Учредитель — Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Южно-Уральский государственный университет» (национальный исследовательский университет)

Основной целью издания является пропаганда научных исследований в следующих областях:

- Вычислительная математика и численные методы
- Математическое программирование
- Распознавание образов
- Вычислительные методы линейной алгебры
- Решение обратных и некорректно поставленных задач
- Доказательные вычисления
- Численное решение дифференциальных и интегральных уравнений
- Исследование операций
- Теория игр
- Теория аппроксимации
- Информатика
- Математическое и программное обеспечение высокопроизводительных вычислительных систем
- Системное программирование
- Распределенные вычисления, облачные и грид-технологии
- Технология программирования
- Машинная графика
- Интернет-технологии
- Системы электронного обучения
- Технологии обработки баз данных и знаний
- Интеллектуальный анализ данных

South Ural State University

The main purpose of the series is publicity of scientific researches in the following areas:

- Numerical analysis and methods
- Mathematical optimization
- Pattern recognition
- Numerical methods of linear algebra
- Reverse and ill-posed problems solution
- Computer-assisted proofs
- Numerical solutions of differential and integral equations
- Operations research
- Game theory
- Approximation theory
- Computer science
- High performance computer software
- System programming
- Distributed, cloud and grid computing
- Programming technology
- Computer graphics
- Internet technologies
- E-learning
- Database and knowledge processing
- Data mining

Содержание

Contents

УДК 519.63

АЛГОРИТМЫ РЕШЕНИЯ СЛАУ НА СИСТЕМАХ С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ В ПРИМЕНЕНИИ К ЗАДАЧАМ ЭЛЕКТРОМАГНЕТИЗМА[1]

Д.С. Бутюгин

Рассматриваются различные аспекты моделирования гармонических электромагнитных полей на кластерах. Основная вычислительная сложность задачи заключается в решении систем линейных алгебраических уравнений (СЛАУ), возникающих в результате конечно-элементных аппроксимаций соответствующих краевых задач электромагнетизма элементами Неделека различных порядков. Рассмотрены эффективные и экономичные подходы к декомпозиции расчетной области и матрицы системы. Решение распределенных СЛАУ осуществляется итерационными методами в подпространствах Крылова с использованием аддитивного метода Шварца в качестве предобуславливателя. Для повышения эффективности алгоритмов итерации осуществляются в подпространствах следов. Реализованные решатели используют MPI для организации обмена данными. Решение систем в подобластях осуществляется при помощи прямого решателя PARDISO из библиотеки Intel® MKL. Результаты серии численных экспериментов на модельных и практических задачах демонстрируют эффективность предлагаемых алгоритмов.

Ключевые слова: уравнения Максвелла, итерационные алгоритмы, методы декомпозиции подобластей, аддитивный метод Шварца.

Введение

Многие актуальные задачи вычислительной математики требуют решения разреженных СЛАУ высоких порядков. Задача вычисления трехмерного электромагнитного поля в частотной области возникает при расчетах различных волновых устройств, решении задач геоэлектроразведки, таких как электромагнитного каротажа, и других. Требования к точности получаемого решения задач с разномасштабными объектами приводят к необходимости построения сеток с числом конечных элементов до $10^5 \div 10^7$ и порядком получаемых после аппроксимации СЛАУ до $10^7 \div 10^9$. Решение таких СЛАУ невозможно без использования вычислительной мощности кластеров. Это требует подходящих алгоритмов решения систем алгебраических уравнений для машин с распределенной памятью.

В данной работе рассматриваются алгоритмы решения СЛАУ, основанные на аддитивном методе Шварца. Рассмотрено два подхода к декомпозиции задачи. Первый из подходов основан на геометрическом разбиении сетки на связанные подобласти и позволяет построить декомпозицию с пересечениями подобластей. Второй подход основан на переупорядочивании графа матрицы системы. Этот способ представляет из себя вариант одно-направленного разбиения графа и приводит к СЛАУ блочно-трехдиагонального вида. Отличительной особенностью предлагаемых методов является низкая асимптотическая вычислительная сложность алгоритмов, в отличие от вариантов метода вложенных сечений (например, реализованного в библиотеке METIS [1]).

[1]Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012»

Решение полученных систем уравнений осуществляется методами в подпространствах Крылова [2], такими как обобщенный метод минимальных невязок (GMRES). В качестве предобуславливателя выступает аддитивный метод Шварца. Использование такого подхода, в отличие от использования метода Шварца напрямую, позволяет решить более широкий класс задач и повысить скорость сходимости итерационного процесса.

Структура данной работы следующая. В разделе 1 описывается математическая постановка задачи, в разделе 2 — дискретная постановка. В разделе 3 описаны предлагаемые методы декомпозиции. Раздел 4 содержит описание используемых итерационных алгоритмов. В разделе 5 приведены результаты численных экспериментов. Наконец, в последнем разделе обсуждаются полученные в работе результаты.

1. Математическая постановка задачи

Электромагнитное поле с гармонической зависимостью от времени в случае линейности задачи при отсутствии внешнего заряда и магнитной проводимости может быть описано следующей формой системы уравнений Максвелла:

$$\begin{array}{rclcrcl} \nabla \times \vec{E} & = & -i\omega\mu\vec{H}, & \qquad & \nabla \cdot (\varepsilon_r \vec{E}) & = & \rho/\varepsilon_0, \\ \nabla \times \vec{H} & = & i\omega\dot{\varepsilon}\vec{E} + \vec{J}, & & \nabla \cdot (\mu_r \vec{H}) & = & 0, \end{array} \tag{1}$$

где электрическая и магнитная проницаемости имеют вид

$$\dot{\varepsilon} \;=\; \varepsilon_0\varepsilon_r - i\sigma^e/\omega, \qquad \mu \;=\; \mu_0\mu_r - i\sigma^m/\omega.$$

Здесь i — мнимая единица, $\vec{E}$ и $\vec{H}$ — векторы напряженности электрического и магнитного полей соответственно, ω — круговая частота решения, $\vec{J}$ и ρ — плотности внешнего тока и объемного заряда соответственно, ε_0 и μ_0 — диэлектрическая и магнитная проницаемости вакуума, ε_r и μ_r — относительная диэлектрическая и магнитная проницаемости среды, а σ^e и σ^m — электрическая и магнитная проводимости.

Используя предположения о линейности задачи (то есть независимости параметров среды от электромагнитного поля), а также отсутствие объемного внешнего заряда ρ и магнитной проводимости $\sigma^m = 0$, уравнения (1) можно привести к форме комплексного векторного уравнения Гельмгольца

$$\nabla \times \left(\mu_r^{-1}\nabla \times \vec{E}\right) - k_0^2\dot{\varepsilon}_r\vec{E} = -ik_0Z_0\vec{J}. \tag{2}$$

Здесь $k_0 = \omega\sqrt{\varepsilon_0\mu_0}$, $Z_0 = \sqrt{\mu_0/\varepsilon_0}$ и $\dot{\varepsilon}_r = \dot{\varepsilon}/\varepsilon_0$. Параметры среды полагаются кусочно-постоянными. Мы будем предполагать, что k_0 не является Максвелловским собственным числом, то есть рабочая частота ω не является резонансной. Стоит отметить, что в случае, если хотя бы в одной подобласти $\sigma^e > 0$, то рабочая частота может быть любой, поскольку в системе отсутствуют резонансы [3].

Решение ищется в области Ω с границей $S = S_1 \cup S_2$, на каждой из частей которой поставлено одно из следующих граничных условий:

$$\vec{n} \times \vec{E} \;\Big|_{S_1} = \; \vec{n} \times \vec{E}_0, \qquad \vec{n} \times \vec{H} \;\Big|_{S_2} = \; 0, \tag{3}$$

где $\vec{n}$ — внешняя нормаль к границе. При $\vec{E}_0 = 0$ первое из условий соответствует идеальному электрическому проводнику, при $\vec{E}_0 \neq 0$ — волновому входу, а второе

условие соответствует идеальному магнитному проводнику. Полагая область Ω состоящей из подобластей $\Omega = \bigcup \Omega_k$, в каждой из которых физические параметры среды являются константными, введем на каждой из внутренних границ Γ с нормалью $\vec{n}$ условия сопряжения

$$\begin{array}{rclcrcl} \vec{n} \cdot (\dot{\varepsilon}_1 \vec{E}_1 - \dot{\varepsilon}_1 \vec{E}_2) & = & 0, & \qquad & \vec{n} \times (\vec{E}_1 - \vec{E}_2) & = & 0, \\ \vec{n} \cdot (\mu_1 \vec{H}_1 - \mu_2 \vec{H}_2) & = & 0, & & \vec{n} \times (\vec{H}_1 - \vec{H}_2) & = & 0. \end{array} \tag{4}$$

Вводятся стандартные соболевские пространства и подмножества:

$$H^{\mathrm{rot}} = \left\{ \vec{\psi} \in [L^2(\Omega)]^3 : \nabla \times \vec{\psi} \in [L^2(\Omega)]^3 \right\},$$
$$H_0^{\mathrm{rot}} = \left\{ \vec{\psi} \in H^{\mathrm{rot}} : \vec{n} \times \vec{\psi}\big|_{S_1} = 0 \right\},\ H_S^{\mathrm{rot}} = \left\{ \vec{\psi} \in H^{\mathrm{rot}} : \vec{n} \times \vec{\psi}\big|_{S_1} = \vec{n} \times \vec{E}_0 \right\}.$$

Вариационная формулировка уравнения (2) с краевыми условиями (3) в форме Галёркина имеет следующий вид (см. [4]): найти такое $\vec{E} \in H_S^{\mathrm{rot}}$, что для всех $\vec{\psi} \in H_0^{\mathrm{rot}}$ выполнено

$$\int_\Omega \mu_r^{-1} \left(\nabla \times \vec{E} \right) \cdot \left(\nabla \times \vec{\psi} \right) \mathrm{d}\Omega - k_0^2 \int_\Omega \dot{\varepsilon}_r \left(\vec{E} \cdot \vec{\psi} \right) \mathrm{d}\Omega = -ik_0 Z_0 \int_\Omega \vec{J} \cdot \vec{\psi} \mathrm{d}\Omega. \tag{5}$$

2. Дискретная постановка задачи

В работе используются неструктурированные тетраэдральные сетки. Рассматривается соответствующее разбиение расчетной области Ω на непересекающиеся тетраэдральные элементы $\Omega = \bigcup \Omega_T$. В каждом из тетраэдров вводятся базисные функции, соответствующие его степеням свободы. Пусть $\mathcal{W}_l$ — конечномерное пространство базисных функций порядка не выше l, конформное H^{rot}. В работе рассматриваются иерархические базисные функции, предложенные в [5]:

$$\begin{array}{rcl} \mathcal{W}_l & = & \tilde{\mathcal{W}}_1 \oplus \tilde{\mathcal{W}}_2 \oplus \ldots \oplus \tilde{\mathcal{W}}_l, \\ \tilde{\mathcal{W}}_1 & = & \tilde{A}_1, \quad \tilde{\mathcal{W}}_i = \tilde{\mathcal{A}}_i \oplus \nabla \tilde{\mathcal{V}}_i, \end{array}$$

где $\tilde{\mathcal{W}}_i$ — инкрементальное подпространство с базисными функциями порядка i, $\tilde{\mathcal{V}}_i$ — инкрементальное подпространство со скалярными базисными функциями, конформными H^1, а $\tilde{\mathcal{A}}_i$ — инкрементальное подпространство с роторными базисными функциями.

Подпространства $\mathcal{V}_{l,0}$ и $\mathcal{W}_{l,0}$ вводятся естественным образом, а подмножество $\mathcal{W}_{l,S}$ — как множество функций, у которых коэффициенты разложения u^S по базисным функциям с ненулевым следом на S_1 принимают фиксированные значения, соответствующие краевому условию (3).

Приближенное решение $\vec{E}^h$ будем искать в виде

$$\vec{E}^h = \sum_j u_j^S \vec{\psi}_j^S + \sum_i u_i \vec{\psi}_i^0. \tag{6}$$

Вводятся матрицы соответствующих билинейных форм:

$$M_{i,j} = \int_\Omega \dot{\varepsilon}_r \left(\vec{\psi}_j^0 \cdot \vec{\psi}_i^0 \right) \mathrm{d}\Omega, \quad S_{i,j} = \int_\Omega \mu_r^{-1} \left(\nabla \times \vec{\psi}_j^0 \right) \cdot \left(\nabla \times \vec{\psi}_i^0 \right) \mathrm{d}\Omega,$$

где базисные функции $\vec{\psi}_i^0, \vec{\psi}_j^0 \in \mathcal{W}_{l,0}$. Вектор правой части определяется как

$$f_i = -ik_0 Z_0 \int_\Omega \vec{J} \cdot \vec{\psi}_i^0 \mathrm{d}\Omega + \sum_j u_j^S \int_\Omega \left(\mu_r^{-1} \left(\nabla \times \vec{\psi}_j^S \right) \cdot \left(\nabla \times \vec{\psi}_i^0 \right) - k_0^2 \dot{\varepsilon}_r \vec{\psi}_j^S \cdot \vec{\psi}_i^0 \right) \mathrm{d}\Omega.$$

Тогда итоговая система принимает вид

$$\left[S - k_0^2 M \right] u = f. \tag{7}$$

Для вычисления элементов матрицы и вектора правой части можно воспользоваться поэлементной технологией сборки [6], заменив интегрирование по расчетной области Ω суммой интегралов по каждому из тетраэдров и вычислением в каждом из тетраэдров локальных матриц и векторов.

3. Методы декомпозиции на подобласти

Для решения задачи на системах с распределенной памятью требуется распределить данные между узлами системы, по возможности минимизируя объем коммуникационных данных. Основной проблемой здесь является то, что алгоритмы, генерирующие разбиения высокого качества, сами требуют больших вычислительных ресурсов. Поэтому, как правило, приходится идти на компромисс и выбирать алгоритмы, позволяющие получить достаточно хорошее разбиение за приемлемое время.

3.1. Геометрическая декомпозиция расчетной области

Предлагаемый в работе алгоритм геометрической декомпозиции расчетной области является упрощенным вариантом алгоритма построения BSP-дерева (binary space partitioning, двоичное разбиение пространства). Метод двоичного разбиения — это метод рекурсивного разбиения пространства на выпуклые множества гиперплоскостями [7]. Получающаяся в результате структура данных находит широкое применение в различных областях компьютерной графики для проверки пересечений, определения прямой видимости объектов и др. Стоит отметить, что построение оптимального BSP дерева представляет из себя очень трудоемкую задачу, поэтому, в большинстве случаев, ограничиваются субоптимальными деревьями.

В рамках данной работы предлагается использовать следующее упрощенное двоичное дерево. Упрощение состоит в том, что секущие плоскости предлагается проводить ортогонально осям координат. При этом мы будем каждый раз разделять имеющееся множество тетраэдров на две приблизительно равные части. Алгоритм построения разбиения будет основан на методе обработки событий. Подготовительная часть алгоритма такая:

- для x, y и z: спроектировать тетраэдры на соответствующую ось; для каждого тетраэдра получаем отрезки, соответствующие его началу и концу;
- для x, y и z: сгенерировать отсортированные по координате события — начало и конец каждого из тетраэдров.

В результате мы получим 3 отсортированных списка событий, для осей Ox, Oy и Oz соответственно.

Далее к этим спискам мы рекурсивно будем применять следующую процедуру. Будем последовательно обрабатывать события для каждой из осей, что соответствует движению секущей плоскости в сторону увеличения соответствующей координаты.

При этом мы будем подсчитывать число тетраэдров в каждом из полупространств. Тетраэдры, пересекающиеся и касающиеся плоскости сечения, будем относить к обеим частям одновременно. Схема алгоритма следующая:

- Для x, y и z:
  - left_count $\leftarrow 0$, right_count $\leftarrow T$, где T — число тетраэдров.
  - Для каждой координаты последовательно: обработать события
    - Обработать начала тетраэдров: для каждого left_count $\leftarrow$ left_count $+ 1$.
    - Проверить left_count и right_count на оптимальность.
    - Обработать концы тетраэдров: right_count $\leftarrow$ right_count $- 1$.
  - Запомнить оптимальное разбиение для оси.
- Выбрать наилучшую ось для разбиения.
- Построить разбиение: для x, y и z:
  - Линейным проходом разбить список событий для соответствующей оси на 2 в соответствии с разбиением. При этом упорядоченность сохранится.
- Рекурсивно запустить алгоритм для двух полученных частей.

Алгоритм останавливает рекурсивное разбиение тогда, когда количество тетраэдров в обрабатываемой части становится достаточно мало, либо когда в результате деление количество тетраэдров в одной из частей оказался равен исходному их числу.

Проверка на оптимальность для каждой из осей может быть следующей: выбирается положение плоскости, при котором размеры левой и правой частей максимально близки друг к другу. Наилучшая из осей выбирается по минимуму $\min(\text{left\_count}, \text{right\_count})$ для оптимального положения плоскости, что позволяет выбрать минимальный из разрезов подпространства и уменьшить коммуникации между подобластями. Отметим, что «толщину» слоя пересечения легко варьировать. Для этого достаточно осуществить проход по событиям двумя указателями вместо одного. Один из указателей при этом будет соответствовать началу пересечения подобластей, второй — концу. При этом легко поддерживать как нужную геометрическую толщину пересечения подобластей, так и минимальное число тетраэдров в пересечении.

Легко видеть, что асимптотическая сложность алгоритма равна $O(T \log T)$, где T — исходное число тетраэдров, при условии, что каждый раз в пересечении оказывается небольшое число тетраэдров и части делятся примерно поровну. Действительно, сложность сортировки событий в начале алгоритма равна $O(T \log T)$. Далее, если число тетраэдров растет несущественно по сравнению с величиной T (например, конечное число тетраэдров есть $O(T)$), то на каждом уровне дерева требуется линейный проход по всем событиям для каждого из узлов дерева этого уровня, суммарное количество которых есть $O(T)$. При этом число подобластей при переходе на уровень ниже увеличивается в 2 раза, поэтому общее число уровней не превзойдет $O(\log T)$. Отсюда получаем заявленную сложность.

Замечание 1. Алгоритм осуществляет разбиение сетки на подобласти с пересечениями, причем никакие две подобласти не имеют протяженной общей границы кроме, возможно, части внешней границы расчетной области.

Доказательство этого факта следует из следующего тривиального наблюдения: два тетраэдра с общей гранью невозможно разбить плоскостью так, чтобы один тет-

раэдр лежал по одну сторону от нее, второй по другую и плоскость не касалась хотя бы одного из тетраэдров. При этом, как следует из алгоритма, тетраэдры, касающиеся либо пересекающиеся с плоскостью сечения, помещаются в пересечение подобластей. Данный факт оказывается очень важным для дальнейшего решения задачи методом декомпозиции на подобласти.

Рассмотрим теперь постановку задачи нахождения поля для подобластей. Пусть имеется разбиение расчетной области Ω на подобласти $\Omega = \bigcup \Omega_p$. Будем искать решение задачи (5) на области Ω путем решения задачи (5) в каждой из подобластей Ω_p. На внутренней границе Γ каждой из подобластей Ω_p будем ставить условия Дирихле из (3). Эти условия будут определяться из значений поля на внутри подобластей, соседних с Ω_p, и, на конечно-элементном уровне, из коэффициентов разложения поля $\vec{E}$ по базисным функциям, соответствующим граничным ребрам и граням. То есть, требуется найти поля в каждой из подобластей так, чтобы они были согласованы друг с другом по краевым условиям.

Теорема 1. *Решение задачи для подобластей для предлагаемого алгоритма декомпозиции существует. Оно единственно, если частота излучения не совпадает с резонансными частотами полостей, образованных подобластями и их пересечениями.*

Доказательство. Ясно, что решение исходной задачи на всей области Ω удовлетворяет задаче на подобластях. Единственность следует из следующей леммы.

Лемма 1. *Пусть поле $\vec{E}$ является решением задачи для подобластей. Тогда для любых Ω_p и $\Omega_{p'}$, имеющих непустое пересечение, значения $\vec{E}$ в общих точках Ω_p и $\Omega_{p'}$ совпадают.*

Доказательство напрямую следует из того, что для пересечения подобластей мы имеем краевую задачу, которая, если частота не резонансная для данной задачи, имеет единственное решение. Поэтому решение задачи для подобластей однозначно определяет электрическое поле в любой точке исходной области Ω.

Доказательство теоремы 1 завершается замечанием того, что

- для любой внутренней точки области Ω существует замкнутая окрестность, что эта точка вместе с окрестностью целиком лежит внутри какой-либо подобласти Ω_p;
- для любой базисной функции из $\mathcal{W}_{l,0}$ ее носитель целиком лежит внутри какой-нибудь подобласти Ω_p;

Отсюда следует, что электрическое поле, являющееся решением задачи для подобластей, удовлетворяет и вариационной формулировки (5) в непрерывном случае, и дискретному аналогу этой формулировке в конечно-элементном случае. Однако решение вариационной формулировки единственно при указанных ограничениях на параметры среды и частоту излучения [3], поэтому единственным оказывается и решение задачи для подобластей. □

3.2. Алгебраическая декомпозиция

Другим вариантом декомпозиции задачи на подобласти, рассматриваемым в рамках работы, является модифицированный алгоритм одно-направленного разбиения

графа матрицы системы [2]. Алгоритм строит разбиение неизвестных системы на множества — алгебраические подобласти Ω_p^A. В дальнейшем индекс A в Ω_p^A мы будем опускать.

Первым шагом алгоритма является нахождение псевдо-периферийной вершины v. Поиск такой вершины осуществляется следующим образом [2].

1. Выбирается произвольная вершина v. Псевдо-диаметр графа D полагается равным 0.
2. Запускается обход в ширину по графу, начиная от вершины v.
3. Находится любая максимально удаленная вершина v'.
4. Если расстояние $d(v, v')$ больше D, то $v \leftarrow v'$, $D \leftarrow d(v, v')$, переход на шаг 2.

Информация с последнего обхода графа в ширину используется для дальнейшего построения разбиения. Для начала все вершины разбиваются во «фронты» F_k — множества вершин, равноудаленных от v на расстояние k. Далее фронты объединяются в алгебраические подобласти по следующему принципу.

С помощью бинарного поиска мы будем максимизировать размер минимальной подобласти. Проверка того, что при данном минимальном размере подобласти можно получить необходимое число подобластей, может быть осуществлена жадным алгоритмом [8]. Пример кода, определяющего максимальное число подобластей, представлен на рис. 1.

```
int get_max_domains(int k, int front_size[], int min_size) {
    int domains = 0, domain_size = 0, i;
    for(i = 0; i < k; i++) {
        domain_size += front_size[i]; // Размер в вершинах
        if(domain_size >= min_size) {
            ++domains;
            domain_size = 0;
        }
    }
    // Оставшиеся вершины отходят к последней подобласти
    return domains;
}
```

Рис. 1. Алгоритм определения максимального числа подобластей

Представленный алгоритм генерирует подобласти без пересечений. Однако получить разбиение с пересечениями довольно легко — достаточно при начале генерации следующей подобласти отступить от границы предыдущей внутрь необходимое число фронтов. Графическое представление объединения фронтов в подобласти с пересечениями изображено на рис. 2, где фронты изображены вертикальными линиями.

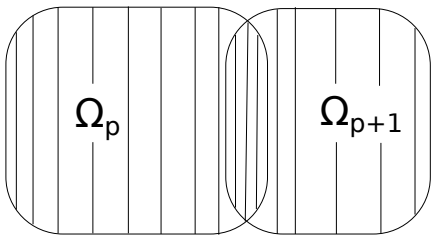


Рис. 2. Объединение фронтов матрицы в подобласти

Обозначив l_p и r_p — границы подобласти Ω_p, мы имеем разбиение на подобласти

$$\Omega_p = \bigcup_{k=l_p}^{r_p} F_k,$$

Необходимо отметить, что ребра в графе матрицы могут соединять вершины либо одного, либо двух соседних фронтов. Это означает, что каждая подобласть Ω_p может граничить одновременно максимум с двумя подобластями — Ω_{p-1} и Ω_{p+1} (при наличии пересечений всегда можно считать, что два граничных с Ω_p фронта F_{l_p-1} и F_{r_p+1} принадлежат соседним подобластям Ω_{p-1} и Ω_{p+1} соответственно). Тогда последовательно занумеровав переменные в каждой из подобластей, а в каждой из подобластей — последовательно в каждом из фронтов, мы получим следующую блочно-трехдиагональную СЛАУ:

$$\begin{bmatrix} D_1 & U_1 & & \\ L_1 & D_2 & \ddots & \\ & \ddots & \ddots & U_{p-1} \\ & & L_{p-1} & D_p \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_p \end{bmatrix}.$$

При этом если разбиение исходной СЛАУ выполнялось с пересечениями, то в итоговой СЛАУ окажется по несколько переменных, соответствующих вершинам в пересечении.

4. Алгоритмы решения СЛАУ

После декомпозиции задачи на подобласти и построения блоков матрицы на соответствующих узлах кластера, мы имеем распределенную СЛАУ

$$Au = f.$$

Для решения таких СЛАУ широко используются методы Шварца. Одна итерация аддитивного метода Шварца в матричном виде может быть записана как (см. [2])

$$u_{i+1} = u_i + \sum_p R_p^T A_p^{-1} R_p (f - Au_i),$$

где R_p — оператор сужения на подобласть Ω_p. При этом, на самом деле, итерировать достаточно только переменные, соответствующие внутренним границам подобластей, поскольку зная граничные значения, легко определить значения внутри подобластей, решив соответствующую задачу в подобласти. То есть

$$u_p = A_p^{-1} R_p (f - A\hat{R}^T u^\Gamma), \tag{8}$$

где u^Γ — вектор граничных переменных, а $\hat{R}$ — оператор сужения на граничные переменные. Подпространство граничных переменных принято называть подпространством следов. Тогда итерация метода Шварца в подпространстве следов переписывается как

$$u_{i+1}^\Gamma = u_i^\Gamma + \hat{R} \sum_p R_p^T A_p^{-1} R_p (f - A\hat{R}^T u_i^\Gamma).$$

Эту итерацию можно представить в виде $u^\Gamma_{i+1} = S(u^\Gamma_i) = Tu^\Gamma_i + g$, где

$$T = I - \hat{R}\sum_p R_p^T A_p^{-1} R_p A \hat{R}^T, \quad g = \hat{R}\sum_p R_p^T A_p^{-1} R_p f.$$

Сходимость данного метода имеется при спектральном радиусе $\rho(T) < 1$. В то время как для эллиптических уравнений это условие выполнено, для уравнения Гельмгольца это не так. Однако в этом случае можно рассмотреть решение системы

$$[I - T]\, u^\Gamma = g. \tag{9}$$

Теорема 2. *Система (9) является совместной. В случае геометрической декомпозиции при выполнении условий теоремы 1 оно единственно.*

Доказательство. Действительно, система (9) эквивалентна системе

$$\hat{R}\sum_p R_p^T A_p^{-1} R_p A u = \hat{R}\sum_p R_p^T A_p^{-1} R_p f,$$

поэтому решение $u^\Gamma = \hat{R}u$, где u — решение $Au = f$ удовлетворяет ей. С другой стороны любое решение (9) является неподвижной точкой итерации Шварца. В случае геометрической декомпозиции на подобласти у неподвижной точки итерации Шварца есть простая интерпретация. Это вектор, задающий поле в подобластях, согласованное по граничным условиям. Однако при выполнении условий теоремы 1 последняя утверждает, что такое поле единственно. Значит, единственна и неподвижная точка итерации Шварца. □

Можно отметить, что решение системы методом Шварца есть решение предобусловленной системы $M^{-1}Au = M^{-1}f$ с M^{-1}, связанным с A и T соотношением $T = I - M^{-1}A$, откуда $M^{-1} = [I - T]A^{-1}$ и предобусловленная система переходит в систему (9). Эту систему можно решать и итерационными методами в подпространствах Крылова, например методом обобщенных минимальных невязок (GMRES) [2]. Для методов в подпространствах Крылова не требуется явный вид матрицы системы, а достаточно только операции умножения матрицы на вектор. Запишем результат умножения $I - T$ на вектор v через результат одной итерации метода аддитивного Шварца $S(v)$:

$$[I - T]\, v = v - Tv = v - S(v) + g, \quad g = S(0).$$

Таким образом, для решения СЛАУ (9) методом GMRES достаточно реализовать обычный метод аддитивного Шварца. Для решения задач в подобластях $A_p x = b$ можно использовать какой-нибудь прямой решатель для разреженных систем. В работе использовался решатель PARDISO из библиотеки Intel MKL [9]. Такой подход оказывается достаточно эффективным, поскольку на каждой итерации требуется решать системы с одними и теми же матрицами, и LU-разложение матриц A_p можно посчитать только один раз. Кроме того, такой подход оказывается существенно эффективнее простого использования решателя PARDISO для исходной системы, поскольку время, требуемое PARDISO для разложения матрицы, растет практически как $O(N^3)$, где N — порядок системы. Дополнительно эффективность достигается

за счет использования подпространства следов, так как в этом случае существенно уменьшается объем коммуникационных затрат, а также объем памяти, требуемый для хранения базиса подпространства Крылова в методе GMRES, поскольку хранить и пересылать требуется только граничные переменные. После того, как решение системы найдено, искомое поле $\vec{E}$ восстанавливается в каждой из подобластей.

5. Численные эксперименты

5.1. Модельная задача

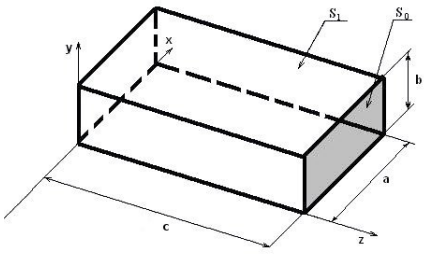


Рис. 3. Расчетная область для модельной задачи

В качестве одного из тестов рассматривалась модельная задача с расчетной областью в виде волновода с линейными размерами $a = 72$, $b = 34$, $c = 200$ мм (см. рис. 3. Параметры задачи: $\mu_r = 1$, $\dot{\varepsilon}_r = 1 - 0.1i$, частота $\omega = 6\pi \cdot 10^9$ Гц. Граничные условия: на грани $z = 200$ задавалось условие

$$\vec{E}_0 \times \vec{n} = \vec{e}_y \sin(\pi x/a),$$

на остальных гранях — условие металлической стенки ($\vec{E}_0 = 0$). Аналитическое решение

$$\vec{E} = \vec{e}_y \sin\left(\frac{\pi x}{a}\right) \frac{\sin \gamma z}{\sin \gamma c}.$$

При решении СЛАУ критерием остановки итерационного процесса служило выполнение условия $(r, r) < \varepsilon^2 (f, f)$, где f — вектор правой части системы, $r = f - Au$ — вектор невязки. В проведенных экспериментах ε было выбрано равным 10^{-7}.

Для построения сетки расчетная область делилась по каждому из измерений на некоторое число шагов, и каждый из получающихся параллелепипедов делился на 6 тетраэдров. В тестах использовалась квазиравномерная сетка с разбиениями по каждой из координат с некоторым шагом h. Были рассмотрены сетки с разбиениями $4 \times 2 \times 10$, $8 \times 4 \times 20$ и $15 \times 7 \times 40$. Для аппроксимации использовались базисные функции второго порядка.

Тесты проводились на двух системах: системе Intel Core2 Duo E6550 @ 2.33 ГГц, 2 ядра, с оперативной памятью объемом 4 ГБ, и на узлах HP BL2x220 G6 кластера НКС-30Т. Каждый узел — Intel Xeon E5540, 2.53 ГГц, 2x4 ядер, 16 ГБ памяти.

В табл. 1 приведены результаты тестирования решателя при использовании геометрической декомпозиции. В таблице nproc — число MPI процессов (и подобластей), N — порядок системы для nproc=1, N_b — размерность пространства следов, N_p — максимальный порядок СЛАУ для подобластей, n — число итераций решателя GMRES,

δE — относительная ошибка для поля $\vec{E}$ в барицентрах тетраэдров. Тесты для сетки $4 \times 2 \times 10$ при 16, 32 и 64 подобластях не проводились, так как при этом не хватает тетраэдров сетки для построения нужного числа подобластей. Тесты проводились на двухъядерном Core2 Duo, поэтому время работы решателей не приводится.

Таблица 1

Решение модельной задачи с геометрической декомпозицией

Сетка	N	δE		nproc					
				2	4	8	16	32	64
$4 \times 2 \times 10$	2392	3,1e-02	n	8	11	22	—	—	—
			N_b	136	408	2656	—	—	—
			N_p	1408	916	670	—	—	—
$8 \times 4 \times 20$	21664	7,3e-03	n	9	14	23	40	53	70
			N_b	592	1776	4556	15764	40396	105200
			N_p	11782	6292	4522	2610	1834	1058
$15 \times 7 \times 40$	149874	2,1e-03	n	10	17	27	50	64	80
			N_b	2012	6036	14084	34560	67872	143256
			N_p	78206	40486	21626	12818	8346	5478

В табл. 2 приведены результаты тестирования для случая алгебраической декомпозиции. Параметр пересечения подобластей был установлен в 4, то есть подобласти пересекались, если это было возможно, по 4 фронтам. Из таблицы видно, что при декомпозиции матрицы на слишком большое число подобластей число итераций начинает быстро расти. В этих случаях на каждую подобласть приходится всего по нескольку фронтов.

Таблица 2

Решение модельной задачи с алгебраической декомпозицией

Сетка	N	δE		nproc				
				2	4	8	16	32
$4 \times 2 \times 10$	2392	3,1e-02	n	6	12	49	—	—
			N_b	378	980	2261	—	—
			N_p	1542	1334	796	—	—
$8 \times 4 \times 20$	21664	7,3e-03	n	9	13	21	258	—
			N_b	1540	4678	10708	23052	—
			N_p	12420	9232	6306	3200	—
$15 \times 7 \times 40$	149874	2,1e-03	n	13	18	25	38	517
			N_b	5210	16646	37670	78446	162720
			N_p	81132	49618	31168	22678	11282

5.2. Задача электромагнитного каротажа

Вторая из рассматриваемых задач — задача электромагнитного каротажа. Расчетная область представляет из себя куб с центром в начале координат со стороной 10 метров. Куб заполнен средой с $\sigma = 0,1$ См/м, в среде имеется слой с $\sigma = 0,05$

См/м. Координаты слоя по оси Oz: от $-0,425$ м до $-0,275$ м. В среде пробурена вертикальная цилиндрическая скважина радиуса $0,108$ м с центром в начале координат в плоскости Oxy. В скважине имеется цилиндрическая каверна внешнего радиуса $0,118$ м и положением по оси Oz от $-0,0725$ до $0,0725$ м. Проводимость скважины и каверны $\sigma = 5$ См/м. В скважину вставлен полый цилиндрический прибор радиуса $0,043$ м, смещенный по оси x относительно центра скважины на $-0,064$. Проводимость прибора равна нулю. В приборе имеется одна генераторная петля при $z = 0,5$ м и две приемные петли при $z = 0,0$ и $z = 0,1$ м. Радиус петель — $0,0365$ м. По генераторной петле течет ток 1 А с частотой 14 МГц. Искомой является разность фаз ЭДС в приемниках.

Генерация неравномерной сетки проводилась при помощи утилиты NETGEN v4.9.13 [10]. В результате была получена сетка с 71892 узлами и 388836 тетраэдрами.

Результаты решения задачи на кластере с использованием алгебраической декомпозиции представлены в табл. 3. В ней дополнительно указаны времена t_{fact} — максимальное время факторизации матриц D_p в подобластях и t_{tot} — общее время работы решателя. Результаты тестирования для nproc < 4 не приведены, так как в этом случае PARDISO не хватает 16 Гб памяти на узле для хранения LU факторов матрицы, а режим Out-of-Core с хранением факторов на диске является слишком медленным.

Таблица 3

Решение задачи каротажа с алгебраической декомпозицией матрицы

nproc	N	N_b	t_{fact}, с	t_{tot}, с	n
4	2398750	371126	8,52e+02	9,49e+02	21
8	2398750	833744	3,30e+02	4,26e+02	28

Результат решения задачи показал хорошее соответствие с результатом группы НГТУ, которая решала эту задачу другим методом — методом выделения поля источника, при этом искомая разность фаз ЭДС в приемниках совпала с ошибкой 1.8%. Стоит отметить также, что для данной задачи обычные итерационные алгоритмы в подпространствах Крылова, примененные к СЛАУ для всей подобласти, показывают отсутствие сходимости.

Заключение

В работе рассматривается задача моделирования электромагнитного поля в частотной области. Предложено два эффективных алгоритма геометрической и алгебраической декомпозиции задачи, причем последний может применяться и для решения других задач. Предлагаемые итерационные алгоритмы в подпространствах следов с предобуславливателем Шварца требуют невысоких коммуникационных затрат и хорошо подходят для вычислительных систем с распределенной памятью. Реализация алгоритмов на языке C++ с использованием средств MPI продемонстрировала хорошую производительность на методических и практических задачах. Предложенные решатели позволили решить задачу электромагнитного каротажа с высокой точностью.

Работа выполнена при поддержке РФФИ (грант 11-01-00205).

Литература

1. Karypis, G. A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs / G. Karypis, V. Kumar // SIAM Journal on Scientific Computing. – 1999. – Vol. 20, № 1. – P. 359–392.
2. Saad, Y. Iterative Methods for Sparse Linear Systems, Second Edition. / Y. Saad – Society for Industrial and Applied Mathematics, 2003.
3. Monk, P. Finite Element Methods for Maxwell's Equations. / P. Monk – Oxford University Press, 2003.
4. Соловейчик, Ю.Г. Метод конечных элементов для решения скалярных и векторных задач / Ю.Г. Соловейчик, М.Э. Рояк, М.Г. Персова – Новосибирск: Изд-во НГТУ, 2007.
5. Ingelstrom, P. A new set of H(curl)-conforming hierarchical basis functions for tetrahedral meshes / P. Ingelstrom // IEEE Transactions on Microwave Theory and Techniques. – 2006. – Vol. 54, № 1. – P. 160–114.
6. Ильин, В.П. Методы и технологии конечных элементов. / В.П. Ильин – Новосибирск: Изд-во ИВМиМГ СО РАН, 2007.
7. Fuchs, H. On visible surface generation by a priori tree structures / H. Fuchs, Z.M. Kedem, B.F. Naylor // ACM Computer Graphics. – 1980. – Vol. 14, № 3. – P. 124–133.
8. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест – М., МЦНМО: БИНОМ. Лаборатория знаний, 2004.
9. Intel. The Flagship High-Performance Computing Math Library for Windows*, Linux*, and Mac OS* X. Intel® Math Kernel Library from Intel. URL: http://software.intel.com/en-us/articles/intel-mkl/ (дата обращения: 22.01.2012)
10. Schöberl, J. NETGEN — An advancing front 2D/3D-mesh generator based on abstract rules / J. Schöberl // Computing and Visualization in Science. – 1997. – Vol. 1, № 1. – P. 41–52.

Дмитрий Сергеевич Бутюгин, младший научный сотрудник, Институт вычислительной математики и математической геофизики СО РАН, аспирант, Новосибирский государственный университет, dm.butyugin@gmail.com.

ALGORITHMS OF SLAES SOLUTION FOR THE SYSTEMS WITH DISTRIBUTED MEMORY APPLIED TO THE PROBLEMS OF ELECTROMAGNETISM

D.S. Butyugin, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation)

Paper presents various aspects of harmonic electromagnetic fields simulation on clusters. The major computational complexity comes from the solution of the systems of linear algebraic equations (SLAEs) arising from the approximations of corresponding electromagnetic boundary value problems by Nedelec elements of various orders. Effective and efficient approaches to the decomposition of the computational domain and the matrix of the system are considered. Distributed SLAEs are solved using iterative Krylov subspace methods preconditioned by additive Schwarz method. In order to increase the effectiveness of the algorithms iterations are performed in the trace space. Implementation of the solvers is based on MPI for data transfers. The solution of the systems in subdomains is performed by PARDISO direct solver from Intel® MKL library. Numerical experiments results on a series of model and real-life problems show the effectiveness of the presented algorithms.

Keywords: Maxwell equations, iterative algorithms, domain decomposition methods, additive Schwarz method.

References

1. Karypis G., Kumar V. A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM Journal on Scientific Computing. 1999. Vol. 20, № 1. P. 359–392.
2. Saad Y. Iterative Methods for Sparse Linear Systems, Second Edition. Society for Industrial and Applied Mathematics, 2003.
3. Monk P. Finite Element Methods for Maxwell's Equations. Oxford University Press, 2003.
4. Soloveychick Y.G., Royak M.E., Persova M.G. Metod konechnykh elementov dlya resheniya skalarnykh i vectornykh zadach [Finite Element Method for the Solution of Scalar and Vector Problems]. Novosibirsk, NSTU Publ., 2007.
5. Ingelstrom P. A New Set of H(curl)-conforming Hierarchical Basis Functions for Tetrahedral Meshes IEEE Transactions on Microwave Theory and Techniques. 2006. Vol. 54, № 1. P. 160–114.
6. Il'in V.P. Metody i tekhnologii konechnykh elementov [Methods and Technologies of Finite Elements]. Novosibirsk, ICM&MG SBRAS Publ., 2007.
7. Fuchs H., Kedem Z.M., Naylor B.F. On Visible Surface Generation by A Priori Tree Structures ACM Computer Graphics. 1980. Vol. 14, № 3. P. 124–133.
8. Cormen T., Leiserson C., Rivest R. Introduction to Algorithms. MIT Press, 2001.
9. Intel (R) Math Kernel Library from Intel:
URL: http://software.intel.com/en-us/articles/intel-mkl/
10. Schöberl J. NETGEN — an Advancing Front 2D/3D-mesh Generator Based on Abstract Rules Computing and Visualization in Science. 1997. Vol. 1, № 1. P. 41–52.

Поступила в редакцию 14 марта 2012 г.

УДК 536.42

МОДЕЛИРОВАНИЕ ПРОЦЕССА РОСТА НАНОПЛЕНОК МЕТОДОМ ХИМИЧЕСКОГО ОСАЖДЕНИЯ ИЗ ГАЗОВОЙ ФАЗЫ[1]

Ю.Я. Болдырев, К.Ю. Замотин, Е.П. Петухов

Большинство задач, которые связаны со многими аспектами развития нанотехнологий, по своей природе существенно междисциплинарны. Одним из наиболее характерных примеров этого является проблематика применения газофазного синтеза в нанотехнологиях. По своему существу такие технологии являются реализацией процессов химического осаждения вещества из газообразного состояния, подаваемого в реакционную зону, в твердое состояние. Междисциплинарность рассматриваемых в газофазном синтезе процессов порождает серьезные трудности при их изучении. При этом в рамках традиционного физического эксперимента не удается получить хорошего результата, так как такой эксперимент: не является наглядным, не позволяет изучать зависимость конечного материала от различных физических параметров системы, занимает много времени, дорог. Поэтому естественно искать пути решения задач на базе математического моделирования, которое лежит в основе виртуального эксперимента. В основе работы — разработка и апробация технологий математического моделирования с использованием высокопроизводительных вычислений в области процессов газофазного синтеза наноразмерных структур и наноматериалов с целью изучения и обеспечения визуализации протекающих физико-химических процессов.

Ключевые слова: наноиндустрия, газофазный синтез наноматериалов, математическое моделирование, газовая динамика, физико-химические процессы.

Химическое осаждение из газовой фазы — получение твердых веществ с помощью химических реакций, реагенты подаются в реакционную зону в газообразном или плазменном состоянии [1]. Используют для получения текстурированных покрытий, монокристаллов, эпитаксиальных и монокристаллических пленок, нитевидных монокристаллов, порошков, барьерных слоев др. Выражение «химическое осаждение из газовой фазы» является наиболее точным переводом с английского языка термина chemical vapor deposition (общепринятая аббревиатура — CVD), который был впервые введен Blocher в 1966 году и с тех пор общепринят во всем мире.

Появление и бурное развитие микроэлектроники придало мощный импульс для разработки разнообразных CVD технологий. Этим методом получают тонкие пленки металлов, диэлектриков и полупроводников, выращивают монокристаллы и эпитаксиальные пленки. Особо следует подчеркнуть, что необходимость получения пленок заданного состава и с требуемым комплексом физических и химических слоев для применения в электронике обусловила проведение тщательных исследований физико-химических закономерностей процессов, что с неизбежностью привело к более глубокому пониманию сущности и механизмов CVD процессов.

К настоящему времени в мировой практике накоплен большой экспериментальный материал по результатам исследования разнообразных процессов химического осаждения из газовой фазы тонких пленок, нанопорошков, нановолокон, наностержней и наноструктур. Однако, несмотря на тот факт, что исследованию некоторых

[1]Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012»

конкретных технологических процессов, посвящены сотни и даже тысячи публикаций, их детерминированные модели, достоверно и однозначно описывающие физико-химические закономерности, отсутствуют. Это обусловлено чрезвычайной сложностью механизма CVD процессов, характеризующихся многомаршрутностью химических реакций, присутствием нескольких гомогенных и гетерогенных стадий, а также многоступенчатостью превращений [2, 3]. Используя современные вычислительные технологии удается надежно описывать лишь процессы массо- и теплопереноса, что, в ряде случаев, позволяет успешно оптимизировать и разрабатывать конструкции реакционных камер, применяемых для реализации процессов ХОГФ [4]. Именно на основе таких расчетов, принимая во внимание экспериментально полученные сведения о кинетических закономерностях некоторых конкретных процессов ХОГФ, осуществляется проектирование промышленных реакторов в крупных международных корпорациях (Applied Materials, Samsung Electronics и др.).

Для успешного исследования и моделирования CVD систем необходимы строгие представления о схемах используемых на практике установок, технологических параметрах и их влиянии на условия синтеза, о химии, физике и физико-химии элементарных явлений, сопровождающих синтез, и характер их взаимодействии, иметь экспериментальные данные о процессах и возможности их управления [5]. Схематично CVD процесс может быть проиллюстрирован рис. 1. Моделирование CVD-процессов

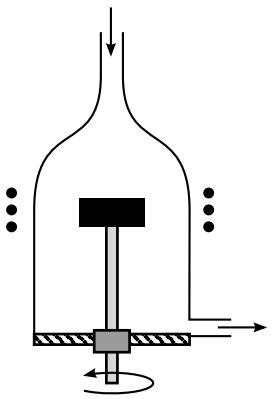


Рис. 1. Схематическое изображение CVD реактора

базируется на системе уравнений газовой динамики — уравнениях Навье — Стокса — записанной для многокомпонентной реагирующей среды [6]. Кроме того, требуется построение «подмоделей», описывающих химические реакции, турбулентность и теплообмен за счет излучения.

Задача осложняется требованиями расчета температуры и состава смеси, которые сводятся к решению уравнений для компонентов смеси и температуры. Последние остаются незамкнутыми, поскольку требуют определения величины средней скорости реакции. Таким образом, ее расчет является основной целью химической модели.

Поскольку современные программные комплексы позволяют решать весьма широкий круг междисциплинарных задач, для проведения численного моделирования был выбран программный комплекс ANSYS FLUENT [7]. В случае решения уравнения сохранения для химических веществ, этот комплекс получает значение локальной массовой доля каждого вещества Y_i через решение уравнения конвекции-диффузии

для i вещества. Запишем уравнение сохранения для i-компоненты в следующем общем виде:

$$\frac{\partial}{\partial t}(pY_i) + \bigtriangledown \cdot (p\vec{\nu}Y_i) = - \bigtriangledown \cdot \vec{J}_i + R_i + S_i, \tag{1}$$

где R_i является нетто-коэффициентом воспроизводства i-компоненты вещества в результате химической реакции, а S_i скорость воспроизводства от добавления из дисперсной фазы с учетом дополнительных источников, задаваемых пользователем. Уравнение такого вида должно решаться для $N-1$ химического вещества, где N есть общее число химических компонент, представленных в газовой фазе. Поскольку сумма массовых долей всех компонент должна быть тождественно равна единице, то массовая доля N-ого компонента будет равна единице минус сумма массовых долей первых $N-1$ веществ. Чтобы свести к минимуму численные ошибки, в согласии со стандартными подходами N-м веществом должен быть выбрана компонента с наибольшей массовой долей, например $N2$ в случае, когда окислителем является воздух.

Отдельно остановимся на проблеме вычислений потока диффузии. В уравнении (1) есть поток диффузии компоненты i, который возникает в результате появления градиентов концентраций. По умолчанию, комплекс ANSYS FLUENT использует так называемую «слабую» аппроксимацию, при которой поток диффузии можно записать в виде:

$$\vec{J}_i = -\rho D_{i,m} \bigtriangledown Y_i, \tag{2}$$

где $\mathrm{D}_{i,m}$ — коэффициент диффузии для i-й компоненты смеси. Подобное приближение не всегда может быть приемлемым, когда требуется моделирование полной многокомпонентной диффузии. В таких случаях, к системе может быть добавлено и решено уравнение Максвелла — Стефана.

Скорости реакции, которые появляются в качестве источниковых членов в уравнении (1), вычисляются по следующей модели: эффект турбулентных флуктуаций игнорируется, и скорость реакции определяется формулой Аррениуса [1]. В рамках данной ламинарной модели конечной скорости вычисляются химические источники в терминах выражения Аррениуса и игнорируются эффекты турбулентных флуктуаций. Модель является точной для ламинарных течений, но, как правило, неточна для турбулентных течений из-за высокой нелинейной химической кинетики Аррениуса. Ламинарная модель, однако, может быть приемлемым для процессов с относительно медленно протекающими реакциями и малыми турбулентными флуктуациями.

Чистый источник химического вещества i определяется как Аррениусовская сумма по N_R реакциям, в которую данное вещество входит:

$$R_i = M_{\omega,i} \sum_{r=1}^{N_r} \hat{R}_{i,r}, \tag{3}$$

где $\mathrm{M}_{\omega,i}$ — это молекулярная масса вещества i, а $\hat{R}_{i,r}$ Аррениусовская молярная скорость притока / оттока вещества i в реакции r [12]. При этом, реакция может происходить:

- в непрерывной фазе;
- между веществами в непрерывной фазе;
- на поверхности стенки в результате осаждения;
- в результате развития данной фазы.

Рассмотрим некоторую реакцию r, записанную в общем виде:

$$\sum_{r=1}^{N_r} \nu'_{i,r} M_i = ^{k_{f,r}} \Longrightarrow \Longleftarrow ^{k_{b,r}} = \sum_{r=1}^{N_r} \nu''_{i,r} M_i, \tag{4}$$

где N — количество химических веществ в системе;

$\nu'_{i,r}$ — стехиометрический коэффициент для i реагента в реакции r;

$\nu''_{i,r}$ — стехиометрический коэффициент для i продукта в реакции r;

M — символ, обозначающий i вещество;

$k_{f,r}$ — константа скорости прямой реакции r;

$k_{b,r}$ — константа скорости обратной реакции r.

Уравнение (4) справедливо как для обратимых, так и необратимых реакций, заметим, что по умолчанию реакции не являются обратимыми. Для необратимых реакций константа обратной скорости опускается.

Суммирование в уравнении (4) производится для всех химических веществ, участвующих в процессе. Но только вещества, которые появляются в качестве реагентов или продуктов будут иметь ненулевые стехиометрические коэффициенты. Таким образом, элементы, которые не участвуют в реакции, выпадают из уравнения.

Для необратимых реакций, молярная скорость притока / оттока материала i в реакции r (в уравнении (3)) дается следующей формулой:

$$\hat{R}_{i,r} = \Gamma(\nu''_{i,r} - \nu'_{i,r})(k_{f,r} \prod_{j=1}^{N} [C_{j,r}^{(\eta''_{i,r} - \eta'_{i,r})}), \tag{5}$$

где $C_{j,r}$ — молярная концентрация j-го элемента в реакции r (кмоль/м3),

$\eta'_{i,r}$ — экспонента скорости для j-го реагента в реакции r,

$\eta''_{i,r}$ — экспонента скорости для j-го продукта в реакции r,

Для обратимой реакции молярная скорость создания / уничтожения материала i в реакции r определяется:

$$\hat{R}_{i,r} = \Gamma(\nu''_{i,r} - \nu'_{i,r})(k_{f,r} \prod_{j=1}^{N} [C_{j,r}]^{(\eta'_{i,r})} - k_{b,r} \prod_{j=1}^{N} [C_{j,r}]^{(\eta''_{j,r})}), \tag{6}$$

Отметим, что показатель скорости для обратной части реакции в уравнении (5) всегда равен стехиометрическому коэффициенту вещества ($\nu''_{i,r}$). Величина учитывает влияние от присутствия третьих тел в реакции на ее скорость. Третьи тела — вещества, которые не являются реагентами в данной реакции, но влияют на ее протекание. Данный член определяется из следующего соотношения:

$$\Gamma = \sum_{j}^{N} \gamma_{j,r} C_j, \tag{7}$$

где $\gamma_{j,r}$ являются коэффициентами влияния третьего тела j-го элемента в r-ой реакции. По умолчанию, комплекс ANSYS FLUENT не включает эффекты влияния третьего тела в процессе расчета скоростей реакций. Можно, однако, принудительно включать в учет влияние третьих тел, когда имеются основания их учитывать.

Переходя к скорости прямой реакции r — величине $k_{f,r}$, укажем, что она вычисляется с использованием уравнения Аррениуса:

$$k_{f,r} = A_r T^{\beta_r} e^{-E_r/RT}, \tag{8}$$

где A_r — предэкспоненциальный множитель (размерная единица)

β_r — температурный показатель экспоненты (безразмерная величина)

E_r — энергия активации (Дж/кмоль)

R — универсальная газовая постоянная (Дж/кмоль·К)

Для корректной постановки задачи необходимо знать (или получить из базы данных) значения для $\nu'_{i,r}$, $\nu''_{i,r}$, $\eta'_{j,r}$, $\eta''_{j,r}$, β_r, A_r, E_r, $\gamma_{j,r}$. В том случае, если реакция обратима, константа скорости обратной реакции вычисляется через скорость прямой реакции по следующему соотношению [12]:

$$k_{b,r} = \frac{k_{f,r}}{K_r}, \tag{9}$$

где K_r — константа равновесия r-ой реакции, вычисляемая по формуле:

$$K_r = e^{(\frac{\Delta S_r^0}{R} - \frac{\Delta H_r^0}{RT})} \Big(\frac{p_{atm}}{RT}\Big)^{\sum_{i=1}^{N}(\nu''_{i,r} - \nu'_{i,r})}. \tag{10}$$

В этой формуле величина p_{atm} обозначает атмосферное давление (101325 Па). Показатель степени (в круглых скобках) экспоненциальной функции представляет собой изменение свободной энергии Гиббса, и его компоненты рассчитываются следующим образом [7]:

$$\frac{\Delta S_r^0}{R} = \sum_{i=1}^{N}(\nu''_{i,r} - \nu'_{i,r})\frac{\Delta S_i^0}{R}, \tag{11}$$

$$\frac{\Delta H_r^0}{RT} = \sum_{i=1}^{N}(\nu''_{i,r} - \nu'_{i,r})\frac{\Delta h_i^0}{RT}, \tag{12}$$

где величины S_i^0 и h_i^0 стандартные состояния энтропии и энтальпии (теплоты образования). Эти значения определяются в комплексе ANSYS FLUENT как свойства смеси материала. Свойства энтропии и энтальпии для использованных в работе материалов задавались в полиномиальном формате CHEMKIN [9] и были получены из термодинамического справочника [10].

Для моделирования процесса металлорганического химического осаждения из газовой фазы (МО ХОГФ) тонких пленок была выбрана система материалов GaAs в реакционном наборе. Выбор материалов обусловливается тем, что данные полупроводниковые материалы широко используются при получении наногетероструктур, на основе которых создается множество приборов (полевые транзисторы, лазеры, светодиоды и фотоприемники и др.). Приведенная реакционная система уравнений (11–12) отличается большим количеством реакций протекающих в объеме и на поверхности. Для каждой реакции должны быть указаны следующие (были взяты из общедоступных источников, таблица 1) значения параметров:

- предэкспоненциальный множитель (A),

- показатель температуры (b),
- энергия активации (E) для скорости реакции в Аррениусовом виде (9)

Таблица 1

Параметры скоростей реакций

Объемные реакции	A	b	E
TMG => DMG + CH3	1.6E17	0	30057
DMG => MMG + CH3	2.5E15	0	17883
CH3 + H2 => CH4 + H	1.2E9	0	6300
ASH3 + CH3 = ASH2 + CH4	9.7E8	0	900
TMG + H = DMG + CH4	5.0E10	0	5051
DMG + H => MMG + CH4	5.0E10	0	5051
2H + M = H2 + M	1.0E13	0	0
2CH3 = C2H6	2.0E10	0	0
CH3 + H + M => CH4 + M	2.4E19	-1	0
TMG + CH3 => ADDUCT + CH4	2.0E8	0	5051
MMG => GA + CH3	1.0E16	0	39052
Поверхностные реакции	A	b	E
H + OPENAS(S) => H-AS(S)	4.95E9	0.5	0
H + OPENG(S) => H-G(S)	4.95E9	0.5	0
CH3 + OPENG(S) = CH3-G(S)	1.27E9	0.5	0
CH3 + OPENAS(S) = CH3-AS(S)	1.27E9	0.5	0
MMG + OPENAS(S) = MMG-AS(S)	5.37E8	0.5	0
DMG + OPENAS(S) => MMG-AS(S) + CH3	4.95E8	0.5	0
ASH + OPENG(S) = ASH(S)	5.68E8	0.5	0
ASH2 + OPENG(S) => ASH(S) + H	5.68E8	0.5	0
ASH3 + OPENG(S) => ASH(S) + H2	5.68E8	0.5	0
CH3 + H-AS(S) => CH4 + OPENAS(S)	1.26E8	0.5	0
CH3 + H-G(S) => CH4 + OPENG(S)	1.26E8	0.5	0
H + CH3-AS(S) => CH4 + OPENAS(S)	4.94E8	0.5	0
H + CH3-G(S) => CH4 + OPENG(S)	4.94E8	0.5	0
H-AS(S) + CH3-G(S) => CH4 + OPENAS(S) + OPENG(S)	1.0E16	0	5051
H-G(S) + CH3-AS(S) => CH4 + OPENAS(S) + OPENG(S)	1.0E16	0	5051
H-G(S) + H-AS(S) => H2 + OPENAS(S) + OPENG(S)	1.2E16	0	10102
CH3-G(S) + CH3-AS(S) => C2H6 + OPENAS(S) + OPENG(S)	1.0E16	0	10102
MMG-AS(S) + ASH(S) => CH4 + OPENG(S) + OPENAS(S) + GAAS(B)	5.0E17	0	14801
MMG-AS(S) + AS(S) => CH3 + OPENG(S) + OPENAS(S) + GAAS(B)	5.0E17	0	10103
2ASH(S) => AS2 + H2 + 2OPENG(S)	1.0E16	0	19681
CH3 + ASH(S) => AS(S) + CH4	1.28E8	0.5	10103
2AS(S) = AS2 + 2OPENG(S)	1.0E17	0	15155
TMG + OPENAS(S) => MMG-AS(S) + 2CH3	4.62E8	0.5	0
GA + OPENAS(S) = GA(S)	5.9E8	0.5	0
GA(S) + AS(S) => OPENAS(S) + OPENG(S) + GAAS(B)	1.1E9	0	505

В данной работе моделирование осаждения нанопленок осуществляются в реакторе вертикального типа с вращающимся диском (см. рис. 1). Все геометрические размеры реактора (высота, диаметр, высота расположения диска, способ подачи газов в реактор), а также физические (температуры стенок реактора и диска, рабочее давление в реакторе, скорость вращения подложки, расходы газов через входные отверстия) были выбраны в соответствии с данными приведенными в работе [8]. Такой выбор данных позволил обеспечить верификацию полученных результатов, которые представлены на рис. 2–4, с экспериментальными данными.

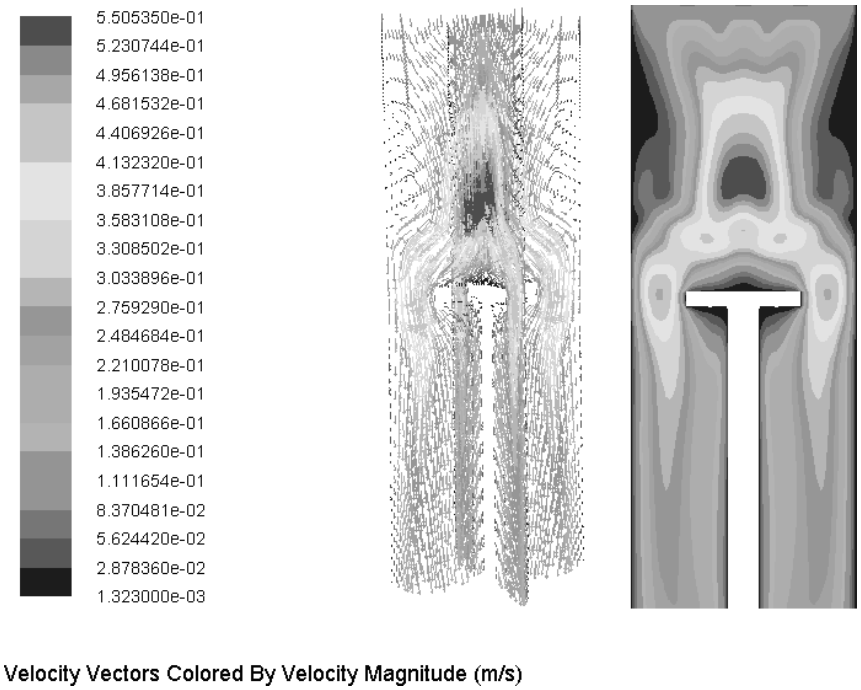


Рис. 2. Скорость потока внутри реактора

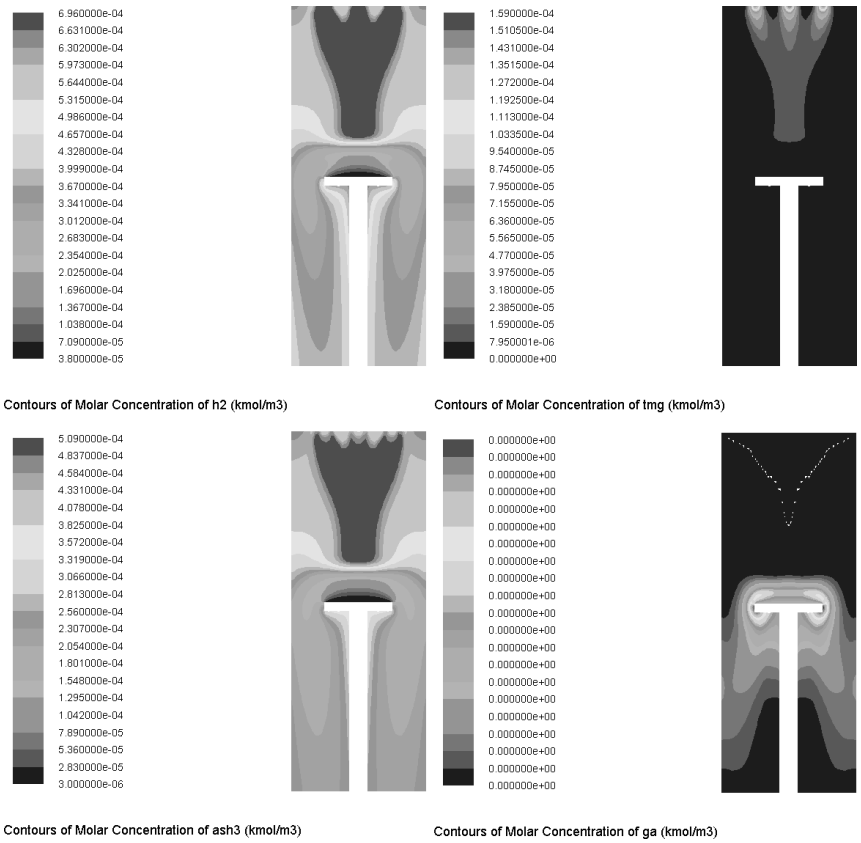


Рис. 3. Молярные концентрации компонент смеси

Задача решалась с применением программного комплекса Ansys Fluent 12.1 на вычислительном кластере СПбГПУ (64 узла: 2x AMD Opteron 280, 8Гб ОЗУ, IB 4x SDR). В связи с относительно небольшим размером расчетной сетки задачи, для одного расчета использовался только один вычислительный узел. Для получения исчерпывающей информации о физических зависимостях протекающих в реакторе процессов требовалось проведение массовых расчетов. Для этого уже были задействовано до 16 узлов кластера. Всего было решено и исследовано более 1000 расчетных случаев.

Представляется, что главным достижением применяемых технологий математического моделирования является получение всего спектра физических значений (см. рис. 2, 3) реагирующего газового потока внутри реактора: компоненты скорости, давление, температура, концентрации как исходных, так и результирующих веществ-реагентов. Кроме того, основным результатом моделирования можно считать получение распределения скорости осаждения пленки по площади подложки, представленные на рис. 4, находящейся на вращающемся диске, на котором происходит осаждение.

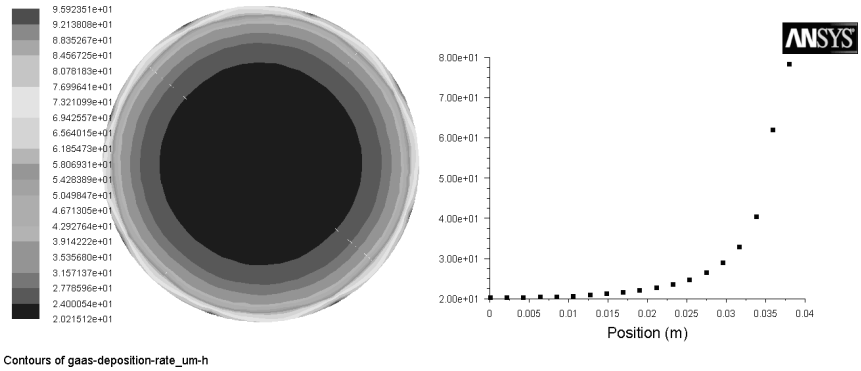


Рис. 4. Скорость осаждения арсенида галлия

Вместе с тем, несмотря на положительные результаты данного исследования, отсутствие детерминированных моделей, достоверно и однозначно описывающие физико-химические закономерности является существенной проблемой. Как уже отмечалось, данное обстоятельство обусловлено чрезвычайной сложностью механизма CVD процессов, характеризующихся многомаршрутностью химических реакций, присутствием нескольких гомогенных и гетерогенных стадий, а также многоступенчатостью превращений.

Согласованность результатов вычислений с опытным экспериментом демонстрирует рис. 5. Был выполнен ряд расчетов для вертикального реактора с варьируемыми параметрами: концентрацией триметилгаллия и температурой подложки. По полученным массивам данных построена зависимость скорости осаждения. Сравнение результатов расчета с экспериментом показывает полное качественное, а также достаточно хорошее количественное совпадение полученных зависимостей. Что является, несомненно, лучшим подтверждением адекватности выбранных моделей при решении задач химического осаждения тонких пленок из газовой фазы.

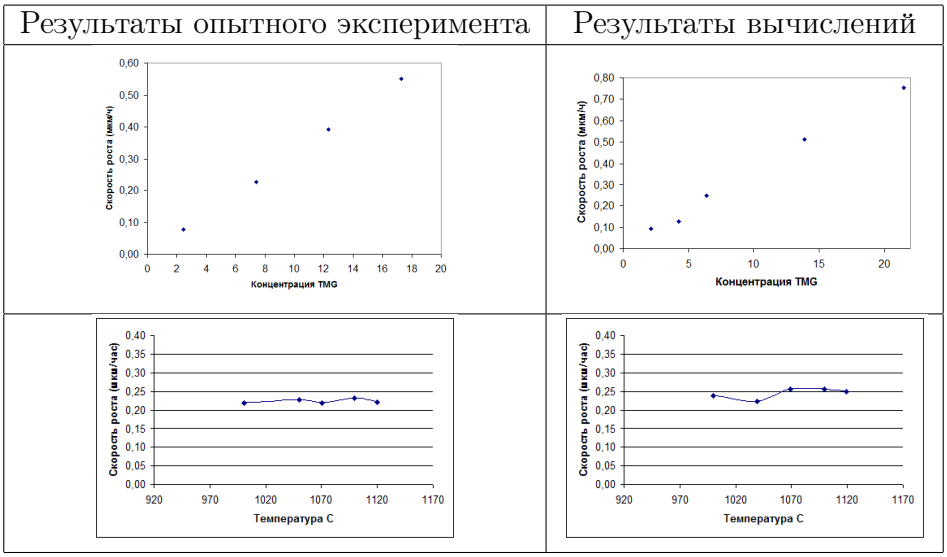


Рис. 5. Сравнение результатов вычислений с экспериментом

Заключение

По результатам исследования можно утверждать, что выбранные модели физико-химических процессов адекватно описывают реагирующее течение газовой среды и тех инструментов, на основе которых реализуется синтез наноразмерных структур и наноматериалов (подложки). Результаты моделирования осаждения GaAs легко согласуются с экспериментальными данными [8] как количественно, так и качественно.

Использование высокопроизводительных вычислительных технологий позволило расширить круг решаемых задач в области нанотехнологий путем математического моделирования различного технологических инструментов (оборудования) (установки химического осаждения из газовой фазы), технологических параметров (температуры подложки, рабочего давления и др.), а также различных материалов (пленки, порошки и др.).

Использование технологий математического моделирования на базе – высокопроизводительных вычислений позволяет получать информацию о процессе ХОГФ и полученном продукте за достаточно короткое время (в пределах несколько лабораторных работ) и не требует применения дорогостоящего оборудования, которое отсутствует в большинстве учебных заведениях. При этом отметим, что весьма важную роль в образовательном процессе с использованием виртуальных лабораторных практикумов играют технологии удаленного доступа к среде моделирования [11].

Статья подготовлена в рамках выполнения государственного контракта с Министерством образования и науки № 16.647.12.2020 от 25 ноября 2010 г.

Литература

1. Jones, A.C. Chemical Vapour Deposition. Precursors, Processes and Application / A.C. Jones, M.L. Hitchman. – London: RSC Publishing, 2009. – 582 с.

2. Протопопова, В.С. Химическое осаждение из газовой фазы слоев Ni из бис-(этилциклопентадиенил) никеля / В.С. Протопопова, С.Е. Александров // Научно-технические ведомости СПбГПУ, серия «Физико-математические науки»,

№ 126. – СПб.: Изд-во Политехн. ун-та, 2011. – с. 145–150.

3. Уваров, А.А. Химическое осаждение из газовой фазы диэлектрических пленок политетрафторэтилена / А.А. Уваров, С.Е. Александров. // Научно-технические ведомости СПбГПУ, серия «Физико-математические науки», № 126. – СПб.: Изд-во Политехн. ун-та, 2011. – с. 141–145.
4. Александров, С.Е. Технология материалов электронной техники. Процессы химического осаждения из газовой фазы: учеб. пособие. – СПб.: Изд-во Политехн. ун-та, 2005. – 92 с.
5. Hitchman, M.L. Chemical Vapor Deposition, Principals and Application / M.L. Hitchman, K.F. Jencen – London: Academic Press, 1993. – 678 p.
6. Лойцянский, Л.Г. Механика жидкости и газа: учеб. для вузов. / Л.Г. Лойцянский. Изд. 6-е, перераб. и доп. – М.: Наука, 1987. – 600 с.
7. FLUENT 6.3 User's Guide
URL: http://hpce.iitm.ac.in/website/Manuals/Fluent_6.3/ (дата обращения: 12.03.2012)
8. Mazumder, S. The Importance of Predicting Rate-limited Growth for Accurate Modeling of Commercial MOCVD Reactors / S. Mazumder, S. Lowry // J. Crystal Growth, 2001. – Vol. 224. № 1–2. – P. 165–174
9. CHEMKIN/CHEMKIN-PRO Input Manual (August 2010)
10. Chase, M.W. NIST-JANAF Thermochemical Tables, 4th Edition. Monograph No. 9 / M.W. Chase – National Institute of Standards and Technology, 1998. – 1952 p.
11. Иванов, Д.И. Визуализация результатов моделирования процессов газофазного синтеза наноразмерных структур при сетевом доступе к кластерному вычислителю / Д.И. Иванов, Н.В. Захаревич, И.А. Цикин. – СПб.: Изд-во Политехн. ун-та, Научно-технические ведомости СПбГУ (в печати).
12. Laidler, K.J. Chemical Kinetics, Third Edition. / K.J. Laidler – Benjamin-Cummings, 1997.

Юрий Яковлевич Болдырев, д.т.н., профессор, заведующий кафедрой Математического и программного обеспечения высокопроизводительных вычислений, директор Отделения вычислительных ресурсов Информационно-телекоммуникационного комплекса, Санкт-Петербургский государственный политехнический университет. e-mail: boldyrev@phmf.spbstu.ru

Кирилл Юрьевич Замотин, начальник отдела прикладных программных систем Информационно-телекоммуникационного комплекса, Санкт-Петербургский государственный политехнический университет.

Евгений Павлович Петухов, начальник отдела системного программного обеспечения Информационно-телекоммуникационного комплекса, Санкт-Петербургский государственный политехнический университет.

MODELING OF CHEMICAL VAPOR DEPOSITION FOR GROWTH OF THIN FILMS

Y. Boldyrev, Saint Petersburg State Polytechnical University (Saint Petersburg, Russian Federation)**,**
K. Zamotin, Saint Petersburg State Polytechnical University (Saint Petersburg, Russian Federation)**,**
E. Petukhov, Saint Petersburg State Polytechnical University (Saint Petersburg, Russian Federation)

Most of the tasks that are associated with many aspects of nanotechnology development are essentially interdisciplinary by its nature. One of the most striking example is the use of gas-phase synthesis problems in nanotechnology. In essence, these technologies are the realization of the processes of the solid state chemical deposition from gaseous substance supplied to the reaction zone.

A classic experiment in the learning process has shown its weakness: not clear; does not allow to study the dependence of the final material characteristics from the different physical parameters of the system; time consuming and expensive. By these reasons experiment was replaced by a virtual experiment i.e. simulation.

At the base of the work lies the development and testing of mathematical models using high-performance computing in the processes of gas-phase synthesis of nanostructures and nanomaterials in order to study and provide visualization of proceeding physical and chemical processes.

Keywords: nanotechnology industry, gas-phase synthesis nanomaterials, simulation, gas dynamics, physical and chemical processes.

References

1. Jones A.C., Hitchman M.L. Chemical Vapour Deposition. Precursors, Processes and Application. London: RSC Publishing, 2009. 582 p.
2. Protopova V.S., Alexandrov S.E. Himicheskoe osazhdenie iz gazovoj fazy sloev Ni iz bis-(jetilciklopentadienil) nikelja [Chemical Deposition of Ni-layers from Gas Phase]. SPb.: Izd-vo Politehn. un-ta. Nauchno-Tehnicheskie vedomosti SPbGPU, seria "Fiziko-matematicheskie nauki" [Scientific and Technical Bulletin of SPbSTU: Physics and Mathematics], No. 126, 2011. P. 145–150.
3. Uvarov A.A, Aleksandrov S.E. Himicheskoe osazhdenie iz gazovoj fazy dijelektricheskih plenok politetraftorjetilena [Chemical Deposition of Dielectric Films of Polytetrafluoroethylene from Gas Phase]. Nauchno-Tehnicheskie vedomosti SPbGPU, seria "Fiziko-matematicheskie nauki" [Scientific and Technical Bulletin of SPbSTU: Physics and Mathematics], No. 126, 2011. P. 141–145.
4. Aleksandrov S.E. Tehnologija materialov jelektronnoj tehniki. Processy himicheskogo osazhdenija iz gazovoj fazy: Ucheb. posobie [Electronic Material Technology. Processes of Chemical Deposition from Gas Phase: Tutorial]. SPb.: Izd-vo Politehn. un-ta, 2005. 92 p.
5. Hitchman M.L., Jencen K.F. Chemical Vapor Deposition, Principals and Application. London: Academic Press, 1993. 678 p.

6. Loicansky L.G. Mekhanika zhidkosti i gaza [Mechanics of Fluids and Gases]. Moscow, Nauka, 1987. 600 p.
7. FLUENT 6.3 User's Guide
URL: http://hpce.iitm.ac.in/website/Manuals/Fluent_6.3/
8. Mazumder S., Lowry S. The Importance of Predicting Rate-limited Growth for Accurate Modeling of Commercial MOCVD Reactors. J. Crystal Growth, 2001. Vol. 224. № 1–2. P. 165–174
9. CHEMKIN/CHEMKIN-PRO Input Manual (August 2010)
10. NIST-JANAF Thermochemical Tables, 4th Edition, M. Chase Monograph No. 9: 1998, 1952 pages, 2 volumes, hardcover, ISBN 1-56396-831-2
11. Ivanov D.I., Zaharevich N.V., Cikin I.A. Vizualizacija rezul'tatov modelirovanija processov gazofaznogo sinteza nanorazmernyh struktur pri setevom dostupe k klasternomu vychislitelju SPb.: Izd-vo Politehn. un-ta Nauchno-Tehnicheskie vedomosti SPbGPU, seria "Fiziko-matematicheskie nauki" (v pechati) [Scientific and Technical Bulletin of SPbSTU: Physics and Mathematics (pending)].
12. Laidler K.J. Chemical Kinetics, Third Edition. Benjamin-Cummings, 1997.

Поступила в редакцию 2 апреля 2012 г.

УДК 519.63

ПАРАЛЛЕЛЬНЫЕ МЕТОДЫ И ТЕХНОЛОГИИ ДЕКОМПОЗИЦИИ ОБЛАСТЕЙ[1]

В.П. Ильин

Рассматриваются параллельные методы декомпозиции областей для решения трехмерных сеточных краевых задач, получаемых в результате конечно-элементных или конечно-объемных аппроксимаций. Данные проблемы являются «узким горлышком» среди различных этапов математического моделирования, поскольку современные требования к разрешающей способности сеточных алгоритмов приводят к необходимости решения систем линейных алгебраических уравнений с числом неизвестных в сотни миллионов и с очень плохой обусловленностью, что вызывает экстремальную ресурсоемкость расчетов. Описываются многопараметрические варианты алгоритмов с различной размерностью декомпозиции — одномерной, двумерной и трехмерной, — с пересечением или без пересечения подобластей, при использовании величин перехлеста как оптимизирующих параметров, а также с различными видами внутренних условий сопряжения на смежных границах (Дирихле, Неймана или третьего рода). Исследуются вариационные итерационные процессы крыловского типа в пространствах следов с разными предобуславливающими подходами: операторы Пуанкаре–Стеклова, блочный метод Чиммино, альтернирующий метод Шварца аддитивного типа, а также грубо-сеточная коррекция, являющаяся в определенном смысле упрощенным вариантом алгебраического многосеточного подхода. Проводится сравнительный анализ критериев эффективности распараллеливания на многопроцессорных вычислительных системах.

Ключевые слова: декомпозиция областей, трехмерные краевые задачи, сеточные аппроксимации, параллельные итерационные алгоритмы в пространствах Крылова, предобуславливающие операторы.

Введение

Разбиение сложной задачи на более простые подзадачи традиционно является распространенным подходом к построению эффективных численных методов решения многомерных задач математического моделирования, описываемых системами дифференциальных или/и интегральных уравнений, а также эквивалентными вариационными постановками. Здесь можно выделить такие структурные принципы, как расщепление по координатным осям (неявные методы переменных направлений), расщепление операторов (алгоритмы дробных шагов), расщепление по физическим процессам, а также декомпозиция расчетной области на подобласти [1–6]. Эти же подходы, из которых мы остановимся только на последнем — методе декомпозиции областей (МДО), — лежат в основе достижения масштабируемого распараллеливания на современных многопроцессорных — многоядерных вычислительных системах (МВС), в том числе гетерогенной архитектуры, когда каждый вычислительный узел содержит один или несколько графических процессорных элементов (ГПЭ) общего назначения, обладающих высокой скоростью выполнения арифметических операций.

С точки зрения эффективного распараллеливания, необходимо рассмотреть типовую структуру алгоритмов, реализуемых в крупномасштабных вычислительных экспериментах посредством различных вложенных циклов. К самому внутреннему уровню следует отнести решение систем линейных алгебраических уравнений (СЛАУ)

[1]Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012».

высокого порядка (десятки или сотни миллионов), которое осуществляется с помощью предобусловленных крыловских итерационных процессов. Эти «линейные» итерации могут иметь многоуровневый характер, когда СЛАУ имеют крупноблочную структуру, что имеет место при сеточной аппроксимации систем исходных функциональных уравнений с неизвестными векторными функциями. Следующий уровень — проведение нелинейных итераций, если коэффициенты и правые части алгебраической системы зависят от искомого решения. Для нестационарных задач предыдущие операции выполняются в цикле по временным шагам. Все указанные выше действия составляют реализацию прямых задач, а в случае решения обратных задач с применением оптимизационных подходов осуществляется внешний цикл с последовательным решением прямых задач и корректировкой целевого функционала на каждой итерации. «Узким горлышком» являются как раз самые простые — линейные — задачи, представляющие наибольший интерес с точки зрения распараллеливания, поскольку требуемый ими объем вычислений нелинейным образом растет при увеличении размерности, или числа степеней свободы.

Итоговая производительность программной реализации численных методов, конечно, значительно зависит от технологических аспектов, возникающих на различных стадиях наукоемких расчетов: геометрического и функционального моделирования, включающего интеллектуальные пользовательские интерфейсы, генерации адаптивных сеток (дискретизации расчетной области), аппроксимации исходной задачи методами конечных элементов, конечных объемов (МКЭ, МКО) и т.д., решения возникающих алгебраических систем сверхвысоких порядков, реализации оптимизационных процедур (в обратных задачах), а также постпроцессинга и визуализации результатов, см. [7]. Параллельные алгоритмы декомпозиции и формирование соответствующих структур данных, распределенных по вычислительным узлам МВС, должны рассматриваться комплексно на всех указанных этапах математического моделирования.

Однако главной целью данной работы является обзор и сравнительный анализ итерационных алгебраических методов декомпозиции областей, базирующихся, во-первых, на эффективном предобуславливании исходных сеточных СЛАУ и, во-вторых, на построении крыловских алгоритмов в пространствах следов, т.е. для функций и операторов типа Пуанкаре–Стеклова, определенных на внутренних границах смежных подобластей.

Помимо приводимых нами в списке литературы основных монографий [3–6] и некоторых статей [8–17] (отметим здесь значительный вклад новосибирских и московских математиков), по данной теме огромное количество материала имеется на интернетовском сайте [18], включая труды уже состоявшихся 20 специальных конференций.

1. Математические и технологические вопросы декомпозиции областей

Представим общую идею МДО следующим образом. Пусть в расчетной трехмерной области Ω с границей Γ требуется решить краевую задачу

$$Lu = f(\vec{r}), \;\; \vec{r} \in \Omega; \quad lu|_\Gamma = g, \tag{1}$$

где линейные дифференциальные операторы L и l обеспечивают существование единственного решения $u(\vec{r})$ в некотором функциональном пространстве. Разобъем область Ω на P пересекающихся или не пересекающихся подобластей Ω_q и введем следующие обозначения:

$$\begin{aligned} &\Omega = \bigcup_{q=1}^{P} \Omega_q, \;\; \bar{\Omega} = \Omega \bigcup \Gamma, \;\; \bar{\Omega}_q = \Omega_q \bigcup \Gamma_q, \\ &\Gamma_q = \bigcup_{q' \in \omega_q} \Gamma_{q,q'}, \;\; \Gamma_{q,q'} = \Gamma_q \bigcap \bar{\Omega}_{q'}, \;\; q' \neq q, \end{aligned} \tag{2}$$

где ω_q есть множество номеров подобластей, соседних к Ω_q, Γ_q — вся граница q-й подобласти, а $\Gamma_{q,0}$ — ее внешняя часть, т.е. Ω_0 обозначает «внешнюю» подобласть — дополнение к Ω в $\mathcal{R}^3$. Вместо исходной задачи (1) рассматриваем P краевых подзадач в подобластях Ω_q:

$$\begin{aligned} &Lu_q(\vec{r}\,) = f_q, \quad \vec{r} \in \Omega_q, \\ &l_{q,q'}(u_q)|_{\Gamma_{q,q'}} = g_{q,q'} \equiv l_{q',q}(u_{q'})\big|_{\Gamma_{q',q}}, \;\; u_q|_{\Gamma_{q,q'}} = u_{q'}|_{\Gamma_{q,q'}}, \\ &q' \in \omega_q, \quad l_{q,0}u_q|_{\Gamma_{q,0}} = g, \quad q = 1, ..., P, \end{aligned} \tag{3}$$

где $l_{q,q'}$ и $g_{q,q'}$ при $q' \neq 0$ определяют некоторые «внутренние» граничные условия такие, которые не «портили» бы исходное решение u, т.е. $u_q(\vec{r}) = u(\vec{r})$ при $\vec{r} \in \Omega_q$. Например, в достаточно общем случае эти условия имеют вид

$$\begin{aligned} &\alpha_q u_q + \beta_q \frac{\partial u_q}{\partial n_q}\Big|_{\Gamma_{q,q'}} = g_{q,q'} \equiv \alpha_{q'} u_{q'} + \beta_{q'} \frac{\partial u_{q'}}{\partial n_{q'}}\Big|_{\Gamma_{q',q}}, \\ &|\alpha_q| + |\beta_q| > 0, \quad \alpha_q \cdot \beta_q \geq 0, \end{aligned} \tag{4}$$

причем для $\beta_q = \beta_{q'} = 0$ они соответствуют условию 1-го рода (Дирихле), при $\alpha_q = \alpha_{q'} = 0$ — условию 2-го рода (Неймана), а в остальных случаях — условию 3-го рода (Робина). Естественным методом решения системы краевых задач (3) является организация простых итераций, т.е. блочного метода Якоби вида

$$Lu_q^n = f_q, \;\; l_{q,q'}u_q^n|_{\Gamma_{q,q'}} = l_{q',q}u_{q'}^{n-1}|_{\Gamma_{q',q}}, \tag{5}$$

где правые части граничных условий для каждой подобласти определяются по значениям решения с прошлой итерации в смежных подобластях.

Далее в качестве иллюстрации исходной задачи (1) мы будем рассматривать аппроксимацию уравнения Пуассона в кубической области с граничными условиями Дирихле на кубической сетке с общим числом узлов M^3:

$$\begin{aligned} &-\Delta u = f, \quad u|_\Gamma = g, \;\; \Omega = [0 \times 1]^3; \;\; \Omega^h = \{i, j, k\}, \\ &(Au^h)_{i,j,k} = 6u^h_{i,j,k} - u^h_{i-1,j,k} - u^h_{i,j-1,k} - \\ &-u^h_{i+1,j,k} - u^h_{i,j+1,k} - u^h_{i,j,k-1} - u^h_{i,j,k+1} = f^h_{i,j,k}; \\ &i, j, k = 1, ..., M, \;\; f^h = \{f^h_{i,j,k}\}, \;\; u^h = \{u^h_{i,j,k}\} \in \mathcal{R}^{M^3}. \end{aligned} \tag{6}$$

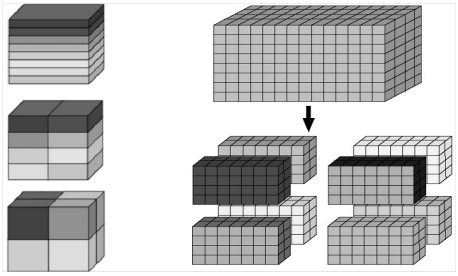


Рис. 1. Примеры структурированных 1D-, 2D- и 3D-декомпозиций области

Отметим, что с точек зрения как скорости сходимости итераций, так и технологии распараллеливания, существенное значение имеет размерность декомпозиции, варианты которой представлены на рис. 1. Для простой расчетной области в форме параллелепипеда размерность очевидным образом определяется как количество типов координатных плоскостей, используемых при построении подобластей. Например, при 1-D декомпозиции область разбивается на P слоев с помощью параллельных плоскостей.

2. Итерационные методы в пространстве следов

Рассмотрим основные алгоритмические вопросы МДО на примере одномерной декомпозиции с пересечением подобластей описанной в (6) сеточной краевой задачи, см. рис. 2.

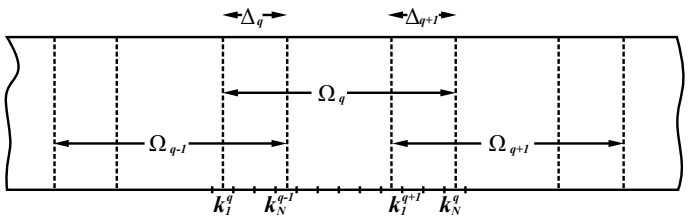


Рис. 2. Пример одномерной декомпозиции с пересечениями подобластей

Через Ω_q^h обозначаем сеточную подобласть, у которой первая и последняя координатные плоскости $z = z_k$ имеют номера k_1^q и k_N^q соответственно. Все подобласти, как и их пересечения Δ_q, считаем одинаковыми. Введем также следующие обозначения:

$$\begin{aligned} &N = dim(\Omega_q^h),\ m = dim(\Delta_q^h),\ q = 1, ..., P, \\ &N = k_N^q - k_1^q + 1,\ \ m = k_N^q - k_1^{q+1} + 1,\ \ M = PN - (P-1)m, \end{aligned} \tag{7}$$

Здесь N и m означают число z-плоскостей в Ω_q и Δ_q, M есть «размерность по z» всей сеточной области Ω^h,а P — общее количество подобластей.

Вводя подвекторы $u_q = (u_{k_1^q}, ..., u_{k_N^q})^T,\ u_{k_l^q} = \{u_{i,j,k_l^q};\ i, j, = 1, ..., M\} \in \mathcal{R}^{M^2}$, а также аналогичные подвекторы правых частей f_q порядка M^2N, системы уравнений для подобластей можно записать в следующем блочно-трехдиагональном виде:

$$\begin{aligned} &-A_{q,q-1}u_{q-1} + A_{q,q}u_q - A_{q,q+1}u_{q+1} = f_q,\ \ q = 1, ..., P, \\ &A_{1,0} = A_{P,P+1} = 0,\ \ A_{q,q}, A_{q,q\pm1} = A_{q\pm1,q}^T \in \mathcal{R}^{M^2N, M^2N}. \end{aligned} \tag{8}$$

Итерационный процесс (5) при использовании значений $u^{n-1}_{q\pm1}$ с предыдущего шага из соседних подобластей имеет вид

$$\begin{aligned} &A_{q,q}u^n_q = \bar f^{n-1}_q \equiv f_q + \hat f^{n-1}_q + \check f^{n-1}_q, \\ &\hat f^{n-1}_q = A_{q,q-1}u^{n-1}_{q-1},\ \check f^{n-1}_q = A_{q,q+1}u^{n-1}_{q+1}, \end{aligned} \tag{9}$$

который в по-компонентной форме записывается следующим образом:

$$(A_{q,q}u^n_q)_k \equiv \begin{cases} (C-\theta I)u^n_{k^q_1} - u^n_{k^q_1-1} = f_{k^q_1} + v^{n-1}_{q-1}, \\ v^{n-1}_{q-1} = u^{n-1}_{k^q_1-1} - \theta u^{n-1}_{k^q_1}, \qquad k = k^q_1, \\ (C-\theta I)u^n_{k^q_N} - u^n_{k^q_N+1} = f_{k^q_N} + w^{n-1}_{q+1}, \\ w^{n-1}_{q+1} = u^{n-1}_{k^q_N+1} - \theta u^{n-1}_{k^q_N}, \qquad k = k^q_N, \\ -u^n_{k-1} + Cu^n_k - u^n_{k+1} = f_k, \\ k = k^q_1+1, ..., k^q_N - 1. \end{cases}$$

Здесь I — единичная, а C — пятидиагональная матрицы порядка M^2, т.е.

$$\begin{aligned} &(C-\theta I)u^n)_k)_{i,j} = \{(6-\theta)u^n_{i,j,k} - u^n_{i-1,j,k} - u^n_{i+1,j,k} - u^n_{i,j-1,k} - u^n_{i,j+1,k}\}, \\ &C \in \mathcal{R}^{M^2},\ \theta \in [0,1],\ u^{n-1}_{k^q_1} \in \Omega_{q-1}, \end{aligned}$$

а θ — итерационный параметр, регулирующий тип итерируемого граничного условия на смежных границах подобластей: значения $\theta = 0$, $\theta = 1$ и $0 < \theta < 1$ соответствуют условиям Дирихле, Неймана и Робина.

Вводя далее подвекторы v_q и w_q размерности M^2, соответствующие лежащим в Ω_q сеточным слоям с номерами k^{q-1}_N и k^{q+1}_1, итерационные соотношения (9) приведем к редуцированному виду

$$\begin{aligned} &v^n_q = \hat B_{q,q-1}w^{n-1}_{q-1} + \hat B_{q,q+1}v^{n-1}_{q+1} + \hat g_q,\ \ q = 2, ..., P, \\ &w^n_q = \check B_{q,q-1}w^{n-1}_{q-1} + \check B_{q,q+1}v^{n-1}_{q+1} + \check g_q,\ \ q = 1, ..., P-1, \\ &\hat B_{1,0} = \hat B_{P,P+1} = 0, \end{aligned} \tag{10}$$

где используются следующие обозначения (см. подробнее [9]):

$$\begin{aligned} &v_q = C_{q,q-1}u_q,\ \ w_q = C_{q,q+1}u_q,\ \ A_{q,q\pm1} = Q_{q,q\pm1}C_{q,q\pm1}, \\ &\check g_q = C_{q,q+1}A^{-1}_{q,q}f_q, \\ &\hat B_{q,q\pm1} = C_{q,q-1}A^{-1}_{q,q}Q_{q,q\pm1},\ \ \check B_{q,q\pm1} = C_{q,q+1}A^{-1}_{q,q}Q_{q\pm1}. \end{aligned}$$

Система уравнений (10) получена из (9) исключением «внутренних» неизвестных для подобластей и содержит только подвекторы, соответствующие смежным границам. Объединяя их в «большие» векторы следов $s = (w_1, v_2, ..., w_{p-1}, v_p)^T$, $g = (\check g_1, \hat g_2, ..., \check g_{p-1}, \hat g_p)^T$, размерности $2M^2(P-1)$, итерации (10) перепишем в сжатом виде

$$s^n = Ts^{n-1} + g, \quad n = 1, 2, ..., \tag{11}$$

где явный вид матрицы T нам не потребуется. Отметим только, что умножение на нее включает решение алгебраических подсистем для всех подобластей. Очевидно,

что если итерационный процесс (11) в пространстве следов сходится, т.е. $s^n \to s$, то предельный вектор удовлетворяет системе уравнений

$$\bar{A}s \equiv (I - T)s = g, \tag{12}$$

в которой $\bar{A}$ есть некоторая предобусловленная (по отношению к A из (6)) матрица.

Эффективное численное решение полученной предобусловленной СЛАУ реализуется с помощью какого-либо из итерационных алгоритмов в подпространствах Крылова, см. [19] и цитируемую там литературу. Например, если $\bar{A}$ есть симметрично положительно определенная матрица, то целесообразно применять методы сопряженных градиентов или сопряженных невязок, которые при $\nu = 0, 1$ соответственно описываются следующими единообразными формулами:

$$r^0 = g - \bar{A}s^0 = \hat{s}^1 - s^0, \;\; \hat{s}^1 = Ts^0 + g, \;\; p^0 = r^0, \;\; n = 1, 2, \ldots :$$
$$s^{n+1} = s^n + \alpha_n^{(\nu)} p^n, \;\; \alpha_n^{(\nu)} = \rho_n^{(\nu)} / \delta_n^{(\nu)}, \;\; \rho_n^{(\nu)} = (\bar{A}^\nu r^n, r^n),$$
$$\delta_n^{(\nu)} = (\bar{A}p^n, \bar{A}^{(\nu)} p^n), \;\; \beta_n^{(\nu)} = \rho_{n+1}^{(\nu)} / \rho_n^{(\nu)},$$
$$r^{n+1} = r^n - \alpha_n^{(\nu)} \bar{A}p^n, \;\; p^{n+1} = r^{n+1} + \beta_n^{(\nu)} p^n.$$

В более общем случае, когда $\bar{A}$ несимметрична, алгоритмы строятся на основе A^ν-ортогонализации Арнольди ($\nu = 0$ или $\nu = 1$):

$$u^n = u^0 + y_1 v^1 + \ldots + y_n v^n, \;\; (v^n, A^\nu v^k) = d_n^{(\nu)} \delta_{k,n}, \;\; d_n^{(\nu)} = (v^n, A^\nu v^n),$$
$$v^{n+1} = Av^n - \textstyle\sum_{k=1}^n h_{k,n}^{(\nu)} v^k, \;\; v^1 = r^0 = f - Au^0,$$
$$h_{k,n}^{(\nu)} = \tfrac{(Av^n, A^\nu v^k)}{(A^s v^k, v^k)}, \;\; k = 1, \ldots, n+1, \;\; V_{n+1} = (v^1, \ldots, v^{n+1})$$
$$\bar{H}_n = \{h_{k,n}^{(\nu)}\} = \begin{bmatrix} H_n \\ e_n^t \end{bmatrix} \in \mathcal{R}^{n+1,n}, \;\; H_n \in \mathcal{R}^{n,n}.$$

При этом коэффициенты $y_k, k = 1, \ldots, n$, разложения итерационного приближения u^n по базисным векторам $v^1, \ldots, v^n$ находятся из условий ортогональности или минимальности векторов невязок r^k, что приводит к методам полной ортогональности или обобщенных минимальных невязок (FOM, A-FOM, GMRES, A-GMRES).

3. Итерационные процессы на основе операторов Пуанкаре–Стеклова

В данном пункте мы рассмотрим частный случай декомпозиции на две подобласти без пересечения ($P = 2, \Delta = 0$), в которых решаются уравнения Пуассона

$$-\Delta u_q = f_q, \;\; u|_\Gamma = g, \; q = 1, 2, \tag{13}$$

совместно определяющие в расчетной области $\bar{\Omega} = \bar{\Omega}_1 \bigcup \bar{\Omega}_2$ решение задачи Дирихле, обладающее свойствами непрерывности на общей границе подобластей $\Gamma_{1,2}$:

$$-\Delta u = f, \;\; u = u_1 \bigcup u_2; \;\; \Gamma_{1,2} : u_1 = u_2, \;\; -\frac{\partial u_1}{\partial n_1} = \frac{\partial u_2}{\partial n_2}. \tag{14}$$

Задавая на внутренней границе произвольное начальное приближение и вычисляя решения соответствующих подзадач

$$u_1^0 = u_2^0\big|_{\Gamma_{1,2}} = u^0_{\Gamma_{1,2}},\ \ -\Delta u_q^0 = f_q,\ \ u^0|_\Gamma = g, \tag{15}$$

для функций ошибки $v_q \equiv u_q - u_q^0,\ q = 1, 2$, мы получаем следующие уравнения:

$$-\Delta v_q = 0,\ \ v_q|_\Gamma = 0,\ \ v_q|_{\Gamma_{1,2}} = u_{\Gamma_{1,2}} - u^0_{\Gamma_{1,2}} = v_{\Gamma_{1,2}}. \tag{16}$$

Отметим, что в силу условий непрерывности исходного решения, имеет место равенство

$$\frac{\partial v_1}{\partial n_1} + \frac{\partial v_2}{\partial n_2} = \varphi \equiv -\Big(\frac{\partial u_1^0}{\partial n_1} + \frac{\partial u_2^0}{\partial n_2}\Big). \tag{17}$$

Определяя теперь на $\Gamma_{1,2}$ операторы Пуанкаре–Стеклова S_q

$$\frac{\partial v_q}{\partial n_q} = S_q^{-1} v_{\Gamma_{1,2}} = S_q^{-1}(u_{\Gamma_{1,2}} - u^0_{\Gamma_{1,2}}),\ \ q = 1, 2, \tag{18}$$

мы приходим к уравнению

$$Au_{\Gamma_{1,2}} \equiv (S_1^{-1} + S_2^{-1})u_{\Gamma_{1,2}} = \psi \equiv (S_1^{-1} + S_2^{-1})u^0_{\Gamma_{1,2}} - \Big(\frac{\partial u_1^0}{\partial n_1} + \frac{\partial u_2^0}{\partial n_2}\Big), \tag{19}$$

которое фактически представляет новую формулировку метода декомпозиции областей. Очевидно, что умножение на оператор S_q^{-1} включает решение краевой подзадачи в Ω_q.

Оказывается, что уравнение (19) может быть эффективно предобусловлено следующим образом:

$$\bar{A}u \equiv BAu = B\psi,\ \ B = S_1 + S_2. \tag{20}$$

Предобусловленный оператор $\bar{A}$ обладает при этом замечательными свойствами:

$$\begin{aligned}
&\bar{A} = (S_1 + S_2)(S_1^{-1} + S_2^{-1}) = I + S_1 S_2^{-1} + I + S_2 S_1^{-1} = \\
&= A_1 + A_2,\ \ A_1 = I + S_1 S_2^{-1},\ \ A_2 = I + S_2 S_1^{-1}, \\
&A_1 A_2 = A_1 + A_2 = A_2 A_1,
\end{aligned}$$

т.е. он представим в виде суммы перестановочных операторов A_1, A_2. Это позволяет, в частности, применять для решения (20) сверхбыстрые неявные методы переменных направлений [1, 2, 19]. Отметим также, что в работе [9] на основе операторов Пуанкаре–Стеклова построен оптимальный (сходящийся за 2 итерации) метод Шварца при декомпозиции области на две непересекающиеся подобласти.

4. Блочный метод Чиммино

Фактически проективный подход к алгебраической декомпозиции реализуется в блочном методе Чиммино, см. [20] и цитируемую там литературу. Разбивая вектор правой части $f = (f_1, ..., f_P)^T$ на подвекторы, мы можем записать СЛАУ в блочном виде

$$Au \equiv \begin{bmatrix} A_1 \\ \vdots \\ A_P \end{bmatrix} u = \begin{bmatrix} f_1 \\ \vdots \\ f_P \end{bmatrix} \equiv f,\ \ A \in \mathcal{R}^{N,N}; u, f \in \mathcal{R}^N, \tag{21}$$

где A_k суть блочные строки матрицы, которые для простоты считаем содержащими одинаковое число M строк. Записывая соответствующие подсистемы в виде

$$A_k u = f_k, \;\; f_k \in \mathcal{R}^M, \;\; A_k \in \mathcal{R}^{M,N}, \;\; k = 1, ..., P, \;\; N = PM, \tag{22}$$

и вводя вспомогательные подвекторы

$$v_k^n = A_k^+ r_k^n, \;\; r_k^n = f_k - A_k u^n, \;\; A^+ = A_k^T (A_k A_k^T)^{-1}, \tag{23}$$

мы можем определить итерационный процесс

$$u^{n+1} = u^n + \omega \sum_{k=1}^{P} v_k^n, \;\; n = 0, 1, ..., \tag{24}$$

где ω есть некоторый итерационный параметр, а A^+ — псевдообратная матрица, корректно определяемая, если A_k есть матрица полного ранга.

Легко видеть, что на каждом n-м шаге вычисляются («одновременно» для всех k) в некотором смысле приближенные решения в соответствующих подобластях.

Стационарный итерационный процесс (24), очевидно, может быть ускорен с помощью какого-то из крыловских методов. Рассмотренный алгоритм можно усилить, если в (23) A_k^+ заменить на обобщенную псевдообратную матрицу

$$A_k^{+G} = G_k^{-1} A_k^T (A_k G_k^{-1} A_k^T)^{-1}, \;\; G_k \in \mathcal{R}^{M,M}, \tag{25}$$

где G_k — некоторая симметричная положительно определенная матрица. Умножение на участвующую в (25) обратную матрицу может быть реализовано с помощью решения вспомогательной системы с седловой точкой

$$\begin{gathered} \left[\begin{array}{cc} G_k & A_k^T \\ A_k & 0 \end{array} \right] \left[\begin{array}{c} w_k^n \\ v_k^n \end{array} \right] = \left[\begin{array}{c} 0 \\ r_k^n \end{array} \right], \\ G_k w_k^n = -A_k^T v_k^n, \;\; v_k^n = A_k^{+G} r_k^n. \end{gathered} \tag{26}$$

Понятно, что существующий большой произвол в выборе матриц G_k дает потенциально значительные возможности в ускорении итераций.

5. Аддитивный метод Шварца с грубо-сеточной коррекцией

Если расчетную облалсть представить в форме $\Omega = \Omega_q \bigcup \tilde{\Omega}_q$, где $\bar{\Omega}_q$ — дополнение к подобласти Ω_q в Ω, а также ввести обозначения

$$u = \left[\begin{array}{c} u_q \\ \tilde{u}_q \end{array} \right], \;\; u_q = R_q u, \tag{27}$$

где $R_q = [0I0]$ — матрица продолжения, а $R_q^T : u \to u_q$ — матрица сужения вектора, то итерационный метод декомпозиции записывается в виде

$$u^{n+1} = u^n + \sum_{q=1}^{P} B_q (f - A u^n) = u^n + B r^n, \;\; B = \sum_{q=1}^{P} B_q, \tag{28}$$

который называется аддитивным алгоритмом Шварца. Здесь предобуславливающая матрица B состоит из слагаемых

$$B_q = R_q^T (R_q A R_q^T)^{-1} R_q, \tag{29}$$

каждое из которых является симметричным при $A = A^T$ и положительно определенным, если таким же свойством обладает A, а матрица R_q имеет полный ранг.

Матричное произведение $P_q = B_q A$ обладает свойством $P_q^2 = P_q$, то есть является ортогональным проектором в смысле A-скалярного произведения

$$(P_q u, v)_A = u^T P_q^T A v = u^T A B_q A v = (U, P_q v)_A.$$

Сходимость итерационного процесса (28) можно ускорить за счет какого-либо «улучшения» предобуславливателя B. Мы рассмотрим метод грубосеточной коррекции [4–6], который заключается в следующем.

Запишем исходную алгебраическую систему в виде

$$A_F u_F = f_F, \;\; u_F, f_F \in \mathcal{R}^{N_F}, \;\; A_F \in \mathcal{R}^{N_F, N_F}, \tag{30}$$

предполагая, что она получена из аппроксимации некоторой краевой задачи на густой сетке Ω_F и имеет достаточно большую размерность N_F. Определим теперь подвектор

$$u_c = R_c u_F, \;\; R_c \in \mathcal{R}^{N_c, N_F}, \;\; N_c \ll N_F,$$

который будем считать соответствующим некоторой редкой (грубой) сетке Ω_c, вложенной в Ω_F, а также дополнительный предобуславливающий оператор

$$B_c = R_c^T A_c^{-1} R_c, \;\; A_c = R_c A R_c^T \in \mathcal{R}^{N_c, N_c}.$$

Тогда для решения СЛАУ можно определить аддитивный метод Шварца с грубосеточной коррекцией следующего вида:

$$u_F^{n+1} = u_F^n + (B_c + B_F)(f_F - A_F u_F^n), \tag{31}$$

где $B_F = \sum_{q=1}^{P} B_q$ — новое обозначение предобуславливателя B из (28).

В частности, если матрица R_c^T соответствует оператору продолжения интерполяционного типа с сетки Ω_c на Ω_F, то получаемый алгоритм называется методом галеркинского вида. Естественно, что итерационный процесс (31) может быть ускорен с помощью крыловских подходов. Введение дополнительного предобуславливателя B_c реализует на каждой итерации взаимосвязи между дальними подобластями и устанавливает некоторый «мостик» между декомпозицией областей и многосеточными методами.

В заключение данного пункта отметим еще такой независимый подход к ускорению итераций, как метод дефляции, см. [17] и приведенный там обзор. В применении к СЛАУ (30) его можно описать с помощью матрицы некоторого проективного пространства $R_d \in \mathcal{R}^{N_F, k}$ с полным рангом и заданным $k < N_F$. Используя R_d, можно определить проектор

$$P_d = I - A_F B, \;\; B = R_d (R_d^T A_F R_d)^{-1} R_d^T$$

и решать вместо (30) предобусловленную им систему $P_d A_F u_F = P_d f_F$. При определенных выборах R_d (например, использование приближенных собственных векторов A_F) метод дефляции дает значительное ускорение итераций.

6. Вопросы распараллеливания методов декомпозиции

Основные критерии распараллеливания алгоритмов — это коэффициенты ускорения S_p и эффективности использования процессоров E_p, которые выражаются через время выполнения арифметических операций на P процессорах T_P^a и длительности межпроцессорных коммуникаций T_P^c:

$$\begin{aligned} &S_P = T_1/T_P, \quad E_P = S_P/P, \\ &T_P = T_P^a + T_P^c \approx \tau_a V_a + N_c(\tau_0 + \tau_c V_c). \end{aligned} \tag{32}$$

Здесь V_a означает общее количество арифметических действий, τ_a – некоторое усредненное время выполнения одной операции, N_c — число обменов одного процессора, V_c — количество передаваемых чисел за одну коммуникацию, τ_c — время передачи одного числа, а τ_0 – длительность задержки при каждом обмене, причем можно считать $\tau_a \ll \tau_c \ll \tau_0$. При этом вынужденно используется грубая модель вычислительного процесса, которая не учитывает многих факторов современных МВС: гетерогенности архитектуры и неоднородности иерархической памяти, использования конвейеризации и совмещения во времени арифметических действий с обменами, топологии физической реализации межпроцессорных соединений, особенностей функционирования многоядерных вычислительных узлов и т.д. В отсутствие реальной возможности реального имитационного моделирования компьютерной системы теоретические оценки ее производительности могут иметь только качественный характер, и практически единственным средством исследования эффективности распараллеливания алгоритмов решения определенного класса задач является численный эксперимент.

Проведем качественный сравнительный анализ эффективности распараллеливания d-D декомпозиции трехмерной области для описанной в (6) модельной сеточной задачи для различных размерностей $d = 1, 2, 3$, изображенных на рис. 1. Пусть во всех рассматриваемых случаях область разбивается на P одинаковых сеточных подобластей с числом узлов M^3/P. Количество же граничных узлов каждой подобласти равно

$$N_\Gamma^{(1)} = 2M^2\Big(\frac{2}{P} + 1\Big), \;\; N_\Gamma^{(2)} = 2M^2\Big(\frac{1}{P} + \frac{2}{P^{1/2}}\Big), \;\; N_\Gamma^{(3)} = \frac{6M^2}{P^{3/2}}$$

для $d = 1, 2, 3$ соответственно. Поскольку при $P \gg 1$ имеем $N_\Gamma^{(1)} \gg N_\Gamma^{(2)} \gg N_\Gamma^{(3)}$, а объем данных, передаваемых от одного процессора ко всем остальным, пропорционален числу поверхностных сеточных узлов, то очевидным образом получаем, что относительный вклад коммуникационных временных потерь уменьшается с увеличением размерности d.

Большую роль играет также величина, которую можно назвать псевдодиаметром $æ$ графа макросети, образуемой совокупностью подобластей. Расстоянием между двумя вершинами графа называется минимальное число ребер, по которым можно пройти из одной вершины в другую. А псевдодиаметр — это максимальное расстояние между какими-либо парами вершин. Если кубическую сеточную область из (6) разбить на P подобластей с помощью d-D декомпозиции, то псевдодиаметры будут равны $æ = \sqrt{d}P^{1/d}$, т.е. длине диагонали квадрата или куба при $d = 2, 3$ соответственно.

Очевидно, что при реализации блочного метода Якоби по подобластям информация от одной подобласти дойдет за одну итерацию только до соседних подобластей и дойдет до самой дальней подобласти за $æ$ итераций в худшем случае. Можно сделать естественное предположение (теоремы такой пока нет), что число итераций по подобластям, необходимое для достижения требуемой точности $\varepsilon \ll 1$, пропорционально $æ^{\gamma}|ln\varepsilon|$, $\gamma > 0$, т.е. при $P \geq 8$ (минимальное число подобластей в трехмерной декомпозиции), т.е. будет убывать с ростом размерности d.

Таким образом, с двух различных теоретических точек зрения, 3-D декомпозиция является самой предпочтительной. Однако на практическое соотношение производительности программных кодов, реализующих различные подходы, естественно, могут повлиять многочисленные технологические факторы. Для рассмотренной в п.3 одномерной декомпозиции на одной итерации времена выполнения арифметических действий и обменов для каждого процессора равны (C_1 — не зависящая от M и N константа)

$$T_P^a \cong C_1 M^2 N^{\gamma+1} \tau_a,$$
$$T_P^c \cong 2(\tau_0 + \tau_c M^2)$$

и при этом критерии эффективности распараллеливания данного алгоритма при $N \gg 1$ оцениваются величинами

$$S_P \cong T_P^a \cdot P/(T_P^a + T_P^c) \approx P, \quad E_P \approx 1,$$

т.е. обеспечивают примерно линейное ускорение с ростом P.

Как уже отмечалось, фактическая производительность программной реализации может сильно отличаться от теоретических оценок. В частности, для распараллеливания существенное значение имеет такая нетривиальная в общем случае сложных неструктурированных сеток задача, как построение сбалансированных по числу узлов подобластей, которую логично осуществлять на этапе дискретизации расчетной области. А вопросы оптимизации величин пересечений и типов внутренних граничных условий требуют дополнительных и аналитических, и экспериментальных исследований.

Работа поддержана грантом РФФИ №11-01-00205, а также грантами Президиума РАН №2.5 и ОМН РАН № 1.3.4.

Литература

1. Марчук, Г.И. Методы вычислительной математики / Г.И. Марчук. – М.: Наука, 1980.

2. Ильин, В.П. Методы конечных разностей и конечных объемов для эллиптических уравнений / В.П. Ильин. – Новосибирск, изд. ИВМ СО РАН, 2000.

3. Лебедев, В.И. Операторы Пуанкаре – Стеклова и их приложения в анализе / В.И. Лебедев, В.И. Агошков, – М.: Отдел вычислительной математики АН СССР, изд. ВИНИТИ, 1983.

4. Quarteroni, A. Domain Decomposition Methods for Partial Differential Equations / A. Quarteroni, A. Valli – Clarendon Press, Oxford, 1999.

5. Smith, B.F. Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations / B.F. Smith, P.E. Bjorstad, W.D. Gropp – Cambridge University Press, 2004.

6. Toselli, A. Domain Decomposition Methods. Algorithms and Theory / A. Toselli, O. Widlund – Springer, Berlin, 2005.

7. Ильин, В.П. Параллельные процессы на этапах петафлопного моделирования / В.П. Ильин // Вычислительные методы и программирование. – 2011. – Т. 12, № 1. – С. 93–99.

8. Nataf, F. Optimized Schwarz Methods. // Lecture Notes in Computer Science and Engineering. – Springer–Verlag, Berlin, 2009. – P. 233–240.

9. Ильин, В.П. Параллельные методы декомпозиции в пространствах следов / В.П. Ильин, Д.В. Кныш // Вычислительные методы и программирование. – Изд. МГУ, 2011. – Т. 12, № 1. – С. 100–109.

10. Смелов, В.В. Принцип итерирования по подобластям в задачах с эллиптическим уравнением. / В.В. Смелов В.В., Т.Б. Журавлева. – М.: Изд. ВИНИТИ, 1981. – (Препринт / ОВМ РАН; № 14).

11. Сандер, С.А. Модификация алгоритма Шварца для решения сеточных краевых задач в областях, составленных из прямоугольников и параллелепипедов. / С.А. Сандер. – Новосибирск, 1981. – (Препринт / Изд. ВЦ СО АН СССР; № 83).

12. Мацокин, А.М. Применение окаймления при решении систем сеточных уравнений / А.М. Мацокин, С.В. Непомнящих // Вычислительные алгоритмы в задачах матемаической физики – Новосибирск: Изд. ВЦ СО АН СССР, 1983. – С. 99–109.

13. Лебедев, В.И. Вариационные алгоритмы метода разделения области / В.И. Лебедев, В.И. Агошков – М., 1983. – (Препринт / ОВМ РАН, № 54).

14. Непомнящих, С.В. О применении метода окаймления к смешанной краевой задаче для эллиптических уравнений и осеточных нормах в $W_2^{1/2}(S)$. / С.В. Непомнящих. – Новосибирск, 1984. – (Препринт / Изд. ВЦ СОАН СССР, № 106).

15. Кузнецов, Ю.А. Новые алгоритмы приближенной реализации неявных разностных схем / Ю.А. Кузнецов – М., 1987. – (Препринт / ОВМ АН СССР, № 142).

16. Свешников, В.М. Построение прямых и итерационных методов декомпозиции / В.М. Свешников // Сиб. журн. индустр. математики. – 2009. – Т. 12, № 3(39). – С. 99–109.

17. Tang, J.M. Comparison of Two-level Preconditioners Derived from Deflation, Domain Decomposition and Multigrid Methods / J.M. Tang, R. Nabben, C. Vuik, Y.A. Erlangga // J. Sci. Comput. – 2009. – V. 39. – P. 340–370.

18. Domain Decomposition Methods.
URL: http://ddm.org (дата обращения: 14.03.2012)

19. Ильин, В.П. Методы и технологии конечных элементов / В.П. Ильин – Новосибирск, изд. ИВМиМГ СО РАН, 2007.

20. Ильин, В.П. Об итерационном методе Качмажа и его обобщениях. // Сиб. журн. индустр. математики. – 2006. – Т. 9, № 3. – С. 39–49.

Ильин Валерий Павлович, доктор физико-математических наук, профессор, главный научный сотрудник ИВМиМГ СО РАН, профессор кафедры вычислительной математики НГУ, ilin@sscc.ru.

PARALLEL METHODS AND TECHNOLOGIES OF DOMAIN DECOMPOSITION

V.P. Il'in, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation)

Parallel domain decomposition methods for solving 3-D grid boundary value problems, which are obtained by finite-element or finite-volume approximations are considered. These problems present the bottle neck between different stages of mathematical modelling, because the modern requirements to accuracy of grid algorithms provide the necessity of solving the systems of linear algebraic equations with the hundred millions of degrees of freedom and with super-high condition numbers which demand the extremal computing resourses. Multi-parameter versions of algorithms with various domain decomposition dimensions — one-dimensional, two-dimensional and three-dimensional, — with or without overlapping of subdomains and with different kinds of internal conjecture conditions on the adjacent boundaries (Dirichlet, Neuman and Robin). The iterative Krylov processes in the trace spaces are investigated for the different preconditioning approaches: Poincare – Steklov operators, block Cimmino method, alternating Schwartz algorithm of additive type, as well as coarse grid correction which is, in a sense, the simplified version of algebraic multigrid method. The comparative analysis of the criteria of parallelezation for the multiprocessor computer systems is made.

Ключевые слова: domain decomposition, tridimensional boundary value problems, grid approximations, parallel iterative algorithms in Krylov spaces, preconditioning operators.

References

1. Marchuk G.I. Metody vychislitelnoj matematiki [Numerical Analysis Methods] Moscow, Nauka, 1980.
2. Il'in V.P. Metody konechnyh raznostej i konechnyh ob"emov dlja jellipticheskih uravnenij [Finite Difference and Finite-volume Methods for Elliptic Partial Difference Equations]. Novosibirsk, Institute of Computational Modelling SB RAS, 2000.
3. Levedev V.I., Agoshkov V.I. Operatory Puankare – Steklova i ih prilozhenija v analize [Poincaré–Steklov Operators and their Application in Analysis] Moscow, Computational Mathematics Department of USSRAS, VINITI, 1983.
4. Quarteroni A. Valli A. Domain Decomposition Methods for Partial Differential Equations. Clarendon Press, Oxford, 1999.
5. Smith B.F., Bjorstad P.E., Gropp W.D. Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press, 2004.
6. Toselli A., Widlund O. Domain Decomposition Methods. Algorithms and Theory. Springer, Berlin, 2005.
7. Il'in V.P. Parallel'nye processy na jetapah petaflopnogo modelirovanija [Parallel Processes in Petaflopic Modeling]. Vychislitel'nye metody i programmirovanie [Numerical Methods and Programming], 2011. V. 12, No 1. P. 93–99.

8. Nataf F. Optimized Schwarz Methods. Lecture Notes in Computer Science and Engineering. Springer–Verlag, Berlin, 2009. P. 233–240.
9. Il'in V.P., Knysh D.V. Parallel'nye metody dekompozicii v prostranstvah sledov [Parallel Methods of Decomposition in Trace Spaces] Vychislitel'nye metody i programmirovanie [Numerical Methods and Programming]. MSU Publ., 2011. V. 12, No 1. P. 100–109.
10. Smelov V.V., Zhuravleva T.B. Princip iterirovanija po podoblastjam v zadachah s jellipticheskim uravneniem [Subdomain Iteration Principle in Partial Differential Equation Problems]. Moscow, VINITI, 1981.
11. Sander S.A. Modifikacija algoritma Shvarca dlja reshenija setochnyh kraevyh zadach v oblastjah, sostavlennyh iz prjamougol'nikov i parallelepipedov [Schwartz Algorithm Modification for Solving Grid Boundary Value Problems in Areas of Rectangles and Parallelepipeds. Novosibirsk, Preprint No 83, CC SB USSRAS, 1981.
12. Matsokin A.M., Nepomnyaschikh S.V. Применение окаймления при решении систем сеточных уравнений [Using the Bordering Method for Solving Systems of Mesh Equations]. Vychislitel'nye algoritmy v zadachah matematicheskoj fiziki [Numerical Methods in Mathematical Physics Problems]. Novosibirsk, CC SB USSRAS, 1981. P. 99–109.
13. Lebedev V.I., Agoshkov V.I. Variacionnye algoritmy metoda razdelenija oblasti [The Variational Algorithms of Domain Decomposition Method]. Moscow, Preprint No 54, Department of Numerical Mathematics of RAS, 1983.
14. Nepomnyaschikh S.V. O primenenii metoda okajmlenija k smeshannoj kraevoj zadache dlja jellipticheskih uravnenij i osetochnyh normah v $W_2^{1/2}(S)$] [On the Application of the Bordering Method to the Mixed Boundary Value Problem for Elliptic Equations and on Mesh Norms in $W_2^{1/2}(S)$]. Novosibirsk, Preprint No 106, CC SB USSRAS, 1984.
15. Kuznetsov Yu.A. Novye algoritmy priblizhennoj realizacii nejavnyh raznostnyh shem [New Algorithms for Approximate Implementation of Implicit Difference Schemes] Moscow, Preprint No 142, Department of Numerical Mathematics of RAS, 1987.
16. Свешников В.М. Построение прямых и итерационных методов декомпозиции [Construction of Direct and Iterative Decomposition Methods]. Sib. Zh. Ind. Mat. [J. Appl. Industr. Math.], 2009. V. 12, No 3(39). P. 99–109.
17. Tang J.M., Nabben R., Vuik C., Erlangga Y.A. Comparison of Two-level Preconditioners Derived from Deflation, Domain Decomposition and Multigrid Methods J. Sci. Comput. 2009. V. 39. P. 340–370.
18. Domain Decomposition Methods.
URL: http://ddm.org
19. Il'in V.P. Metody i tehnologii konechnyh jelementov [Finite Element Methods and Technologies] Novosibirsk, ICMMG SB RAS, 2007.
20. Il'in V.P. Ob iteracionnom metode Kachmazha i ego obobwenijah [On Iterational Kachmazh Method and its Generalizations]. Sib. Zh. Ind. Mat. [J. Appl. Industr. Math.], 2006. V. 9, No 3. P. 39–49.

Поступила в редакцию 14 июня 2012 г.

УДК 004.45

ТЕХНОЛОГИЯ ФРАГМЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ[1]

В.Э. Малышкин

Кратко представлена технология фрагментированного программирования и реализующие ее язык и система фрагментированного программирования LuNA, разрабатываемые в ИВМиМГ СО РАН. Технология ориентирована на поддержку разработки параллельных программ, реализующих большие численные модели, и их исполнения на суперкомпьютерах. Система LuNA автоматически обеспечивает такие динамические свойства параллельных программ как динамическая настройка на все доступные ресурсы, динамическая балансировка нагрузки, учет динамики поведения моделируемого явления и т.п.

Ключевые слова: фрагментированное программирование, численные алгоритмы, большие численные модели, параллельное программирование

Введение

В течение последних 15 лет в ИВМиМГ СО РАН велись работы по созданию методов и средств параллельной реализации больших численных моделей на суперкомпьютерах, а также параллельной реализации таких моделей. Накопленный опыт позволил сформировать идеи технологии фрагментированного программирования. Основная проблема параллельной реализации больших численных моделей состоит в том, что необходимая сложность программирования заметно превосходит квалификацию программистов, работающих обычно в численном моделировании, так как в программе моделирования понадобилось реализовывать динамические системные функции.

Чтобы преодолеть эту проблему, нужна поддержка процесса параллельного программирования таких моделей. Необходимая технология — технология фрагментированного программирования и реализующие ее язык и система программирования LuNA — разработаны в ИВМиМГ СО РАН.

Поддерживая технику разработки программ численного моделирования также хотелось бы обеспечивать:

- должную степень непроцедурности [1] представления алгоритма в параллельной программе (не знать MPI, свойства вычислителя и его коммуникационной сети, методы и средства параллельного программирования и т.д.),
- неизменность алгоритма, его независимость от оборудования (раздельное описание алгоритма и реализующей его программы), автоматическая генерация фрагментированной программы, что требуется для обеспечения ее переносимости,
- автоматическое включение реализации динамических свойств в прикладные параллельные программы, такие как динамическая балансировка нагрузки, настраиваемость на все доступные ресурсы вычислителя, динамическое распределение ресурсов, выполнение коммуникаций на фоне счета, учет поведения моделируемого явления.

[1]Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012»

Теоретическую базу проекта LuNA составляет теория синтеза параллельных программ на вычислительных моделях [2]. В проекте системы LuNA учтен опыт разработки как больших численных моделей [3, 4, 13, 14], так и различных систем сборочного программирования в мире [5–8]. Текущие технологические результаты представлены в настоящей статье. Теоретические аспекты проекта не рассматриваются. Более ранние результаты опубликованы в [9–11].

1. Идеология системы фрагментированного программирования LuNA

Перечисленные мотивы, и ряд других, после общего анализа трансформировались в следующие исходные проектные решения в системе фрагментированного программирования LuNA.

1.1. Основные проектные решения

1. Технология фрагментированного программирования поддерживает процесс сборки целой программы из фрагментов вычислений (модулей, процедур, их входных/выходных фрагментов данных и т.п.) и ее исполнение.
2. Каждый фрагмент вычислений – независимая единица программы (рис. 1), содержит описание входных/выходных переменных и кода (модуля, процедуры) фрагмента.

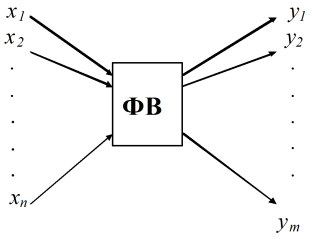


Рис. 1. Фрагмент вычислений

3. Фрагментированная программа – это рекурсивно перечислимое множество фрагментов вычислений и их входных/выходных переменных. Таким образом, фрагментированная программа определяется как множества переменных (фрагментов данных) и фрагментов вычислений. К фрагменту вычислений можно обращаться по-разному, например, как к обычной процедуре в последовательном языке программирования.
4. В отличие от технологии модульного программирования, в ходе исполнения фрагментированная структура программы сохраняется. Каждый фрагмент вычислений определит в ходе исполнения независимо исполняющийся процесс программы, взаимодействующий с другими процессами.
5. Следуя С. Клини [12], в качестве базового представления алгоритма взято рекурсивно перечислимое множество функциональных термов. Фрагментированный алгоритм при необходимости извлекается алгоритмом вывода из множества фрагментов вычислений.

1.2. Исполнение фрагментированной программы

Базовый алгоритм:

- Фрагмент вычислений исполняется, если все его входные переменные получили значения.
- После выполнения фрагмента вычислений получают значения его выходные переменные.
- Алгоритм может реализоваться либо управлением в сгенерированной программе, либо run-time системой. В системе LuNA для обеспечения необходимой степени асинхронности выбрано исполнение фрагментированной программы run-time системой.

Эти правила определяют асинхронное исполнение фрагментированной программы, в которой порядок исполнения фрагментов вычислений определяется лишь информационными зависимостями между ними. При исполнении фрагментированной программы run-time система ищет лучшие способы исполнения фрагментированного алгоритма.

2. Шаги разработки ФП

Фрагментированная программа разрабатывается в несколько последовательных этапов.

- Разработка исходного алгоритма решения задачи.
- Фрагментация исходного алгоритма. Фрагментация рассматривается здесь как универсальный способ распараллеливания алгоритмов, что позволяет поддержать его реализацию системой программирования (примеры фрагментации приведены в разделе 3). Фрагментация нередко является очень сложной работой. Например, фрагментация алгоритма прогонки [15] заняла 2 года и завершилась защитой кандидатской диссертации. На тестах получено ускорение исполнения алгоритма прогонки в 6000 раз на 8000 процессоров [15].
- Описание фрагментированной программы на языке LuNA. Входной язык системы LuNA устроен просто, в него в основном включены средства для определения множеств фрагментов данных и вычислений, синтаксис языка не очень интересен. Пример фрагментированной программы на языке LuNA можно видеть в разделе 3.1.
- Компиляция и анализ фрагментированного алгоритма. Генерация платформо-ориентированной фрагментированной программы.
- Исполнение фрагментированной программы.

3. Примеры фрагментированных алгоритмов

Несколько примеров поясняют технологию фрагментированного программирования и проблемы разработки системы LuNA.

3.1. Исходный алгоритм умножения квадратных матриц

На рис. 2 представлен исходный алгоритм умножения квадратных матриц. Вычисления проводятся по формулам:

$$c_{i,j} = \sum_{l=1}^{N} a_{i,l} \times b_{l,j},\ \ i,j = 1,2,\ldots,N.$$

Так как множество функциональных термов хранить нехорошо, то в языке LuNA вместо множества функциональных термов определяются множества фрагментов вычислений и фрагментов данных, а нужные термы конструируются из них по мере необходимости алгоритмом вывода.

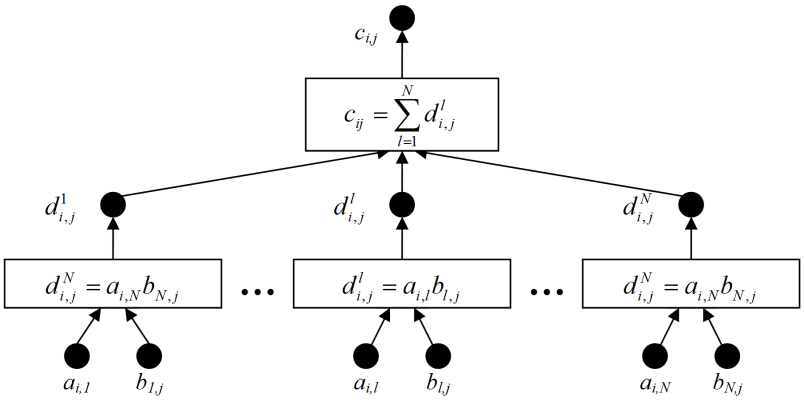


Рис. 2. Множество функциональных термов, представляющих исходный алгоритм

3.2. Фрагментированный алгоритм умножения квадратных матриц

В системе LuNA исходный алгоритм может быть запрограммирован как фрагментированный, т.е. каждая операция $c_{i,j} = \sum_{l=1}^{N} d_{i,j}^{l}$ объявлена фрагментом вычислений (и будет реализоваться как независимый процесс программы), переменные $a_{i,l}$, $b_{l,j}$, $c_{i,j}$, $d_{i,j}^{l}$ объявлены фрагментами данных. Но такая фрагментированная программа будет исполняться с большим замедлением, примерно с 1000 кратным, по сравнению с программой с использованием MPI из-за больших расходов на реализацию управления. Поэтому для численных алгоритмов, отличающихся высоко регулярностью, в процессе фрагментации проводится агрегация и переменных и операций. На рис. 3 представлена схема фрагментации алгоритма умножения матриц, которая показывает способ агрегации данных и вычислений, а на рис. 4 — фрагментированный алгоритм. Здесь:

$$C_{I,J} = \sum_{L=1}^{K} A_{I,L} \times B_{L,J},\ \ I,J = 1,2,\ldots,N.$$

Даже в таком простом примере, как алгоритм умножения матриц, информационные зависимости не определяют хорошего исполнения фрагментированной программы. Например, если для всех I и J выполнить сначала все, кроме K-го, фрагменты вычислений $D_{I,J}^{L} = A_{I,L} B_{L,J}$, а потом исполнить все K-ые фрагменты, то память вычислителя должна будет хранить все выработанные, но своевременно не потребленные,

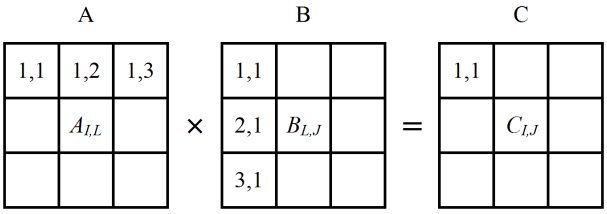


Рис. 3. Агрегация данных и фрагментов вычислений исходного алгоритма

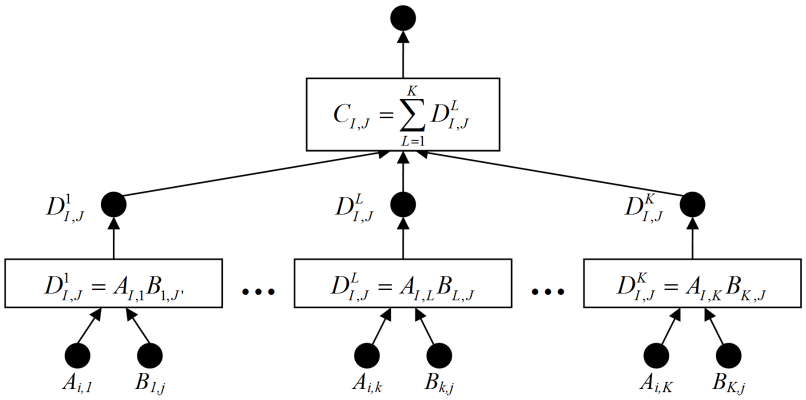


Рис. 4. Фрагментированный алгоритм

промежуточные данные. Объем хранимых промежуточных данных в K раз большие, чем может понадобиться при хорошей организации вычислений. В результате будет ограничен размер решаемой задачи и замедлится исполнение программы. Поэтому основными задачами, решаемыми run-time системой, являются распределение ресурсов и выбор наиболее подходящего очередного фрагмента вычислений на исполнение.

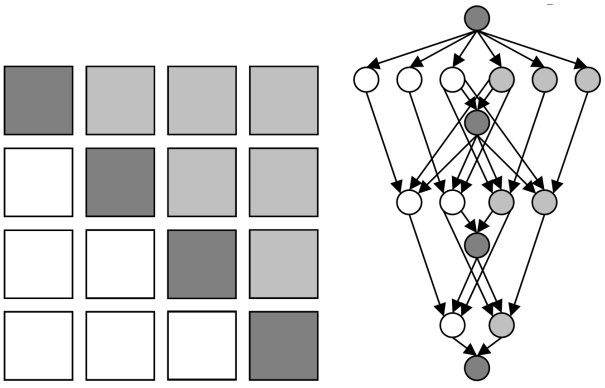
В качестве переменных фрагментированной программы используются агрегаты переменных исходного алгоритма, составляющих подматрицы исходной матрицы. Аналогично, код фрагмента вычислений является агрегатом кодов операций исходного алгоритма. В системе LuNA конструируются программы, в которых размер фрагментов данных является входным параметром. Ниже в качестве примера приведена часть фрагментированной программы умножения матриц, записанная в языке LuNA:

Множество фрагментов данных:

```
df a[i,k] := block(4*M*M) | i=0..K-1, k=0..K-1;
df b[k,j] := block(4*M*M) | k=0..K-1, j=0..K-1;
df c[i,j] := block(4*M*M) | i=0..K-1, j=0..K-1;
df d[i,j,k] := block(4*M*M) | i=0..K-1, j=0..K-1, k=0..K-1;
```

Множество фрагментов вычислений:

```
cf initc[i,j] := proc_zero<M,M> (out: c[i,j]) | i=0..K-1, j=0..K-1;
cf mul[i,j,k] := proc_mmul<M,M,M> (in: a[i,k],b[k,j]; out: d[i,j,k])
   | i=0..K-1, j=0..K-1, k=0..K-1;
cf sum[i,j,k] := proc_add<M,M> (in: d[i,j,k],c[i,j]; out: c[i,j])
   | i=0..K-1, j=0..K-1, k=0..K-1;
```



(a) (b)

Рис. 5. Фрагментация вычислений алгоритма LU-разложения (a) и информационные зависимости между фрагментами вычислений (b)

3.3. LU-разложение

LU-разложение преобразует квадратную матрицу A по формулам $l_{i,j} = a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}u_{k,j}$, $u_{i,j} = \frac{1}{l_{i,i}}\left(a_{i,j} - \sum_{k=1}^{i-1} l_{i,k}u_{k,j}\right)$ к виду $A = LU$, где:

$$L = \begin{pmatrix} l_{1,1} & 0 & 0 & \cdots \\ l_{2,1} & l_{2,2} & 0 & \\ l_{3,1} & l_{3,2} & l_{3,3} & \\ \cdots & & & \cdots \end{pmatrix}, \; U = \begin{pmatrix} 1 & u_{1,2} & u_{1,3} & \cdots \\ 0 & 1 & u_{2,3} & \\ 0 & 0 & 1 & \\ \cdots & & & \cdots \end{pmatrix}.$$

Матрица делится на фрагменты данных (рис. 5a), каждый фрагмент данных – подматрица матрицы A, для обработки каждого фрагмента данных формируется фрагмент вычислений. Фрагменты вычислений должны выполняться в следующем порядке (таковы информационные зависимости): вначале исполняется фрагмент $(1, 1)$, затем могут выполняться все фрагменты первого столбца и первой строки, затем может исполняться фрагмент $(2, 2)$ и т.д. Информационные зависимости между фрагментами вычислений показаны на рис. 5b.

Этот порядок нехорош тем, что создает неравномерную нагрузку на вычислитель: есть моменты времени, когда только один фрагмент вычислений готов к исполнению, и есть моменты времени, когда много более одного фрагмента готовы исполняться. Лучше было бы организовать исполнение фрагментов вычислений по гиперплоскостями (рис. 6), при котором формируется более равномерная нагрузка процессоров вычислителя.

3.4. Неравномерность загрузки процессоров в модели эволюции облака пыли

Параллельная реализация модели эволюции протопланетного диска описана в [13, 14]. На рис. 7 показана неравномерность распределения нагрузки на узлы мультикомпьютера в процессе моделирования. Каждый прямоугольник изображает фрагмент вычислений и потребляемые им ресурсы. Динамическая балансировка нагрузки узлов вычислителя планируется run-time системой и реализуется миграцией фраг-

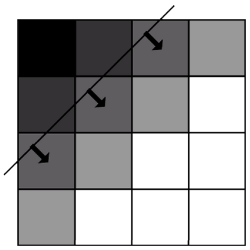


Рис. 6. Исполнение фрагментов вычислений гиперплоскостями

ментов вычислений с перегруженного узла на соседние, менее загруженные. На этом основана автоматическая реализация динамических свойств фрагментированной программы.

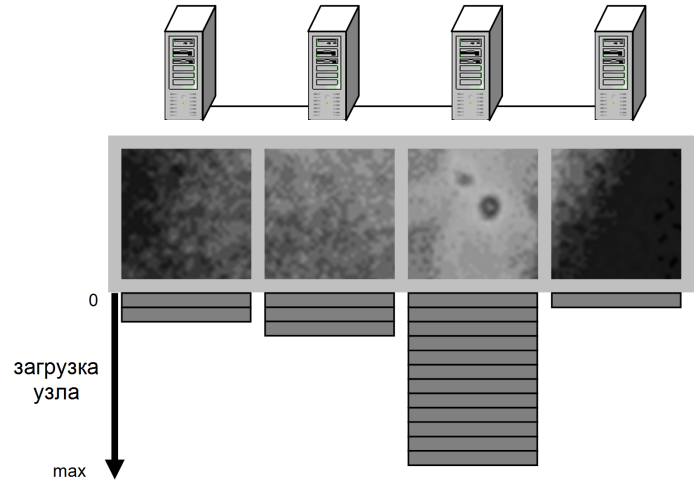


Рис. 7. Неравномерная нагрузка узлов вычислителя

Планирование, точный расчет балансировки нагрузки не делается, а моделируется физический процесс диффузии в жидких и/или газообразных средах. Средой здесь является множество фрагментов вычислений. Точный расчет балансировки нагрузки в условиях динамически меняющегося состояния системы взаимодействующих процессов фрагментированной программы не имеет смысла, и, как минимум, не технологичен.

Завершая обсуждение фрагментации численных алгоритмов, необходимо указать на одну важную качественную характеристику исполнения фрагментированной программы (рис. 8).

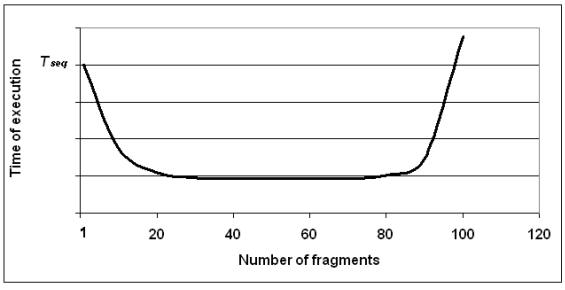


Рис. 8. Качественный график времени исполнения фрагментированной программы

Фрагментированная программа исполнялась в одном узле, сравнение производилось с последовательной программой. На рисунке T_{seq} – это время исполнения фрагментированной программы, состоящей из одного фрагмента, т.е. это время исполнения последовательной программы. Затем число фрагментов данных увеличивалось (соответственно, размер фрагментов данных уменьшался), при этом наблюдалось уменьшение времени исполнения программы. Минимальное время исполнения программы получалось, когда фрагмент вычислений со всеми обрабатывае-мыми им фрагментами данных попадал целиком в кэш-память. Увеличение времени исполнения программы начиналось с ростом числа фрагментов данных и вычислений, что приводило к увеличению накладных расходов на реализацию управления и динамического распределения ресурсов.

4. Язык и система фрагментированного программирования LuNA

Входной язык LuNA – теоретико-множественный, единственного присваивания и единственного исполнения фрагментов вычислений. Фрагменты данных и вычислений задаются рекурсивно перечислимыми множествами с использованием индексных выражений. Управление в LuNA-программе задается отношением частичного порядка на множестве фрагментов вычислений.

Отношение соседства на множестве фрагментов данных определяет, какие фрагменты данных должны размещаться в одном либо в соседних узлах. Дополнительно имеются операторы-рекомендации по распределению ресурсов вычислителя, по определению требуемого порядка исполнения фрагментов вычислений. Средства задания приоритетов исполнения фрагментов вычислений используются run-time системой для выбора наиболее подходящего фрагмента на исполнение.

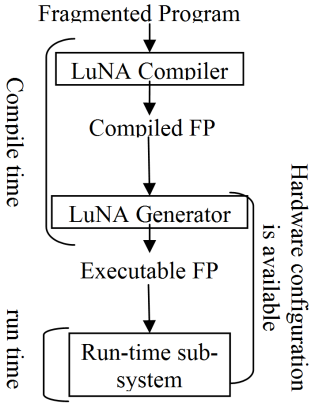


Рис. 9. Компоненты системы LuNA

Функциональная структура системы фрагментированного программирования LuNA представлена на рис. 9. Язык, некоторые теоретические и технологические аспекты системы программирования LuNA описаны в ряде публикаций [10, 11].

LuNA имеет три уровня преобразования фрагментированного алгоритма в программу: Компиляция, Генерация и Исполнение:

- Компилятор принимает решения (и вносит их в формируемую программу), которые можно принять, используя только информацию о свойствах алгоритма.

- Генератор принимает решения, которые зависят от свойств конкретного вычислителя (количество и типы доступных ресурсов, производительность ресурсов, нумерация узлов и т.п.), на котором программа должна исполняться.
- Run-time система принимает те решения, которые могут быть сделаны только динамически, в ходе вычислений. В их числе выбор фрагмента вычислений на исполнение, динамическая балансировка нагрузки узлов вычислителя, динамическое распределение ресурсов, включая назначение процессора для исполнения фрагмента вычислений и многое другое.

В системе LuNA есть еще один компонент — профилировщик, который собирает реальную информацию о ходе исполнения фрагментированной программы. Эта информация используется затем для улучшения последующих исполнений программы.

5. Родственные работы

Общее представление о работах, посвященных сборке программ из готовых фрагментов, дают проекты [5–8]. Проект Charm [5] за долгое время развития стал хорошо известной системой программирования. Ее основным недостатком является отсутствие глобальной оптимизации исполнения и не соответствующие входной язык и технология программирования, не позволяющие использовать в полной мере достоинства системы программирования, что свойственно и другим проектам.

Работа выполнена при частичной поддержке РФФИ, грант 10-07-00454-а.

Литература

1. Вальковский, В.А. К уточнению понятия непроцедурности языков программирования / В.А. Вальковский, В.Э. Малышкин. // Кибернетика. – 1981. – №3. – С. 55.
2. Вальковский, В.А. Синтез параллельных программ и систем на вычислительных моделях / В.А. Вальковский, В.Э. Малышкин. – Новосибирск: Наука. Сиб. отдние, 1988.
3. Kraeva, M.A. Assembly Technology for Parallel Realization of Numerical Models on MIMD-Multicomputers / M.A. Kraeva, V.E. Malyshkin. // International Journal on Future Generation Computer Systems. – 2001. – V. 17, № 6. – P. 755–765.
4. Malyshkin, V.E. Assembling of Parallel Programs for Large Scale Numerical Modeling. In Handbook of Research on Scalable Computing Technologies, ed. Kuan-Ching Li, Ching-Hsien Hsu, Laurence Tianruo Yang, Jack Dongarra and Hans Zima / V.E. Malyshkin. – IGI Global, 2010. – P. 295–311.
5. Charm++. URL: http://charm.cs.uiuc.edu/papers
6. ProActive Parallel Suite. URL: http://proactive.inria.fr
7. S-Net home page. URL: http://www.snet-home.org
8. Berzins, M. DAG-Based Software Frameworks for PDEs / M. Berzins, Q. Meng, J. Schmidt, J.C. Sutherland // Lecture Notes in Computer Science. – 2012. – V. 7155. – P. 324–333.

9. Kireev, S. Fragmentation of numerical algorithms for parallel subroutines library / S. Kireev, V. Malyshkin // The Journal of Supercomputing. – 2011. – V. 57, № 2. – P. 161–171.

10. Kireev, S. The LuNA Library of Parallel Numerical Fragmented Subroutines / S. Kireev, V. Malyshkin, H. Fujita // Lecture Notes in Computer Science. – 2011. – V. 6873. – P. 290–301.

11. Malyshkin, V. Optimization Methods of Parallel Execution of Numerical Programs in the LuNA Fragmented Programming System / V. Malyshkin, V. Perepelkin // The Journal of Supercomputing. DOI: 10.1007/s11227-011-0649-6. – P. 1–14.

12. Клини, С. Введение в математику / С. Клини. – М.: Иностранная литература, 1957.

13. Киреев, С.Е. Параллельная реализация метода частиц в ячейках для моделирования задач гравитационной космодинамики / С.Е. Киреев // Автометрия. – 2006. – №3. – С. 32–39.

14. Kireev, S.E. A Parallel 3D Code for Simulation of Self-gravitating Gas-Dust Systems / S.E. Kireev // Lecture Notes in Computer Science. – 2009. – V. 5698. – P. 406–413.

15. Terekhov, A.V. Parallel Dichotomy Algorithm for solving tridiagonal system of linear equations with multiple right-hand sides / A.V. Terekhov // Parallel Computing. – 2010. – V. 36, № 8. – P. 423–438.

Виктор Эммануилович Малышкин, доктор технических наук, профессор, кафедра параллельных вычислений, Новосибирский национальный исследовательский государственный университет, кафедра параллельных вычислительных технологий, Новосибирский государственный технический университет, зав. лаб. синтеза параллельных программ, Институт вычислительной математики и математической геофизики СО РАН, malysh@ssd.sscc.ru.

FRAGMENTED PROGRAMMING TECHNOLOGY

V.E. Malyshkin, Novosibirsk State University (Novosibirsk, Russian Federation)

Shortly the technology of fragmented programming is presented. This technology is now under development in the Institute of Computational Mathematics and Mathematical Geophysics. Also the LuNA language and system of fragmented programming are presented. The technology is oriented to support the parallel implementation of the large scale numerical models in physics. The LuNA system provides automatically such dynamic properties of parallel programs as dynamic tuning of the program to all the available resources of a supercomputer, dynamic balancing of a workload, the computation organization to the new details of the model behavior.

Keywords: fragmented programming, numerical algorithms, large scale numerical models, parallel programming.

References

1. Valkovsky V.A., Malyshkin V.E. K utochneniju ponjatija neprocedurnosti jazykov programmirovanija [Clarifying the Term of Non-Procedural Languages]. Kibernetika [Cybernetics], 1981. No 3. P. 55.

2. Valkovsky V.A., Malyshkin V.E. Sintez parallel'nyh programm i sistem na vychislitel'nyh modeljah [Synthesis of Parallel Programs and Systems on the Basis of Computational Models]. Nauka, Novosibirsk, 1988.
3. Kraeva M.A., Malyshkin V.E. Assembly Technology for Parallel Realization of Numerical Models on MIMD-Multicomputers. International Journal on Future Generation Computer Systems. 2001. V. 17, No 6. P. 755–765.
4. Malyshkin V.E. Assembling of Parallel Programs for Large Scale Numerical Modeling. In Handbook of Research on Scalable Computing Technologies, ed. Kuan-Ching Li, Ching-Hsien Hsu, Laurence Tianruo Yang, Jack Dongarra and Hans Zima. IGI Global, 2010. P. 295–311.
5. Charm++. URL: http://charm.cs.uiuc.edu/papers
6. ProActive Parallel Suite. URL: http://proactive.inria.fr
7. S-Net home page. URL: http://www.snet-home.org
8. Berzins M., Meng Q., Schmidt J., Sutherland J.C. DAG-Based Software Frameworks for PDEs. Lecture Notes in Computer Science. 2012. V. 7155. P. 324–333.
9. Kireev S., Malyshkin V. Fragmentation of Numerical Algorithms for parallel subroutines library. The Journal of Supercomputing. 2011. V. 57, No 2. P. 161–171.
10. Kireev S., Malyshkin V., Fujita H. The LuNA Library of Parallel Numerical Fragmented Subroutines. Lecture Notes in Computer Science. 2011. V. 6873. P. 290–301.
11. Malyshkin V., Perepelkin V. Optimization Methods of Parallel Execution of Numerical Programs in the LuNA Fragmented Programming System. The Journal of Supercomputing. DOI: 10.1007/s11227-011-0649-6. P. 1–14.
12. Kleene S.C. Introduction to Mathematics New York, D. Van Nostrand Company, Inc., 1952.
13. Kireev S.E. Parallel Implementation of the Particle-in-Cell Method for Gravitational Cosmodynamics Problem Modeling. Avtometriya, 2006. No 3. P. 32–39.
14. Kireev S.E. A Parallel 3D Code for Simulation of Self-gravitating Gas-Dust Systems. Lecture Notes in Computer Science. 2009. V. 5698. P. 406–413.
15. Terekhov A.V. Parallel Dichotomy Algorithm for solving tridiagonal system of linear equations with multiple right-hand sides. Parallel Computing. 2010. V. 36, № 8. P. 423–438.

Поступила в редакцию 26 марта 2012 г.

УДК 004.42, 517.97

ПАРАЛЛЕЛЬНОЕ ВЫЧИСЛЕНИЕ ОЦЕНКИ ПРИБЛИЖЕННО ОПТИМАЛЬНЫХ УПРАВЛЕНИЙ[1]

О.В. Фесько

Предложен метод расчета априорной оценки на основе достаточных условий оптимальности Кротова, позволяющей судить о качестве приближенного решения, полученного в ходе работы программы улучшения управления для задач оптимизации динамических систем. Метод реализован в виде параллельного алгоритма, являющегося частью программного комплекса оптимизации динамических систем на множествах кусочно-постоянных и кусочно-линейных управлений. Представленная процедура, кроме того, используется на этапе поиска начального управления при решении задач оптимального управления. Применение алгоритма и анализ эффективности его распараллеливания в рамках системы параллельного программирования с открытой архитектурой OpenTS демонстрируется в вычислительных экспериментах на примерах решения задач об оптимизации бифункциональной каталитической смеси и оптимального производства белка в биореакторе.

Ключевые слова: оптимальное управление, достаточные условия оптимальности Кротова, оценка управления, параллельный алгоритм.

Введение

При использовании различных численных методов решения задач оптимального управления крайне важное значение представляет собой возможность получения количественной оценки найденного приближенного управления, которая позволяет судить о точности, близости к точному оптимуму. Такую возможность открывают достаточные условия оптимальности Кротова [1].

В работе [2] рассматривалась задача оптимального управления вида

$$\begin{gathered}\dot{x}(t) = f(t, x(t), u(t)), \quad x(t_I) = x_I, \quad t \in [t_I, t_F], \\ u(t) \in D_u = \{u(t) |\, \underline{u} \le u(t) \le \overline{u}\}, \quad F(x(t_F)) \to \min,\end{gathered} \tag{1}$$

где $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, $x_i(t), i = \overline{1,n}$ – кусочно-гладкие. Управление $u = (u_1, \ldots, u_p)^T \in \mathbb{R}^p$ принадлежит одному из двух классов: кусочно-линейному

$$u(t) = \frac{((w^{2i+1} - w^{2i})\, t - (\tau_i w^{2i+1} - \tau_{i+1} w^{2i}))}{(\tau_{i+1} - \tau_i)}, \quad t \in [\tau_i, \tau_{i+1}], \tag{2}$$

или кусочно-постоянному $u(t) = w^i,\ t \in [\tau_i, \tau_{i+1}],\ i = \overline{0,m}$, где m — число моментов переключений при разбиении $t_I = \tau_0 < \tau_1 < \tau_2 < \ldots < \tau_{m+1} = t_F$. На параметры управления наложены ограничения типа $w^i \in W = \{w^i |\, \underline{u} \le w^i \le \overline{u}\}$.

Задача (1) была сведена к задаче условной конечномерной минимизации функции многих переменных $G(w, \tau)$ [3, 4]. Процесс решения полученной задачи состоит в поочередном применении численных алгоритмов: метода Рунге–Кутта для решения задачи Коши и комбинации метода Ньютона с модифицированным методом градиентного спуска для минимизации многоэкстремальной функции $G(w, \tau)$. Требуется количественно оценить полученное при использовании данного подхода приближенное управление.

[1]Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012».

1. Оценки приближенно оптимального решения

В силу рассматриваемых классов управлений непрерывную задачу (1) сведем к дискретной задаче оптимального управления. Введем множество индексов $K = \{0, 1, \ldots, m+1\}$ для моментов переключений $t_I = \tau_0 < \tau_1 < \tau_2 \ldots < \tau_{m+1} = t_F$, где m — число моментов переключений. В случае *кусочно-постоянного* управления $u(t) = w(k),\ t \in [\tau_k, \tau_{k+1}],\ k \in K$, эквивалентная непрерывной задаче (1) дискретная задача примет вид

$$\begin{aligned} &z(k+1) = \mu(k, z(k), w(k)), \quad k \in \{0, 1, \ldots, m+1\}, \\ &z(0) = z_0 = x_I, \quad w(k) \in W = \{w(k) |\ \underline{u} \le w(k) \le \overline{u}\}, \\ &F(z(m+1)) \to \min, \end{aligned} \tag{3}$$

где функция $\mu(k, z(k), w(k))$ — решение задачи Коши

$$\dot{x}(\xi) = f(\xi, x(\xi), w(k)), \quad \xi \in [\tau_k, \tau_{k+1}], \quad x(\tau_k) = z(k),$$

взятое в точке $\xi = \tau_{k+1}$.

Для случая *кусочно-линейного* управления $u(t) = w^1(k) + w^2(k)t,\ t \in [\tau_k, \tau_{k+1}]$, эквивалентная дискретная задача будет выглядеть как

$$\begin{aligned} &z(k+1) = \eta\left(k, z(k), w^1(k), w^2(k)\right), \quad k \in \{0, 1, \ldots, m+1\}, \\ &z(0) = z_0 = x_I, \quad w(k) \in W, \quad F(z(m+1)) \to \min, \end{aligned} \tag{4}$$

где функция $\eta(k, z(k), w^1(k), w^2(k))$ — решение задачи Коши

$$\dot{x}(\xi) = f\left(\xi, x(\xi), w^1(k) + w^2(k)\xi\right), \quad \xi \in [\tau_k, \tau_{k+1}], \quad x(\tau_k) = z(k),$$

взятое в точке $\xi = \tau_{k+1}$.

Пусть $(\tilde{z}(k), \tilde{w}(k))$ — допустимое решение задачи (3). Любой функции $\varphi(k, z)$ согласно [5, 6] соответствует нижняя граница минимизируемого функционала $F(x(t_F))$ и оценка $\Delta(z, w, \varphi) \ge 0$ его близости к оптимуму. Запишем оценочную функцию Кротова для задачи (3):

$$\begin{aligned} \Delta(\tilde{z}, \tilde{w}, \varphi) = F(\tilde{z}(m+1)) - \min_{z \in \mathbb{R}^n} (F(z) + \varphi(m+1, z)) + \\ + \sum_{k=1}^{m} \max_{\substack{w \in W, \\ z \in \mathbb{R}^n}} (\varphi(k+1, \mu(k, z, w)) - \varphi(k, z)) + \max_{w \in W} \varphi(1, \mu(0, z_0, w)). \end{aligned} \tag{5}$$

Оценочная функция Кротова для задачи (4) имеет вид:

$$\begin{aligned} \Delta\left(\tilde{z}, \tilde{w}^1, \tilde{w}^2, \varphi\right) = F(\tilde{z}(m+1)) - \min_{z \in \mathbb{R}^n} (F(z) + \varphi(m+1, z)) + \\ + \sum_{k=1}^{m} \max_{\substack{w^1, w^2 \in W, \\ z \in \mathbb{R}^n}} \left(\varphi\left(k+1, \eta\left(k, z, w^1, w^2\right)\right) - \varphi(k, z)\right) + \\ + \max_{w^1, w^2 \in W} \varphi\left(1, \eta\left(0, z_0, w^1, w^2\right)\right), \end{aligned} \tag{6}$$

где $(\tilde{z}(k), \tilde{w}^1(k), \tilde{w}^2(k))$ — допустимое решение задачи.

Если функция $\varphi(k,z)$ удовлетворяет условиям (схема Беллмана [1])

$$\begin{gathered}\varphi\left(k,z\right)=\max_{w^1,w^2\in W}\varphi\left(k+1,\eta\left(k,z,w^1,w^2\right)\right),\\ \varphi\left(m+1,z\right)=-F(z),\quad k\in\{0,1,\ldots,m+1\},\end{gathered}\tag{7}$$

то формулы оценок принимают более простой вид:

$$\Delta\left(\tilde{z},\tilde{w},\varphi\right)=F\left(\tilde{z}\left(m+1\right)\right)+\varphi\left(0,z_0\right),\quad \Delta\left(\tilde{z},\tilde{w}^1,\tilde{w}^2,\varphi\right)=F\left(\tilde{z}\left(m+1\right)\right)+\varphi\left(0,z_0\right)\tag{8}$$

вместо формул (5) и (6) соответственно. Управление, на котором достигается максимум в равенствах (7), может быть выбрано в качестве оптимального решения задач (3) и (4).

2. Описание алгоритма

Параллельный алгоритм поиска начального приближения для решения задачи оптимального управления (1) и вычисления количественной оценки приближенно оптимального управления представлен на рис. 1 и состоит из нескольких блоков.

На начальном этапе формулируется задача оптимального управления: задаются ее параметры, правые части системы дифференциальных уравнений $f(t,x(t),u(t))$ и терминальный функционал качества $F(x(t_F))$ в виде синтаксически правильных выражений языка `C++`. Далее исполняются следующие блоки:

Блок A: Вычисление начального приближения для ЗОУ.

Шаг 1. Задание параметров: начального состояния системы $z_0 = x_I$; ограничений на управление $\underline{u},\overline{u}$; шага $\tilde{h}$ для построения расчетной сетки по управлению, моментов переключений τ.

Шаг 2. Построение сетки по управлению $\left\{\underline{u},\underline{u}+\tilde{h},\ldots,\overline{u}-\tilde{h},\overline{u}\right\}$.

Шаг 3. Рекурсивно разрешается цепочка относительно функции Кротова $\varphi(k,z)$ на построенной сетке по управлению.

Шаг 4. Вычисление траекторий $\tilde{z}$, соответствующих найденному оптимальному управлению $\tilde{u}$.

Блок B: Улучшение управления [4].

Шаг 1. Задание начального состояния системы x_I; управления $\tilde{u}$ (найденного в *блоке* ***A***), области поиска, моментов переключений управления τ.

Шаг 2. При данном управлении численно интегрируется система обыкновенных дифференциальных уравнений (метод Рунге–Кутта 4-го порядка или метод Рунге–Кутта–Фельберга с адаптивным шагом).

Шаг 3. Вычисляется значение целевого функционала $G(\tilde{u},\tau)$.

Шаг 4. Вызывается функция поиска очередного приближения к решению.

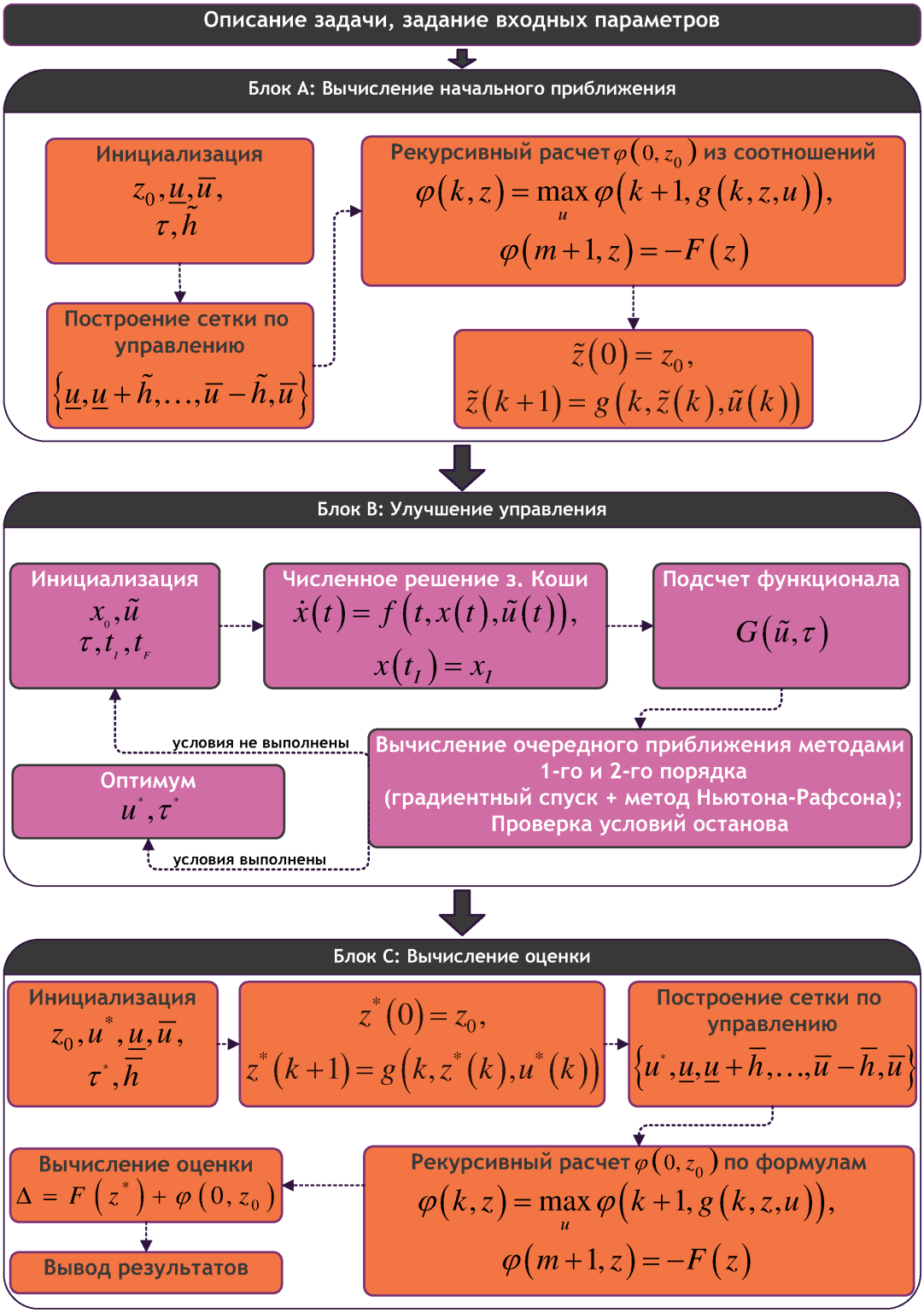


Рис. 1. Вычислительная схема решения поставленной задачи

Шаг 5. Проверка условий останова. Если условия останова не выполняется, переходим к шагу 2, иначе *блок* ***B*** завершает свою работу.

Блок C: Вычисление оценки приближенно оптимального решения.

Шаг 1. Задание параметров: начального состояния системы $z_0 = x_I$; допустимого управления u^* (найденного в *блоке* ***B***), которое требуется оценить; ограничений на управление $\underline{u}, \overline{u}$; шага $\overline{h} \le \tilde{h}$ для построения расчетной сетки по управлению, моментов переключений τ.

Шаг 2. Вычисление траекторий z^* и функционала качества $F(z^*)$, соответствующих заданному допустимому управлению.

Шаг 3. Построение сетки по управлению $\left\{u^*, \underline{u}, \underline{u} + \overline{h}, \ldots, \overline{u} - \overline{h}, \overline{u}\right\}$.

Шаг 4. Рекурсивно разрешается цепочка относительно функции Кротова $\varphi(k, z)$ на построенной сетке по управлению.

Шаг 5. Вычисление оценки согласно формулам (8).

Шаг 6. Вывод результатов в виде текстовых файлов и графиков. Завершение работы программы.

Шаги 3 и 4 рекурсивного вычисления функции Кротова $\varphi(k, z)$ блоков ***A*** и ***C*** соответственно могут выполняться независимо для различных наборов управлений, сгенерированных при построении сетки, за счет чего алгоритм допускает параллельное исполнение.

Распараллеливание вычислительного процесса осуществляется по схеме процессорной фермы. Главный процессор считывает входные данные и формирует сетку по управлению, после чего распределяет наборы управлений между процессорами-подчиненными, на которых производится рекурсивный расчет функции Кротова; после этого главный процессор собирает результаты.

Параллельный алгоритм для решения задачи оптимального управления (1), в основе которого лежит геометрическая декомпозиция расчетной области, описан в работах [3, 4].

3. Вычислительные эксперименты

Программа решения задачи оптимального управления была реализована на языке `C++`, а в параллельной версии исполнена в среде `OpenTS` на языке `T++` [7]. Преимущество данного подхода заключается в том, что данная система позволяет в динамике выполнять распараллеливание кусков кода программы, планировку вычислений, распределение данных по узлам и пр. без участия пользователя.

Расчеты проводились на высокопроизводительном вычислительном кластере «`BLADE`» Института Программных Систем РАН, оснащенном 8 вычислительными узлами с двумя процессорами Intel Xeon E5472 (4 ядра по 3,0 Ггц) и 16 ГБ оперативной памяти на каждом узле. Используя предложенный выше подход, решим следующие задачи.

3.1. Задача об оптимизации бифункциональной каталитической смеси

Рассматривается трубчатый реактор, в котором протекает процесс получения бензола из метилциклопентана. Процедура состоит из смешения двух монофункциональных каталитических компонентов, приводящих к реакциям гидрирования и изомеризации. Роль управления u играет массовая доля гидрирующего катализатора (отношение массы компонента гидрогенизации к общей массе катализатора). Характерное время t определяется отношением массы катализатора в конкретном разделе реактора к входному молярному расходу метилциклопентана. На выходе из реактора, финальный момент времени t_F определяется из отношения общей массы катализатора в реакторе к молярному расходу метилциклопентана в реакторе.

Задача оптимизации состоит в отыскании оптимальной каталитической смеси по всей длине реактора с целью максимизации концентрации бензола. То есть необходимо максимизировать функционал

$$F\left(x(t_F)\right) = x_7(t_F) \to \max,$$

где $t_F = 2000$ г·ч·моль$^{-1}$.

Имеющие место химические реакции описываются системой дифференциальных уравнений

$$\begin{aligned}
&\dot{x}_1(t) = -k_1x_1, \quad \dot{x}_2(t) = k_1x_1 - (k_2+k_3)x_2 + k_4x_5,\\
&\dot{x}_3(t) = k_2x_2, \quad \dot{x}_4(t) = -k_6x_4 + k_5x_5,\\
&\dot{x}_5(t) = k_3x_2 + k_6x_4 - (k_4+k_5+k_8+k_9)x_5 + k_7x_6 + k_{10}x_7,\\
&\dot{x}_6(t) = k_8x_5 - k_7x_6, \quad \dot{x}_7(t) = k_9x_5 - k_{10}x_7,
\end{aligned} \tag{9}$$

где константы скорости протекания реакций выражаются кубическими функциями от управления u каталитической смеси $k_i = c_{i0} + c_{i1}u + c_{i2}u^2 + c_{i3}u^3,\ i = \overline{1,10}$. Коэффициенты $c_{ij},\ j = \overline{0,3}$ представлены в табл. 1.

Таблица 1

Коэффициенты для констант k_i, определяющих скорость реакций

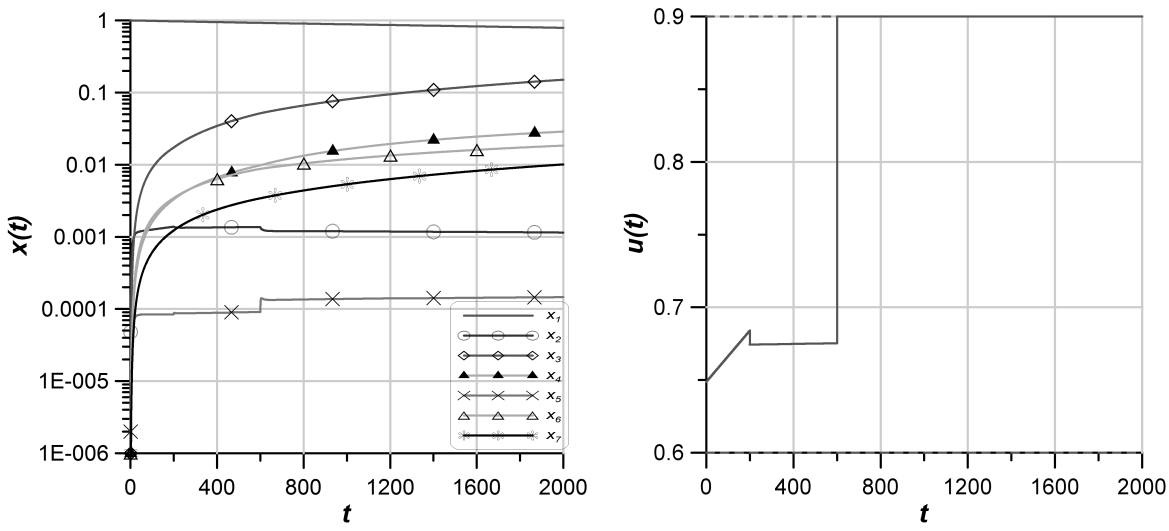
i	c_{i0}	c_{i1}	c_{i2}	c_{i3}
1	0,2918487e−002	−0,8045787e−002	0,6749947e−002	−0,1416647e−002
2	0,9509977e+001	−0,3500994e+002	0,4283329e+002	−0,1733333e+002
3	0,2682093e+002	−0,9556079e+002	0,1130398e+003	−0,4429997e+002
4	0,2087241e+003	−0,7198052e+003	0,8277466e+003	−0,3166655e+003
5	0,1350005e+001	−0,6850027e+001	0,1216671e+002	−0,6666689e+001
6	0,1921995e−001	−0,7945320e−001	0,1105666e+000	−0,5033333e−001
7	0,1323596e+000	−0,4696255e+000	0,5539323e+000	−0,2166664e+000
8	0,7339981e+001	−0,2527328e+002	0,2993329e+002	−0,1199999e+002
9	−0,3950534e+000	0,1679353e+001	−0,1777829e+001	0,4974987e+000
10	−0,2504665e−004	0,1005854e−001	−0,1986696e−001	0,9833470e−002

Переменные состояния представляют собой массовые доли химических соединений: при $i = \overline{1,6}$ для метилциклопентана, при $i = 7$ — бензола. Начальное состоя-

ние системы задано: $x(0) = (1, 0, 0, 0, 0, 0, 0)^T$. На управление наложено ограничение: $0,6 \le u \le 0,9$.

Задача имеет множество локальных максимумов [8]. Необходимо найти глобальный. Управление ищется в классе кусочно-линейных функций с двумя моментами переключений.

При поиске начального кусочно-линейного управления (***блок A*** схемы 1) с шагом по управлению $\tilde{h} = 0,1$ были найдены следующие параметры $(w^0; w^1; w^2; w^3; w^4; w^5) = (0,6; 0,7; 0,7; 0,7; 0,9; 0,9)$. Значение функционала составило $F = 0,0099729$. На этапе улучшения управления (***блок B***) найдено $(w^0; w^1; w^2; w^3; w^4; w^5) = (0,648; 0,683;\ 0,674; 0,675; 0,9; 0,9)$. При этом значение функционала $F = 0,0100956$ против $F = 0,0100942$ (см. [8]). При вычислении оценки приближенно оптимального управления (***блок C*** схемы) с шагом по управлению $\overline{h} = 0,05$ получено $\Delta = 0$. На рис. 2(a) и 2(b) изображены оптимальное кусочно-линейное управление и соответствующие ему траектории (на логарифмической шкале). В табл. 2 приведены результаты времени счета задачи оптимизации бифункциональной каталитической смеси на суперЭВМ.



(a) Траектории

(b) Управление

Рис. 2. Оптимальные процессы и управление

Таблица 2

Анализ эффективности параллельной версии программы

Число узлов, p	1	2	3	4	5	6	7
Время, t_p (с)	15473,371	7998,311	5555,059	4284,082	3469,615	3081,801	2609,322
Ускорение, t_1/t_p	1	1,93	2,78	3,61	4,45	5,02	5,93
Эфф-ть, $t_1/t_p/p$	1	0,97	0,92	0,9	0,89	0,84	0,85

3.2. Оптимальное производство белка в биореакторе

Рассмотрим модель по производству секретируемого белка в реакторе периодического действия с подпиткой, представленную в работе [9]. Эта модель использовалась некоторыми исследователями при разработке робастных методов решения задач такого характера. Трудности нахождения оптимального управления здесь частично связаны с низкой чувствительностью функционала к управлению. Биореактор описывается системой дифференциальных уравнений

$$\begin{aligned}
&\dot{x}_1(t) = g_1(x_2 - x_1) - \frac{u}{x_5}x_1, \quad \dot{x}_2(t) = g_2 x_3 - \frac{u}{x_5}x_2,\\
&\dot{x}_3(t) = g_3 x_3 - \frac{u}{x_5}x_3, \quad \dot{x}_4(t) = -7,3 g_3 x_3 + \frac{u}{x_5}(20 - x_4),\\
&\dot{x}_5(t) = u,
\end{aligned} \tag{10}$$

где $g_1 = \dfrac{4,75 g_3}{0,12 + g_3}$, $g_2 = \dfrac{x_4 e^{-5x_4}}{0,1 + x_4}$, $g_3 = \dfrac{21,87 x_4}{(x_4 + 0,4)(x_4 + 62,5)}$, с начальным состоянием $x(0) = (0, 0, 1, 5, 1)^T$. Здесь x_1 — концентрация секретируемого белка `SUC-s2` в культуре (л$^{-1}$), x_2 — общая концентрация белка `SUC-s2` в культуре (л$^{-1}$), x_3 — плотность клеток культуры (г·л$^{-1}$), x_4 — уровень глюкозы в культуре (г·л$^{-1}$), x_5 — объем культуры (л). Роль управления u играет скорость потока подпитки (л·ч$^{-1}$), $0 \le u \le 2$. Определение оптимальной скорости подпитки в реакторе для получения максимального количества желаемого продукта является очень сложной задачей оптимального управления.

Функционал составлен с целью максимизации количества секретируемого белка `SUC-s2`:

$$F\left(x(t_F)\right) = x_1(t_F)x_5(t_F) \to \max,$$

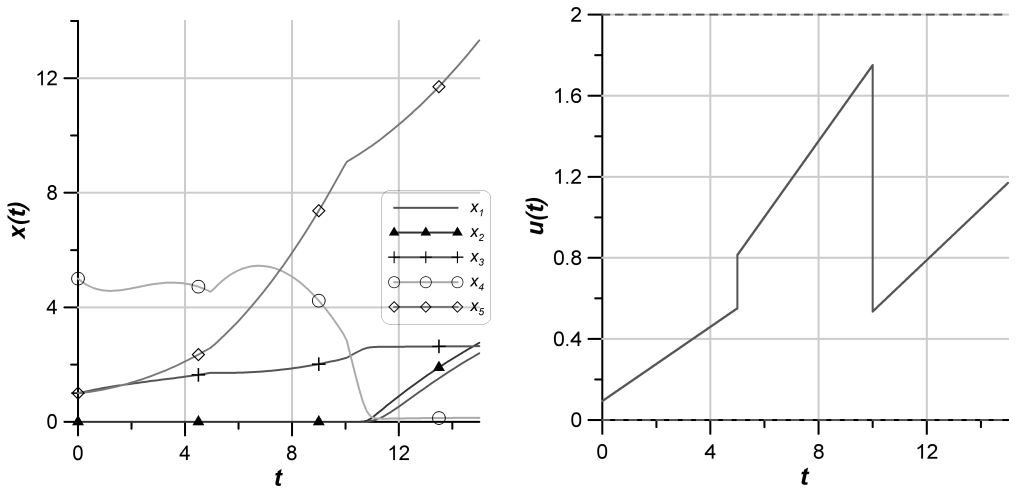
где $t_F = 15$ ч.

Управление ищется в виде (2) с двумя точками переключений. При поиске начального кусочно-линейного управления (***блок A***) с шагом по управлению $\tilde{h} = 0,25$ были найдены следующие параметры $(w^0; w^1; w^2; w^3; w^4; w^5) = (0; 0,75; 0,5; 2; 0,5; 1,25)$. Значение терминального функционала — $F = 31,62$. На этапе улучшения управления (***блок B***) найдено $(w^0; w^1; w^2; w^3; w^4; w^5) = (0,117; 0,526; 0,8; 1,74; 0,54; 1,14)$. При этом значение функционала составило $F = 31,82$ против $F = 32,67$ (см. [10], где функция управления искалась в кусочно-постоянном виде с 15 точками переключений). При вычислении оценки приближенно оптимального управления (***блок C***) с шагом по управлению $\overline{h} = 0,2$ получено $\Delta = 0$. На рис. 3(a) и 3(b) изображены оптимальное кусочно-линейное управление и соответствующие ему траектории. В табл. 3 приведены результаты времени счета оптимального производства белка в биореакторе на кластерной установке.

Таблица 3

Анализ эффективности параллельной версии программы

Число узлов, p	1	2	3	4	5	6	7
Время, t_p (с)	2281,76	1170,653	803,972	617,148	510,432	440,973	399,476
Ускорение, t_1/t_p	1	1,95	2,84	3,7	4,47	5,17	5,71
Эфф-ть, $t_1/t_p/p$	1	0,98	0,95	0,93	0,89	0,86	0,82



(a) Траектории (b) Управление

Рис. 3. Оптимальные процессы и управление

Заключение

В работе предложен параллельный алгоритм вычисления начального приближения для решения задач оптимального управления, а также расчета качества полученного решения в виде количественной оценки на основе достаточных условий оптимальности Кротова. Проведено исследование масштабируемости параллельной программы. Описанный алгоритм — неотъемлемая часть программного комплекса оптимизации динамических систем на множествах управлений простой структуры [3], в котором задействованы возможности современных суперЭВМ, что позволяет, в свою очередь, существенно сократить время вычислений.

Работа выполнена при поддержке РФФИ (грант 12-01-00256-а).

Литература

1. Кротов, В.Ф. Методы и задачи оптимального управления / В.Ф. Кротов, В.И. Гурман. – М.: Наука, 1973. – 216 с.

2. Фесько, О.В. Алгоритм поиска кусочно-линейного управления с нефиксированными моментами переключений / О.В. Фесько // Вестник Бурятского государственного университета, сер. Математика и информатика. – 2011. – №9. – С. 52–56.

3. Фесько, О.В. Программный комплекс поиска оптимальных управлений на мно-

жествах простой структуры / О.В. Фесько // Параллельные вычислительные технологии (ПаВТ2011): Труды международной научной конференции. – 2011. С. 712.

4. Фесько, О.В. Параллельный алгоритм оптимизации динамических систем на множестве кусочно-линейных управлений / О.В. Фесько // Вестник Бурятского государственного университета, сер. Математика и информатика. – 2010. – №9. – С. 79–87.
5. Гурман, В.И. Принцип расширения в задачах управления / В.И. Гурман. – М.:Наука-Физматлит, 1985.
6. Трушкова, Е.А. Оценка приближенно оптимальных решений на основе преобразований модели объекта / Е.А. Трушкова. // Вестник Бурятского государственного университета, сер. Математика и информатика. – 2011. – №9. – С. 47–51.
7. Moskovsky, A. Parallelism Granules Aggregation with the T-system / A. Moskovsky, V. Roganov, S. Abramov // 9th International Conference on Parallel Computing Technologies. LNCS 4671. – 2007. – P. 293–302.
8. Luus, R. Multiplicity of Solutions in the Optimization of a Bifunctional Catalyst Blend in a Tubular Reactor / R. Luus, J. Dittrich, F.J. Keil // Can. J. Chem. Eng. 70. – 1992. – P. 780–785.
9. Park, S. Optimal Production of Secreted Protein in Fed-batch Reactors / S. Park, W.F. Ramirez // AIChE J. 34. – 1988. – P. 1550–1558.
10. Luus, R. Iterative Dynamic Programming / R. Luus. – Boca Raton: Chapman and Hall/CRC, 2000.

Фесько Олесь Владимирович, аспирант, Исследовательский центр системного анализа ИПС им. А.К. Айламазяна РАН (г. Переславль-Залесский, Российская Федерация), fov@pereslavl.ru.

A PARALLEL APPROACH TO ESTIMATION OF THE APPROXIMATE OPTIMAL CONTROL

O.V. Fesko, Ailamazyan Program Systems Institute of the Russian Academy of Sciences (Pereslavl-Zalesskii, Russian Federation)

In this paper the method for computing a priori estimates of the approximate optimal control based on the Krotov sufficient conditions for optimality is considered. These estimates provide us with information about the quality of the approximate optimal solution obtained by applying the improvement control procedure. The method is implemented in the form of a parallel algorithm and may be used at the stage of finding out initial control. This algorithm is an essential part of the developed software package intended for optimization of controllable dynamical systems with piecewise constant and piecewise linear control. We also consider the scalability of the parallel algorithm in the OpenTS parallel programming system for bifunctional catalyst blend optimization problem and production of secreted protein in a fed-batch reactor problem.

Keywords: optimal control, Krotov's sufficient conditions of optimality, estimation of control, parallel algorithm.

References

1. Krotov V.F., Gurman V.I. Metody i zadachi optimal'nogo upravleniya [Methods and Problems of Optimal Control]. Moscow, Nauka Publ., 1973.
2. Fesko O.V. Algoritm poiska kusochno-lineynogo upravleniya s nefiksirovannymi momentami pereklyucheniy [Algorithm for Piecewise Linear Control Searching with Movable Switching Points]. Vestnik Buryatskogo gosudarstvennogo universiteta [Buryat State University Bulletin]. 2011. No 9, P. 52–56.
3. Fesko O.V. Programmnyy kompleks poiska optimal'nykh upravleniy na mnozhestvakh prostoy struktury [Software Package for Optimal Control Searching on the Simple Control Set]. Trudy mezhdunarodnoy nauchnoy konferencii Parallel'nye vychislitel'nye tekhnologii [Proceedings of the 2011 International Scientific Conference on Parallel Computational Technologies]. Moscow, 2011. P. 712.
4. Fesko O.V. Parallel'nyy algoritm optimizatsii dinamicheskikh sistem na mnozhestve kusochno-lineynykh upravleniy [Parallel Algorithm for Optimization Dynamical Systems on Piecewise Linear Control Set]. Vestnik Buryatskogo gosudarstvennogo universiteta [Buryat State University Bulletin]. 2010. No 9, P. 79–87.
5. Gurman V.I. Printsip rasshireniya v zadachakh upravleniya [The Extension Principle in Control Problems]. Moscow, Nauka Publ., 1997.
6. Trushkova E.A. Otsenka priblizhenno optimal'nykh resheniy na osnove preobrazovaniy modeli ob"ekta [The Estimate of Approximate Solutions on the Base of Model Transformation]. Vestnik Buryatskogo gosudarstvennogo universiteta [Buryat State University Bulletin]. 2011, no.9, pp. 47 – 51.
7. Moskovsky A., Roganov V., Abramov S. Parallelism Granules Aggregation with the T-system. 9th International Conference on Parallel Computing Technologies. LNCS, 2007. No 4671. P. 293–302.
8. Luus R., Dittrich J., Keil F.J. Multiplicity of Solutions in the Optimization of a Bifunctional Catalyst Blend in a Tubular Reactor. Can. J. Chem. Eng. 1992. No 70. P. 780–785.
9. Park S., Ramirez W.F. Optimal Production of Secreted Protein in Fed-batch Reactors. AIChE J. 1988. No 34. P. 1550–1558.
10. Luus R. Iterative Dynamic Programming. Boca Raton, Chapman and Hall/CRC, 2000.

Поступила в редакцию 3 марта 2012 г.

УДК 004.7-004.9

ПОСЛЕ EGI — WGI?[1]

В.П. Шириков

Статья посвящена краткому обзору истории и авторской оценке состояния реализации проектов сбора и распределенной обработки данных, основанной на использовании Грид-технологий. Особое внимание уделяется этапам реализации и областям их применений в рамках панЕвропейского проекта EGI (European Grid Initiative), а также перспектив его развития для возможной реализации проекта типа WGI (Wordlewide Grid Initiative).

Ключевые слова: обработка данных, грид-технологии.

По существу, данный обзор можно считать продолжением тех вводных, что были сделаны автором на наших конференциях в Абрау в 2004 и 2007 годах (см. [1, 2]). При этом частично используется материал, нашедший отображение в докладе [3] на юбилейной конференции по электронным библиотекам и коллекциям в 2008 году (см. [3]), а также в авторских обзорах в периодических изданиях Информационных бюллетеней ЛИТ ОИЯИ [4, 5].

Речь шла и идет о том, как и в какой степени реализуются и развиваются идеи основоположников GRID – тематики (К.Кессельмана, Я.Фостера), ставших ключевыми фигурами для объединений Globus Alliance и Globus Grid Forum, занявшихся организацией проработки и реализации систем типа Grid: их технической и программной основы , т.е. тех наборов программных средств (Globus Toolkits, GT), с помощью которых можно создавать эксплуатационные варианты систем. Исходной целью было сравняться по масштабу и общедоступности с реализацией « Всемирной информационной паутины World Wide Web», созданной на основе идей и программного задела Тима Бернерс-Ли почти 20 лет назад. К сожалению, несмотря на то, что указанные выше объединения начали разработку универсальных пакетов программой поддержки подобных структур более 10 лет назад – единой вычислительной структуры не получилось, а история как-то изложена в указанных выше обзорах. Получилось своеобразное «лоскутное одеяло» использования вычислительных ресурсов: в Европе свое, (с применением версий пакетов GT стали строить «локальные гридики» в рамках локальных сетей организаций или стран (как NorduGrid для северных стран), в Америке свое. Реализация проекта EGEE (Enabling Grids for E-sciencE), в рамках которого до 2010-го года работали в основном те, кто был связан с обработкой данных с ускорителя LHC (и не только), вынудила ответственных за программное обеспечение своих GRID-структур организовывать программные системные мосты для перехода к использованию EGEE (эта ситуация охарактеризована в обзоре [4]); возникла проблема обеспечения интероперабельности средств EGEE и Американского OSG (Open Science Grid)... Наконец, в рамках расширения возможностей EGEE и унификации его использования по крайней мере для Европейских стран была запущена реализация панЕвропейского проекта EGI (European Grid Initiative) как преемника EGEE. Целью было и укрепление общей компьютерной ресурсной базы (например, включением в состав совместно используемого странами-участницами оборудования суперкомпьютерных центров из 15 европейских стран) плюс унификация использования того программного системного обеспечения, которое необходимо для доступа

[1]Статья рекомендована к публикации программным комитетом международной научной конференции «Научный сервис в сети Интернет 2011»

и использования объединенного Европейского Грид. Как указывалось в обзоре [5], всеми организационными и финансовыми вопросами занялся Совет EGI Council, куда входят и представители от России и Белоруссии: в их ответственность входит и предоставить для общего использования: например, грид-инфраструктуру RDIG (Russian Data Intensive Grid) и оборудование федерации суперкомпьютерных центров «Скиф – полигон» (в которую вошли суперкомпьютерные центры ряда университетов и институтов России).

Ситуация с расширением рамок EGI за пределы Европы (скажем, объединением с Американскими Грид- структурами и не только, что позволило бы говорить о проекте WGI (Wordwide Grid Initiative)), не очевидная, хотя, казалось бы, общей системной программной основой начала работ по созданию всех грид- структур были упомянутые выше пакеты Globus Toolkits и их развитие. Так, в статье по адресу http://x-com.parallel.ru/about.html авторами из МГУ под руководством В.В.Воеводина отмечается: «Направление создания универсальных средств по созданию глобальных полигонов, объединяющих в рамках высокоскоростных сетей значительные распределенные ресурсы — интересное, однако реальные системы крайне тяжелы в установке, администрировании и сопровождении; организация расчетов на доступных компьютерах требует привилегированных административных полномочий, многие компьютерные платформы вообще не поддерживаются, тиражирование крайне затруднено. Примером работ в этом направлении является инфраструктура EGEE. . . ». Правда, в рамках проекта EGI усилия по преодолению указанных трудностей предпринимаются, но все же. Для ряда прикладных задач типа той, которая описана в статье «Grids for Experimental Science: The Virtual Control Room» (см. http://www.globus.org/alliance/publications/papers/clade_submitted_corrected.pdf), авторам вполне достаточно было взаимодействия с системой Access Grid, когда для контроля и интерпретации результатов в проведении экспериментов по термоядерному синтезу на установке Токамак требовалось оперативное привлечение вычислительного ресурса...

Отдельной проблемой можно считать и проблему создания информационных систем и коллекций, которые называют «Digital Libraries» (DL) и VDL («Virtual Digital Libraries»). Речь не идет в основном о библиотеках в традиционном смысле, к этому понятию относят цифровые коллекции разного типа – например, коллекцию фотографий или снимков событий в экспериментах, дополненную средствами поиска через Web интересующей фотографии (снимка) по определенным признакам. Для реализации таких средств должна быть предварительно проведена обработка каждого элемента коллекции, что может потребовать значительных вычислительных ресурсов. В своем авторском обзорном докладе на конференции RCDL‘2008 (Десятой Всероссийской конференции по тематике электронных библиотек и коллекций) я приводил пример реализации проекта DILIGENT (Digital Library Infrastructure on Grid Enabled Technology) и его предвидевшемся развитии в последующие годы в рамках проекта D4Science (сейчас он представлен на сайте по адресу http://www.d4science.eu). Одной из первых прикладных целей проекта DILIGENT было создание сервисов для проекта SAPIR (Search in Audio Visual Content Using Peer-to-Peer IR) как части проекта Chorus, т.е. для задачи создания в интересах этих проектов нового типа представления и поиска данных, отсутствовавших в традиционно используемых поисковых системах типа Google и Yandex. Указанным проектом DILIGENT авторов

из CNR-ISTI (Пиза, Италия) заинтересовались в ЦЕРН и помогли выделением компьютерных мощностей из ресурсов EGEE для создания и формализованного описания информационных объектов: с применением сервисов «gCube on top of gLite» (см. http://www.gcube-system.org), разработанных авторами проекта, был проведен на инфраструктуре EGEE 16-недельный прогон (data challenge) по обработке 37 млн. фотографий из on-line базы данных Flickr (известного модифицированного Web-приложения для поиска и обмена фотографиями), сгенерировано около 112 млн. текстовых и image-объектов...

Может быть, полезно еще раз вспомнить и старую статью 2002-го года «The Semantic Grid: a Future e-Science Infrastruture» (http://www.semanticgrid.org/documents/semgrid-journal/semgrid-journal.pdf), где авторы предсказывали, что программная среда компьютеризованной науки и все Grids должны будут включать в себя трехуровневую систему сервисов:

1) Data/Computation Services, средства размещения данных и их транспортировки между обрабатывающими программами, обеспечение вычислительных и сетевых ресурсов;

2) Information Services, средства представления, запоминания и доступа к информации, управления ею;

3) Knowledge Services, средства накопления, представления, обновления, «публикации» (сетевого распространения) знаний для помощи ученому в его исследовательском процессе.

Все положения демонстрировались детальным формализованным примером цикла полной автоматизации обработки экспериментальных данных в сетевой компьютерной среде (от начала поступления данных на анализ до подведения итогов результата обработки научным сообществом) с применением конкретного перечня сервисов каждого из указанных уровней; подчеркивалась роль семиуровневой системы онтологий (аппарата формализованного представления информации) для нормального функционирования всей клиент-сервисной структуры приведенного примера.

Когда-то, комментируя эту статью в обзорном докладе на конференции «Научный сервис в сети ИНТЕРНЕТ» в 2003 году (см. [7], я отмечал следующее (на основе ее авторских определений):

Разделение понятий «информация» и «знание» сделано просто: информация – это какие-то данные и их значения, определение, смысл («данное целое число относится к температуре во время реакции», «эта строка – имя человека»), а знание – это информация, побуждающая к действию («данное значение температуры критическое, необходима остановка реакции»). Соответственно “сервис” можно определить как программный процесс реализации какого-то действия из набора служебных и прикладных программ в какой-то научной предметной области или в междисциплинарных сферах: например, сервис автоматического уведомления ученых, заинтересованных в результатах проведенной другими сервисами обработки какого-то набора данных. Агенты в этой схеме – своеобразные “брокеры” на бирже (рынке) программных услуг-сервисов, программные инициаторы процессов : агент по своей инициативе или поручению от другого агента организует поиск нужного сервиса в каком-то репозитории, сверяет полномочия поручителя с указаниями в описании сервиса, запускает сервис в работу и предпринимает какие-то действия по концу его работы. Что касается упомянутой системы онтологий (документов или файлов с метадан-

ными, которые формально определяют классы, типы и свойства объектов, понятий, терминов, а также отношения между ними за счет использования описаний свойств классов и подклассов и логических правил вывода), то в упомянутой статье отмечается, что проблемы аннотирования контента (содержания коллекций информации разных типов) и сервисов определяют необходимость порождения аппаратом онтологий следующих типов метаданных:

- Domain ontologies: описания (концептуализация) важных объектов, их свойств и отношений между ними (согласованный набор аннотаций, понятий, определений в предметной области...);
- Task ontologies: описания задач и процессов, их свойств и отношений (например, набора характеристик фаз процесса химического анализа...);
- Quality ontologies: описание атрибутов знания (например, аннотации к тому, могут ли результаты, полученные какими-то средствами, быть более эффективно получены и расширены более совершенными средствами);
- Value ontologies: характеристика тех атрибутов, которые относятся к установлению значимости (важности) контента ("стоимость" полученных в эксперименте физических данных, например);
- Argumentation ontologies: широкий набор аннотаций, имеющих отношение к описанию причин – почему контент был накоплен (например, данные с какого-то эксперимента), почему он был использован тем или иным способом, кто его одобряет или не признает...

Понятно, что в реализации такой архитектуры накопления, обработки и использования ее результатов в значительной степени замешаны и понятие семантического Grid, и понятие семантического Web... В этом смысле интересен доклад Хорошевского В.Ф. из ВЦ РАН «Онтологические модели и Semantic Web: откуда и куда мы идем?» (http://ontology.ipi.ac.ru/files/f/f0/OM2008-khoroshevskiy.ppt#1).

Должен отметить, что многие работы по рассматриваемой в обзоре тематике рассматривались на четырех международных конференциях «Распределенные вычисления и грид-технологии в науке и образовании» в ЛИТ ОИЯИ: последняя прошла в 2010 году. Тезисы (ISBN 978-5-9530-0253-0) и полные тексты докладов (ISBN 978-5-9530-0269-1) опубликованы. Впрочем, скажем, полный текст работы «Mediation Based Semantic Grid» сотрудников из ИПИ РАН (соучастников реализации и развития международного проекта AstroGrid) на русском языке и сейчас доступен по адресу http://synthesis.ipi.ac.ru/synthesis/publications/10semgrid/10semgrid.pdf.

Наконец, в заключение можно продолжить разговор по модной теме, которой посвящался заключительный раздел редакторского обзора [6]: о совместном использовании грид-технологии и технологии «облачной обработки данных» (Cloud computing). Будет ли общий всемирный грид WGI или попрежнему будет многогридовская структура – от указанной темы не уйти. В этом смысле интересующимся можно рекомендовать материалы Европейского исследовательского консорциума по информатике и математике (ERCIM), подготовившего в октябре 2010 года специальный выпуск по этой теме (см. http://ercim-news.ercim.eu/en83), где в принципе со страницы по этому адресу можно организовать скачивание 64-х страниц общим объемом в 17 мегабайт (файл в pdf-формате).

Литература

1. Ширикова, В.П. Программное обеспечение Grid : переоценка ценностей / В.П. Ширикова // Научный сервис в сети Интернет: тр. Всерос. науч. конф. (20–25 сент. 2004 г., г. Новороссийск). – М., 2004. – С. 142–144.
2. Ширикова, В.П. Системное обеспечение «бесшовной» структуры и средств использования «Computational/Data Grid of Grids»для разных областей деятельности: достижения, нерешенные проблемы, виды на реализацию / В.П. Ширикова // Научный сервис в сети Интернет: тр. Всерос. науч. конф. (24–29 сент. 2007 г., г. Новороссийск). – М., 2007. – С. 10–13.
3. Ширикова, В.П. RCDL'1999 – RCDL'2008: DL, VDL, Semantic WEB/GRID... / В.П. Ширикова // Научный сервис в сети Интернет: решение больших задач: тр. 10 Всерос. науч. конф. (22–27 сент. 2008 г., г. Новороссийск). – М., 2008. – С. 24-27.
4. Ширикова, В.П. Программное обеспечение Grid: состояние и перспективы // http://lit.jinr.ru/Inf_Bul_3/bullet.htm#_Toc98590864 (дата обращения: 10.03.2012)
5. Ширикова, В.П. Обеспечение «бесшовной» структуры и средств использования «Computational /Data Grid of Grids»// http://lit.jinr.ru/Inf_Bul_4/bullet_6.htm#_Toc190687952 (дата обращения: 10.03.2012)
6. Ширикова, В.П. О новом проекте общеевропейской GRID-инфраструктуры // http://Inf_Bul_5/bullet_8.htm (дата обращения: 11.03.2012)
7. Ширикова, В.П. Как у нас с интеллектом в Web и Grid для создания полноценного научного сервиса? / В.П. Ширикова // Научный сервис в сети Интернет: тр. Всерос. науч. конф. – М., 2002. – С. 33–38.

Владислав Павлович Ширикова, доктор физико-математических наук, профессор, Лаборатория информационных технологий Объединенного института ядерных исследований, shirikov@jinr.ru

---

AFTER EGI — WGI?

V.P. Shirikov, Joint Institute for Nuclear Research (Dubna, Russian Federation)

This article concerns a short review of history and author's estimation of realization state in projects for data handling, based on use of Grid-technologies (in particular, in frames of European Grid Initiative project: EGI). Some problems are mentioned, which concern the possible WGI (Worldwide Grid Initiative project) realization.

Keywords: data handling, grid technologies.

References

1. Shirikov V.P. Programmnoe obespechenie Grid: pereocenka cennostej [Grid Software: Reappraisal]. Proceedings of the "Nauchnyj servis v seti Internet" (Novorossiysk, 2004, Sept. 20–25). P. 142–144.

2. Shirikov V.P. Sistemnoe obespechenie "besshovnoj" struktury i sredstv ispol'zovanija "Computational/Data Grid of Grids" dlja raznyh oblastej dejatel'nosti: dostizhenija, nereshennye problemy, vidy na realizaciju [System Support of "Seamless" Structure and "Computational/Data Grid of Grids" for Different Areas. Progress, Unresolved Issues and Prospects.]. Proceedings of the "Nauchnyj servis v seti Internet" (Novorossiysk, 2007, Sept. 24–29). P. 10–13.

3. Shirikov V.P. RCDL'1999 – RCDL'2008: DL, VDL, Semantic WEB/GRID... Proceedings of the "Nauchnyj servis v seti Internet" (Novorossiysk, 2008, Sept. 22–27). P. 24–27.

4. Shirikov V.P. Programmnoe obespechenie Grid: sostojanie i perspektivy [Grid Software: Current State and Prospects]. URL: http://lit.jinr.ru/Inf_Bul_3/bullet.htm#_Toc98590864 (дата обращения: 13.03.2012)

5. Shirikov V.P. Obespechenie "besshovnoj" struktury i sredstv ispol'zovanija "Computational/Data Grid of Grids" [System Support of "Seamless" Structure and "Computational/Data Grid of Grids"]. URL: http://lit.jinr.ru/Inf_Bul_4/bullet_6.htm#_Toc190687952 (дата обращения: 14.03.2012).

6. Shirikov V.P. O novom proekte obweevropejskoj GRID-infrastruktury [On a New Project of European GRID-infrastructure] http://Inf_Bul_5/bullet_8.htm (дата обращения: 10.03.2012).

7. Shirikov V.P. Kak u nas s intellektom v Web i Grid dlja sozdanija polnocennogo nauchnogo servisa? [Do We Have Good-enough AI for Full-fledged Scientific Service?] Proceedings of the "Nauchnyj servis v seti Internet" (Novorossiysk, 2002). P. 33–38.

Поступила в редакцию 11 ноября 2011 г.

УДК 65.01

О СТРАТЕГИЧЕСКОМ ПЛАНИРОВАНИИ РАЗВИТИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В КОРПОРАЦИИ

Ю.А. Зеленков

Существующие методы формирования корпоративной стратегии развития информационных технологий (ИТ) опираются на требования, которые должны быть сформированы в бизнес-стратегии компании, однако часто формализованная бизнес-стратегия отсутствует. В данной статье решается задача формирования паттерна стратегического поведения ИТ-подразделения крупной корпорации в условиях отсутствия формальной бизнес-стратегии. Рассмотрен общий процесс принятия решений в корпорации, предложена модель принятия стратегических решений о развитии ИТ. Предложенная модель позволяет определить уровень сложности инициатив по внедрению новых ИТ на основе их влияния на трансформационные и транзакционные затраты. Уровень сложности внедрения в свою очередь накладывает ограничения на использование различных элементов архитектуры предприятия. Разработан метод принятия стратегических решений базе указанной модели. Продемонстрировано использование предложенного метода на примере создания виртуальной среды проектирования машиностроительной корпорации, участвующей в качестве соисполнителя в создании нового продукта.

Ключевые слова: управление ИТ, ИТ-стратегия, информационный менеджмент.

Введение

В работе [8] отмечено, что компании с высоким уровнем инвестиций в информационные технологии (ИТ) получают больше, чем просто приобретение новых технологий. Они фактически инвестируют в новые формы существования корпорации, улучшение бизнес-процессов, более эффективное распространение информации, децентрализацию принятия решений, устранение неключевых продуктов и компетенций, повышение квалификации персонала. Таким образом, задача создания стратегического плана развития ИТ (далее ИТ-стратегия) является важнейшим инструментом построения эффективного предприятия.

1. Обзор методов формирования ИТ-стратегии

В работах зарубежных исследователей сформулированы несколько подходов к разработке ИТ-стратегии. Во-первых, это «выравнивание бизнеса и ИТ» (Business and IT Alignment), задача которого понимается либо как обеспечение ИТ-поддержки бизнеса, либо как реализация в бизнесе новых возможностей, которые предлагают современные ИТ. Одной из основополагающих теоретических работ по данному вопросу является статья [11], где предложена стратегическая модель соответствия. Эта модель предполагает, что установление соответствия ИТ и бизнеса может быть достигнуто за счет выравнивания четырех областей: бизнес-стратегия, ИТ-стратегия, организационная инфраструктура и процессы предприятия, ИТ-инфраструктура и процессы. «Направление» этого выравнивания характеризует стратегическую роль ИТ-подразделения, а также определяет методы стратегического планирования и критерии эффективности. Например, использование последних достижений ИТ для создания новых возможностей ведения бизнеса (движение от ИТ-стратегии через бизнес-стратегию к изменению организационной инфраструктуры и процессов) превращает

ИТ-менеджера в архитектора бизнеса, а ИТ-подразделение в полноправного поставщика продукции.

Основная критика этой модели сводится к тому, что она определяет «что» делать, но не указывает «как» это делать. Существует также проблема оценки качества выравнивания ИТ и бизнеса. Один из способов решения этой проблемы предложен в [14], где рассматривается шесть критериев соответствия ИТ и бизнеса (зрелость коммуникаций между ИТ и бизнесом, качество измерения эффективности ИТ, качество общего управления ИТ, уровень вовлеченности ИТ в решение бизнес-проблем, зрелость ИТ-архитектуры и уровень компетентности ИТ-персонала) и предложено описание пяти уровней зрелости для каждого из них.

Второй подход к разработке ИТ-стратегии базируется на архитектуре предприятия (Enterprise Architecture). Под архитектурой предприятия понимается строгое описание его структуры, ее декомпозиции на подсистемы, связей между подсистемами и с внешней средой, а также используемая терминология и руководящие принципы проектирования и развития предприятия [10]. Впервые понятие архитектуры предприятия было введено Дж. Захманом в 1987 г. К настоящему моменту данная область достаточно хорошо исследована, предложено несколько моделей описания архитектуры предприятия. Следует отметить, что все эти модели базируются на рассмотрении нескольких архитектурных доменов, как правило, это архитектура бизнес-процессов, данных, приложений и техническая архитектура. Согласно данному подходу, необходимо спроектировать целевую архитектуру предприятия, которая должна соответствовать целям и стратегии бизнеса. ИТ-стратегия в этом случае будет представлять набор действий по созданию целевой архитектуры. Наиболее последовательно архитектурный подход к созданию ИТ-стратегии сформулирован в [15]. Согласно этой работе предлагается выполнить три шага: 1. Разработать операционную модель, которая определяется видением того, как корпорация будет обеспечивать достижение стратегических целей, и зависит от степени интеграции и стандартизации бизнес-процессов. 2. Разработать архитектуру предприятия, поддерживающую операционную модель, 3. Повышать зрелость архитектуры предприятия (определены 4 уровня: бизнес-силос, стандартизация технологий, оптимизированное ядро и модульность бизнеса).

Третьим направлением в разработке ИТ-стратегии является разработка процедур управления ИТ (IT Governance). Это сфокусированная на ИТ часть корпоративного управления, которая определяется как «ответственность высшего руководства и заключается в обеспечении управления, организационных структур и процессов, гарантирующих, что информационные технологии поддерживают и дополняют стратегию организации и ее цели» [9]. Широко известным фреймворком IT Governance является CobiT [9], где применительно к корпоративным ИТ рассматриваются следующие области: определение направлений для внедрения новых решений и предоставления сервисов (PO), приобретение новых решений и их реализация в виде сервисов (AI), предоставление и поддержка сервисов (DS), мониторинг и оценка всех процессов (ME). Для каждой области выделены основные процессы, для процессов предложены метрики и модель оценки зрелости. Отметим, что в CobiT ИТ-подразделение рассматривается как часть архитектуры предприятия, которая создается в соответствии с ИТ-целями, которые, в свою очередь, выводятся из бизнес-целей и бизнес-стратегии, но конкретных рекомендаций по организации ИТ-службы не дается. Можно сказать,

что наряду со стандартом ISO/IEC 27002, определяющим требования к информационной безопасности, CobiT определяет «что» нужно делать для управления ИТ. Руководством по тому «как» это нужно делать служит стандарт ITIL V3, определяющий процедуры планирования, развертывания и поддержки ИТ-сервисов.

Среди других подходов к управлению ИТ следует упомянуть модель [17], разработчиками которой являются авторы книги [15]. В [17] ими предложена формальная комплексная оценка эффективности управления ИТ, а также рассмотрено влияние на принятие решений (инвестиции, архитектура, выбор приложений...) различных архетипов управления (монархия бизнеса или ИТ, дуополия и т.д.).

В отечественных исследованиях проблеме разработки ИТ-стратегии уделяется гораздо меньше внимания. Основной является работа [2], где предлагается метод стратегического планирования, базирующийся на архитектуре предприятия. Стратегия понимается как множество проектов, обеспечивающих последовательный переход к целевой архитектуре, сформированное при соответствующем наборе ресурсных ограничений. При этом планирование изменений в управлении ИТ не рассматривается. Для преодоления этих ограничений в работе [3] автором настоящей статьи был предложен метод стратегического планирования развития ИТ, который предполагает рассмотрение не только четырех традиционных доменов архитектуры предприятия (бизнес-процессы, данные, приложения и техническая архитектура), но и пятого домена — архитектуры процессов управления ИТ-сервисами. Фактически этот метод синтезирует подходы на базе архитектуры и управления. Также домен технической архитектуры был детализирован применительно к особенностям бизнеса машиностроительной корпорации. Все это позволило дополнительно к собственно методу стратегического планирования ИТ разработать и обосновать его организационно-техническое обеспечение (метод внедрения новых ИТ-сервисов, организационная структура ИТ-подразделения, процедуры финансового планирования).

2. Постановка задачи

Из приведенного выше краткого обзора следует, что, несмотря на различие подходов, все способы разработки ИТ-стратегии так или иначе должны учитывать следующие аспекты: согласование возможностей ИТ с целями бизнеса; эффективность (зрелость) развития ИТ на предприятии; архитектуру информационных систем и технологий и управление ИТ, включая модель процессов создания и поддержки ИТ-сервисов, организационную структуру ИТ-подразделения, развитие компетенции в ИТ.

При этом ключевой задачей является соответствие бизнес-стратегии. Однако, как отмечается в [4], в крупной корпорации крайне редко удается обнаружить априорные заявления (формализованную бизнес-стратегию), которым она действительно следует. Это объясняется тем, что стратегия имеет в основном дело не с неопределенными, а с неизвестными факторами. В то же время для непредвзятого наблюдателя наличие определенной стратегии, которая базируется на спонтанно возникающих паттернах достижения стратегических целей, очевидно. Для преодоления этого парадокса в [4] предложено следующее понимание процесса формирования стратегии. В центре внимания должен находиться процесс формирования паттернов стратегического поведения, которые изменяются вместе с новыми ситуациями. Большую часть времени

организация может быть описана как некая устойчивая конфигурация ее составных частей. Такие периоды стабильности время от времени прерываются трансформациями — квантовыми скачками в иную конфигурацию. Потребность в трансформации выявляется в процессе инкрементального самообучения организации.

О том, что наличие бизнес-стратегии не является обязательным условием формирования ИТ-стратегии косвенно свидетельствуют результаты исследования мнения ИТ-директоров зарубежных корпораций о влиянии различных факторов на успех стратегического планирования ИТ, приведенные в [12]. Большинство из них (54%) гораздо выше оценивает вовлечение и поддержку топ-менеджмента, т.е. фактически неформальные связи с руководством, чем наличие бизнес-стратегии (18%). Таким образом, ключевую проблему формирования ИТ-стратегии (как и любой другой функциональной стратегии) можно переформулировать следующим образом: необходим метод формирования стратегического поведения ИТ-подразделения, соответствующего паттернам стратегического поведения корпорации в целом. При этом формализованное описание стратегических паттернов корпорации или других ее подразделений отсутствует.

3. Модель корпорации

Для решения сформулированной выше задачи построим математическую модель корпорации. Согласно современным представлениям любая организация (в том числе и корпорация) может рассматриваться как открытая система, эволюционирующая вместе с внешней средой. Она является целенаправленной системой, входит как часть в одну или более целенаправленных систем верхнего уровня, ее части (люди) имею собственные цели (что означает недопустимость проведения аналогии с организмом) [1]. При этом границы между корпорацией и внешней средой становятся все более условными. Сама корпорация имеет многомерную организационную структуру, в которой на каждом уровне имеются структурные единицы трех разных видов: определяемые (а) их функцией (продукция этих единиц потребляется преимущественно внутри корпорации), (б) их продукцией (которая потребляется преимущественно на внешнем рынке) и (в) их пользователями (рынками, определяемыми типом или местонахождением покупателей) [1]. Очевидно, что в большинстве случаев ИТ-служба является функциональным подразделением первого вида. Поэтому задачу формирования ИТ-стратегии можно рассматривать как частный случай разработки функциональной стратегии.

На основании сказанного выше можно предложить следующую модель корпорации. В ее структуре выделим стратегический управляющий центр M, подразделения, ответственные за производство продуктов и услуг, потребляемых на внешнем рынке P_i, $i \in [1, \ldots, n]$, подразделения, ответственные за оказание услуг внутри корпорации S_j , $j \in [1, \ldots, q]$ и подразделения, отвечающие за работу на рынках C_k , $k \in [1, \ldots, l]$. Здесь n, q и l — количества соответствующих подразделений.

В [5] предложена модель организационной системы, согласно которой принятие решений стратегического центра описывается кортежем $\Psi_0 = \{U_A, U_v, U_I, A_0, \Theta, w(\cdot), v_0(\cdot), I_0\}$, соответствующий кортеж подразделения (агента) имеет вид $\Psi = \{A, R, \Theta, w(\cdot), v(\cdot), I\}$. Здесь A — множество действий агента, R — множество результатов действий, Θ — множество возможных значений

обстановки, I — информация, которой обладает агент на момент принятия решения. Под обстановкой понимается взаимодействие не только с внешней средой, но и всеми элементами организационной системы. Предпочтения агента на множестве возможных результатов деятельности заданы его функцией полезности $v(\cdot)$, а результат деятельности зависит от действия и обстановки $w(\cdot)\colon A \times \Theta \to R$. Соответствующие переменные с нижним индексом $_0$ описывают центр. Действием центра является формирование вектора управления $u = (u_A, u_v, u_I)$, включающего институциональное, мотивационное и информационное управления $u_A \in U_A$, $u_v \in U_v$, $u_I \in U_I$, задающего для агента соответственно допустимое множество действий, функцию полезности и доступную информацию.

Следует подчеркнуть, что рассмотренная модель описывает только процесс принятия решений, не вводя переменные для описания состояния элементов организационной системы, потому ее необходимо расширить для целей проводимого здесь исследования. Для того, что бы записать сформулированную выше проблему формирования ИТ-стратегии в математическом виде, в соответствии с результатами работы [4] введем понятие конфигурации системы, которую определим кортежем $\Sigma = \langle \Sigma_M, \Sigma_R, \Sigma_P, \Sigma_S, p(\Sigma_S\Sigma_P), \Gamma \rangle$. Здесь Σ_M — система потребителей продуктов и услуг, Σ_R — продуктовый портфель, Σ_P — подмодель, определяющая поведение (бизнес-процессы) системы, Σ_S — подмодель, определяющая структуру системы, $p(\Sigma_S\Sigma_P)$ — предикат целостности, определяющий семантику преобразования $\Sigma_P \to \Sigma_S$, Γ — цель системы.

Введем в рассмотрение набор m показателей $F = (F_1, F_2, \ldots, F_m)$, отражающих состояние системы или ее подразделения в любой момент времени τ (для каждого подразделения значение m различно). Каждому состоянию системы соответствует точка в пространстве показателей, а совокупность таких точек при различных значениях τ образует траекторию $\Phi(\tau) = \{F_\tau\}$. Задачу перевода системы из одного состояния в другое можно представить в виде кортежа $\gamma = \{\Sigma, |F_\gamma^0 - F_\gamma^*|, A_\gamma, t\}$, где Σ — организационная система; F_γ — подмножество показателей состояния системы, изменяемых в рамках данной задачи, $F_\gamma \subseteq F$; F_γ^0, F_γ^* — исходное и целевое состояние системы; A_γ — множество действий по достижению цели, $A_\gamma \subseteq A$; t — время, за которое задачу предполагается решить. Это означает необходимость сократить расстояние между векторами F_γ^0 и F_γ^* до нуля. Отметим, что такая постановка является общей как для тактических, так и для стратегических задач, которые различаются лишь расстоянием $|F_\gamma^0 - F_\gamma^*|$ и интервалом времени t, за которое это расстояние предполагается сократить. При этом также не выделяется некий специальный момент времени для стратегического планирования, поскольку стратегические проблемы возникают также нерегулярно, как и тактические. Набор решаемых задач формирует вектор целей Γ, $\gamma \in \Gamma$. При этом возникают два вида ограничений — ресурсные и фазовые. Ограничения первого вида зависят от наличия необходимых для решения задачи ресурсов. Фазовые ограничения определяют те участки пространства показателей системы, попадание в которые нежелательно. В качества примера такого участка применительно к ИТ можно привести временное ухудшение параметров соглашения об уровне сервиса, которое может произойти при запуске в эксплуатацию новой системы.

Подобные рассуждения можно повторить для любого элемента организационной системы, поэтому все записанные выражения для конфигурации и целей справедливы и для подразделений. Очевидно, что стратегический центр будет влиять на определе-

ние целей подразделений. Таким образом, вектор управления u необходимо записать в виде $u = (u_A, u_v, u_I, u_\Gamma)$, где $u_\Gamma \in U_\Gamma$ — стратегическое управление, а в кортежи Ψ и Ψ_0 добавить соответственно Γ и Γ_0: $\Psi_0 = \{U_A, U_v, U_I, U_\Gamma, A_0, \Theta, w(\cdot), v_0(\cdot), I_0, \Gamma_0\}$, $\Psi = \{A, R, \Theta, w(\cdot), v(\cdot), I, \Gamma\}$.

Кроме того, необходимо заметить, что на практике внутри корпорации существуют связи не только между подразделениями и стратегическим центром, но и между подразделениями тоже. Сервисное подразделение получает запросы на реализацию тех или иных услуг не только от руководства корпорации, но и от подразделений P_i,S_j,C_k также в виде векторов управления, аналогичных по структуре $u = (u_A, u_v, u_I, u_\Gamma)$. Это означает, что допустимое множество действий A , функция полезности $v(\cdot)$, доступная информация I и стратегические цели Γ формируются не столько центром управления, сколько внутри самого сервисного подразделения на основании агрегирования всех векторов управления. Эта деятельность и является содержанием формирования стратегического поведения. Ключевыми вопросами при этом являются выбор способа агрегирования управляющих векторов, поступающих от всех подразделений корпорации, и задание целевых состояний F^* и выбор соответствующего набора проектов $\gamma \in \Gamma$ по их достижению. На основании изложенных соображений очевидно, что режим сотрудничества между подразделениями потребителями внутренних ИТ-услуг возможен крайне редко (для этого должны выполняться условия согласования интересов всех подразделений и достаточности ресурсов ИТ-подразделения). Более часто наблюдается режим конкуренции.

Тогда модель стратегического поведения на основе паттернов и конфигураций может быть представлена в виде, показанном на рис. 1.

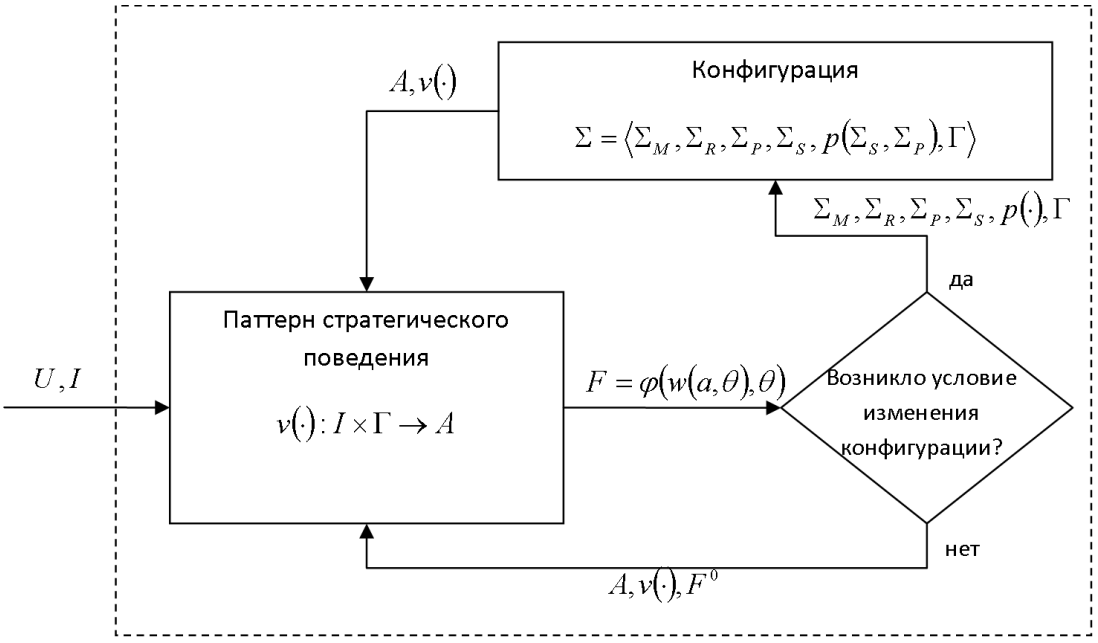


Рис. 1. Модель стратегического поведения

Введенное уточнение касается вида функции полезности $v(\cdot)$, которая должна учитывать цель системы. При наступлении в момент времени τ какого-либо события, требующего реакции, подразделение выбирает действие на основании цели Γ и

информации об обстановке I в этот момент времени $v(\cdot)\colon I \times \Gamma \to A$. Выбранное действие $a = v(I, \Gamma)$, $a \in A$ в зависимости от обстановки приводит к результату $r = w(a, \theta)$, $r \in R$, $\theta \in \Theta$. Новое состояние системы является функцией результата и обстановки $F = \phi(r, \theta)$ или $F = \phi(w(v(I, \Gamma), \theta), \theta)$. Следует особо подчеркнуть разницу между результатом и состоянием. Например, в целях сокращения затрат руководитель ИТ-подразделения выбирает действие «внедрить систему Service Desk». Целевой переменной может служить фонд зарплаты ИТ-подразделения. Результатом этого действия может стать внедренная система, но сокращения затрат можно не достичь, например, из-за роста зарплат специалистов, вызванным увеличением спроса на рынке труда.

4. Предлагаемый метод

П. Страссман показал [16], что инвестиции в ИТ основное влияние оказывают на снижение транзакционных расходов корпорации, к которым относятся административные, маркетинговые и коммерческие расходы, а также затраты на исследования и разработку. Очевидно, данное утверждение будет справедливым и для внутрифирменных транзакционных расходов. Таким образом, будем полагать, что основной задачей ИТ-подразделения в общем случае является снижение внешних и внутренних транзакционных расходов за счет создания информационных систем в подразделениях корпорации.

Запишем функции полезности всех типов подразделений корпорации. Эти функции для подразделений видов C_k, P_i и S_j имеют вид соответственно: $f_C = H - \sum_P \sigma_P - \sum_S \sigma_S - C_C$, $f_P = \sum_C \sigma_P - \sum_S \sigma_S - C_P$, $f_S = \sum_{C+P} \sigma_S - C_S$, где H — доход от продажи продукции и услуг на рынках; $\sum_P \sigma_P$ — размер компенсации за производство продукции и услуг, выплачиваемый производственным центрам P_i; $\sum_S \sigma_S$ — размер компенсации, выплачиваемый сервисным подразделениям S_j за внутрифирменные услуги; $\sum_C \sigma_P$ — компенсация, получаемая производственным подразделением от коммерческих; $\sum_{C+P} \sigma_S$ — компенсация, получаемая сервисным подразделением от коммерческих и производственных; C_C, C_P, C_S — внутренние затраты соответственно коммерческих, производственных и сервисных подразделений. Тогда функция полезности корпорации в целом будет иметь вид $f = f_C + f_P + f_S = H - (C_C + C_P + C_S + C_M)$, где C_M — затраты на корпоративное управление. Затраты $C_\Sigma = C_C + C_P + C_S + C_M$ включают: затраты на трансформацию сырья и материалов в готовые продукты и услуги T_w; транзакционные затраты на управление процессом трансформации T_m; транзакционные затраты на согласование действий между подразделениями внутри корпорации T_a^{int}; транзакционные затраты на достижение согласия с внешними агентами T_a^{ext}. Отсюда $C_\Sigma = T_w + T_m + T_a^{int} + T_a^{ext}$.

Отметим, что порядок перечисления этих затрат соответствует возрастанию сложности проектов по их снижению. Так для сокращения трансформационных затрат T_w отдельно взятого офисного работника, бухгалтера или инженера достаточно предоставить им персональный компьютер с установленным соответствующим программным обеспечением (офисный или бухгалтерский пакет, система подготовки чертежей и т.п.). При этом работники сразу ощущают значительную личную выгоду от внедрения и обычно способствуют изменениям, если не возникает проблемы освоения новых инструментов. Проекты, связанные с сокращением транзакционных затрат ре-

ализуются, как правило, с большими трудностями, поскольку необходимо согласовывать интересы все большего количества людей (работников подразделения, организации в целом и даже внешних организаций). Эти соображения позволяют построить «пирамиду зрелости предприятия», которая может служить для оценки соответствия ИТ и бизнеса.

На рис. 2 представлена модель принятия стратегических решений по поводу развития ИТ на предприятии. В ее состав входит упомянутая выше пирамида зрелости предприятия, определяющая уровень сложности инициатив по внедрению тех или иных ИТ. Данный уровень сложности накладывает ограничения на использование различных элементов архитектуры предприятия. На рис. 2 в качестве примера перечислены некоторые варианты различных организационных и технических решений, относящихся к различным доменам архитектуры предприятия (бизнес-процессы, данные, приложения и техническая архитектура) и соответствующие различным уровням зрелости. Аналогичные ограничения в зависимости от достигнутого и планируемого уровней зрелости накладываются и на использование тех или иных методов управления ИТ, но в этом случае все затраты рассматриваются применительно к ИТ-подразделению (его внутренние трансформационные T_w и управленческие T_m затраты, затраты на согласование с другими подразделениями корпорации T_a^{int} или ИТ-службами партнеров корпорации T_a^{ext}). Более подробно вопрос оптимальной организации ИТ-подразделения рассмотрен в [3].

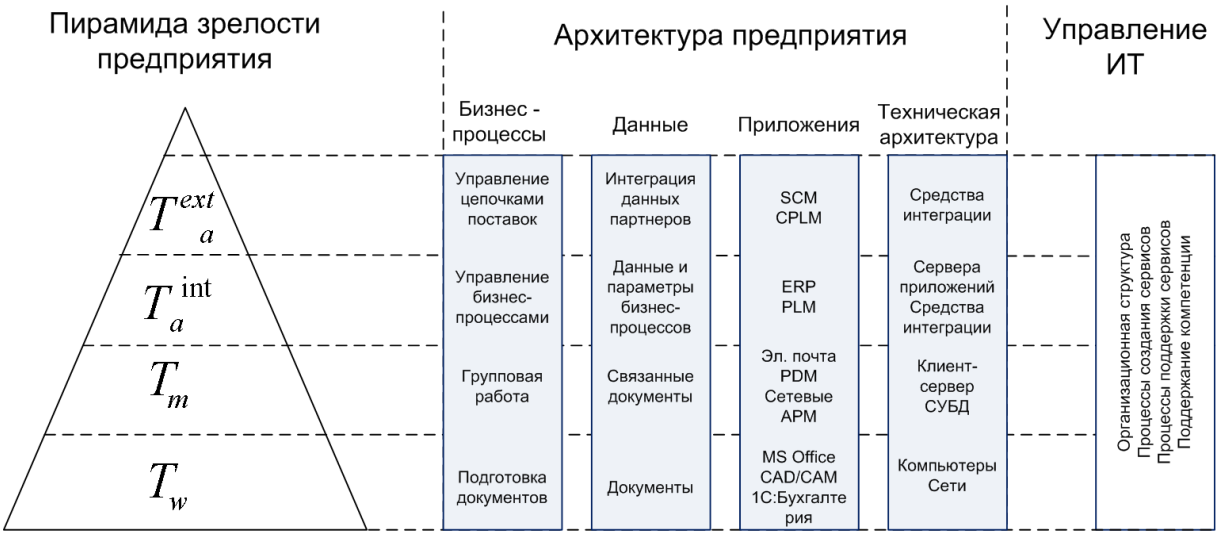


Рис. 2. Модель принятия стратегических решений

На базе предложенной модели сформирован метод выбора стратегических решений $\gamma = \{\Sigma, |F^0_\gamma - F^*_\gamma|, A_\gamma, t\}$ на уровне ИТ-подразделения, который предполагает следующие шаги при возникновении очередной инициативы по созданию какого-либо ИТ-сервиса или системы (независимо от источника этой инициативы):

1) определить тип транзакционных затрат, снижению которых будет способствовать данный сервис или система;
2) построить модель организационной системы до и после внедрения сервиса и провести качественную оценку необходимых и достаточных условий снижения транзакционных затрат;

3) убедиться, что выполняются следующие условия — все элементы архитектуры предприятия и управления ИТ «нижележащих» уровней уже реализованы, на траектории развития организационной системы отсутствуют фазовые ограничения;
4) если указанные условия соблюдаются, а также отсутствуют ресурсные ограничения, реализацию рассматриваемой инициативы можно принимать к исполнению и планировать с помощью одного из методов управления проектами.

5. Пример использования предложенного метода

Рассмотрим применение описанного выше метода к внедрению ИТ-поддержки процесса проектирования новой продукции машиностроительной корпорации. В [7] описана виртуальная среда проектирования, включающая подсистемы цифрового проектирования на основе трехмерной мастер-модели, инженерных расчетов и управления данными испытаний. Данная виртуальная среда проектирования была создана в ОАО «НПО «Сатурн» (г. Рыбинск), российской компании, занимающейся проектированием, производством и послепродажным обслуживанием газотурбинных двигателей и промышленных установок на их базе. Создание системы шло последовательно, «снизу — вверх» в соответствии с моделью, представленной на рис.2. На первом этапе были реализованы системы, направленные на снижение трансформационных затрат T_w (сети, обеспечение ПК, локальные системы CAD/CAM/CAE). На втором — обеспечена групповая работа в подразделениях, ответственных за разработку продуктов и процессов их изготовления (интеграция CAD/CAM/CAE на базе мастер-моделей под управлением PDM-системы), что привело к сокращению затрат T_m. На третьем этапе — интегрированы все данные, связанные с разработкой продукта (модели, фактические данные производства, данные испытаний), что позволило сократить T_a^{int}.

Одним из важнейших вопросов при разработке системы был выбор сценария сокращения транзакционных затрат на взаимодействие с внешними контрагентами T_a^{ext}. Помимо традиционных отношений «поставщик-потребитель» в последнее время развивается модель открытых инноваций, предполагающая активное взаимодействие с внешними источниками новых идей и технологий [13]. В частности, в аэрокосмической отрасли широкое распространение получил инновационный аутсорсинг, когда конечный продукт разбивается на отдельные подсистемы, каждая из которых обладает значительной автономностью и может разрабатываться независимо от других. При этом поставщики отдельных компонентов или сервисных элементов конечного продукта отвечают за их разработку и производство, а головной разработчик фокусируется на интеграции, общем контроле инновационного процесса и своих ключевых компетенциях [6]. В результате сегодня до 80% работ по созданию нового продукта выполняется внешними подрядчиками, которые взаимодействуют с головным разработчиком. Для того, чтобы предприятие было готово к вступлению в подобные инновационные альянсы, необходимо обеспечить простую возможность интеграции его информационной системы с системами других членов альянса. При этом надо отметить, что на международном рынке российские машиностроительные компании сегодня в лучшем случае могут претендовать только на роль разработчиков подсистем готового продукта.

В соответствии с предлагаемым методом построим модель взаимодействия разработчиков и определим оптимальные сценарии сокращения T_a^{ext}. Рассмотрим организационную систему, состоящую из n центров и одного агента. Центры представляют собой головных разработчиков, которые выдают заказы на выполнение субподрядов по разработке агенту. Все работы выполняются с использованием информационных систем (ИС). Будем считать, что тип работ, выполняемых агентом, одинаков для всех центров, например, проектирование изделий в 3D САПР, при этом центры не используют одинаковые системы. В этом случае возникает задача выбора оптимальной стратегии развития ИС для агента. Возможны следующие варианты: (1) внедрение множества Θ различных ИС, соответствующих информационным системам центров. В этом случае, при появлении заказчика с системой $\theta_i \notin \Theta$, агент обязан внедрить систему θ_i; (2) использование единственной внутренней ИС и создание интерфейсов со всеми системами центров; (3) комбинация двух вышеперечисленных стратегий.

В дальнейшем будем считать, что если i-й центр и агент используют одинаковые информационные системы, взаимодействие между ними в процессе выполнения субподряда осуществляется без дополнительных затрат T_a^{ext}. Если они используют различные информационные системы, оба осуществляют затраты на конвертацию данных.

Предпочтения n центров описываются их функциями полезности

$$f_i(w_i) = z_i(w_i) + x_i(w_i) + \sigma_i, \ \mathbf{min} f_i(w_i), \tag{1}$$

где $i \in N\{1, 2, \ldots, n\}$ — множество центров, $z_i(w_i)$ — затраты i-го центра на выполнение части работы w_i (в общем случае работа в рамках одного субподряда может включать несколько заданий, т.е. w_i — вектор); $x_i(w_i)$ — компенсация агенту за выполнение другой части той же работы со стороны i-го центра; σ_i- затраты на конвертацию данных в процессе взаимодействия i-го центра и агента. Затраты на конвертацию данных зависят только от вида используемых информационных систем и не зависят от вида работы, в случае использования одинаковых ИС $\sigma_i = 0$. Данная функция полезности определяет целесообразность передачи работ агенту, очевидно, что это имеет смысл только при $f_i^0(w_i) = z_i^0(w_i) > f_i(w_i)$ или $z_i^0(w_i) > z_i(w_i) + x_i(w_i) + \sigma_i$, где $z_i^0(w_i)$ — затраты i-го центра на выполнение работы без привлечения агента, т.е. когда передача работы агенту позволяет снизить затраты на ее выполнение.

Предпочтения агента представлены функцией полезности

$$f_a(W) = \sum_{i \in N} [x_i(w_i) - c_i(w_i) - \sigma_i], \ \mathbf{max} f_a(W), \tag{2}$$

где $W = \{w_1, w_2, \ldots, w_n\}$ — множество всех работ, выполняемых агентом, $c_i(w_i)$ — затраты агента на выполнение работы w_i. Затраты агента на выполнение работы с помощью информационной системы можно представить как $c_i(w_i) = I + S(t_i)/e$, где I — инвестиции в создание ИС, $S(t_i)$ — затраты на выполнение работ при помощи ИС, t_i — время выполнения работы w_i, e — эффективность использования ИС. Отметим, что инвестиции в создание информационной системы равны нулю, когда агент использует уже существующую у него ИС без доработок.

Определим эффективность использования ИС исходя из следующих соображений. Если агент применяет только одну ИС, считаем, что его работники освоили 100%

ее функций и эффективность использования (т.е. производительность труда с применением системы) равна 1. Если используется более одной ИС, работники вынуждены применять для выполнения однотипных работ различные системы, в результате они имеют меньше времени на полное освоение функций систем и эффективность их использования меньше 1. Будем считать зависимость эффективности использования от количества систем линейной $e = 1 - (k-1)/\alpha$, где k — количество используемых систем, $\alpha > 1$ — произвольное целое число. Очевидно, что предложенная модель имеет смысл только при $e > 0$, это приводит к ограничению на количество используемых систем

$$k < \alpha + 1. \tag{3}$$

Данное ограничение означает, что при увеличении числа ИС может наступить момент, когда персонал агента будет просто не в состоянии освоить работу с ними.

Из формулы (1) следует, что с точки зрения центра целесообразно навязывать агенту использование той же ИС, что использует данный центр. Это ведет к устранению затрат на интерфейс σ_i. С точки зрения агента ситуация не столь однозначна. Рассмотрим построенную модель более подробно.Для простоты положим, что все работы имеют одинаковую сложность $w_i = w$, $i \in N$ и могут быть выполнены агентом за одинаковое время $t_i = t$, $i \in N$, соответственно одинаковы и затраты $S(t_i) = S$, $z_i = z$, $x_i = x$, $i \in N$ на их выполнение, а сложность интерфейсов (т.е. затраты на их создание) между любыми рассматриваемыми ИС описывается функцией $\sigma(\theta_i, \theta_j) = \begin{cases} \sigma_0 = \text{const}, i \neq j \\ 0, i = j \end{cases}$. Также будем считать, что все ИС созданы и инвестиции $I = 0$. Тогда функции полезности (1) и (2) примут вид $f_i = z + x + \sigma(\theta_i, \theta_a)$ для центра и $f_a = \sum_{i \in N} [x - S/e - \sigma(\theta_i, \theta_a)]$ для агента.

Рассмотрим случай, когда агент использует только одну ИС ($k = 1$), не совпадающую с ИС ни одного из центров, и обменивается со всеми центрами данных через интерфейсы. Функции полезности в этом случае:

$$f_i = z + x + \sigma_0, f_a = N(z - S - \sigma_0). \tag{4}$$

В случае, когда агент использует набор ИС, соответствующий множеству ИС центров Θ ($k = N$) получаем следующие функции полезности:

$$f_i = z + x, f_a = N\left(z - S\frac{\alpha}{\alpha - N + 1}\right). \tag{5}$$

Из анализа функций полезности (4), (5) следует, что агенту не выгодно использовать одну ИС и интерфейсы обмена только при $S + \sigma_0 > S\frac{\alpha}{\alpha - N + 1}$. После упрощений получаем неравенство

$$\frac{\sigma_0}{S} > \frac{1}{\alpha/(N-1) - 1}. \tag{6}$$

Исходя из практики затраты на создание интерфейса к системе можно оценить как 10% от затрат на внедрение собственно системы, т.е $\sigma_0/S \approx 0,1$. Также исходя из практических соображений будем считать, что если эффективность использования первой системы равна 1, то для второй аналогичной она не будет превосходить 0,8, для третьей — 0,6 и т.д. Это соответствует $\alpha \leq 5$. Учитывая, что из ограничения

(3) следует $N - 1 < \alpha$, приходим к выводу, что при данных условиях неравенство (6) никогда не выполняется. Требуемых значений правая часть данного неравенства достигает при $N = 2$ и $\alpha \geq 11$, но такое значение α предполагает, что эффективность освоения второй ИС составит 91%.

Таким образом, с точки зрения агента эффективная стратегия развития ИС сводится к использованию единой внутренней системы и созданию универсальных интерфейсов с системами потенциальных заказчиков. Интересы центров при этом удовлетворяются в случае $x_i(w_i) + \sigma_i \leq x_i^{IS}(w_i)$, где $x_i^{IS}(w_i)$ — затраты на компенсацию агенту в случае внедрения им новой системы.

В результате такого анализа в виртуальную среду проектирования ОАО «НПО «Сатурн» были включены интерфейсы, обеспечивающие интеграцию данных о продукте, параметрах его производства и испытаний с другими разработчиками (рис. 3). Данная система использовалась при разработке авиационного газотурбинного двигателя SaM146, созданного в альянсе с французской компанией Snecma, для регионального самолета Сухой СуперДжет-100.

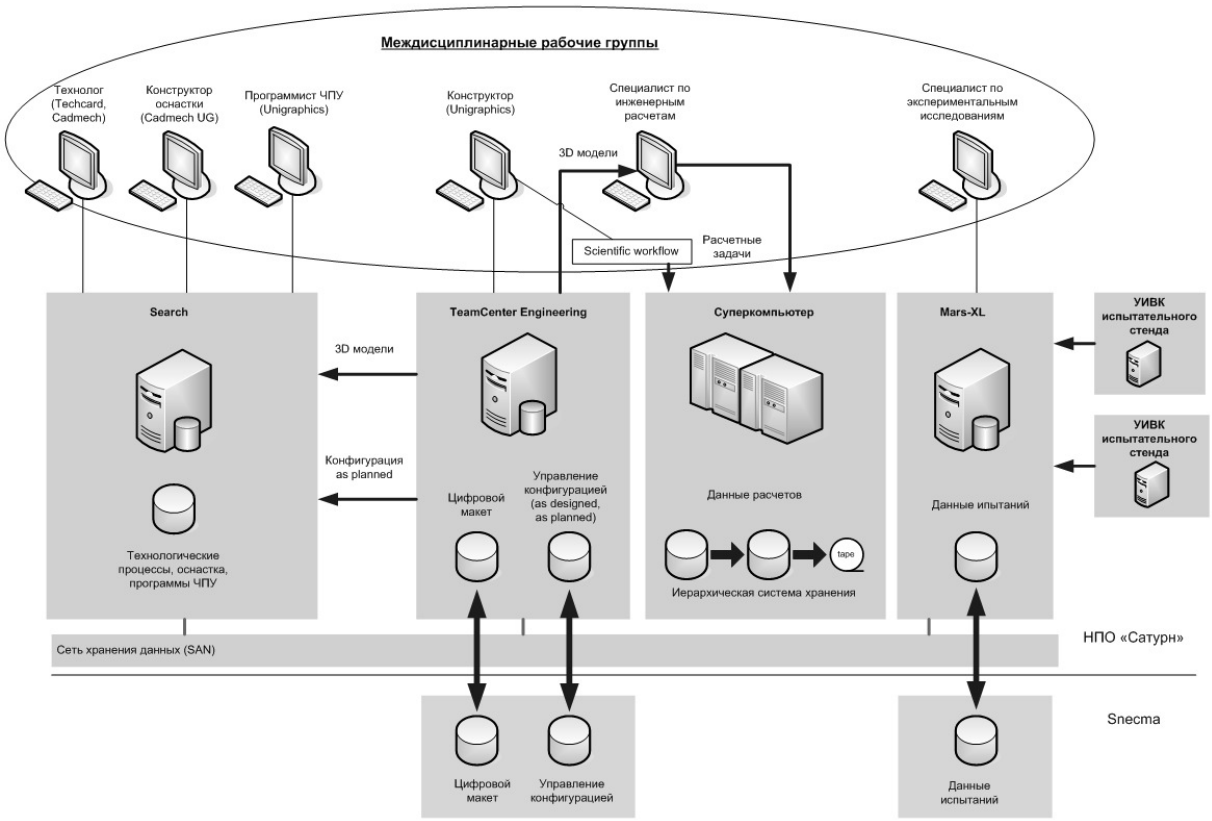


Рис. 3. Виртуальная среда проектирования

6. Выводы

Предложенный метод формирования стратегического поведения ИТ-подразделения интегрирует подходы обеспечения соответствия бизнеса и ИТ, архитектуры предприятия и управления ИТ и позволяет сформировать стратегическую модель оценки инициатив по внедрению различных ИТ-систем и сервисов

независимо от их источника (руководство корпорации, другие ее подразделения, внешний мир, само ИТ-подразделение). При этом формализованное описание стратегических паттернов корпорации или других ее подразделений не требуется. Использование предложенного здесь метода допустимо для рассмотрения инициатив, связанных с внедрением систем и сервисов, направленных на снижение трансформационных и транзакционных затрат. Если формальная бизнес-стратегия имеется, для стратегического планирования развития ИТ можно воспользоваться методом, описанным в [3].

Литература

1. Акоф, Р. Планирование будущего корпорации. / Р. Акоф. – М.: Прогресс, 1985. – 328 с.
2. Данилин, А. Архитектура и стратегия. «Инь» и «Янь» информационных технологий предприятия / А. Данилин, А. Слюсаренко. – М. Интернет Ун-т Инф. Технол., 2005. – 504 с.
3. Зеленков, Ю.А. Формирование ИТ-стратегии предприятия: архитектура, проекты, организация / Ю.А. Зеленков // Вестник РГАТА имени П.А.Соловьева. – 2010. – №3(18). – С. 190-198.
4. Минцберг, Г. Стратегический процесс: Концепции, проблемы, решения / Г. Минцберг, Дж.Б. Куин, С. Гошал – СПб: Питер, 2001. – 688 с.
5. Новиков, Д.А. Теория управления организационными системами. / Д.А. Новиков. – М.: МПСИ, 2005. – 584 с.
6. Управление исследованиями и разработками в российских компаниях. Национальный доклад. – М.: Ассоциация менеджеров, 2011. – 80 с.
7. Шмотин, Ю.Н. Виртуальная среда проектирования / Ю.Н. Шмотин, П.В. Чупин, Ю.А. Зеленков // Открытые системы. – 2010. – No 7. – С. 42–45.
8. Brynjolfsson, E. Wired for Innovation: How Information Technology is Reshaping Economy / E. Brynjolfsson, A. Saunders. – Cambridge: MIT Press, 2010. – 154 p.
9. CobiT 4.1. – IT Governance Institute, 2007. – 213 p.
10. Giachetti, R.E. Design of Enterprise Systems, Theory, Architecture, and Methods. – Boca Raton, FL.: CRC Press, 2010. – 447 p.
11. Henderson, J.C. Strategic Alignment: Leveraging Information Technology for Transforming Organizations / J.C. Henderson, N. Venkatraman // IBM systems journal. – 1993. – 32(1). – P. 472–484.
12. Information Management: The Organizational Dimension / M.J. Earl, ed. – NY: Oxford University Press, 1998. – 514 p.
13. Lichtenthaler, U. Open Innovations: Past Research, Current Debates, and Future Directions // Academy of Management Perspectives. – 2011. – V. 25, No 1 – P. 75–93.
14. Luftman, J.N. Competing in the Information Age: Align in the Sand. – NY.: Oxford University Press, 2003. – 432 p.
15. Ross, J.W. Enterprise Architecture As Strategy: Creating a Foundation for Business Execution / J.W. Ross, P. Weill, D. Robertson. – Harvard Business School Press, 2006 – 288 p.

16. Strassman, P.A. The Squandered Computer – Evaluation the Business Alignment of Information Technologies. – New Canaan, CO.: Information Economics Press, 1997. – 402 p.
17. Weill, P. IT Governance. How Top Performers Manage IT Decision Rights for Superior Results / P. Weill, J.W. Ross. – Harvard Business School Press, 2004. – 286 p.

Юрий Александрович Зеленков, кандидат физико-математических наук, директор по информационным технологиям, ОАО «Научно-производственное объединение «Сатурн», yuri.zelenkov@npo-saturn.ru.

ABOUT STRATEGIC PLANNING OF INFORMATION TECHNOLOGIES DEVELOPMENT IN CORPORATION

Yu.A. Zelenkov, NPO Saturn (Rybinsk, Russian Federation)

Existing methods of corporate IT-strategy development are based on the requirements that must be described in the company's business-strategy, but often formal business-strategy is not available. In this paper we solve the problem of creation a pattern of strategic behavior of IT-division in large corporation in the absence of a formal business-strategy. The general process of decision making in a corporation is described, a model of strategic decision-making about the development of IT is proposed. The proposed model allows determining the level of complexity of IT initiatives implementation on the basis of their impact on transformational and transactional costs. The level of complexity, in turn, imposes restrictions on the use of the various elements of enterprise architecture. A method for making strategic decisions based on this model is developed. Usage of this method by creating a virtual design environment for manufacturing corporation involved as a collaborator in the new product development is demonstrated.

Keywords: IT governance, IT strategy, information management.

References

1. Ackoff R. Creating the Corporate Future: Plan or Be Planned For. New York, John Wiley & Sons, 1981.
2. Danilin A., Slyusarenko A. Architectura i strategia. 'Yin' i 'yang' informacionnyh tehnologiy predpriatiya [Architecture and Strategy. 'Yin' and 'Yang' of Corporate Information Technology]. Moscow, Internet University of IT, 2005. 504 p.
3. Zelenkov Yu.A. Formirovanie IT-strategii predpriyatia: architectura, proecty, organizacia [Corporate IT-strategy Development: Architecture, Projects, Organization]. Vestink RGATA [Bulletin of RGATA], 2010, No 3(18), P .190-198.
4. Mintzberg H., Lampel J., Goshal S., Quinn J.B. The Strategy Process. Pearson, 2003.
5. Novikov D.A. Teoriya upravleniya organizacionnymi systemami [Organizational Systems Management Theory]. Moscow, MPSI, 2005. 584 p.
6. Upravlenie issledovaniyami i razrabotkami v rossiyskih companiyah [Research and Development Management in Russian Companies]. Moscow, Association of Managers, 2011. 80 p.
7. Shmotin Yu.N., Chupin P.V., Zelenkov Yu.A. Virtualnaya sreda proectirovaniya [Virtual Design Environment]. Otkrytye sistemy [Open Systems], 2010. No 7, P. 42–45.

8. Brynjolfsson E., Saunders A. Wired for Innovation: How Information Technology is Reshaping Economy. Cambridge: MIT Press, 2010. 154 p.
9. CobiT 4.1. IT Governance Institute, 2007. 213 p.
10. Giachetti R.E. Design of Enterprise Systems, Theory, Architecture, and Methods. Boca Raton, FL.: CRC Press, 2010. 447 p.
11. Henderson J.C., Venkatraman N. Strategic Alignment: Leveraging Information Technology for Transforming Organizations. IBM Systems Journal, 1993. No 32(1). P. 472–484.
12. Earl M.J. (ed.) Information Management: The Organizational Dimension. NY: Oxford University Press, 1998. 514 p.
13. Lichtenthaler U. Open Innovations: Past Research, Current Debates, and Future Directions. Academy of Management Perspectives, 2011, V. 25, No 1, P. 75–93.
14. Luftman J.N. Competing in the Information Age:Align in the Sand. NY: Oxford University Press, 2003. 432 p.
15. Ross J.W., Weill P., Robertson D. Enterprise Architecture As Strategy: Creating a Foundation for Business Execution. Boston, Harvard Business School Press, 2006. 288 p.
16. Strassman P.A. The Squandered Computer — Evaluation the Business Alignment of Information Technologies. New Canaan, CO, Information Economics Press, 1997. 402 p.
17. Weill P., Ross J.W. IT Governance. How Top Performers Manage IT Decision Rights for Superior Results. Boston, Harvard Business School Press, 2004. 286 p.

Поступила в редакцию 3 февраля 2012 г.

УДК 004.457

БРОКЕР РЕСУРСОВ ДЛЯ ПОДДЕРЖКИ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ГРИД-СРЕД

А.В. Шамакина

Статья посвящена созданию методов и алгоритмов планирования ресурсов, а также разработке на их основе брокера ресурсов для поиска оптимальных ресурсов в проблемно-ориентированных грид-средах. Разработанный алгоритм планирования ресурсов учитывает дополнительные знания о специфике предметной области задания и представление о потоке задач. Приведенный алгоритм основан на алгоритме кластеризации доминирующей последовательности DSC. В отличие от оригинального алгоритма для отображения задач на вычислительные ресурсы используется раскраска графа задач, а объединение задач в один кластер производится с учетом наличия свободных слотов на вычислительных ресурсах. Предложены метод двухфазного резервирования ресурсов и учет проблемных параметров задачи для оценки времени ее выполнения. Приведены варианты использования брокера ресурсов, описаны процесс выделения ресурсов и архитектура брокера ресурсов CAEBeans Broker.

Ключевые слова: брокер ресурсов. алгоритмы планирования ресурсов, грид, резервирование, UNICORE.

Введение

В настоящее время перспективным является направление, связанное с применением грид-технологий [1] для решения ресурсоемких научных задач в разных областях: медицине, инженерном проектировании, нанотехнологиях, прогнозировании климата и т.д. Примером системы, предоставляющей доступ к программным системам класса CAE (Computer Aided Engineering [2]) является система CAEBeans [3], которая позволяет выполнять декомпозицию задач на типовые подзадачи; поиск вычислительных ресурсов, согласно требованиям; постановку задач соответствующим базовым компонентам CAE-систем; мониторинг хода решения задач; предоставление результатов решения задания пользователю.

Технологический цикл решения CAE-задания в общем случае предполагает формирование геометрии задачи, генерацию вычислительной сетки, определение граничных условий, проведение компьютерного моделирования, визуализацию и анализ результатов решения. CAE-задания имеют специфику, в рамках которой необходимо учитывать не только характеристики ресурсов, но и наличие установленных инженерных пакетов, количество доступных лицензий на них и др.

Многие грид-среды осуществляют поддержку сложных приложений с потоком задач (workflow [4]), которые обычно моделируют DAG. Можно перечислить такие инструменты, как Condor DAGMan [5], CoG [6], Pegasus [7], GridFlow [8] и ASKALON [9]. Использование дополнительных знаний о специфике области задачи и представление о потоке задач может существенно улучшить эффективность методов планирования ресурсов. Однако, ни один из существующих на сегодняшний день инструментов не учитывает эту специфику.

Цель данной работы состоит в создании методов и алгоритмов планирования ресурсов, а также в разработке на их основе брокера ресурсов для поиска оптимальных ресурсов в проблемно-ориентированных грид-средах.

В настоящей статье рассмотрены алгоритм планирования ресурсов и реализация на его основе брокера ресурсов CAEBeans Broker. Статья организована следующим

образом. В разделе 1 описывается назначение брокера ресурсов и платформа для его реализации, во 2 разделе описываются варианты использования брокера ресурсов, 3 раздел содержит описание архитектуры брокера ресурсов. Процесс выделения ресурсов компонента CAEBeans Broker рассматривается в 4 разделе и в 5 разделе приводится алгоритм планирования ресурсов. В заключении суммируются основные результаты, полученные в данной работе.

1. Брокер ресурсов CAEBeans Broker

CAEBeans Broker — это компонент системы CAEBeans, который принимает задания от пользователя, согласовывает требования к ресурсам и находит наиболее подходящие вычислительные элементы для каждой из задач. Можно выделить следующие основные задачи брокера ресурсов: обработка базы данных ресурсов грид-среды; анализ запросов на предоставление ресурсов, поступающих от внешних клиентов; сбор и предоставление информации об актуальном состоянии грид-среды.

Реализация CAEBeans Broker производится на языке программирования Java в виде сервиса на базе платформы UNICORE. Данный подход обеспечивает независимость компонента от вычислительной платформы, предоставление полной информации о текущем состоянии экземпляра сервиса, а также поддерживает возможность надежного и безопасного исполнения, управление временем жизни; рассылку уведомлений об изменении состояния экземпляра сервиса, управление политикой доступа к ресурсам, управление сертификатами доступа.

2. Варианты использования компонента CAEBeans Broker

Программный компонент CAEBeans Broker представляет собой сервис, предоставляющий интерфейс для выбора наиболее подходящих ресурсов. В процессе работы брокера поиск требуемых ресурсов выполняется для нескольких заданий одновременно. Кроме этого, с брокером ресурсов взаимодействует вспомогательный компонент для сбора информации. Диаграмма вариантов использования компонента CAEBeans Broker приведена на рис. 1.

Вариант использования «Выделить ресурсы» передает запрос от клиента к брокеру ресурсов системы CAEBeans. В роли клиента выступает компонент CAEBeans Server [10], отвечающий за исполнение заданий и их мониторинг. Вариант использования начинается, когда CAEBeans Server указывает в запросе требования к ресурсам необходимым для выполнения задания. Вариант использования «Выделить ресурсы» включает в себя вариант использования «Установить резервирование», который позволяет установить резервирование выбранных ресурсов.

Вариант использования «Освободить ресурсы» уведомляет брокера ресурсов о необходимости освобождения выбранных ресурсов. Данный вариант использования также предполагает удаление заявки на резервирование ресурсов [11]. Соответствующий запрос на удаление ресурсов формируется вариантом использования «Удалить резервирование».

Вариант использования «Получить статус» запрашивает информацию о выделении ресурсов. В случае успешного выделения ресурсов клиент запрашивает список

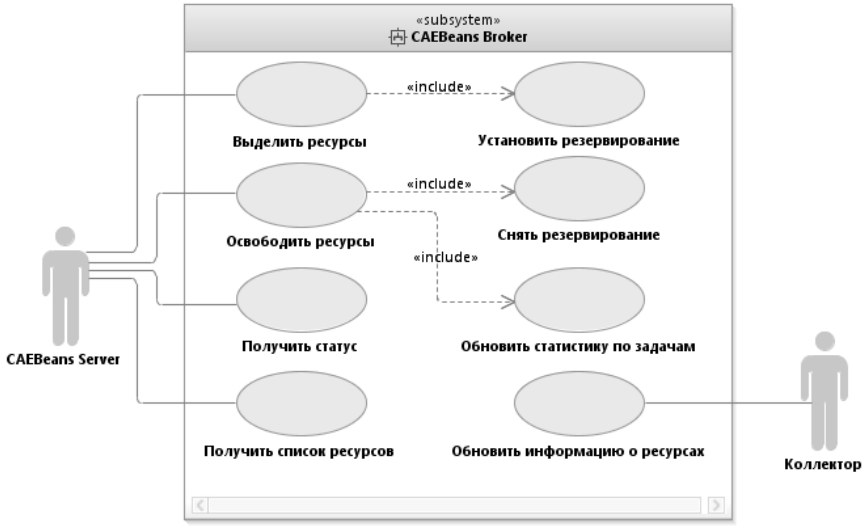


Рис. 1. Диаграмма вариантов использования подсистемы CAEBeans Broker

выделенных для его задания ресурсов с помощью варианта использования «Получить список ресурсов».

Варианты использования «Обновить информацию о ресурсах» и «Обновить статистику по задачам» обновляют информацию о характеристиках ресурсов и статистике выполнения заданий соответственно в каталоге брокера ресурсов.

3. Архитектура брокера ресурсов CAEBeans Broker

Согласно представленной на рис. 1 диаграмме вариантов использования была разработана следующая архитектура брокера ресурсов CAEBeans Broker (рис. 2). CAEBeans Broker состоит из следующих компонентов.

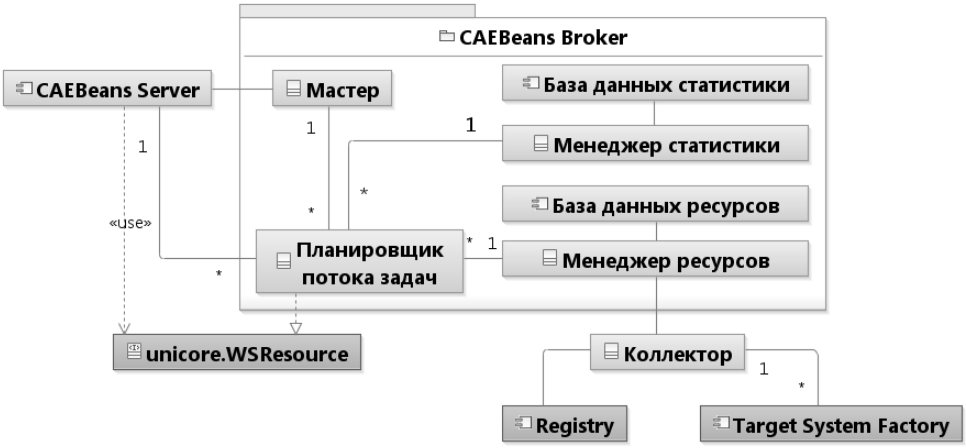


Рис. 2. Архтектура брокера ресурсов CAEBeans Broker

- Мастер принимает запросы от CAEBeans Server и создает экземпляр планировщика потока задач, представляющий собой WS-ресурс в терминах UNICORE.

- Экземпляр планировщика потока задач осуществляет обработку одного запроса. Формирует список требуемых для исполнения задания ресурсов и производит их резервирование.
- Менеджер ресурсов управляет базой данных ресурсов, содержащей информацию о целевых системах и резервировании ресурсов.
- Менеджер статистики управляет базой данных статистики, содержащей информацию о статистике выполнения задач.
- Коллектор работает независимо от CAEBeans Broker и осуществляет сбор информации для базы данных ресурсов.

Темным цветом на рис. 2 выделены компоненты платформы UNICORE, с которыми взаимодействуют компоненты системы CAEBeans.

4. Процесс выделения ресурсов компонента CAEBeans Broker

Взаимодействие компонентов брокера ресурсов, представленных на рис. 2, осуществляется согласно диаграммам последовательности выделения ресурсов в CAEBeans Broker (рис. 3) и сбора информации (рис. 4).

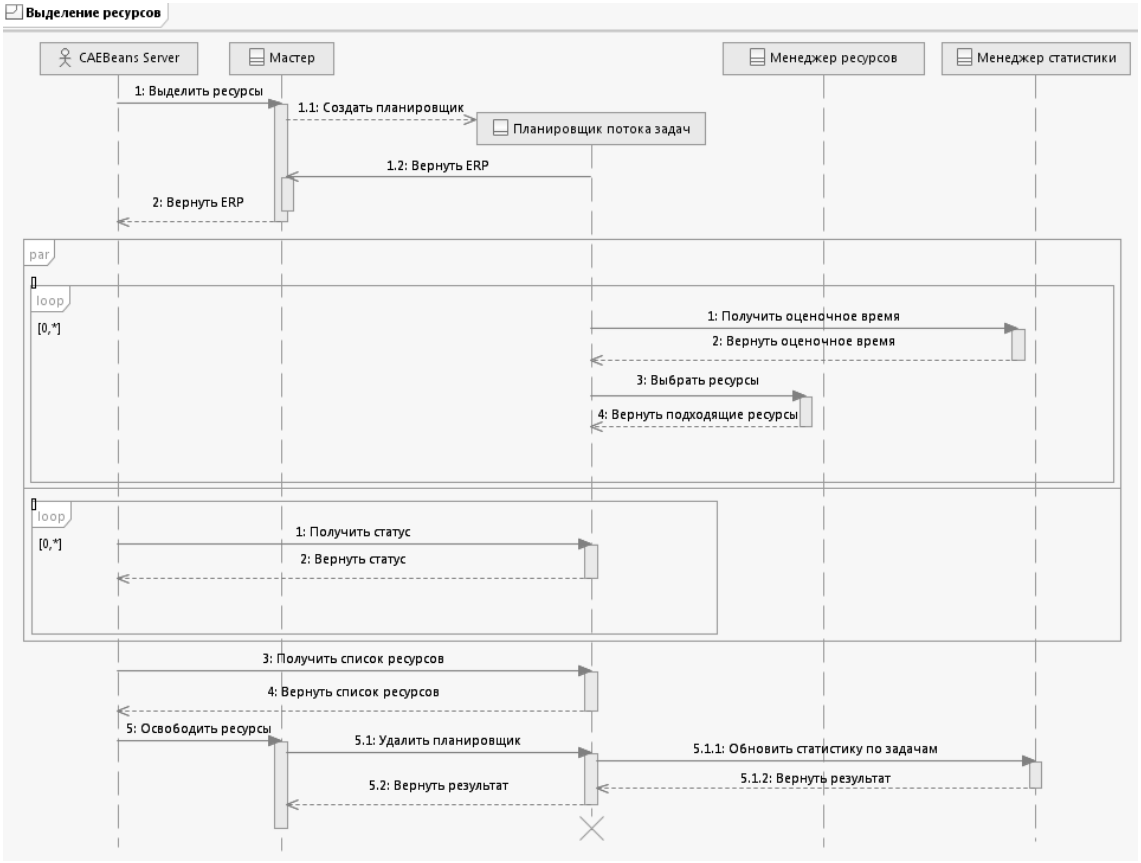


Рис. 3. Диаграмма последовательности выделения ресурсов в CAEBeans Broker

Рассмотрим процесс выделения ресурсов компонентом CAEBeans Broker.

1. Компоненту CAEBeans Server предоставляется доступ к брокеру ресурсов посредством сервиса Мастер. CAEBeans Server подключается к нему с помощью имеющегося сертификата безопасности. Затем CAEBeans Server вызывает метод

«Выделить ресурсы» для выделения ресурсов заданию и передает методу в качестве входного параметра абстрактный поток задач, содержащий требования к ресурсам для каждой из задач.

2. После вызова комонентом CAEBeans Server метода «Выделить ресурсы» сервис Мастер выполняет следующую последовательность действий.
   (a) Мастер создает экземпляр планировщика потока задач в виде WS-ресурса. Планировщик потока задач возвращает сервису Мастер свой уникальный идентификатор. В дальнейшем, любую информацию о процессе поиска ресурсов для задания можно получить непосредственно у экземпляра планировщика потока задач, обратившись к нему с помощью данного идентификатора.
   (b) Мастер записывает уникальный идентификатор созданного экземпляра планировщика потока задач в свой пул.
   (c) Мастер инициализирует специальный объект — конкретный поток задач, извлекая необходимую информацию из абстрактного потока задач. Конкретный поток задач содержит отображение каждой задачи из задания на конкретный ресурс. Мастер записывает конкретный поток задач с помощью фреймворка Ehcahe сначала в кеш, а потом на диск. В качестве ключа к данному объекту используется уникальный идентификатор планировщика потока задач.
   (d) Мастер возвращает компоненту CAEBeans Server уникальный идентификатор созданного экземпляра планировщика потока задач с помощью метода «Вернуть EPR». В качестве идентификатора выступает адрес конечной точки (Endpoint References, EPR).
3. После создания планировщик потока задач считывает соответствующий конкретный поток задач из кеша или с диска. Дальнейшие шаги последовательно повторяются до завершения процесса выделения ресурсов.
   (a) Планировщик потока задач получает оценку времени выполнения для каждой задачи у менеджера статистики, вызвав метод «Получить оценочное время».
   (b) Планировщик потока задач начинает поиск требуемых ресурсов в базе данных ресурсов помощью метода «Выбрать ресурсы» посредством запроса к менеджеру ресурсов. В случае успешного поиска ресурсов планировщик потока задач получает список ресурсов, в противном случае — получает пустое сообщение. Выбранные ресурсы резервируются.
   (c) CAEBeans Server проверяет статус выделения ресурсов, обращаясь непосредственно к своему экземпляру планировщика потока задач.
4. В случае успешного выделения ресурсов CAEBeans Server запрашивает список выделенных для его задания ресурсов с помощью метода «Получить список ресурсов».
5. После выполнения задания CAEBeans Server отправляет компоненту Мастер запрос на освобождение ресурсов.
6. После вызова компонентом Мастер метода «Удалить планировщик» экземпляр планировщика потока задач снимает резервирование с выделенных ресурсов и обновляет статистику по выполнившемуся заданию в базе данных статистики.

Процесс сбора информации для базы данных ресурсов представлен на рис. 4. Шаги по обновлению информации повторяются последовательно в течение всего времени исполнения компонента Коллектор. Обновление информации о ресурсах подразумевает сбор информации об аппаратных, программных и лицензионных ресурсах, доступных CAEBeans Broker. Статистика выполнения задач включает информацию о реальном времени выполнения задач с конкретными параметрами на конкретных вычислительных узлах.

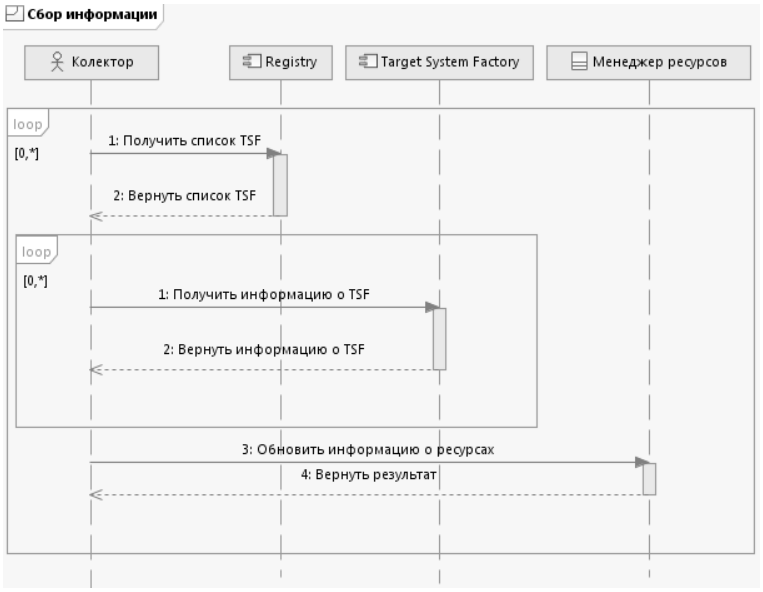


Рис. 4. Диаграмма последовательности сбора информации в системе CAEBeans Broker

5. Алгоритм планирования брокера ресурсов

В рамках данной работы предлагается разработать алгоритм планирования ресурсов в распределенных проблемно-ориентированных вычислительных средах, учитывающий дополнительные знания о специфике предметной области задания и представление о потоке работ; использующий резервирование ресурсов; управляющий как аппаратными, программными, так и лицензионными ресурсами распределенной вычислительной сети (РВС).

Предлагается использовать следующие методы:

- метод двухфазного резервирования ресурсов;
- метод кластеризации доминирующей последовательности;
- учет проблемных параметров задачи для оценки ее времени выполнения.

МЕТОД ДВУХФАЗНОГО РЕЗЕРВИРОВАНИЯ РЕСУРСОВ. Резервирование ресурсов в РВС в настоящее время является сложной задачей, поскольку требует наличия в системе управления задачами компонента для централизованного планирования ресурсов и мгновенного принятия решения о резервировании ресурсов. Для решения данной проблемы предлагается использовать метод двухфазного резервирования ресурсов. Данный метод позволяет выполнять на первой фазе предварительное резервирование (маркировку) требуемых ресурсов и одновременно производить поиск оптимального расписания для потока задач. Для планирования отдельного пото-

ка задач отводится некоторое время, ограниченное приоритетом выполнения потока задач. По истечении данного времени первая фаза завершится. В рамках выполнения второй фазы происходит окончательное выделение ресурсов для потока задач или освобождение всех маркированных ресурсов. В обоих случаях операции выделения/освобождения выполняться атомарно.

МЕТОД КЛАСТЕРИЗАЦИИ ДОМИНИРУЮЩЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ. Отображение потока задач на вычислительные ресурсы производится с помощью адаптированного метода кластеризации доминирующей последовательности (Dominant Sequence Clustering, DSC [12]). Суть метода DSC сводится к минимизации параллельного времени выполнения потока задач. Однако в оригинальном методе DSC планирование осуществляется в два этапа. На первом этапе происходит объединение нескольких задач в кластеры (группы), на втором этапе — отображение кластеров задач на реальные вычислительные ресурсы. Адаптация данного метода подразумевает использование предварительного резервирования ресурсов на этапе объединения кластеров: поиск свободных слотов на вычислительных ресурсах производится непосредственно при планировании потока задач.

УЧЕТ ПРОБЛЕМНЫХ ПАРАМЕТРОВ ЗАДАЧИ ДЛЯ ОЦЕНКИ ЕЕ ВРЕМЕНИ ВЫПОЛНЕНИЯ. Время выполнения отдельных задач в потоке задач по умолчанию указывается пользователем. Однако в большинстве случаев пользователи указывают существенно завышенное время. Для более точной оценки времени выполнения задач статистика по задачам хранится в отдельной базе данных, которой управляет менеджер статистики. Менеджер статистики предоставляет информацию о точном времени исполнения задач, исходя из имеющихся данных о значениях параметров задачи и архитектуре вычислительных узлов, на которых данная задача исполнялась ранее.

На рис. 5 представлен алгоритм планировани ресурсов CAEBeans Broker. На начальном этапе задание представлено в виде ориентированного ациклического графа. Вершинами графа являются задачи. Каждая вершина является отдельной группой. Необходимо разбить вершины графа на группы таким образом, чтобы количество групп равнялось числу имеющихся вычислительных ресурсов.

Первая фаза алгоритма начинается со строки номер 1. Данная фаза ограничена по времени. В соответствии с оригинальным алгоритмом находим максимальное параллельное время (Parallel Time, PT) и доминирующую последовательность (Dominant Sequence, DS). Затем производим поиск ребра с максимальным весом. Если объединение вершин, соединенных данным ребром, уменьшит параллельное время, то произведем их объединение в одну группу.

Адаптируем оригинальный алгоритм DSC следующим образом. Сопоставим каждому вычислительному ресурсу некоторый цвет. Объединение вершин графа будем производить только в том случае, если для данных вершин существуют подходящие свободные слоты на одном вычислительном ресурсе. Вершины, объединенные в одну группу, раскрасим в один цвет – цвет ресурса. Если подходящих слотов найдено не было, то находим следующее ребро с максимальным временем и повторим процедуру поиска слотов для его вершин.

На шаге 17 начинается вторая фаза алгоритма. Произведем поиск вершин графа, для которых не было зарезервировано слотов на вычислительных ресурсах. Обозначим множество нераспланированных вершин как $V_{nonsched}$. Выделим для каждой

вершины из множества $V_{nonsched}$ первые подходящие свободные слоты на вычислительных ресурсах. Произведем окончательное выделение ресурсов для всех вершин графа.

```
1.  while time > 0 do
2.    while количество групп вершин > количество вычислительных ресурсов do
3.      найти PT_max и соответствующий ему DS_max
4.      выполнить сортировку весов дуг DS_max, результат записать в массив W_E
5.      for i=1 to |W_E| do
6.        найти вершины n_k и n_m, соединяемые i-той дугой
7.        if существуют подходящие слоты для вершин n_k и n_m
8.        на одном вычислительном ресурсе then
9.          объединить вершины в одну группу
10.         раскрасить вершины цветом, соответствующим выбранному ресурсу
11.         break for
12.       end if
13.     end for
14.   end while
15.   time --;
16. end while
17. if V_nonsched == ∅ then
18.   for j=1 to |V_nonsched| do
19.     найти слот S_j, на котором задача j будет запущена раньше;
20.     зарезервировать слот S_j;
21.   end for
22. end if
23. выделить ресурсы;
```

Рис. 5. Алгоритм планирования ресурсов CAEBeans Broker

6. Заключение

В данной статье рассмотрен алгоритм планирования ресурсами в распределенных проблемно-ориентированных вычислительных средах, учитывающий дополнительные знания о специфике предметной области задания и представление о потоке задач; использующий резервирование ресурсов; управляющий как аппаратными, программными, так и лицензионными ресурсами РВС. Приведены варианты использования брокера ресурсов, описаны процесс выделения ресурсов и архитектура брокера ресурсов CAEBeans Broker.

В качестве дальнейших направлений работы можно выделить следующее: проведение вычислительных экспериментов для оценки эффективности алгоритма планирования на суперкомпьютерах «СКИФ-Аврора ЮУрГУ» и «СКИФ Урал» Суперкомпьютерного центра ЮУрГУ.

Работа выполнена при поддержке грантов РФФИ № 11-07-00478-а и № 12-07-31076, гранта Президента Российской Федерации МК-1987.2011.9, в рамках государственного задания Министерства образования и науки РФ 8.3786.2011.

Литература

1. Foster, I. The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure / I. Foster, C. Kesselman. – San Francisco: Morgan Kaufman, 2003. – P. 748.
2. Raphael, B. Fundamentals of computer aided engineering / B. Raphael, I. F. C. Smith. – London: John Wiley, 2003. – P. 324.
3. Радченко, Г.И. Сервисно-ориентированный подход к использованию систем инженерного проектирования и анализа в распределенных вычислительных средах / Г.И. Радченко // Параллельные вычислительные технологии (ПаВТ'2011): Труды международной научной конференции (Москва, 28 март. – 1 апр. 2011 г.). – Челябинск: Издательский центр ЮУрГУ, 2011. – С. 606 – 616.
4. Yu, J. A Taxonomy of Workflow Management Systems for Grid Computing / J. Yu, R. Buyya // Grid Computing. – 2005. – V. 3, № 3. – P. 171–200.
5. Condor. High Throughput Computing. URL: http://www.cs.wisc.edu/condor/ (дата обращения 20.05.2012)
6. Laszewski, G. CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids / G. Laszewski, I.Foster // Java Grande of the ACM. – Jun. 2000. – P. 97–106.
7. Deelman, E. Pegasus: Mapping Scientific Workflows onto the Grid / E. Deelman, J. Blythe // Grid Computing: Second European AcrossGrids Conference (AxGrids 2004). – Jan. 2004. – P. 11–26.
8. Cao, J. GridFlow: Workflow Management for Grid Computting / J. Cao, S. A. Jarvis // International Symposium on Cluster Computing and the Grid (CCGrid'03). – May. 2003. – P. 198–205.
9. Wieczorek, M. Scheduling of Scientific Workflows in the ASKALON Grid Environment / M. Wieczorek, R. Prodan, T. Fahringer // ACM SIGMOD Record. – 2005. – V. 34, № 3. – P. 56–62.
10. Федянина, Р.С. CAEBeans Server: среда выполнения проблемно-ориентированных оболочек над инженерными пакетами / Р.С. Федянина // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 март. – 2 апр. 2010 г.). – Челябинск: Издательский центр ЮУрГУ, 2010. – С. 621 – 628.
11. Mateescu, G. Quality of Service on the Grid via Metascheduling with Resource Co-Scheduling and Co-Reservation / G. Mateescu // High Performance Computing Applications. – 2003. – V. 17, № 3. – P. 209–218.
12. Yang, T. DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors / T. Yang, A. Gerasoulis // IEEE Transactions on Parallel and Distributed Systems. – 1994. – V. 5, № 9. – P. 951–967.

Анастасия Валерьевна Шамакина, старший преподаватель, кафедра «Системное программирование», Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), sham2004@bk.ru.

BROKERING SERVICE FOR SUPPORTING PROBLEM-ORIENTED GRID ENVIRONMENT

A.V. Shamakina, South Ural State University (Chelyabinsk, Russian Federation)

This paper describes scheduling methods and algorithms of resources, and also the development on their base of the broker resource for search optimum resources in problem-oriented grid-environment. The developed scheduling algorithm considers additional knowledge about subject domain specifics of tasks and the representation about a workflow. The algorithm is based on a dominant sequence clustering algorithm (DSC). Unlike the original algorithm is that, for mapping tasks on the computing resources used by task graph coloring and the merging of tasks in a cluster is based on the availability of free slots on computing resources. Proposed a diphasic reservation method of resources and accounting problem of the problem parameters to estimate the time of its execution. Use cases of the broker resource are also given, process of resource allocation and architecture of the resource broker CAEBeans Broker are described.

Keywords: broker resource, scheduling algorithms of resources, grid, reservation, UNICORE.

References

1. Foster I., Kesselman C. The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure. San Francisco: Morgan Kaufman, 2003. 748 p.
2. Raphael B., Smith I.F.C. Fundamentals of Computer-Aided Engineering. London: John Wiley, 2003. 324 p.
3. Radchenko G.I. Servisno-orientirovannyi podkhod k ispol'zovaniyu system inzhenernogo proektirovaniya i analiza v raspredelennykh vychislitel'nykh sredakh [A Service-Oriented Approach to Using of CAE-systems in Distributed Computing Environments]. Parallelnye vychislitelnye tekhnologii (PaVT'2011): Trudy mezhdunarodnoj nauchnoj konferentsii (Moskva, 28 marta — 1 aprelya 2011) [Parallel Computational Technologies (PCT'2011): Proceedings of the International Scientific Conference (Moscow, Russia, March 28 — April 1, 2011)]. Chelyabinsk, Publishing of the South Ural State University, 2011. P. 606–616.
4. Yu J., Buyya R. A Taxonomy of Workflow Management Systems for Grid Computing. Grid Computing, 2005. V. 3, No 3. P. 171–200.
5. Condor. High Throughput Computing. URL: http://www.cs.wisc.edu/condor/
6. Laszewski G., Foster I. CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids. Java Grande of the ACM. 2000. P. 97–106.
7. Deelman E., Blythe J. Pegasus: Mapping Scientific Workflows onto the Grid. Grid Computing: Second European AcrossGrids Conference (AxGrids 2004). 2004. P. 11–26.
8. Cao J., Jarvis S.A. GridFlow: Workflow Management for Grid Computting. International Symposium on Cluster Computing and the Grid (CCGrid'03). 2003. P. 198–205.

9. Wieczorek M., Prodan R., Fahringer T. Scheduling of Scientific Workflows in the ASKALON Grid Environment. ACM SIGMOD Record, 2005. V. 34, No 3. P. 56–62.

10. Fedyanina R.S. CAEBeans Server: sreda vypolneniya problemno-orientirovannykh obolokhek nad inzhenernymi paketami [CAEBeans Server: the Runtime Environment of Problem-oriented Shells over Engineering Packages]. Parallelnye vychislitelnye tekhnologii (PaVT'2010): Trudy mezhdunarodnoj nauchnoj konferentsii (Ufa, 29 marta — 2 aprelya 2010) [Parallel Computational Technologies (PCT'2010): Proceedings of the International Scientific Conference (Ufa, Russia, March 29 — April 2, 2010)]. Chelyabinsk, Publishing of the South Ural State University, 2010. P. 621–628.

11. Mateescu G. Quality of Service on the Grid via Metascheduling with Resource Co-Scheduling and Co-Reservation. High Performance Computing Applications, 2003. V. 17, No 3, P. 209–218.

12. Yang T., Gerasoulis A. DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors. IEEE Transactions on Parallel and Distributed Systems, 1994. V. 5, No 9, P. 951–967.

Поступила в редакцию 6 августа 2012 г.

СВЕДЕНИЯ ОБ ИЗДАНИИ

Серия основана в 2012 году.

Свидетельство о регистрации ПИ № ФС77-26455 выдано 13 декабря 2006 г. Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии `http://vestnikvmi.susu.ru`. **Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются и назад авторам не высылаются.**

2. Адрес редакции научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:

   Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, Южно-Уральский государственный университет, факультет Вычислительной математики и информатики, кафедра СП, ответственному секретарю, доценту Цымблеру Михаилу Леонидовичу.

3. Адрес электронной почты редакции: `vestnikvmi@gmail.com`

4. **Плата с авторов за публикацию рукописей не взимается, и гонорары авторам не выплачиваются.**

5. Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год (февраль, май, август и ноябрь).