

Автоматизированное проектирование эффективных программ на основе логического анализа численных алгоритмов

В. Н. Алеева

Южно-Уральский государственный университет (НИУ)

Мальцевские чтения
Новосибирск, 2022

Актуальность исследования

- ▶ Повышение производительности параллельных вычислительных систем — актуальная проблема для развития современных фундаментальных и прикладных вычислений.
- ▶ Ее решением может быть повышение эффективности параллельных вычислений.
- ▶ Особое значение при этом имеет автоматизированное проектирование эффективных программ, использующих ресурс параллелизма алгоритмов полностью.

Цель и задачи исследования

Цель исследования — решение описанной проблемы в случае численных алгоритмов.

Для достижения цели решены следующие задачи.

1. Создана программная Q -система для исследования ресурса параллелизма численных алгоритмов
2. Разработан метод проектирования Q -эффективных программ, использующих ресурс параллелизма алгоритмов полностью.
3. Разработана технология Q -эффективного программирования для повышения эффективности реализации численных методов и алгоритмических проблем.

Цель и задачи исследования

В настоящее время решается задача

4. Создание единой программной системы для эффективной реализации численных алгоритмов.

Теоретический фундамент для исследования

- ▶ Теоретическим фундаментом для исследования является концепция Q -детерминанта.
- ▶ Концепция Q -детерминанта — один подходов к распараллеливанию численных алгоритмов.
- ▶ Концепция основана на универсальном описании численных алгоритмов — представлении в форме Q -детерминанта.

Алгоритмическая проблема

Алгоритмическая проблема может быть представлена в виде

$$\bar{y} = F(N, B),$$

где $N = \{n_1, \dots, n_k\}$ — множество параметров размерности проблемы или пустое множество,

B — множество входных данных,

$\bar{y} = (y_1, \dots, y_m)$ — множество выходных данных, m является вычислимой функцией параметров N .

Далее через \bar{N} будем обозначать вектор $(\bar{n}_1, \dots, \bar{n}_k)$, где \bar{n}_i — некоторое заданное значение параметра n_i ($1 \leq i \leq k$).

Кратко о понятии Q -детерминанта алгоритма

- ▶ Пусть \mathcal{A} — численный алгоритм для решения алгоритмической проблемы,
 Q — множество операций, используемых алгоритмом \mathcal{A} ,
 B — множество входных и $\bar{y} = (y_1, \dots, y_m)$ — множество выходных данных алгоритма \mathcal{A} .
- ▶ Для каждого y_i ($i = 1, \dots, m$) строится Q -терм, который описывает все способы вычисления y_i в зависимости от B в соответствии с алгоритмом \mathcal{A} .
- ▶ Алгоритм \mathcal{A} определяет свой Q -детерминант, как набор Q -термов.
- ▶ С помощью Q -детерминанта фактически выполняется логический анализ алгоритма \mathcal{A} , позволяющий строить все возможные вычисления y_i ($i = 1, \dots, m$) по B согласно алгоритму.

Примеры представления алгоритмов в форме Q -детерминантов

- ▶ Алгоритм умножения плотных матриц сразу представлен в форме Q -детерминанта $c_{ij} = \sum_{s=1}^k a_{is} b_{sj}$ ($i = 1, \dots, n; j = 1, \dots, m$) из nm безусловных Q -термов.
 - ▶ Однако численные алгоритмы чаще не представлены в форме Q -детерминанта, но их представление можно найти.
- 1) Представление в форме Q -детерминанта алгоритма, реализующего метод Гаусса—Жордана решения СЛАУ (n — размер матрицы):

$$x_j = \{(u_1, w_1^j), \dots, (u_{n!}, w_{n!}^j)\} (j = 1, \dots, n).$$

Q -детерминант состоит из n условных Q -термов длины $n!$.

- 2) Представление в форме Q -детерминанта алгоритма, реализующего метод Якоби решения СЛАУ (n — размер матрицы, ε — точность вычислений):

$$x_i = \{(\|\bar{x}^1 - \bar{x}^0\| < \varepsilon, x_i^1), \dots, (\|\bar{x}^k - \bar{x}^{k-1}\| < \varepsilon, x_i^k), \dots\} (i = 1, \dots, n).$$

Q -детерминант состоит из n условных бесконечных Q -термов.

Q-эффективная реализация алгоритма

- ▶ Реализацией алгоритма \mathcal{A} , представленного в форме Q -детерминанта, называется вычисление Q -термов.
- ▶ Если реализация такова, что Q -термы вычисляются одновременно (параллельно) и при вычислении каждого из них операции выполняются по мере их готовности, то реализация называется *Q*-**эффективной**.
- ▶ Если алгоритм допускает распараллеливание, то его *Q*-эффективная реализация полностью использует ресурс параллелизма алгоритма.
- ▶ Реализация алгоритма называется *выполнимой*, если одновременно необходимо выполнять конечное число операций.

Характеристики ресурса параллелизма алгоритма

Пусть $W(\bar{N})$ — выражения, из которых состоят Q -термы Q -детерминанта алгоритма \mathcal{A} , $T^{w(\bar{N})}$ — уровень вложенности выражения $w(\bar{N})$.

Введем характеристики:

- 1) $D_{\mathcal{A}}(\bar{N})$ — максимальное число уровней вложенности выражений $W(\bar{N})$, т.е. $D_{\mathcal{A}}(\bar{N}) = \max_{w(\bar{N}) \in W(\bar{N})} T^{w(\bar{N})}$;
- 2) $P_{\mathcal{A}}(\bar{N})$ — максимальное из значений, представляющих количество операций на каждом из уровней вложенности всех выражений $W(\bar{N})$, т.е. $P_{\mathcal{A}}(\bar{N}) = \max_{1 \leq r \leq D_{\mathcal{A}}(\bar{N})} \sum_{w(\bar{N}) \in W(\bar{N})} O_r^{w(\bar{N})}$, где $O_r^{w(\bar{N})}$ — количество операций уровня вложенности r выражения $w(\bar{N})$.

$D_{\mathcal{A}}(\bar{N})$ будем называть высотой алгоритма, $P_{\mathcal{A}}(\bar{N})$ — его шириной.

$D_{\mathcal{A}}(\bar{N})$ характеризует время выполнения Q -эффективной реализации алгоритма,

$P_{\mathcal{A}}(\bar{N})$ — количество процессоров, необходимое для ее выполнения.

ЗАДАЧА 1. Программная *Q*-система для исследования ресурса параллелизма численных алгоритмов

В режиме просмотра информации *Q*-система (*Q-system*)
доступна всем по адресу

<https://qclient.herokuapp.com>

ЗАДАЧА 1. Программная Q -система для исследования ресурса параллелизма численных алгоритмов

Q -система позволяет:

- 1) оценить характеристики ресурса параллелизма любого численного алгоритма \mathcal{A} —
высоту $D_{\mathcal{A}}(\bar{N})$ и ширину $P_{\mathcal{A}}(\bar{N})$;
- 2) из множества численных алгоритмов, решающих одну и ту же алгоритмическую проблему, выбрать алгоритм с лучшим ресурсом параллелизма.

ЗАДАЧА 1. Программная Q -система для исследования ресурса параллелизма численных алгоритмов

Q -система состоит из двух подсистем:

1. для формирования Q -детерминантов алгоритмов,
2. для вычисления ресурса параллелизма алгоритмов.

ЗАДАЧА 2. Проектирование программ, использующих ресурс параллелизма численных алгоритмов полностью

- ▶ Модель концепции Q -детерминанта дает возможность исследовать все машинно-независимые свойства численных алгоритмов, однако не учитывает особенности их выполнения на параллельных вычислительных системах.
- ▶ В связи с этим модель была расширена путем добавления двух подмоделей, которые представляют собой модели параллельных вычислений *PRAM* и *BSP*, отражающие в абстрактной форме архитектуру параллельных вычислительных систем с общей и с распределенной памятью соответственно.

ЗАДАЧА 2. Проектирование программ, использующих ресурс параллелизма численных алгоритмов полностью

На основе расширенной модели был предложен метод проектирования параллельной программы для выполнения Q -эффективной реализации численного алгоритма.

Метод состоит из трех этапов:

1. Построение Q -детерминанта алгоритма.
2. Описание Q -эффективной реализации алгоритма.
3. Если Q -эффективная реализация выполнима, то по ее описанию разрабатывается параллельная программа.

ЗАДАЧА 2. Проектирование программ, использующих ресурс параллелизма численных алгоритмов полностью

- ▶ Полученная с помощью метода программа называется *Q-эффективной*, а процесс ее создания *Q-эффективным программированием*.
- ▶ *Q-эффективная* программа полностью использует ресурс параллелизма алгоритма, так как выполняет его *Q-эффективную* реализацию.
- ▶ Таким образом, *Q-эффективная* программа имеет самый высокий параллелизм среди программ, реализующих алгоритм.

ЗАДАЧА 2. Проектирование программ, использующих ресурс параллелизма численных алгоритмов полностью

Метод был апробирован на алгоритмах, Q -детерминанты которых содержат Q -термы различного типа, в их числе:

- ▶ алгоритм умножения плотных матриц (безусловные Q -термы),
- ▶ алгоритм умножения разреженных матриц (условные Q -термы),
- ▶ метод Гаусса–Жордана для решения СЛАУ (условные Q -термы),
- ▶ метод Якоби для решения СЛАУ (условные бесконечные Q -термы),
- ▶ метод Гаусса–Зейделя для решения СЛАУ (условные бесконечные Q -термы).

ЗАДАЧА 2. Проектирование программ, использующих ресурс параллелизма численных алгоритмов полностью

- ▶ Для перечисленных алгоритмов и других студентами ЮУрГУ разработаны *Q*-эффективные программы для общей и распределенной памяти.
- ▶ Использовался язык программирования C++.
- ▶ Для общей памяти применялась технология OpenMP, для распределенной MPI и OpenMP.
- ▶ Исследование проводилось на суперкомпьютере «Торнадо ЮУрГУ».
- ▶ Программы для общей памяти выполнялись на одном процессорном узле, для распределенной памяти на нескольких процессорных узлах.

ЗАДАЧА 3. Повышение эффективности реализации алгоритмических проблем

Покажем, как описанные результаты могут применяться для повышения эффективности реализации алгоритмических проблем.

- ▶ Предположим, что для решения алгоритмической проблемы могут использоваться численные методы $M_i (i = 1, \dots, k)$, а для реализации метода M_i применяются численные алгоритмы $A_j^i (j = 1, \dots, l)$.
 - ▶ Тогда технология повышения эффективности реализации алгоритмической проблемы состоит в следующем.
- 1) С помощью Q -системы для каждого метода $M_i (i = 1, \dots, k)$ находится алгоритм $A_{j_i}^i$ с минимальным значением высоты $D(\bar{N})$.
 - 2) С помощью Q -системы из алгоритмов $A_{j_i}^i (i = 1, \dots, k)$ выбирается алгоритм $A_{j_{i_0}}^{i_0}$ с минимальным значением высоты $D(\bar{N})$,
 - 3) С помощью метода проектирования Q -эффективных программ разрабатывается Q -эффективная программа для алгоритма $A_{j_{i_0}}^{i_0}$.

Описанная технология названа Q -эффективным программированием в широком смысле.

ЗАДАЧА 4. Создание единой программной системы для эффективной реализации численных алгоритмов

- ▶ Концепция Q -детерминанта позволяет автоматизировать проектирование эффективных программ.
- ▶ Поэтому проводятся исследования по созданию единой программной системы для проектирования и исполнения эффективных программ, реализующих численные алгоритмы.
- ▶ Созданы две версии прототипа системы, использующие разные подходы:
 - 1) на основе компилятора и виртуальной машины;
 - 2) на основе лексического анализатора.
- ▶ Проведено экспериментальное исследование версий прототипа программной системы.

ЗАДАЧА 4. Создание единой программной системы для эффективной реализации численных алгоритмов

Что общего у версий прототипа?

- 1) Используют Q -детерминанты алгоритмов, сформированные из блок-схем алгоритмов с помощью первой подсистемы Q -системы.
- 2) Проектируют Q -эффективные программы и исполняют их.
- 3) Тестирование проведено на алгоритмах, исследованных с помощью Q -системы.

ЗАДАЧА 4. Создание единой программной системы для эффективной реализации численных алгоритмов

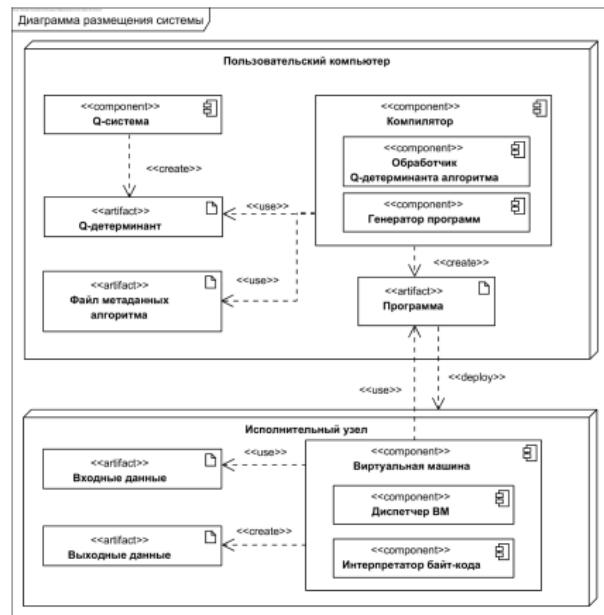


Рис.1. Диаграмма размещения компонентов прототипа системы для эффективной реализации численных алгоритмов на основе компилятора и виртуальной машины

ЗАДАЧА 4. Создание единой программной системы для эффективной реализации численных алгоритмов

Особенности прототипа на основе компилятора и виртуальной машины:

- 1) В настоящее время прототип предназначен для вычислительных систем с общей памятью. Проводится его модификация для возможности использования на распределенной памяти.
- 2) Компилятор преобразует представление алгоритма в форме Q -детерминанта в файл программы.
 - ▶ Подкомпонент компилятора «Обработчик Q -детерминанта алгоритма» строит в памяти структуру, соответствующую Q -эффективной реализации алгоритма.
 - ▶ Получаемая компилятором на основе этой структуры очередь команд записывается в файл программы в бинарном виде.
- 3) Виртуальная машина исполняет программу, созданную компилятором.

Разработка является результатом ВКР магистра Александра Юферова

http://omega.sp.susu.ru/publications/masterthesis/2020_220_yuferovav.pdf

ЗАДАЧА 4. Создание единой программной системы для эффективной реализации численных алгоритмов

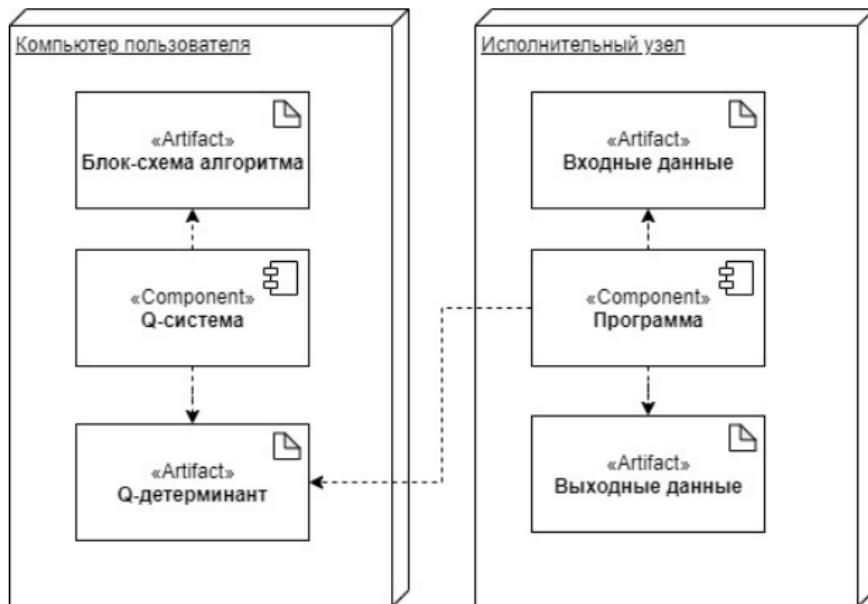


Рис.2. Диаграмма размещения компонентов прототипа системы для эффективной реализации численных алгоритмов на основе лексического анализа

ЗАДАЧА 4. Создание единой программной системы для эффективной реализации численных алгоритмов

Особенности прототипа на основе лексического анализатора:

- 1) Прототип может использовать как общую, так и распределенную память.
- 2) Программа с помощью Q -детерминанта и входных данных алгоритма получает выходные данные алгоритма, используя Q -эффективную реализацию алгоритма.
- 3) Основу программы составляет лексический анализатор, реализованный с использованием стека для хранения промежуточных данных и модели конечного автомата.

Разработка является результатом ВКР бакалавра Алексея Иванова

http://omega.sp.susu.ru/publications/bachelorthesis/2022_404_ivanovad.pdf

ЗАДАЧА 4. Создание единой программной системы для эффективной реализации численных алгоритмов

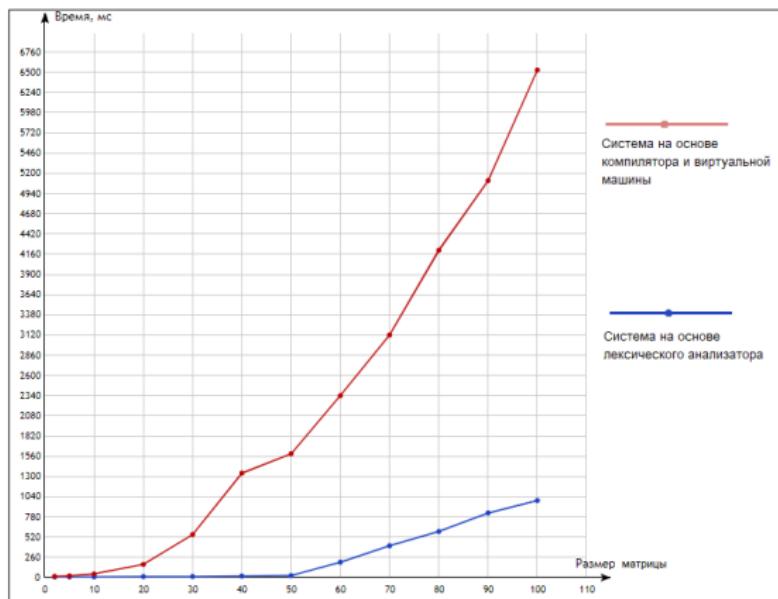


Рис.3. Результаты вычислительного эксперимента для алгоритма умножения матриц без использования схемы сдваивания на суперкомпьютере «Торнадо ЮУрГУ»

ЗАКЛЮЧЕНИЕ

Таким образом, с помощью проведенного исследования доказана возможность создания единой для численных алгоритмов программной системы для проектирования программ, использующих ресурс параллелизма алгоритмов полностью, и их исполнения.

Спасибо за внимание!