

В.В. Крохин, Н.О. Кузьменко

УСКОРЕНИЕ ВЫЧИСЛЕНИЙ С ПОМОЩЬЮ МАТЛАБ ПУТЕМ РАСПАРАЛЛЕЛИВАНИЯ

Аннотация. Реализация параллельных вычислений в настоящее время доступна как при наличии многоядерных центральных процессоров, так и графических процессоров. В статье анализируется возможность ускорения вычислений в МАТЛАБ за счет распараллеливания. Рассмотрено применение надстройки Parallel Computing Toolbox (PCT) фирмы MathWorks, а также применение Jacket – программного продукта фирмы AccelerEyes, который позволяет производить математические расчеты в среде МАТЛАБ с использованием ресурсов современных видеокарт от NVIDIA.

Ключевые слова: Параллельные вычисления, многоядерный процессор, графический процессор, программный продукт, имитационное моделирование, ускорение вычислений.

Введение. При проведении исследований с помощью имитационного моделирования часто приходится выполнять расчеты, которые содержат одни и те же вычисления, но с различными данными. Такого рода вычисления, например, требуются при вычислении статистик на основе множества реализаций. При этом если все реализации являются доступными для обработки одновременно, то существенное ускорение вычислений может быть достигнуто за счет распараллеливания. Многие современные компьютеры, даже персональные, имеют центральные процессоры (CPU) с несколькими ядрами, что принципиально обеспечивает возможность применения распараллеливания вычислений. В то же время, всё чаще в архитектуру компьютера включаются графические процессоры (GPU). Такие процессоры являются многоядерными по своей природе.

Следовательно, существует несколько возможностей ускорения вычислений за счет распараллеливания.

Основная часть. В данной статье рассматривается применение распараллеливания вычислений осуществляемых с помощью

МАТЛАБ. За счет распараллеливания можно существенно уменьшить время, затрачиваемое на численные эксперименты, в которых многократно повторяются одни и те же вычисления, но с различными исходными данными.

Для ускорения вычислений нами использовались следующие подходы:

- распараллеливание вычислений на многоядерном CPU;
- параллельные вычисления с использованием GPU.

Для выполнения распараллеливания с использованием GPU применялись два инструмента:

- Parallel Computing Toolbox (PCT) фирмы MathWorks;
- Jacket фирмы AccelerEyes.

Распараллеливание с использованием многоядерного CPU. В последних версиях МАТЛАБ предусмотрена возможность применения распараллеливания вычислений с помощью встроенного механизма Parallel Computing Toolbox (PCT)[1]. PCT позволяет решать вычислительные задачи с большими объемами данных, используя многоядерные процессоры. PCT содержит до двенадцати потоков (worker), которые запускают программы локально на многоядерной вычислительной машине. Таким образом, без изменения кода можно запускать параллельные вычисления в интерактивном или пакетном режиме (с использованием MATLAB Distributed Computing Server).

Распараллеливание с использованием GPU. В отличие от традиционных процессоров, которые включают в себя не более чем несколько ядер, графический процессор имеет сотни ядер. Однако использование GPU для распараллеливания вычислений связано с определенными затратами. Данные должны быть отправлены с центрального процессора на графический процессор для вычислений, а затем извлекаются из него. Поскольку графический процессор присоединяется к центральному процессору посредством шины PCI Express, доступ к памяти происходит медленнее, чем у традиционного CPU. Это означает, что общее ускорение вычислений ограничивается количеством передач данных между CPU и GPU, которое нужно осуществлять в соответствии с используемым алгоритмом обработки данных. Поэтому необходимо тратить время на тонкую настройку кода для GPU с целью оптимизации программ для достижения максимальной производительности. Такая настройка для графических про-

цессоров NVIDIA осуществляется с помощью программно-аппаратной архитектуры CUDA (Compute Unified Device Architecture)[2]. CUDA позволяет производить вычисления с использованием графических процессоров NVIDIA, поддерживающих технологию GPGPU (произвольных вычислений на видеокартах). CUDA дает разработчику возможность по своему усмотрению организовывать доступ к набору инструкций графического ускорителя и управлять его памятью, организовывать на нем сложные параллельные вычисления. Интегрирование CUDA с МАТЛАБ обеспечивается с помощью специально разработанного фирмой Acceleereyes программного продукта под названием Jacket [2]. Jacket – это CUDA плагин, который позволяет производить математические расчеты в среде МАТЛАБ с использованием ресурсов современных видеокарт от NVIDIA. Это настоящий ускоритель расчетов, который перегружает основные процедуры МАТЛАБ, предоставляя полную прозрачность выполнения кода на GPU. Изменения, которые нужно вносить в код программы, минимальны (например, plot ->gplot, surf ->gsurf). Синтаксис многих функций вообще остается без изменений (sin , cos и т.д.).

Характеристики использованных аппаратных и программных средств.

Оборудование:

- Процессор - IntelCore i7, 2.00 GHz, количество ядер – 4
- Оперативная память - 4 GB
- Графический процессор – NVIDIA GeForce GT 555M, 1.35 GHz, количество ядер – 144
- Память графического процессора - 2 GB

Программные средства:

- Windows 7 Ultimate SP1, 64 Bit
- Matlab2011b
- CUDA 4.1
- Parallel Computing Toolbox 6.0
- Jacket 2.1.

Тестирование эффективности механизмов распараллеливания. Описанные выше механизмы распараллеливания вычислений, осуществляемых с помощью МАТЛАБ были применены нами при проведении исследования эффективности различных критериев выбо-

ра оптимальной модели множественной линейной регрессии. Под выбором оптимальной модели подразумевается определение подмножества регрессоров из совокупности всех возможных независимых переменных, которое наилучшим образом «объясняет» наблюдаемые значения зависимой переменной [4]. Определение эффективности различных критериев выбора оптимальной модели проводилось на основе исследования множества реализаций значений всех возможных независимых переменных, полученных с помощью имитационного моделирования [5]. Такой подход связан с необходимостью выполнять однотипные вычисления с различными данными.

Один из методов выбора оптимальной модели множественной регрессии это «метод всех регрессий» [4,5]. Если оценивается модель с k входными переменными, выбираемыми из множества содержащего m переменных, то общее число моделей, которые необходимо проанализировать, будет C_m^k . Расчет критерия качества модели (например, критерия минимума исправленного коэффициента детерминации [5]) можно производить одновременно, если значения всех входных переменных модели известны заранее. Именно это обстоятельство использовалось для распараллеливания вычислений: критерии качества могут быть рассчитаны одновременно для всех C_m^k моделей.

Оптимальная модель выбиралась на основе критерия минимума исправленного коэффициента детерминации [5]. Проводилось по 300 независимых экспериментов для выборок объемом в 500 элементов. Если число входных переменных, которые могут быть включены в модель, равно m , то в каждом эксперименте нужно оценить $m!$ моделей.

Использовалось распараллеливание вычислений на многоядерном CPU с помощью ParallelComputingToolbox (PCT), а также вычисления на GPU. В последнем случае для распараллеливания использовались два механизма – тот же PCT или Jacket. Полученные данные приведены в таблице 1. В таблице приводится время выполнения расчетов. Кроме того, в скобках указана величина ускорения, полученного за счет распараллеливания.

Таблица 1

m	m!	Вычисления на CPU		Вычисления на GPU	
		без распар.	с распар. (PCT)	с использов. PCT	с использов. Jacket
15	32768	5,2531	3,00539 (1,75)	74,77234 (0,07)	1,70391(3,08)
12	4096	0,6597	0,39113 (1,69)	9,561550 (0,07)	0,43767(1,50)
10	1024	0,1817	0,08285 (2,19)	2,389080 (0,08)	0,07154 (2,54)
7	128	0,0327	0,01339 (2,44)	0,288970 (0,11)	0,06903 (0,47)
5	32	0,0140	0,00450 (3,13)	0,079550 (0,18)	0,05534 (0,25)
3	8	0,0044	0,00226 (1,99)	0,016960 (0,26)	0,01248 (0,36)

Приведенные в таблице данные показывают, что при использовании многоядерного CPU с помощью PCT удалось достичь весьма существенного ускорения. При использовании GPU инструмент PCT вместо ускорения дает значительное замедление. Это связано, очевидно, с тем, что когда интерпретатор МАТЛАБ управляет выполнением программы, каждая команда отправляется отдельно в GPU [3]. Это резко замедляет рабочий процесс и убивает производительность. При использовании Jacket однопоточные M-коды превращаются в GPU-приложения, которые полностью используют аппаратные средства [2,3]. Кроме этого, Jacket включает в себя менеджер, который руководит передачей данных между памятью CPU и памятью GPU для оптимизации производительности кода [2]. Эффективность распараллеливания оказывается более высокой, когда число одновременно выполняемых действий велико. Именно такая ситуация возникает, когда число рассматриваемых входных переменных не менее 10. При небольшом числе входных переменных, ускорение за счет параллельных вычислений нивелируется замедлением, связанным с передачей данных между GPU и CPU [3].

Выводы. При проведении расчетов, которые содержат одни и те же вычисления, но с различными данными, имеется принципиальная возможность распараллеливания вычислений. Такая возможность может быть реализована при наличии многоядерных процессоров или графических процессоров в составе применяемого оборудования. В МАТЛАБ использование многоядерного CPU для параллельных вычислений эффективно поддерживается настройкой Parallel Computing Toolbox (PCT). Однако использование данной настройки для GPU не дает ускорения в вычислениях. Такого ускорения удастся

достичь, применяя программный продукт Jacket фирмы AccelerEyes. Jacket обеспечивает эффективное сопряжение МАТЛАБ с программно-аппаратной архитектурой CUDA (Compute Unified Device Architecture), которая позволяет организовывать вычисления с использованием графических процессоров NVIDIA.

ЛИТЕРАТУРА

1. [Электронный ресурс] Режим доступа: <http://www.accelereyes.com/>
2. [Электронный ресурс] Режим доступа: <http://developer.nvidia.com/cuda-gpus>
3. [Электронный ресурс] Режим доступа: <http://computeraidedfinance.com/2012/06/11/matlab-gpu-toolbox-benchmarking-accelereyes-vs-mathworks-vs-gp-you/>
4. Дрейпер Н., Смит Г. – Прикладной регрессионный анализ: В 2-х кн. /Пер. с англ. – 2-е изд., перераб. и доп. – М.: Финансы и статистика. 1987. – 351 с.
5. Крохин В. В., Кузьменко Н. О. Автоматизация выбора оптимальной модели линейной регрессии // Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 1(78). – Днепропетровск, 2012. – С. 73-83.