

УДК 519.683.4

## СРАВНЕНИЕ ИТЕРАЦИОННЫХ МЕТОДОВ РЕШЕНИЯ РАЗРЕЖЕННЫХ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ В ЗАДАЧАХ ФИЛЬТРАЦИИ НА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ

К. Ю. Богачев<sup>1</sup>, Я. В. Жабицкий<sup>1</sup>, А. А. Климовский<sup>1</sup>,  
А. Р. Миргасимов<sup>1</sup>, А. Е. Семенко<sup>1</sup>

Проводится сравнение трех алгоритмов BiCGSTAB, ORTHOMIN и GMRES (последний в двух вариантах — QGMRES и DQGMRES) решения систем линейных уравнений с несимметричными разреженными матрицами по скорости работы и масштабируемости на параллельных ЭВМ: на SMP-системах, на системах с распределенной памятью и на гибридных системах (системах с распределенной памятью, состоящих из SMP-узлов).

**Ключевые слова:** высокопроизводительные вычисления, гибридные MPI-многопоточные программы, итерационные алгоритмы, BiCGSTAB, ORTHOMIN, GMRES, Message Passing Interface (MPI).

**1. Введение.** Среди алгоритмов решения несимметричных разреженных систем линейных уравнений наиболее широкое применение получили три алгоритма: BiCGSTAB (стабилизированный метод бисопряженных градиентов), ORTHOMIN (вариант обобщенного метода сопряженных невязок с неполной ортогонализацией) и GMRES (метод обобщенных минимальных невязок) [1]. Целью настоящей статьи является эмпирический поиск наиболее приспособленного из них для решения систем линейных уравнений, возникающих в задаче фильтрации вязкой сжимаемой многофазной смеси в анизотропной пористой среде [2].

В работе сравнивается скорость работы алгоритмов, а также их масштабируемость на SMP-системах (Symmetric MultiProcessing Systems), системах с распределенной памятью и на гибридных системах (системах с распределенной памятью, состоящих из SMP-узлов). Проводится исследование масштабируемости каждого из алгоритмов (время работы алгоритма сравнивается с временем его работы в другой конфигурации узлов/потоков) и сравнение алгоритмов между собой (время работы алгоритма сравнивается с временем работы другого алгоритма на той же конфигурации узлов/потоков). Все алгоритмы были реализованы с максимальными оптимизациями для параллельных ЭВМ, включая учет неоднородности доступа к памяти (NUMA: Non-Uniform Memory Access) [3].

В результате тестирования на наиболее сложных и высокоразмерных системах, полученных при аппроксимации задач фильтрации для реальных месторождений, оказалось, что наилучшим образом себя проявил алгоритм BiCGSTAB, хотя другие алгоритмы отстали незначительно и могут дать преимущество в отдельных задачах при выборе оптимальных значений параметров.

**2. Постановка задачи.** Рассматриваются матрицы, полученные с помощью гидродинамического пакета моделирования tNavigator из моделей реальных нефтяных месторождений. Характеристики матриц, участвующих в тестировании, приведены в таблице.

Системы решались на кластере из 16 двухпроцессорных узлов, с шестиядерными процессорами Intel Xeon X5650 с частотой 2.67 GHz и соединенных InfiniBand QDR с пропускной способностью 40 Gbit/s.

В качестве предобусловливателя использовался параллельный вариант предобусловливателя ILU(0) с “алгебраическим” разбиением матрицы, показавший наилучший результат на матрицах, полученных при аппроксимации фильтрационных течений для ряда месторождений [4].

**3. Особенности реализации.** Для гибридного MPI-многопоточного распараллеливания векторы и матрица делились поровну между всеми потоками исполнения. При вычислении скалярного произведения, линейной комбинации векторов или произведения матрицы и вектора каждый поток считал свою часть. Далее, в случае вычисления скалярного произведения результат получался посредством общей

<sup>1</sup> Московский государственный университет им. М. В. Ломоносова, механико-математический факультет, Ленинские горы, 119899, Москва; К. Ю. Богачев, доцент, e-mail: bogachev@mech.math.msu.su; Я. В. Жабицкий, аспирант, e-mail: jjv@fromru.com; А. А. Климовский, студент, e-mail: arseny@klimovsky.ru; А. Р. Миргасимов, аспирант, e-mail: mirgasimov.almaz@gmail.com; А. Е. Семенко, студент, e-mail: asemenko@gmail.com

Характеристики тестовых матриц

№	число уравнений	число ненул. элементов	макс. число ненул. элементов в строке	ширина ленты матрицы	коэффициент кососимметричности
1	2 188 842	30 044 852	175	10 094	$1.9 \times 10^7$
2	2 400 402	29 509 724	93	36 096	$1.3 \times 10^6$
3	2 715 768	34 268 312	13	12 716	$2.8 \times 10^7$
4	815 932	10 941 448	89	14 228	$4.8 \times 10^6$
5	1 023 378	19 898 604	118	10 500	$1.4 \times 10^7$
6	858 382	11 691 772	135	15 508	$5.2 \times 10^4$
7	733 713	14 319 189	58	10 167	$1.6 \times 10^5$
8	1 324 920	25 038 036	64	8868	$3.3 \times 10^5$
9	679 262	8 767 636	43	38 600	$1.2 \times 10^6$
10	2 018 124	31 892 472	64	14 466	$5.3 \times 10^6$
11	14 024 025	275 653 395	58	35 334	$2.1 \times 10^8$

MPI-синхронизации. Перед каждой операцией умножения вектора на матрицу и применения предобусловителя происходил MPI-обмен, в ходе которого каждый узел получал элементы вектора, необходимые для подсчета его части в соответствующей операции.

Поскольку каждый узел является NUMA-системой, для повышения быстродействия при распараллеливании на уровне узла был применен подход из [3]: каждый поток был привязан к конкретному физическому ядру и процессору и именно в банке этого процессора выделялась память, с которой этот поток преимущественно работал.

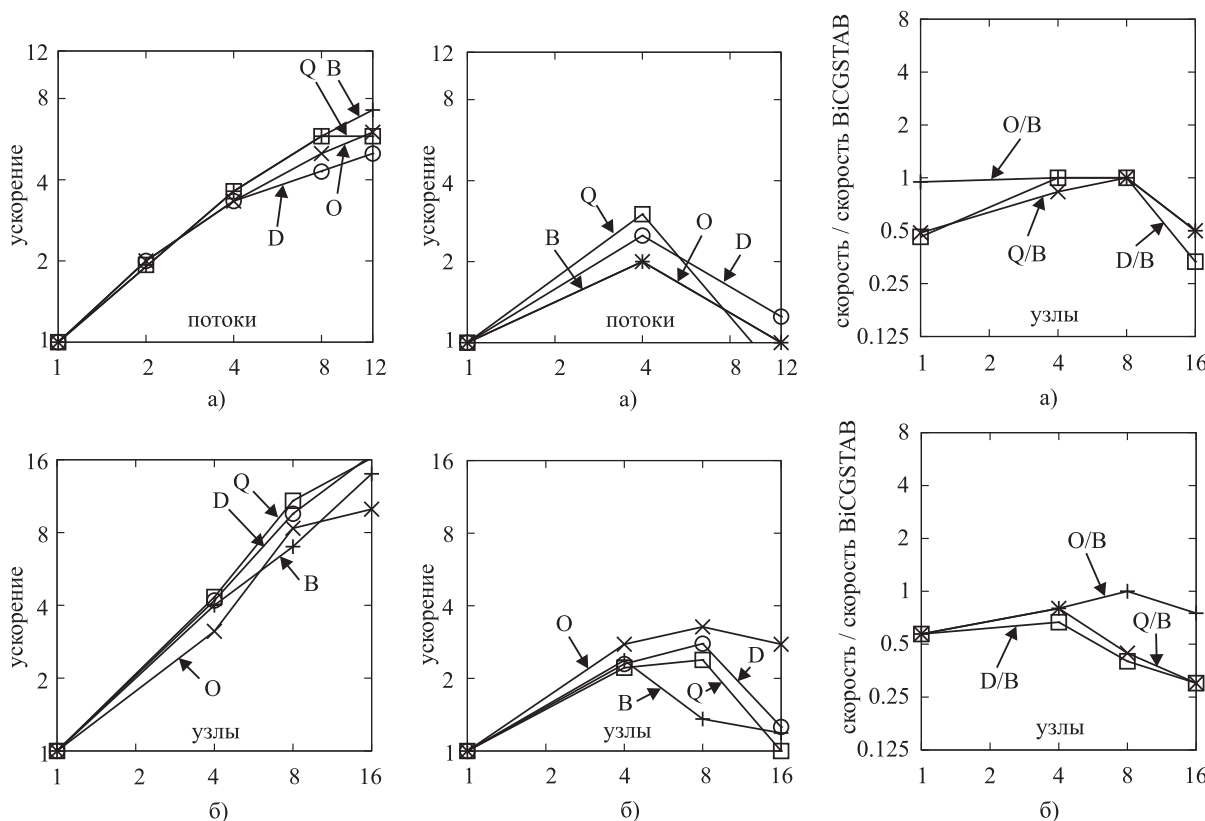


Рис. 1. Ускорение при негибридном распараллеливании: а) один узел, разное количество потоков; б) один поток, разное количество узлов

Рис. 2. Ускорение при гибридном распараллеливании с использованием всех имеющихся (а) узлов (16 узлов, разное количество потоков на каждом узле) и (б) ядер (12 потоков на каждом узле, разное количество узлов)

Рис. 3. Графики сравнения скоростей алгоритмов в зависимости от количества узлов: а) 1 поток, б) 12 потоков

**4. Параметры алгоритмов.** Были проведены тестовые расчеты для определения оптимального

параметра  $k$  в алгоритмах QGMRES( $k$ ), DQGMRES( $k$ ) и ORTHOMIN( $k$ ). Наилучшим значением для всех трех алгоритмов оказалось значение  $k = 5$  и параметр рестарта 50.

Из двух отмеченных в [1] возможностей реализации частичной ортогонализации в этих алгоритмах — подсчет скалярных произведений со всеми векторами базиса с последующим вычислением полной линейной комбинации или последовательная ортогонализация относительно каждого вектора базиса — был выбран первый вариант ввиду меньшего количества синхронизаций ( $O(i)$  против  $O(ik)$ , где  $i$  — количество итераций), что существенно для системы с распределенной памятью (правильность выбора была подтверждена тестовыми расчетами).

**5. Результаты.** Усредненные по матрицам графики ускорения алгоритмов при запуске на одном узле с разным количеством потоков и с одним потоком на разном количестве узлов показаны соответственно на рис. 1а и 1б. На рис. 1–3 буквой В обозначается BiCGSTAB, О — ORTHOMIN, Q — QGMRES и D — DQGMRES. Из рис. 1б видно, что ускорение как функция числа узлов практически линейно (ускорение близко к идеальному). При использовании одной SMP-системы (рис. 1а) ускорение линейно только при небольшом числе потоков из-за недостаточной полосы пропускания оперативной памяти.

На рис. 2 показан результат расчетов в разных комбинациях числа узлов и числа потоков исполнения на узле. На рис. 2а показана зависимость ускорения от числа потоков, использующихся при расчете на 16 узлах, а на рис. 2б — от числа узлов при использовании всех 12 потоков на каждом узле.

В этом случае ускорение невелико: видимо, вычислительные возможности узлов превосходят возможности интерфейса. Оптимальное количество узлов для решения систем с указанными матрицами находится в пределах от 4 до 8. Замедление при включении всех потоков на всех процессорах связано с деградацией предобусловливателя при большом количестве потоков исполнения. Тестируемые матрицы слишком малы для кластера: действительно, для самой большой матрицы (14 миллионов уравнений) каждый поток работает всего с несколькими мегабайтами памяти.

Соотношение скорости работы алгоритмов: на графиках (рис. 3) представлено отношение скорости алгоритмов ORTHOMIN, QGMRES и DQGMRES к BiCGSTAB в зависимости от количества узлов на одном потоке и на 12 потоках. Алгоритмы показывают сопоставимое время, но, как правило, BiCGSTAB оказывается быстрее.

**6. Заключение.** Для ряда больших несимметричных разреженных матриц, полученных при аппроксимации важной в практических приложениях задачи фильтрации, проведено тестирование алгоритмов BiCGSTAB, ORTHOMIN, QGMRES и DQGMRES на параллельных ЭВМ при использовании от 1 до 192 потоков исполнения. В среднем лучше всего как с точки зрения масштабируемости, так и абсолютного времени решения себя проявил алгоритм BiCGSTAB.

#### СПИСОК ЛИТЕРАТУРЫ

1. Saad Y. Iterative methods for sparse linear systems. Philadelphia: SIAM, 2003.
2. Богачев К.Ю., Мельниченко Н.С. О пространственной аппроксимации методом подсеток для задачи фильтрации вязкой сжимаемой жидкости в пористой среде // Вычислительные методы и программирование. 2008. 9, № 2. 42–50.
3. Богачев К.Ю., Миргасимов А.Р. Об оптимизации вычислительных приложений для многопроцессорных систем с общей неоднородной памятью // Вычислительные методы и программирование. 2010. 11, № 2. 40–44.
4. Богачев К.Ю., Жабичский Я.В. Метод Капорина–Коньшина параллельной реализации блочных предобусловливателей для несимметричных матриц в задачах фильтрации многокомпонентной смеси в пористой среде // Вестн. Моск. ун-та. Матем. Механ. 2010. № 1. 46–52.

Поступила в редакцию  
22.12.2010