

СИНХРОНИЗАЦИЯ ПРОЦЕССОВ ПРИ ПАРАЛЛЕЛЬНОМ МОДЕЛИРОВАНИИ ДИСКРЕТНЫХ СОБЫТИЙ

Л. Н. Щур^{a,b*}, Л. Ф. Зиганурова^{b,c**}

^a *Институт теоретической физики им. Л. Д. Ландау Российской академии наук
142432, Черногловка, Московская обл., Россия*

^b *Национальный исследовательский университет Высшая школа экономики
101000, Москва, Россия*

^c *Научный центр Российской академии наук в Черногловке
142432, Черногловка, Московская обл., Россия*

Поступила в редакцию 22 апреля 2019 г.,
после переработки 22 апреля 2019 г.
Принята к публикации 23 апреля 2019 г.

Метод параллельного моделирования дискретных событий (ПМДС) является одним из перспективных методов эффективного использования суперкомпьютерных систем. В работе приводятся результаты анализа моделей синхронизации процессорных элементов при выполнении вычислений методом ПМДС. Для классификации моделей синхронизации используется аналогия между изменением профиля локальных времен процессорных элементов и ростом поверхности образца при молекулярной эпитаксии. Два важных класса таких моделей — это модели консервативного ПМДС и оптимистического ПМДС. Консервативная модель принадлежит к классу универсальности уравнения Кардара–Паризи–Жанга, а оптимистическая модель — к классу универсальности задач направленного протекания. Мы приводим результаты анализа моделей синхронизации в случае, если граф обмена сообщениями между процессорными элементами принадлежит к классу сетей малого мира.

Статья для специального выпуска ЖЭТФ, посвященного 100-летию И. М. Халатникова

DOI: 10.1134/S0044451019100201

1. ВВЕДЕНИЕ

Начало развитию вычислительной физики положили научно-практические работы по разработке ракетно-ядерного щита. С этой целью были созданы ЭВМ в США и в СССР. Основной круг решаемых задач находился в области гидродинамики экстремального состояния вещества. В процессе работ над такими вычислениями была предложена очень красивая и эффективная схема решения уравнений гиперболического и параболического типов. Это было сделано Л. Д. Ландау и И. М. Халатниковым в соавторстве с математиком Н. Н. Мейманом [1]. Ландау и Халатников разработали программу развития методов вычислений, в которую были вклю-

чены возможные интересные физические приложения, они понимали перспективность нового направления исследований, основанных на использовании ЭВМ [2], но известные трагические события прервали их совместную работу. Предложенная ими схема бегущего счета, известная как «метод прогонки», вошла в учебники, она широко используется специалистами по вычислительной физике во всем мире, хотя большинство из них и не знает о существовании основополагающей работы [1], вошедшей во второй том Собрания трудов Ландау. В этом мы имели возможность убедиться на прошедшей в 2013 г. в Москве международной конференции по вычислительной физике IUPAP CCP2013 [3], на которой И. М. Халатников впервые представил доклад по материалам этой работы шестидесятилетней давности.

Первые вычислительные системы были скалярными и проводили в единицу времени одно вычисление. В настоящее время процесс развития вычис-

* E-mail: lev@landau.ac.ru

** E-mail: ziganurova@gmail.com

лительных технологий находится на очередном витке, которое состоит в переходе на иерархическую структуру вычислительных устройств. Если центральный процессор ранних ЭВМ состоял из одного арифметически-логического устройства, то современный процессор имеет несколько таких устройств в одном процессорном элементе, называемом ядром. Более того, процессор может иметь до нескольких десятков таких ядер. Каждое ядро может выполнять свой независимый процесс вычислений. Для эффективного использования в вычислениях ЭВМ с такой конструкцией необходим особый подход к организации вычислений.

ЭВМ с иерархической структурой можно использовать как большой набор традиционных ЭВМ, каждая из которых выполняет свою задачу на одном из ядер. При этом процесс выполнения задач каждым ядром не будет зависеть от процессов, выполняемых другими ядрами. Если же требуется использовать всю мощность современной супер-ЭВМ для решения одной большой задачи, то нам надо составить процесс вычислений таким образом, чтобы синхронизировать работу большого числа процессов вычислений, выполняемых всеми ядрами ЭВМ. Современные суперкомпьютерные системы насчитывают несколько миллионов вычислительных ядер¹⁾ и обеспечить их синхронную работу таким образом, чтобы минимизировать простои каждого из ядер, является актуальной и в общем случае нерешенной задачей.

Один из методов загрузки суперкомпьютера на полную мощность одной задачей (а это и есть основное предназначение супер-ЭВМ) — это метод параллельного моделирования дискретных событий [5]. Его назначение состоит в синхронизации вычислений одной задачи, которая предварительно разбивается на части (логические процессы), каждая из которых выполняется отдельным процессорным элементом (например, ядром). В случае решения упомянутых выше дифференциальных уравнений область вычислений может быть разбита на подобласти и каждый логический процесс будет проводить

вычисления в одной из подобластей. Свою часть вычислений логические процессы могут проводить независимо друг от друга, а взаимодействие логических процессов происходит только на границах областей с целью обмена обновленными значениями сеточных переменных.

Первые принципы параллельного моделирования дискретных событий были описаны Любачевским в 1987 г. [6] на примере модели Изинга. Главная особенность метода состоит в том, что он не изменяет динамику физической системы при моделировании [7]. Например, в спиновых системах перевороты спинов происходят в случайные моменты времени в разных узлах системы. Если моделировать спиновую систему традиционным синхронным способом, то расчеты переворота спина будут происходить в моменты времени $0, \Delta t, 2\Delta t, \dots$, где Δt — заранее заданный временной шаг моделирования. Метод параллельного моделирования дискретных событий (ПМДС) позволяет моделировать асинхронную динамику системы: время при моделировании течет непрерывно, но изменения в частях системы (т. е. перевороты спинов) происходят в случайные моменты времени. Эти изменения называются событиями, и времена между событиями распределены по закону Пуассона. Параллельное моделирование таких дискретных событий при правильной синхронизации множества логических процессов дает ту же последовательность событий, что и при последовательном моделировании, но процесс вычислений можно значительно ускорить за счет параллелизма вычислений ядрами.

Локальное время [8] каждого логического процесса изменяется при наступлении события в моделируемой этим процессом части общей системы, поэтому времена всех логических процессов в системе различны. Локальные виртуальные времена процессов составляют профиль локальных виртуальных времен (ЛВВ). Виртуальное время процесса — это неубывающая локальная переменная логического процесса, и профиль таких времен может быть использован для характеристики степени синхронизации параллельных вычислений [5].

В процессе моделирования профиль локальных времен растет, что делает его схожим с процессом случайного роста поверхности в физике [9, 10]. Рост локальных времен в консервативном алгоритме в одномерном случае имеет аналогию с ростом поверхности, описываемым уравнением Кардара – Паризи – Жанга (KPZ [11]), а в оптимистическом алгоритме — с ростом поверхности в модели SOS (Solid-on-Solid [12]). Аналогия между эволюцией про-

¹⁾ Мы не углубляемся в основном тексте статьи в реальную схему современных суперкомпьютеров, которые дополнительно к центральным процессорам имеют устройства для выполнения вычислений, так называемые ускорители вычислений, например, графические процессоры. Каждый такой графический процессор имеет свою иерархическую структуру, в которой насчитываются тысячи ядер, и которые имеют иную схему организации вычислений, чем ядра классических центральных процессоров [4]. Общее число таких ядер в современных суперкомпьютерах намного превышает общее число ядер центральных процессоров.

филя локальных виртуальных времен в консервативном алгоритме и ростом поверхности по уравнению KPZ дает возможность классифицировать алгоритмы синхронизации на три класса: консервативный, оптимистический и FaS. Эти классы соответствуют периодическим, свободным и фиксированным граничным условиям уравнения Кардара – Паризи – Жанга [13].

Сделать выводы о фундаментальных свойствах этих алгоритмов возможно, исследуя поведение локальных виртуальных времен процессов. Для этого мы используем модели роста профиля локальных времен для оптимистического [10] и консервативного [9] алгоритмов синхронизации. Эти модели учитывают такие свойства реальных систем, как количество логических процессов, граф их взаимодействия, среднее время между событиями и другие. Мы исследовали поведение моделей на графах двух типов: регулярных графах и графах с топологией малого мира. Регулярные графы содержат только локальные взаимодействия между узлами, в то время как сети малого мира учитывают также и удаленные взаимодействия.

В статье мы описываем результаты моделирования профиля локальных виртуальных времен для консервативного и оптимистического алгоритмов ПМДС и приводим основания для соотнесения моделей с известными классами универсальности.

2. МОДЕЛЬ РОСТА ПРОФИЛЯ ЛОКАЛЬНЫХ ВРЕМЕН В АЛГОРИТМАХ СИНХРОНИЗАЦИИ ПМДС

В этом разделе мы опишем модели роста профиля ЛВВ для консервативного и оптимистического алгоритмов ПМДС [9, 10, 14, 15]. Отметим, что существует множество различных реализаций алгоритмов обоих классов. Мы не учитываем частные особенности конкретных реализаций, а фокусируемся на поведении профиля локальных времен. Поведение профиля содержит информацию о степени синхронизации процессорных элементов и об эффективности использования процессорного времени [16, 17].

Степень синхронизации процессорных элементов и эффективность параллельного моделирования зависят не только от выбранного алгоритма синхронизации, но и от топологии взаимодействия логических процессов. При моделировании реальных физических процессов чаще всего элементы системы взаимодействуют локально, но в некоторых зада-

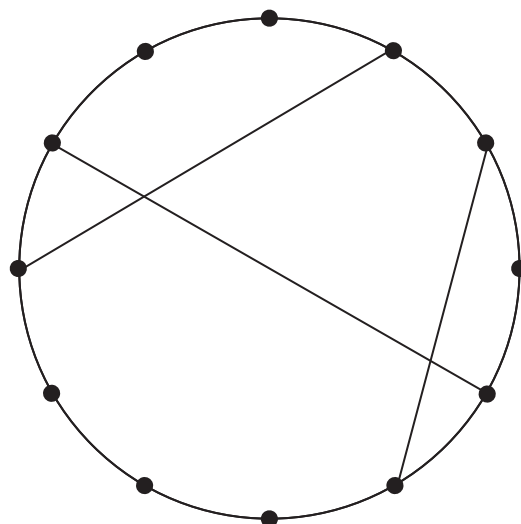


Рис. 1. Пример топологии малого мира

чам иногда могут возникать и связи дальнего порядка [18], поэтому в моделях мы организуем логические процессы как в регулярный граф, так и в граф малого мира [19]. В этих графах вершинами являются логические процессы (ЛП), а ребрами — взаимодействия между ними. Если два ЛП не соединены ребром, это означает, что они независимы друг от друга и синхронизация между ними не требуется. Мы описываем коммуникационный граф матрицей смежности D . Если ЛП $_i$ зависит от ЛП $_j$, то элемент матрицы $D(i, j) = 1$, если процессы ЛП $_i$ и ЛП $_j$ независимы, то $D(i, j) = 0$.

Регулярная топология представляет собой кольцо, где каждый ЛП взаимодействует только с двумя соседними ЛП. Топология малого мира получается из регулярного графа путем добавления дальних связей между двумя случайно выбранными логическими процессами (рис. 1). Количество таких дальних связей в наших моделях регулируется параметром $p \in [0; 1]$, фиксированным числом дальних связей, равным pN . Случай $p = 0$ соответствует регулярной топологии.

При топологии малого мира средний кратчайший путь $l(N, p)$ в графе растет логарифмически с количеством вершин N [20]:

$$l(N, p) = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij} \propto \ln(N), \quad (1)$$

где расстояние d_{ij} измеряется как минимальное количество вершин, через которые нужно пройти, чтобы попасть из вершины i в вершину j (также для d_{ij} используется название chemical distance). Напомним, что на регулярном графе средний кратчайший

путь растет линейно с N . Соотношение (1) выполняется в тех графах, которые мы строим и используем при моделировании на сетях малого мира. Мы численно проанализировали зависимость среднего кратчайшего пути в этих графах от размера системы и параметра p . Результаты численных экспериментов хорошо аппроксимируются функцией [14]

$$l(p) = A \frac{\ln(pN)}{p} + C, \quad (2)$$

где A, C — константы.

Взаимодействие между ЛПП реализуется с помощью обмена сообщениями (messages). При наступлении события (т. е. при изменении состояния в моделируемой логическом процессом подобласти) логический процесс создает сообщение с указанием сущности события и времени события (ЛВВ), измеренного в единицах глобального времени моделируемого события [8]. Сравнение времени в полученном сообщении с текущим локальным временем дает возможность каждому ЛПП делать вывод о сохранении причинности в процессе моделирования. Если полученное логическим процессом сообщение содержит время меньшее, чем локальное время процесса, то это означает нарушение причинности. Способ обработки нарушения причинности и отличает классы алгоритмов [13, 21].

2.1. Модель для консервативного алгоритма

Консервативный алгоритм полностью избегает ошибки причинности, т. е. такие ситуации, при которых логический процесс получает сообщение «из прошлого» — с временной меткой меньшей, чем его текущее локальное время. Это осуществляется путем обмена сообщениями, содержащими времена наступления событий в логических процессах [5, 22]. Говоря на языке локальных времен, в консервативном алгоритме обработка событий логическим процессом будет безопасна, если его локальное время меньше или равно локальному времени соседей, а также других ЛПП, от которых он зависит в соответствии с графом взаимодействия. Если это условие не выполняется, то ЛПП остается заблокированным и ждет, пока соседние ЛПП не «догонят» его во времени.

Рассмотрим систему из N логических процессов, объединенных в коммуникационную топологию. Пусть p — средняя доля дальних связей на один ЛПП, тогда общее число дальних связей равно pN .

Обозначим локальное время одного логического процесса на шаге моделирования t как $\tau_i(t)$, тогда профиль ЛВВ — это набор $\{\tau_1(t), \tau_2(t), \dots, \tau_N(t)\}$.

Мы начинаем моделирование эволюции ЛВВ с плоского профиля: $\tau_i(t = 0) = 0, i = 1, 2, \dots, N$. Далее увеличиваем локальное время только у тех процессов, чье время меньше или равно локальному времени соседей. Если его время больше, чем у его соседей, то на данном шаге моделирования его локальное время не изменяется. Учитывая, что события в системе распределены экспоненциально, мы обновляем профиль локальных времен на каждом шаге моделирования t согласно следующему правилу:

$$\tau_i(t + 1) = \begin{cases} \tau_i(t) + \eta_i, & \tau_i(t) \leq \{\tau_j(t)\}_{D(i,j)=1}, \\ \tau_i(t), & \tau_i(t) > \{\tau_j(t)\}_{D(i,j)=1}, \end{cases} \quad (3)$$

где η_i — это случайная величина, распределенная по закону Пуассона с единичным математическим ожиданием, $\{\tau_j(t)\}_{D(i,j)=1}$ — это локальные времена логических процессов, которые соединены с ЛПП _{i} локальной либо дальней связью, $i = 1, \dots, N$.

2.2. Модель для оптимистического алгоритма

Идея оптимистического алгоритма заключается в том, что вместо блокирующих конструкций (как в консервативных алгоритмах) используется механизм обнаружения и исправления ошибок [23]. ЛПП параллельно обрабатывают события в течение некоторого установленного временного окна, а затем происходит проверка причинности вычислений. Если ЛПП получил сообщение с меткой меньшей, чем его ЛВВ, то состояние этого ЛПП «откатывается» до того времени, когда получение этого события было бы безопасным. Если при этом ЛПП отправлял сообщения другим ЛПП, то для отмены этих преждевременно отправленных сообщений рассылаются так называемые антисообщения. Антисообщения отличаются от обычных сообщений только наличием отрицательного знака. При попадании в одну очередь двух одинаковых сообщений с разными знаками, происходит аннигиляция, оба сообщения уничтожаются.

Поскольку в оптимистическом алгоритме локальное время логических процессов может как увеличиваться, так и уменьшаться, мы разбиваем один шаг моделирования на две части: продвижение вперед и откат назад. Во время продвижения вперед каждый ЛПП _{i} увеличивает свое ЛВВ на случайную

величину η_i , распределенную по закону Пуассона со средним значением, равным единице:

$$\tau_i(t+1) = \tau_i(t) + \eta_i, \quad i = 1, 2, \dots, N. \quad (4)$$

Откат происходит в том случае, когда нарушена причинность вычислений. Будем считать, что количество откатов во время одного шага моделирования — это пуассоновская случайная величина со средним b . Во время каждого отката случайным образом выбирается один ЛПП $_j$. Считаем, что причинность может быть нарушена равновероятно любым ЛПП, от которого зависит j -й ЛП. Выбираем равновероятно ЛПП $_r$ из числа тех, для которых $D(j, r) = 1$. Далее τ_j сравнивается со временем τ_r и в случае, если τ_j больше ЛВВ ЛПП $_r$, τ_j уменьшается до τ_r [10]:

$$\tau_j = \begin{cases} \tau_r, & \tau_j > \tau_r, \\ \tau_j, & \tau_j < \tau_r. \end{cases} \quad (5)$$

2.3. Наблюдаемые величины

По описанным выше правилам мы моделируем рост профиля локальных времен в течение $t = 10^4$ – 10^6 дискретных шагов моделирования. После каждого шага t , т. е. после каждого обновления профиля, мы рассчитываем среднее значение локального времени по всем логическим процессам $\tau(t)$, среднюю скорость профиля $v(t)$ и среднюю квадратичную ширину профиля $w^2(t)$.

Среднее локальное виртуальное время иногда называют средней высотой профиля. Оно представляет собой среднее арифметическое по локальным временам всех логических процессов:

$$\tau(t) = \left\langle \frac{1}{N} \sum_{i=1}^N \tau_i(t) \right\rangle. \quad (6)$$

Угловые скобки в этом уравнении и далее означают усреднение по независимым реализациям профиля на шаге моделирования t .

Средняя скорость профиля показывает загруженность процессорных элементов, или эффективность работы алгоритма. Для консервативного алгоритма средняя скорость рассчитывается по формуле

$$v(t) = \left\langle \frac{N_{\text{акт}}(t)}{N} \right\rangle, \quad (7)$$

где $N_{\text{акт}}(t)$ — это количество активных ЛПП, т. е. тех ЛПП, которые изменили свое время на данном шаге моделирования.

Для оптимистического алгоритма средняя скорость также отображает эффективность обработки событий, но рассчитывается по другой формуле:

$$v(t) = \langle \tau(t) - \tau(t-1) \rangle, \quad (8)$$

где $\tau(t)$ — это среднее по логическим процессам время на шаге t , а $\tau(t-1)$ — среднее по логическим процессам время на предыдущем шаге. В случае, когда в модели оптимистического алгоритма нет откатов, т. е. параметр $b = 0$, средняя скорость будет равна единице.

Средняя квадратичная ширина профиля отражает степень синхронизации процессорных элементов и рассчитывается по формуле

$$w^2(t) = \left\langle \frac{1}{N} \sum_{i=1}^N [\tau_i(t) - \tau(t)]^2 \right\rangle. \quad (9)$$

Чем меньше разброс локальных времен, тем лучше ЛПП синхронизированы. Большая дисперсия локального времени процессов указывает на то, что часть ЛПП отстает, а часть — слишком торопится. Это увеличивает время ожидания для тех ЛПП, которые находятся впереди по времени.

Мы изучаем изменение средней скорости профиля локальных времен v и средней квадратичной ширины профиля локальных времен w^2 как функций шага моделирования t , количества логических процессов N и концентрации дальних связей между логическими процессами p .

3. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

В этом разделе мы описываем поведение модели для консервативного алгоритма синхронизации и модели для оптимистического алгоритма синхронизации. Акцент сделан на аналогии моделей динамики локальных времен с моделями роста поверхности в физике. Таким образом, исследуемые нами свойства профиля ЛВВ в алгоритмах ПМДС могут быть интерпретированы как в терминах степени синхронизации и эффективности этих алгоритмов, так и в терминах случайного роста поверхности.

3.1. Консервативный алгоритм

3.1.1. Класс универсальности Кардара – Паризи – Жанга

Эволюцию профиля ЛВВ в консервативном алгоритме ПМДС на одномерной топологии можно

относит к классу универсальности Кардара–Паризи–Жанга (KPZ) [9]. Класс универсальности KPZ описывается уравнением [11]

$$\frac{\partial}{\partial t} h(x, t) = \nu_0 + \nu \nabla^2 h + \frac{\lambda}{2} (\nabla h)^2 + \eta(x, t), \quad (10)$$

где $h(x, t)$ — это высота поверхности в точке x в момент времени t , а $\eta(x, t)$ — это гауссовский шум (дельта-коррелированный), который задается уравнением

$$\begin{aligned} \langle \eta(x, t) \rangle &= 0, \\ \langle \eta(x, t) \eta(x', t') \rangle &= D \delta(x - x') \delta(t - t'). \end{aligned} \quad (11)$$

Угловые скобки означают усреднение по ансамблю, ν_0, ν, λ, D — константы.

Первый член уравнения ν_0 — это постоянная действующая сила. Второй член уравнения соответствует поверхностному натяжению и отвечает за сглаживание поверхности. Нелинейность возникает из геометрических соображений: поверхность растет в каждой точке перпендикулярно краю, а высота $h(x, t)$ измеряется вдоль оси y . Это лишь один из источников нелинейности, среди прочих причин может быть также нелинейное взаимодействие частиц, из которых состоит поверхность.

Известно, что квадратичная ширина поверхности $w^2(L, t)$ масштабируется как [24]

$$w^2(L, t) \propto t^{2\beta} F(Lt^{-1/z}) \propto \begin{cases} N^{2\alpha}, & L \ll \xi(t), \\ t^{2\beta}, & L \gg \xi(t), \end{cases} \quad (12)$$

где L — линейный размер системы, $\xi(t) \propto t^{1/z}$ — корреляционная длина, $z = \alpha/\beta$. Критический показатель α называется показателем роста, β — показателем шероховатости, z — динамическим критическим показателем. В одномерном случае для класса универсальности KPZ известны точные значения критических показателей: $\alpha = 1/2$, $\beta = 1/3$, $z = 3/2$.

3.1.2. Результаты численных экспериментов

В модели консервативного алгоритма на регулярной топологии квадратичная ширина профиля растет со временем логарифмически:

$$w^2(t) \propto t^{2\beta}, \quad (13)$$

а затем насыщается (рис. 2). Величина насыщения (максимальная ширина профиля ЛВВ) зависит от количества логических процессов линейно (рис. 3):

$$w_\infty^2(N) \propto N^{2\alpha}. \quad (14)$$

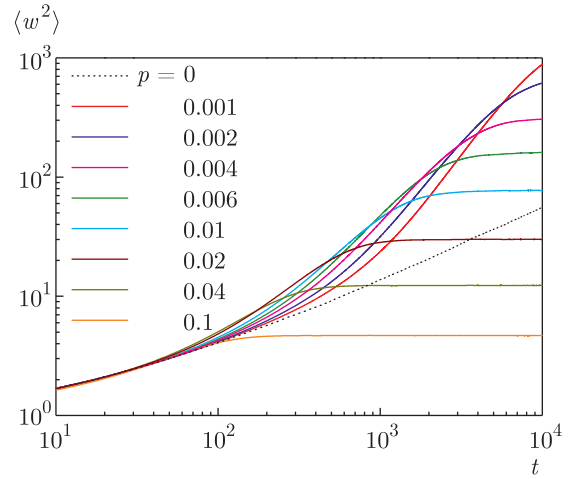


Рис. 2. Средняя квадратичная ширина w^2 как функция времени для системы из $N = 10^4$ логических процессов и для различных значений p . Усреднение взято по 1500 независимым реализациям. Пунктиром обозначена кривая для регулярной топологии $p = 0$. Из статьи [14]

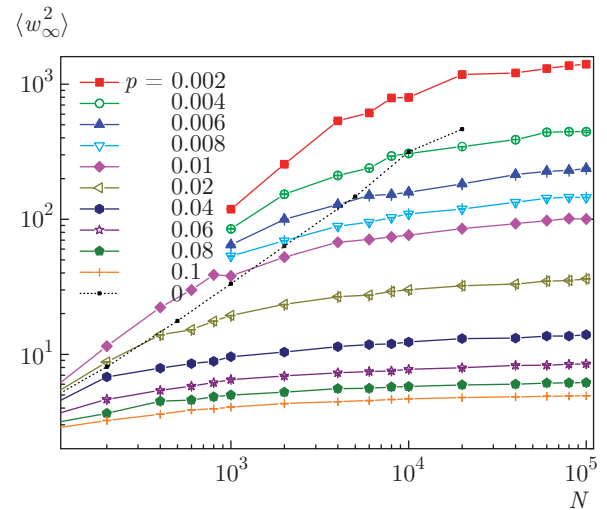


Рис. 3. Зависимость средней величины насыщения w_∞^2 от количества N логических процессов при разных p [14]

Значения показателей α и β , полученные путем аппроксимации данных, равны соответственно 0.49(1) и 0.326(5). Значения этих показателей близки к критическим индексам класса KPZ.

Средняя скорость роста профиля ЛВВ в пределе бесконечного размера системы ν_0 равна 0.246410(7) [13]. Из этого следует, что при использовании консервативной синхронизации с локальными взаимодействиями только четверть логических процессов активна в один момент времени. Степень рассинхронизации между логическими процессами увеличивается пропорционально их числу ($w^2 \propto N$).

Добавление дальних взаимодействий между логическими процессами кардинально меняет поведение профиля ЛВВ в консервативном алгоритме [14, 15]. Синхронизация между ЛП в модели консервативного алгоритма становится лучше на топологии малого мира. Ширина профиля растет со временем как $t^{2\beta}$ и насыщается (рис. 2), как и в случае локальной топологии, однако величина насыщения не увеличивается с ростом количества ЛП (рис. 3). Критический показатель шероховатости α в этом случае равен 0. Это означает, что степень синхронизации остается одинаковой для систем любого размера.

Средняя скорость профиля на топологии малого мира незначительно снижается. Так, например, при значении параметра $p = 0.01$ (p — доля случайных добавленных дальних связей) средняя скорость равна 0.221370(7). Обозначим разность между скоростью роста профиля на регулярной топологии v_0 и на топологии малого мира v как Δv [14],

$$\Delta v = v_0 - v.$$

Эта разность изменяется как

$$\Delta v(p, N) \propto p^{B(N)}. \tag{15}$$

Степень $B(N)$ уменьшается с ростом количества логических процессов:

$$B(N) \approx B + A \frac{\ln N}{\sqrt{N}}, \tag{16}$$

асимптотически она равна $B = 0.306(4)$.

Уменьшение скорости профиля ЛВВ на топологии малого мира происходит потому, что добавляется число связей, потенциально тормозящих процесс. Однако малое число добавочных связей приводит к небольшому замедлению эволюции профиля времен. Иными словами, дальние связи незначительно уменьшают эффективность параллельного моделирования, но подавление рассинхронизации дает общий выигрш в предсказуемости процесса вычисления.

Таким образом, консервативная модель роста профиля ЛВВ на регулярной топологии может быть отнесена к классу универсальности KPZ. Рассинхронизация локальных времен в этом алгоритме растет с размером системы, а средняя скорость роста профиля постоянна и примерно равна 1/4. Добавление случайных дальних взаимодействий между логическими процессами улучшает синхронизацию. Ширина профиля остается постоянной вне зависимости от размера системы. При этом скорость профи-

Таблица 1. Сводная таблица поведения скорости v и квадратичной ширины w^2 профиля локальных времен в модели консервативного алгоритма

	Регулярная топология $p = 0$	Топология малого мира $p > 0$
$w^2(t)$	$w^2(t) \propto t^{2\beta}, t < t_x$ $\beta = 0.326(5)$	$w^2 \propto t^{2\beta}$ $\beta \sim \ln(p)$
$w_\infty^2(N)$	$w_\infty^2(N) \propto N^{2\alpha}$ $\alpha = 0.49(1)$	$w_\infty^2(N) \sim \text{const}$ $\alpha = 0$
v	$v_0 = 0.246410(7)$	$v = v_0 - \Delta v$ $\Delta v \propto p^{0.306(4)}$

ля снижается незначительно. Основные результаты для консервативного алгоритма ПМДС кратко представлены в табл. 1.

3.2. Оптимистический алгоритм

В этом разделе мы опишем результаты исследования модели для оптимистического алгоритма и покажем связь нашей модели с классами универсальности KPZ с коррелированным шумом (Quenched KPZ, qKPZ) [25] и классом направленной перколяции (Directed Percolation, DP) [26].

3.2.1. Класс универсальности направленной перколяции

Уравнение KPZ с коррелированным шумом отличается от обычного KPZ (10) тем, что шум $\eta_q(x, h(x, t))$ теперь является функцией x и h и не зависит от времени:

$$\frac{\partial}{\partial t} h(x, t) = F + \nu \nabla^2 h + \frac{\lambda}{2} (\nabla h)^2 + \eta_q(x, h(x, t)). \tag{17}$$

Такая модель описывает рост поверхности в некоторой гетерогенной среде (например, распространение влажного края при намокании бумаги). F в уравнении (17) может быть интерпретирована как сила, заставляющая поверхность расти.

В qKPZ (17) есть фазовый переход в точке F_c . Когда сила достаточно большая, поверхность растет непрерывно и в этом случае шум эквивалентен шуму в обычном KPZ. Когда же сила F маленькая, появляется часть поверхности, где рост подавляется сильным негативным шумом. Фазовый переход к шероховатости (вблизи F_c) относится к классу универсальности направленной перколяции. В одномерном случае универсальные критические показатели

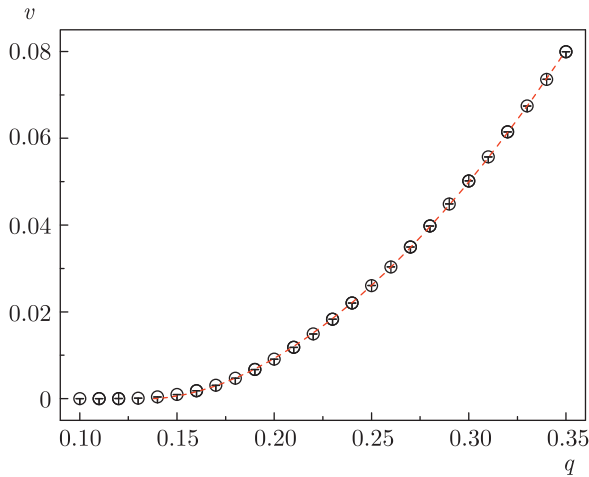


Рис. 4. Средняя скорость v профиля как функция параметра q для системы из $N = 1000$ логических процессов на регулярной топологии $p = 0$

поверхности выражаются через критические показатели класса универсальности одномерной направленной перколяции: $\alpha = \beta = \nu_{\perp}^{DP} / \nu_{\parallel}^{DP} \approx 0.633$ и $z = 1$, где ν_{\perp}^{DP} и ν_{\parallel}^{DP} — это критические показатели, соответственно описывающие расхождение корреляционной длины и времени в классе DP.

К классу направленной перколяции, согласно гипотезе Грассбергера, относятся системы, совершающие непрерывный фазовый переход в одно абсорбирующее состояние (состояние, которое система не может покинуть) [27]. Эта гипотеза была подтверждена множеством численных экспериментов. Обсуждаемая нами модель для оптимистического алгоритма демонстрирует аналогичное поведение.

3.2.2. Результаты численных экспериментов

Удобно ввести параметр $q = 1/(1 + b)$, который может быть интерпретирован как интенсивность роста поверхности (growth rate).

Определение критического индекса ν_{\parallel} . Средняя скорость роста профиля ЛВВ $v(t)$ — это среднее приращение высоты профиля за один шаг моделирования (8). Средняя скорость не зависит от времени, но зависит от параметра q , интенсивности роста. С ростом b при некотором значении $q = q_c$ скорость профиля снижается до нуля, т. е. слишком большое количество откатов не позволяет профилю ЛВВ продвигаться вперед. Система оказывается в поглощающем состоянии. Мы можем ожидать, согласно гипотезе Грассбергера, что модель будет принадлежать к классу универсальности направленной перколяции.

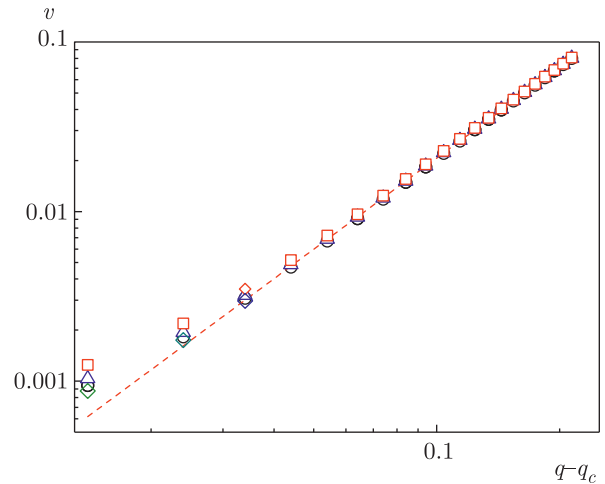


Рис. 5. (В цвете онлайн) Средняя скорость v профиля как функция параметра $(q - q_c)$ в системах разного размера на регулярной топологии $p = 0$. Зеленые ромбы для $N = 5000$; черные кружки для $N = 1000$; синие треугольники для $N = 500$; красные квадраты для $N = 250$

Для нахождения зависимости $v(q)$ при каждом значении параметра q проводится усреднение функции $v(t)$ по шагам моделирования, при этом в усреднении не учитываются значения для шага времени, меньшего типичного интервала релаксации скорости. График зависимости средней скорости v от параметра q представлен на рис. 4. Предполагая, что $v \sim \xi_{\parallel}^{-1} \sim (q - q_c)^{\nu_{\parallel}}$ [26, 28, 29], мы аппроксимировали график функцией вида

$$v(q) = v_0(q - q_c)^{\nu_{\parallel}}. \quad (18)$$

Полученные значения параметров подгонки: $v_0 = 1.26(2)$, $q_c = 0.136(1)$, $\nu_{\parallel} = 1.78(2)$.

На рис. 5 показана зависимость средней скорости v от $(q - q_c)$ в дважды логарифмической шкале. В целом результаты численного моделирования хорошо попадают на кривую $(q - q_c)^{1.78(2)}$, но вблизи критической точки данные отклоняются. Это можно объяснить недостаточной точностью вычислений и эффектом конечного размера системы. Для того чтобы проверить влияние конечного размера, мы провели моделирование систем из 250, 500, 1000 и 5000 логических процессов.

На рис. 6 изображен график зависимости значений скорости v от количества логических процессов N вблизи точки перехода ($q - q_c = 0.034$). При увеличении размера системы средняя скорость профиля снижается. Мы аппроксимировали график функцией $v(N) = v_{N \rightarrow \infty} + A/N^C$. Полученные значения $v_{N \rightarrow \infty}$ для различных q вблизи критического зна-

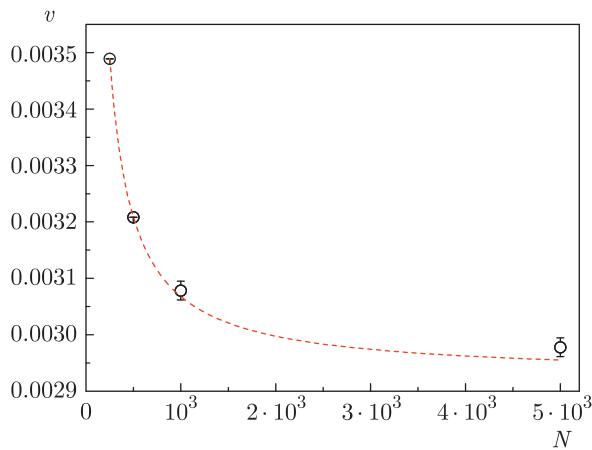


Рис. 6. (В цвете онлайн) Средняя скорость профиля v как функция количества логических процессов N для $q - q_c = 0.034$ на регулярной топологии $p = 0$. Красной штриховой линией показана аппроксимация вида $v(N) = v_{N \rightarrow \infty} + A/N^C$. Найденное значение $v_{N \rightarrow \infty} = 0.0016 \pm 1.5 \cdot 10^{-8}$, $\chi^2 = 1.61894$

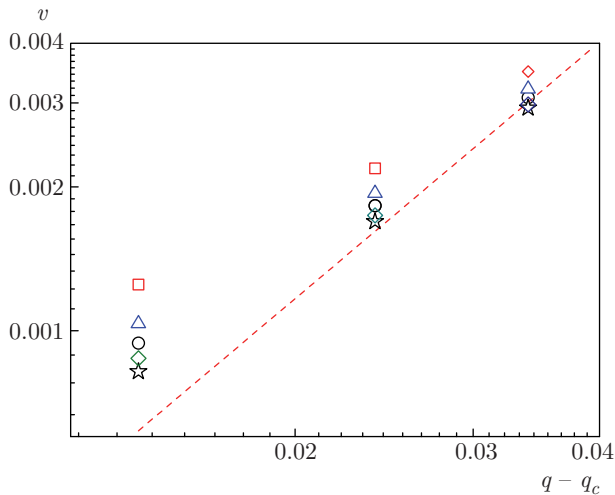


Рис. 7. (В цвете онлайн) Средняя скорость профиля v как функция параметра $(q - q_c)$ в системах разного размера на регулярной топологии $p = 0$. Зеленые ромбы для $N = 5000$; черные кружки для $N = 1000$; синие треугольники для $N = 500$; красные квадраты для $N = 250$. Черными звездочками показаны рассчитанные значения скорости в пределе бесконечного размера системы (рис. 6)

чения изображены на рис. 7. Чем больше моделируемая система, тем ближе полученные значения к предсказанной кривой, однако даже для достаточно больших систем ($N = 5000$) заметно влияние конечности системы.

Таким образом, скорость роста профиля ЛВВ в модели оптимистического алгоритма на регулярной

Таблица 2. Значения параметров функции (18) для регулярной топологии и топологий малого мира с различными значениями параметра p

p	v_0	q_c	ν
0	1.26(2)	0.136(1)	1.78(2)
0.001	1.27(4)	0.140(2)	1.80(3)
0.01	1.32(4)	0.158(2)	1.82(3)
0.1	1.64(7)	0.213(3)	1.96(5)

топологии имеет фазовый переход между активной фазой (когда скорость роста профиля ненулевая) и состоянием пиннинга (когда профиль не растет) в точке $q_c = 0.136(1)$. Найденная нами критическая экспонента $\nu_{||} = 1.78(2)$ близка к экспоненте в классе направленной перколяции, $\nu_{||}^{DP} \approx 1.73$. К классу направленной перколяции также относятся контактные процессы [30, 31], модели распространения эпидемий без иммунизации [32], каталитические реакции [33–35], модели случайного блуждания (branching-annihilating random walks) [36, 37], модели роста поверхностей [12, 38] и другие [26].

Поведение скорости роста профиля для модели оптимистического алгоритма на сетях малого мира изменяется несущественно. Поскольку дополнительные зависимости между ЛП провоцируют больше откатов, в среднем скорость профиля снижается, положение критической точки q_c и критический показатель $\nu_{||}$ при этом смещаются в сторону больших значений (табл. 2).

Ширина профиля локальных времен. В моделях роста ширина характеризуется t_x , типичным значением шага моделирования для выхода на насыщение. Известно, что $t_x \propto N^z$, где z — динамическая экспонента. В нашей модели в период $t < t_x$ ширина растет степенным образом:

$$w^2(t) \propto t^{2\beta}.$$

График изменения ширины профиля ЛВВ со временем для разных топологий представлен на рис. 8. Чем меньше параметр q , тем меньше ширина профиля ЛВВ. Поскольку откаты выравнивают времена логических процессов, увеличение их количества, т. е. уменьшение параметра q , делает профиль ЛВВ более гладким. Добавление небольшого числа дальних связей также улучшает синхронизацию — ширина профиля при одинаковом параметре q на сетях малого мира ниже, чем на регулярной топологии.

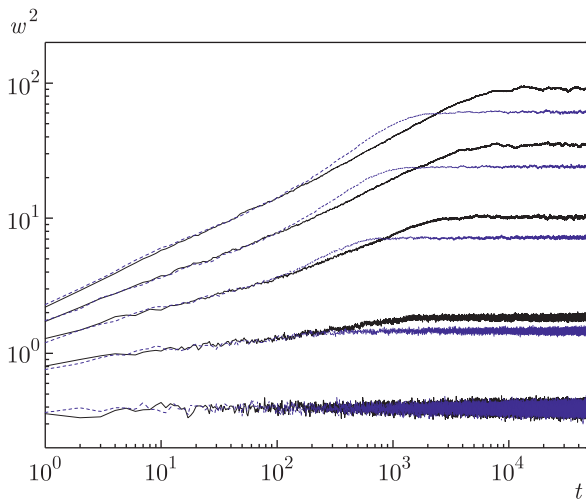


Рис. 8. (В цвете онлайн) Зависимость средней ширины профиля ЛВВ от времени для разных значений параметра q снизу вверх: $[0.1, 0.2, 0.3, 0.4, 0.5]$. Черной линией обозначена ширина для регулярной топологии $p = 0$, синей штриховой — для топологии малого мира с параметром $p = 0.01$. Результаты усреднены по 1000 независимым реализациям

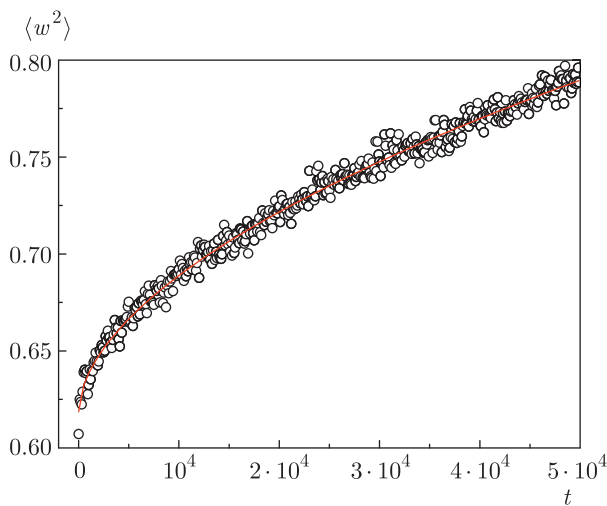


Рис. 9. (В цвете онлайн) Рост ширины w^2 профиля ЛВВ со временем в модели оптимистического алгоритма для системы из $N = 10^5$ логических процессов в критической точке при $p = 0$. Каждая точка на графике — это усреднение по бину в 100 временных шагов моделирования t . Красная линия показывает функцию $w^2(t) = 0.618(2) + 0.00043(5)t^{0.553(9)}$

Мы также исследовали, как изменяется показатель роста β в зависимости от параметра q , доли дальних связей p и размера системы N .

Таблица 3. Результаты аппроксимации ширины в области $t < t_x$ функцией $w^2(t) = A + Bt^{2\beta}$ для систем разного размера N в критической точке q_c

N	t_x	A	B	β
10^3	< 100	0.42(2)	0.17(2)	0.016(9)
10^4	≈ 20000	0.600(6)	0.006(2)	0.13(3)
10^5	> 50000	0.618(2)	0.00043(5)	0.277(9)

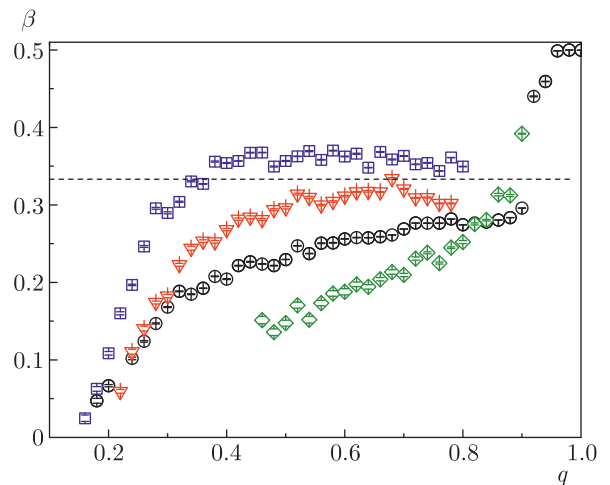


Рис. 10. (В цвете онлайн) Зависимость критического индекса β от параметра q на различных топологиях для системы из $N = 10^3$ логических процессов. Черными кружками обозначены результаты на регулярной топологии $p = 0$, синими квадратами — результаты для топологии малого мира с $p = 0.001$, красными треугольниками — для $p = 0.01$, зелеными ромбами — для $p = 0.1$. Черной штриховой линией приведено значение критического индекса для класса KPZ $\beta = 1/3$

В критической точке $q \approx 0.136$ в системе из $N = 10^5$ логических процессов на регулярной топологии ($p = 0$) ширина профиля в области $t < t_x$ растет по степенному закону (рис. 9):

$$w^2(t < t_x) = 0.618(2) + 0.00043(5)t^{0.553(9)},$$

т. е. показатель $\beta \approx 0.277$. В табл. 3 приведено значение показателя β в критической точке q_c для регулярной топологии ($p = 0$) в зависимости от количества логических процессов. По таблице видно, что β увеличивается с количеством логических процессов. Нас интересует показатель β в термодинамическом пределе бесконечного размера системы, $N \rightarrow \infty$.

На рис. 10 изображен график зависимости критического индекса роста β от параметра q для различных топологий взаимодействия логических про-

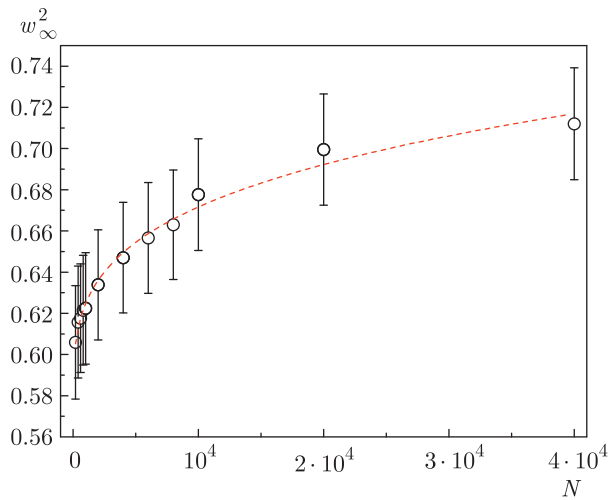


Рис. 11. Величина насыщения ширины профиля ЛВВ w_∞^2 в зависимости от количества логических процессов N в точке $q_c = 0.136$. Каждая точка на графике получена усреднением функции $w^2(t)$ по диапазону $[t_\times, t_{max}]$, где t_\times — точка, где ширина равна значению насыщения, t_{max} — максимальное время моделирования

цессов. Поведение критического индекса можно разделить на три режима: при малых, умеренных и больших значениях параметра q . При малых q , т. е. в фазе пиннинга, профиль локальных времен практически не растет, поэтому и ширина профиля w^2 не растет и остается близкой к нулю. Для умеренных $q \in [0.4; 0.8]$, или в активной фазе, где профиль растет с ненулевой скоростью v и имеется ненулевое количество откатов, во всех топологиях показатель β либо равен константе, близкой к $1/3$, либо стремится к этому значению. Для точного определения показателей β к рис. 10 необходимы поправки на конечный размер системы. При $q > 0.8$ наблюдается режим, при котором профиль растет практически свободно, без откатов ($q = 1$ соответствует чистому свободному росту). В этом случае ширина профиля растет быстрее всего, показатель β стремится к значению $1/2$.

По-видимому, в активной фазе, при q значительно большем q_c , но несколько меньшем 1, показатель β совпадает с показателем из класса KPZ, как в консервативном алгоритме. При q вблизи единицы, система растет по закону случайного роста. Таким образом, модель оптимистического алгоритма в одномерном случае ($p = 0$) демонстрирует несколько режимов поведения в зависимости от значения q — она имеет две критические точки ($q = q_c$ и $q = 1$) и поведение в классе KPZ для промежуточных значений q .

Таблица 4. Результаты аппроксимации ширины в области $t > t_\times$ функцией $w_\infty^2(N) = A + BN^{2\alpha}$

	A	B	α
$q = q_c$	0.57(2)	0.009(6)	0.13(5)
$q - q_c = 0.05$	0.629(5)	0.0007(3)	0.29(5)
$q - q_c = 0.1$	0.671(6)	0.0003(1)	0.36(5)
$q - q_c = 0.15$	0.710(7)	0.0002(1)	0.39(5)

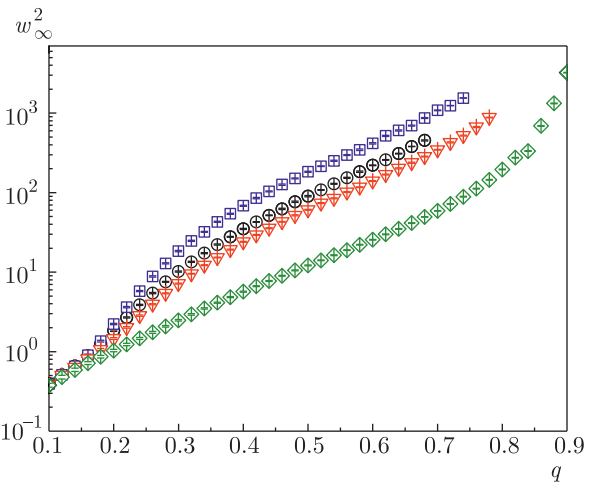


Рис. 12. (В цвете онлайн) Зависимость ширины профиля ЛВВ w_∞^2 от параметра q на различных топологиях для системы из $N = 10^3$ логических процессов. Черными кружками обозначены результаты на регулярной топологии $p = 0$, синими квадратами — результаты для топологии малого мира с $p = 0.001$, красными треугольниками — для $p = 0.01$, зелеными ромбами — для $p = 0.1$

Мы также изучили, как изменяется величина насыщения w_∞^2 с размером системы. Мы усреднили функцию $w^2(t)$ по диапазону $[t_\times, t_{max}]$, где t_\times — точка, где ширина равна значению насыщения, t_{max} — максимальное время моделирования. График $w_\infty^2(N)$ в критической точке для регулярной топологии представлен на рис. 11. Мы аппроксимировали график функцией $w_\infty^2(N) = A + BN^{2\alpha}$. Значения показателя шероховатости α в области критической точки приведены в табл. 4. В самой критической точке показатель α намного ниже, чем в ее окрестности.

Ширина насыщения w_∞^2 также имеет несколько режимов в зависимости от параметра q . Для малых q w_∞^2 близка к нулю. В области $q \in [0.4; 0.7]$ ширина насыщения w_∞^2 растет с параметром q примерно одинаково для всех видов топологий (рис. 12). Для

больших значений q определить значение ширины сложно из-за очень большого времени выхода на насыщение t_{\times} .

4. ЗАКЛЮЧЕНИЕ

В работе обсуждаются модели синхронизации при параллельном моделировании дискретных событий. Оказывается, эволюция локальных времен параллельных процессов имеет аналогию с ростом поверхности. Это дает возможность применить для анализа синхронизации и эффективности вычислений язык и аппарат статистической механики. Для консервативного алгоритма модель может быть сведена к модели роста по уравнению Кардара–Паризи–Жанга. Для оптимистического алгоритма модель по формулировке похожа на модель роста SOS. При этом, однако, у нее есть три режима — режим в классе направленной перколяции, режим случайного роста и промежуточный режим, близкий к классу универсальности KPZ. Такое поведение может быть также сопоставлено с поведением поверхности роста в qKPZ.

С точки зрения проведения параллельных вычислений введение связей типа малого мира улучшает синхронизацию процессорных элементов при небольшом замедлении вычислений. Консервативный алгоритм свободен от блокировок (deadlock), тогда как в оптимистическом алгоритме при увеличении частоты нарушения причинности возможен переход шероховатости в абсорбирующее состояние с нулевой общей производительностью. При умеренных значениях частоты нарушения причинности общая производительность положительна и десинхронизация насыщается с ростом числа параллельных процессов.

Финансирование. Работа инициирована при выполнении гранта № 14-21-00158 Российского научного фонда и завершена в рамках государственного задания 0236-2019-0001.

ЛИТЕРАТУРА

1. Л. Д. Ландау, И. Н. Мейман, И. М. Халатников, в книге: Л. Д. Ландау, *Собрание трудов*, Т. 2, под ред. Е. М. Лифшица, И. М. Халатникова, Физматлит, Москва (2008), с. 357.
2. И. М. Халатников, частное сообщение.
3. Conference on Computational Physics, 20–24 August, Moscow (2013), <http://ccp2013.ac.ru/>.
4. Nvidia-V100. <https://www.nvidia.com/ru-ru/data-center/tesla-v100/>.
5. R. M. Fujimoto, *Comm. ACM* **33**, 30 (1990).
6. B. D. Lubachevsky, *Complex Syst.* **1**, 1099 (1987).
7. B. D. Lubachevsky, *J. Comput. Phys.* **75**, 103 (1988).
8. D. R. Jefferson, *ACM Transactions on Programming Languages and Systems (TOPLAS)* **7**, 404 (1985).
9. G. Korniss, Z. Toroczkai, M. A. Novotny, and P. A. Rikvold, *Phys. Rev. Lett.* **84**, 1351 (2000).
10. L. Ziganurova, M. Novotny, and L. Shchur, *J. Physics: Conf. Ser.* **681**, 012047 (2016).
11. M. Kardar, G. Parisi, and Y. C. Zhang, *Phys. Rev. Lett.* **56**, 889 (1986).
12. U. Alon, M. R. Evans, H. Hinrichsen, and D. Mukamel, *Phys. Rev. Lett.* **76(15)**, 2746 (1996).
13. L. N. Shchur and M. A. Novotny, *Phys. Rev. E* **70**, 026703 (2004).
14. L. Ziganurova and L. N. Shchur, *Phys. Rev. E* **98**, 022218 (2018).
15. G. Korniss, M. Novotny, H. Guclu, Z. Toroczkai, and P. A. Rikvold, *Science* **299**, 677 (2003).
16. L. Ziganurova and L. Shchur, in: *Lecture Notes in Computer Science* **10421**, Springer (2017), p. 246.
17. L. Shchur and L. Ziganurova, *Lobachevskii J. Math.* **38(5)**, 967 (2017).
18. P. Crawford, S. J. Eidenbenz, P. D. Barnes, and P. A. Wilsey, in *Proc. Winter Simulation Conference*, IEEE Press, Las Vegas (2017), p. 1025.
19. D. J. Watts and S. H. Strogatz, *Nature* **393**, 440 (1998).
20. A. Barrat and M. Weigt, *Eur. Phys. J. B* **13**, 547 (2000).
21. R. M. Fujimoto, C. Carothers, A. Ferscha, D. Jefferson, M. Loper, M. Marathe, and S. J. Taylor, in *Proc. Winter Simulation Conference*, IEEE Press, Las Vegas, USA (2017).
22. V. Y. Vee and W. J. Hsu, *Parallel Discrete Event Simulation: A Survey*, Technical Report. Nanyang Technological Univ., Singapore (1999).

23. D. Jefferson and R. M. Fujimoto, in: *Advances in Modeling and Simulation*, Springer (2017), p. 97.
24. F. Family and T. Vicsek, *J. Physics A: Mathematical and General* **18**, L75 (1985).
25. K. A. Takeuchi, *An Appetizer to Modern Developments on the Kardar–Parisi–Zhang Universality Class*, arXiv:1708.06060.
26. H. Hinrichsen, *Adv. Phys.* **49(7)**, 815 (2000).
27. P. Grassberger, *Z. Physik B: Condens. Matt.* **47**, 365 (1982).
28. G. Odor, *Rev. Mod. Phys.* **76**, 663 (2004).
29. F. D. Reis and F. D. Brazilian, *J. Phys.* **33(3)**, 501 (2003).
30. T. M. Liggett, *Interacting Particle Systems*, Springer Science and Business Media (2012).
31. R. Dickman and M. A. Burschka, *Phys. Lett. A* **127(3)**, 132 (1988).
32. P. Grassberger, *Math. Biosci.* **63(2)**, 157 (1983).
33. R. M. Ziff, E. Gulari, and Y. Barshad, *Phys. Rev. Lett.* **56(24)**, 2553 (1986).
34. F. Scholgl, *Z. Phys.* **253(2)**, 147 (1972).
35. E. V. Albano, *Physica A: Statistical Mechanics and its Applications* **216(3)**, 213 (1995).
36. H. Takayasu and A. Y. Tretyakov, *Phys. Rev. Lett.* **68(20)**, 3060 (1992).
37. M. Hoyuelos, E. V. Albano, and H. O. Martin, *J. Phys. A: Mathematical and General* **30(2)**, 431 (1997).
38. J. Kertesz and D. E. Wolf, *Phys. Rev. Lett.* **62(22)**, 2571 (1989).