

УДК 004.272:519.67:004.942

МОДЕЛИРОВАНИЕ СЕТЕЙ РАСПРЕДЕЛЕНИЯ ПИТАНИЯ СБИС НА МНОГОЯДЕРНОМ ВЫЧИСЛИТЕЛЕ

В. Ю. Воронов¹, Н. Н. Попова¹

Представлена параллельная реализация модели сети распределения питания СБИС, ориентированная на вычислитель с многоядерными процессорами. Определение динамики потенциалов сети питания большой размерности сводится к анализу электрических цепей и численному решению обыкновенных дифференциальных уравнений. В качестве вычислительного ядра решения используются решатели разреженных систем линейных алгебраических уравнений из пакета Intel MKL. Обсуждается производительность и масштабируемость полученной реализации. Работа выполнена при поддержке компании Интел.

Ключевые слова: сеть распределения питания, обобщенный метод узловых потенциалов, решатель СЛАУ PARDISO, многоядерные процессоры, Intel Math Kernel Library.

Введение. Современные средства автоматизации проектирования сверхбольших интегральных схем (СБИС) требуют применения высокопроизводительных вычислений. Одна из важных стадий жизненного цикла проектирования интегральных схем — моделирование сети распределения питания (power distribution network).

Сеть распределения питания представляет собой регулярную металлическую структуру, служащую для подключения активных устройств интегральной схемы к источнику питания. Рост частоты переключения активных устройств, уменьшение их линейных размеров и рост количества элементов интегральной схемы усложняют задачу эффективного обеспечения питания. При моделировании структуры сети питания требуется определить падения напряжений активных устройств и потенциалы узлов сети питания. Эта задача приводит к исследованию нелинейных моделей в связи с наличием индуктивности проводов, соединяющих сеть распределения питания с источником питания. Таким образом, при переключении состояний активных устройств интегральной схемы в сети питания возникают переходные процессы. Все сказанное означает, что для разработки эффективных структур сети питания необходимы методы моделирования, позволяющие определять потенциалы в узлах сети и падения напряжения на активных устройствах на временном отрезке. Определение потенциалов на активных устройствах СБИС является составной частью алгоритмов анализа сети питания более высокого уровня. К наиболее важным из таких задач относятся анализ потенциалов в сети питания при работе устройств на максимальной нагрузке и анализ сети при возникновении неисправностей активных устройств или линий подачи питания.

Для решения задачи моделирования переходят к эквивалентной электрической цепи. Параметры сети питания на постоянном токе (анализ на постоянном токе, DC-analysis) могут быть найдены применением законов Кирхгоффа и решением полученной разреженной системы линейных алгебраических уравнений (СЛАУ). Наличие проводов в соединении сети с источником питания приводит к возникновению в сети питания переходных процессов. Эти особенности исследуются на этапе анализа переходного процесса (transient analysis), где параметры сети питания определяются из решения системы обыкновенных дифференциальных уравнений (ОДУ) с заданными начальными условиями, полученными при решении задачи на постоянном токе. Отметим, что на сегодняшний день решение ОДУ является основным подходом к численному моделированию сетей распределения питания, примером альтернативного подхода к моделированию на основе стохастических процессов является работа [1].

Анализ сетей питания реальных устройств требует решения разреженных систем линейных уравнений большой размерности (10^5 – 10^{12} неизвестных). Существуют исследования, посвященные применению различных подходов к этой задаче для достижения высокой производительности. Так, в статье [2] представлены оценки производительности моделирования сетей питания при использовании многосеточных

¹ Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, Ленинские горы, 119992, Москва; В. Ю. Воронов, аспирант, e-mail: basrav@angel.cmc.msu.ru; Н. Н. Попова, доцент, e-mail: popova@cmc.msu.ru

методов решения численной задачи. Исследование [3] посвящено применению итерационных методов решения с переобусловливателем для двухуровневых сетей питания. Отметим работы, которые применяют иерархический анализ сети питания [4] и параллельные методы решения, основанные на декомпозиции области [5].

В настоящей статье представлен результат создания параллельного симулятора сетей питания, ориентированного на вычислительные системы с многоядерными процессорами. Многоядерные процессоры стали стандартом современных вычислителей. Тенденция к дальнейшему увеличению числа ядер таких процессоров делает архитектуры с общей памятью доступной вычислительной базой для высокопроизводительных вычислений. Для таких архитектур необходимо разрабатывать параллельные приложения, учитывающие специфику и особенности вычислителей, которые позволят приложениям масштабироваться с ростом числа используемых нитей. В статье для этого предлагается использовать многонитевые реализации решателей разреженных СЛАУ из библиотеки высокопроизводительных математических операций Intel Math Kernel Library (MKL) [6].

Структура данной статьи следующая. В первом разделе представлена постановка задачи моделирования сети питания. Второй раздел посвящен вопросам численного решения поставленной задачи и реализации симулятора сетей питания для архитектуры с общей памятью. Третий раздел описывает разработанную программную реализацию. В четвертом разделе описан численный эксперимент для оценки симулятора. Пятый и шестой разделы содержат результаты вычислительного эксперимента при использовании прямых и итерационных решателей разреженной СЛАУ. В седьмом разделе представлен анализ полученных результатов, а также сравнение с другим исследованием. В заключении указаны пути дальнейшего развития исследования.

1. Задача моделирования сети питания СБИС. Для современных устройств сеть распределения питания представляет собой регулярную многоуровневую металлическую структуру (отсюда другой распространенный термин — решетка питания, power grid), нижний уровень которой подключается к активным элементам интегральной схемы, а верхний слой соединен с постоянным источником питания.

В данной работе рассматривается следующая модель сети питания.

Сеть состоит из h слоев металлизации. Слои металлизации расположены друг над другом вдоль координатной оси OZ . Каждый слой металлизации состоит из n_i линий металла, регулярно расположенных параллельно друг другу в плоскости OXY . Для четных слоев линии металла ориентированы вдоль оси OX , для нечетных — вдоль оси OY .

Каждая линия металла может быть одного из двух типов $\{V_{cc}, V_{ss}\}$, конфигурация каждого слоя металлизации задается параметрами

$$\begin{aligned} L_{CC} &= \{offset_{i_{V_{cc}}}, width_{i_{V_{cc}}}, pitch_{i_{V_{cc}}}, resistance_{i_{V_{cc}}}\}, \\ L_{SS} &= \{offset_{i_{V_{ss}}}, width_{i_{V_{ss}}}, pitch_{i_{V_{ss}}}, resistance_{i_{V_{ss}}}\}, \end{aligned}$$

где значения $offset$, $width$ и $pitch$ определяют смещение, ширину и шаг между каждой линией металла, величина $resistance$ задает удельное сопротивление металла.

Между каждыми двумя соседними слоями металлизации $(i, i + 1)$ определяются точки переходов. Так как линии металла двух соседних слоев расположены друг над другом взаимно перпендикулярно, точки перехода для двух линий металла отмечаются там, где у линий перехода одинакового типа из разных слоев совпадают абсцисса и ордината. Точки перехода соответствующих линий соединяются металлической линией перехода сопротивлением r_i .

На первом слое металлизации активные элементы соединяют ближайшие пары точек переходов линий разных типов. В рассматриваемой модели конденсатор емкости C моделирует закрытый транзистор; порядка 1% от общего числа активных элементов моделируют открытый транзистор, т.е. имеют параллельно присоединенный к конденсатору C импульсный источник тока I .

Для последнего слоя металлизации h определяется виртуальный слой $h + 1$, координаты точек перехода к которому определяют подключение к постоянному источнику напряжения V . Точку перехода $h + 1$ слоя и источник напряжения соединяет провод, имеющий заданные удельное сопротивление R и индуктивность L . Один из узлов соединения источника питания считается землей, к данному узлу присоединены все провода, идущие от линий металла типа V_{cc} из h -го слоя. Линии металла h -го слоя, имеющие тип V_{ss} , соединяются проводами с противоположным узлом источника питания.

2. Численное моделирование параметров сети питания. Для построения модели от физической модели переходят к эквивалентной электрической цепи. Наиболее известный метод построения модели — усовершенствованный узловый анализ (modified nodal analysis) [3]. Данный метод построения системы уравнений относительно неизвестных потенциалов в узлах сети основан на использовании зако-

нов Кирхгоффа.

Суть метода состоит в следующем. Для электрической схемы и описывающего ее ориентированного графа $G = (V, E)$, где множество V — узлы электрической цепи и E — элементы электрической цепи, строится матрица смежности, которая является сильно разреженной и имеет в каждом столбце только один элемент, равный 1, и только один элемент, равный -1 :

$$A = \{a_{ij}\} = \begin{cases} 1, & \text{если элемент цепи } j \text{ выходит из узла } i, \\ -1, & \text{если элемент цепи } j \text{ оканчивается в узле } i, \\ 0 & \text{иначе.} \end{cases}$$

Для электрической цепи с n узлами и b элементами выполняется первый закон Кирхгоффа

$$Ai = 0, \tag{1}$$

где $i = (i_1, \dots, i_b)^T$ — вектор значений тока на элементах цепи; второй закон Кирхгоффа:

$$v = A^T e, \tag{2}$$

где $v = (v_1, \dots, v_b)^T$ — значения напряжений на элементах цепи и $e = \{e_1, \dots, e_n\}$ — значения потенциалов в узлах цепи. Если для элементов электрической цепи нумерация выполнена так, что сначала пронумерованы все резисторы, затем конденсаторы, затем индуктивности, затем источник питания, затем источники тока, то столбцы матрицы смежности образуют матрицы смежности элементов цепи определенного типа:

$$A = [A_G A_C A_L A_V A_I], \tag{3}$$

где $A_G \in \mathbb{B}^{n \times N_G}$, $A_C \in \mathbb{B}^{n \times N_C}$, $A_L \in \mathbb{B}^{n \times N_L}$, $A_V \in \mathbb{B}^{n \times N_V}$ и $A_I \in \mathbb{B}^{n \times N_I}$. Учитывая вольт-амперные характеристики элементов электрической цепи

$$i_G = Gv = \left(\frac{1}{R}\right)v, \quad i_C = \frac{d}{dt}(Cv), \quad v_L = \frac{d}{dt}(Li), \tag{4}$$

из соотношений (1)–(4) получим

$$A_C \frac{d}{dt} C A^T e + A_R G A_R^T e + A_L i_L = -A_I I, \quad \frac{d}{dt} Li_L - A_L^T e = 0, \quad A_V^T e = E, \tag{5}$$

где $E = (E_1, \dots, E_{N_V})^T$ — значения напряжений на источниках питания, $I = (I_1, \dots, I_{N_I})^T$ — значения силы тока на источниках тока, $G = \text{diag}\left(\frac{1}{R_1}, \dots, \frac{1}{R_{N_G}}\right)$ — диагональная матрица проводимости электрической цепи, $C = \text{diag}(C_1, \dots, C_{N_C})$ — матрица емкостей и $L = \text{diag}(L_1, \dots, L_{N_L})$ — матрица индуктивностей.

Иначе говоря, (5) задает систему обыкновенных дифференциальных уравнений

$$\begin{pmatrix} A_C C A^T & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & 0 \end{pmatrix} \frac{d}{dt} \begin{pmatrix} e \\ i_L \\ i_V \end{pmatrix} + \begin{pmatrix} A_G G A_G^T & A_L & A_V \\ -A_L^T & 0 & 0 \\ A_V^T & 0 & 0 \end{pmatrix} \begin{pmatrix} e \\ i_L \\ i_V \end{pmatrix} = \begin{pmatrix} -A_I I \\ 0 \\ E \end{pmatrix}, \tag{6}$$

или в более компактной форме записи

$$\tilde{G}x + C\dot{x} = F(t), \tag{7}$$

где \tilde{G} , $C \in \mathbb{R}^{n+N_L+N_V \times n+N_L+N_V}$ и x , $F(t) \in \mathbb{R}^{n+N_L+N_V \times 1}$. В модели необходимо учесть заземление на одном из узлов соединения источника питания. По соглашению нумерация узлов сети выполняется так, что данному узлу соответствует первый номер. Это означает наличие в (6) граничного условия

$$x_1 = 0. \tag{8}$$

Проводя подстановку (8) в (6), получим, что первый столбец матриц \tilde{G} и C можно отбросить, тем самым понижая число неизвестных модели на 1. Кроме того, из уравнения отбрасывается первая компонента решения x_1 и правой части F_1 . Далее подразумевается, что этот этап выполнен.

Для численного решения (7) применяется неявный метод Эйлера. Вводя шаг времени h и производя замену $\dot{x} = \frac{x(t) - x(t-h)}{h}$, систему (7) приводим к виду

$$\left(\tilde{G} + \frac{C}{h}\right)x(t+h) = F(t+h) + \frac{C}{h}x(t), \quad (9)$$

или

$$Ax^{(i)} = b^{(i)}, \quad b^{(i)} = F(t+ih) + \frac{C}{h}x^{(i-1)}. \quad (10)$$

Начальное условие x_0 может быть получено решением задачи на постоянном токе. Для построения этой задачи в (6) конденсаторы заменяются диэлектриками, индуктивности — совершенными проводниками, тем самым формируется СЛАУ вида

$$\begin{pmatrix} A_G G A_G^T & A_V \\ A_V^T & 0 \end{pmatrix} \begin{pmatrix} e \\ i_V \end{pmatrix} = \begin{pmatrix} -A_I I \\ E \end{pmatrix}. \quad (11)$$

Таким образом, решения $x^{(0)}, \dots, x^{(n)}$ систем (10), (11) определяют решение задачи (7) в узлах $\{t_0, t_0 + h, \dots, t_0 + nh\}$.

3. Программная реализация. Для проведения моделирования сетей питания был разработан параллельный симулятор, который состоит из нескольких модулей.

1. Генератор сети питания. Данный модуль на основе входной спецификации генерирует так называемый netlist элементов электрической цепи, соответствующий физической модели. Спецификация определяет число слоев металлизации сети питания, число линий металла на каждом слое и удельные физические характеристики элементов сети. Применяемый формат хранения выходной электрической цепи является подмножеством распространенного формата SPICE, что позволяет работать с электрической цепью из других симуляторов.

2. Модуль анализа электрической цепи и формирования численной задачи. Данный модуль выполняет считывание файла электрической цепи и для выбранного шага по времени h формирует СЛАУ для задачи на постоянном токе (11) и задачу анализа переходного процесса (10). Модуль сохраняет на диск сформированные разреженные матрицы СЛАУ в распространенном формате Compressed Sparse Row [7].

3. Модуль решения задачи на основе решателей разреженных СЛАУ из пакета Intel Math Kernel Library. Данный модуль осуществляет чтение файлов СЛАУ и выполняет параллельное решение задач на постоянном токе и на заданном числе шагов по времени анализа переходного процесса.

В качестве решателей разреженной СЛАУ в работе используются прямой разреженный решатель PARDISO [8] и итерационные решатели GMRES с переобусловливателем, основанным на неполной факторизации (ILU) [9]. PARDISO является прямым решателем, в основе которого лежат панельная LU-факторизация и выбор ведущего элемента, факторизация Холецкого, алгоритмы переупорядочивания и алгоритмы прямой и обратной подстановки для определения решения СЛАУ. Применение прямого решателя имеет ряд сложностей с точки зрения эффективного использования ресурсов вычислительной системы. Так, в рамках решателя PARDISO параллельно выполняется факторизация и выбор ведущего элемента, фазы переупорядочивания элементов матрицы и решения задачи реализованы последовательно. Следует отметить, что в других широко распространенных параллельных прямых решателях для архитектур с общей памятью (таких как SuperLU [10] и WSMP [11]) эти фазы решения также выполняются последовательно.

При использовании итерационных решателей требования на оперативную память существенно ниже, чем для прямого решателя. Однако для достижения высокой скорости сходимости необходимо настраивать параметры итерационного метода и переобусловливателя.

Отметим также, что в рассматриваемой постановке задачи при выборе постоянного шага по времени в (9) вид матрицы A в задаче (10) не изменяется. Таким образом, для анализа переходного процесса необходимо выполнить факторизацию матрицы A только на первом временном шаге, а на дальнейших шагах — использовать факторизацию для получения решения. То же самое относится и к переобусловливателю, который следует создавать на первом шаге анализа переходного процесса и использовать для

Таблица 1

Описание параметров сетей питания эксперимента и параметров соответствующих СЛАУ

Модель	число узлов	nC	nR	nL	nI	n	nnz
n500	300661	124750	466182	1278	11	300661	984025
n500_DC	300650	0	124750	0	0	300672	1233574
n1000	1201972	499500	867851	5082	33	1201972	4938800
n1000_DC	1201939	0	867851	0	0	1201939	3939641
n1500	2703746	1124250	4202347	11346	60	2703746	11110158
n1500_DC	2703686	0	4202347	0	0	2703686	8861364
n2000	4806369	1999000	7474170	20226	127	4806369	19757145
n2000_DC	4806115	0	7474170	0	0	4806242	15758516
n3000	10810870	4498500	16815922	45544	263	10810870	44446590
n3000_DC	10810607	0	16815922	0	0	10810607	35448281

последующих шагов по времени. Эти особенности приводят к значительному сокращению временных затрат и вычислительной сложности моделирования.

4. Описание численного эксперимента. Для оценки производительности разработанного симулятора был проведен численный эксперимент. Эксперимент проводился на наборах сетей питания, основные параметры которых представлены в табл. 1. В таблице использованы следующие обозначения: nC — число конденсаторов сети питания, nR — число резистров, nL — число индуктивностей, nI — число источников тока. Кроме того, указаны характеристики СЛАУ задачи, где n — размерность СЛАУ и nnz — число ненулевых элементов матрицы СЛАУ.

Отметим, что размерность рассматриваемых сетей питания (10^7 узлов) приблизительно соответствует функциональным устройствам современного процессора (для сравнения, процессор, изготовленный по технологическому процессу $180 \mu m$, имеет в сети питания порядка 2×10^8 узлов). Заметим, что максимальная размерность моделей для проведения вычислительного эксперимента ограничивалась ресурсами, требуемыми для генерации сети питания. Формирование СЛАУ (11) и (10) требует выполнения операций умножения вида “матрица–матрица” и “матрица–вектор”, транспонирования матриц, конкатенации нескольких матриц друг с другом. С ростом размерности сети питания для выполнения этих операций ограничением является объем оперативной памяти вычислительной системы, что требует разработки параллельных алгоритмов формирования численной задачи на нескольких узлах.

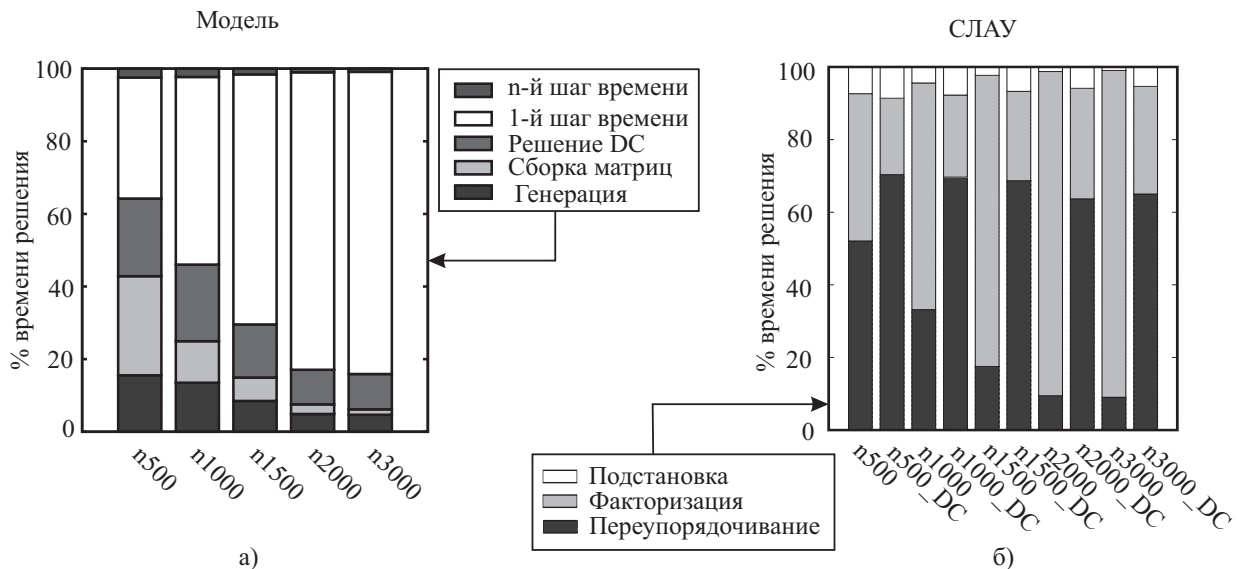


Рис. 1. Доля времени этапов решения задачи в общем времени моделирования (а) и распределение времени решения СЛАУ по этапам решения при использовании решателя PARDISO (б). СЛАУ с суффиксом _DC соответствуют задаче на постоянном токе

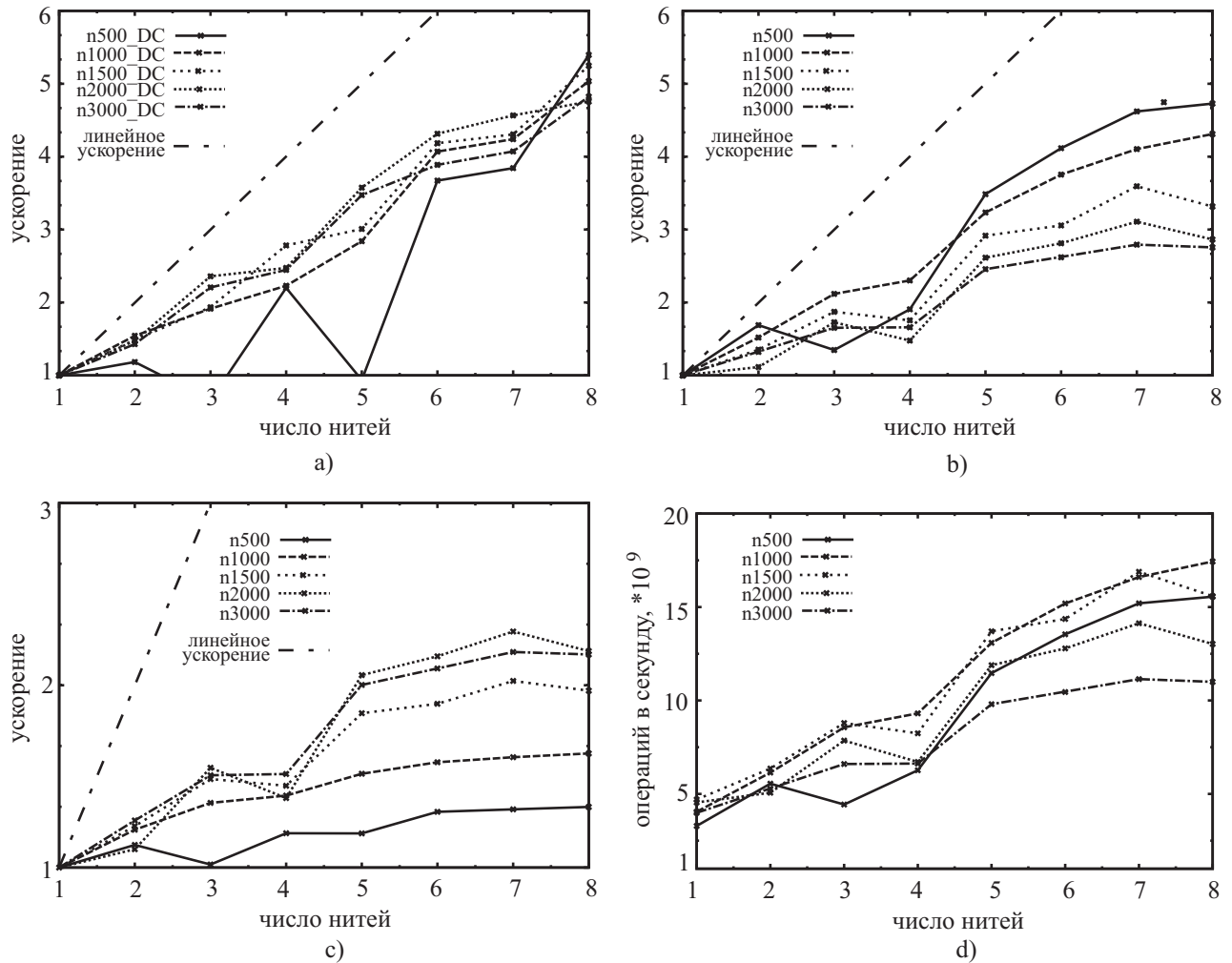


Рис. 2. Ускорение решателя PARDISO с ростом числа нитей: ускорение при факторизации СЛАУ для постоянного тока (а), ускорение при факторизации СЛАУ переходного процесса (б), ускорение при решении СЛАУ переходного процесса (с), увеличение GFLOPS при факторизации СЛАУ переходного процесса (д)

Эксперимент проводился на вычислительном узле платформы СКИФ-МГУ "Чебышев" с двумя четырехъядерными процессорами 2 x Intel Xeon 5472 3.0GHz 64Kb L1 Cache, 4Mb L2 Cache, 8Gb RAM. В рассматриваемой модели временной отрезок моделирования составляет 10^{-9} с. Переключение активного элемента моделируется импульсом тока на источниках тока первого слоя. Импульс тока имеет форму равнобедренного треугольника длительностью 10^{-12} с и с максимальным значением 10^{-3} А. Шаг по времени моделирования равен 10^{-12} с; таким образом, решение задачи определения потенциалов в узлах сети питания при заданных параметрах сводится к решению одной задачи на постоянном токе (11) и 1000 шагов анализа переходного процесса (10). На рис. 1а представлены результаты анализа производительности симулятора по этапам работы. С ростом размерности модели сети питания доля времени генерации сети питания и формирования матриц численных задач уменьшается вплоть до 13% от времени всего моделирования. Таким образом, основная доля времени приходится на численное решение разреженных систем линейных уравнений задач на постоянном токе и анализа переходного процесса. В данной работе выполняется распараллеливание именно этих этапов, поскольку они вносят наиболее существенный вклад во временные затраты моделирования сети питания. Если отдельно рассмотреть этапы решения СЛАУ, соответствующей задаче на постоянном токе, и решение СЛАУ для одного шага времени анализа переходного процесса, то при использовании прямого решателя PARDISO в общем времени решения доминирует операция факторизации СЛАУ (рис. 1б). Эта операция может занимать до 80% времени решения СЛАУ. При использовании итерационных решателей в общем времени решения СЛАУ наибольшую долю времени занимает составление переобусловливателя: эта доля достигает 45%.

Далее представлены результаты анализа производительности симулятора в зависимости от исполь-

зуюемого решателя СЛАУ.

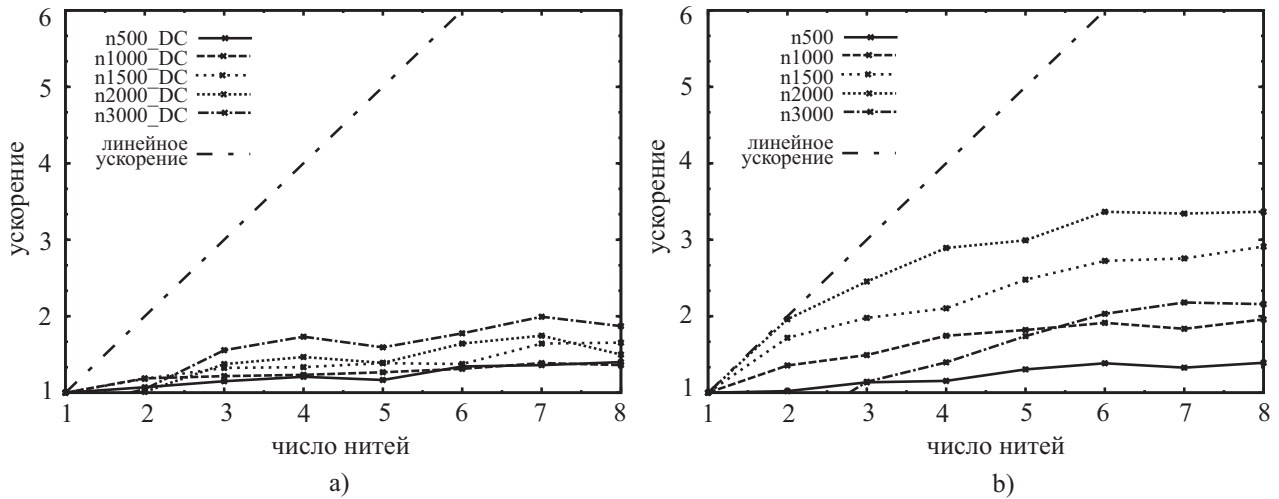


Рис. 3. Данные по ускорению итерационного решателя GMRES(35)+ILU(0): решение СЛАУ для постоянного тока (а), решение СЛАУ переходного процесса (б)

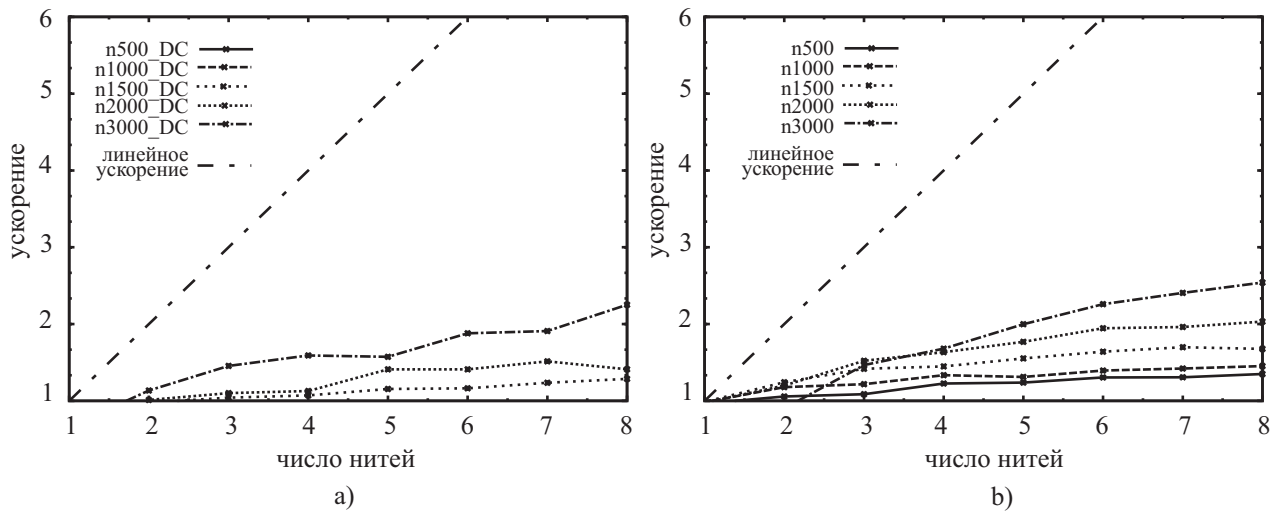


Рис. 4. Данные по ускорению итерационного решателя GMRES(35)+ILUT: решение СЛАУ для постоянного тока (а), решение СЛАУ переходного процесса (б)

5. Результаты производительности при использовании решателя PARDISO. Как уже отмечалось, в решателе PARDISO распараллелен алгоритм факторизации СЛАУ, поэтому будут представлены результаты, в основном касающиеся этой стадии решения СЛАУ. Ускорение в разработанном симуляторе при росте числа нитей представлено на рис. 2. Как видно из указанных графиков, этап факторизации матрицы СЛАУ переходного процесса допускает ускорение до 4,8 раз при переходе на 8 нитей. При этом ускорение всего решения СЛАУ переходного процесса снижается до 2,4 раз. Время решения СЛАУ для постоянного тока в среднем в 15 раз меньше времени решения СЛАУ переходного процесса, поэтому его можно считать несущественным. Отметим, что ускорение факторизации СЛАУ для постоянного тока составляет до 5,2 раз.

Поскольку решатель PARDISO собирает статистику о числе операций с плавающей точкой, эта информация также приведена на рис. 2d. Пиковая производительность вычислительной системы составляет порядка 162 GFLOPS, или 20 GFLOPS для одного ядра. Запуск для одной нити показывает загрузку ядра приблизительно на 25% от пиковой. При использовании 8 ядер процессора общая производительность факторизации увеличивается приблизительно втрое и составляет приблизительно 13% от пиковой.

6. Результаты моделирования с использованием итерационных решателей. Обратимся к итерационным решателям СЛАУ. Для вычислительного эксперимента максимальное количество ите-

раций было ограничено 1000, критерий получения решения — достижение величины допуска нормы невязки $\|r_i\| < 10^{-6}$. Начальное приближение решения для задачи на постоянном токе было выбрано $x_0 = (1, \dots, 1)^T$, на каждом шаге анализа переходного процесса начальное приближение выбиралось равным решению на предыдущем шаге по времени. На рис. 3 представлено ускорение итерационного метода GMRES с использованием переобусловливателей ILU(0). Максимальная величина ускорения решения СЛАУ переходного процесса составила 3,27 раз, решения СЛАУ для постоянного тока — 2,03 раз. Ускорение решения при использовании итерационного метода GMRES и переобусловливателя ILUT представлено на рис. 4. Для переобусловливателя были заданы следующие параметры: уровень заполнения $f = 4,0$, порог отсечения $T = 10^{-6}$. В результате наибольшая величина ускорения для решения СЛАУ переходного процесса составила 3,35 раза, а в случае СЛАУ для постоянного тока — 2,28 раз.

7. Анализ результатов апробации симулятора. Подведем итоги по представленным результатам производительности моделирования при использовании решателей MKL. Решатель PARDISO в сравнении с итерационными решателями показывает более высокую масштабируемость. При этом необходимо понимать, что ускорение при использовании решателя PARDISO получено за счет этапа факторизации. При моделировании сотен шагов по времени для переходного процесса решающий вклад в общее время моделирования будут иметь этапы прямой и обратной подстановки для определения решения на основе полученного разложения матрицы СЛАУ. Таким образом, в данной постановке задачи моделирования время факторизации матрицы пренебрежимо мало в сравнении со временем подстановки решения для сотен и тысяч шагов по времени переходного процесса.

В отличие от прямого решателя, ускорение, полученное для итерационных решателей, будет достигаться при решении СЛАУ на каждом временном шаге переходного процесса. Таким образом, его значения могут быть распространены на все время моделирования, и в данной постановке задачи моделирования использование итерационного решателя более предпочтительно.

Эти выводы верны при моделировании сетей питания, приводящем к стационарной на шагах по времени матрице A в (10). При рассмотрении моделей, приводящих к нестационарной матрице A_i на каждом шаге, ускорение решателя PARDISO будет достигаться на каждом шаге. Такая же картина будет верна и для итерационных решателей. Таким образом, выбор прямого либо итерационного решателя будет диктоваться в зависимости от рассматриваемой модели сети питания и требуемого количества шагов по времени.

Таблица 2

Пиковый объем оперативной памяти при моделировании сетей питания с использованием решателя PARDISO и итерационных решателей

Модель	решатель PARDISO		решатель GMRES + ILU(0)		решатель GMRES + ILUT	
	переходный процесс (Мб)	постоянный ток (Мб)	переходный процесс (Мб)	постоянный ток (Мб)	переходный процесс (Мб)	постоянный ток (Мб)
n500	85.792	42.283	57.855	41.617	53.941	44.199
n1000	466.852	215.308	228.234	163.316	211.715	172.769
n1500	1435.057	547.884	512.066	366.035	474.586	386.961
n2000	3243.633	1079.778	909.613	649.988	842.695	686.918
n3000	6800.104	2577.417	1591.304	1460.800	1471.281	1251.221

Кроме того, на выбор решателя СЛАУ существенно влияет доступный на вычислительной системе объем оперативной памяти. В табл. 2 представлена статистика по необходимому объему оперативной памяти в зависимости от типа решателя СЛАУ, применяемого при моделировании сетей питания. Требования к объему памяти решателя PARDISO в несколько раз выше, чем для итерационных решателей. В случае применения прямых решателей основной объем оперативной памяти приходится на операции с элементами заполнения, которые появились в матрицах L и U после факторизации. Число таких элементов многократно превышает число ненулевых значений в исходной матрице. Объем оперативной памяти вычислителя ограничивает максимальный размер сети питания, для которой может быть эффективно выполнено моделирование.

Результаты сравнения времени выполнения этапов моделирования для самой большой модели сети питания представлены в табл. 3. Время решения указано в секундах. Заметим, что время решения с помощью PARDISO одной СЛАУ многократно больше, чем для итерационных решателей. Ускорение при

Таблица 3

Время моделирования сети питания $n=3000$ по этапам

Тип решателя	Число нитей			
	1 нить	2 нити	4 нити	8 нитей
Решение задачи на постоянном токе				
PARDISO	137,004	125,185	113,519	105,841
GMRES+ILU(0)	4,052	3,575	2,528	2,4362
GMRES+ILUT	4,526	3,591	3,062	2,546
1 шаг анализа переходного процесса				
PARDISO	1286,361	779,996	585,069	554,076
GMRES+ILU(0)	33,092	29,364	25,423	18,590
GMRES+ILUT	24,280	20,590	16,880	14,490
1000 шагов анализа переходного процесса				
PARDISO	12271,871	12193,839	12151,397	11962,988
GMRES+ILU(0)	14752,374	11139,497	7332,261	5806,912
GMRES+ILUT	14816,243	11227,561	7430,267	5427,334

росте числа нитей PARDISO в этом случае происходит за счет многонитевой реализации факторизации, в то время как создание переобусловливателя итерационного решателя в MKL реализовано только последовательно.

При решении задачи анализа переходного процесса для 1000 шагов времени оба метода решения дают приблизительно сопоставимое время работы. Как уже отмечалось выше, алгоритмы обратной подстановки в PARDISO являются последовательными, что объясняет практически отсутствующее ускорение с ростом числа нитей для этого решателя (всего 1,03 раз). Напротив, итерационные решатели MKL являются многонитевыми, и с ростом числа нитей суммарное время решения задач переходного процесса ускоряется приблизительно в 2,7 раза.

В заключение приведем результаты сравнения полученного симулятора с результатами из других существующих исследований. Заметим, что результаты по данной тематике с использованием многонитевого параллелизма на общей памяти получены впервые.

Сравнение полученной реализации симулятора было выполнено с результатами работы [5] как наиболее близкой к нашей работе по рассматриваемым данным и используемой вычислительной системе. В этой работе, как уже отмечалось во введении, предложено использовать методы декомпозиции области для решения задачи моделирования сети питания на архитектурах с распределенной памятью. Эксперимент в работе проводится для одноуровневых сетей питания размерностью порядка 4 и 10 миллионов узлов на кластере из четырех узлов, с AMD Opteron 146 2.0 GHz 4 Gb RAM на каждом узле. В табл. 4 представлены результаты производительности моделирования сетей питания для дальнейшего сопоставления. Время решения указано в секундах. Приведенные в таблице аббревиатуры методов решения СЛАУ соответствуют следующим методам:

- 1) DD+LU — декомпозиция области, в каждой области применяется LU-разложение из пакета SuperLU и обратная подстановка;
- 2) DD+IF — декомпозиция области, в каждой области используется итерационный решатель с переобусловливателем неполной факторизации;

Таблица 4

Производительность методов декомпозиции области для моделирования сети питания

Метод решения	Размерность модели	
	4×10^6 узлов	10^7 узлов
задача на постоянном токе		
DD+LU	157,20	449,32
DD+IF	4340,94	—
DD+MG	178,43	653,17
AS	234,73	—
20 шагов переходного процесса		
DD+LU	183,80	517,21
DD+IF	13297,53	—
DD+MG	376,11	1377,16
AS	1543,72	—

3) DD+MG — декомпозиция области, в каждой области используется итерационный метод с многосеточным переобусловливателем; реализация методов взята из параллельного пакета Trilinos [12];

4) AS — итерационный метод решения без декомпозиции области, с аддитивным переобусловливателем Шварца.

Отметим, что сопоставление результатов табл. 3 и 4 не может быть сделано напрямую ввиду того, что условия экспериментов в работах существенно различаются. Тем не менее, сравнение результатов показательно в том плане, что полученный в нашей работе симулятор имеет производительность по меньшей мере не хуже представленного в [5] при моделировании сетей питания приблизительно одинаковой размерности. В то же время, моделирование на платформах с распределенной памятью позволяет рассматривать сети питания большей размерности, чем в нашей работе.

Заключение. Представленные результаты позволяют оценить эффективность использования многоядерных вычислителей для решения задачи. Получены оценки размерности сетей питания, которые эффективно моделируются на одной вычислительной платформе с общей памятью. Кроме того, указаны предполагаемые особенности эффективности использования ресурсов вычислительной системы при усложнении модели сети питания.

Дальнейшее развитие работы заключается в следующем. Для моделирования сетей питания размерности 10^8 – 10^{12} узлов необходимо переходить к массивно-параллельным архитектурам. Для этого предлагается использовать решатели СЛАУ из библиотек научных вычислений, ориентированных на архитектуры с распределенной памятью. Перспективным направлением также является предложение гибридных вычислительных моделей вида “MPI + нитевой параллелизм” для рассматриваемой задачи. При этом программная реализация, представленная в данной статье, может использоваться как общий алгоритм для решения локальных подзадач в рамках памяти одного узла. В этом смысле перспективным является создание симулятора для массивно-параллельной платформы с многоядерными процессорами на узлах на основе метода декомпозиции области. Переход к сетям питания большой размерности также требует параллельной реализации формирования численных задач, поскольку память одного узла является существенным ограничителем для формирования СЛАУ (11) и (10).

Авторы благодарят К. К. Малинаускаса, К. А. Трушина и А. В. Жмурина за обсуждение работы и ценные замечания.

СПИСОК ЛИТЕРАТУРЫ

1. Qian H., Nassif S., Sapatnekar S. Power grid analysis using random walks // IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems. 2005. **24**, N 8. 1204–1224.
2. Kozhaya J., Nassif S., Najm F. A multigrid-like technique for power grid analysis // IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems. 2002. **21**, N 10. 1148–1160.
3. Chen T., Chen C. Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods // Proc. Design Automation Conference. Las Vegas, 2001. 559–562.
4. Zhao M., Panda R.V., Sapatnekar S.S., Blaauw D. Hierarchical analysis of power distribution networks // IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems. 2002. **21**, N 2. 159–168.
5. Sun K., Zhou Q., Mohanram K., Sorensen D. Parallel domain decomposition for simulation of large-scale power grids // Proc. of the 2007 IEEE/ACM International Conference on Computer-Aided Design. San Jose, 2007. 54–59.
6. <http://www.intel.com/cd/software/products/asm-na/eng/307757.htm>
7. Баладин М.Ю., Чепурина Э.П. Методы решения СЛАУ большой размерности. Новосибирск: Изд-во НГТУ, 2000.
8. Schenk O., Gärtner K. Solving unsymmetric sparse systems of linear equations with PARDISO // Future Generation of Computer Systems. 2004. **20**, N 3. 475–487.
9. Ильин В.П. Методы неполной факторизации для решения алгебраических систем. М: Наука, Физматлит, 1995.
10. Li X. An overview of SuperLU: algorithms, implementation and user interface // ACM Trans. on Mathematical Software (TOMS). 2005. **31**, N 3. 302–325.
11. Gupta A., Joshi M., Kumar V. WSMP: a high-performance shared and distributed-memory parallel sparse linear equation solver // Tech. Report. University of Minnesota and IBM Thomas J. Watson Research Center, 2001.
12. Heroux M., Phipps E., Salinger A., Thornquist H., et al. An overview of the Trilinos project // ACM Trans. on Mathematical Software (TOMS). 2005. **31**, N 3. 397–423.

Поступила в редакцию
23.01.2009