

Технология параллельного решения нелинейных систем булевых уравнений

Г.А. Опарин, В.Г. Богданова, Н.Г. Макеева

Рассматривается технология булева моделирования, архитектура и принципы функционирования инструментальных средств, предназначенных для автоматизации параллельного решения нелинейных систем булевых уравнений.

1. Введение

Хорошо известно, что с точки зрения человеческого восприятия булевы системы имеют сложную специфическую природу. В связи с этим важной и весьма нетривиальной проблемой остается построение средств, позволяющих человеку моделировать задачи на языке булевых ограничений. Такие задачи, как правило, обладают тяжелыми комбинаторными характеристиками с высокой оценкой сложности, что заставляет вести поиск методов, действующих наиболее эффективно для отдельных (интересных с практической точки зрения) классов булевых систем, и одновременно повышать производительность решения систем булевых уравнений путем использования многопроцессорной техники.

В последние годы активно ведутся работы по разработке технологий, методов и средств параллельного проведения ресурсоемких расчетов для различных типов архитектур многопроцессорной техники. Так, в работе [1] приводится обзор методов распараллеливания алгоритмов решения ряда задач вычислительной дискретной математики (в том числе комбинаторных задач) и отмечается, что попытки выполнения обычных последовательных алгоритмов комбинаторных задач на многопроцессорных вычислительных системах во многих случаях не приводят к повышению быстродействия. В работе [2] представлены параллельные методы обхода дерева, ориентированные на использование в многопроцессорных системах кластерного типа, а в [3] рассматриваются параллельные алгоритмы решения комбинаторных задач, в частности, метод линеаризационного множества, использующий параллельный алгоритм обхода дерева по методу выделяемых поддеревьев для решения системы логических уравнений. Для параллельного решения задач булевой выполнимости применяются различные эвристические методы расщепления булевой функции, разработаны параллельные SAT-решатели дискретных задач. В качестве входного языка таких программ используется фрагмент международного формата DIMACS представления булевых ограничений в виде конъюнктивной нормальной формы пропозициональной логики. Следует отметить, что средства автоматизации построения булевых моделей по содержательной (ориентированной на конечного пользователя) постановке задачи в этих программах отсутствуют. В большинстве существующих программ-решателей, использующих метод расщепления [4] для крупноблочного распараллеливания булевых моделей, выбор тактики расщепления зависит от задачи и возлагается на пользователя.

Таким образом, авторам представляется актуальной разработка интегрированной инструментальной среды, использующей технологию синтеза булевых ограничений по содержательному описанию дискретной задачи, технологию распараллеливания решения системы булевых уравнений как на уровне узлов кластера, так и на уровне процессорных ядер, содержащей методы и средства автоматизации представления, накопления, модификации и обработки знаний (в частности, алгоритмы полного поиска) для решения задач в булевых ограничениях и ориентированной на использование в фундаментальных и прикладных исследованиях, где естественным образом возникают дискретные модели в виде систем булевых уравнений.

В докладе представлен являющийся такой средой инструментальный комплекс (ИК) РЕБУС, рассматриваются архитектура этого комплекса и используемые в нем технологии булева моделирования и параллельного решения комбинаторных задач.

2. Постановка задачи

Формально задача решения булевых уравнений формулируется следующим образом: пусть $X = (x_1, \dots, x_n)$ – упорядоченное множество булевых переменных ($x_i \in B_2$ для всех $i = \overline{1, n}$; $B_2 = \{0, 1\}$). Задано m булевых соотношений (система уравнений) вида

$$f_i(x_{i_1}, \dots, x_{i_{k(i)}}) = 0, \quad (1)$$

где f_i ($i = \overline{1, m}$) – произвольные булевы функции своих аргументов. Требуется найти одно, несколько или все решения системы (1) или установить ее несовместность.

Следует заметить, что первая NP-полная задача – задача “выполнимости КНФ” (задача булевой выполнимости или SAT задача) есть, по существу, задача распознавания совместности систем уравнений [5]

$$x_{i_1}^{\sigma_{i_1}} \vee \dots \vee x_{i_{k(i)}}^{\sigma_{i_{k(i)}}} = 1 \quad (i = \overline{1, m}), \quad (2)$$

где переменная

$$x_{i_p}^{\sigma_{i_p}} = \begin{cases} \bar{x}_{i_p}, & \sigma_{i_p} = 0, \\ x_{i_p}, & \sigma_{i_p} = 1. \end{cases}, \quad p = \overline{1, k(i)}.$$

Двойственная к (2) система уравнений вида (1) записывается следующим образом:

$$\bar{x}_{i_1}^{\sigma_{i_1}} \wedge \dots \wedge \bar{x}_{i_{k(i)}}^{\sigma_{i_{k(i)}}} = 0 \quad (i = \overline{1, m}). \quad (3)$$

Известно, что любая система вида (1) может быть представлена одним эквивалентным уравнением $f(x_1, \dots, x_n) = \bigcup_{i=1}^m f_i(x_{i_1}, \dots, x_{i_{k(i)}}) = 0$. Поэтому далее будем рассматривать в основном одно стандартное уравнение

$$f(x_1, \dots, x_n) = 0. \quad (4)$$

В рассматриваемой инструментальной среде алгоритмы решения булева уравнения (4), составляющие ее вычислительное ядро, ориентированы на часто встречающиеся в приложениях так называемые большие разреженные системы булевых уравнений, когда в (1) выполняется условие $m \gg n \gg k(i)$ (символом “ \gg ” обозначено отношение “много больше”). В частности, для многих практических задач $k(i)$ принимает только два фиксированных значения: $k(i) = 2$ и $k(i) = p$ ($p > 2$), причем бинарных ограничений в системе (1) много больше, чем p -арных. Алгоритмы решения систем булевых уравнений ИК РЕБУС разработаны на основе дискретного подхода и относятся к классу полных методов поиска с возвратом. Преимуществом первых двух алгоритмов является то, что они предназначены для решения уравнения (4), левая часть которого $f(x_1, x_2, \dots, x_n)$ представлена булевой формулой общего вида. Третий алгоритм разработан для решения частного случая уравнения (4), когда булева формула представлена в дизъюнктивной нормальной форме (система вида (3)). Алгоритмы используют метод интеллектуального возврата, статические и динамические стратегии для определения локальной несогласованности, схемы обратного просмотра, трехзначную и многозначную логику.

3. Автоматизация параллельных вычислений в ИК РЕБУС

Методологической основой автоматизации проведения вычислительных экспериментов в области параллельного решения дискретных задач является использование подхода, базирующегося на знаниях. В связи с этим вопросы представления, накопления, модификации и использования знаний занимают центральное место в технологии, используемой ИК РЕБУС (подробное описание специфики предметной области изложено в работе [6]). При параллельном программировании в булевых ограничениях естественным образом выделяется ряд тесно связанных слоев знаний, представляющих: 1) булевы модели дискретных задач; 2) методы ре-

шения систем булевых уравнений, алгоритмы, разработанные на основе этих методов и программы, реализующие алгоритмы на одном из языков программирования; 3) спецификации параметров, модулей и операций, используемые для планирования вычислительного процесса в процедурной и не процедурной постановках задач; 4) статистические данные, накапливаемые в результате проведения вычислительных экспериментов, и результаты их обработки.

Технология решения комбинаторной задачи в ИК РЕБУС соответственно включает четыре этапа обработки информации: моделирование, декомпозиция булевой модели, проведение параллельных вычислений и обработка результатов. Первый, второй и четвертый этапы выполняются на рабочей станции, на третьем этапе строится план решения задачи, генерируется задание для проведения параллельных вычислений, которое передается в систему управления заданиями (СУПЗ) на кластер. Вычисления проводятся в узлах кластера, после чего результаты передаются на рабочую станцию.

3.1 Технология построения булевой модели

Базовая технология автоматического синтеза булевых ограничений по содержательному описанию дискретной задачи, применяемая в ИК РЕБУС, была разработана в процессе исследования построения булевых моделей для ряда классических и практических дискретных задач. Анализ этого процесса позволил выделить характерные особенности, классифицирующие дискретные задачи по ряду признаков. Первый признак – это зависимость булева вектора X от дискретного времени $t = 0, 1, 2, \dots, k$. С этой точки зрения мы различаем статические (или одношаговые) задачи, вектор состояния которых не зависит от времени t , и динамические (или многошаговые) задачи, вектор состояния которых зависит от времени t .

В первом случае булевы ограничения (в векторной форме) имеют вид $F(X) = 0$, во втором случае рассматривается уравнение вида $F(X_0, X_1, \dots, X_k) = 0$, где $X_i = X(t)$ при $t = i$.

Второй признак – это структуризация множества X булевых переменных задачи. Здесь рассматриваются варианты линейного упорядочения множества X (X рассматривается как вектор независимых переменных) и вариант упорядочения множества X в виде многомерного массива булевых переменных.

Приведенная классификация является в определенной степени условной, и существуют дискретные задачи, которые занимают промежуточное положение в рамках этой классификации. В общем случае, размерность динамических дискретных задач (при прочих равных условиях) существенно выше размерности статических задач, так как для динамической (многошаговой) задачи необходимо определить значение вектора состояния для каждого момента времени $t = 0, 1, 2, \dots, k$.

Построение компьютерной модели включает следующие этапы:

- задание множества булевых переменных $X = \{x_1, x_2, \dots, x_n\}$;
- задание системы булевых ограничений (множество $C = \{C_1, C_2, \dots, C_m\}$);
- представление системы ограничений в одном из форматов входных данных программ-решателей ИК РЕБУС.

Для автоматизации второго и третьего этапов была разработана специальная методика моделирования, подробное изложение которой приведено в работе [7]. Эта методика предоставляет два различных способа построения компьютерной модели: автоматический синтез булевой модели в виде текста программ на языке программирования (Фортран, С++) либо синтез системы булевых ограничений в международном формате DIMACS. Для компактной записи булевой модели разработан язык ЯСФОР (язык содержательных формулировок). Технология построения булевых моделей обеспечивает простоту, ясность и выразительность интерфейса с пользователем и позволяет формализовать постановку задачи на содержательном уровне. Компактная запись булевой модели повышает эффективность ее переноса на вычислительные узлы.

Для распараллеливания булевой модели наиболее предпочтительным с точки зрения решения уравнения (1) является метод расщепления, позволяющий выполнить декомпозицию уравнения (1) на произвольное число независимых уравнений, объединение решений которых даст

решение исходного уравнения. Каждое независимое уравнение может решаться в своем вычислительном узле. Анализатор булевой модели выбирает тактику расщепления и генерирует схему расщепления, используемую для последующей декомпозиции этой модели.

3.2 Параллельное решение

Для проведения вычислений в узлах кластера используются решатели – вычислительные модули ИК РЕБУС. Три метода решения булевых уравнений, лежащие в основе этих модулей, показали сравнительно высокую эффективность на ряде классических и практических дискретных задач. Специфика этих методов заключается в ориентации на решение больших разреженных систем булевых уравнений, а в качестве преимуществ первых двух методов следует отметить возможность представления левой части булева уравнения в общем виде, в отличие от существующих SAT-решателей задач удовлетворения булевых ограничений, требующих соответствующего представления в конъюнктивной нормальной форме.

Параллельный исполнитель – подсистема вычислительного эксперимента ИК РЕБУС – отвечает за проведение вычислений на кластере. ИК РЕБУС допускает две схемы проведения параллельных вычислений: 1) генерация параллельной программы, использующей библиотеку MPI с последующим сохранением, компиляцией и выполнением этой программы на кластере; 2) интеграция с ИК DISCOMP [8]. Обе схемы требуют средства интерактивного взаимодействия с кластером. В первом случае это взаимодействие обеспечивает ИК РЕБУС, использующий штатные средства системы управления заданиями кластера и утилиты удаленного доступа. Во втором случае взаимодействие с кластером осуществляется средствами доступа ИК РЕБУС к программному интерфейсу XML-RPC ИК DISCOMP. При этом перед проведением вычислительного эксперимента требуется обеспечить формирование в рабочей области пользователя на кластере область расчетных данных, содержащую спецификации объектов предметной области (ПО) и входные данные. Затем средствами интерфейса XML-RPC производится запуск вычислительного процесса. Мониторинг выполнения этого процесса, выявление его завершения и считывание результатов вычислений из области расчетных данных кластера в ИК РЕБУС также отслеживаются при помощи средств доступа к XML-RPC сервису. В этом случае на верхнем уровне абстракции вычислительная модель [9] ИК РЕБУС может быть представлена в виде двудольного графа $\langle P, O, E \rangle$, где доля P содержит множество параметров, доля O – множество операций ИК РЕБУС, а множество ребер E содержит направленные связи между параметрами и операциями. Списки параметров, операций и модулей хранятся в базе знаний ИК РЕБУС. В соответствии со спецификой решаемой задачи множество T допустимых типов параметров наряду с традиционными типами языков программирования дополнено специальным типом *par* – "параллельный список". Интерпретация операции $f : x \rightarrow y$, где x , y – параметры типа *par*, выполняется следующим образом: 1) в базе знаний осуществляется поиск всех вычислительных ресурсов (модуль + узел), реализующих операцию f ; 2) организуется параллельное применение i -го параллельного ресурса к i -му элементу списка x ; 3) результат применения присваивается i -му элементу списка y .

4. Архитектура ИК РЕБУС

В инструментальной среде ИК РЕБУС можно выделить три основных компонента: систему моделирования, систему проведения вычислительного эксперимента, систему работы с базой знаний. Система моделирования включает инструментальные средства автоматизации построения булевых ограничений и расщепления булевой модели. Система вычислительного эксперимента содержит инструментарий для построения плана решения задачи, организации процесса решения (последовательного или параллельного проведения вычислений), визуализации результатов. Информация, необходимая для работы этих систем, содержится в базе знаний ИК РЕБУС. Система работы с базой знаний содержит три компонента: справочник (предоставление справочной информации по базе знаний в соответствии с правами пользователя в результате выполнения запроса или составления отчета); администратор (администрирование пользователей, работа с нормативно-справочной информацией и мониторинг системы); пользователь

(просмотр разрешенных таблиц базы знаний и таблиц собственной базы данных, сохранение результатов).

Рассмотрим функциональные возможности тех компонентов ИК РЕБУС, которые отвечают за организацию параллельных вычислений: подсистемы расщепления модели, содержащей модули конвертирования, анализа и расщепления, подсистемы параллельного исполнителя, отвечающего за организацию интерактивного взаимодействия с кластером и подсистемы обработки результатов решения, содержащей модули постпроцессора, слияния и визуализации результатов работы. Эти программные средства обеспечивают выполнение следующих функциональных возможностей: анализ структуры булевой модели, выбор тактики расщепления, формирование схемы расщепления, генерация спецификаций параметров и операций предметной области и схем решения задачи, генерация паспорта задания для кластера, получение остаточной функции в узле в результате применения схемы расщепления к исходной функции, упрощение остаточной функции, параллельное решение полученных в результате расщепления подсистем булевых уравнений решателями вычислительного ядра ИК РЕБУС или внешними SAT-решателями, установленными в узлах кластера, слияние и визуализацию результатов решения, интерактивное взаимодействие с кластером.

Интерактивное взаимодействие ИК РЕБУС с кластером (рис. 1) как в случае генерации параллельной программы, так и в случае использования ИК DISCOMP, требует, во-первых, предварительного сохранения в рабочей области пользователя либо текста параллельной программы, либо спецификаций параметров, операций и схемы решения задачи в области расчетных данных для первого и второго случаев соответственно; во-вторых, формирования паспорта задания, содержащего перечень исполняемых программ, названия файлов с исходными данными и результатами работы, инструкции по выполнению программ, требования к ресурсам. Средства, поддерживающие эти функциональные требования, реализованы в параллельном исполнителе, являющемся компонентом системы проведения вычислительного эксперимента ИК РЕБУС.

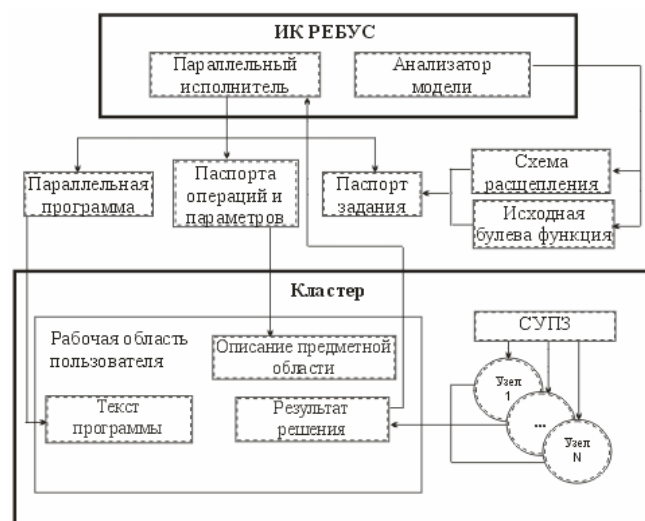


Рис. 1. Схема проведения параллельных вычислений в ИК РЕБУС

5. Заключение

В целом, применение указанных подходов к решению поставленной задачи обеспечивает создание программного комплекса со следующими функциональными возможностями: построение булевой модели (переход от вербального описания задачи к формальному в виде булева уравнения), распараллеливание булевой модели с использованием метода расщепления, обеспечение интерактивного взаимодействия с кластером, параллельное решение полученных в результате расщепления остаточных булевых уравнений, слияние и визуализация результатов решения. Представленная в данной статье инструментальная среда разработана на основе принципов создания многоплатформенных приложений, обеспечивающих переносимость на уровне исходного кода и использование приложений, разработанных для операционных систем

Windows, Linux в качестве модулей. Технология булева моделирования применялась для решения ряда дискретных задач, например: поиск гамильтонова цикла, нахождение покрытия конечного множества системой его подмножеств, составление плана ремонта электропоездов при равномерной загрузке локомотивного депо, составление оперативного плана отправления локомотивных бригад в условиях жесткого расписания грузовых поездов, генерация туннельных маршрутов (подзадача задачи о равномерной загрузке каналов связи СПД), построение параллельного плана требуемой длины в распределенных вычислительных системах [6, 10, 11]. Вычислительные эксперименты проводились на “горизонтальном” и ”вертикальном” кластерах ИДСТУ СО РАН под управлением соответственно ОС Windows и Linux и показали эффективность рассматриваемого подхода. Дальнейшее направление исследований связано с анализом накопленных статистических данных с целью определения оптимального числа узлов кластера и наиболее приемлемой тактики расщепления для конкретных практических дискретных задач, разработкой гибридного алгоритма распараллеливания технологий работы с MPI и OpenMP.

Литература

1. Ефимов С.С. Обзор методов распараллеливания алгоритмов решения некоторых задач вычислительной дискретной математики // Математические структуры и моделирование. – 2007. Вып. 17. – С.72-93.
2. Тимошевская Н.Е. Параллельные методы обхода дерева // Математическое моделирование. – 2004. – Т.16. – № 1. – С. 105-114.
3. Тимошевская Н.Е. Разработка и исследование параллельных комбинаторных алгоритмов // Вычислительные методы в дискретной математике. – 2009. – № 2(4). – С.95-103.
4. Данцин Е.Я. Алгоритмика задачи выполнимости // Вопросы кибернетики. Проблемы сокращения перебора. – М.: АН СССР, 1987. – С. 7-29.
5. Горшков С.П. Применение теории NP-полных задач для оценки сложности решения систем булевых уравнений // Обозрение прикладной и промышленной математики, серия дискретной математики, 1995. – Т.2, вып.3. – С. 325-398.
6. Опарин Г.А., Богданова В.Г. РЕБУС – интеллектуальный решатель комбинаторных задач в булевых ограничениях // Вестник НГУ. Серия: Информационные технологии, 2008. – Т. 6, вып. 1. – С. 60-68.
7. Опарин Г.А., Богданова В.Г. Ребус – интеллектуальный решатель задач в булевых ограничениях // Вестник ТГУ. Приложение, 2006. – №18. – С. 243-247.
8. Васильев С.Н., Опарин Г.А., Феоктистов А.Г., Сидоров И.А. Интеллектуальные технологии и инструментальные средства создания вычислительной инфраструктуры в сети Интернет // Вычислительные технологии. Специальный выпуск, 2006. – Т.11. – С.34-44.
9. Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Новопашин А.П. Решение булевых уравнений большой размерности в распределенной вычислительной среде // Распределенные вычисления и Грид-технологии в науке и образовании (Дубна, 29 июня-2 июля 2004 г.). Труды Межд. конф. – Дубна: ОИЯИ, 2004. – С. 164-169.
10. Опарин Г.А., Богданова В.Г., Новопашин А.П., Максимович В.В. Математические модели, методы и средства управления загрузкой каналов связи в магистральных сетях передачи данных // Инфокоммуникационные и вычислительные технологии и системы (ИКВТС-06): Материалы II Всерос. конф. с межд. участием. – Т. 2. – Улан-Удэ: Изд-во Бурятского государственного университета, 2006. – С. 74-87.
11. Опарин Г.А., Новопашин А.П. Булево моделирование планирования действий в распределенных вычислительных системах // Известия РАН. Теория и системы управления. – 2004. – №5. – С. 105-108.