество вершин графа, представляющих выполняемые операции алгоритма, R — множество дуг графа. Вершины без входных дуг могут использоваться для задания операций ввода, а вершины без выходных дуг – для операций вывода.

Основной алгебраической операцией выражения (1) является матричное умножение. Код на C++ для такой операции выглядит следующим образом:

```
for (i=0; i<n; i++ )
    for (j=0; j<n; j++ ) {
        a[i][j] = 0;
          for (k=0; k<n; k++ )
              a[i][j] = a[i][j] + b[i][k]*c[k][j];
              }
```

Используя этот код, можно построить граф операции матричного умножения, который представлен на рис. 2:

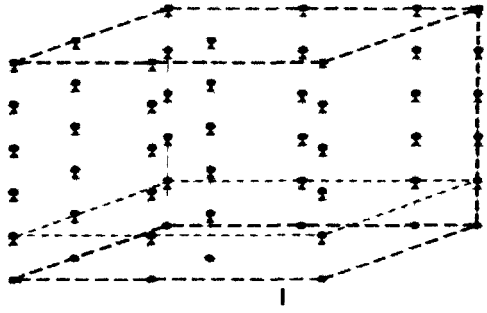


Рис. 2
Граф операции матричного умножения

Вершины графа, которые не соединены между собой дугой, называются информационно независимыми. Чем больше в графе информационно независимых вершин, тем более эффективно распараллеливается алгоритм (другими словами, можно распараллелить только те участки кода, в которых вершины графа информационно независимы).

Процедуру матричного перемножения можно представить как скалярное произведение векторов-строк одной матрицы на векторы-столбцы другой. Таким образом, для распараллеливания алгоритма достаточно написать кусок кода следующего вида:

```
for (k=0; k<n; k++ )
l_mult = l_mult + b[i][k]*c[k][j];
MPI_Reduce(&l_mult,&res,1,MPI_DOUBLE,MPI_SUM,0,MPI_COMM_WORLD),
MPI_Barrier(MPI_COMM_WORLD);
```

Здесь функция MPI_Reduce выполняет независимые глобальные операции суммирования над соответствующими элементами массивов l_mult. Результат получается в массиве res процесса под номером ноль. Работа процессов блокируется до тех пор, пока все оставшиеся процессы коммуникатора не выполнят эту процедуру. Все процессы должны вызвать MPI_Barrier, хотя реально исполненные вызовы различными процессами коммуникатора могут быть расположены в разных местах программы.

Вычислительные эксперименты обработки аэрокосмических изображений были проведены на суперкомпьютере IBM Blue Gene/P (рис. 3), принадлежащем МГУ им. М.В. Ломоносова. Каждая из стоек (их две) имеет следующие характеристики:

- 1024 4-ядерных вычислительных узлов;
- Производительность одного вычислит. узла –13.6 GF/s;
- Производительность одной стойки– 13.9 Tflops;
- Оперативная память одного узла – 2 GB;
- Суммарная оперативная память в стойке– 2 TB;
- Узлов ввода/вывода 8 — 64;
- Размеры - 1.22 x 0.96 x 1.96;
- Занимаемая площадь 1.17 $м^2$;
- Энергопотребление (1 стойка) - 40 kW (max).

Состав программного обеспечения на суперкомпьютере Blue Gene/P следующий:

- Операционная система Linux на узлах ввода\вывода;
- Библиотеки MPI (MPICH2) и OpenMP (2.5);
- Стандартное семейство компиляторов IBM XL: XLC/C++, XLF;
- Компиляторы GNU;
- Система управления заданиями LoadLeveler;

- Файловая система GPFS;
- Инженерная и научная библиотека подпрограмм (ESSL).

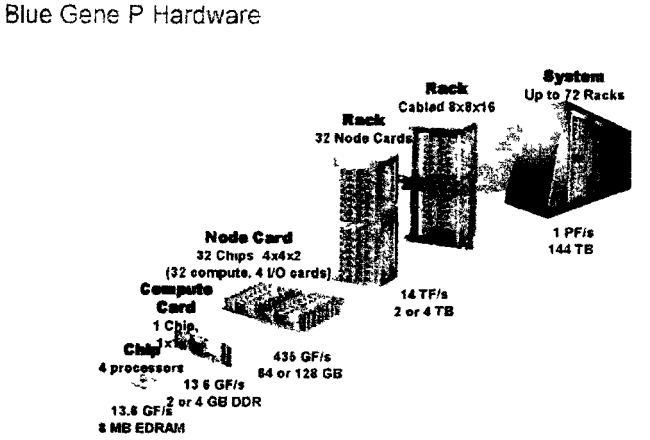


Рис. 3
Архитектура вычислительной системы IBM Blue Gene/P

В результате проведения вычислительных экспериментов была исследована зависимость ускорения (2) программы от числа процессоров для различных размеров изображений. Эта зависимость представлена в табл. 1. Здесь для каждых значений количества процессоров и размерности изображения представлено соответствующее значение ускорения.

Таблица 1

Зависимость ускорения программы
от числа процессоров для различных размеров изображений

	Количество процессоров		
Размерность изображения	32	64	128
2896x2896	31.1546	61.334	400.444
4096x4096	31.4553	62.5177	459.614
5792x5792	31.8656	62.8969	481.963
8192x8192	31.9387	63.715	494.081
11585x11585	31.9481	63.8413	499.991

Таким образом, величина ускорения больше зависит от количества процессоров, чем от размерности изображения. Особенность архитектуры суперкомпьютера IBM Blue Gene/P позволяет не нести потери в эффективности при увеличении количества вычислительных узлов.

Работа выполнена при поддержке ФЦП «Научные и научно-педагогические кадры для инновационной России» на 2009-2013 годы, гос. контракт № 14.740.11.0390.

Литература

1. Гергель В.П. Теория и практика параллельных вычислений. М., Бином, 2007. 424 с.

2. Гонсалес Р. и Вудс Р. Цифровая обработка изображений. М., Техносфера, 2006. 1072 с.

3. Жиляков Е.Г. и Чрноморец А.А. Оптимальная фильтрация изображений на основе частотных представлений. – "Вопросы радиоэлектроники", сер. ЭВТ, 2008, вып. 1, с. 18-132.

4. Жиляков, Е.Г. и Лихошерстный А.Ю. Алгоритмы обработки аэрокосмических изображений на основе частотных представлений. – "Вопросы радиоэлектроники", сер. ЭВТ, 2010, вып. 1, с.73-84.

5. Лихошерстный, А.Ю. Обработка космических изображений на основе частотных представлений. – "Научные ведомости Белгородского государственного университета", 2009, вып. 11\1, № 9 (64), с..123-131.

6. Шпаковский Г.И. и Серикова Н.В. Программирование для многопроцессорных систем в стандарте MPI. Минск, 2002. 323 с.

Статья поступила 12 10 2010