

## АДАПТАЦИЯ НЕКОТОРЫХ РЕШАТЕЛЕЙ СЛАУ ДЛЯ АРИФМЕТИЧЕСКИХ УСКОРИТЕЛЕЙ NVIDIA

*Ю.Г. Бартнев, А.П. Карпов, И.А. Крючков, И.А. Пищулин*

*Российский федеральный ядерный центр – Всероссийский НИИ экспериментальной физики, Саров*

Решению СЛАУ на GPGPU (General Purpose Graphical Processor Unit) уделяется большое внимание ([1-4] и др.). Известно, что эффективная реализация для GPGPU ряда актуальных алгоритмов решателей затруднительна. Это относится к решателям для разреженных матриц, особенно использующим этапы решения треугольных линейных систем (Гаусс-Зейдель, ILU [5]).

Выбор решателя СЛАУ, учет в решателе специфики свойств СЛАУ, потока СЛАУ в приложении позволяет лучше воспользоваться высокой производительностью GPGPU. Авторами разработаны, усовершенствованы программы решателей для NVIDIA GPGPU – наискорейшие (или близкие к ним) на одном ядре универсального микропроцессора в ряде 3D-приложений. Для ЭВМ с одним GPGPU и одним ядром микропроцессора реализованы:

- 1) предобусловливатель ADI [6] (метод переменных направлений) и решатель BiCGSTAB [5] для регулярно разреженных матриц СЛАУ задач теплопроводности;
- 2) предобусловливатель Якоби [5] (обратная точечная или мелкоблочная диагональ) и BiCGSTAB для нерегулярно разреженных матриц простых СЛАУ задач гидродинамики;
- 3) предобусловливатель ILU0/ILDU0 [5] и BiCGSTAB для мелкоблочных нерегулярно разреженных матриц СЛАУ задач аэродинамики;
- 4) возможность сочетания с BiCGSTAB (помимо CG [5]), оптимизирован предобусловливатель AMG библиотеки Parallel Toolbox [7] для потока СЛАУ давления задач гидродинамики с нерегулярно разреженными матрицами.

Реализация выполнена на языке CUDA [8]. Тестирование проведено на СЛАУ размером  $10^6$ , выработанных программами [9-10]. Собственно BiCGSTAB выполняется на Nvidia Tesla C2050® ~ в 15 раз быстрее, чем на ядре Intel Westmere® с частотой 2,93 ГГц.

В методе 1 предобусловливатель представляется произведением нескольких трехдиагональных матриц, производных от исходной матрицы, путем естественного перепорядочивания, количество которых не больше размерности сетки; используются форматы DIA, ELL [5] матриц; решение СЛАУ на GPGPU ускоряется в 7-10 раз.

В методе 2 используется формат CSR [5] исходной матрицы и плотный формат для предобусловливающей обратной блочной диагонали; решение СЛАУ на GPGPU ускоряется в 10 раз.

В методе 3 используется формат CSR матриц; выполняется предварительное вычисление этапов конвейера решения нижней и верхней треугольной системы с возможностью использования результата этой процедуры для решения треугольных СЛАУ в предобусловливателе на итерациях BiCGSTAB (и следующей СЛАУ при совпадении портретов матриц); решение СЛАУ на GPGPU ускоряется в 3-6 раз.

В методе 4 реализован более быстрый алгоритм  $A_c = P^T A_r P$  вычисления грубых матриц в расчете на известное расположение ненулевых коэффициентов в них, которое определяется при решении первой в потоке СЛАУ. Сохраненная пирамида матриц про-

лонгации (P) и указанный алгоритм используются для решения следующей СЛАУ, обладающей матрицей с таким же портретом. Это резко сокращает среднее время этапа построения предобусловливателя AMG, выполняемого на ядре универсального процессора; решение СЛАУ на GPGPU ускоряется в 3-5 раз.

Проведенные эксперименты показывают:

- более быстрое решение СЛАУ пакета ЛОГОС в последовательном режиме с использованием GPGPU при включении данных разработок в LParSol [11] возможно;
- реализация параллельного режима будет менее эффективной из-за выполнения обмена информацией через универсальный микропроцессор;
- скорость решения СЛАУ с разреженными матрицами на TESLA C2050 сдерживают скорость доступа к оперативной памяти и механизма управления потоками. Перевод решателей на OpenCL [12] позволит испытать GPGPU AMD и MIC [13] Intel.

Авторы благодарны А.В. Ерзунову, Е.Б. Щаниковой за консультации по решателям.

### Литература

1. Демидов Д.Е., Егоров А.Г., Нуриев А.Н. Использование технологии NVIDIA CUDA для решения задач гидромеханики // XI Международный семинар «Супервычисления и математическое моделирование»: Тез. докл. Саров, 2009. С. 58-59.
2. Naumov M. Parallel Incomplete-LU and Cholesky Factorization in the Preconditioned Iterative Methods on the GPU: NVIDIA Technical Report NVR-2012-003. – NVIDIA, 2012.
3. Dalton S., Olson L., Bell N. Efficient Algebraic Multigrid Methods on GPUs // Exposing Fine-Grained Parallelism in Algebraic Multigrid Methods: NVIDIA Technical Report NVR-2011-002. Illinois, 2011.
4. Березин С.Б., Каргапольцев И.С., Марковский Н.Д., Сахарных Н.А. Параллельная реализация метода расщепления для системы из нескольких GPU с применением в задачах аэрогидродинамики // Межд. научн. конф. «Параллельные вычислительные технологии 2012»: Сб. тр. Новосибирск, 2012. С. 54–64.
5. Saad Y. Iterative Methods for Sparse linear Systems. Boston: International Thomson Publishing Company, 1996.
6. Meurant G. Computer Solution of Large Linear Systems. Amsterdam: Elsevier Science B.V., 1999.
7. Liebmann M. Algebraic Multigrid Methods on GPU-Accelerated Hybrid Architectures. Institute for Mathematics and Scientific Computing, University of Graz, 2011.
8. NVIDIA CUDA Programming Guide. Ver 4.0. NVIDIA Corporation, 2011.
9. Бабанов А.В., Бельков С.А., Бондаренко С.В. и др. Методика МИМИЗА-НДЗД. Расчет трехмерных задач спектрального переноса излучения // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2012. Вып. 2. С. 64-72.
10. Козелков А.С., Дерюгин Ю.Н., Зеленский Д.К., Глазунов В.А., Голубев А.А., Денисова О.В., Лашкин С.В., Жучков Р.Н., Тарасова Н.В., Сизова М.А. Многофункциональный пакет программ ЛОГОС для расчета задач гидродинамики и теплопереноса на суперЭВМ: базовые технологии и алгоритмы // XII Международный семинар «Супервычисления и математическое моделирование»: Тез. докл. Саров, 2010. С. 65-66.
11. Бартенев Ю.Г., Бондаренко Ю.А., Ерзунов В.А., Карпов А.П., Макаров Л.С., Наумов А.О., Петров Д.А., Романова М.Д., Стаканов А.Н., Щаникова Е.Б., Харченко С.А., Сушко Г.Б., Коньшин И.Н., Прилуцкий М.Х., Старостин Н.В., Филимонов

А.В. Комплекс LParSol для решения СЛАУ // Труды XIII Международного семинара «Супервычисления и математическое моделирование» / Под ред. Р.М. Шагалиева. Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2011.

12. OpenCL. The Open Standard for Heterogeneous Parallel Programming. Khronos Group, 2009.
13. Klemm M. Programming for the Intel Many Integrated Core Architecture. Intel Corporation, 2012.