УДК 004.925.83

РЕШЕНИЕ СЛАУ В ЗАДАЧЕ ТРЕХМЕРНОЙ ИНТЕРПОЛЯЦИИ НА БАЗЕ GPU и CPU

В.С. Бабков, К.К. Титаренко
Донецкий национальный технический университет

В работе рассмотрены практические реализации методов решения СЛАУ большой размерности, как прямых, так и итерационных с использованием графического процессора и технологии CUDA, а также одно- и многоядерного CPU. Даны практические рекомендации по организации вычислительного процесса на данных архитектурах.

Задача построения изоповерхности по множеству дискретных точек в трехмерном пространстве в области компьютерной графики возникает при обработке результатов 2D и 3D-сканирования [1]. Математически задача сводится к интерполяции в трехмерном пространстве и одним из часто используемых в данном случае методов является метод радиальных симметричных функций (RBF). Основной недостаток метода – значительная временная сложность, связанная с решением СЛАУ большой размерности. В работе [2] предложен иерархический метод RBF, который может быть использован для построения поверхностей (2D), а в работе [3] предложена модификация иерархического метода RBF, ориентированная на построение изоповерхностей 3D. Оба указанных метода адаптированы к распараллеливанию.

Иерархические методы, предложенные в [2, 3], с точки зрения реализации на GPU, имеют тот недостаток, что GPU – это SIMD-система, а наибольший выигрыш во временных характеристиках иерархический метод может обеспечить только на MIMD-системах.

Поэтому в работе ставится задача исследовать эффективную, с точки зрения временных затрат, реализацию этапа решения СЛАУ при построении изоповерхности методом RBF с помощью графического процессора и CPU на основе многопоточной модели.

В работе [4] были рассмотрены реализации таких методов решения СЛАУ, как LDLT-разложение и метод Якоби на базе GPU и CPU с использованием многопоточности.

Для метода LDLT были реализованы две модификации, а метод Якоби был полностью реализован с применением CUDA.

В ходе рассмотрения эффективности реализаций различных методов было выяснено, что использовать CUDA целесообразно только для высокопараллельных методов, допускающих возможность оптимизации доступа к памяти.

Т.к. в задаче реконструкции изоповерхности в реальном времени ставятся жесткие временные ограничения, а матрица системы обладает свойством симметричности, но не гарантирует положительной определенности, диагонального преобладания и т.п. качеств, то целесообразны два подхода:

- прямое решение LDLT со значительным увеличением временных затрат;

- итерационное решение, но с предварительной подготовкой матрицы к виду, обеспечивающему сходимость метода.

При этом использование GPU как платформы для решения СЛАУ оправдано только в случае итерационного метода решения и значения $t < 0.1$с для систем с $n \to 10^4$ являются удовлетворительным результатом для реконструкции изоповерхности в реальном времени.

Очередной этап исследований предполагал рассмотрение в качестве альтернативы метода сопряженных градиентов, реализованного с применением многопоточности на базе CPU.

Метод сопряженных градиентов является итерационным методом решения СЛАУ, всегда сходящимся к точному решению при условии положительной определенности и симметричности матрицы коэффициентов максимум за число итераций, не превышающее размерность системы [5].

Итерация метода сопряженных градиентов состоит в вычислении очередного приближения к точному решению в соответствии с правилом:

$$x^k = x^{k-1} + s^k d^k, \tag{1}$$

где

x^k - новое значение приближения;

x^{k-1} - приближение, построенного на предыдущем шаге;

s^k - скалярный шаг;

d^k - вектор направления.

Перед выполнением первой итерации векторы x^0 и d^0 полагаются равными нулю, а для вектора g^0 устанавливается значение, равное - b. Далее каждая итерация для вычисления очередного значения приближения x^k включает выполнение четырех шагов:

- вычисление градиента:
- вычисление вектора направления:
- вычисление величины смещения по выбранному направлению:

- вычисление нового приближения:

Как можно заметить, данные выражения включают две операции умножения матрицы на вектор, четыре операции скалярного произведения и пять операций над векторами. Как результат, общее количество операций, выполняемых на одной итерации, составляет

$$t=2n^2+13n \tag{2}$$

Как уже отмечалось ранее, для нахождения точного решения системы линейных уравнений с положительно определенной симметричной матрицей необходимо выполнить n итераций. Таким образом, для нахождения решения системы необходимо выполнить

$$t=2n^3+13n^2 \tag{3}$$

шагов и, тем самым, сложность алгоритма имеет порядок $O(n^3)$.

При реализации многопотоковой модели решения системы были разработаны классы Vector и SquarteMatrix, инкапсулирующие соответственно вектор и квадратную матрицу и реализующие необходимые векторно-матричные операции с применением технологии OpenMP. Факторы, повышающие в данной реализации производительность, следующие:

- преобладающе последовательное обращение к памяти (использовано свойство матрицы системы – симметричность);
- выравнивание памяти по 16-байтной границе;
- использование OpenMP (выполнение ресурсоемких операций в несколько потоков).

Практические результаты показаны на рис. 1. Зависимости показаны при использовании одного и двух ядер процессора (Intel Core 2 Duo E8500).

Полученные результаты говорят о масштабируемости полученной реализации в контексте многопоточности, т. к. решение задачи на двух ядрах процессора осуществляется в среднем в два раза быстрее, чем на одном ядре. Проведенные исследования позволяют сделать вывод о том, что решение СЛАУ в задаче трехмерной интерполяции при необходимости осуществлять данный процесс в реальном времени целесообразно реализовывать с применением итерационных алгоритмов, распараллеленных в соответствие с моделью SIMD или SIMT на базе GPU или CPU.

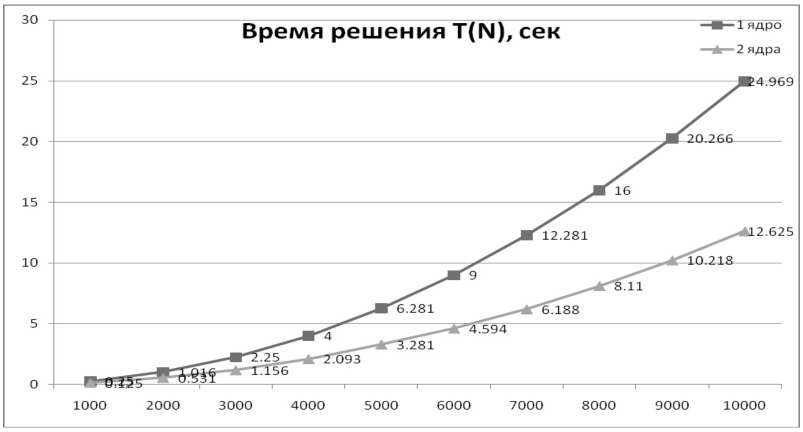

Рис. 1. Зависимость времени расчета от размерности задачи при использовании различного количества ядер CPU

При этом использование GPU предпочтительнее в случае использования методов, в которых расчетные операции преобладают над пересылочными. Также GPU позволяет организовать большее количество параллельных потоков выполнения, чем многоядерный CPU (по крайней мере, на данный момент).

Литература

1. Бабков В.С. 3D-моделювання об’єктів на основі 2D та 3D-проекційних даних / В.С. Бабков // Матеріали IV науково-практичної конференції „Донбас-2020: наука і техніка – виробництву”, 27-28 травня 2008 р. – Донецьк: ДонНТУ Міністерства освіти і науки, 2008. – С. 383–387

2. Pouderox J. Adaptive hierarchical RBF interpolation for creating smooth digital elevation models / J. Pouderox [et al.] // Proc. 12-th ACM Int. Symp. Advances in Geographical information Systems, 2004. – ACP Press. – P. 232–240

3. Бабков В.С. Модифікація ієрархічного методу RBF для отримання 3D-моделей за результатами лазерного сканування / В.С. Бабков // IV Міжнародна науково-практична конференція „Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій”: тези доповіді, 24-26 вересня, 2008 р. – Запоріжжя, ЗНТУ. – С. 116–117

4. Бабков В.С. Решение СЛАУ большой размерности на базе графического процессора в задаче построения изоповерхности / В.С. Бабков, К.К. Титаренко // Наукові праці Донецького національного технічного університету, Серія „Інформатика, кібернетика та обчислювальна техніка”, ДонНТУ, 2009. - № 10 (153). – С. 160-166

5. Golub G.H. Matrix computations / G.H. Golub , C.F. Van Loan: 3-rd ed. – London: John Hopkins University Press, 1996. – 687 p.

Получено 16.05.09