

РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ ИССЛЕДОВАНИЯ РЕСУРСА ПАРАЛЛЕЛИЗМА ЧИСЛЕННЫХ АЛГОРИТМОВ С ПОМОЩЬЮ ПРОГРАММНОЙ Q-СИСТЕМЫ

© 2021 В.Н. Алеева, П.С. Зотова, Д.С. Склезнев

Южно-Уральский государственный университет

(454080 Челябинск, пр. им. В.И. Ленина, д. 76)

E-mail: aleevavn@susu.ru, zotovaps2000@gmail.com, skleznevdm@gmail.com

Поступила в редакцию: 16.03.2021

Программная Q -система предназначена для исследования ресурса параллелизма численных алгоритмов. С помощью Q -системы можно вычислить ресурс параллелизма любого численного алгоритма, а также найти алгоритм с лучшим ресурсом параллелизма из множества алгоритмов, решающих алгоритмическую проблему. Теоретической основой Q -системы является концепция Q -детерминанта, где Q — множество операций, используемых алгоритмом. Любой численный алгоритм имеет Q -детерминант и может быть представлен в форме Q -детерминанта. Такое представление является универсальным описанием численных алгоритмов. Q -детерминант состоит из Q -термов. Их число равно числу выходных данных алгоритма. Каждый Q -терм описывает все возможные способы вычисления одного из выходных данных на основе входных данных. Q -детерминант делает алгоритм понятным с точки зрения структуры и реализации. Q -система может быть использована для повышения эффективности параллельных вычислений. В статье описаны новые возможности Q -системы. Новая функциональность позволяет использовать классификацию алгоритмов при их исследовании. Также новая функциональность дает возможность вычислять функции для аппроксимации высоты и ширины алгоритмов более точно и создавать их графическое представление. В результате интерфейс пользователя Q -системы стал удобнее, появилось больше возможностей для исследования ресурса параллелизма алгоритмов.

Ключевые слова: Q-детерминант алгоритма, представление алгоритма в форме Q-детерминанта, Q-эффективная реализация алгоритма, высота алгоритма, ширина алгоритма, ресурс параллелизма алгоритма, программная Q-система.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Алеева В.Н., Зотова П.С., Склезнев Д.С. Расширение возможностей исследования ресурса параллелизма численных алгоритмов с помощью программной Q -системы // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 66–81. DOI: 10.14529/cmse210205.

Введение

Одна из самых актуальных проблем — повышение эффективности параллельных вычислений. Эта проблема тесно связана с проблемой использования ресурса параллелизма алгоритмов. Если иметь возможность выявлять ресурс параллелизма алгоритмов, а затем использовать его полностью при разработке программ, то алгоритмы будут реализовываться эффективно. Более того, если иметь возможность сравнивать ресурсы параллелизма алгоритмов, решающих одну и ту же алгоритмическую проблему, а затем использовать алгоритм с лучшим ресурсом параллелизма, то алгоритмическая проблема будет реализована эффективнее. Программная Q -система, созданная на основе концепции Q -детерминанта, предназначена для вычисления и сравнения ресурсов параллелизма численных алгоритмов, поэтому исследование, связанное с расширением ее возможностей, является актуальным.

Целью исследования является расширение возможностей программной Q -системы для исследования ресурса параллелизма численных алгоритмов. Для достижения цели решаются следующие задачи:

- 1) разработка функциональности Q -системы для использования классификации алгоритмов при их исследовании;
- 2) повышение точности аппроксимации функций высоты и ширины алгоритмов;
- 3) реализация графического представления аппроксимации функций высоты и ширины алгоритмов.

Статья организована следующим образом. Раздел 1 содержит обзор работ по теме исследования. В разделе 2 приводятся основные понятия концепции Q -детерминанта. В разделе 3 рассмотрены теоретические основы и программная реализация Q -системы. В разделе 4 приведено описание новой функциональности Q -системы. Заключение содержит краткое изложение полученных результатов, выводы об их применении и описание дальнейшего направления исследований. Статья продолжает исследования, представленные в [1, 7–10].

1. Обзор работ по теме исследования

Наиболее близким к теме исследования, рассматриваемого в статье, является направление, созданное В.В. Воеводиным. Основополагающими работами этого направления являются [2, 15]. В рамках данного направления проводятся очень важные и развитые исследования параллельной структуры алгоритмов и программ для их реализации на параллельных вычислительных системах (ПВС). Эти исследования применяются для создания открытой энциклопедии свойств алгоритмов AlgoWiki [4, 11]. В AlgoWiki для определения ресурса параллелизма алгоритмов используется индивидуальный подход к каждому алгоритму. Единое программное обеспечение для исследования и сравнения ресурсов параллелизма алгоритмов данное направление исследований в настоящее время не содержит.

Исследований, использующих универсальные подходы к решению проблем, связанных с параллельными вычислениями, очень мало. В основном они посвящены разработке параллельных программ. Приведем примеры таких исследований. Т-система [13] является средой разработки с возможностью автоматического динамического распараллеливания программ. Синтез параллельных программ заключается в разработке новых параллельных алгоритмов с использованием базы знаний параллельных алгоритмов для решения более сложных задач. Технология фрагментированного программирования, язык ее реализации и система программирования LuNA разработаны на основе метода синтеза параллельных программ [5]. Чтобы снять ограничения на ресурсы ПВС, рассматривается проектирование параллельных программ с помощью функциональных языков программирования. Работа [12] является примером такого исследования. Приведенные примеры решают проблему построения параллельных программ, но при этом не исследуется ресурс параллелизма алгоритмов, реализуемых программами, и насколько программы его используют.

подавляющее большинство исследований, связанных с параллельными вычислениями, состоят в разработке параллельных программ, реализующих конкретные алгоритмы или учитывающих особенности архитектуры ПВС. Примерами таких исследований являются [14, 16]. Как правило, разработчики параллельных программ не подбирают для решаемой ими задачи алгоритм с лучшим ресурсом параллелизма, а берут некоторый подходящий алгоритм, выбирают некоторую его реализацию, которая их устраивает, не оценивая, является ли эта реализация эффективной.

Данный обзор основан на доступных авторам источниках и, возможно, не является полным. Однако факт, что повышение эффективности параллельных вычислений является одной из самых актуальных проблем, дает основание заключить, что либо инструменты для автоматизированного исследования ресурса параллелизма алгоритмов разработаны, но не достаточно широко известны, либо они не существуют.

2. Концепция Q -детерминанта

Алгоритмическую проблему опишем в виде $\bar{y} = F(N, B)$, где $N = \{n_1, \dots, n_k\}$ — множество параметров размерности или N — пустое множество, B — множество входных данных, $\bar{y} = \{y_1, \dots, y_m\}$ — множество выходных данных, целое число m является либо константой, либо значением вычисляемой функции параметров N , если $N \neq \emptyset$. Значение n_i ($i \in \{1, \dots, k\}$) равно любому положительному целому числу. Если $N = \{n_1, \dots, n_k\}$, то через $\bar{N} = \{\bar{n}_1, \dots, \bar{n}_k\}$ обозначим набор из k положительных целых чисел, где \bar{n}_i — некоторое заданное значение параметра размерности n_i для каждого $i \in \{1, \dots, k\}$. Через $\{\bar{N}\}$ обозначим множество всех возможных k -наборов \bar{N} . Пусть α — алгоритм для решения алгоритмической проблемы, Q — набор операций, используемых алгоритмом α .

Определение 1. Выражение над B и Q определим, как терм в стандартном смысле математической логики [3]. Каждое выражение w имеет уровень вложенности, обозначим его через T^w .

Пример 1. Выражения и их уровни вложенности:

- 1) $w_1 = b_1 \times (b_2 + b_3)/b_4$, $T^{w_1} = 3$;
- 2) $w_2 = (b_1 \geq b_2) \vee ((b_3 \times b_4) \leq b_5 \times (b_6 + b_7))$, $T^{w_2} = 4$.

Определение 2. Выражение называется цепочкой длины n , если оно является результатом применения некоторой ассоциативной операции из Q к n выражениям.

Определение 3. Если $N = \emptyset$, то любое выражение w над B и Q мы называем безусловным Q -термом. Пусть $N \neq \emptyset$ и V — множество всех выражений над B и Q . Тогда любое отображение $w : \{\bar{N}\} \rightarrow V \cup \emptyset$ также называется безусловным Q -термом.

Пусть $N = \emptyset$ и w — безусловный Q -терм. Предположим, что выражение w над B и Q имеет значение логического типа при любой интерпретации переменных B . Тогда безусловный Q -терм w называется безусловным логическим Q -термом. Пусть $N \neq \emptyset$ и w — безусловный Q -терм. Если выражение $w(\bar{N})$ для каждого $\bar{N} \in \{\bar{N}\}$ имеет значение логического типа при любой интерпретации переменных B , то безусловный Q -терм w называется безусловным логическим Q -термом.

Пусть u_1, \dots, u_l — безусловные логические Q -термы, w_1, \dots, w_l — безусловные Q -термы. Тогда множество l пар $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$ называется условным Q -термом длины l .

Пусть $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i=1,2,\dots}$ — счетное множество пар безусловных Q -термов. Предположим, что $\{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$ — условный Q -терм для любого $l < \infty$. Тогда (\hat{u}, \hat{w}) называется условным бесконечным Q -термом.

Опишем нахождение значения безусловного Q -терма w при интерпретации переменных B . Если $N = \emptyset$, то нахождение значения выражения w означает нахождение значения безусловного Q -терма w при любой интерпретации переменных B . Если $N \neq \emptyset$ и $w(\bar{N}) \neq \emptyset$, то $w(\bar{N})$ является выражением над B и Q . Можно найти значение выражения $w(\bar{N})$ при

любой интерпретации переменных B . Конечно, мы опускаем значение $w(\bar{N}) = \emptyset$. Теперь опишем нахождение значения условного Q -терма (\hat{u}, \hat{w}) при интерпретации переменных B . Пусть $N = \emptyset$. Находим значения выражений u_i, w_i для $i \in \{1, \dots, l\}$. При этом мы можем найти пару u_{i_0}, w_{i_0} такую, что u_{i_0} имеет значение **true**. Следовательно, мы можем найти значение w_{i_0} . Тогда считаем, что (\hat{u}, \hat{w}) имеет значение w_{i_0} . В противном случае считаем, что значение (\hat{u}, \hat{w}) при интерпретации переменных B не определено. Пусть $N \neq \emptyset$ и $\bar{N} \in \{\bar{N}\}$. Находим выражения $u_i(\bar{N}), w_i(\bar{N})$ для $i \in \{1, \dots, l\}$. При этом мы можем найти пару $u_{i_0}(\bar{N}), w_{i_0}(\bar{N})$ такую, что $u_{i_0}(\bar{N})$ имеет значение **true**. Следовательно, мы можем найти значение $w_{i_0}(\bar{N})$. Тогда мы считаем, что (\hat{u}, \hat{w}) имеет значение $w_{i_0}(\bar{N})$. В противном случае считаем, что значение (\hat{u}, \hat{w}) для \bar{N} и при такой интерпретации переменных B не определено. Аналогично можно определить значение условного бесконечного Q -терма.

Определение 4. Пусть $M = \{1, \dots, m\}$ и алгоритм α состоит в нахождении для каждого $i \in M$ значения y_i путем вычисления значения Q -терма f_i . Тогда набор Q -термов $\{f_i \mid i \in M\}$ называется Q -детерминантом алгоритма α . Система уравнений $\{y_i = f_i \mid i \in M\}$ называется представлением алгоритма α в форме Q -детерминанта.

Определение 5. Процесс вычисления Q -термов $\{f_i \mid i \in M\}$ алгоритма α называется реализацией алгоритма α . Если реализация алгоритма α состоит в том, что некоторые операции выполняются одновременно, то она называется параллельной реализацией.

Для дальнейшего изложения будем использовать некоторое разбиение множества M . Предположим, что U, C и I образуют разбиение множества $M = \{1, \dots, m\}$ с пустыми членами, то есть: $U \cup C \cup I = M$; $U \cap C = U \cap I = C \cap I = \emptyset$; кроме того, одно или два подмножества U, C и I могут быть пустыми. Предположим также, что подмножества U, C и I можно связать с подмножествами множества Q -термов $\{f_i \mid i \in M\}$ таким образом:

- 1) для каждого $i \in U$ существует Q -терм f_i , который является безусловным, и $f_i = w^i$;
- 2) для каждого $i \in C$ существует Q -терм f_i , который является условным, и $f_i = \{(u_j^i, w_j^i)\}_{j \in \{1, \dots, l(i)\}}$, где $l(i)$ — либо константа, либо значение вычисляемой функции от N , если $N \neq \emptyset$;
- 3) для каждого $i \in I$ существует Q -терм f_i , который является условным бесконечным, и $f_i = \{(u_j^i, w_j^i)\}_{j \in \{1, 2, \dots\}}$.

Определение 6. Опишем реализацию алгоритма α , называемую Q -эффективной.

Пусть $N = \emptyset$. Зададим интерпретацию переменных B . Будем вычислять выражения

$$W = \{w^i(i \in U); u_j^i, w_j^i(i \in C, j \in \{1, \dots, l(i)\}); u_j^i, w_j^i(i \in I, j \in \{1, 2, \dots\})\} \quad (1)$$

одновременно, параллельно. Мы говорим, что операция готова к выполнению, если вычислены значения всех ее операндов. При вычислении каждого из выражений W (см. (1)) мы выполняем операции, как только они готовы к выполнению. Если несколько операций цепочки готовы к выполнению, то они выполняются по схеме сдваивания. Например, вычисление цепочки $a_1 + a_2 + a_3 + a_4$ по схеме сдваивания выполняется так: сначала вычисляем $b_1 = a_1 + a_2$ и $b_2 = a_3 + a_4$ одновременно, затем $c = b_1 + b_2$. Если для $i \in C \cup I$ и $j \in \{1, 2, \dots\}$ u_j^i имеет значение **false**, то вычисление соответствующего w_j^i прекращается. Если для $i \in C \cup I$ и $j \in \{1, 2, \dots\}$ вычисление пары (u_j^i, w_j^i) приводит к тому, что значение одного выражения не определено, то вычисление второго выражения прекращается. Если

для $i \in C \cup I$ вычисление некоторой пары $(u_{j_0}^i, w_{j_0}^i)$ приводит к определению их значений и значение $u_{j_0}^i$ **true**, то вычисление выражений u_j^i, w_j^i прекращается для любого $j \neq j_0$.

Теперь пусть $N \neq \emptyset$. Зададим интерпретацию переменных B и $\bar{N} \in \{\bar{N}\}$. Получаем множество выражений

$$W(\bar{N}) = \{w^i(\bar{N})(i \in U); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in C, j \in \{1, \dots, l(i)\}); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in I, j \in \{1, 2, \dots\})\}. \quad (2)$$

Выражения $W(\bar{N})$ (см. (2)) вычисляются по аналогии с выражениями W (см. (1)).

Определение 7. Реализация алгоритма α называется выполнимой, если одновременно должно выполняться конечное (непустое) множество операций.

Замечание 1. Есть алгоритмы, Q -эффективная реализация которых невыполнима.

Пример 2. Алгоритм для вычисления суммы S ряда $\sum_{m=1}^{\infty} (-1)^m \frac{1}{m}$ с заданной точностью ϵ имеет следующее представление в форме Q -детерминанта

$$S = \left\{ \left(\frac{1}{2} < \epsilon, -1 \right), \left(\frac{1}{3} < \epsilon, -1 + \frac{1}{2} \right), \dots, \left(\frac{1}{m} < \epsilon, -1 + \frac{1}{2} - \dots + (-1)^{m-1} \frac{1}{m-1} \right), \dots \right\}.$$

Первый уровень вложенности выражений содержит счетное множество операций деления, поэтому Q -эффективная реализация алгоритма не является выполнимой.

Q -эффективная реализация полностью использует ресурс параллелизма алгоритма, который опишем далее. Для этого будем использовать следующие условия и обозначения.

1. Алгоритм α представлен в форме Q -детерминанта $y_i = f_i$ для всех $i \in M$.
2. Значения Q -термов f_i для всех $i \in M$ определяются при любой интерпретации переменных B и для любого $\bar{N} \in \{\bar{N}\}$, если $N \neq \emptyset$.
3. Пусть $N = \emptyset$ и $I \neq \emptyset$. Тогда при заданной интерпретации переменных B для любого $i \in I$ существует пара выражений $u_{j_i}^i, w_{j_i}^i$ такая, что значение $u_{j_i}^i$ равно **true** и значение $w_{j_i}^i$ определено. Введем обозначение

$$\widetilde{W} = \{w^i(i \in U); u_j^i, w_j^i(i \in C, j \in \{1, \dots, l(i)\}); u_{j_i}^i, w_{j_i}^i(i \in I)\}.$$

4. Пусть $N \neq \emptyset$ и $I \neq \emptyset$. Тогда при заданной интерпретации переменных B для любого $\bar{N} \in \{\bar{N}\}$ и $i \in I$ существует пара выражений $u_{j_i}^i(\bar{N}), w_{j_i}^i(\bar{N})$ такая, что значение $u_{j_i}^i(\bar{N})$ равно **true**, а значение $w_{j_i}^i(\bar{N})$ определено. Введем обозначение

$$\widetilde{W}(\bar{N}) = \{w^i(\bar{N})(i \in U); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in C, j \in \{1, \dots, l(i)\}); u_{j_i}^i(\bar{N}), w_{j_i}^i(\bar{N})(i \in I)\}.$$

5. Q -эффективная реализация алгоритма α выполнима.

Введем понятия высоты и ширины алгоритма α , которые характеризуют его ресурс параллелизма.

Определение 8. Обозначим через D_α высоту и через P_α ширину алгоритма α и определим их следующим образом.

Если $N = \emptyset$, то

$$D_\alpha = \begin{cases} \max_{w \in W} T^w & \text{если } I = \emptyset, \\ \max_{w \in \widetilde{W}} T^w & \text{если } I \neq \emptyset; \end{cases} \quad (3)$$

$$P_\alpha = \max_{1 \leq r \leq D_\alpha} \sum_{w \in W} O_r^w, \quad (4)$$

где O_r^w — количество операций уровня вложенности r выражения w .

Если $N \neq \emptyset$, то

$$D_\alpha(\bar{N}) = \begin{cases} \max_{w(\bar{N}) \in W(\bar{N})} T^{w(\bar{N})} & \text{если } I = \emptyset, \\ \max_{w(\bar{N}) \in \widetilde{W}(\bar{N})} T^{w(\bar{N})} & \text{если } I \neq \emptyset; \end{cases} \quad (5)$$

$$P_\alpha(\bar{N}) = \max_{1 \leq r \leq D_\alpha(\bar{N})} \sum_{w(\bar{N}) \in W(\bar{N})} O_r^{w(\bar{N})}, \quad (6)$$

где $O_r^{w(\bar{N})}$ — количество операций уровня вложенности r выражения $w(\bar{N})$.

В формулах (3), (4), (5), (6) обозначения W и $W(\bar{N})$ те же, что в (1) и (2) соответственно. Содержательный смысл формул (3) и (4) следующий. Если параметров размерности нет и Q -детерминант не содержит условные бесконечные Q -термы, то высота алгоритма — это максимальный уровень вложенности выражений W , а ширина алгоритма — максимальное количество операций одного уровня вложенности выражений W . В случае, если параметров размерности нет, но Q -детерминант содержит условные бесконечные Q -термы, то высота алгоритма — это максимальный уровень вложенности выражений \widetilde{W} , а ширина алгоритма — максимальное количество операций одного уровня вложенности выражений W , не превосходящего высоту алгоритма. Содержательный смысл формул (5) и (6) можно описать аналогично. D_α характеризует время выполнения Q -эффективной реализации алгоритма, а P_α — количество процессоров, необходимое для выполнения Q -эффективной реализации алгоритма. Понятия D_α и P_α соотносятся с понятиями высоты и ширины минимальной ярусно-параллельной формы алгоритма.

3. Q -система

3.1. Теоретические основы Q -системы

Теоретические основы Q -системы составляют методы исследования ресурса параллелизма численных алгоритмов, разработанные с помощью концепции Q -детерминанта:

- 1) метод построения Q -детерминанта алгоритма;
- 2) метод получения Q -эффективной реализации алгоритма;
- 3) метод вычисления характеристик ресурса параллелизма алгоритма;
- 4) метод сравнения характеристик ресурса параллелизма алгоритмов.

Опишем эти методы и поясним их на простых примерах.

Q -детерминанты алгоритмов можно построить с помощью блок-схем алгоритмов. При этом блок-схемы используют блоки «терминал», «решение», «процесс», «данные» и имеют ограничения, описанные в работе [9]. Для построения Q -детерминанта алгоритма его блок-схема анализируется при фиксированных значениях параметров размерности алгоритмической проблемы N , если $N \neq \emptyset$.

Если Q -детерминант алгоритма имеет безусловные Q -термы, то проход по блок-схеме идет последовательно в соответствии с выполнением алгоритма. Для каждого $i \in U$ из содержимого блоков «процесс», участвующих в вычислении f_i , формируются выражения w^i , если $N = \emptyset$ и $w^i(\bar{N})$, если $N \neq \emptyset$. Если Q -детерминант содержит условные Q -термы, то блок-схема имеет хотя бы один блок «решение» с условием, содержащим входные данные. Прохождение такого блока порождает ветвление. После прохода по каждой из ветвей для всех $i \in C$ и $j \in \{1, \dots, l(i)\}$ получаются пары выражений (u_j^i, w_j^i) , если $N = \emptyset$ и $(u_j^i(\bar{N}), w_j^i(\bar{N}))$, если $N \neq \emptyset$ для всех условных Q -термов f_i . Выражения u_j^i , если $N = \emptyset$ и $u_j^i(\bar{N})$, если $N \neq \emptyset$ формируются из условий блоков «решение», которые содержат входные данные, а w_j^i , если $N = \emptyset$ и $w_j^i(\bar{N})$, если $N \neq \emptyset$ на основе содержимого блоков «процесс». Условные бесконечные Q -термы содержат Q -детерминанты итерационных алгоритмов. Ограничив количество итераций, можно свести случай Q -детерминанта с условными бесконечными Q -термами к случаю Q -детерминанта с условными Q -термами. Для ограничения количества итераций используется параметр L . Его значение обозначим через \bar{L} . Оно может принимать целые положительные значения. \bar{L} задает длину Q -термов f_i для $i \in I$.

Пример 3. Рассмотрим алгоритм вычисления скалярного произведения векторов $\vec{a}^1 = (a_1^1, \dots, a_n^1)$ и $\vec{a}^2 = (a_1^2, \dots, a_n^2)$, не использующий схему сдваивания. В этом случае $N \neq \emptyset$ и $I = \emptyset$. С помощью метода построения Q -детерминанта для любого $n \in \{2, 3, \dots\}$ может быть получено представление алгоритма в форме Q -детерминанта

$$(\vec{a}^1, \vec{a}^2) = (\dots (a_1^1 \times a_1^2 + a_2^1 \times a_2^2) + a_3^1 \times a_3^2) + \dots + a_n^1 \times a_n^2).$$

В разделе 2 определено, что при Q -эффективной реализации алгоритма выполняется одновременное вычисление выражений W (см. (1)), если $N = \emptyset$ и выражений $W(\bar{N})$ (см. (2)), если $N \neq \emptyset$, при этом операции выполняются по мере вычисления их операндов. Если $N = \emptyset$ и $I \neq \emptyset$, то вместо выражений W будем использовать выражения

$$W(\bar{L}) = \{w^i(i \in U); u_j^i, w_j^i(i \in C, j \in \{1, \dots, l(i)\}); u_j^i, w_j^i(i \in I, j \in \{1, \dots, \bar{L}\})\}.$$

Если $N \neq \emptyset$ и $I \neq \emptyset$, то вместо выражений $W(\bar{N})$ будем использовать выражения

$$W(\bar{N}, \bar{L}) = \{w^i(\bar{N})(i \in U); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in C, j \in \{1, \dots, l(i)\}); u_j^i(\bar{N}), w_j^i(\bar{N})(i \in I, j \in \{1, \dots, \bar{L}\})\}.$$

Метод получения Q -эффективной реализации алгоритма заключается в вычислении уровней вложенности операций, входящих в выражения: W , если $N = \emptyset$ и $I = \emptyset$; $W(\bar{L})$, если $N = \emptyset$ и $I \neq \emptyset$; $W(\bar{N})$, если $N \neq \emptyset$ и $I = \emptyset$; $W(\bar{N}, \bar{L})$, если $N \neq \emptyset$ и $I \neq \emptyset$.

Пример 4. Для алгоритма из примера 3 используем множество $W(\bar{N})$, состоящее из выражения $(\dots (a_1^1 \times a_1^2 + a_2^1 \times a_2^2) + a_3^1 \times a_3^2) + \dots + a_n^1 \times a_n^2$. Операции умножения имеют уровень вложенности 1, уровень вложенности i -й ($i \in \{1, \dots, n-1\}$) операции сложения равен $i+1$.

Вычисление характеристик ресурса параллелизма алгоритма α выполняется по следующим правилам. Если $N = \emptyset$ и $I = \emptyset$, то высота и ширина алгоритма α не зависят от параметров N и L . Высота обозначается D_α и вычисляется по формуле (3). Ширина обозначается P_α и вычисляется по формуле (4). Если $N = \emptyset$ и $I \neq \emptyset$, то высота и ширина алгоритма α зависят от параметра L . Значение высоты алгоритма обозначается $D_\alpha(\bar{L})$ и вычисляется

по формуле $D_\alpha(\bar{L}) = \max_{w \in W(\bar{L})} T^w$. Значение ширины алгоритма обозначается $P_\alpha(\bar{L})$ и вычисляется по формуле $P_\alpha(\bar{L}) = \max_{1 \leq r \leq D_\alpha(\bar{L})} \sum_{w \in W(\bar{L})} O_r^w$. Если $N \neq \emptyset$ и $I = \emptyset$, то высота и ширина алгоритма α зависят от параметров N . Значение высоты алгоритма обозначается $D_\alpha(\bar{N})$ и вычисляется по формуле (5). Значение ширины алгоритма обозначается $P_\alpha(\bar{N})$ и вычисляется по формуле (6). Если $N \neq \emptyset$ и $I \neq \emptyset$, то высота и ширина алгоритма α зависят от параметров N и L . Значение высоты алгоритма обозначается $D_\alpha(\bar{N}, \bar{L})$ и вычисляется по формуле $D_\alpha(\bar{N}, \bar{L}) = \max_{w(\bar{N}) \in W(\bar{N}, \bar{L})} T^{w(\bar{N})}$. Значение ширины алгоритма обозначается $P_\alpha(\bar{N}, \bar{L})$ и вычисляется по формуле $P_\alpha(\bar{N}, \bar{L}) = \max_{1 \leq r \leq D_\alpha(\bar{N}, \bar{L})} \sum_{w(\bar{N}) \in W(\bar{N}, \bar{L})} O_r^{w(\bar{N})}$.

Пример 5. Высота алгоритма из примера 3 должна вычисляться по формуле (5), а ширина по формуле (6), следовательно, высота и ширина равны n .

Допустим, что алгоритмы α и β решают одну и ту же алгоритмическую проблему. Метод сравнения характеристик ресурса параллелизма алгоритмов сравнивает высоту и ширину алгоритмов α и β , если они удовлетворяют одному из четырех наборов условий.

1. Пусть $N = \emptyset$ и $I = \emptyset$. Тогда D_α , P_α , D_β и P_β определены.
2. Пусть $N = \emptyset$ и $I \neq \emptyset$. Тогда
 - 1) $D_\alpha(\bar{L})$ и $P_\alpha(\bar{L})$ определены, если $\bar{L} \in V_\alpha(L)$;
 - 2) $D_\beta(\bar{L})$ и $P_\beta(\bar{L})$ определены, если $\bar{L} \in V_\beta(L)$;
 - 3) $V_\alpha(L) \cap V_\beta(L) \neq \emptyset$.
3. Пусть $N \neq \emptyset$ и $I = \emptyset$. Тогда
 - 1) $D_\alpha(\bar{N})$ и $P_\alpha(\bar{N})$ определены, если $\bar{N} \in V_\alpha(N)$;
 - 2) $D_\beta(\bar{N})$ и $P_\beta(\bar{N})$ определены, если $\bar{N} \in V_\beta(N)$;
 - 3) $V_\alpha(N) \cap V_\beta(N) \neq \emptyset$.
4. Пусть $N \neq \emptyset$ и $I \neq \emptyset$. Тогда
 - 1) $D_\alpha(\bar{N}, \bar{L})$ и $P_\alpha(\bar{N}, \bar{L})$ определены, если $(\bar{N}, \bar{L}) \in V_\alpha(N, L)$;
 - 2) $D_\beta(\bar{N}, \bar{L})$ и $P_\beta(\bar{N}, \bar{L})$ определены, если $(\bar{N}, \bar{L}) \in V_\beta(N, L)$;
 - 3) $V_\alpha(N, L) \cap V_\beta(N, L) \neq \emptyset$.

Пример 6. Поясним метод сравнения характеристик ресурса параллелизма алгоритмов на примере алгоритма вычисления скалярного произведения векторов, не использующего схему сдваивания (см. пример 3), и алгоритма вычисления скалярного произведения векторов $\vec{a}^1 = (a_1^1, \dots, a_n^1)$ и $\vec{a}^2 = (a_1^2, \dots, a_n^2)$, в котором схема сдваивания применяется. В качестве алгоритма α используем первый алгоритм, а в качестве алгоритма β второй. С помощью метода построения Q -детерминанта для любого значения $n \in \{2, 3, \dots\}$ может быть получено представление алгоритма β в форме Q -детерминанта $(\vec{a}^1, \vec{a}^2) = a_1^1 \times a_1^2 + a_2^1 \times a_2^2 + \dots + a_n^1 \times a_n^2$. Так как алгоритм β использует схему сдваивания, то $D_\beta = 1 + \lceil \log_2 n \rceil$, $P_\beta = n$. Пусть высота и ширина алгоритма α определены для значений размерности множества $V_\alpha(N) = \{2, 3, 4, 5, 9, 15, 20, 50\}$, а высота и ширина алгоритма β для значений размерности множества $V_\beta(N) = \{2, 4, 5, 7, 9, 20, 35, 50\}$. Тогда $V_\alpha(N) \cap V_\beta(N) = \{2, 4, 5, 9, 20, 50\} \neq \emptyset$. Алгоритмы α и β удовлетворяют набору условий 3, поэтому к ним можно применить метод сравнения.

С помощью метода сравнения вычисляются значения ΔD и ΔP по правилам.

1. Если $N = \emptyset$ и $I = \emptyset$, то $\Delta D = D_\alpha - D_\beta$, $\Delta P = P_\alpha - P_\beta$.

2. Если $N = \emptyset$ и $I \neq \emptyset$, то

$$\Delta D = \sum_{\bar{L} \in (V_\alpha(L) \cap V_\beta(L))} (D_\alpha(\bar{L}) - D_\beta(\bar{L})), \quad \Delta P = \sum_{\bar{L} \in (V_\alpha(L) \cap V_\beta(L))} (P_\alpha(\bar{L}) - P_\beta(\bar{L})).$$

3. Если $N \neq \emptyset$ и $I = \emptyset$, то

$$\Delta D = \sum_{\bar{N} \in (V_\alpha(N) \cap V_\beta(N))} (D_\alpha(\bar{N}) - D_\beta(\bar{N})), \quad \Delta P = \sum_{\bar{N} \in (V_\alpha(N) \cap V_\beta(N))} (P_\alpha(\bar{N}) - P_\beta(\bar{N})).$$

4. Если $N \neq \emptyset$ и $I \neq \emptyset$, то

$$\Delta D = \sum_{(\bar{N}, \bar{L}) \in (V_\alpha(N, L) \cap V_\beta(N, L))} (D_\alpha(\bar{N}, \bar{L}) - D_\beta(\bar{N}, \bar{L})),$$

$$\Delta P = \sum_{(\bar{N}, \bar{L}) \in (V_\alpha(N, L) \cap V_\beta(N, L))} (P_\alpha(\bar{N}, \bar{L}) - P_\beta(\bar{N}, \bar{L})).$$

По результатам вычислений делаются следующие выводы. Если $\Delta D < 0$, то высота алгоритма α меньше высоты алгоритма β . Если $\Delta D = 0$, то алгоритмы имеют одинаковую высоту. Если $\Delta D > 0$, то высота алгоритма α больше высоты алгоритма β . Если $\Delta P < 0$, то ширина алгоритма α меньше ширины алгоритма β . Если $\Delta P = 0$, то алгоритмы имеют одинаковую ширину. Если $\Delta P > 0$, то ширина алгоритма α больше ширины алгоритма β .

Пример 7. Применим метод сравнения для алгоритмов α и β из примера 6. Используя правило 3, получим $\Delta D = 63$, $\Delta P = 0$. Следовательно, можно сделать вывод, что высота алгоритма α больше высоты алгоритма β , а ширина алгоритмов одинакова.

3.2. Программная реализация Q-системы

Q-система состоит из двух подсистем: для создания Q-детерминантов и для вычисления ресурса параллелизма численных алгоритмов. Первая подсистема реализует метод построения Q-детерминанта алгоритма на основе его блок-схемы. Во второй подсистеме реализованы остальные методы исследования ресурса параллелизма численных алгоритмов. Диаграмма развертывания Q-системы представлена на рис. 1.

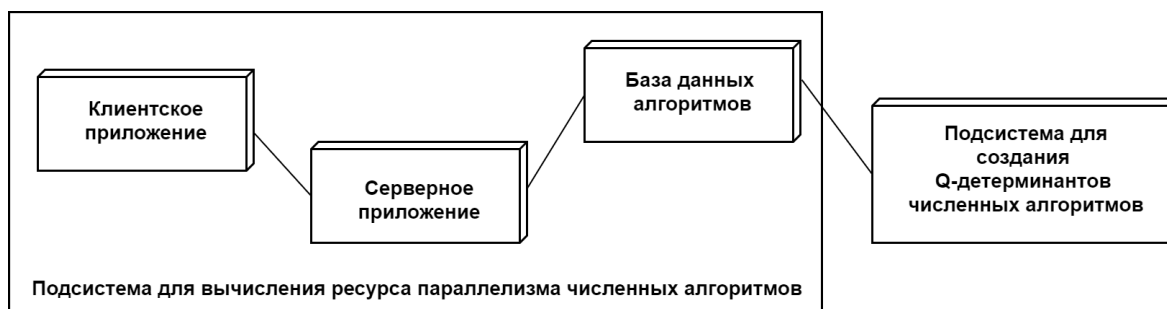


Рис. 1. Диаграмма развертывания Q-системы

Подсистема для создания Q-детерминантов численных алгоритмов является .NET-приложением, разработанным на объектно-ориентированном языке программирования C#. Формат обмена данными JSON применяется для описания входных и выходных данных подсистемы. Формат описания и примеры блок-схем алгоритмов и выходных файлов приведены в работах [7–9].

Подсистема для вычисления ресурса параллелизма численных алгоритмов включает базу данных, серверное и клиентское приложения. Для разработки использовалась СУБД **PostgreSQL**. База данных состоит из таблиц **Algorithms** и **Determinants**. Таблица **Algorithms** имеет атрибуты: идентификатор, имя, описание алгоритма, количество Q -детерминантов алгоритма, загруженных в базу данных. Атрибуты таблицы **Determinants**: идентификатор Q -детерминанта, идентификатор алгоритма, значения параметров размерности N , если $N \neq \emptyset$, иначе 0, Q -детерминант, значение D_α , значение P_α , значение параметра L , если $I \neq \emptyset$, иначе 0. Серверное приложение реализует методы: запись нового алгоритма, обновление информации об алгоритме, получение списка алгоритмов с полной информацией о них, сравнение характеристик ресурса параллелизма алгоритмов, удаление алгоритма и его Q -детерминантов, загрузка Q -детерминанта и характеристик ресурса параллелизма, получение списка Q -детерминантов, скачивание Q -детерминанта, удаление Q -детерминанта. На главной странице клиентского приложения находится список алгоритмов, записанных в базу данных. Также на ней показано количество Q -детерминантов, загруженных в базу данных для каждого алгоритма, расположены элементы интерфейса, которые позволяют любому пользователю сравнить алгоритмы, перейти на страницу со списком Q -детерминантов, загруженных в базу данных для выбранного алгоритма, аппроксимировать функции $D(\bar{N}, \bar{L})$ и $P(\bar{N}, \bar{L})$. Страница со списком Q -детерминантов содержит для каждого Q -детерминанта значения параметров размерности \bar{N} , количество итераций \bar{L} , значения $D(\bar{N}, \bar{L})$ и $P(\bar{N}, \bar{L})$. Интерфейс для авторизованных пользователей обеспечивает добавление, редактирование и удаление алгоритмов, загрузку и удаление Q -детерминантов.

Пример 8. В настоящее время в Q -системе описано десять специально подобранных алгоритмов с различными структурами Q -детерминантов, в том числе алгоритмы α и β (см. пример 6), используемых для тестирования функциональности Q -системы. Для алгоритмов α и β в базу данных загружено соответственно 36 и 78 Q -детерминантов. При загрузке каждого Q -детерминанта выполнялось вычисление и запись в базу данных значений высоты и ширины. Результатом сравнения алгоритмов α и β будет вывод, приведенный в примере 7.

4. Расширение функциональности Q -системы

4.1. Обзор алгоритмов в соответствии с их классификацией

В процессе эксплуатации Q -системы количество алгоритмов в базе данных может расти. В этом случае использование общего списка алгоритмов будет не всегда удобным, поэтому был реализован способ обзора алгоритмов в соответствии с их классификацией. Возможность обзора алгоритмов в виде общего списка сохранена.

Примером использования при исследовании алгоритмов их классификации является открытая энциклопедия свойств алгоритмов AlgoWiki [4]. Под классификацией алгоритмов в нашем исследовании мы понимаем следующий принцип систематизации алгоритмов. Исследуемые алгоритмы делятся на классы, классы могут иметь подклассы первого уровня, которые могут иметь подклассы второго уровня и так далее. Количество классов, подклассов и уровней подклассов не ограничено. Один и тот же алгоритм может принадлежать нескольким классам или подклассам. Критерии распределения алгоритмов по классам и подклассам должны определять их исследователи — пользователи Q -системы.

Программное обеспечение, разработанное для обзора алгоритмов в соответствии с их классификацией, позволяет создать для классификатора систему папок и подпапок, соот-

ветствующих классам и подклассам классификатора. Структуры строк, соответствующих алгоритмам, при обзоре на основе иерархической классификации и с помощью общего списка одинаковы. Таким образом, при обзоре алгоритмов в соответствии с классификацией для работы с алгоритмами используется та же функциональность Q -системы, что и при обзоре алгоритмов в виде общего списка. Например, внутри любого класса или подкласса алгоритмов можно проводить сравнение характеристик ресурса параллелизма двух алгоритмов, переходить на страницу для аппроксимации функций высоты и ширины, на страницу с Q -детерминантами выбранного алгоритма. Авторизованный пользователь имеет право производить операции добавления, редактирования и удаления элементов классификации алгоритмов, записывать новые алгоритмы, редактировать и удалять существующие.

Пример 9. Для алгоритмов, описанных в Q -системе, был создан классификатор и осуществлена его программная поддержка. В результате обзор алгоритмов можно осуществлять с использованием следующей системы папок и подпапок.

1. Папка «**1. Решение СЛАУ**» содержит две подпапки:
 - подпапка «**1.1. Прямые методы решения СЛАУ**» содержит метод Гаусса—Жордана с выбором в качестве ведущего первого ненулевого элемента ведущей строки;
 - подпапка «**1.2. Итерационные методы решения СЛАУ**» содержит методы Гаусса—Зейделя и Якоби.
2. Папка «**2. Умножение матриц**» содержит два алгоритма: не использующий и использующий схему сдваивания.
3. Папка «**3. Скалярное произведение векторов**» содержит два алгоритма: не использующий и использующий схему сдваивания.
4. Папка «**4. Вычисление максимального элемента в массиве**» содержит алгоритм поиска максимального элемента в массиве чисел.
5. Папка «**5. Решение квадратных уравнений**» содержит два алгоритма: без использования операции унарного минуса и с использованием операции унарного минуса.

4.2. Аппроксимация характеристик ресурса параллелизма алгоритмов и ее графическое представление

Характеристики ресурса параллелизма алгоритмов $D(\bar{N}, \bar{L})$ и $P(\bar{N}, \bar{L})$ хранятся в базе данных в табличном виде, то есть их значения известны только для некоторых наборов значений параметров размерности и количества итераций. Вместе с тем при исследовании ресурса параллелизма алгоритмов важно иметь возможность вычислять характеристики ресурса параллелизма при любых значениях \bar{N} и \bar{L} . Поэтому важно решить задачу аппроксимации, а точнее, интерполяции и экстраполяции функций высоты $D(\bar{N}, \bar{L})$ и ширины $P(\bar{N}, \bar{L})$. Для наглядности целесообразно использовать кроме аналитического представления аппроксимирующей функции ее графическое представление.

Мы разработали новую функциональность Q -системы для аппроксимации характеристик ресурса параллелизма алгоритмов $D(\bar{N}, \bar{L})$ и $P(\bar{N}, \bar{L})$ и графического представления аппроксимирующих функций. При описании и использовании данной функциональности параметры размерности и количества итераций обозначаются через $\{x_i\}_{i=1, \dots, k}$. Сначала перечисляются $k - 1$ параметров размерности в том порядке, в каком они хранятся в базе данных, переменная x_k обозначает количество итераций. Если $\bar{L} = 0$, то x_k не используется.

Ранее в составе Q -системы было разработано программное обеспечение для аппроксимации характеристик ресурса параллелизма алгоритмов с помощью полиномиальных

функций, зависящих от $\{x_i\}_{i=1,\dots,k}$. При этом степень аппроксимирующей полиномиальной функции вводилась пользователем по запросу системы. Это программное обеспечение не обеспечивало достаточную точность аппроксимации при больших значениях параметров в случае, если характеристика ресурса параллелизма зависела от $\log_2(x_i)$, где $x_i \in \{x_i\}_{i=1,\dots,k}$. Графическое представление аппроксимирующих функций реализовано не было.

Для реализации новой функциональности Q -системы для аппроксимации характеристик ресурса параллелизма алгоритмов и графического представления аппроксимирующих функций было разработано программное обеспечение, которое для любого алгоритма, исследуемого с помощью Q -системы, решает следующие задачи.

1. Вычисление аппроксимирующих полиномиальных функций для высоты $D(\bar{N}, \bar{L})$ и ширины $P(\bar{N}, \bar{L})$ с помощью метода наименьших квадратов и минимизации невязки. В процессе вычисления подбирается степень и тип полинома. Выбор типа аппроксимирующей полиномиальной функции производится между полиномиальной функцией, зависящей от $\{x_i\}_{i=1,\dots,k}$, и полиномиальной функцией, зависящей от $\{\log_2(x_i)\}_{i=1,\dots,k}$.
2. Создание графического представления аппроксимирующих функций для высоты и ширины при условии, что они зависят не более чем от двух параметров $\{x_i\}_{i=1,\dots,k}$. Графическое представление содержит график аппроксимирующей функции, дополненный изображениями точек из базы данных.
3. Расчет значений аппроксимирующих функций для любых значений параметров $\{x_i\}_{i=1,\dots,k}$, заданных пользователем.

Для аппроксимации функций высоты и ширины алгоритма пользователь должен выбрать алгоритм из списка и кликнуть на элементе со знаком суммы Σ . В результате будет выведена страница с аппроксимацией для выбранного алгоритма, содержащая аппроксимирующие функции и их графические представления, если они были созданы. На этой же странице пользователь может задать значения параметров $\{x_i\}_{i=1,\dots,k}$ и получить вычисленные с помощью аппроксимирующих функций значения высоты и ширины алгоритма.

На рис. 2 и рис. 3 показано графическое представление аппроксимирующих функций высоты и ширины алгоритма β (см. пример 6) и метода Гаусса—Жордана с выбором в качестве ведущего первого ненулевого элемента ведущей строки. Оценка $3x_1$ высоты метода Гаусса—Жордана, где x_1 — размер матрицы СЛАУ, ранее была доказана в [1] математически. Рисунки 2 и 3 получены с помощью Q -системы.

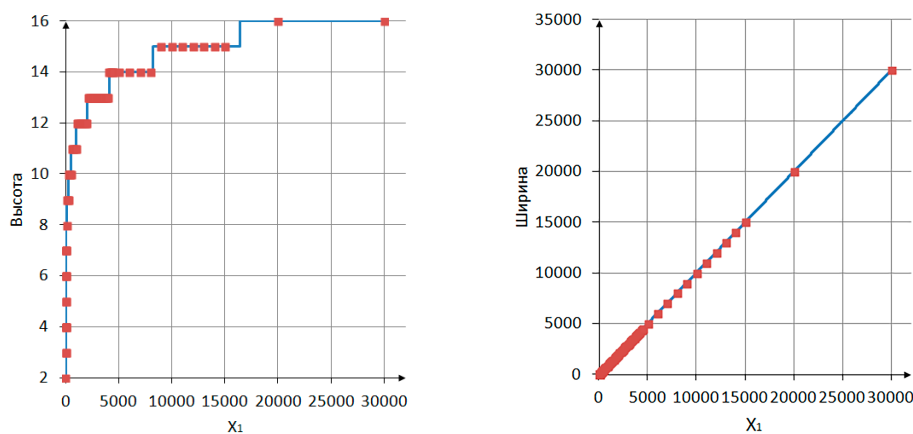


Рис. 2. Графическое представление аппроксимирующих функций высоты и ширины алгоритма вычисления скалярного произведения векторов

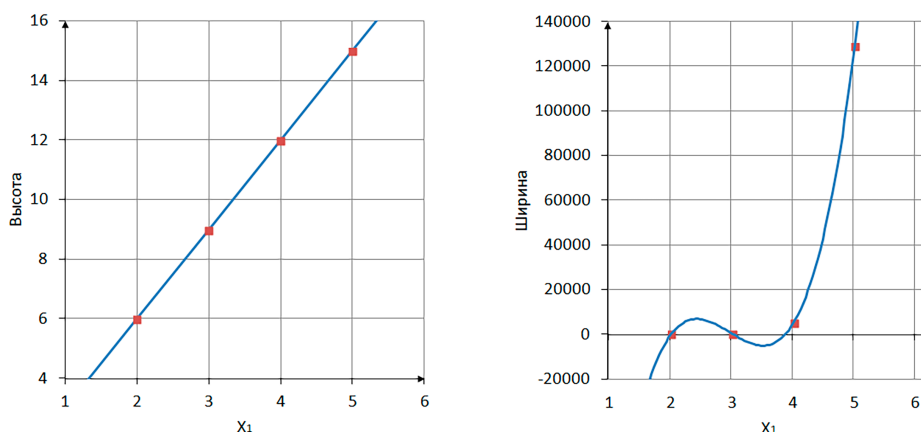


Рис. 3. Графическое представление аппроксимирующих функций высоты и ширины метода Гаусса–Жордана

Заключение

В статье описано расширение возможностей программной Q -системы для исследования ресурса параллелизма численных алгоритмов в результате решения следующих задач:

- 1) разработана функциональность Q -системы для использования классификации алгоритмов при их исследовании;
- 2) повышена точность аппроксимации функций высоты и ширины алгоритмов;
- 3) реализовано графическое представление аппроксимации функций высоты и ширины алгоритмов.

Полученные результаты дают пользователям больше возможностей для исследования ресурса параллелизма алгоритмов, а также делают интерфейс пользователя более удобным. Q -система, метод проектирования Q -эффективных программ [6] и технология Q -эффективного программирования [7] в совокупности решают проблему повышения эффективности параллельных вычислений. Планируется дальнейшее расширение возможностей Q -системы, а также ее доведение до промышленной эксплуатации. Q -система свободно доступна в сети Интернет по адресу <https://qclient.herokuapp.com>, что позволяет ознакомиться с ее возможностями, для которых авторизация не требуется.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-00865 а и при поддержке Правительства РФ в соответствии с Постановлением № 211 от 16.03.2013 г. (соглашение № 02.А03.21.0011).

Литература

1. Алеева В.Н. Анализ параллельных численных алгоритмов. Препринт № 590. Новосибирск: ВЦ СО АН СССР, 1985. 23 с.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
3. Ершов Ю.Л., Палотин Е.А. Математическая логика. М.: Наука, 1987. 336 с.
4. Открытая энциклопедия свойств алгоритмов. URL: <https://algowiki-project.org/ru> (дата обращения: 06.03.2021).
5. Akhmed-Zaki D., Lebedev D., Malyshev V. et al. Automated Construction of High Performance Distributed Programs in LuNA System // Parallel Computing Technologies (PaCT 2019). Lecture Notes in Computer Science. 2019. Vol. 11657. P. 3–9. DOI: 10.1007/978-3-030-25636-4_1.

6. Aleeva V. Designing a Parallel Programs on the Base of the Conception of Q -Determinant // Supercomputing. RuSCDays 2018. Communications in Computer and Information Science. 2019. Vol. 965. P. 565–577. DOI: 10.1007/978-3-030-05807-4_48.
7. Aleeva V.N. Improving Parallel Computing Efficiency // Proceedings – 2020 Global Smart Industry Conference (GloSIC 2020). IEEE, 2020. P. 113–120. Article number 9267828. DOI: 10.1109/GloSIC50886.2020.9267828.
8. Aleeva V.N., Aleev R.Zh. High-Performance Computing Using Application of Q -determinant of Numerical Algorithms // Proceedings – 2018 Global Smart Industry Conference (GloSIC 2018). IEEE, 2018. 8 p. Article number 8570160. DOI: 10.1109/GloSIC.2018.8570160.
9. Aleeva V., Bogatyreva E., Skleznev A. et al. Software Q -system for the Research of the Resource of Numerical Algorithms Parallelism // Supercomputing. RuSCDays 2019. Communications in Computer and Information Science. 2019. Vol. 1129. P. 641–652. DOI: 10.1007/978-3-030-36592-9_52.
10. Aleeva V.N., Sharabura I.S., Suleymanov D.E. Software System for Maximal Parallelization of Algorithms on the Base of the Conception of Q -determinant // Parallel Computing Technologies (PaCT 2015). Lecture Notes in Computer Science. 2015. Vol. 9251. P. 3–9. DOI: 10.1007/978-3-319-21909-7_1.
11. Antonov A.S., Dongarra J., Voevodin V.V. AlgoWiki Project as an Extension of the Top500 Methodology // Supercomputing Frontiers and Innovations. 2018. Vol. 5, no. 1. P. 4–10. DOI: 10.14529/jsfi180101.
12. Legalov A.I., Vasilyev V.S., Matkovskii I.V. et al. A Toolkit for the Development of Data-Driven Functional Parallel Programmes // Parallel Computational Technologies (PCT'2018). Communications in Computer and Information Science. 2018. Vol. 910. P. 16–30. DOI: 10.1007/978-3-319-99673-8_2.
13. Moskovsky A., Roganov V., Abramov S. Parallelism Granules Aggregation with the T-System // Parallel Computing Technologies (PaCT 2007). Lecture Notes in Computer Science. 2007. Vol. 4671. P. 293–302. DOI: 10.1007/978-3-540-73940-1_30.
14. Rajashri A. Parallelization of shortest path algorithm using OpenMP and MPI // International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2017. P. 304–309. DOI: 10.1109/I-SMAC.2017.8058360.
15. Voevodin V.V., Voevodin V.I. The V-Ray technology of optimizing programs to parallel computers // Numerical Analysis and Its Applications (WNAA 1996). Lecture Notes in Computer Science. 1997. Vol. 1196. P. 546–556. DOI: 10.1007/3-540-62598-4_136.
16. Wang Q., Liu J., Tang X. et al. Accelerating embarrassingly parallel algorithm on Intel MIC // IEEE International Conference on Progress in Informatics and Computing. 2014. P. 213–218. DOI: 10.1109/PIC.2014.6972327.

Алеева Валентина Николаевна, к.ф.-м.н., доцент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Зотова Полина Сергеевна, студент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

Склезнев Дмитрий Сергеевич, студент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

ADVANCEMENT OF RESEARCH FOR THE PARALLELISM RESOURCE OF NUMERICAL ALGORITHMS WITH THE HELP OF SOFTWARE Q -SYSTEM

© 2021 V.N. Aleeva, P.S. Zotova, D.S. Skleznev

South Ural University (pr. Lenina 76, Chelyabinsk, 454080 Russia)

E-mail: aleevavn@susu.ru, zotovaps2000@gmail.com, skleznevdm@gmail.com

Received: 16.03.2021

The software Q -system is designed to study the parallelism resource of numerical algorithms. By use of Q -system, we can calculate the parallelism resource of any numerical algorithm. Also, for the set of algorithms solving a given algorithmic problem, we can find an algorithm with the best parallelism resource. The theoretical basis of the Q -system is the concept of Q -determinant where Q is the set of operations used by the algorithm. Any numerical algorithm has a Q -determinant and can be represented in the form of a Q -determinant. Such a representation is a universal description of numerical algorithms. The Q -determinant consists of Q -terms. Their number is equal to the number of output data items. Each Q -term describes all possible ways to calculate one of the output data items based on the input data. Q -determinant makes the algorithm transparent in terms of structure and implementation. The Q -system can be used for the improvement of parallel computing efficiency. In this paper we describe new features of the Q -system. With the new functionality, we can use the classification of algorithms when studying them. Also with the new functionality, we can calculate the functions for approximating the height and width of the algorithms more precisely and create a graphical representation of them. As a result, we have a more convenient user interface of the Q -system and more opportunities to study the parallelism resource of algorithms.

Keywords: Q -determinant of algorithm, representation of algorithm in form of Q -determinant, Q -effective implementation of algorithm, algorithm height, algorithm width, parallelism resource of algorithm, software Q -system.

FOR CITATION

Aleeva V.N., Zotova P.S., Skleznev D.S. Advancement of Research for the Parallelism Resource of Numerical Algorithms with the Help of Software Q -system. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2021. Vol. 10, no. 2. P. 66–81. (in Russian) DOI: 10.14529/cmse210205.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Aleeva V.N. Analysis of Parallel Numerical Algorithms. Preprint no. 590. Novosibirsk, Computing Center of the Siberian Branch of the Academy of Sciences of the USSR, 1985. 23 p. (in Russian)
2. Voevodin V.V., Voevodin V.I. Parallel Computing. St.Petersburg, BHV-Petersburg, 2002. 608 p. (in Russian)
3. Ershov Yu.L., Palyutin E.A. Mathematical Logic. Moscow, Mir, 1984. 303 p.
4. Open Encyclopedia of Parallel Algorithmic Features. URL: <https://algowiki-project.org/en> (accessed: 06.03.2021).

5. Akhmed-Zaki D., Lebedev D., Malyshkin V. et al. Automated Construction of High Performance Distributed Programs in LuNA System. Parallel Computing Technologies (PaCT 2019). Lecture Notes in Computer Science. 2019. Vol. 11657. P. 3–9. DOI: 10.1007/978-3-030-25636-4_1.
6. Aleeva V. Designing a Parallel Programs on the Base of the Conception of Q -Determinant. Supercomputing. RuSCDays 2018. Communications in Computer and Information Science. 2019. Vol. 965. P. 565–577. DOI: 10.1007/978-3-030-05807-4_48.
7. Aleeva V.N. Improving Parallel Computing Efficiency. Proceedings – 2020 Global Smart Industry Conference (GloSIC 2020). IEEE, 2020. P. 113–120. Article number 9267828. DOI: 10.1109/GloSIC50886.2020.9267828.
8. Aleeva V.N., Alev R.Zh. High-Performance Computing Using Application of Q -determinant of Numerical Algorithms. Proceedings – 2018 Global Smart Industry Conference (GloSIC 2018). IEEE, 2018. 8 p. Article number 8570160. DOI: 10.1109/GloSIC.2018.8570160.
9. Aleeva V., Bogatyreva E., Skleznev A. et al. Software Q -system for the Research of the Resource of Numerical Algorithms Parallelism. Supercomputing. RuSCDays 2019. Communications in Computer and Information Science. 2019. Vol. 1129. P. 641–652. DOI: 10.1007/978-3-030-36592-9_52.
10. Aleeva V.N., Sharabura I.S., Suleymanov D.E. Software System for Maximal Parallelization of Algorithms on the Base of the Conception of Q -determinant. Parallel Computing Technologies (PaCT 2015). Lecture Notes in Computer Science. 2015. Vol. 9251. P. 3–9. DOI: 10.1007/978-3-319-21909-7_1.
11. Antonov A.S., Dongarra J., Voevodin V.V. AlgoWiki Project as an Extension of the Top500 Methodology. Supercomputing Frontiers and Innovations. 2018. Vol. 5, no. 1. P. 4–10. DOI: 10.14529/jsfi180101.
12. Legalov A.I., Vasilyev V.S., Matkovskii I.V. et al. A Toolkit for the Development of Data-Driven Functional Parallel Programmes. Parallel Computational Technologies (PCT'2018). Communications in Computer and Information Science. 2018. Vol. 910. P. 16–30. DOI: 10.1007/978-3-319-99673-8_2.
13. Moskovsky A., Roganov V., Abramov S. Parallelism Granules Aggregation with the T-System. Parallel Computing Technologies (PaCT 2007). Lecture Notes in Computer Science. 2007. Vol. 4671. P. 293–302. DOI: 10.1007/978-3-540-73940-1_30.
14. Rajashri A. Parallelization of shortest path algorithm using OpenMP and MPI. International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2017. P. 304–309. DOI: 10.1109/I-SMAC.2017.8058360.
15. Voevodin V.V., Voevodin V.I. The V-Ray technology of optimizing programs to parallel computers. Numerical Analysis and Its Applications (WNAA 1996). Lecture Notes in Computer Science. 1997. Vol. 1196. P. 546–556. DOI: 10.1007/3-540-62598-4_136.
16. Wang Q., Liu J., Tang X. et al. Accelerating embarrassingly parallel algorithm on Intel MIC. IEEE International Conference on Progress in Informatics and Computing. 2014. P. 213–218. DOI: 10.1109/PIC.2014.6972327.