

# Параллельная реализация клеточно-автоматных алгоритмов моделирования пространственной динамики\*

О.Л.Бандман

## Аннотация

Клеточный автомат (КА) является математической моделью пространственной динамики, которая используется, в основном, в тех случаях, когда моделируемое явление представлено нелинейными или разрывными функциями. Поскольку задачи КА-моделирования имеют большие размеры, для их решения необходимо иметь эффективные методы и алгоритмы распараллеливания. Укоренившееся мнение о том, что мелкозернистость КА-алгоритмов обеспечивает простое и эффективное крупноблочное распараллеливание не всегда верно. На самом деле проблема требует решения. В статье делается попытка обосновать общий подход к распараллеливанию КА-алгоритмов, основываясь на условиях поведенческой корректности КА алгоритмов. Сначала дается формальная модель представления КА (алгоритм параллельных подстановок) и приводятся условия их корректности. Затем представлены методы распараллеливания и оценки их эффективности для синхронных и асинхронных КА. Для последних рассмотрен метод достижения высокой эффективности путем аппроксимации их блочно-синхронными КА. Все аналитические положения иллюстрируются результатами вычислительных экспериментов.

**Ключевые слова.** *Клеточный автомат, синхронный режим, асинхронный режим, мелкозернистый параллелизм, блочно-синхронный режим, эффективность параллельных реализаций, алгоритм параллельных подстановок, условия корректности*

## Аннотация

Cellular Automaton (CA) is a mathematical model for spatial dynamics simulation which is capable to model phenomena with strong nonlinearity and discontinuity. Since CA simulation problems size is usually very large, highly efficient methods, algorithms, and software for coarse grained parallelization are urgently needed. The engrained opinion that fine-grained parallelism of CA eliminates the problem of coarse-grained parallelization is shown to be incorrect. The problems needs to be solved. So, a general approach to CA coarse-grained parallelization based on CA-correctness conditions is presented. First, the formal model used for CA representation (Parallel Substitution Algorithm) and CA correctness conditions are given. Then parallelization methods are considered for synchronous and asynchronous CA. To achieve acceptable efficiency for asynchronous CA, a method of approximation it with a block-synchronous CA is proposed. All presented methods are illustrated by computer simulation results.

**Key words:** *Cellular Automaton, synchronous mode, asynchronous mode, block-synchronous mode, fine-grained parallelism, coarse-grained parallelism, efficiency of parallelization, parallel substitution algorithm, correctness conditions.*

---

\*Поддержано 1) Программой фундаментальных исследований Президиума РАН № 14-16(2006)б 2) Интеграционным проектом СО РАН № 29 (2006)

# 1 Введение

Клеточный автомат (КА), предложенный полвека тому назад фон-Нейманом [1] для иллюстрации возможности построения машины, имитирующей биологические явления, получает в наши дни второе дыхание как математическая модель пространственной динамики. Основные свойства КА - естественная мелкозернистая параллельность и локальность взаимодействий при функционировании - делают эту модель привлекательной для создания вычислительных технологий, соответствующих современным и быстро развивающимся многомашинным и многопроцессорным вычислительным комплексам. Предполагается, что КА-подход к моделированию пространственной динамики станет дополнением к традиционному моделированию, основанному на решении дифференциальных уравнений в частных производных [2, 3]. Справедливость такого предположения базируется на двух главных факторах, характеризующих современное состояние вычислительной науки: техническом и математическом. Технический фактор обусловлен развитием многопроцессорных кластеров и компьютерных сетей, которые требуют простых и эффективных технологий распараллеливания задач больших размеров, что легко достигается благодаря мелкозернистости КА-алгоритмов. Математический фактор связан с трудностями, с которыми сталкиваются традиционные вычислительные методы при моделировании нелинейной пространственной динамики, особенно в части распараллеливания. Однако, если трудности распараллеливания с разной степенью эффективности могут быть преодолены, то в случаях когда процессы, подлежащие моделированию, существенно нелинейны, необратимы или прерывисты, дифференциальные уравнения становятся вообще малоприменимыми. Это особенно сильно ощущается в биологии, химии, микроэлектронике, материаловедении, где возникает необходимость моделирования поведения реальных молекул и атомов. Поскольку эти науки в современном мире быстро развиваются, то возрастает и интенсивность поиска новых (альтернативных дифференциальным уравнениям) математических моделей. Одно из направлений такого поиска основано на желании вернуться к истокам математического моделирования, т.е. к описанию явлений на микро- и нано- уровнях, оперируя дискретными событиями, такими как перемещения, изменения состояний, преобразования и взаимодействия реальных или абстрактных частиц, иногда наделяя их скоростью движения в дискретном пространстве. Такие представления явлений легко выражаются в терминах КА, который представляет собой множество связанных по входам и выходам одинаковых конечных автоматов (клеток) с простыми детерминированными или вероятностными правилами переходов (смены состояний).

В настоящее время уже известно много клеточно-автоматных моделей природных явлений, которые в зависимости от назначения различаются алфавитом состояний, структурой дискретного пространства, правилами переходов и "режимами" функционирования. Совместно, они определяют класс моделей и алгоритмов, объединенных понятием *мелкозернистый параллелизм* [4]. Базовой моделью мелкозернистого параллелизма является классический КА фон-Неймана, который имеет булев алфавит состояний, детерминированные правила переходов и синхронный режим функционирования. Классические КА могут моделировать диффузионные и волновые процессы [5], химические реакции [6], фазовые переходы [7], пространственную самоорганизацию [8] и др. Более сложный класс синхронных КА с состояниями в виде булевых векторов, отображающих скорости движения абстрактных частиц, получил большое развитие в виде так называемой Lattice-Gas - гидродинамики [9]. К классу синхронных мелкозернистых моделей относятся и представления явлений вещественными числами в дискретном пространстве и времени. Этот класс моделей совпадает с явными схемами дискретизации дифференциальных уравнений в частных производных.

К мелкозернистым моделям относится также класс асинхронных вероятностных КА, которые применяются, в основном, для моделирования кинетических процессов в наносистемах. Известны результаты их применения для изучения поверхностных химических

реакций на катализаторах [10], адсорбции, сублимации и диффузия атомов на кристаллических решетках [11], в исследованиях морфогенезиса [12] и др. В своих областях знаний такое моделирование часто называется "методом Монте-Карло в [13] этот класс КА определен как "кинетические КА".

Очевидно, что КА моделирование реальных процессов требует больших вычислительных мощностей, поскольку для выявления каких-либо свойств моделируемых процессов необходимо оперировать большими количествами частиц ( $10^{10} - 10^{12}$ ) в течение длительного времени ( $10^3 - 10^5$  итераций). Иными словами, *КА-моделирование составляет широкий класс больших задач, для решения которых необходимо иметь, во-первых, мощные параллельные суперкомпьютеры, и, во-вторых, эффективные методы и алгоритмы распараллеливания, простые и удобные для обычного пользователя.*

Целью статьи является представить и проанализировать методы крупноблочного распараллеливания КА-алгоритмов с синхронным и асинхронным режимом работы и показать их высокую эффективность. Помимо Введения и Заключения статья содержит еще четыре раздела. В следующем за Введением разделе дано формальное представление КА алгоритмов. В третьем разделе обсуждаются условия их корректности, на которых основаны методы распараллеливания. Четвертый раздел посвящен крупно-блочному распараллеливанию КА, в нем приводятся оценки эффективности для синхронных КА и обозначается проблема достижения приемлемой эффективности для асинхронных. В пятом разделе предлагается решение этой проблемы путем аппроксимации асинхронных КА блочно-синхронными.

enddocument

## 2 Формальное представление КА-модели

Клеточный автомат можно себе представить в виде множества клеток, плотно заполняющих дискретное  $D$ -мерное пространство. Расстояния между центрами соседних клеток равны единицам. В каждой клетке помещен конечный автомат, называемый элементарным. Входами элементарного автомата являются выходы других автоматов, находящихся по отношению к нему на некотором ограниченном расстоянии. Все элементарные автоматы одинаковы. Если они меняют свои состояния все одновременно, переводя КА в следующее глобальное состояние, то такой "режим" функционирования КА называется синхронным. Если смена состояний элементарных автоматов происходит стохастически, то КА называется асинхронным. Для формального представления КА далее используется математическая модель мелкозернистого параллелизма, называемая "алгоритмом параллельных подстановок"[14], так как она, в силу своей универсальности, удобна для представления КА-алгоритмов моделирования.

Формально, КА определяется четырьмя понятиями:  $\aleph_\rho = \langle A, M, \Theta, \rangle$ , где  $A$  - алфавит состояний клеток,  $M$  - множество их имен,  $\Theta$  - локальный оператор, индекс  $\rho$  - режим функционирования. Алфавитом состояний может быть любое множество чисел, символов, векторов или матриц, множеством имен - наборы координат точек дискретного пространства. Конкретные значения  $A$  и  $M$  определяют класс *клеточных массивов*  $\Omega(A, M) \in A \times M$ , каждый представитель которого  $\Omega = \{(a, m) : a \in A, m \in M\}$  есть множество пар вида  $(a, m)$ , называемых *клетками*. Клетке соответствует элементарный автомат с именем  $m \in M$  и состоянием  $a \in A$ . На множестве имен вводятся *именующие функции*  $\phi : M \rightarrow M$ , значения которых связывают с любой клеткой  $m \in M$  множество ее соседей

$$S(m) = \{(v_0, m), \dots, (v_k, \phi_k(m)), \dots, (v_q, \phi_q(m))\}, \quad (1)$$

которое называется локальной конфигурацией. Состояния клеток,  $v_0, \dots, v_q$  локальной конфигурации  $S(m)$  играют роль переменных, определенных в алфавите  $A$ , а именующие функции составляют ее определяющий шаблон

$$T_S(m) = \{m, \dots, \phi_k(m), \dots, \phi_q(m)\}. \quad (2)$$

Две локальных конфигурации:

$$S(m) = \{(v_0, m), \dots, (v_q, \phi_q(m))\} \quad \text{и} \quad S'(m) = \{(v'_0, m), \dots, (v'_p, \phi_p(m))\},$$

записанные в виде подстановки

$$\Theta(m) : S(m) \rightarrow S'(m), \quad (3)$$

такой, что

$$T_{S'}(m) \subseteq T_S(m), \quad v'_k = f_k(v_0, \dots, v_q), \quad p \leq q, \quad k = 0.1 \dots, p, \quad (4)$$

составляют локальный оператор, в котором  $(S(m))$  -называется *базой* подстановки или базовой локальной конфигурацией [14].

Локальная конфигурация  $S(m)$  называется совместимой с клеточным массивом  $\Omega \in \Omega(A, M)$ , что обозначается как  $S(m) \sqsubseteq \Omega$ , если

- переменные состояний  $v_k$  во всех клетках  $S(m)$  определены на алфавите  $A$ ,
- клетки с константными состояниями  $(a_k, \phi_k(m)) \in S(m) \cap \Omega$ .

Локальный оператор  $(m)$  вида (3) применим к клетке  $m \in M$ , если выполняется условие:  $S(m) \sqsubseteq \Omega$ . Применение  $\Theta(m)$  к клетке  $m$  состоит в следующем:

- 1) вычисление функций перехода  $f_k(v_0, \dots, v_q)$ ,  $k = 0, \dots, p$ ,
- 2) изменение состояний клеток  $(v_k, \phi_k(m)) \in S'(m)$  на  $v'_k = f_k(v_0, \dots, v_q)$ .

Применение  $\Theta(m)$  к одной клетке происходит за определенный отрезок дискретного времени  $\tau$ , называемый *тактом*. Применение  $\Theta(m)$  ко всем клеткам клеточного массива  $\Omega$  приводит к изменению его глобального состояния  $\Omega(t) \xrightarrow{\Theta(M)} \Omega(t+1)$  и называется итерацией. Последовательность

$$\Sigma(\Omega) = \Omega, \dots, \Omega(t), \Omega(t+1), \dots, \Omega(T),$$

полученная в результате итеративного функционирования КА называется *эволюцией*. Эволюция КА является дискретным представлением пространственно-временной функции. Если итеративный процесс сходится, то эволюция КА имеет завершение, т.е. существует  $t = T$ , такое что  $\Omega(T) = \Omega(T+1) = \Omega(T+2) = \dots$ . В иных случаях КА моделирует колебательный или хаотический процесс [3].

Режим работы КА  $\rho = \{\alpha, \beta, \dots, \sigma\}$  определяет порядок применения локальных операторов к клеткам массива за время одной итерации. Основными режимами являются: синхронный, обозначаемый символом  $\sigma$ , и асинхронный, обозначаемый символом  $\alpha$ . Соответственно, синхронный КА обозначается  $\aleph_\sigma$ , а асинхронный -  $\aleph_\alpha$ . Названные два режима являются двумя крайними случаями. Существуют также смешанные и производные режимы, например, блочно-синхронный  $\aleph_\beta$  [5] и асинхронный упорядоченный [15].

При синхронном режиме изменение состояний происходят только после того, как все функции  $f_k(v_0, \dots, v_q)$  для всех клеток вычислены, т.е. известны новые состояния всех клеток массива. В машине САМ-8[5], которая является аппаратным воплощением КА, все клетки выполняют вычисления функций параллельно и синхронно меняют свои состояния, реализуя потенциальный внутренний параллелизм КА, далее называемый *клеточным параллелизмом*, при котором итерация равна одному такту. В обычном последовательном компьютере синхронная параллельная смена состояний клеток имитируется путем вычисления новых состояний всех клеток последовательно при сохранении их в другом массиве, который служит глобальным состоянием на следующей итерации. Такая параллельность (одновременность) называется *виртуальной*, и вычисление нового глобального состояния занимает время  $\Delta t = \mu \cdot \tau$  тактов, где  $\mu = |M|$ .

При асинхронном режиме никакой одновременности смены состояний клеток не предполагается. Внутренний параллелизм проявляется в том, что смены состояний клеток происходят в произвольном порядке (так называемый "истинный параллелизм" или "true

сопцигенсу"). Каждая клетка вычисляет функцию перехода от тех состояний соседей, которые на данный момент имеют место, и сразу производит замену текущего состояния на новое. Это значит, что функции переходов  $f_k$  (4) могут иметь в качестве аргументов как текущие, так и новые состояния, например,  $(v_k, \phi_k(m)) \in \Omega(t)$ , а  $(v_l, \phi_l(m)) \in \Omega(t+1)$ . Аппаратурные реализации, позволяющие реализовать истинную параллельность асинхронных КА неизвестны. В последовательном компьютере каждая итерация выполняется за  $|M| = \mu$  тактов,  $(\Delta t = \mu \cdot \tau)$  на каждом из которых происходит применение  $\Theta(m)$  к клетке с именем  $m \in M$ . Множество  $\Gamma_\alpha$  возможных последовательностей  $\gamma_\alpha = (m_1, m_2, \dots, m_\mu)$ , к которым применяется  $\Theta(m)$  для перехода в новое глобальное состояние, называемых *трассами* имеет мощность

$$|\Gamma_\alpha| = \mu!. \quad (5)$$

Важен и тот факт, что в общем случае разные трассы приводят к разным результатам, т.е. две трассы  $\Gamma_\alpha^{(1)}$  и  $\Gamma_\alpha^{(2)}$ , могут привести от  $\Omega(t)$  к  $\Omega^{(1)}(t+1) \neq \Omega^{(2)}(t+1)$ . Множество разных результатов глобальных переходов при асинхронном режиме обозначается  $\Psi(\Theta, \Omega)$ , причем

$$|\Psi(\Theta, \Omega)| \leq |\Gamma_\alpha|.$$

Однако, существуют и такие  $\aleph_\alpha$ , для которых  $|\Psi(\Theta, \Omega)|=1$  для любого  $\Omega \in A \times M$  [16]. Тот факт, что в функциях переходов при синхронном режиме используются переменные текущей итерации, а при асинхронном - переменные могут быть как текущие, так и новые, приводят к тому, что при одном и том же наборе операторов и одном и том же исходном клеточном массиве, эволюции синхронного и асинхронного КА совершенно различны. Более того, различными могут быть их принадлежности к разным классам по поведенческим свойствам [3]. Следующий пример показывает, насколько могут быть разными эволюции КА, отличающиеся только режимами функционирования.

**Пример1.** КА-модель распространения волны - это первая и самая простая 2D-модель газовой динамики [9]. Газ представлен четырехслойным клеточным массивом  $\Omega = \{(v, (i, j, k))\}$  с алфавитом состояний  $A = \{0, 1\}$ . Множество имен  $M = \{(i, j, k) : i, j = 0, \dots, N; k = 1, 2, 3, 4\}$ . Значение состояния  $v = 1$  клетки с именем  $(i, j, k)$  означает, что в точке с координатами  $(i, j, k)$  двумерного дискретного пространства находится "модельная" частица, движущаяся с единичной скоростью вдоль оси  $i$  или  $j$ , оставаясь при этом в  $k$ -м слое. В каждом  $k$ -м слое все частицы движутся в одном направлении, соответствующим значению именуемой функции в шаблонах

$$T_k(i, j, k) = \{(i, j, k), \phi_k(i, j, k)\},$$

где

$$\begin{aligned} \phi_1(i, j, 1) &= (i-1, j, 1), & \phi_2(i, j, 2) &= (i, j+1, 2), \\ \phi_3(i, j, 3) &= (i+1, j, 3), & \phi_4(i, j, 4) &= (i, j-1, 4). \end{aligned}$$

Режим функционирования этой модели синхронный, двухфазный, т.е. каждая итерация делится на две фазы: столкновение и сдвиг. На фазе столкновения ( $\Theta_1$ ) происходит смена направления движения на  $90^\circ$  тех пар частиц, которые попали в клетки с одноименными  $(i, j)$ , двигаясь навстречу друг другу.

$$\Theta_1(i, j, k) : \{(u, (i, j, 1)), (\bar{u}, (i, j, 2)), (u, (i, j, 3)), (\bar{u}, (i, j, 4))\} \rightarrow \{(\bar{u}, (i, j, 1)), (u, (i, j, 2)), (\bar{u}, (i, j, 3)), (u, (i, j, 4))\}$$

На фазе сдвига ( $\Theta_2$ ) каждая частица перемещается в ту соседнюю клетку, на которую указывает ее вектор скорости.

$$\Theta_2(i, j, k) : (u, (i, j, k)) \rightarrow (u, \phi_k(i, j, k)) \quad k = 1, 2, 3, 4.$$

На рис. 1 показано 3 глобальных состояния в эволюции этого КА. Видно, что при начальном состоянии, имеющем вид квадрата большой концентрации в центре клеточного массива, появляется круговая волны, распространяющаяся от центра.

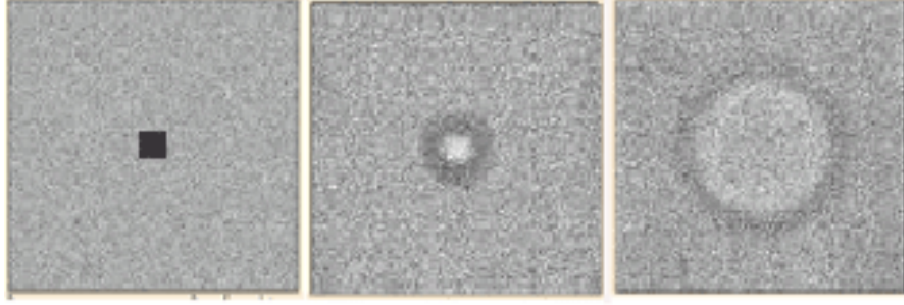


Рис. 1: Три глобальных состояния ( $t=0$ ,  $t=30$ ,  $t=100$ ) из эволюции синхронного КА, моделирующего распространение круговой волны в газовой среде.

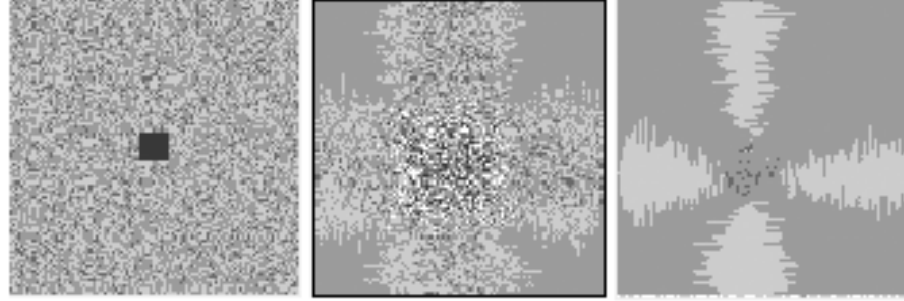


Рис. 2: Три глобальных состояния ( $t=0$ ,  $t=100$ ,  $t=320$ ) из эволюции асинхронного КА, отличающегося от синхронного (рис.1) только режимом функционирования.

При асинхронном функционировании КА с теми же локальными операторами, при тех же начальных данных, имеет совсем другую эволюцию (рис.2).

Эквивалентные преобразования  $\aleph_\alpha \leftrightarrow \aleph_\beta$ , сохраняющие эволюции, изучены слабо. Известна теорема о том, что для каждого синхронного КА можно построить такой асинхронный, который будет моделировать эволюцию исходного с растяжкой во времени [17]. К сожалению, способа построения синхронного КА, воспроизводящего эволюцию заданного асинхронного не существует. А именно этот способ был бы полезен для решения проблемы крупно-блочного распараллеливания асинхронных КА, поскольку для синхронных КА эта проблема решается просто и эффективно.

### 3 Условия корректности КА-моделей

Как всякая математическая модель КА должен удовлетворять условиям поведенческой и вычислительной корректности. Эти условия для КА просты и естественны, но легко могут быть нарушены при крупноблочном распараллеливании.

**Условие детерминированности:** *локальный оператор  $\Theta(m) : S(m) \rightarrow S'(m)$  в один и тот же момент  $\tau$  может изменять состояние не более чем одной клетки из каждого подмножества клеток  $S'(m) \in \{S'(m_1), \dots, S'(m_\mu)\}$ .*

Это условие гарантирует отсутствие коллизий, т.е. таких ситуаций когда одной и той же клетке на одном и том же такте предписано сменить свое состояние разными функциями переходов  $f_k(m)$  и  $f_l(\phi(g(m)))$ . Это значит, что одновременность (пусть виртуальная) применения  $\Theta(m)$  допускается только к такому подмножеству клеток  $S' \subseteq S$ , для которого выполняется условие:

$$T_{S'}(m'_g) \cap T_{S'}(m'_h) = \emptyset \quad \forall (m'_g, m'_h) \in M'. \quad (6)$$

Очевидно, что классические синхронные КА, у которых  $|S'(m)| = 1$ , всегда удовлетворяют условию (6). Но, если в синхронном КА правая часть  $\Theta(m)$  (3) имеет  $|S'(m)| > 1$ ,

то при применении  $\Theta(m)$  ко всем  $m \in M$  одновременно (пусть виртуально одновременно) вычисление некорректно. Устранение такой некорректности выполняется путем разделения вычисления функций переходов (4) во времени. Это приводит к применению блочно-синхронного режима, в котором каждая итерация разделена на  $p = |S'(m)|$  этапов, а  $\Omega$  разделен на  $b = |M|/p$  блоков, и вычисление организовано так, чтобы на каждом этапе локальный оператор применяется к клеткам из разных блоков. Первые синхронно-блочные КА были описаны в [5], среди которых хорошо исследована и широко применяется КА-диффузия Марголуса [18].

**Условие равноправия клеток:** на каждой  $t$ -й итерации локальный оператор  $\Theta(m)$  применяется ко всем клеткам  $(v, m) \in \Omega(t)$ , причем к каждой клетке только один раз.

Это условие гарантирует, что все клетке участвуют в процессе эволюции в равной степени. Нарушение этого условия может быть вызвано несовершенством генератора случайных чисел, если он не обеспечивает биномиального закона распределения вероятности, а также неправильной организацией взаимодействий между областями КА, если они расположены на разных процессорах. Очевидно, что синхронные КА всегда удовлетворяют условию равноправия, так как режим предусматривает детерминированный порядок применения локального оператора ко всем клеткам из  $\Omega(t)$ .

## 4 Многопроцессорная реализация КА-алгоритмов

Некоторый уже накопленный опыт клеточно-автоматного моделирования реальных процессов позволяет сделать вывод о том, что необходимые размеры и время моделирования требует выполнения вычислений на многопроцессорных системах. В настоящее время в распоряжении исследователей имеется два типа многопроцессорных систем: кластеры, объединяющие сотни и тысячи компьютеров, и многоядерные компьютеры с общей памятью, объединяющие десятки процессоров, которые затем тоже объединяются в кластеры. Распараллеливание КА выполняется путем разрезания клеточного массива на части, называемые доменами, размещения каждого домена на одном процессоре многопроцессорной системы и пересылок данных соседним доменам для выполнения межклеточных взаимодействий.

Пусть клеточный массив  $\Omega \in A \times$  имеет структуру параллелограмма,  $M = \{(i, j, k)\}$ , который разделен на  $n$  равных по количеству клеток доменов, размерами  $|Dom| = l_i \times l_j \times l_k$ . Тогда каждый домен содержит  $|Dom| = |M|/n$  клеток и имеет 2D соседей (при периодических граничных условиях). Известную формулу эффективности распараллеливания

$$Q(n) = \frac{t_{mono}}{n \cdot t_{dom} + t_{ex}}, \quad (7)$$

для КА моделирования удобно интерпретировать следующим образом:

-  $t_{mono}, t_{dom}$  - времена, необходимые для выполнения одной итерации на всех КА и на одном домене, соответственно, т.е.  $t_{mono} = \tau|M| = \tau \cdot \mu$ ,  $t_{dom} = \tau \cdot |Dom|$ ,  $\tau$  - время выполнения локального оператора в одной клетке.

-  $t_{ex}$  - время, затрачиваемое на межпроцессорные обмены за одну итерацию. Если таких обменов  $h$ , то  $t_{ex} = h(t_{lat} + V \cdot t_b)$ , где  $t_{lat}$  - время латентности (организации канала),  $t_b$  - время пересылки байта,  $V$  - объем пересылаемого пакета в байтах. Отсюда

$$Q(n) = \frac{\tau \cdot |Dom|}{\tau \cdot |Dom| + h(t_{lat} + V \cdot t_b)}. \quad (8)$$

Очевидно, что для получения высокой эффективности необходимо, чтобы

$$\tau \cdot |Dom| > h(t_{lat} + V \cdot t_b), \quad (9)$$

чтобы пересылки можно было выполнять на фоне вычислений.

В синхронных КА обмен данными производится после каждой итерации, т.е.  $h=1$ . Пакет формируется как клеточный массив, состоящий из пограничных клеток домена, толщина которого зависит от размеров шаблона  $T_S(m)$  (2). Чтобы условия корректности сохранились, необходимо, чтобы получаемые от соседних процессоров состояния клеток использовались только как переменные для вычисления функций переходов пограничной клетки, и не рассматривались бы как полноправные члены домена, к которым можно применить локальный оператор. Объем пакета  $V = c \cdot P$ , где  $P$  периметр (для 2D случая) и поверхность (для 3D случая), а константа  $c$  зависит от размеров  $T_S(m)$  (2) и от длины двоичного представления состояния клетки. Следовательно, (8) можно представить в виде

$$\tau \cdot |Dom| > (t_{lat} + c \cdot P \cdot f_b). \quad (10)$$

Соотношение (10) показывает, что при малых размерах доменов (больших  $n$ ), эффективность распараллеливания падает. Это происходит из-за увеличения доли потерь времени на межпроцессорные обмены, хотя, могут быть и технические причины. Например, влияние архитектуры коммуникационной сети. Разброс времени вычислений по доменам, сказывающийся на эффективности распараллеливания конечно-разностных алгоритмов, при КА-моделировании незначителен из-за простоты и однородности клеточный вычислений.

Приведенные оценки наглядно иллюстрируются экспериментально полученной зависимостью  $Q(n)$  при распараллеливании КА-алгоритмов трехмерного потока вязкой жидкости, приведенных в [19]. На рис.3<sup>1</sup> показаны зависимости  $Q(n)$ , полученные при размещении КА размерами  $2000 \times 2000 \times 100$  клеток на кластерах МВС-1000М и МВС-15000ВМ. Из рисунка легко видеть, что до значений  $n = 32$  (процессоры расположены в одной стойке) значение эффективности превышает 90% в обоих случаях. Далее эф-

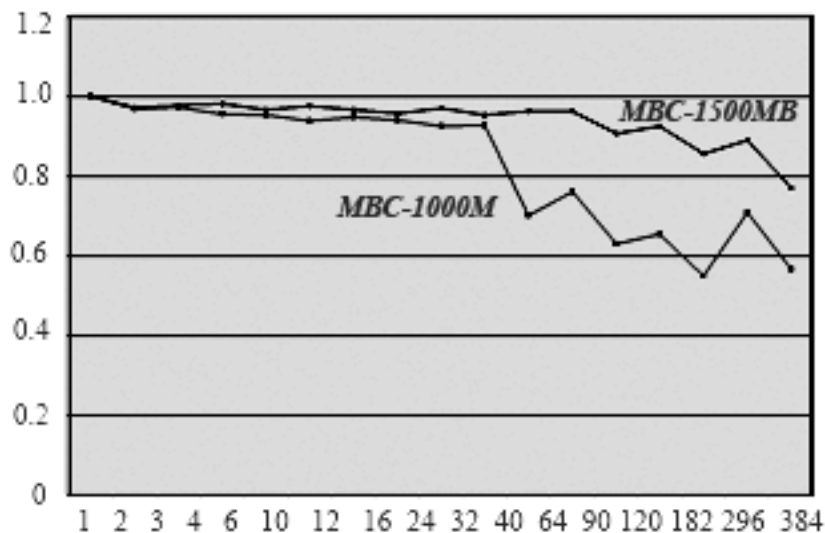


Рис. 3: Зависимость эффективности распараллеливания КА-алгоритма моделирования потока вязкой жидкости от числа процессоров (шкала числа процессоров неравномерная)

фективность для нижней кривой (кластер МСВ-1000) снижается за счет того, что используются сетевые переключатели, которые могли быть одновременно использованы и другими задачами (увеличивается  $t_{lat}$ .) В кластере МВС-15000ВМ влияние коммуникационного фактора не заметно. Однако, в обоих случаях при  $n > 200$  начинает сказываться влияние малого размера домена, так как правая и левая части (10) становятся сравнимыми.

<sup>1</sup>Рисунок составлен по данным работы [19] с разрешения автора



Для асинхронных КА межпроцессорный обмен должен выполняться всякий раз, когда изменяется состояние хотя бы одной клетки, находящейся на границе домена. Если это условие не выполняется, то не выполнится условие равноправия клеток, поскольку при стохастическом выборе клеток невозможно гарантировать, что новое состояние пограничной клетки  $(v, m_g)$ ,  $m_g \in Dom_g$ , не понадобится на следующем же такте для вычисления нового состояния клетки другого жомена  $(u, m_l)$ ,  $m_l \in Dom_l$ , если  $m_l = \phi_k^{-1}(m_g)$ ,  $\phi_k(m_g) \in T_S$ . Отсюда следует, что для асинхронных КА  $h = P$ ,  $V = c' \cdot t_b$ , где  $c'$  — число бит в двоичном представлении состояний, и соотношение (9) принимает вид

$$\tau \cdot |Dom| > P(t_{lat} + c' \cdot t_b). \quad (11)$$

Поскольку  $c'$  обычно не более двух байт и  $t_{lat}/t_b \approx 10^3$  (что имеет место в кластере МВС-1000М), можно принять  $t_{lat} \gg c' \cdot t_b$  и упростить условие высокой эффективности (11) до соотношения

$$\tau \cdot |Dom| > P \cdot t_{lat}, \quad (12)$$

которое при условии  $l_i = l_j = l_k = l$  и  $t_{lat}/\tau \approx 10^3$  превращается в следующие малопримлемые соотношения:  $l > 2 \cdot 10^3$  для двумерного КА и  $l > 6 \cdot 10^3$  — для трехмерного.

Существует еще одна проблема распараллеливания асинхронных КА — это настройка генераторов случайных чисел в процессорах, чтобы они имитировали работу единого генератора. Для этого можно следовать методу [20] или использовать простой но грубый способ получения разных последовательностей в разных процессорах, состоящий в том, что в каждом из них программе генератора задается свое исходное число.

## 5 Распараллеливание асинхронных КА

Поскольку, как показано в предыдущем разделе, асинхронный режим является препятствием для эффективного распараллеливания, возникает соблазн ввести некоторый порядок в вычислительный процесс, допустив тем самым частичную синхронизацию режима функционирования, при этом

- а) по возможности, сохранив эволюцию КА,
- б) не нарушив условий корректности.

Как показано в [21], это можно сделать, изменив асинхронный режим функционирования на блочно-синхронный. Следовательно, алгоритм распараллеливания асинхронного КА должен включать в себя преобразование  $\aleph_\alpha = \langle A, M, \Theta \rangle$  в  $\aleph_\beta = \langle A, M, \Theta \rangle$ . В результате, для распараллеливания  $\aleph_\alpha$  выполняется следующее.

- 1). На множестве имен  $M$  определяется шаблон, называемый *блоком*

$$B(m) = \{\phi_0(m), \phi_1(m), \dots, \phi_q(m), \phi_{q+1}(m), \dots, \phi_r(m)\},$$

который обладает следующими свойствами:

- а) блок включает в себя определяющий шаблон  $T(m)$  базовой локальной конфигурации  $S(m)$ ,

$$T(m) \subseteq B(m), \quad (13)$$

- б) блок индуцирует на множестве имен разбиение на *независимые подмножества*  $\Pi = \{M_1, \dots, M_r\}$ , для которых выполняются соотношения:

$$|M_k| = \frac{|M|}{r}, \quad \forall k \in \{1, \dots, r\}, \quad (14)$$

$$\bigcup_{k=1}^r M_k = M, \quad M_k \cap M_l = \emptyset \quad \forall (k, l) \in \{1, \dots, r\}, \quad (15)$$

$$\bigcup_{m \in M_k} B(m) = M, \quad B(m_g) \cap B(m_l) = \emptyset, \quad \forall (m_g, m_l) \in M_k. \quad (16)$$

Иными словами, вводится  $r$  покрытий множества имен одинаковыми блоками, причем каждый блок включает в себя базовую локальную конфигурацию  $S(m)$ .

2. Смена глобального состояния  $\Omega(t) \xrightarrow{\Theta(M)} \Omega(t+1)$  производится за  $r$  этапов:  $\theta_0, \theta_1, \dots, \theta_r$ . На каждом  $k$ -м этапе  $\theta_k$  за  $\lambda$  тактов ( $\Delta\theta = \lambda \cdot \tau$ ) производится применение  $\Theta(m)$  ко всем клеткам из одного случайно выбранного из  $\Pi$  независимого подмножества  $M_j$ ,  $j = 1, 2, \dots, r$ . Порядок выбора клеток при этом совершенно не имеет значения, так как функции переходов для них не пересекаются по переменным, что гарантируется условием (13). Следовательно, результат глобального перехода к следующему этапу  $\Omega(t + \theta_k) \xrightarrow{\Theta(M_k)} \Omega(t + \theta_{k+1})$  – единственный и одинаков как для синхронного так и для асинхронного режимов, т.е. множество возможных трасс  $\Gamma_\beta(\theta)$  для получения результата на каждом этапе имеет мощность

$$|\Gamma_\beta(\theta)| = \lambda!, \quad (17)$$

а каждая трасса  $\gamma(\theta_k)$  является фрагментом одной из трасс  $\gamma_\alpha \in \Gamma_\alpha$ .

3. Каждая трасса полной итерации является конкатенацией этапных трасс,  $\gamma_\beta(\Delta t) = \gamma_\beta(\theta_1)\gamma_\beta(\theta_2)\dots\gamma_\beta(\theta_r)$ . При случайном порядке обработки подмножеств  $M_j \in \Pi$  число таких конкатенаций равно  $r!$ , откуда общее количество возможных трасс в итерации

$$|\Gamma_\beta| = r! \cdot \lambda! = r! \left( \frac{|M|}{r} \right)!, \quad (18)$$

что существенно меньше, чем  $|\Gamma_\alpha| = |M|!$ . Однако, поскольку каждая  $\gamma_\beta$  имеет себе равную в  $\Gamma_\alpha$ , можно считать, что построенный по вышеприведенному алгоритму КА  $\aleph_\beta = \langle A, M, \Theta \rangle$  аппроксимирует  $\aleph_\alpha = \langle A, M, \Theta \rangle$ . Термин "аппроксимация" здесь имеет аналогичный общепринятому смысл приближенного равенства решений при равных исходных данных. В данном случае это понимается как приближенное равенство эволюций при равных начальных клеточных массивах. Косвенную оценку этого приближения можно сделать по отношению  $|\Gamma_\beta|/|\Gamma_\alpha|$ . Чем оно больше, тем выше точность аппроксимации, т.к. тем больше глобальных переходов в эволюции  $\Sigma_\beta(\Omega)$  точно совпадут с переходами  $\Sigma_\alpha(\Omega)$ . Точность аппроксимации зависит также и от функций перехода (4) в  $\aleph_\alpha$ . Если они таковы, что все трассы приводят к одному результату,  $|\Sigma_\alpha(\Omega)| = 1$ , то для любого  $\Omega \in A \times M$  аппроксимация имеет абсолютную точность.

**Пример 2.** Упрощенная КА модель эпитаксиального роста кремниевого кристалла [11] отображает два элементарных действия: 1) адсорбцию Si-атомов из газа с вероятностью  $p_a$ ; 2) диффузию адсорбированных атомов по поверхности кристалла с вероятностью  $p_d$ . Поверхность разделена на квадратные площадки ((клетки) так, что в каждой клетке может поместиться один атом или столбик из  $a$  атомов. Атом диффундирует в одну из  $n$  соседних клеток ( $n \in \{1, 2, 3, 4\}$ ), если там столбик ниже, чем в ней самой. Это происходит с вероятностью  $p' = 0.05^{4-n}$ , причем одно из  $n$  возможных направлений выбирается равновероятно, т.е.  $p_d = p'/n$ . Процесс моделируется КА  $\aleph_\alpha = \langle A, M, \Theta \rangle$ , где  $A$  – множество целых чисел,  $M = \{(i, j) : i, j = 0, \dots, N\}$ . Определяющий шаблон базовой конфигурации  $T_S(i, j) = \{(i, j), (i-1, j), (i, j+1), (i+1, j), (i, j-1)\}$ . Клетка  $(a, (i, j))$  соответствует площадке на кристалле, над которой толщина адсорбированного слоя равна  $a$  атомов. Локальный оператор  $\Theta(i, j)$  равен суперпозиции  $\Theta_a(i, j)$  (адсорбция) и  $\Theta_d(i, j)$  (диффузия), где

$$\begin{aligned} \Theta_a(i, j) &: \{(v_0, (i, j))\} \xrightarrow{p_a} \{(v_0 + 1, (i, j))\}, \\ \Theta_d(i, j) &: \{(v_0, (i, j)), (v_k, \phi_k(i, j))\} \xrightarrow{p_d} \{(v_0 - 1, (i, j)), (v_k + 1, \phi_k(i, j))\}. \end{aligned}$$

Одной из важных характеристик процесса эпитаксиального роста является динамика формирования островков из адсорбированных на поверхности атомов, которая оценивается колебаниями их суммарного периметра  $P(t)$ . На рис.4 показаны зависимости  $P_\alpha(t)$  и  $P_\beta(t)$ , полученные путем использования КА-моделей  $\aleph_\alpha = \langle A, M, \Theta \rangle$  (рис.4 а) и

$\aleph_\beta = \langle A, M, \Theta \rangle$  (рис.4 б) при размерах клеточного массива  $300 \times 300$ ,  $p_a = 0.2$ . Средняя квадратичная разность между этими экспериментально полученными зависимостями равна  $E = 0.012$ .

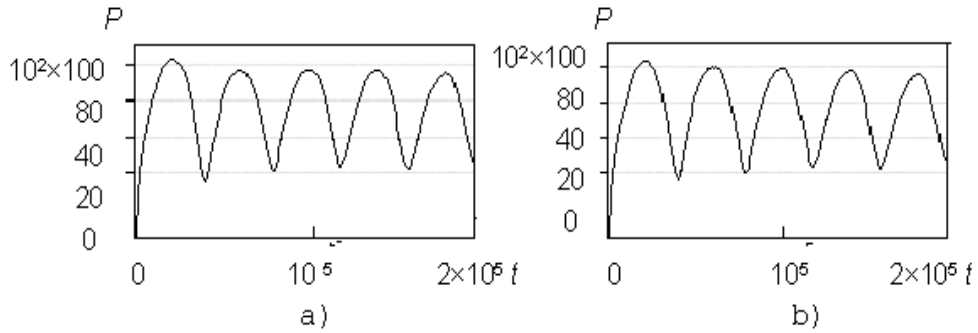


Рис. 4: Зависимости суммарного периметра островков из атомов кремния от количества итераций при моделировании эпитаксиального роста кристалла : а) асинхронным КА , б) блочно-синхронным КА

Поскольку при блочно-синхронном режиме на каждой стадии функции переходов разделены по переменным, т.е. обмен данными между доменами можно производить после завершения вычислений на каждом этапе  $\theta_k$ . При этом ни одно условие корректности не нарушается, следовательно, допускается формирование пакета в котором содержатся новые состояния клеток из обрабатываемого на этой стадии независимого подмножества. Объем пакета  $V = c \cdot P_{dom}/r$ , где  $P_{dom}$  - периметр домена,  $r = |B(m)|$ . С учетом того, что этапы обрабатываются последовательно, условие эффективности распараллеливания (11) принимает вид

$$\tau \cdot |Dom| > r \cdot \left( t_{lat} + c \cdot t_b \frac{P_{dom}}{r} \right). \quad (19)$$

Из (20) следует, что чем больше размер блока  $r$ , тем скорее начнется деградация эффективности распараллеливания. Следовательно, блок надо выбирать минимального размера, но так, чтобы удовлетворялись условия (13) и (16). Результаты экспериментального моделирования процесса эпитаксиального роста с помощью блочно-синхронного КА размером  $6000 \times 6000$ , аппроксимирующего  $\aleph_\alpha$  из примера 2 на кластере МСВ-1000 при различных количествах процессоров приведены в табл. 1.

**Таблица**

Время вычислений  $T(n)$ (мин) на  $n$  процессорах, ускорение  $Sp = T(n)/T(1)$  и эффективность  $Q(n) = 1/Sp(n)$  распараллеливания при выполнении  $10^4$  итераций КА  $\aleph_\beta$  с размерами  $6000 \times 6000$  клеток.

$n$	1	4	9	16	25
$T(n)$	817,08	212,91	113,66	55,52	41,65
$Sp(n)$	1	3,83	7,18	14,78	19,64
$Q(n)$	1	0,95	0,79	0,92	0,78

Данные таблицы свидетельствуют о двух фактах: 1) эффективность распараллеливания синхронно-блочного КА вполне приемлема, 2) влияние архитектуры коммутационной сети существенна, что показывает разница в значениях эффективности для четного и нечетного количества процессоров, как следствие того что кластер МСВ-1000М состоит из двухпроцессорных модулей. Однако, реальное моделирование процессов типа "эпитаксиальный рост кристалла" требует на 2-3 порядка больших размеров КА и на 2-3 порядка большего времени моделирования, чем в приведенном примере. Поэтому, распараллеливание необходимо проводить на сотни и тысячи процессоров.

## 6 Заключение.

Из приведенного материала можно сделать следующие выводы. 1) Существует широкий класс больших задач моделирования пространственной динамики, которые можно решать только с использованием КА-моделей.

2) Вследствие внутреннего параллелизма и локальности взаимодействий КА-алгоритмы легко и эффективно распараллеливаются.

3) Для синхронных КА это не требует никаких изменений алгоритма. Для асинхронных достижение высокой эффективности возможно при использовании предложенного метода их аппроксимации синхронно-блочными КА.

## Список литературы

- [1] von Neumann J. Theory of self reproducing automata - University of Illinois. USA. 1966.
- [2] Toffoli T. Cellular Automata as an Alternative to (rather than Approximation of) Differential Equations in Modeling Physics // Physica D. - Vol.10 - 1884 - 117-127.
- [3] Wolfram S. Statistical mechanics of Cellular Automata // Review of Modern Physics. -Vol.55. - 1993. - 607-640.
- [4] Бандман О.Л. Клеточно-автоматные модели пространственной динамики. - в кн. Системная информатикаю - вып. 10. - 2006. - 58-16.
- [5] Toffoli T., Margolus N. Cellular Automata Machines - USA: MIT Press. - 1987.
- [6] Madore B. Freedman W. Computer simulations of the Belousov-Zhabotinsky reaction. - Science. - vol. 222. - 1983. - 615-619.
- [7] Packard N. Lattice models for solidification and aggregation. - In: Theory and application of Cellular Automata (ed. S.Wolfram). - Singapore: World Scientific - 1986 - 305-310.
- [8] Wolfram S. A new kind of science - Wolfram Media Inc., Champaign, Ill.,USA. - 2002.
- [9] Rothman B.H., Zaleski S. Lattice-Gas Cellular Automata. Simple Models of Complex Hydrodynamics - London: Cambridge Univ. Press - 1997.
- [10] Elokhin V.i, Latkin E.I., Matveev A.V., Gorodetskii V.V. Application of Statistical Lattice Models to the Analysis of Oscillatory and Autowave Processes in the Reaction of Carbom Monoxide Oxidation over Platinum and Palladium Surfaces . -Kinetics and Catalysis. - Vol.44, No 5. - 2003 - 692-700.
- [11] Neizvestny I.G., Shwartz N.L., Yanovitskaya Z.Sh. 3D-model of epitazial growth on porous 111 and 100 Si surfaces. - Computer Physics Communications. - Vol .147. - 2002. - 272-275.
- [12] Chen N., Glazier J.A, Alber M.A. A Parallel Implementation of the Cellular Potts Model for Simulation of Cell-Based Morphogenesis // Lecture Notes in Computer Science . - Vol. 4173. - Berlin: Springer. - 2006. - 58-67.
- [13] Bandman O.L. Synchronous versus Asynchronous Cellular Automata for Simulating Nano-System Kinetics. - Bullrtin of Novosibirsk Computer Center, series Computer Science. - issue 23. - 2006. - 1-12.
- [14] Achasova S., Bandman O., Markova V., Piskunov S. Parallel Substitution Algorithm. Theory and Application - Singapore: World Scientific. - 1994
- [15] Park J.K., Steiglitz K., ThurstonW.P. Soliton-like behavior in automata // Physica D. - Vol.19. -1986.- 423-432.

- [16] Ачасова С.М., Бандман О.Л. Корректность параллельных процессов. Новосибирск: Наука. - 1999.
- [17] Golze U. (A)synchronous (N0n)deterministic Cell Spaces Simulating Each Other // Journal of Computer and System Science. - Vol.17. - 1978. -176-188.
- [18] Малинецкий Г.Г., Степанцов М.Е. Моделирование диффузионных процессов клеточными автоматами с окрестностью Марголдуса. // Журнал вычислительной математики и мат. Физики. - Том 36, № 6. - 1998. - 1017-1021.
- [19] Медведев Ю.Г. Исследование вычислительных характеристик программной реализации трехмерной клеточно-автоматной модели вязкой жидкости // Труды Всероссийской научной конференции "Научный сервис в сети ИНТЕРНЕТ: технологии параллельного программирования. - Новороссийск. - 2006 -79-85.
- [20] Марченко М.А. Комплекс программ MONC для распределенных вычислений методом Монте-Карло //Сибирский журнал вычислительной математики. Том 7, № 1. - 2004. - 43-56.
- [21] Bandman O. Parallel Simulation of Asynchronous Cellular Automata Evolution // Lecture Notes in Computer Science . - Vol. 4173. - Berlin: Springer. - 2006. - 41-48.