



Проблемы параллельных вычислений

Рассматриваются проблемы параллельных вычислений в аспекте развития сложных вычислительных систем. Рассматривается проблема оценки производительности вычислительной системы в широком и узком смысле. Описано влияние семантического разрыва на качество параллельных алгоритмов. Рассмотрены методы формализованного построения структур вычислительных систем. Дается анализ различных архитектур вычислительных систем.

Ключевые слова: информация, философия информации, параллельные вычисления, сложные вычислительные системы, структуры алгоритмов



Problems of parallel computing systems

The article explores the challenges of parallel computing in terms of the development of complex computer systems. The paper describes the problem of assessing the performance of a computer system in the wide and narrow sense. The article describes the effect of the semantic gap in the quality of parallel algorithms. This article describes methods of building formal structures of computing systems. The article analyzes the different architectures of computer systems.

Keywords: Information, philosophy of information, parallel computing, complex computer systems, structures of algorithms

Введение

Возрастающие требования к скорости вычислений и ограниченность последовательных систем обработки информации, обуславливают необходимость применения параллельных вычислительных систем. Например, решение задач при обработке космических изображений [1] требует от современных вычислительных систем производительности от 25 млрд. до 1000 трлн. операций/секунду [2]. Существуют противоречия между усложнением создаваемых систем и традиционными подходами к их проектированию. Для вычислительных систем (ВС) приемлем аппарат проектирования сложных систем [3]. Сложной вычислительной системой (СВС) называют вычислительную систему, закон функционирова-

ния которой допускает ее декомпозицию на отдельные составляющие, сохраняющие свойство внутренней эмерджентности. Отличие структуры СВС от структуры типовой сложной системы заключается в гетерогенности типов связей, элементов [4] и распределении функций, что подчеркивает гетерогенность СВС в целом. Функциональными особенностями СВС являются параллелизм, недетерминированность, наличие информационных взаимодействий [5], сочетание синхронного и асинхронного управления и др. Поэтому фундаментальная проблема параллельных вычислений заключается в формализации параллельных алгоритмов и структур параллельных вычислительных систем. Проблемы анализа и синтеза структур и алгоритмов СВС в полном объеме не решена и требует исследований.

Некоторые тенденции в развитии вычислительных систем

Недостатки архитектуры фон Неймана заключаются не только в ограниченной производительности последовательных вычислений. Существует семантический разрыв [5] между архитектурой ЭВМ и средой, в которую ее помещают для использования. Эта проблема приводит к высокой стоимости разработки программного обеспечения, его ненадежности, и эксплуатационной неэффективности и др. [6].

Семантический разрыв оказывает влияние на интерпретацию понятия «производительность ВС или СВС», которое понимают по-разному. Основная причина разной интерпретации производительности заключается в том, что ее можно определять в широком и узком понимании. В широком смысле при решении задач существует многоуровневая логическая цепочка [7] действий, которая включает человеческий фактор, как компонент общего решения.

«1. ПОСТАНОВКА ЗАДАЧИ → 2. ФОРМАЛИЗАЦИЯ ЗАДАЧИ → 3. НАПИСАНИЕ АЛГОРИТМА → 4. ВЕРИФИКАЦИЯ АЛГОРИТМА → 5. ВЫЧИСЛЕНИЕ → 6. РЕШЕНИЕ → 7. АНАЛИЗ РЕШЕНИЯ»

Многие разработчики, используя традиционный подход к проектированию "снизу-вверх" [8] и стремясь минимизировать стоимость аппаратных средств, концентрируют свое внимание только на уровне (вычисление 5.) и возлагают на плечи аналитиков и программистов решение всех остальных трудных проблем.

В качестве примера подобного несоответствия возьмем данные из работы [9]. Какая ВС имеет более высокую производительность: система, выполняющая программу X за 1 минуту, но требующая 10 человеко-недель для программирования, или система Y, выполняющая ту же программу X за 2 минуты при затратах на программирование 2 человеко-недели?

С позиций микроуровня быстродействие первой системы по сравнению со второй в два раза выше, с позиций макроуровня - в пять раз ниже. Если же исходить из известных статистических данных о том, что 85% всех программ выполняется только один раз, то такая характеристика архитектуры ЭВМ, как простота программирования, является не менее важной, чем скорость выполнения команд.

В ряде работ, например [8-10], показано, что традиционная технология проектирования универсальных ЭВМ снизу-вверх неэффективна при разработке параллельных ВС. Кроме того, показано, что на заданном наборе алгоритмов максимальная производительность может быть гарантирована только в проблемно-ориентированной ВС при проектировании ее сверху-вниз. Доказано, что для эффективного проектирова-

ния параллельных ВС необходима единая форма описания алгоритмов и структур ВС. Поэтому в настоящее время применяют специальные подходы к построению параллельных СВС.

Локальность связей. Традиционный способ построения параллельных ВС и языков программирования не подходит для системы обработки данных в реальном масштабе времени из-за большого числа вспомогательных операций, связанных с распределением вычислительных ресурсов и препятствующих повышению производительности [2]. Поэтому в многопроцессорных системах стали использовать локальные связи, что уменьшает взаимную зависимость процессоров и задержек, вносимых длинными соединениями. Такой подход привел к разработке нескольких специализированных устройств обработки данных с локальными связями и с архитектурой, ориентированной на исполнение в виде СБИС: трансверсальная фильтрация, столбчатые матрицы, потоковые процессорные матрицы [8, 9]. Использование локальных связей существенно упрощает процесс проектирования архитектуры ВС и взаимодействия между ее элементами. Однако данное преимущество достигается за счет ограничения области применения ВС.

Программируемая архитектура и динамическая реконфигурация вычислительных систем. Из многочисленных архитектур построения супер ЭВМ необходимо отметить ВС с перестраиваемой архитектурой [2]. Эта технология опирается на "бесшовную" архитектуру, позволяющую плавно наращивать вычислительную мощность системы для точного соответствия требованиям заказчика. Эта архитектура позволяет создавать семейства CS (Computing Surface - вычислительная среда) в любой конфигурации из готовых процессорных и периферийных схемных плат, благодаря чему, сокращаются сроки проектирования разработки, изготовления и поставок.

Многопроцессорные системы (МПС) с перестраиваемой архитектурой разделяют на два основных класса: МПС с программируемой архитектурой; МПС с динамической реконфигурацией. В МПС с программируемой архитектурой [11] нужная конфигурация определяется программным путем.

Под динамической реконфигурацией МПС [12] понимается способность системы к динамическому перераспределению параллельных ресурсов каждого типа между задачами и их ветвями. Данное перераспределение осуществляется автоматически путем анализа процесса выполнения программ аппаратными средствами в соответствии с текущими, заранее непредсказуемыми требованиями задач на вычислительные ресурсы.

Особую группу реконфигурируемых систем образуют ВС с параллельной архитектурой, построенные на основе клеточных автоматов и искусственных нейронных сетей [13]. Клеточные

автоматы рассматриваются как сложные динамические системы самоорганизующиеся из достаточно простых дискретных компонентов и характеризующиеся дискретностью, однородностью структуры, детерминированностью. Основным преимуществом искусственных нейронных сетей является свойство адаптивного самообучения.

Пространственно-временное распределение данных. В каждом алгоритме для параллельной ВС важно определить момент времени, когда данные достигают процессорного элемента (ПЭ). В программе, написанной на обычном языке высокого уровня, нет взаимодействия пространства и времени. Иначе обстоит дело при разработке систем на процессорных матрицах. Данные, которые расположены далеко от ПЭ, требуют больших энергетических затрат и имеют большее время доступа. Поэтому алгоритм, считающийся хорошим для традиционного языка программирования, не обязательно хорош, когда в расчет принимаются явления, происходящие на физическом уровне. Таким образом, при проектировании параллельных систем на матричных структурах необходимо учитывать пространственное и временное распределение данных.

Декомпозиция вычислительных структур и данных. При обработке данных на параллельных ВС часто возникают вопросы отображения вычислительных задач большой размерности на ограниченные структуры ПЭ. Поэтому необходимо проектировать только такие специализированные устройства, которые могут быть использованы для решения задачи произвольного размера и делать это не хуже, чем программные средства универсальных компьютеров. Для решения данных вопросов часто используются аддитивное разбиение и тензорное исчисление [14].

Параллельное представление алгоритмов. Многие важные алгоритмы обработки данных имеют очень специализированные структуры или совсем не структурированы. Вопросы реализации таких алгоритмов на параллельных и конвейерных устройствах достаточно сложны. Поэтому для достижения желаемой скорости обработки необходимы средства, позволяющие максимально использовать возможности параллелизма и конвейерной организации. Иначе говоря, задача нахождения отношения параллельности реализуемых алгоритмов на множестве ПЭ весьма актуальна.

Аппаратные средства параллельной обработки. Из числа новых разработок элементной базы для высокопроизводительных ЭВМ следует отметить создание СБИС центральных процессоров с высоким быстродействием, создание высокопроизводительных МПС с ограниченным набором команд (RISC-процессоров), реализация параллельных процессоров в виде одной СБИС и появление транспьютеров - нового мощного направления развития элементной базы.

Современный уровень развития технологий БИС, используемых при построении суперкомпьютеров, уже сейчас позволяет получать СБИС с периодом тактовых импульсов системы 2 нс и производительность от 20 до 25 млрд.оп./с.

Термин "транспьютер" в настоящее время используется в двух значениях: 1) как наименование программируемых СБИС, разработанных фирмой INMOS и предназначенных в качестве вычислительных модулей для создания сверхпроизводительных параллельных систем; 2) как наименование любых подобных транспьютерам вычислительных модулей. Основное назначение транспьютеров заключается в построении параллельных вычислительных систем высокой производительности. Транспьютерную систему можно использовать для организации конвейерных вычислений, систолической обработки, векторной и матричной обработки, обработки методом потока данных.

Обобщая результаты анализа основных тенденций в развитии современных ВС, следует отметить, что наиболее перспективным способом достижения высокой производительности является построение ВС на основе большого числа однородных ЭВМ. Данные системы привлекательны из-за локальности линий связи, что обеспечивает большую пропускную способность. Стремление эффективно решать задачи на таких системах должно обязательно сопровождаться согласованием структуры численных методов и архитектуры ВС.

Методы формализованного описания структур вычислительных систем

Целью формализованного описания структур ВС является представление имеющихся данных и параллельных процессов в виде специальных формальных моделей, удобных для проведения вычислительных и имитационных экспериментов на компьютере. Поэтому выбор формализованного языка, в наибольшей степени учитывающего особенности параллельных ВС, является основной задачей начального этапа проектирования.

Решение таких задач связано с применением специальных методов построения синхронных и асинхронных моделей дискретных систем. Среди этих методов наибольшую известность получили: методы, использующие методы систем массового обслуживания; методы автоматного и агрегативного моделирования; методы расширения языков программирования [15]; методы структурных нотаций [16]; сетевые и алгебраические методы [17]; методы использующие графовые модели [18].

Подход, предложенный в работе [10] для формализации анализа сложной системы и состоящий из трех шагов: структуризации объекта,

формализации элементов сложной системы и взаимодействия между этими объектами, обладает существенным недостатком. Данный недостаток заключается в том, что такая формализация элементов системы и взаимодействия между ними обладает "неформульным" заданием схемы сопряжения (в виде рисунков) и операторов сопряжения (в виде таблиц).

Подобная формализация не представляет возможность формализовать область эквивалентных структурных преобразований схемы сопряжения. Этим недостатком лишена формализация элементов сложной системы и взаимодействия между ними с помощью так называемых R-модулей [19], которые используются не только при описании детерминированных динамических систем, функционирующих в дискретном времени, но и при описании стохастических систем, представляемых вероятностными автоматами. Однако, подобные автоматные модели не перекрывают все возникающие задачи.

Использование параллельных языков программирования и структурных нотаций для формализованного описания структур ВС эффективно лишь для анализа одной структуры вычислительной системы.

Это обусловлено тем, что данные описания громоздки и не приспособлены для поддержки процедур синтеза новых структур ВС. Если ВС рассматривать как множество взаимодействующих функциональных блоков (объектов), а не как вычислительную сеть или многопроцессорную систему, то для исследования процессов, протекающих в ВС, может быть использован сетевой подход [18]. В работе [20] для описания параллельных процессов с синхронизацией описаны предложены OS-сети. Потребляемые ресурсы в этой модели явно не задаются, повторно используемые ресурсы используются самые простые. Предложенные OS-сети используются, в основном, для анализа тупиковых ситуаций, которые могут возникнуть в параллельных ВС.

Эффективным средством анализа и синтеза параллельных ВС и процессов является алгебраический подход [17], который основан на формульном выражении сетевых моделей. Использование алгебраического подхода позволяет аналитическими методами путем проведения эквивалентных преобразований формул получать оптимальные структуры ВС [20]. Недостатками данного подхода являются: во-первых, ограниченность, т.к. не все сетевые структуры могут быть описаны алгебраически; во-вторых, сложность, т.к. процесс построения алгебраического описания сложной сетевой структуры требует определенных навыков и умения.

В последнее время во многих работах отмечается тот факт, что графовые модели являются наиболее удобными и эффективными средствами описания и исследования параллельных структур и процессов. К настоящему времени су-

ществует несколько формализмов, основанных на графовых моделях и служащих для описания параллельных процессов. Наиболее общими из них являются [18]:

- схемы параллельных программ Карпа-Милнера;
- А-программы Котова-Нариньяни;
- билогические графы;
- вычислительные модели;
- операторы Хоара.

К концу 70-х годов указанные модели были практически вытеснены сетями Петри (СП) [18, 21] - формализмом, описывающим структуру и взаимодействие параллельных процессов. Широкое распространение СП обусловлено рядом преимуществ, среди которых можно выделить следующие.

1. СП позволяют моделировать асинхронность и недетерминизм параллельных независимых процессов, параллелизм конвейерного типа, конфликтные взаимодействия между процессами.

2. СП включают в себя возможности ряда других моделей, предложенных для описания и исследования параллельных ВС (семафоры Дейкстры, системы векторного сложения, вычислительные сети, сетевые структуры, модели повторно используемых ресурсов и др.).

3. СП, расширенные такими обобщениями, как ингибиторные дуги, приоритетность и время срабатывания переходов, цветные метки и др., позволяют моделировать сложные ВС с учетом таких факторов, как приоритетность процессов, временные параметры событий, совместное отображение структуры управления и потоков данных.

4. В отличие от других формализмов (таких, как А-программы, схемы Карпа-Милнера и др.) СП допускают произвольную интерпретацию элементов модели как в смысле выполняемого фрагмента (выражения, операторы, подпрограммы, аппаратные преобразования информации), так и по уровню абстракции. Это позволяет с помощью СП производить иерархическое построение аппаратных и программных модулей ВС.

5. Эффективность представления знаний в ЭС. Современные языки программирования имеют существенные ограничения, которые связаны с выполнением действий в определенном порядке. Исследование в этой области, направленные на устранение указанного ограничения путем ввода новых примитивов, привели к использованию СП. Между СП и представлением знаний в ЭС существует глубокая связь. В частности, предикатные СП являются продукционными системами, основанными на логике первого порядка. Кроме того, модульность системы правил является существенным показателем производительности ЭС. С этой точки зрения СП имеют преимущества, так как активизация правил, представленных в терминах СП, происходит

ассоциативным образом, а не в порядке, строго заданном процедурой.

6. СП, обладая однородностью и аналитическими зависимостями, которые описывают функционирование переходов, удовлетворяют необходимым условиям для использования в тензорной методологии.

7. Для СП, которые являются двудольным ориентированным динамическим помеченным мультиграфом, справедливы все положения теории графов.

Заключение

На основе проведенного исследования проблем анализа и синтеза структур параллельных ВС можно сделать следующие выводы.

1. Большинство современных вычислительных структур характеризуются такими свойствами, как параллелизм, недетерминированность, многоуровневость представления, сочетание синхронных и асинхронных процессов, однородность и др. Эффективное решение задач на таких структурах должно обязательно сопровождаться согласованием структуры численных методов и архитектуры ВС.

2. При решении проблемы синтеза эффективных вычислительных структур основополагающим вопросом является выбор математического фундамента, на котором может строиться изучение таких разнородных компонент, как численные методы, алгоритмы, структуры ВС и их

математические модели. В качестве подобного математического фундамента предлагается использовать тензорное исчисление.

3. Из проведенного анализа следует, что реализация современных проектов ВС должна вестись при поддержке эффективных средств автоматизации проектирования, моделирования и верификации. В силу NP-сложности задач синтеза альтернативных вариантов проектируемой ВС, автоматизация структурного и параметрического синтеза является трудно реализуемой даже с использованием САПР, реализованных на высокопроизводительных ЭВМ. Поэтому большинство из существующих подходов к решению данной задачи носят эвристический характер и предусматривают включение в контур машинного проектирования человека. На основе этого можно заключить, что современные САПР параллельных ВС должны иметь интеллектуальную составляющую.

4. Анализ методов формализованного описания современных структур ВС и их особенностей, параллельных процессов и алгоритмов, интеллектуальных САПР позволил остановиться на аппарате СП, который в рамках единого формализма не только дает возможность описывать указанные разнообразие объекты, но, кроме того, предоставляет развитые методы анализа параллельных процессов и, обладая однородностью и аналитическими зависимостями, удовлетворяет условиям для использования в тензорной методологии.

ЛИТЕРАТУРА

1. Бондур В.Г. Современные подходы к обработке больших потоков гиперспектральной и многоспектральной аэрокосмической информации // Исследование Земли и космоса. 2014. № 1. С. 4-17.
2. Суперкомпьютерные технологии в науке, образовании и промышленности / Под редакцией: академика В. А. Садовниченко, академика Г. И. Савина, чл.-корр. РАН Вл. В. Воеводина. М.: Издательство Московского университета, 2009. 232 с.
3. Монахов С.В., Савиних В.П., Цветков В.Я. Методология анализа и проектирования сложных информационных систем. М.: Просвещение, 2005. 264 с.
4. S. A. Kuja, I. V. Solovjev, V. Ya. Tsvetkov. System Elements Heterogeneity // European Researcher, 2013, Vol.(60), № 10-1, p.2366-2373.
5. Tsvetkov V. Ya. Information Interaction as a Mechanism of Semantic Gap Elimination // European Researcher, 2013, Vol.(45), № 4-1, p.782-786.
6. Dorai C, Venkatesh S. Bridging the Semantic Gap with Computational Media Aesthetics.
7. Цветков В.Я. Логика в науке и методы доказательств. LAP LAMBERT Academic Publishing GmbH & Co. KG, Saarbrücken, Germany 2012. 84 с.
8. Цапко И. В., Цапко С. Г. Архитектура вычислительных систем. Томск, Из-во Томского политехнического университета, 2011. 168 с.
9. Майерс Г. Архитектура современных ЭВМ: В 2-кн. Кн.1. М.: Мир, 1985. 364 с.
10. Букатов А. А., Дацюк В. Н., Жегуло А. И. Программирование многопроцессорных вычислительных систем. Ростов-на-Дону. Издательство ООО «ЦВВР», 2003, 208 с.
11. Митропольский Ю.И. Концепция иерархического построения высокопроизводительных вычислительных систем // Кибернетика и вычислительная техника (Москва). 1991. № 5. С.7-14.
12. Kramer J., Magee J. Dynamic configuration for distributed systems // IEEE Trans. Software Eng., 1985, 11, N4. P.424-436
13. Цой Ю.Р. К применению нейронных сетей для аппроксимации таблицы правил клеточного автомата // Нейроинформатика и ее приложения: Материалы XIV Всероссийского семинара, 6-8 октября 2006 г. / Под ред. А.Н. Горбаня, Е.М. Миркеса. - ИВМ СО РАН, Красноярск, 2006. С. 129-130.
14. Кулагин В.П. Тензорные методы проектирования структур вычислительных систем // АВТ. 1989. № 2. С. 64-71.
15. Армстронг Дж.Р. Моделирование цифровых схем на языке VHDL: концепция моделирования на уровне интегральных схем. М.: Мир, 1992. 174 с.
16. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. Н.Новгород, ННГУ, 2001.
17. Кулагин В.П. Алгебра сетевых моделей для описания параллельных вычислительных систем // Автоматизация и современные технологии. 1993. № 2. С.25-30.
18. Кулагин В.П. Моделирование структур параллельных ВС на основе сетевых моделей. М.: МИЭМ, 1998. 102 с.
19. Иванов П.М. Формализация анализа сложных систем // Кибернетика и системный анализ. 1992. № 4. С. 87-92.
20. Воеводина В.В. Параллельные структуры алгоритмов и программ. М.: ОВМ АН СССР, 1987. 148 с.
21. Кулагин В.П., Цветков В.Я. Философия сетей Петри // Вестник МГТУ МИРЭА «MSTU MIREA HERALD» 2014. № 4 (5). С. 18-38.

Информация об авторе

Кулагин Владимир Петрович
(Россия, Москва)

Профессор, доктор технических наук.
Заведующий кафедрой
Московский институт электроники
и математики НИУ ВШЭ
E-mail: kvp@miem.ru

Information about the author

Kulagin Vladimir Petrovich
(Russia, Moscow)

Professor, doctor of technical Sciences.
Head of the Department
Moscow Institute of electronics
and mathematics HSE
E-mail: kvp@miem.ru