

Специализированный решатель разреженных систем линейных алгебраических уравнений на вычислительных кластерах, оснащенных графическими процессорами

Р.Р. Губайдуллин, Н.В. Репин, Р.Ф. Сайфутдинов, А.В. Юлдашев

Уфимский государственный авиационный технический университет

В данной статье рассмотрены результаты разработки и тестирования параллельной программы, предназначенной для решения на гибридных кластерных системах с графическими процессорами NVIDIA систем линейных алгебраических уравнений с разреженной матрицей, возникающих, в частности, при численном моделировании многофазных фильтрационных потоков в пористой среде. Решение систем уравнений производится с помощью итерационных методов подпространств Крылова с использованием различных предобуславливателей: блочных модификации неполного LU-разложения и двухступенчатого предобуславливателя CPR. Программа разработана средствами MPI, CUDA, а также оптимизированных математических библиотек NVIDIA cuBLAS, cuSPARSE и AmgX.

Ключевые слова: графические процессоры, итерационные методы, многоядерные вычислительные системы, разреженные матрицы, параллельные вычисления.

1. Введение

Одной из практических задач, в рамках которой необходимо многократное высокопроизводительное решение плохо обусловленных систем линейных алгебраических уравнений (СЛАУ) с разреженной матрицей большой размерности, является численное моделирование многофазных фильтрационных потоков в пористой среде (в процессе построения и адаптации трехмерных геолого-гидродинамических моделей нефтегазовых месторождений). В данной статье рассматривается специализированный решатель, предназначенный для решения СЛАУ, возникающих в указанной задаче, на гибридных вычислительных кластерах, оснащенных графическими процессорами NVIDIA. Идея адаптации и реализации алгоритмов решения СЛАУ, характерных для расчета многофазной фильтрации углеводородов, для гибридных вычислительных систем не нова: существуют как специализированные библиотеки, так и полноценные гидродинамические симуляторы [1-4], поддерживающие вычисления на графических процессорах. Однако, все известные авторам разработки в этой области являются закрытыми либо коммерческими программными продуктами, что ограничивает возможность оценки их производительности и масштабируемости на представительной выборке тестовых данных. Это и привело к созданию собственного решателя, реализующего как универсальные, так и специализированные методы решения СЛАУ. Далее приводится информация о ключевых возможностях и особенностях разработанной программы, а также экспериментальная оценка масштабируемости реализованных алгоритмов на современном гибридном вычислительном кластере при решении СЛАУ, содержащих 1,5 – 13,6 миллионов неизвестных.

2. Используемые методы и технологии

В нашем решателе реализованы метод бисопряженных градиентов со стабилизацией (BiCGStab) и гибкий обобщенный метод минимальных невязок (FGMRES). В качестве предобуславливателей выступают как достаточно универсальное неполное LU-разложение без заполнения (ILU(0)), так и специализированный двухступенчатый предобуславливатель CPR [5], применяемый при моделировании фильтрации вязкой сжимаемой жидкости в пористой среде.

Крупноблочное распараллеливание ILU(0), выступающего в роли самостоятельного предобуславливателя либо второй ступени предобуславливателя CPR, реализовано в нашем решателе

с помощью двух методов: блочного метода Якоби (BILU(0)) и аддитивного метода Шварца с ограничениями [6] (RAS ILU(0)). Описание комбинированного подхода к распараллеливанию ILU(0), основанного на совместном использовании блочного метода Якоби и алгоритма разделения на уровни (level scheduling) и позволяющего проводить параллельное построение и применение предобуславливателя ILU(0) на системах с несколькими GPU, приведено в [7].

Решатель СЛАО написан на C/C++. Распределение работы между несколькими GPU реализовано средствами стандарта MPI, а параллельные вычисления в рамках отдельно взятых GPU – средствами CUDA C, а также с помощью следующих библиотек NVIDIA:

- cuBLAS (выполнения векторных операций);
- cuSPARSE (построение и решение треугольных систем, матрично-векторного умножения);
- AmgX [8] (реализация алгебраического многосеточного метода).

Матрица предобуславливателя ILU(0) хранится в формате Block Compressed Sparse Row (BSR), а исходная матрица СЛАО хранится в гибридном формате (HYB), что позволяет обеспечить наибольшую производительность выполнения базовых операций [7].

3. Экспериментальная часть

В таблице 1 приведены характеристики матриц, которые далее используются для оценки масштабируемости разработанного решателя на двух узлах гибридного кластера со следующими характеристиками: 2x CPU Intel Xeon E5-2698 v3 (2.3 GHz, 16 ядер), 4x GPU NVIDIA TESLA M40 (1 GHz, 3072 ядер CUDA, ECC On), коммуникационная сеть Infiniband FDR (Рис. 1). Тестовые матрицы получены при решении уравнений трехфазной фильтрации, дискретизированных по времени с помощью полностью неявной разностной схемы. Для них характерна сильная разреженность (видно из таблицы 1), а также группировка ненулевых элементов в блоки 3x3, что, в частности, учитывается путем хранения матрицы ILU(0) в формате BSR.

Таблица 1. Характеристики тестовых матриц

Название матрицы	Размерность	Количество ненулевых элементов	Среднее количество ненулевых элементов в строке
imsh	1 500 000	55 815 624	37,210
immn	2 304 102	42 859 314	18,601
krrv	4 320 921	85 471 137	19,781
mmnt	5 637 747	109 595 799	19,440
fdrv	6 610 263	118 221 633	17,885
kmms	8 630 895	167 332 329	19,388
lkms	13 665 705	254 102 823	18,594

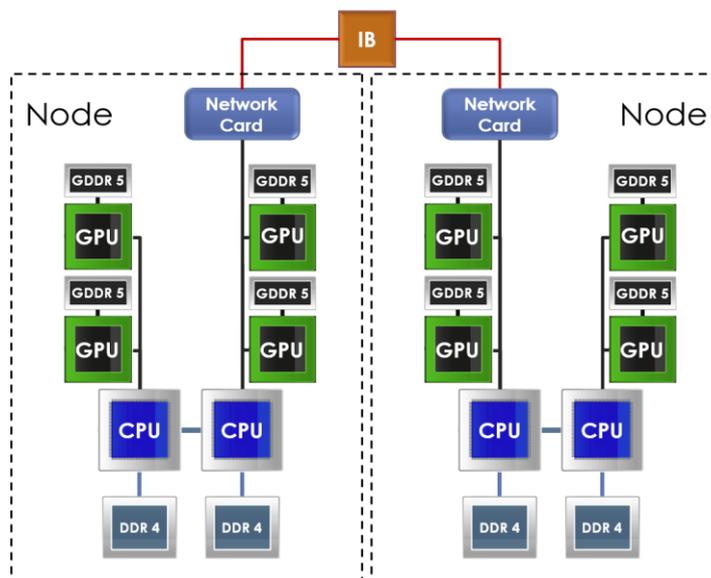


Рис. 1. Двухузловой гибридный вычислительный кластер

Параметры тестирования: условие остановки итерационного процесса – достижение относительной невязки величины $\varepsilon = 10^{-6}$; начальное приближение – нулевой вектор; расчеты проводились с двойной точностью.

3.1 Итерационный метод BiCGStab с предобуславливателем BILU(0)

В Таблице 2 представлены результаты работы итерационного метода BiCGStab с предобуславливателем BILU(0).

Таблица 2. BiCGStab с предобуславливателем BILU(0)

Матрица	Количество GPU	1	2	4	8
imsh	время решения, с.	0,124	0,070	0,046	0,037
	число итераций	1,5	1,5	1,5	1,5
immn	время решения, с.	1,298	0,706	0,414	0,278
	число итераций	20,5	21	20,5	20
krrv	время решения, с.	2,831	1,438	0,789	0,486
	число итераций	24,5	24	24,5	25
mmnt	время решения, с.	3,409	1,747	0,930	0,550
	число итераций	22,5	22,5	22,5	22,5
fdrv	время решения, с.	0,846	0,436	0,253	0,166
	число итераций	4	4	4	4
kmms	время решения, с.	14,483	8,441	4,470	2,424
	число итераций	64	73,5	75	73,5
lkms	время решения, с.	19,449	10,610	5,185	2,933
	число итераций	53,5	58	56	60

На Рисунке 2 представлен график ускорения решения СЛАУ методом BiCGStab с предобуславливателем BILU(0) на 2-8 GPU относительно одного GPU.

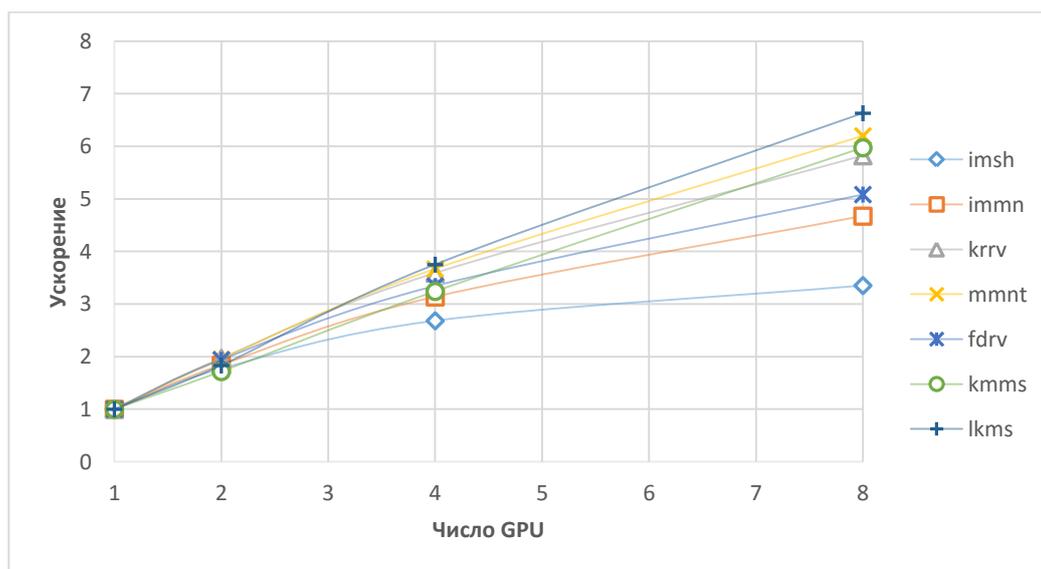


Рис. 2. Ускорение решения СЛАУ методом BiCGStab с предобуславливателем BILU(0).

Худшее ускорение проявляется на матрице наименьшей размерности, в то время как на матрицах наибольшей размерности, несмотря на увеличение числа итераций на 12-14%, уско-

рение получается существенно выше. Это обусловлено тем, что на матрицах меньшей размерности недостаточно ресурса параллелизма для загрузки нескольких GPU.

3.2. Итерационный метод BiCGStab с предобуславливателем RAS ILU(0)

В Таблице 3 представлены результаты работы итерационного метода BiCGStab с предобуславливателем RAS ILU(0).

Таблица 3. BiCGStab с предобуславливателем RAS ILU(0)

Матрица	Количество GPU	1	2	4	8
imsh	время решения, с.	0,124	0,073	0,047	0,039
	число итераций	1,5	1,5	1,5	1,5
immn	время решения, с.	1,298	0,703	0,417	0,287
	число итераций	20,5	20,5	20,5	20
krrv	время решения, с.	2,831	1,508	0,813	0,477
	число итераций	24,5	24,5	24,5	24
mmnt	время решения, с.	3,409	1,775	0,937	0,551
	число итераций	22,5	22,5	22	22
fdrv	время решения, с.	0,846	0,456	0,264	0,160
	число итераций	4	4	4	4
kmms	время решения, с.	14,483	7,812	3,895	2,522
	число итераций	64	66	62	69,5
lkms	время решения, с.	19,449	9,500	5,064	2,713
	число итераций	53,5	50,5	51,5	51

На Рисунке 3 представлен график ускорения решения СЛАУ методом BiCGStab с предобуславливателем RAS ILU(0) на 2-8 GPU относительно одного GPU.

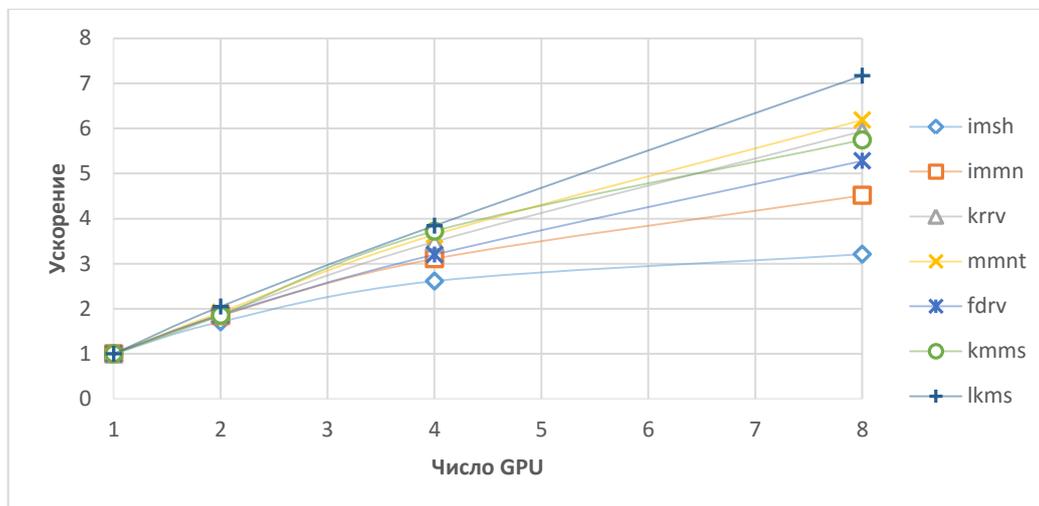


Рис. 3. Ускорение решения СЛАУ методом BiCGStab с предобуславливателем RAS ILU(0)

На отдельных матрицах предобуславливатель RAS ILU(0) позволяет уменьшить число итераций на 18% относительно BiLU(0). Однако, снижение числа итераций не всегда компенсирует увеличение времени решения, которое происходит за счет увеличения размерности локальных треугольных систем, решаемых на каждом GPU, а также дополнительных коммуникаций. В итоге на 8 GPU время решения СЛАУ при использовании RAS ILU(0) составляет от 92% до 104% от времени решения соответствующих систем с предобуславливателем BiLU(0). Макси-

мальный эффект от применения аддитивного метода Шварца с ограничениями в смысле минимизации ухудшения скорости сходимости предобусловленного метода BiCGStab наблюдается при решении СЛАУ с матрицами kmms и lkms наибольшей размерности.

3.3. Итерационный метод BiCGStab с предобуславливателем CPR

В Таблице 4 представлены результаты работы итерационного метода BiCGStab с предобуславливателем CPR.

Таблица 4. BiCGStab с предобуславливателем CPR

Матрица	Количество GPU	1	2	4	8
imsh	время решения, с.	0,282	0,226	0,191	0,192
	число итераций	1,5	1,5	1,5	1,5
immn	время решения, с.	0,469	0,316	0,268	0,195
	число итераций	3,5	3	4	3
krrv	время решения, с.	0,833	0,531	0,372	0,291
	число итераций	3,5	3,5	3,5	3,5
mmnt	время решения, с.	1,165	0,709	0,470	0,313
	число итераций	4	4	4	3,5
fdrv	время решения, с.	0,756	0,483	0,339	0,231
	число итераций	1,5	1,5	1,5	1,5
kmms	время решения, с.	2,034	1,188	0,870	0,602
	число итераций	5	5	6,5	7
lkms	время решения, с.	2,749	1,670	0,981	0,681
	число итераций	4	4,5	4,5	5

На Рисунке 4 показан график ускорения решения СЛАУ методом BiCGStab с предобуславливателем CPR на 2-8 GPU относительно одного GPU.

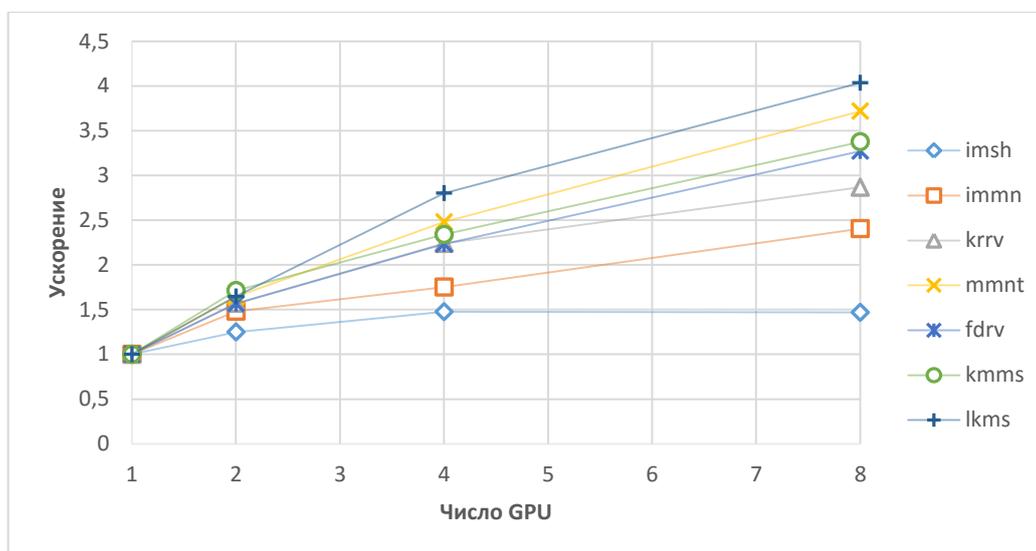


Рис. 4. Ускорение решения СЛАУ методом BiCGStab с предобуславливателем CPR

Как и ранее лучшее ускорение достигается при решении СЛАУ большей размерности. Максимальным результатом при использовании предобуславливателя CPR является ускорение в 4,04 раз на восьми GPU при решении СЛАУ с матрицей lkms.

4. Анализ масштабируемости решателя

Для сравнительной оценки масштабируемости на гибридном вычислительном кластере реализованных в решателе методов, рассчитаем эффективность работы программы на нескольких графических процессорах:

$$E = \frac{S_p}{p} \cdot 100\% = \frac{T_1}{T_p \cdot p} \cdot 100\%, \text{ где } S_p \text{ – ускорение выполнения программы на } p \text{ графиче-}$$

ских процессорах, относительно времени работы программы на одном GPU.

В Таблице 5 приведена эффективность работы программы на 2, 4 и 8 GPU при использовании различных предобуславливателей.

Таблица 5. Эффективность работы решателя при использовании BILU(0), RAS ILU(0), CPR

Матрица	Количество GPU	2	4	8
imsh	BILU(0)	0,884	0,671	0,419
	RAS ILU(0)	0,854	0,653	0,401
	CPR	0,625	0,369	0,184
immn	BILU(0)	0,920	0,784	0,584
	RAS ILU(0)	0,922	0,777	0,565
	CPR	0,741	0,438	0,301
krrv	BILU(0)	0,984	0,897	0,728
	RAS ILU(0)	0,939	0,871	0,742
	CPR	0,785	0,560	0,358
mmnt	BILU(0)	0,976	0,917	0,775
	RAS ILU(0)	0,961	0,910	0,773
	CPR	0,822	0,620	0,465
fdrv	BILU(0)	0,970	0,836	0,636
	RAS ILU(0)	0,929	0,800	0,660
	CPR	0,783	0,558	0,409
kmms	BILU(0)	0,858	0,810	0,747
	RAS ILU(0)	0,927	0,930	0,718
	CPR	0,856	0,585	0,422
lkms	BILU(0)	0,917	0,938	0,829
	RAS ILU(0)	1,024	0,960	0,896
	CPR	0,823	0,700	0,504

На Рисунке 5 приведен график зависимости эффективности для 8 GPU от размерности решаемых СЛАУ.

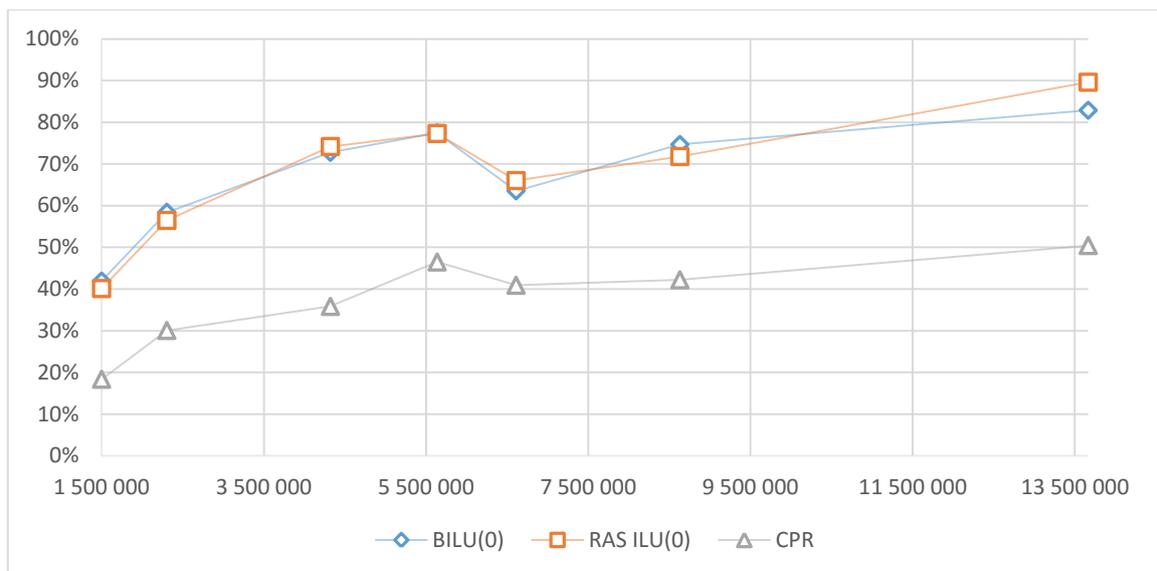


Рис. 5. Зависимость эффективности работы программы на 8 GPU от размерности СЛАУ

Как видно из графика, наибольшая эффективности на 8 GPU достигается при решении СЛАУ с матрицей lkms размерности более 13 миллионов: BILU(0) – 83%, RAS ILU(0) – 90 % и CPR – 50%. CPR демонстрирует более низкую эффективность по сравнению с блочными модификациями ILU(0). Это связано с тем, что AMG, являющийся первой ступенью CPR, оперирует с матрицей СЛАУ на давление, размерность которой меньше в три раза чем у исходной.

Сравнение числа итераций метода BiCGStab при использовании BILU(0), RAS ILU(0) и CPR на 8 GPU представлено на Рисунке 6, а сравнение времени решения СЛАУ на Рисунке 7.

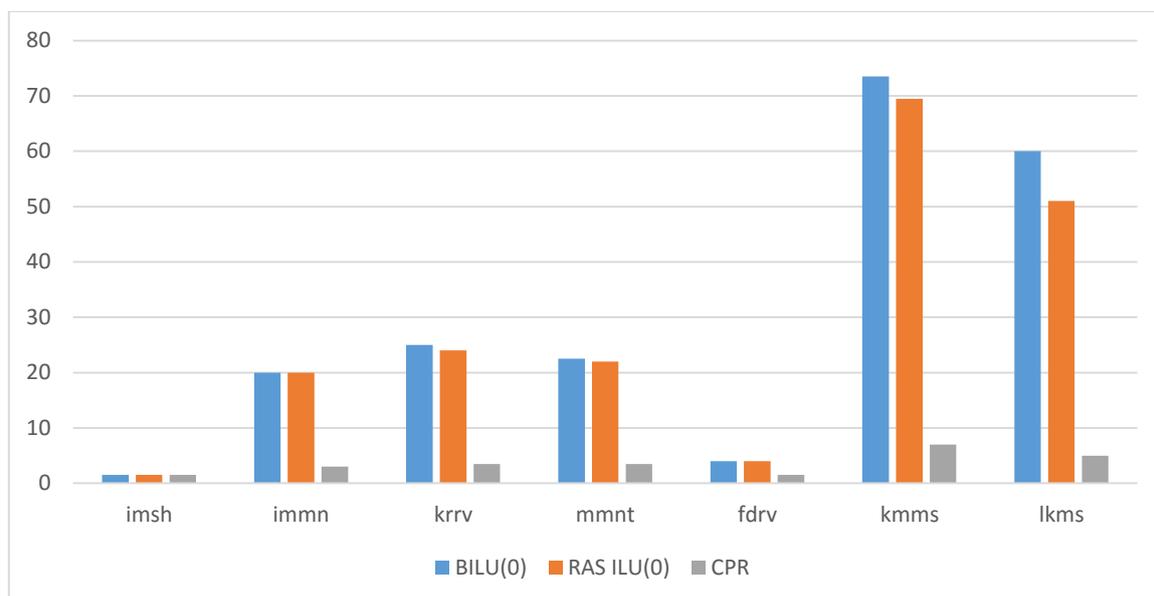


Рис. 6. Число итераций при решении СЛАУ с различными предобуславливателями.

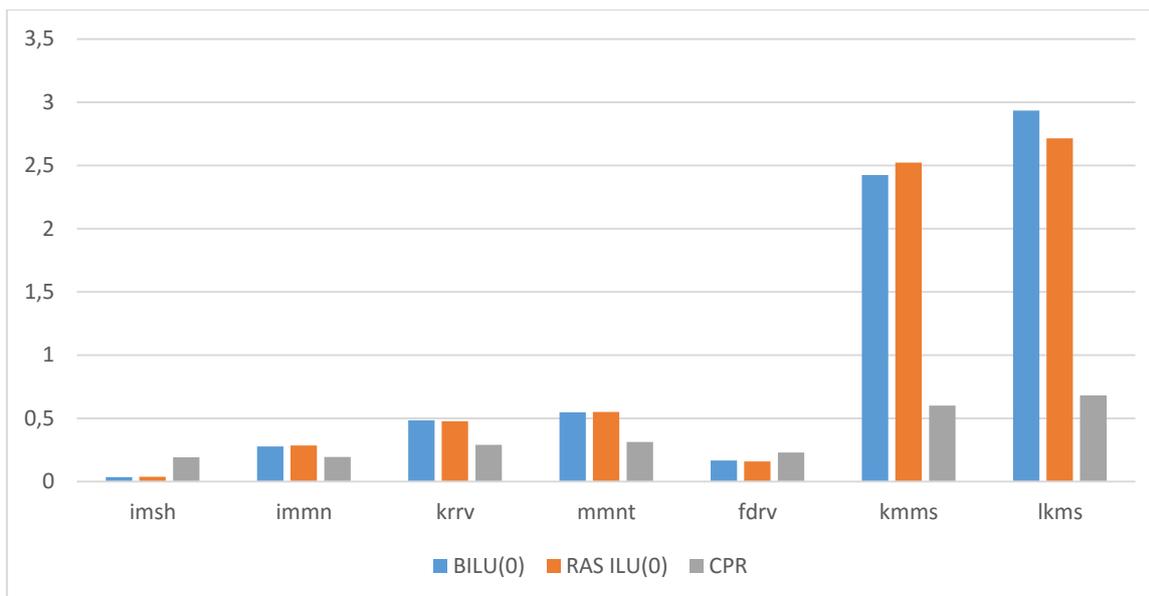


Рис. 7. Время решения СЛАУ методом BiCGStab с различными предобуславливателями.

Видно, что CPR проигрывает во времени решения СЛАУ только с теми матрицами, для которых мало число итераций при использовании блочных предобуславливателей на основе ILU(0). Для матриц, где число итераций при использовании BILU(0) / RAS ILU(0) составляет несколько десятков (все кроме imsh и fdrv), использование CPR позволяет добиться значительно меньшего времени решения СЛАУ. Даже с учетом относительно слабой масштабируемости CPR, выигрыш на 8 GPU относительно BILU(0) / RAS ILU(0) составляет 1,4 – 4,3 раза.

В результате проделанной работы создан и апробирован специализированный решатель разреженных СЛАУ, целевой аппаратной платформой для которого являются кластерные системы, оснащенные графическими процессорами NVIDIA. Эксперименты, проведенные с тестовыми матрицами, выгруженными в ходе численного моделирования многофазных фильтрационных потоков в пористой среде, показали, что масштабируемость решателя при использовании до 8 GPU зависит как от размерности решаемой СЛАУ, так и от используемого предобуславливателя. Наибольшая эффективность работы на гибридном кластере достигается при использовании параллельных блочных предобуславливателей, однако, наименьшее время решения СЛАУ преимущественно (для 5 из 7 рассмотренных матриц) наблюдается при использовании двухступенчатого предобуславливателя CPR.

Литература

1. Z. Chen, H. Liu, S. Yu, B. Hsieh and L. Shao, GPU-based parallel reservoir simulators, 21st International Conference on Domain Decomposition Methods, 2012, France.
2. Z. Chen, H. Liu, and B. Yang, Accelerating iterative linear solvers using multiple graphical processing units, International Journal of Computer Mathematics, 92(7): 1422-1438, 2015.
3. K. Esler, V. Natoli, A. Samardžic. GAMPACK (GPU Accelerated Algebraic Multigrid PACKAGE). ECMOR XIII - 13th European Conference on the Mathematics of Oil Recovery. 2012.
4. K. Esler, K. Mukundakrishnan, V. Natoli, J. Shumway, Y. Zhang and J. Gilman. Realizing the Potential of GPUs for Reservoir Simulation, ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery Catania, Sicily, Italy, 8-11 September 2014.
5. Wallis J. R., Kendall R.P., Little T. E. Constrained residual acceleration of conjugate residual methods // SPE 1353. 1985. P. 415–428.

6. Xiao-Chuan Cai and Marcus Sarkis. A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems. *SIAM Journal on Scientific Computing*, 1999, Vol. 21, No. 2. P. 792–797.
7. Р.Р. Губайдуллин, Н.В. Репин, А.В. Юлдашев. Опыт применения графических процессоров для решения разреженных систем линейных алгебраических уравнений в рамках задачи гидродинамического моделирования нефтегазовых месторождений // *Вестник УГАТУ*, 2015. – Т.19. – №4 (70). – С. 118-123.
8. M. Naumov, M. Arsaev, P. Castonguay, J. Cohen, J. Demouth, J. Eaton, S. Layton, N. Markovskiy, I. Reguly, N. Sakharnykh, V. Sellappan, and R. Strzodka. AmgX: A Library for GPU Accelerated Algebraic Multigrid and Preconditioned Iterative Methods. *SIAM Journal on Scientific Computing*, 2015, Vol. 37, No. 5. P. S602–S626.

Specialized solver of sparse linear systems of algebraic equations on GPU clusters

A.V. Yuldashev, R.R. Gubaidullin, N.V. Repin, R.F. Sayfutdinov

Ufa state aviation technical university

This paper describes the results of development and testing of parallel program, designed for the solution on hybrid cluster systems with NVIDIA graphics processors of big and sparse systems of linear algebraic equations, in particular arising in the numerical simulations of multiphase fluid flows in porous media. Systems of linear algebraic equations are solved using Krylov subspace iterative methods with different preconditioners: block modifications of incomplete LU factorization and two-step CPR preconditioner. Our solver was developed using MPI, CUDA and following optimized mathematical libraries: NVIDIA cuBLAS, cuSPARSE and AmgX.

Keywords: graphics processors, iterative methods, multi- and many-core systems, sparse matrices, parallel computing.

References

1. Z. Chen, H. Liu, S. Yu, B. Hsieh and L. Shao, GPU-based parallel reservoir simulators, 21st International Conference on Domain Decomposition Methods, 2012, France.
2. Z. Chen, H. Liu, and B. Yang, Accelerating iterative linear solvers using multiple graphical processing units, International Journal of Computer Mathematics, 92(7): 1422-1438, 2015.
3. K. Esler, V. Natoli, A. Samardžic. GAMPACK (GPU Accelerated Algebraic Multigrid PACKAGE). ECMOR XIII - 13th European Conference on the Mathematics of Oil Recovery. 2012.
4. K. Esler, K. Mukundakrishnan, V. Natoli, J. Shumway, Y. Zhang and J. Gilman. Realizing the Potential of GPUs for Reservoir Simulation, ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery Catania, Sicily, Italy, 8-11 September 2014.
5. Wallis J. R., Kendall R.P., Little T. E. Constrained residual acceleration of conjugate residual methods // SPE 1353. 1985. P. 415–428.
6. Xiao-Chuan Cai and Marcus Sarkis. A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems. SIAM Journal on Scientific Computing, 1999, Vol. 21, No. 2. P. 792–797.
7. R. R. Gubaydullin, N.V. Repin, A. V. Yuldashev. The experience of using graphics processors for solving sparse systems of linear algebraic equations in the scope of hydrodynamic modeling of oil and gas fields. // Vestnik UGATU, 2015. Vol. 19, No. 4 (70). P. 118–123.
8. M. Naumov, M. Arsaev, P. Castonguay, J. Cohen, J. Demouth, J. Eaton, S. Layton, N. Markovskiy, I. Reguly, N. Sakharnykh, V. Sellappan, and R. Strzodka. AmgX: A Library for GPU Accelerated Algebraic Multigrid and Preconditioned Iterative Methods. SIAM Journal on Scientific Computing, 2015, Vol. 37, No. 5. P. S602–S626.