

Р. Р. Губайдуллин, А. В. Юлдашев

## Разработка MPI-версии линейного решателя для гидродинамического моделирования нефтегазовых месторождений на графических процессорах

### Введение

В процессе решения различных задач компьютерного моделирования часто возникают системы линейных алгебраических уравнений (СЛАУ), для которых характерны большая размерность, сильная разреженность и плохая обусловленность. Для решения таких систем обычно используют предобусловленные итерационные методы подпространства Крылова. В нашем университете разработан специализированный линейный решатель СЛАУ [1], предназначенный для решения систем, возникающих при численном расчете гидродинамических моделей нефтегазовых месторождений. В нем реализован предобусловленный метод бисопряженных градиентов со стабилизацией (BiCGStab), а в качестве предобуславливателей поддержаны неполное LU-разложение без заполнения (ILU(0)) и двухступенчатый предобуславливатель CPR, который считается более перспективным при решении задачи фильтрации вязкой сжимаемой многофазной смеси в пористой среде на параллельных вычислительных системах [2]. Реализация указанных алгоритмов базируется на CUDA, библиотеках cuBLAS, cuSPARSE и AmgX, а также технологии OpenMP, что позволяет задействовать при проведении расчетов от одного до нескольких графических процессоров (GPU) NVIDIA, размещенных в рамках одного гибридного вычислительного узла.

В докладе планируется представить текущие результаты проекта по разработке новой версии решателя СЛАУ, в которой наряду с технологией CUDA и упомянутыми выше библиотеками используется интерфейс передачи сообщений MPI, позволяющий задействовать несколько GPU (multi-GPU) как в рамках одного, так и нескольких вычислительных узлов гибридного кластера, и таким образом повысить

производительность вычислений. В практической части представлено сравнение производительности двух версий решателя, реализующих метод BiCGStab с предобуславливателем ILU(0) с использованием технологий OpenMP и MPI, при решении тестовых СЛАУ на 1-2 узлах гибридного вычислительного кластера, оснащенных GPU NVIDIA Tesla K40.

## 1. Теоретическая часть

В работе метода BiCGStab с предобуславливателем ILU(0) можно выделить три основных ресурсоемких операции:

- построение предобуславливателя ILU(0);
- решение треугольных систем;
- умножение разреженной матрицы на вектор.

Были реализованы следующие подходы к параллельному выполнению указанных операций на системе с несколькими GPU.

Построение предобуславливателя ILU(0) реализуется с использованием блочного метода Якоби на базе блочно-диагональной части исходной матрицы [3], где число блоков равняется числу GPU. В таком случае, за счет отсутствия зависимостей между блоками, становится возможным как параллельное построение предобуславливателя, так и параллельное решение треугольных систем на множестве GPU. При распараллеливании ILU(0) как одной из ступеней предобуславливателя CPR возможно применение комбинированного подхода [4], предполагающего дополнительное разрежение матрицы предобуславливателя в целях увеличения ресурса параллелизма при обработке каждого из блоков.

Умножение разреженной матрицы на вектор декомпозируется явным образом: каждый GPU работает со своим блоком строк умножаемой матрицы. Вектор, на который происходит умножение, хранится распределенно в памяти нескольких GPU и, перед очередным умножением, должен собираться на каждом GPU. Для ускорения обменов между GPU, возможно, при условии соблюдения ряда программно-аппаратных требований, использовать прямую передачу данных между GPU (peer-to-peer), без использования памяти CPU в качестве буфера. Причем, некоторые современные библиотеки MPI, например, OpenMPI и MVAPICH2, позволяют задействовать данный режим неявно в функциях коллективных взаимодействий.

## 2. Практическая часть

В Таблице 1 представлены основные характеристики матриц СЛАУ, на которых производилось тестирование различных версий линейного решателя.

Таблица 1. Характеристики матриц

Название	Размерность	Количество ненулевых элементов	Среднее количество элементов в строке
kslv	872 547	15 397 533	17,64665
imsh	1 500 000	55 815 624	37,21042
immn	2 304 102	42 859 314	18,60131
krrv	4 320 921	85 471 137	19,78077

Все расчеты проводились на 1-2 гибридных узлах (2x CPU Ivy Bridge E5-2690 v2 3.00 GHz, 6x GPU Tesla K40m), объединенных коммуникационной средой Infiniband FDR.

В Таблице 2 приведены результаты работы первой реализации multi-GPU версии решателя, написанной с помощью технологии OpenMP.

Таблица 2. Результаты работы OpenMP-версии решателя

Матрица		Количество задействованных GPU		
		1	2	4
kslv	время решения	0,58699	0,35145	0,57842
	число итераций	19	19	37,5
	ускорение	1	1,670	1,015
imsh	время решения	0,15997	0,09385	0,06428
	число итераций	1,5	1,5	1,5
	ускорение	1	1,696	2,284
immn	время решения	1,62066	0,95534	0,70955
	число итераций	20,5	21	20,5
	ускорение	1	1,696	2,284
krrv	время решения	3,61166	2,00925	1,47184
	число итераций	24,5	24	24,5
	ускорение	1	1,798	2,454

Относительно низкая эффективность распараллеливания в данной версии решателя при использовании нескольких GPU связана главным образом с тем, что возникают накладные расходы на сборку вектора, на который умножается матрица. Так, при увеличении числа GPU с 2 до 4, удельное время, уходящее на соответствующие обмены данными между GPU, в среднем возрастает с 10% до 27% от общего времени решения СЛАУ несмотря на использование режима peer-to-peer, при котором время передачи данных между GPU снижается на 21%-40% для двух GPU и на 16%-30% для четырех GPU.

Для уменьшения времени копирования было предложено минимизировать обмены данными между GPU. Зная, что матрицы решаемых систем сильно разрежены и имеют портрет близкий к блочно-ленточному, было предложено устроить обмены таким образом, чтобы каждый GPU собирал те элементы вектора, которые необходимы для умножения хранящегося на нем блока строк матрицы. Проведение такой оптимизации средствами OpenMP сопряжено с необходимостью разработки оптимальной стратегии обменов с учетом топологии связей между GPU и, по сути, требует реализации коллективных операций передачи данных. Этим обосновывается переход при разработке от стандарта OpenMP к MPI, некоторые реализации которого, например, современные версии OpenMPI, содержат коллективные операции, поддерживающие работу с памятью GPU. В Таблице 3 приведены результаты работы MPI-версии решателя.

Таблица 3. Результаты работы MPI-версии решателя

Матрица		Количество задействованных GPU			
		1	2	4	8
kslv	время решения	0,57644	0,29929	0,33504	0,22103
	число итераций	19	19	37,5	36,5
	ускорение	1	1,926	1,721	2,608
imsh	время решения	0,14947	0,07785	0,04443	0,03635
	число итераций	1,5	1,5	1,5	1,5
	ускорение	1	1,920	3,364	4,112
immn	время решения	1,59157	0,83153	0,44563	0,27594
	число итераций	20,5	21	20,5	20
	ускорение	1	1,914	3,572	5,768

krtv	время решения	3,62313	1,8025	0,93826	0,50705
	число итераций	24,5	24	24,5	25
	ускорение	1	2,010	3,862	7,146

Оптимизация обменов данными позволила сократить их удельное время в среднем с 27% до 3% на 4 GPU, что позволило существенно снизить время решения СЛАУ на 2-4 GPU. Кроме того, использование технологии MPI позволило задействовать 8 GPU, размещенных на 2 узлах гибридного кластера, и, при решении СЛАУ с матрицей наибольшей размерности (krtv), получить ускорение около 7 раз относительно времени вычислений на одном GPU.

В дальнейшем планируется развитие MPI-версии решателя: балансировка нагрузки на GPU, а также реализация более эффективного предобуславливателя CPR.

### Список литературы

- [1] Р. Р. Губайдуллин, Н. В. Репин, А. В. Юлдашев. *Разработка параллельного линейного решателя для гидродинамического моделирования нефтегазовых месторождений на гибридных системах с графическими процессорами* // Суперкомпьютерные дни в России: труды международной конференции (28-29 сентября 2015г., г. Москва). — М.: Изд-во МГУ, 2015, с 392–398. ↑ 1.
- [2] К. Ю. Богачев, И. Г. Горелов. *Применение параллельного предобуславливателя CPR к задаче фильтрации вязкой сжимаемой жидкости в пористой среде* // Вычислительные методы и программирование: НИВЦ МГУ, 2008. Т. 9, с 184–190. ↑ 1.
- [3] Y. Saad. *Iterative methods for sparse linear systems* (2nd ed.) Philadelphia: SIAM, 2003. — 547 p. ↑ 2.
- [4] Р. А. Ахметшин, И. И. Газизов, А. В. Юлдашев. *Комбинированный подход к построению параллельного предобуславливателя для решения задачи фильтрации углеводородов в пористой среде на графических процессорах*, URL [http://2014.nscf.ru/TesisAll/6\\_Supercomputerniy\\_engineering/08\\_215\\_YuldashevAV.pdf](http://2014.nscf.ru/TesisAll/6_Supercomputerniy_engineering/08_215_YuldashevAV.pdf). ↑ 2.

*Об авторах:*

**Ратмир Расилевич Губайдуллин**

Студент 4 курса, направление прикладная математика и информатика.

*e-mail:* [hfnvbhu@gmail.com](mailto:hfnvbhu@gmail.com)

**Артур Владимирович Юлдашев**

Старший преподаватель кафедры высокопроизводительных вычислительных технологий и систем.

*e-mail:* [arthur@mail.rb.ru](mailto:arthur@mail.rb.ru)

*Пример ссылки на эту публикацию:*

Р. Р. Губайдуллин, А. В. Юлдашев. «Разработка MPI-версии линейного решателя для гидродинамического моделирования нефтегазовых месторождений на графических процессорах», *Программные системы: теория и приложения*, 2015, ??:?, с. ??–??.

URL <http://psta.psiras.ru/read/>

Ratmir Gubaidullin, Arthur Yuldashev. *Development of MPI version of linear solver for hydrodynamic modeling of oil and gas fields on graphics processors.*

*Sample citation of this publication:*

Ratmir Gubaidullin, Arthur Yuldashev. “Development of MPI version of linear solver for hydrodynamic modeling of oil and gas fields on graphics processors”, *Program systems: theory and applications*, 2015, ??:?, pp. ??-??. (In Russian.)

URL

<http://psta.psir.ru/read/>