

М. Н. ВОРОНЮК

## МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ФИЛЬТРАЦИИ В ПЕРКОЛЯЦИОННЫХ РЕШЕТКАХ С ИСПОЛЬЗОВАНИЕМ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ СВЕРХВЫСОКОЙ ПРОИЗВОДИТЕЛЬНОСТИ

Рассматривается процесс математического моделирования заводнения нефтяного месторождения в рамках динамической перколяционной модели на решетках, содержащих  $10^9$  узлов, с использованием суперкомпьютеров с GPU. Предложен алгоритм эффективной балансировки загрузки вычислительных узлов.

**Ключевые слова:** нефтяная добыча, GPU, перколяция, псевдослучайные числа, метод Монте-Карло, балансировка загрузки.

**Введение.** Современные методы добычи углеводородов невозможны без использования высокопроизводительных автоматизированных систем управления. Такие автоматизированные системы, базирующиеся на статистических моделях, позволяют оценить связность двух и более скважин и проводимость между ними, время прорыва агента между добывающей и нагнетательной скважинами и его поведение после прорыва (например, время достижения 50%-ной обводненности вокруг добывающей скважины) [1].

Использование данных моделей сопряжено, однако, с большими вычислительными затратами. Применение для моделирования вычислительных систем сверхвысокой производительности, содержащих графические ускорители (Graphics Processing Unit — GPU), позволяет за приемлемое время получать результаты с высокой точностью, но для эффективного проведения расчетов на графических ускорителях используемые алгоритмы требуют адаптации.

**Постановка задачи.** Для описания процессов нефтевытеснения используется динамическая перколяционная модель (ДПМ) [2]. Пласт представляется в виде агрегата из трех взаимопроникающих составляющих: скелета, жидкой фазы и газовой фазы. В модели нефтесодержащие поры и капиллярные каналы представлены в виде правильной кубической решетки, содержащей  $10^9$  узлов. Каждый канал случайным образом помечается „открытым“ с вероятностью протекания жидкости  $p$  ( $0 \leq p \leq 1$ ) или „закрытым“ с вероятностью  $(1-p)$ . Открытым каналам приписывается значение гидросопротивления. Состояние ( $\alpha$ ) каждого узла (поры) описывается линейной комбинацией трех состояний: узел заполнен нефтью:  $\alpha = +1$ , узел заполнен водой:  $\alpha = -1$  или узел не заполнен (вакантен):  $\alpha = 0$ .

Процесс фильтрации описывается системой кинетических уравнений для решеточных функций [2]:

$$\begin{aligned} \frac{d}{dt} G_{\mathbf{m}}(t) &= - \sum_{\gamma_{\mathbf{m}}} \eta_{\gamma_{\mathbf{m}}} \Omega_{\gamma_{\mathbf{m}}} (G_{\mathbf{m}} - G_{\mathbf{m}+\gamma_{\mathbf{m}}}); \\ \frac{d}{dt} F_{\mathbf{m}}^{\alpha}(t) &= - \sum_{\gamma_{\mathbf{m}}} \eta_{\gamma_{\mathbf{m}}} \Omega_{\gamma_{\mathbf{m}}} (\Psi_{\mathbf{m}+\gamma_{\mathbf{m}},\mathbf{m}} F_{\mathbf{m}+\gamma_{\mathbf{m}}}^{\alpha} F_{\mathbf{m}}^0 - \Psi_{\mathbf{m},\mathbf{m}+\gamma_{\mathbf{m}}} F_{\mathbf{m}+\gamma_{\mathbf{m}}}^0 F_{\mathbf{m}}^{\alpha}); \\ F_{\mathbf{m}}^0(t) &= 1 - \sum_{\alpha=\pm 1} F_{\mathbf{m}}^{\alpha}(t), \end{aligned}$$

где  $\mathbf{m}$  — векторный индекс с целочисленными координатами, характеризующий узел с индексом  $ijk$ ; величины  $F_{\mathbf{m}}^{\alpha}(t)$ ,  $F_{\mathbf{m}}^0(t)$  интерпретируются как средние доли нефти, воды и вакан-

сии в узле  $\mathbf{m}$ ;  $G_{\mathbf{m}}(t)$  — давление в узле  $\mathbf{m}$ ;  $\gamma_{\mathbf{m}}$  — вектор, нумерующий ребра, выходящие из узла  $\mathbf{m}$  (одна из координат вектора  $\gamma_{\mathbf{m}}$  равна  $\pm 1$ , остальные — нулевые);  $\Psi_{\mathbf{m}+\gamma_{\mathbf{m}},\mathbf{m}} = \left( \left| G_{\mathbf{m}+\gamma_{\mathbf{m}}} - G_{\mathbf{m}} \right| - (G_{\mathbf{m}+\gamma_{\mathbf{m}}} - G_{\mathbf{m}}) \right) / 2$  — функция-„ниппель“ ребра  $\gamma_{\mathbf{m}}$ ;  $\eta_{\gamma_{\mathbf{m}}}$  — случайная величина со значениями „0“ и „1“ представляет собой индикатор открытости канала  $\gamma_{\mathbf{m}}$ ;  $\Omega_{\gamma_{\mathbf{m}}}$  — безразмерная случайная проницаемость канала, нормированная к единице.

Для численного моделирования процессов фильтрации использована следующая аппроксимация:

$$\bar{G}_{\mathbf{m}} = G_{\mathbf{m}} - \tau \sum_{\gamma_{\mathbf{m}}} \eta_{\gamma_{\mathbf{m}}} \Omega_{\gamma_{\mathbf{m}}} (G_{\mathbf{m}} - G_{\mathbf{m}+\gamma_{\mathbf{m}}}); \quad (1)$$

$$\bar{F}_{\mathbf{m}}^{\alpha} = F_{\mathbf{m}}^{\alpha} - \tau \sum_{\gamma_{\mathbf{m}}} \eta_{\gamma_{\mathbf{m}}} \Omega_{\gamma_{\mathbf{m}}} (\Psi_{\mathbf{m}+\gamma_{\mathbf{m}},\mathbf{m}} F_{\mathbf{m}+\gamma_{\mathbf{m}}}^{\alpha} F_{\mathbf{m}}^0 - \Psi_{\mathbf{m},\mathbf{m}+\gamma_{\mathbf{m}}} F_{\mathbf{m}+\gamma_{\mathbf{m}}}^0 F_{\mathbf{m}}^{\alpha}); \quad (2)$$

$$\bar{F}_{\mathbf{m}}^0 = 1 - \sum_{\alpha=\pm 1} \bar{F}_{\mathbf{m}}^{\alpha}(t), \quad (3)$$

где  $\tau$  — шаг по времени;  $G_{\mathbf{m}}$ ,  $F_{\mathbf{m}}^{\alpha}$ ,  $F_{\mathbf{m}}^0$  и  $\bar{G}_{\mathbf{m}}$ ,  $\bar{F}_{\mathbf{m}}^{\alpha}$ ,  $\bar{F}_{\mathbf{m}}^0$  — значения решеточной функции в моменты времени  $t$  и  $t+\tau$  соответственно.

**Методика расчета.** Рассматриваемая задача (1)—(3) является стохастической и требует проведения многократных расчетов каждого из сценариев заводнения нефтяного месторождения. Соответственно необходимо задание нового набора случайных величин, характеризующих гидросопротивление межпоровых каналов, для каждого из сценариев, и, кроме того, само их моделирование требует значительных затрат процессорного времени.

Использование внутреннего параллелизма, присущего динамической перколяционной модели, позволяет эффективно производить расчеты на основе параллельных систем сверхвысокой производительности.

Перед непосредственным расчетом сценариев заводнения выполняются следующие подготовительные действия, что позволяет, с одной стороны, избежать их многократного повторения, а с другой — эффективно использовать параллельные системы:

- 1) генерация геометрии месторождения с помощью механизма R-функций [3];
- 2) „наложение“ скважин;
- 3) решение задачи балансировки загрузки вычислительных узлов.

Информация о геометрии месторождения задается с помощью базовых элементов различных форм (эллипсоида, параллелепипеда, эллиптического цилиндра, полупространства) и операций над ними (пересечение, объединение, дополнение, поворот, сдвиг).

Скважины задаются набором цилиндрических секций заданной толщины.

При окончательном расчете каждого из сценариев заводнения выполняются следующие операции:

- 1) генерация перколяционной решетки с помощью генератора псевдослучайных чисел;
- 2) „наложение“ скважин на перколяционную решетку;
- 3) решение системы дифференциальных уравнений (1)—(3).

При генерации перколяционной решетки для задания значения случайной величины  $\eta_{\gamma_{\mathbf{m}}}$  используется генератор псевдослучайных чисел на основе M-последовательностей [4, 5]

$$\Phi_{j+1023} = \Phi_{j+511} + \Phi_{j+127} + \Phi_{j+7} + \Phi_j \bmod 2, \quad (4)$$

где  $\Phi_j$  — одноразрядные члены рекуррентной последовательности, и применяется формула для получения очередного псевдослучайного 32-битового числа [5]:

$$w_{j+8} = (w_j \ll 1) \oplus (w_{j+1} \gg 31) \oplus (w_j \ll 8) \oplus (w_{j+1} \gg 24) \oplus w_{j+4} \oplus w_{j+16}, \quad (5)$$

где  $\oplus$  — операция побитового сложения по модулю 2 целых чисел; операции „ $\gg$ “ и „ $\ll$ “ — побитовые сдвиги вправо и влево соответственно на указанное количество разрядов;  $w_i$  — 32-битовые слова кольцевого буфера,

Указанный генератор (4)—(5) был реализован в библиотеке LRND32-GPU [6] для возможности генерации псевдослучайных чисел на процессоре GPU. Для быстрого получения псевдослучайного числа с требуемым номером используется методика, описанная в работах [5, 7—8], что позволяет производить параллельную генерацию участков перколяционной решетки одного месторождения на нескольких процессорах GPU.

Наибольших вычислительных затрат требует операция интегрирования дифференциальных уравнений давления и насыщенностей фаз, т.е. вычисление (1)—(3).

При расчетах используется двумерный блок шириной  $wb = 32$  нити и высотой  $hb = 6$  нитей (всего 192 нити). В его рамках вычисление (1)—(3) для месторождения (или его части) размером  $L_x \times L_y \times L_z$  узлов производится в два этапа: на первом этапе рассчитывается участок размером  $L_x \times hb \times 1$  для каждого из блоков, на втором этапе — участок размером  $wb \times 1 \times L_z$ . Результаты расчетов складываются [9].

При разбиении решетки на блоки (для поддержания обработки на нескольких GPU) каждый из них содержит, кроме основных, теневые узлы, значения переменных в которых вычисляются в другом блоке. Приграничными будем называть узлы, граничащие с теневыми. При обмене приграничными узлами между несколькими процессорами GPU, находящимися на разных вычислительных блоках, для сокращения времени обмена осуществляется предварительное вычисление данных и асинхронные отсылка и прием. Одновременно с обменом производится вычисление в обычных узлах.

**Балансировка загрузки вычислительных узлов.** Для эффективной балансировки загрузки вычислительных узлов использован двухуровневый метод декомпозиции графа, описывающего перколяционную решетку [10]. Вначале расчетная сетка разбивается на большое число компактных блоков — микродоменов, каждому из которых принадлежит некоторое количество узлов сетки, а макрограф описывает топологию сетки блоков. Каждая из вершин макрографа представляет собой микродомен, а ребро — связь между ними. Вершины имеют веса, равные количеству узлов в микродоменах. Балансировка в этом случае достигается при разбиении макрографа на части по числу процессоров.

Особенностями алгоритма (и GPU в целом) обусловлены дополнительные требования к разбиению:

- 1) параллелепипеды, описанные вокруг вершин микродоменов, не должны иметь пересечений (для ускорения „упаковки“ узлов в сообщения);
- 2) заполнение параллелепипедов узлами должно быть максимально плотным (расчет узлов внутри параллелепипеда осуществляется независимо от того, активен узел или нет);
- 3) параллелепипеды не должны быть слишком малых размеров (в результате экспериментов установлено, что размер микродомена должен составлять  $62 \times 36 \times 36$  узлов и более);
- 4) микродомены не должны иметь граничных „соседей“ в направлении оси  $Ox$  (в то же время разбиение перпендикулярно этой оси возможно для пропуска „пустых“ участков); это требование продиктовано особенностью доступа к памяти GPU: минимальное количество информации, извлекаемое из памяти, составляет 32 байта [11], поэтому при обработке узлов, хаотично расположенных в памяти, происходит резкое снижение производительности GPU (на порядок); в этом случае обработка производится при обмене приграничными узлами.

С учетом указанных требований было принято решение для разбиения регулярной сетки на микродомены использовать параллельный алгоритм геометрической декомпозиции сеточных данных [12], в основе которого лежит метод рекурсивной координатной бисекции. Реализация алгоритма предусматривает получение набора микродоменов, число вершин в которых отличается не более чем на одну. Для этого на каждом этапе бисекции разбиение произ-

водится по медиане перпендикулярно координатной оси, вдоль которой разбиваемый домен имеет наибольшую протяженность.

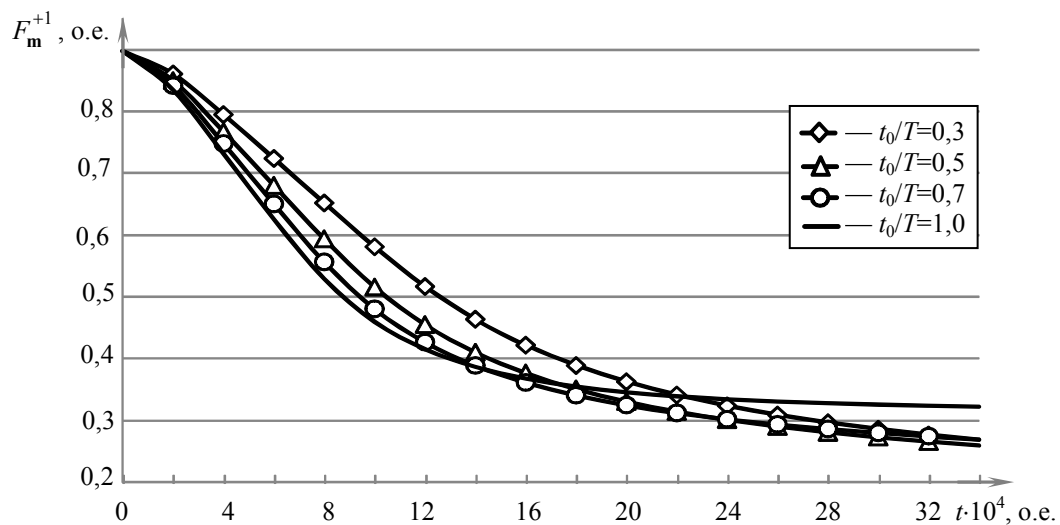
Алгоритм был модифицирован для удовлетворения требований, изложенных в п. 1, так как в исходном варианте получаемые параллелепипеды имели пересечения. В новом варианте разбиения осуществляются не точно по медиане, а так, чтобы плоскость, содержащая медиану, оказывалась полностью на грани одного из получаемых параллелепипедов. В этой модификации разница в числе вершин получаемых микродоменов минимально возможная (в исходном варианте — не более одной вершины). Выполнение требований, изложенных в пп. 2 и 3, обеспечивалось путем определения оптимального количества микродоменов при разбиении; требования п. 4 удовлетворены не были.

Наряду с данным алгоритмом был предложен другой подход, более полно удовлетворяющий указанным требованиям и обладающий вместе с тем достаточной простотой. Исходная сетка в этом случае подвергается разбиению на блоки (микродомены) равной ширины вдоль осей  $Oy$  и  $Oz$ . При этом если полученный блок содержит секции, не имеющие ни одного узла в сечении, параллельном плоскости  $yOz$ , производится попытка его разбиения плоскостями, параллельными  $yOz$ , так чтобы длина блоков была не меньше заданного порогового значения.

Для возможности использования нескольких процессоров GPU макрограф делится на части по количеству карт с помощью алгоритма PartGraphKway пакета Metis [13].

**Результаты.** Моделирование процессов фильтрации нефти было проведено на решетке размером  $2000 \times 2000 \times 250$  узлов для стационарного и нестационарного режимов воздействия на нефтесодержащий пласт. Нагнетательные и добывающие скважины были размещены в шахматном порядке, расстояние между рядами однотипных скважин составляло 80 узлов, каждая скважина занимала  $3 \times 3$  узла. Время расчета — примерно 56 ч с использованием 60 процессоров GPU.

Результаты моделирования приведены на рисунке, где показана зависимость относительного содержания нефти в пласте (среднее значение  $F_m^{+1}$  по всем узлам) от времени при стационарном (отношение времени воздействия  $t_0$  к периоду воздействия  $T=2000$  о.е. равно единице) и нестационарном ( $t_0/T=0,3 \dots 0,7$ ) воздействиях и вероятности открытости каналов  $p=0,4$ . Моделирование проводилось до достижения момента времени  $t=34 \cdot 10^4$  о.е. Наблюдается описанное в работе [2] характерное снижение доли остаточной нефти при импульсном режиме закачки воды по сравнению со стационарным режимом.



Сравнительный анализ методов декомпозиции представлен в таблице, где время выполнения программы приведено в зависимости от примененного метода балансировки загрузки.

В графе „Метод“ символом „—“ обозначен вариант реализации программы без учета геометрии месторождения, в этом случае разбиение решетки осуществляется по методу геометрического параллелизма на равные части (между GPU) параллельно плоскости  $xOz$ ; цифрой 1 обозначен метод геометрической декомпозиции, цифрой 2 — предложенный метод.

Ускорение работы программы в зависимости от используемого метода вычислено по формуле

$$S_{1/2} = \frac{T_-}{T_{1/2}},$$

где  $T_-$  — время работы программы без учета геометрии месторождения,  $T_{1/2}$  — время работы программы с использованием первого или второго метода.

Эффективность рассматриваемых методов вычислена по формуле

$$E_{1/2} = S_{1/2} P_{\text{акт}},$$

где  $P_{\text{акт}}$  — доля активных узлов (в процентах от общего количества).

Количество GPU	Количество узлов	$P_{\text{акт}}, \%$	Метод	Количество микро-доменов	Часть вычисляемых узлов, %	Время выполнения $10^3$ итераций, с	$E_{1/2}, \%$	$S_{1/2}$
1	$0,27 \cdot 10^8$	45,08	—	1	100	44,63	45,08	1
			1	53	54,92	37,31	53,91	1,20
			2	64	62,62	26,87	74,88	1,66
3	$0,91 \cdot 10^8$	44,94	—	3	100	58,81	44,94	1
			1	215	58,07	59,25	44,60	0,99
			2	111	52,42	33,43	79,04	1,76
30	$10^9$	44,60	—	30	100	85,37	30,72	1
			2	495	48,46	42,84	88,90	1,99

Расчеты проводились на гибридной вычислительной системе К-100 (Института прикладной математики им. М. В. Келдыша РАН, Москва), содержащей 192 графических процессора NVIDIA Tesla C2050. Коммуникационная сеть QLogic Infiniband.

**Выводы.** На базе описанной в работе [2] динамической перколяционной модели разработаны алгоритмы и создано программное обеспечение для моделирования процессов фильтрации нефти и активного нефтевытеснения с использованием систем сверхвысокой производительности, содержащих GPU NVIDIA.

Предложен подход, позволяющий осуществлять оптимальную балансировку загрузки вычислительных узлов. Эффективность использования предложенного метода составляет до 74—88 %.

Проведено моделирование на решетке, содержащей  $10^9$  узлов. Согласованность результатов моделирования с результатами, полученными в работе [2] на решетке с меньшим разрешением ( $10^8$  узлов), подтверждает правильность выводов, сделанных в этой работе, и устойчивость динамической перколяционной модели.

Статья подготовлена по результатам работы, выполненной при поддержке Российского фонда фундаментальных исследований, проект № 11-07-00779-а.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Mohsen Masihi, King P. R.* Percolation approach in underground reservoir modeling // *Water Resources Management and Modeling*; Ed. *Purna Nayak* / InTech. 2012. March 3.

2. Лапушкин С. С., Бренерман М. Х., Якобовский М. В. Моделирование динамических процессов фильтрации в перколяционных решетках на высокопроизводительных вычислительных системах // Математическое моделирование. 2004. Т. 16, № 11. С. 77—88.
3. Соллогуб А. В., Вальшин А. Т. Система трехмерного геометрического моделирования пространственных тел с использованием характеристических и R-функций // Программирование. 1991. № 3. С. 86—96.
4. Соболев И.М. Численные методы Монте-Карло. М.: Наука, 1973.
5. Iakobovski M. V., Kornilina M. A., Voroniuk M. N. The scalable GPU-based parallel algorithm for uniform pseudorandom number generation // Proc. of the 2nd Intern. Conf. on Parallel, Distributed, Grid and Cloud Computing for Engineering. 2011. Paper 23.
6. Программные средства, разработанные в ходе выполнения контракта № 07.514.11.4002. [Электронный ресурс]: <[http://apos.imamod.ru/Programms/index\\_prog.html](http://apos.imamod.ru/Programms/index_prog.html)>.
7. Якобовский М. В. Параллельный алгоритм генерации последовательностей псевдослучайных чисел // Математическое моделирование. 2009. Т. 21, № 6. С. 59—68.
8. Brent R. P. On the periods of generalized Fibonacci recurrences // Techn. Report TRCS -92-03, Computer Sciences Laboratory. 1992. March.
9. Воронюк М. Н., Якобовский М. В. Адаптация алгоритмов моделирования динамических процессов фильтрации в перколяционных решетках для графических ускорителей // Математическое моделирование. 2012. Т. 24, № 12. С. 78—85.
10. Abalakin I. V., Boldyrev S. N., Chetverushkin B. N., Zhokhova A. V., Iakobovski M. V. Parallel algorithm for solving flow problems on unstructured meshes // Proc. of 16th IMACS World Congress. Lausanne, Switzerland, Aug. 21—25, 2000.
11. NVIDIA. CUDA C Programming Guide. 2012. Vol. 5.0.
12. Головченко Е. Н. Параллельный пакет декомпозиции больших сеток // Математическое моделирование. 2011. Т. 23, № 10. С. 3—18.
13. METIS—Serial Graph Partitioning and Fill-reducing Matrix Ordering [Электронный ресурс]: <<http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>>.

**Сведения об авторе**

**Михаил Николаевич Воронюк** — аспирант; Московский государственный технологический университет „СТАНКИН“, кафедра информационных систем;  
E-mail: voronyukm@gmail.com

Рекомендована кафедрой  
информационных систем

Поступила в редакцию  
12.02.13 г.