

ВИЗОГРАФ: ОБ ОДНОМ ПОДХОДЕ К ОПИСАНИЮ ДАННЫХ И ГЕНЕРАЦИИ СХЕМ

В.П. Гергель, Е.А. Козин

Нижегородский госуниверситет им.Н.И. Лобачевского

E-mail: Evgeniy.Kozinov@gmail.com

Введение

На протяжении последних лет одной из наиболее острых проблем эффективного использования потенциала компьютерных технологий является трудоемкость разработки программного обеспечения. Проблема обостряется, когда необходимо разработать программное обеспечение для параллельных вычислительных систем. Для уменьшения сложности применяются разные подходы. Среди популярных подходов можно выделить разработку библиотек параллельных методов и системных библиотек для параллельного программирования, расширение языков программирования, добавление функциональности в компиляторы для автоматического распараллеливания, а также создание инструментов для параллельного программирования.

Данная работа посвящена одному из активно развиваемых направлений, уменьшающих сложность, – созданию сред визуального проектирования и анализа параллельных программ. Статья является продолжением серии работ [1-4] и посвящена проблеме описания данных в визуальном языке, а также подходу к автоматической генерации схем параллельных алгоритмов.

Постановка задачи

В работе [4] представлено описание разрабатываемого визуального языка параллельного программирования. На рис. 1 представлен пример визуальной схемы для параллельного суммирования вектора. Схема состоит из трех *модулей* – «Генерация вектора», «Суммирование элементов вектора» и «Вывод суммы». Модули обеспечивают представление отдельных блоков параллельной программы. У модулей окружностями изображены *параметры*. Параметры используются для визуального представления входных и выходных данных модулей. Для установки соответствия (связывания) между параметрами модулей применяются *линкеры*. Линкеры изображены стрелками. Для представления операций по разделению и объединению данных в системе Визограф вводятся визуальные элементы *разделитель* (сплиттер) и *агрегатор* данных. На схеме данные элементы изображены в блоках 1 и 2 соответственно. В [4] были введены основные обозначения, необходимые для описания схем алгоритмов, но не рассматривался вопрос описания данных.

В данной работе предложено визуальное представление данных, позволяющее осуществлять логический и синтаксический контроль в визуальных схемах. Описание должно быть достаточно полным, чтобы впоследствии можно было автоматически генерировать схемы для новых алгоритмов.

Описание типов данных

В разрабатываемой визуальной системе Визограф для описания данных предполагается введение нескольких базовых типов данных, например, скаляр, матрица, вектор

и др. В визуальных схемах тип данных обозначается прямоугольником. Внутри прямоугольника присутствует минимум два элемента – тип элементов внутри базового типа данных и пиктограмма, визуально отображающая рассматриваемый тип. На рис. 2 блок 1 представляет базовый тип матрицы вещественного типа.

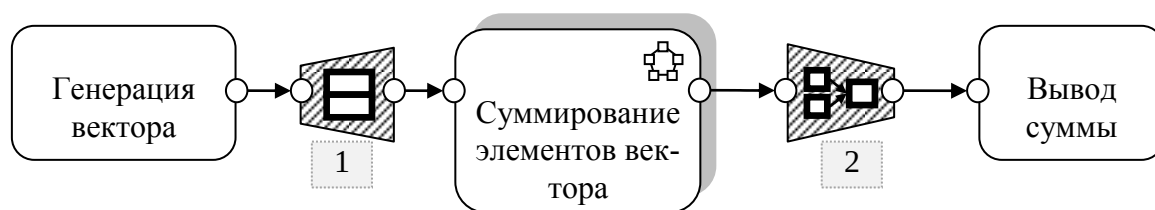


Рис.1. Пример визуальной схемы

Для повышения информативности типов данных и усиления контроля, предполагается введение производных типов данных. В производных типах данных пользователи могут добавлять дополнительные свойства. Каждое свойство добавляет ограничение на данные. Свойства могут быть взяты как из списка заранее реализованных свойств в системе, так и введены пользователем. Если свойство введено пользователем, то код проверки выполнимости свойства пользователю придется разрабатывать самостоятельно. Пользовательские свойства необходимы для расширения возможностей системы Визограф. В типах данных свойства отображаются дополнительным списком. Примеры производных типов данных от базового типа матрицы показаны на рис. 2 блоки 1 и 2.

Предполагается возможность получения производного типа как агрегацию свойств нескольких производных типов одного базового типа. При агрегации новый тип данных содержит объединение списков свойств базовых типов. Например, производный тип данных матричного типа, представленный на рис. 2 блок 4, получен путем агрегации производных типов.

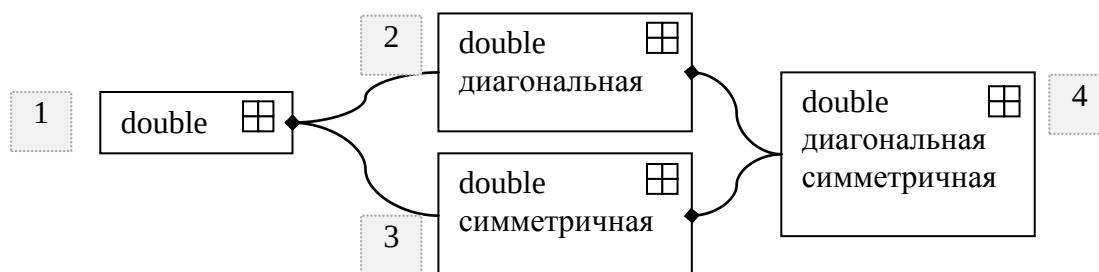


Рис.2. Пример визуального представления матричных типов данных

При описании визуальной системы Визограф введено понятие модуля. Модули позволяют описать часть алгоритма, имеют входные и выходные параметры, соединив которые линкерами, можно получить описание разрабатываемого алгоритма. Подробности синтаксиса можно посмотреть в [4]. В разрабатываемой визуальной системе предполагается, что все модули, используемые для описания алгоритмов, должны быть зарегистрированы. При регистрации, каждому параметру ставится в соответствие тип данных. После регистрации, при соединении модулей линкерами проверяется синтаксическая правильность – нельзя соединять параметры, имеющие разные типы данных. В момент выполнения алгоритма, при каждой передаче данных через параметры, проверяется выполнимость свойств.

Кроме синтаксической корректности схемы, необходимо так же контролировать логическую правильность. Для обеспечения логической корректности схемы разраба-

тываемого алгоритма предполагается введение специального вида свойств – дескрипторов. Все дескрипторы создаются пользователями Визограф и являются глобальными для системы. При регистрации модуля типы данных могут быть связаны с дескрипторами. Дескрипторы, как и свойства, отображаются в списке свойств, но после разделяющей черты. На рис. 3 блок 1 демонстрирует связи параметра с типом данных, содержащим один дескриптор.

В некоторых случаях модули, реализованные пользователем, могут обрабатывать данные, различающиеся с логической точки зрения. Для описания подобных ситуаций предусмотрена возможность использования дескрипторов как самостоятельных элементов без типов данных. Для связи дескриптора с типом данных в этом случае используются линкеры. Если дескрипторы связан с типом данных через линкер, то его можно добавить к списку дескрипторов связанного типа данных. Как и для модулей, для самостоятельных дескрипторов можно создавать зависимость. На рис. 3 представлен пример зарегистрированного модуля, который может перемножить матрицу на вектор. Также модуль может перемножить ленту матрицы на вектор, но при этом в качестве результата в выходном параметре будет только часть перемножаемого вектора.

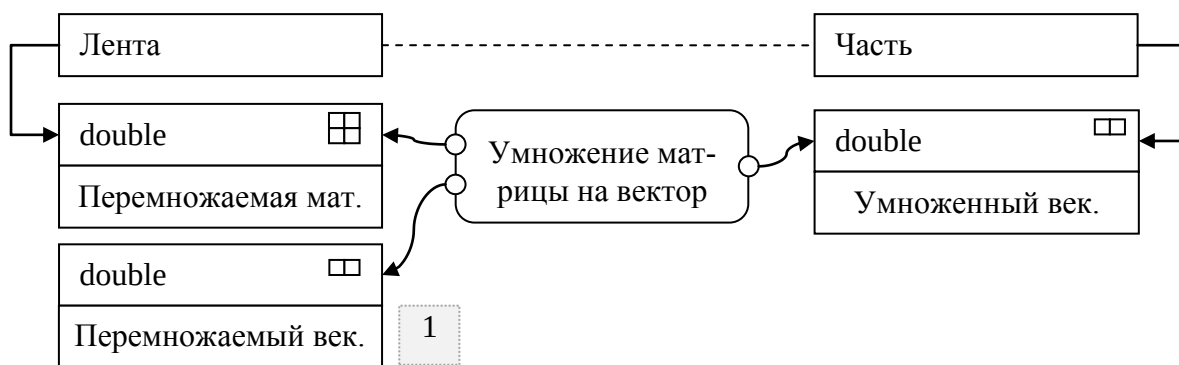


Рис.3. Пример регистрации модуля в системе

Если при регистрации модулей у типов данных были определены дескрипторы, то в этом случае при соединении параметров модулей контролируется логическая корректность схемы – типы данных и набор дескрипторов должны совпадать.

Автоматическая генерация схем

В системе Визограф помимо модулей могут быть созданы и объекты. Объекты всегда создаются на основе типов данных и связаны с источником данных. В качестве источника данных может выступать, например, файл. Между объектами, как и между модулями, могут быть установлены зависимости. На рис. 4 представлено несколько созданных объектов – «Матрица А», «Вектор b» и «Вектор c».

Используя логические связи между модулями, задаваемые дескрипторами и свойствами, можно построить сеть логически правильных схем. Предполагается, что система Визограф может использовать построенную сеть для автоматической генерации алгоритмов для объектов, созданных пользователем. Предполагается следующая последовательность действий для пользователя.

1. Создать две группы объектов – группа входных объектов и объекты, которые хотелось бы получить.
2. Выполнить связывание типов данных с необходимыми дескрипторами, чтобы система Визограф могла логически проинтерпретировать созданные объекты.
3. Выделить требуемые объекты данных и запросить построение алгоритма.

Если Визограф сможет подобрать алгоритм, то создается макромодуль с соответствующей последовательностью модулей, связанных между собой.

Макромодуль пользователь может раскрыть и при необходимости отредактировать последовательность действий. При редактировании также проверяется логическая и синтаксическая правильность получаемой схемы. Макромодуль автоматически регистрируется в системе. На рис. 4 приведен пример автоматически сгенерированной схемы.

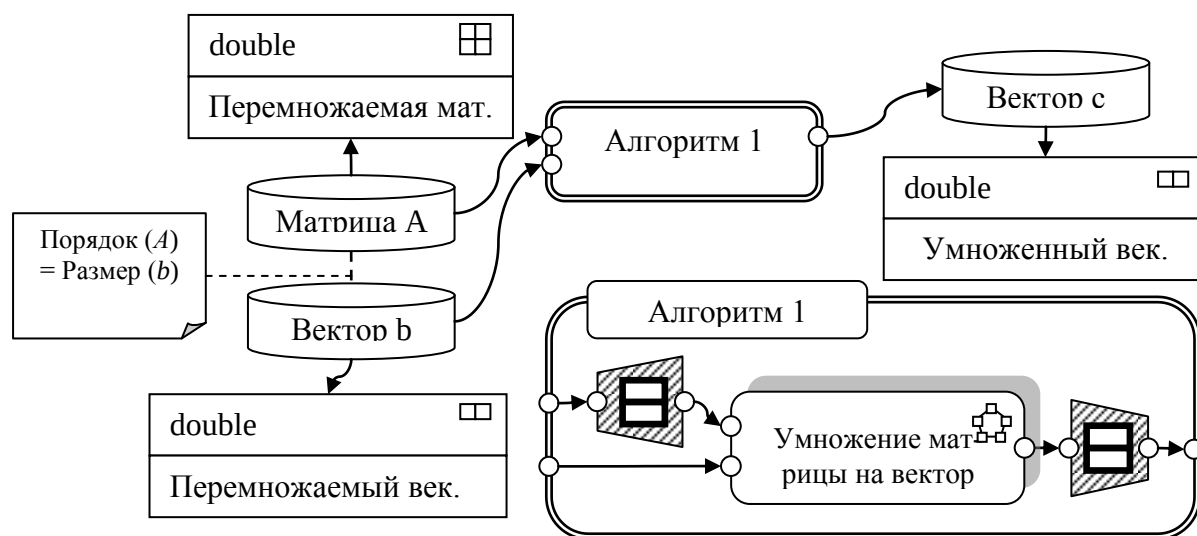


Рис.4. Пример автоматически сгенерированной системы

Заключение

В рамках работы предложен визуальный способ описания данных, позволяющий описывать данные, передаваемые между модулями. При передаче параметров данные контролируются на предмет корректности с точки зрения выполнимости свойств, что исключает возможные проблемы в ходе выполнения алгоритма. Контроль данных позволяет легче понять причину некорректных результатов разрабатываемого алгоритма.

За счет введения в описание типов данных специального вида свойств – дескрипторов, в системе Визограф появилась возможность снизить сложность разработки параллельных алгоритмов за счет автоматической генерации схем.

Важным результатом является то, что разработанное описание позволяет генерировать синтаксически и логически корректные схемы алгоритмов.

Литература

1. Козинев Е.А., Деумени М.-Н.Л. Обзор некоторых визуальных систем для параллельного программирования // Материалы конференции «Технологии Microsoft в теории и практике программирования», Нижегородский госуниверситет им. Н.И. Лобачевского, 11-12 марта 2009. – С. 198-204.
2. Козинев Е.А. Графический язык параллельного программирования // Материалы конференции «Технологии Microsoft в теории и практике программирования», ННГУ им. Лобачевского, 13-14 мая 2010 г. – С. 16-19.
3. Козинев Е.А. Различные подходы к визуальному представлению параллельных программ // Материалы X международной конференции «Высокопроизводительные параллельные вычисления на кластерных системах». Пермский гос. тех. университет 1-3 ноября 2010.

4. Гегель В.П., Козин Е.А. Об одном подходе к визуальной разработке параллельных программ. Направлена в печать (Вестник ННГУ).