

РЕШЕНИЕ СЛОЖНЫХ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ НА БАЗЕ КЛАСТЕРНОГО КОМПЬЮТЕРА

Черемисинова Л. Д.

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: cld@newman.bas-net.by

Описывается технология распараллеливания алгоритмов решения логико-комбинаторных задач, возникающих при автоматизации логического проектирования СБИС, с целью их реализации на многопроцессорных вычислительных системах кластерного типа.

ВВЕДЕНИЕ

Логико-комбинаторные задачи характеризуются тем, что объемы перерабатываемой информации при поиске их решений относительно невелики, а сами процессы переработки достаточно сложны, так как требуют, как правило, перебора и анализа значительного числа вариантов промежуточных решений. В силу этого становится проблематично решать такие задачи большой размерности за приемлемое время даже с использованием самых быстродействующих персональных компьютеров. Для решения таких задач все чаще используют многопроцессорные вычислительные системы, наиболее популярным типом которых являются кластеры.

В настоящей работе предлагается технология выполнения распараллеливания алгоритмов решения логико-комбинаторных на примере выполнения верификации результатов проектирования в рамках САПР логического проектирования заказных КМОП СБИС на основе удаленного решения на кластере задачи нахождения корней логических уравнений.

I. ЛОГИЧЕСКИЕ УРАВНЕНИЯ И ПОИСК ИХ КОРНЕЙ

Логическими уравнениями называются выражения вида $F(X) = 1$ и $F(X) = 0$, где $F(X)$ – формула булевой алгебры. Одним из важнейших классов логических уравнений являются уравнения, в которых формула $F(X)$ задана в виде КНФ K или ДНФ D : $K = 1$, $D = 0$ (задачи двойственны).

Широкий класс практически значимых задач, связанных с управлением и обработкой информации в дискретных системах, а также задач синтеза и верификации в микроэлектронике и других, сводится к проблеме булевой выполнимости КНФ или проверке ДНФ на тавтологию. Эти ключевые операции часто фигурируют во внутренних циклах алгоритмов, решающих эти задачи. Проблема выполнимости КНФ K состоит в проверке существования корня уравнения $K = 1$, т. е. такого набора значений переменных, который обращает в 1 КНФ (как и каждый ее дизъюнкт) и называется выполняющим. Задачу

выполнимости КНФ называют в литературе SAT-задачей.

В матричной интерпретации задача сводится к задаче анализа на выполнимость троичной матрицы T , строки которой задают дизъюнкты исходной КНФ. Проверить троичную матрицу T на выполнимость означает найти троичный вектор t , который обращает в 1 каждый дизъюнкт. Или убедиться в том, что такого вектора не существует (когда матрица T не выполнима, т. е. $K = 0$). Анализ матрицы T производится методом обхода дерева поиска выполняющего вектора t . Текущая ситуация, представляемая в узле дерева, характеризуется значением вектора t , отдельные компоненты которого определены (т. е. имеют значение 0 или 1), а другие (отмеченные «–») подлежат доопределению в процессе движения по дереву. Троичный вектор t однозначно определяет минор $T(t)$ матрицы T , расположенный на пересечении ее строк, соответствующих невыполнимым относительно вектора t дизъюнктам, и столбцов, отмеченных его неопределенными компонентами.

II. РАСПАРАЛЛЕЛИВАНИЕ ЗАДАЧИ ПОИСКА КОРНЕЙ ЛОГИЧЕСКИХ УРАВНЕНИЙ

Распараллеливание алгоритмов решения SAT-задач может производиться на основе динамического или статического распределения нагрузки процессоров. Для систем кластерного типа более подходит метод статического распределения нагрузки процессоров, который не требует интенсивного взаимодействия узлов. Он предполагает декомпозицию исходной SAT-задачи на семейство непересекающихся подзадач, каждая из которых обрабатывается SAT-решателем на отдельном вычислительном узле вычислительной системы.

В основе реализованного в настоящей работе метода лежит идея гибридного распределения нагрузки между подчиненными процессами, когда сначала вся нагрузка делится, а затем освободившиеся процессоры подключаются для решения подзадач, решаемых загруженными процессорами.

Метод распараллеливания с предварительным распределением нагрузки основан на расщеплении анализируемых булевых формул, ис-

пользуя разложение Шеннона по k переменным. Задача разбивается на не более чем $l = 2k$ подзадач (не более, так как некоторые из подзадач могут иметь тривиальное решение, не требующее обсчета на процессоре).

Полученные частные задачи решаются параллельно l процессорами решающими свою задачу над своими исходными данными до конца, а корневой процессор собирает результаты частичных решений и формирует из них итоговое решение общей задачи. Если хотя бы для одной частной задачи найдется решение, то по нему однозначно находится набор значений переменных, выполняющих исходную КНФ. Хотя время решения задачи существенно зависит от выбора переменных для разложения, используется тривиальная процедура разложения, так как она выполняется целиком на корневом процессоре в режиме последовательного выполнения.

III. ПРОГРАММА РЕШЕНИЯ ЛОГИЧЕСКИХ УРАВНЕНИЙ НА КЛАСТЕРНОМ КОМПЬЮТЕРЕ

Метод решения логических уравнений реализован MPI-программой «Вычислитель» для кластерного компьютера [1]. Исходными данными для программы «Вычислитель» служит описание КНФ в текстовом виде или в формате DIMACS, который представляет собой стандартный интерфейс для SAT-решателей. Результатом выполнения программы является текст, содержащий режим и длительность решения, а также выполняющий вектор, если КНФ выполнима.

В мультипроцессорных системах с индивидуальной памятью текст MPI-программы перед исполнением копируется в память каждого из обрабатывающих процессов. Сама параллельная MPI-программа делится на общие, корневые и подчиненные участки, исполняемые всеми, корневым и подчиненными процессами. Для организации обмена сообщениями предусматриваются специальные приемо-передающие буферы. Для рассматриваемой задачи через них передаются векторы, задающие миноры и выполняющие наборы, сигналы окончания работы. При реализации программы «Вычислитель» был выделен минимально необходимый набор MPI-операций: для запуска и завершения работы, определения выделенной конфигурации кластера, выделения и распознавания блоков программы для процессов, работы с приемо-передающими буферами (опроса, приема и отправки), операций обмена сообщениями через буфера.

Распределение нагрузки в MPI-программе «Вычислитель» производится корневым процессом динамически через магазин миноров по мере их формирования при обходе дерева поиска. Минор матрицы T задается вектором t . Когда какой-либо подчиненный процесс при достраивании вектора t в процессе обхода своего поддерева поиска:

1) находит искомым вектор t (выполняющий все строки обрабатываемого минора), тогда задача решена и все остальные процессы прекращают работу, 2) убеждается в невыполнимости минора, тогда он освобождается и получает очередной минор из магазина или минор, полученный (по запросу корневого процесса) на очередном шаге обхода дерева поиска некоторым занятым процессом (делит с ним нагрузку). Поиск выполняющего вектора t продолжается до тех пор, пока он не будет найден (КНФ выполнима) или пока все подчиненные процессы не закончат работу (КНФ не выполнима).

IV. ИНТЕГРАЦИЯ ПРОГРАММЫ РЕШЕНИЯ ЛОГИЧЕСКИХ УРАВНЕНИЙ В МАРШРУТ ПРОЕКТИРОВАНИЯ

MPI-программа «Вычислитель» в целом является частью трехслойной системы, состоящей из программы для суперкомпьютера, сервиса связи и агента пользователя. Протокол работы грид-сервиса состоит из следующих шагов: 1) транспортировки файла с анализируемой КНФ в файловую систему на управляющей машине кластера; 2) запуска программы «Вычислитель»; 3) обратной транспортировки файла с результатами решения на клиентский компьютер.

Методика запуска на кластерном компьютере программ решения комбинаторных задач, возникающих в процессе проектирования интегральных схем, отработана на примере разработанного в лаборатории логического проектирования ОИПИ НАН Беларуси программного комплекса ЭЛС «Энергосберегающий логический синтез» [2]. Программа верификации из ЭЛС была модифицирована следующим образом: после получения КНФ в процессе выполнения операции верификации в среде комплекса ЭЛС файл с анализируемой КНФ, заданной в формате Dimacs, транспортируется по сети на управляющую машину кластера; запускается программа «Вычислитель» на суперкомпьютере; файл с результатами решения возвращается в систему ЭЛС. Если файл содержит найденный вектор (что говорит о том, что КНФ выполнима), делается вывод о неэквивалентности сравниваемых описаний, иначе описания эквивалентны.

V. СПИСОК ЛИТЕРАТУРЫ

1. Черемисинов, Д. И. Использование параллельных вычислений при автоматизированном проектировании СВИС / Д. И. Черемисинов, Л. Д. Черемисина // Проблемы разработки перспективных микро- и наноэлектронных систем – 2016. Сборник трудов / под общ. ред. академика РАН А. Л. Стемпковского. М. : ИППМ РАН, 2016. Часть I. – С. 32-39.
2. Бибилло, П. Н. Автоматизация логического синтеза КМОП схем с пониженным энергопотреблением / П. Н. Бибилло, Л. Д. Черемисина, С. Н. Кардаш и др. // Программная инженерия. – 2013. – № 8. – С. 35-41.