DOI: 10.51790/2712-9942-2024-5-4-13

СУБТАЙЛИНГ В ИТЕРАЦИОННЫХ МЕТОДАХ: ПРИНЦИПЫ ПОСТРОЕНИЯ И ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

В. М. Свешников[а], А. М. Яклюшин[б]

Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск, Российская Федерация

[а] ✍ *victor@lapasrv.sscc.ru,* [б] *a.yaklyushin@g.nsu.ru*

Аннотация: предлагается и экспериментально исследуется новый подход к ускорению итерационных методов — субтайлинг, основанный на идеях классического тайлинга. Суть подхода заключается в повторном использовании данных, загруженных в кэш-память процессора, что значительно сокращает время вычислений и повышает эффективность алгоритмов. Основная идея заключается в формировании субтайлов — вторичных тайлов, смещенных по диагонали на один узел относительно исходных тайлов. Предложенный подход был протестирован на итерационном методе последовательной верхней релаксации (SOR). Результаты численных экспериментов показали, что субтайлинг позволяет ускорить вычисления более чем в 5 раз. Изложен алгоритм формирования и использования субтайлов, проведен анализ его эффективности.

Ключевые слова: численные эксперименты, итерационные методы, ускорение расчетов, тайлинг, субтайлинг, кэш-память.

Благодарности: исследование выполнено за счет гранта Российского научного фонда № 23-21-00385.

Для цитирования: Свешников В. М., Яклюшин А. М. Субтайлинг в итерационных методах: принципы построения и численные эксперименты. *Успехи кибернетики*. 2024;5(4):95–102. DOI: 10.51790/2712-9942-2024-5-4-13.

Поступила в редакцию: 08.10.2024. В окончательном варианте: 16.11.2024.

SUB-TILING IN ITERATIVE METHODS: PRINCIPLES AND NUMERICAL EXPERIMENTS

V. M. Sveshnikov[a], A. M. Yaklyushin[b]

The Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch, Russian Academy of Sciences, Novosibirsk, Russian Federation

[a] ✍ *victor@lapasrv.sscc.ru,* [b] *a.yaklyushin@g.nsu.ru*

Abstract: we proposed and experimentally investigated a new approach to accelerating iterative methods called sub-tiling based on the ideas of conventional tiling. The new approach reuses the data loaded into the CPU cache, which significantly reduces the computation time and increases the efficiency of algorithms. The key idea is to form subtiles, or secondary tiles shifted diagonally by one node relative to the original tiles. We tested this concept using the iterative successive over-relaxation (SOR) method. The results of numerical experiments show that sub-tiling speeds up the computation by more than 5x. The paper presents an algorithm for sub-tile generation and application, and the analysis of the algorithm efficiency.

Keywords: numerical experiments, iterative approach, higher computation performance, tiling, sub-tiling, cache memory.

Acknowledgements: this study is supported by the Russian Science Foundation grant No. 23-21-00385.

Cite this article: Sveshnikov V. M., Yaklyushin A. M. Sub-Tiling in Iterative Methods: Principles and Numerical Experiments. *Russian Journal of Cybernetics*. 2024;5(4):95–102. DOI: 10.51790/2712-9942-2024-5-4-13.

Original article submitted: 08.10.2024. Revision submitted: 16.11.2024.

Введение

Современные вычислительные задачи часто требуют значительных объемов памяти и продолжительных расчетов, особенно при решении краевых задач и других численных задач, использующих

итерационные методы. Для ускорения вычислений эффективно применяется тайлинг — метод, который оптимизирует использование кэш-памяти процессора. Суть тайлинга заключается в разбиении расчетной сетки на небольшие блоки (тайлы), которые загружаются в кэш, что минимизирует задержки при доступе к данным и ускоряет выполнение арифметических операций.

В статье [1] описаны методы создания эффективных решателей для систем линейных уравнений с блочно-ленточными матрицами, а также с использованием структуры данных для оптимального хранения и предкомпилятора для ускорения вычислений. В работе применялся итерационный метод Зейделя, а эксперименты проводились на задаче решения СЛАУ для трехмерной сетки $100 \times 100 \times 100$, где оптимизации позволили ускорить вычисления на 30%.

В статье [2] рассматривается использование тайлинга для улучшения производительности при решении уравнений в частных производных (PDE) в трехмерных областях. Эксперименты проводились на задачах 3D Jacobi, Red-black SOR и RESID из SPEC/NAS benchmarks. Применение тайлинга позволило улучшить производительность на 17–121% за счет снижения пропусков кэша и улучшения локальности данных. Например, для 3D Jacobi достигнуто ускорение на 27% по сравнению с исходной версией.

В статье [3] описаны экспериментальные исследования ускорения решения краевых задач методом декомпозиции области. Применение тайлинга к итерационным методам SOR и SSOR на сетках различного размера позволило ускорить вычисления до 3 раз. Эксперименты проводились на модельной задаче для уравнения Лапласа с граничными условиями Дирихле, решаемой на квадратной области с использованием метода конечных разностей.

В статье [4] показано, что модифицированный алгоритм Гаусса-Зейделя при решении двумерной задачи Дирихле был ускорен в 2,9 раза после применения специальных методов разбиения пространства и тайлинга. Аналогичное ускорение в 2,4 раза было достигнуто для трехмерной задачи Дирихле.

В статье [5] обсуждается применение алгоритма DiamondTorre для численного моделирования волновых процессов. Этот алгоритм ориентирован на эффективное использование иерархии памяти и параллелизма на графических процессорах общего назначения (GPGPU) (иерархический тайлинг). Показано, что распараллеливание алгоритма и применение тайлинга дает ускорение в 5 раз по сравнению с традиционным подходом. Этот результат был получен при моделировании волнового уравнения с использованием схемы второго порядка аппроксимации.

Однако, несмотря на значительные преимущества, классический тайлинг имеет свои ограничения. В частности, эффективность использования кэш-памяти может быть ограничена частотой обращения к оперативной памяти и размерами тайлов, что приводит к необходимости поиска новых методов оптимизации вычислений.

Предлагается новый метод — субтайлинг, который является развитием классического тайлинга и направлен на повышение производительности итерационных методов. Идея субтайлинга заключается в повторном использовании тайлов, уже загруженных в кэш-память процессора, путем формирования и обработки связанных с ними вторичных тайлов. Это позволяет значительно уменьшить количество обращений к оперативной памяти и ускорить процесс вычислений, не меняя сути применяемого алгоритма.

Цель данной работы — разработка и исследование субтайлинга, а также сравнение его эффективности с классическим тайлингом на примере итерационного метода SOR.

Принципы построения тайлинга

Тайлинг — это метод организационной оптимизации вычислений, направленный на эффективное использование кэш-памяти процессора. Суть метода заключается в разбиении исходной сетки задачи на более мелкие блоки, называемые тайлами. Эти тайлы загружаются в кэш-память процессора, что позволяет сократить время доступа к данным и ускорить выполнение арифметических операций.

Основные разделы тайлинга:

1. Разбиение разностной сетки на тайлы: исходная сетка делится на небольшие прямоугольные блоки (тайлы), которые могут эффективно загружаться в кэш-память процессора. Размер тайлов подбирается таким образом, чтобы они максимально использовали доступную кэш-память.

2. Оптимизация доступа к данным: тайлинг позволяет организовать вычисления так, чтобы данные хранились в кэше процессора как можно дольше. Это достигается за счет изменения последова-

тельности вычислений, при которой промежуточные данные остаются в кэше на протяжении большего числа операций. Вместо последовательного прохода по всей сетке (слева направо и снизу вверх), в тайлинге вычисления осуществляются по тайлам, а внутри тайла — по его узлам.

3. Экспериментальный подбор размера тайлов: размер тайлов определяется экспериментально для каждой конкретной задачи и вычислительного устройства. Это помогает найти оптимальный баланс между количеством операций и объемом кэш-памяти. Оптимальные размеры зависят от архитектуры процессора и объема доступной кэш-памяти, поэтому могут варьироваться в зависимости от используемого оборудования.

На рис. 1 приведен пример разбиения сетки на тайлы и последовательности их обработки. Слева показана традиционная схема прохода по сетке, справа — схема, использующая тайлинг.

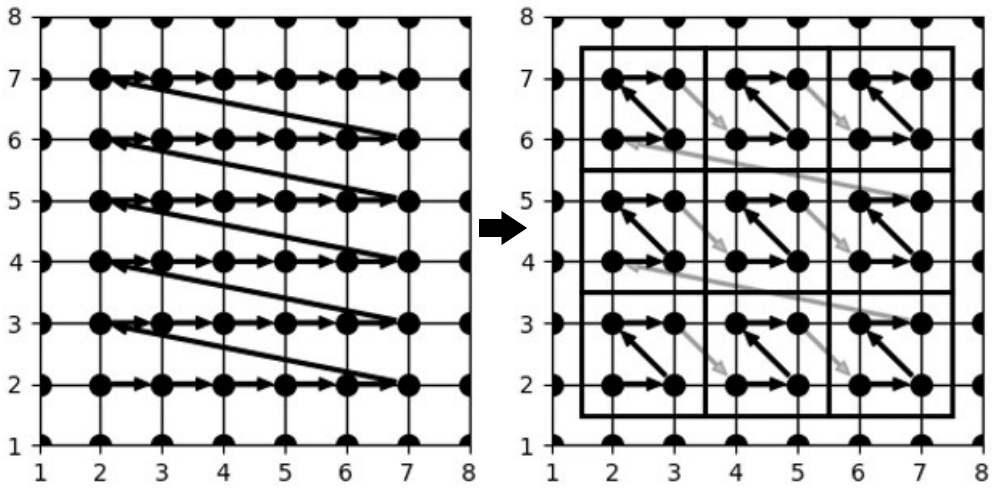

Рис. 1. *Разбиение сетки на тайлы и последовательности их обработки (классический вариант тайлинга)*

Численные эксперименты показывают, что использование тайлинга может значительно ускорить вычислительные процессы. В задачах с большой размерностью сетки и большим числом итераций тайлинг позволяет сократить время вычислений более чем в четыре раза по сравнению с традиционными методами. Оптимизация размера тайлов и порядка вычислений обеспечивает высокую производительность, минимизируя задержки при доступе к данным.

Принципы построения субтайлинга

Субтайлинг — это инновационный подход, направленный на дальнейшее ускорение итерационных методов. Основная концепция субтайлинга заключается в повторном использовании тайлов, загруженных в кэш-память процессора, посредством формирования и обработки субтайлов — дополнительных тайлов, смещенных по диагонали на один узел относительно исходных тайлов.

Основные разделы субтайлинга:

1. Разбиение сетки на тайлы: исходная сетка разбивается на тайлы традиционным образом, обеспечивая их эффективную загрузку в кэш-память процессора.

2. Формирование субтайлов: для каждого тайла создается субтайл, смещенный по диагонали на один узел от исходного, при этом смещение организовано так, чтобы не нарушать порядок вычислений в процессе проведения итераций.

3. Размер субтайла может изменяться в зависимости от его положения на сетке. Для тайлов, удаленных от границы, размер субтайла совпадает с размером исходного тайла. Если тайл расположен вблизи границы области, размер субтайла уменьшается, если он включает в себя граничные узлы сетки, и увеличивается, если субтайл не содержит узлов, непосредственно примыкающих к граничным узлам.

4. Обработка исходного тайла: вычисления начинаются с традиционной обработки узлов исходного тайла.

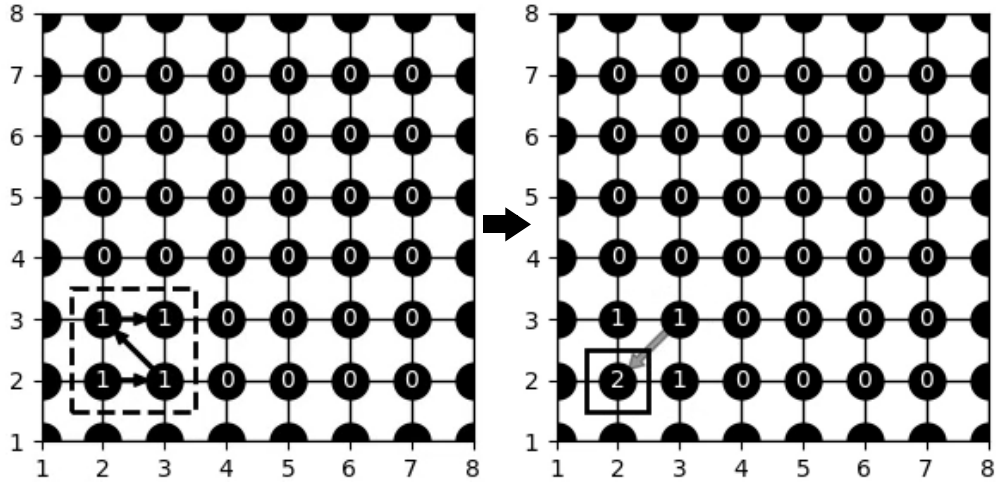

Рис. 2. *Разбиение сетки на субтайлы и последовательность их обработки (тайл вблизи левой нижней границы)*

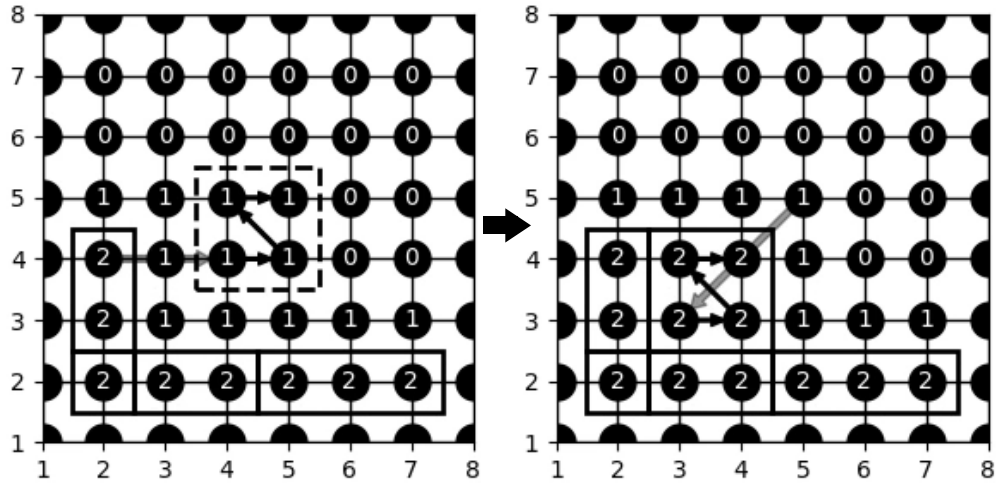

Рис. 3. *Разбиение сетки на субтайлы и последовательность их обработки (тайл вдали от границ)*

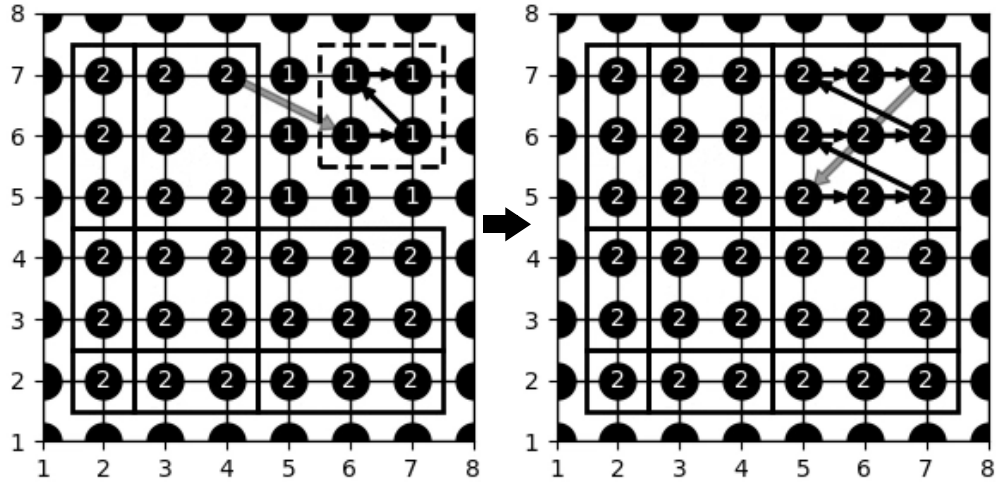

Рис. 4. *Разбиение сетки на субтайлы и последовательность их обработки (тайл вблизи правой верхней границы)*

5. Обработка субтайла: после вычисления узлов исходного тайла производится обработка соответствующего субтайла. Субтайлы обрабатываются аналогично основным тайлам, что позволяет дольше удерживать данные в кэш-памяти. Этот процесс повторяется до тех пор, пока не будут вычислены все тайлы и их субтайлы на текущей итерации.

6. На каждой текущей итерации проводятся субитерации без нарушения последовательности вычислений, заданных итерационным методом.

На рис. 2–4 приведен пример разбиения сетки на субтайлы и последовательности их обработки в трех различных случаях, отличающихся положением тайлов относительно границы расчетной области. Число внутри узла указывает на количество субитераций, совершенных за одну итерацию. Пунктирными линиями обозначены тайлы, сплошными — субтайлы. Черные указатели задают последовательность обработки тайла или субтайла, серые — переход от последнего узла тайла к первому узлу соответствующего субтайла. На рисунках слева показано формирование тайлов, справа — формирование и переход к субтайлам. Всюду для узлов с числом субитераций 2 сплошная линия указывает на принадлежность конкретному субтайлу. На рис. 2 тайл расположен вблизи границы, размер субтайла уменьшен. На рис. 3 тайл находится вдали от границы, размер субтайла равен размеру тайла. На рис. 4 тайл расположен вблизи границы, размер субтайла увеличен.

Подробный разбор рис. 3:

1. На левом и правом рисунках сплошной линией выделены субтайлы, соответствующие предыдущим тайлам.

2. На левом рисунке пунктирной линией выделен новый тайл, расположенный вдали от границы. Обработка его узлов производится в следующем порядке: (4,4) $\to$ (4,5) $\to$ (5,4) $\to$ (5,5).

3. На правом рисунке осуществляется переход от последнего узла тайла (5,5) к первому узлу соответствующего ему субтайла (3,3). Обработка узлов нового субтайла производится в следующем порядке: (3,3) $\to$ (4,3) $\to$ (3,4) $\to$ (4,4).

Расширение субтайлинга заключается в увеличении числа субтайлов для каждого тайла. Принцип формирования новых субтайлов остается прежним: для построения n-го субтайла осуществляется сдвиг на n узлов по диагонали в одном направлении от исходного тайла, а размер корректируется в зависимости от его расположения относительно границы сетки. Необходимое условие — количество субтайлов для каждого тайла должно совпадать, иначе суть итерационного алгоритма будет нарушена.

Уровнем субтайлинга будем называть число субтайлов, соответствующих каждому тайлу. Таким образом, если уровень субтайлинга равен нулю, то речь идет о классическом тайлинге. При уровне субтайлинга n за одну итерацию совершается $n + 1$ субитераций.

Возможны случаи, когда размер некоторых субтайлов может быть равен нулю. Это происходит, если уровень субтайлинга равен размеру тайла. В таком случае, если тайл расположен вблизи нижней или левой границы, последний соответствующий ему субтайл лежит вне расчетной области и исключается из расчетов. Уровень субтайлинга может также превышать размер тайлов, это не нарушает суть итерационного алгоритма, однако выбор такого уровня может быть неэффективным. Наибольшая эффективность метода наблюдается при уровне субтайлинга на единицу меньше размера тайлов. В приведенных численных экспериментах всюду уровень субтайлинга на единицу меньше размера тайлов.

Численные эксперименты показывают, что субтайлинг обеспечивает значительное ускорение вычислений по сравнению с классическим тайлингом. Оптимизация размера тайлов обеспечивает максимальную производительность, минимизируя задержки при доступе к данным и улучшая использование кэш-памяти процессора. Результаты численных экспериментов показали, что субтайлинг позволяет увеличить коэффициент ускорения, достигнутый классическим тайлингом, более чем в 1.5 раза.

Экспериментальные исследования ускорения расчетов

Цель проводимых численных экспериментов — исследование ускорения итерационных методов за счет применения тайлинга и субтайлинга.

Исследования проводились на модельной задаче о расчете электрического поля в цилиндрическом конденсаторе, образованном двумя концентрическими окружностями с радиусами $R_1 = 0.1$, $R_2 = 1$ с заданными на них потенциалами $\varphi(R_1) = 1$, $\varphi(R_2) = 2$, что приводит к задаче

$$\Delta\varphi = 0 \quad \text{в области } G, \tag{1}$$

$$\varphi = g \quad \text{на границе } \Gamma, \tag{2}$$

где Δ — оператор Лапласа в декартовых координатах, φ — искомая, g — заданная функция координат x, y. Рассматриваются декартовы и полярные координаты r, φ, причем $r = \sqrt{x^2 + y^2}$.

Аналитическое решение данной задачи имеет вид

$$\varphi(r) = \ln\left(\frac{rR_2}{R_1^2}\right) \Big/ \ln\left(\frac{R_2}{R_1}\right). \tag{3}$$

В качестве G выбирается квадратная область $\{[0.3, 0.7] \times [0, 0.4]\}$ с соответствующей границей Г. Граничная функция g определяется согласно (3).

На равномерной сетке $\omega_h = \{x_i = ih, y_j = jh;\ i = \overline{1,N};\ j = \overline{1,N}\}$ при помощи обычной пятиточечной схемы

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} = 0 \tag{4}$$

аппроксимируется исходная задача (1), (2) с заданными условиями Дирихле. Сеточные уравнения решались итерационным методом последовательной верхней релаксации SOR, реализуемым по формуле

$$u_{i,j}^{n+1} = (1-\omega)u_{i,j}^n + \omega\left(\frac{u_{i-1,j}^{n+1} + u_{i,j-1}^{n+1} + u_{i+1,j}^n + u_{i,j+1}^n}{4}\right), \qquad i = \overline{1,N}; \qquad j = \overline{1,N},$$

на различных сетках при $N = 8, 16, 32, 64, 128, 256, 512, 1024$ и различных размерах тайла $n = 1, 2, 4, 8, 16$.

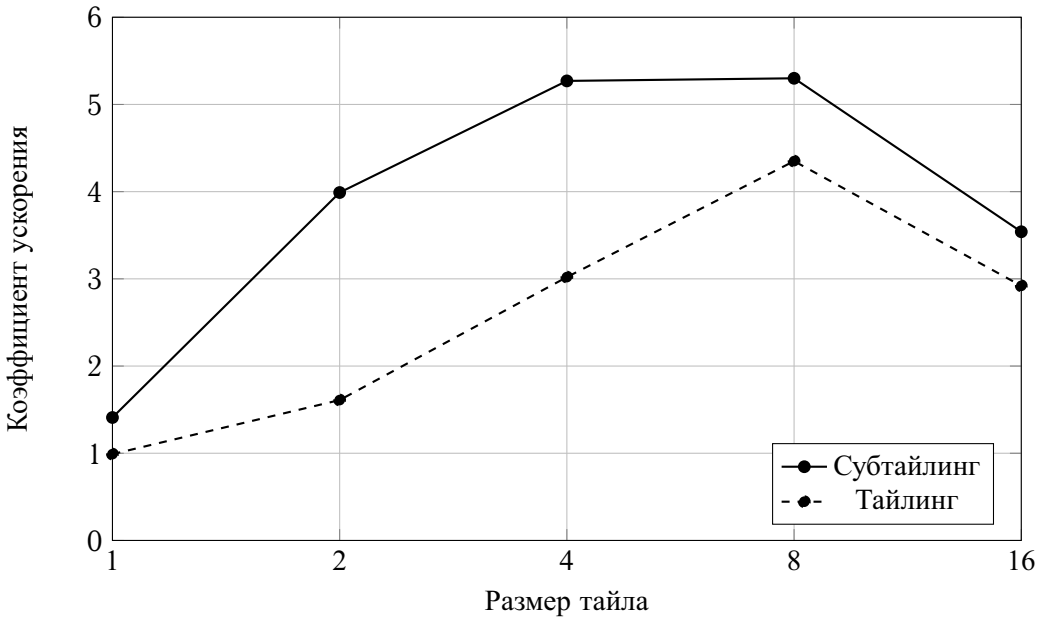

Рис. 5. *Коэффициент ускорения при* $N = 1024$ *(Apple M1)*

Расчеты проводились на персональном компьютере Apple M1 (таблицы 1, 2 и рис. 5) и вычислительных ресурсах МСЦ РАН Intel Xeon Gold 6248R (CLK) (таблицы 3, 4 и рис. 6).

В таблицах приведены коэффициенты ускорения Q_t — с применением тайлинга и Q_s — с применением субтайлинга, равные

$$Q_t = \frac{T_0}{T_t}, \qquad Q_s = \frac{T_0}{T_s},$$

T_0 — время счета без тайлинга и субтайлинга, T_t — время счета с применением тайлинга, T_s — время счета с применением субтайлинга (при Q_t, $Q_s > 1$ рассматриваемый подход выигрывает). На графиках (рис. 5, 6) приведен коэффициент ускорения с применением тайлинга и субтайлинга при N = 1024.

Из данных таблиц видно, что: 1) оптимальный размер тайла $n = 8$, при котором применение тайлинга дает ускорение более 4 раз; 2) при малых N ускорение небольшое, что объясняется частой

Таблица 1

Коэффициент ускорения с применением тайлинга (Apple M1)

$N \setminus n$	1	2	4	8	16
8	1,05	1,03	0,99	0	0
16	0,97	0,99	1,04	1,50	0
32	0,99	1,09	1,55	2,27	1,48
64	0,99	1,14	2,20	3,20	2,10
128	1,00	1,39	2,68	3,88	2,57
256	1,00	1,52	2,90	4,18	2,80
512	1,00	1,58	3,02	4,38	2,91
1024	0,99	1,61	3,02	4,35	2,92

Таблица 2

Коэффициент ускорения с применением субтайлинга (Apple M1)

$N \setminus n$	1	2	4	8	16
8	1,07	1,03	0,99	0	0
16	0,95	1,33	1,54	1,44	0
32	1,05	2,00	2,44	2,35	1,34
64	1,03	2,84	3,60	3,49	2,27
128	1,24	3,48	4,51	4,55	2,94
256	1,33	3,76	4,96	5,02	3,26
512	1,38	3,93	5,18	5,26	3,48
1024	1,41	3,99	5,27	5,30	3,54

сменой содержимого сверхбыстрой памяти; 3) ускорение уменьшается при увеличении размера тайла после оптимального значения, что объясняется увеличением вклада времени выполнения арифметических операций в тайле; 4) субтайлинг при размере тайла $n = 8$ дает ускорение более 5 раз; 5) применение субтайлинга при достаточно большом N умножает коэффициент ускорения классического тайлинга более чем в 1.5 раза.

Заключение

Проведено экспериментальное исследование ускорения итерационных методов на примере последовательной верхней релаксации (SOR). Ускорение достигается путем изменения последовательности перебора узлов сетки с целью как можно более долгого удержания данных в быстрой памяти компьютера без нарушения сути итерационного алгоритма. Данный подход, получивший название тайлинг, заключается в разбиении сетки на прямоугольные блоки (тайлы) и замене традиционного перебора узлов во всей области их переборами по тайлам. Предложен новый подход к построению тайлов, характеризующийся их расширением за счет узлов, примыкающих к исходному тайлу, названный субтайлингом. Численные эксперименты проводились на модельной задаче, которая покрывалась сеткой, содержащей от десятков до миллиона узлов. Размеры тайлинга менялись от 8×8 до 16×16 узлов.

Таблица 3

Коэффициент ускорения с применением тайлинга (CLK)

$N \setminus n$	1	2	4	8	16
8	0,95	1,20	1,29	0	0
16	0,92	1,44	1,80	1,30	0
32	0,99	1,72	2,26	1,62	1,27
64	0,97	1,77	2,36	1,68	1,27
128	0,94	1,75	2,35	1,67	1,27
256	0,97	1,77	2,20	1,65	1,27
512	0,95	1,77	2,04	1,53	1,26
1024	0,99	1,79	2,05	1,55	1,27

Таблица 4

Коэффициент ускорения с применением субтайлинга (CLK)

$N \setminus n$	1	2	4	8	16
8	1,21	1,64	1,34	0	0
16	1,42	2,59	2,16	1,60	0
32	1,64	3,46	2,89	2,16	1,26
64	1,63	3,76	3,17	2,30	1,63
128	1,60	3,83	3,20	2,41	1,76
256	1,62	3,93	3,12	2,39	1,80
512	1,62	3,49	3,00	2,12	1,71
1024	1,63	3,58	3,07	2,32	1,86

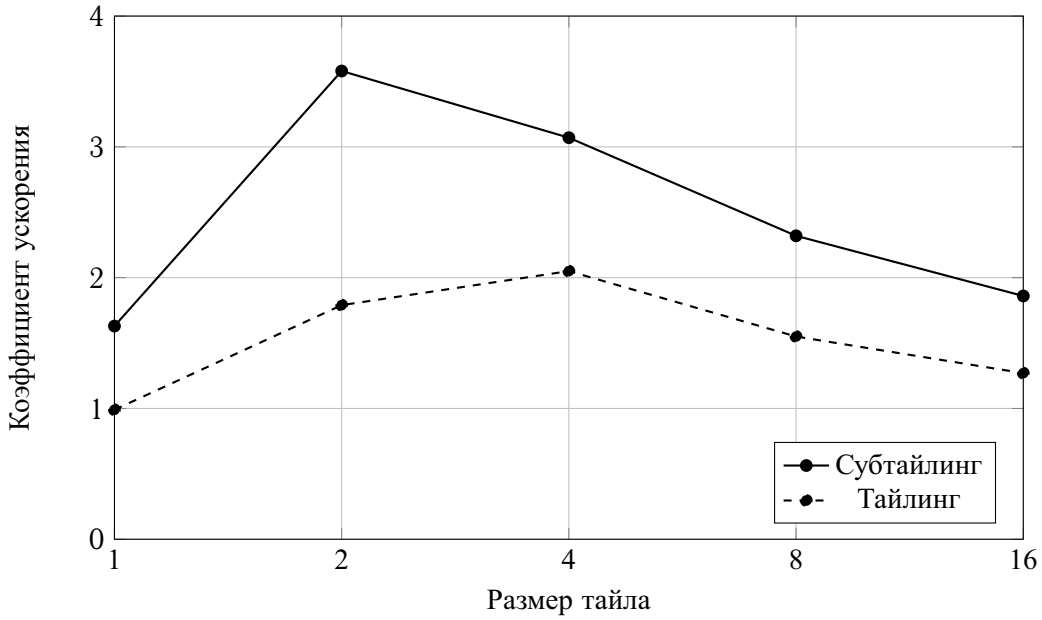

Рис. 6. *Коэффициент ускорения при $N = 1024$ (CLK)*

Расчеты были выполнены на персональном компьютере и на суперЭВМ МСЦ РАН. Полученные результаты показали ускорение метода SOR при помощи субтайлинга более чем в 5 раз по сравнению с традиционными расчетами без тайлинга, причем ускорение субтайлинга по сравнению с ускорением просто тайлингом составляет 1.5 раза. Отметим, что проведенные исследования, на наш взгляд, представляют интерес при реализации метода декомпозиции области, широко применяемого при распараллеливании решения краевых задач.

ЛИТЕРАТУРА

1. Штейнберг Б. Я., Василенко А. А., Веселовский В. В., Живых Н. А. Решатели СЛАУ с блочно-ленточными матрицами. *Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование*. 2021;14(3):106–112. DOI: 10.14529/mmp210309.
2. Rivera G., Tseng C.-W. Tiling Optimizations for 3D Scientific Computations. *Proceedings of the 2000 ACM/IEEE Conference on Supercomputing (SC '00)*. IEEE Computer Society, USA, 2000:32.
3. Свешников В. М., Климонов И. А. Применение тайлинга при решении краевых задач методом декомпозиции области. *Всероссийская конференция по математике и механике*. 2023:129–135.
4. Ammaev S. G., Gervich L. R., Steinberg B. Y. Combining Parallelization with Overlaps and Optimization of Cache Memory Usage. *Lecture Notes in Computer Science*. 2017;10421:257–264. DOI: 10.1007/978-3-319-62932-2_24.
5. Perepelkina A. Yu., Levchenko V. D. DiamondTorre Algorithm for High-Performance Wave Modeling. *Keldysh Institute Preprints*. 2015;18:1–20.