

ОПЫТ ПРИМЕНЕНИЯ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ ДЛЯ РЕШЕНИЯ РАЗРЕЖЕННЫХ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ В РАМКАХ ЗАДАЧИ ГИДРОДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ НЕФТЕГАЗОВЫХ МЕСТОРОЖДЕНИЙ

Р. Р. Губайдуллин¹, Н. В. Репин², А. В. Юлдашев³

¹ hfnvbhu@gmail.com, ² meujin@gmail.com, ³ art@ugatu.su

ФГБОУ ВПО «Уфимский государственный авиационный технический университет» (УГАТУ)

Поступила в редакцию 14 ноября 2015 г.

Аннотация. Исследовано влияние формата хранения разреженных матриц на время выполнения на графическом процессоре NVIDIA Tesla K20X базовых операций стабилизированного метода бисопряженных градиентов с предобуславливателем ILU(0). Показано, что для операций построения ILU(0) и решения соответствующих треугольных систем наибольшая производительность вычислений достигается при использовании блочного формата BSR, а для операции умножения матрицы на вектор – гибридного формата HVB. Предложен подход к распараллеливанию ILU(0), основанный на комбинации блочного метода Якоби и алгоритма разделения на уровни, позволяющий сократить на 3–16% время решения СЛАУ на одном графическом процессоре, а также задействовать при проведении расчетов несколько графических процессоров. Разработан параллельный решатель СЛАУ, в котором реализованы предобуславливатели ILU(0) и CPR. Проведена оценка его эффективности на гибридной вычислительной системе, оснащенной двумя графическими процессорами NVIDIA Tesla K20X, при решении систем линейных алгебраических уравнений с разреженной матрицей, характерных для задачи численного моделирования многофазной фильтрации углеводородов в пористой среде.

Ключевые слова: алгебра разреженных матриц; графические процессоры; итерационные методы; многоядерные вычислительные системы; параллельные вычисления.

ВВЕДЕНИЕ

В процессе решения ряда задач компьютерного моделирования возникают системы линейных алгебраических уравнений (СЛАУ), для которых характерны большая размерность, сильная разреженность и плохая обусловленность. К примеру, одной из таких практических задач является численное моделирование многофазных фильтрационных потоков углеводородов в пористой среде (гидродинамическое моделирование нефтегазовых месторождений). Как правило, львиную долю времени (до 90%) в процессе численного моделирования занимает блок решения разреженных систем линейных алгебраических уравнений (СЛАУ) [1], поэтому ключевую роль играет производительность и масштабируемость линейного решателя.

Работа выполнена при (частичной) финансовой поддержке гранта Правительства РФ в рамках постановления №220, договор № 11.G34.31.0042.

Для решения разреженных и плохо обусловленных СЛАУ большой размерности принято использовать итерационные методы подпространства Крылова, например, стабилизированный метод бисопряженных градиентов (BiCGStab) с различными предобуславливателями. Из-за высокой жесткости системы найти решение без предобуславливателя или с простейшими вариантами (например, Якоби) не удастся, поскольку при большом числе итераций накопление вычислительной ошибки приводит к значительному искажению приближенного решения [2].

Одним из наиболее популярных предобуславливателей, применяемых, в частности, при численном решении уравнений многофазной фильтрации углеводородов, является неполное LU-разложение без заполнения (с нулевым заполнением) – ILU(0), также используются и другие предобуславливатели класса ILU [3, 4]. Более эффективным с точки зрения сходимости итерационного метода считается специализированный двухступенчатый предобуславливатель Constrained Pressure Residual (CPR) и его моди-

фикации [1, 5, 6]. В рамках CPR выделяется локальный предобуславливатель, в качестве которого обычно используется алгебраический многосеточный метод (AMG) [7], а также глобальный предобуславливатель – упомянутый выше $ILU(0)$. В нашей работе локальный предобуславливатель строится на основе классического алгоритма AMG, в качестве сглаживателя используется блочный метод Якоби, а для решения на грубой сетке – полное LU-разложение.

В настоящее время в сфере высокопроизводительных вычислений широкое применение находят гибридные (гетерогенные) системы, в которых наряду с центральными процессорами (CPU) устанавливаются массивно-параллельные ускорители вычислений, в частности, графические процессоры (GPU) NVIDIA. Эффективное применение гибридных систем сопряжено с необходимостью переработки программного обеспечения путем использования новых технологий параллельного программирования (CUDA, OpenACC и др.) и библиотек либо, как минимум, оптимизации существующих исходных кодов под целевую вычислительную платформу. В некоторых случаях необходим переход к специализированным структурам хранения данных, обеспечивающим эффективную работу с памятью, а также альтернативным алгоритмам, обладающим необходимым ресурсом параллелизма. Так, серверный GPU NVIDIA Tesla K20X, используемый нами для экспериментов, содержит 2 688 процессорных ядер, причем для его загрузки требуется породить десятки тысяч легковесных нитей, выполняющих независимые арифметические операции.

В связи с вышесказанным в нашей работе решаются задачи разработки и реализации параллельных алгоритмов решения разреженных СЛАУ, позволяющих задействовать уникальные вычислительные возможности современных гибридных вычислительных систем. В первой части статьи рассматриваются некоторые существующие и предлагается комбинированный подход к распараллеливанию предобуславливателя $ILU(0)$. Во второй части описываются особенности разработанного параллельного решателя СЛАУ. В третьей части приводятся результаты экспериментов: исследуется влияние формата хранения разреженных матриц на производительность базовых операций метода BiCGStab с предобуславливателем $ILU(0)$, а также оценивается эффективность решателя на гибридной вычислительной системе при решении разреженных СЛАУ, характерных для задачи численного моделирования многофазной фильтрации углеводородов в пористой среде.

ПОДХОДЫ К РАСПАРАЛЛЕЛИВАНИЮ

Алгоритмы построения предобуславливателя $ILU(0)$ и решения систем с нижне- и верхнетреугольными матрицами в классической постановке для сильно разреженных СЛАУ обладают незначительным ресурсом параллелизма. Существует множество подходов к распараллеливанию операций построения и применения предобуславливателей класса ILU [3, 8, 9].

Одним из простейших подходов является построение предобуславливателя с помощью блочного метода Якоби на основе блочно-диагональной части исходной матрицы. За счет отсутствия зависимостей между блоками, становится возможным как параллельное построение предобуславливателя, так и параллельное решение множества независимых треугольных систем. Данный подход приводит к утрате части ненулевых элементов в матрице предобуславливателя с увеличением количества блоков, что негативно влияет на сходимость итерационного метода. Например, в [10] нами было показано, что при разбиении матрицы на 8 и более блоков возможно увеличение количества итераций метода BiCGStab более чем в полтора раза. Следовательно, на GPU, содержащем тысячи процессорных ядер, в чистом виде данный метод использовать проблематично.

Более подходящим для распараллеливания $ILU(0)$ на GPU является алгоритм разделения на уровни (level scheduling). Идея алгоритма состоит в том, чтобы сгруппировать уравнения системы в отдельные подмножества (уровни) так, что в рамках каждого из уровней можно организовать независимую обработку. Данный подход, например, реализован и развит [11, 12] в библиотеке cuSPARSE, входящей в состав NVIDIA CUDA Toolkit. Применение алгоритма разделения на уровни существенно не влияет на сходимость итерационного метода, т. к. при построении $ILU(0)$ сохраняются все ненулевые элементы, а меняется лишь порядок обработки строк матрицы СЛАУ.

В алгоритме разделения на уровни важным параметром, влияющим на эффективность распараллеливания $ILU(0)$, является количество уровней: чем их меньше (при фиксированной размерности матрицы СЛАУ), тем больше независимых строк, которые можно обрабатывать параллельно. Одним из способов снижения числа уровней является разреживание матрицы, причем нами предлагается использовать в качестве процедуры разреживания вышеупомянутый блочный метод Якоби.

В результате предлагаемый комбинированный подход к распараллеливанию $ILU(0)$ состо-

ит в построении предобуславливателя на основе блочно-диагональной части матрицы СЛАУ с использованием алгоритма разделения на уровни как при формировании ILU(0), так и при решении соответствующих треугольных систем.

Комбинированный подход к распараллеливанию ILU(0) предлагается использовать в рамках двухступенчатого предобуславливателя CPR, т. к. сходимости итерационного метода менее чувствительна к качеству LU-разложения, чем при использовании ILU(0) в роли основного предобуславливателя. Также отметим, что комбинированный подход позволяет естественным образом распределить работу при построении и применении ILU(0) между несколькими GPU.

В нашем решателе реализованы следующие варианты распараллеливания ILU(0) на GPU:

1) на одном GPU средствами библиотеки cuSPARSE реализуется разделение строк матрицы на уровни и их дальнейшая параллельная поуровневая обработка;

2) на нескольких GPU используется комбинированный подход: алгоритм разделения на уровни применяется к блочно-диагональной части с количеством блоков, равным числу GPU;

3) на одном или нескольких GPU (в рамках CPR) используется комбинированный подход, причем количество блоков в блочно-диагональной части исходной матрицы выбирается под задачу (в нашей работе от 64 до 256).

ОСОБЕННОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ

GPU-версия решателя разработана средствами CUDA, OpenMP, а также базируется на функциях библиотек cuBLAS и cuSPARSE из состава NVIDIA CUDA Toolkit и библиотеки NVIDIA AmgX. В решателе реализован итерационный метод BiCGStab с предобуславливателями ILU(0) и CPR. Распараллеливание ILU(0) осуществляется в соответствии с описанными выше вариантами в зависимости от конфигурации запуска. Операции построения и применения AMG на данный момент выполняются только на одном GPU, т. к. параллельная версия

классического алгоритма AMG, реализованная в библиотеке AmgX, может быть задействована для систем с несколькими GPU только из MPI-программ. В настоящее время нами ведется разработка альтернативной версии решателя с использованием технологии MPI, что позволит, с одной стороны, обойти указанное ограничение и, с другой, обеспечить возможность решения СЛАУ на гибридном кластере.

Для хранения разреженных матриц в памяти в нашем решателе поддерживается универсальный формат CSR (Compressed Sparse Row), формат BSR (Block compressed Sparse Row), предназначенный для хранения разреженных матриц с блочной структурой, а также гибридный формат HYB. В целях обеспечения максимальной производительности GPU-версии линейного решателя в следующем разделе исследуется влияние формата хранения матриц на время выполнения на GPU базовых вычислительных операций.

ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

В табл. 1 рассмотрены характеристики разреженных матриц СЛАУ, которые используются далее для оценки различных аспектов эффективности решения СЛАУ на GPU. Отметим, что тестовые матрицы получены при решении уравнений трехфазной фильтрации, дискретизированных по времени с использованием полностью неявной разностной схемы. Для них характерна группировка ненулевых элементов в блоки размером 3x3, что может быть учтено в рамках формата BSR при хранении матрицы СЛАУ, а также предобуславливателя ILU(0).

Приведенные далее результаты получены на гибридной вычислительной системе (2 x CPU Intel Xeon E5-2670 и 2 GPU NVIDIA Tesla K20X, ECC Off). В ходе тестирования использовались следующие параметры: условие остановки итерационного процесса – достижение относительной невязкой величины $\varepsilon = 1e-6$; начальное приближение – нулевой вектор; расчеты проводились с двойной точностью.

Таблица 1

Характеристики тестовых матриц

Матрица	Размерность	Кол-во ненулевых элементов	Среднее кол-во элементов в строке
Kslv	872 547	15 397 533	17,65
Imsh	1 500 000	55 815 624	37,21
Imms	2 304 102	42 859 314	18,57
Krrv	4 320 921	85 471 137	19,78
Fdrv	6 610 263	118 221 633	17,85

Таблица 2

Время выполнения на GPU базовых операций в зависимости от формата хранения матриц

Матрица	Время построения ILU(0), с		Время решения треугольных систем, с			Время умножения матрицы на вектор, с		
	CSR	BSR	CSR	BSR	НУВ	CSR	BSR	НУВ
kslv	0,0915	0,0314	0,0190	0,0134	0,0340	0,0019	0,0046	0,0013
imsh	0,1807	0,0651	0,0376	0,0250	0,0651	0,0070	0,0080	0,0064
imms	0,2266	0,0827	0,0450	0,0345	0,0896	0,0052	0,0120	0,0034
krrv	0,4216	0,1595	0,0722	0,0651	0,1500	0,0099	0,0199	0,0057
fdrv	0,5737	0,2326	0,1090	0,0969	0,2257	0,0145	0,0298	0,0096

Исследование влияния формата хранения матриц на время выполнения на GPU базовых операций предобусловленного метода BiCGStab.

К наиболее трудоемким (базовым) операциям метода BiCGStab с предобуславливателем ILU(0) относятся следующие:

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

Ранее в нашем решателе как в CPU-, так и в GPU-версиях для хранения разреженных матриц использовался популярный формат CSR, который поддерживается библиотеками Intel Math Kernel Library и NVIDIA cuSPARSE во всех указанных выше операциях. Начиная с CUDA Toolkit версии 6.0, в библиотеке cuSPARSE появилась соответствующая поддержка формата BSR. В связи с этим в GPU-версии решателя в качестве одного из основных форматов хранения матриц был поддержан BSR. Гибридный формат НУВ может использоваться в cuSPARSE во всех базовых операциях кроме построения предобуславливателя ILU(0).

В данном разделе приводятся результаты исследования влияния используемого формата хранения матриц на производительность вы-

полнения на GPU упомянутых выше базовых операций. В рамках эксперимента задействована библиотека cuSPARSE из CUDA Toolkit 7.5.

В табл. 2 приведены времена выполнения базовых операций при работе с различными форматами хранения матриц (жирным шрифтом отмечены минимальные времена). Видно, что на операциях, связанных с ILU(0), наибольшая производительность достигается при использовании формата BSR: время построения ILU(0) меньше в 2,3–2,6 раза, а время решения треугольных систем меньше в 1,2–1,5 раза относительно работы в формате CSR. В то же время операция умножения разреженной матрицы на вектор выполняется максимально быстро при использовании формата НУВ.

В рассматриваемой в данной работе версии нашего решателя используются два формата хранения матриц в памяти GPU: CSR для хранения матрицы СЛАУ, а BSR для хранения неполного LU-разложения. Однако в связи с полученными результатами для хранения матрицы СЛАУ и выполнения операции умножения разреженной матрицы на вектор в новой версии решателя для кластерных систем внедряется формат НУВ.

Таблица 3

Решение СЛАУ на GPU методом BiCGStab с различными предобуславливателями

Матрица	ILU(0)		CPR (AMG + ILU(0))		CPR* (AMG + ILU(0)*)		
	1 GPU	2 GPU	1 GPU	2 GPU	1 GPU	2 GPU	Количество блоков
Количество итераций							
kslv	19	19	5,5	5,5	5	5	256
imsh	1,5	1,5	1,5	1,5	1,5	1,5	64
imms	20,5	21	10,5	10,5	10,5	10,5	64
krrv	24,5	24	3,5	3,5	3,5	3,5	128
fdrv	4	4	1,5	1,5	1,5	1,5	256
Время решения СЛАУ, с							
kslv	0,65	0,41	0,36	0,29	0,30	0,25	256
imsh	0,17	0,10	0,33	0,26	0,32	0,25	64
imms	1,81	1,12	1,44	1,07	1,39	1,05	64
krrv	4,04	2,34	1,22	0,92	1,16	0,89	128
fdrv	1,21	0,71	1,13	0,87	1,03	0,83	256

Оценка эффективности решения СЛАУ на GPU методом BiCGStab с различными предобуславливателями.

Сходимость итерационного метода с различными предобуславливателями на 1–2 GPU проиллюстрирована в табл. 3. Заметно улучшение сходимости с предобуславливателем CPR, например, при решении СЛАУ с матрицей k_{tv} сходимость улучшается до 7 раз. При этом возможно без ухудшения сходимости применение предобуславливателя CPR*, в рамках которого для извлечения дополнительного параллелизма при работе с $ILU(0)$ реализуется комбинированный подход ($ILU(0)^*$).

Также в табл. 3 приведены времена решения СЛАУ с различными предобуславливателями на 1–2 GPU, включающие в себя время на построение предобуславливателя и итерации метода BiCGStab. Видно, что снижение количества итераций, отмеченное выше при использовании CPR, преимущественно приводит к снижению времени на решение СЛАУ. Исключением является матрица imsh , для которой использование CPR приводит к замедлению решения СЛАУ, т. к. построение предобуславливателя CPR является достаточно трудоемкой операцией, в то время как для обеспечения хорошей сходимости метода (1,5 итерации) достаточно $ILU(0)$. Применение комбинированного подхода к распараллеливанию $ILU(0)$ в рамках CPR позволяет снизить время решения СЛАУ на 3–16% для одного GPU и 3–12% для двух.

Ускорение решения СЛАУ с предобуславливателями CPR и CPR* на 2 GPU относительно 1 GPU составляет 1,2–1,34 раза, с предобуславливателем $ILU(0)$ – 1,61–1,73 раза. Низкая масштабируемость CPR и CPR* связана с тем, что значительную часть общего времени решения СЛАУ (от 24%) при расчете на 1 GPU занимает алгебраический многосеточный метод, реализация которого в рассмотренной версии решателя не позволяет загрузить несколько GPU.

ЗАКЛЮЧЕНИЕ

В работе исследована возможность ускорения на GPU NVIDIA решения СЛАУ, возникающих в ходе численного решения уравнений многофазной фильтрации потоков углеводородов в пористой среде. Показано, что для обеспечения максимальной производительности необходимо применение специализированных форматов хранения данных, а также модифицированных алгоритмов. В целях повышения масштабируемости на системах с несколькими

GPU ведется разработка кластерной версии решателя с использованием технологии MPI.

СПИСОК ЛИТЕРАТУРЫ

1. **Борщук О. С.** О модификации двухступенчатого метода предобуславливания при численном решении задачи многофазной фильтрации вязкой сжимаемой жидкости в пористой среде // Вестник УГАТУ. 2009. Т. 12. № 1 (30), С. 146–150. [O. S. Borschuk, About modification of two-stage preconditioning method for the numerical solution of the multiphase flow problem of a viscous compressible fluid in a porous medium, (in Russian), in *Vestnik UGATU*, vol. 12, no. 1 (30), pp. 146–150, 2009.]
2. **Богачев К. Ю.** Эффективное решение задачи фильтрации вязкой сжимаемой многофазной многокомпонентной смеси на параллельных ЭВМ: дис. ... д. ф.-м. н.: 05.13.18. М.: 2012. 201 с. [K. Y. Bogachev, An effective solution to the problem of filtration of viscous compressible multi-component multiphase mixture on parallel computers: dissertation of doctor of physical and mathematical sciences: 05.13.18, (in Russian). Moscow. 2012.]
3. **Saad Y.** Iterative methods for sparse linear systems. 2nd ed. Philadelphia, PA, USA: SIAM, 2003. 528 p. [Y. Saad. Iterative methods for sparse linear systems. 2nd ed. Philadelphia, PA, USA: SIAM, 2003.]
4. **Богачев К. Ю., Жабицкий Я. В.** Блочные предобуславливатели класса ILU для задач фильтрации многокомпонентной смеси в пористой среде // Вестник МГУ. 2009. № 5, С. 19–25. [K. Y. Bogachev, Y. V. Zhabitsky, "Block ILU class of preconditioners for problems of filtration of a multi-component mixture in a porous medium", (in Russian), in *Vestnik MGU*, no. 5, pp. 19–25, 2009.]
5. **Wallis J. R., Kendall R. P., Little T. E.** Constrained residual acceleration of conjugate residual methods // SPE 1353. 1985. P. 415–428. [J.R. Wallis, R.P. Kendall, T. E. Little. Constrained residual acceleration of conjugate residual methods // SPE 13536. 1985, pp. 415–428.]
6. **Богачев К. Ю., Горелов И. Г.** Применение параллельного предобуславливателя CPR к задаче фильтрации вязкой сжимаемой жидкости в пористой среде // Вычислительные методы и программирование. 2008. Т. 9. С. 184–190. [K. Y. Bogachev, I. G. Gorelov, "The parallel application of CPR preconditioner to the problem of filtration of viscous compressible fluid in a porous medium", (in Russian), in *Vychisl. Metody Programm*, vol. 9. pp. 184–190. 2008.]
7. **Ruge J. W., Stuben K.** Algebraic multigrid (AMG) // Multigrid Methods / S. F. McCormick (ed.) Philadelphia, PA, USA: SIAM, 1987. Vol. 3, p. 73–130. [J. W. Ruge, K. Stuben. Algebraic multigrid (AMG) // Multigrid Methods / S. F. McCormick (ed.) Philadelphia, PA, USA: SIAM, 1987, vol. 3, pp. 73–130.]
8. **Saad Y., Li R.** GPU-accelerated preconditioned iterative linear solvers // The Journal of Supercomputing, February, 2013. Vol. 63, no. 2, p. 443–466. [Y. Saad, R. Li. GPU-accelerated preconditioned iterative linear solvers // The Journal of Supercomputing, February, 2013, vol. 63, no. 2, pp. 443–466.]
9. **Богачев К. Ю., Богатый А. С., Лапин А. Р.** Использование графических ускорителей и вычислительных сопроцессоров при решении задачи фильтрации // Вычислительные методы и программирование: 2013. Т. 14. С. 357–

361. [K.Y. Bogachev, A.S. Bogaty, A.P. Lapin, "Use of graphics cards and coprocessors for solving filtration problems", (in Russian), in *Vychisl. Metody Programm.* vol. 14, no. 3, pp. 357–361, 2013.]

10. **И. И. Газизов, А. В. Юлдашев.** Разработка параллельного линейного решателя для задачи гидродинамического моделирования нефтегазовых месторождений // Эвристические алгоритмы и распределенные вычисления, 2014. Т. 1. № 1. С. 88–96. [I.I. Gazizov, A.V. Yuldashev, Development of parallel linear solver for hydro-dynamic modeling of oil and gas fields, (in Russian), in *Evrlist. algoritmy i raspredel. Vichisl.*, vol. 1, no. 1, pp. 88-96, 2014.]

11. **Naumov M.** Parallel Solution of Sparse Triangular Linear Systems in the Preconditioned Iterative Methods on the GPU: Technical Report NVR-2011-001 NVIDIA, June, 2011. [M. Naumov. Parallel Solution of Sparse Triangular Linear Systems in the Preconditioned Iterative Methods on the GPU: Technical Report NVR-2011-001 NVIDIA, June, 2011.]

12. **Naumov M.** Incomplete-LU and Cholesky Preconditioned Iterative Methods Using CUSPARSE and CUBLAS: Technical Report NVR-2012-003 NVIDIA, May, 2012. [M. Naumov. Incomplete-LU and Cholesky Preconditioned Iterative Methods Using CUSPARSE and CUBLAS: Technical Report NVR-2012-003 NVIDIA, May, 2012.]

ОБ АВТОРАХ

ГУБАЙДУЛЛИН Ратмир Расилевич, студ. по направлению «Прикл. матем. и информатика».

РЕПИН Никита Вячеславович, студ. по направлению «Прикл. матем. и информатика».

ЮЛДАШЕВ Артур Владимирович, ст. преп. каф. высокопроизводительных вычислительных технол. и систем. Дипл. инж.-мат-к (УГАТУ, 2006). Иссл. в обл. высокопроизводительных выч.

METADATA

Title: 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.

Authors: R. R. Gubaydullin¹, N.V. Repin², A. V. Yuldashev³

Affiliation:

Ufa State Aviation Technical University (UGATU), Russia.

Email: ¹hfnvbhu@gmail.com, ²meujin@gmail.com, ³art@ugatu.su

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 19, no. 4 (70), pp. 118-123, 2015. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: The impact of sparse matrices storage format on the execution time of basic operations of BiCGStab method with ILU(0) preconditioner on the NVIDIA Tesla K20X GPU is investigated. The approach to the parallelization of ILU(0) based on the combination of the block Jacobi method and level scheduling algorithm is introduced. Parallel solver for systems of linear algebraic equations implementing ILU(0) and CPR preconditioners is developed. Its effectiveness is evaluated on a hybrid computing system equipped with two NVIDIA Tesla K20X GPUs.

Key words: sparse algebra; graphics processors; iterative methods; multi- and many-core systems; parallel computing.

About authors:

GUBAYDULLIN, Ratmir Rasilevich, Undegrad. student, "Applied Mathematics and Informatics" direction.

REPIN, Nikita Vyacheslavovich, Undegrad. student, "Applied Mathematics and Informatics" direction.

YULDASHEV, Artur Vladimirovich, Senior lecturer, Dep. of high-performance computing technologies and systems. Dipl. eng.-mathematician (UGATU, 2006). Research in high-performance computing.