



приложения, конечного пользователя или других облачных сервисов. Новизна последних двух моделей заключается в одновременном описании подзадач и источников данных разного типа с указанием направления и способа передачи данных.

Разработанная структурная модель позволяет описать основные элементы программно-управляемой инфраструктуры виртуального ЦОД.

Исследования выполнены при поддержке РФФИ (проекты № 16-37-60086 и № 16-07-01004).

Литература

1. Bolodurina I., Parfenov D., Shukhman A. Approach to the effective controlling cloud computing resources in data centers for providing multimedia services // Proceedings of International Siberian Conference on Control and Communications (SIBCON), Omsk, Russia, 2015. pp.:1-6, DOI: 10.1109/SIBCON.2015.7147170
2. Полежаев П.Н., Ушаков Ю.А., Шухман А.Е. Система управления ресурсами для высокопроизводительных вычислений, основанная на использовании программно-конфигурируемой сети // "Системы управления и информационные технологии: научно-технический журнал", 2013. - №4(54). - С. 65-69.
3. Парфёнов, Д.И. Управление потоками данных в высоконагруженных информационных системах, построенных на базе облачных вычислений / Болодурина И.П., Парфёнов Д.И. // Системы управления и информационные технологии: научно-технический журнал. – 2015. - № 1.1. – С. 111-118.
4. Mishra M., Sahoo A. "On theory of VM placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach" Cloud Computing (CLOUD), IEEE International Conference, 2011, Washington: IEEE Press, p.275-282.

Э.И. Ватутин, С.Ю. Валяев, В.С. Титов

АНАЛИЗ РЕЗУЛЬТАТОВ ПРИМЕНЕНИЯ МЕТОДА СЛУЧАЙНОГО ПЕРЕБОРА ПРИ ПОСТРОЕНИИ РАЗБИЕНИЙ ГРАФ-СХЕМ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ В ЗАВИСИМОСТИ ОТ РАЗМЕРНОСТИ ЗАДАЧИ И СИЛЫ ОГРАНИЧЕНИЙ

(Юго-Западный государственный университет)

При проектировании цифровых устройств их структура традиционно делится на две части: операционную и управляющую. Операционная часть, как правило, включает в своем составе схемы, обеспечивающие обработку информации в соответствии с требованиями решаемой задачи, а управляющая обеспечивает координацию ее работы путем определения ее состояния в ходе получения вектора сигналов логических условий $X = \{x_1, x_2, \dots, x_p\}$ и выдачи сигналов микроопераций $Y = \{y_1, y_2, \dots, y_q\}$ в соответствии с заданным алгоритмом



управления. Так, например, в качестве сигнала логического условия может служить единичное значение бинарного признака завершения работы одной из схем в составе операционной части [1], достижение счетчиком заданного значения и т.п.; управляющие сигналы обычно используются для фиксации значений в запоминающих элементах схемы (ОЗУ, триггерах и пр.). При проектировании относительно простых систем логического управления (СЛУ), ориентированных на реализацию небольших алгоритмов управления, производится разметка соответствующей граф-схемы, составление графа переходов и проектирование соответствующего управляющего автомата Мили или Мура [2]. Если же алгоритм логического управления $G = \langle A, V \rangle$, где A – множество его вершин, V – множество дуг, является комплексным и содержит большое количество вершин (как правило сотни), параллельные фрагменты, циклы различного вида (в том числе параллельные), то традиционный подход сталкивается с существенными трудностями. С целью их преодоления в [3–5] было предложено использование однородных многомодульных мультисистем, именуемых логическими мультиконтроллерами (ЛМК), включающих в своем составе определенное количество относительно простых логических контроллеров, каждый из которых может выполнять последовательный подалгоритм A_i и $\bigcup_{i=1}^n A_i = A$,

" $i, j = \overline{1, n}, i \neq j: A_i \cap A_j = \emptyset$ ограниченной сложности, являющейся частью исходного комплексного алгоритма управления G . Каждый контроллер в составе ЛМК имеет память микропрограмм ограниченной емкости W_{\max} , в которой хранится соответствующий подалгоритм A_i , и заданное число ножек (или портов ввода/вывода) для приема сигналов логических условий X_{\max} и выдачи сигналов микроопераций Y_{\max} . При обеспечении координации работы отдельных подалгоритмов, назначаемых на различные контроллеры в составе ЛМК, можно добиться выдачи логических условий, соответствующий заданному алгоритму управления G , т.е., другими словами, реализации СЛУ в базисе ЛМК. При этом возникает ряд задач дискретной комбинаторной оптимизации [6], связанных с разбиением заданного алгоритма управления на подалгоритмы (блоки разбиения) в соответствии с ограничениями базиса ЛМК [3–5], размещением полученных блоков в контроллерах, имеющих заданную топологию связей (например, матричную) [3], координацией параллельной работы модулей [3], оптимизацией структуры ЛМК [7] и пр.

В указанной работе рассматривается задача поиска субоптимального разбиения заданного алгоритма логического управления (ее подробное описание можно найти в монографиях [4–5]). При ее решении учитываются структурные и функциональные ограничения базиса ЛМК, к которым относятся соответственно емкость памяти контроллера W_{\max} , число ножек X_{\max} и Y_{\max} и требование отсутствия параллельных вершин в составе блоков разбиения (что вынуждает решать частную задачу классификации бинарных отношений вершин граф-схемы G [8]), и минимизируются значения множества показателей качества, к



которым относятся число блоков разбиения Z_H , степени дублирования сигналов логических условий и микроопераций Z_X и Z_Y , число межблочных связей по управлению Z_a и межконтроллерный график передач управления Z_d . Первые четыре показателя качества напрямую влияют на требования, предъявляемые к ЛМК (в первую очередь, на его аппаратную сложность и, как следствие, на массогабаритные параметры и требования к питанию и тепловыделению), последний определяет требования к организации коммуникационной подсистемы ЛМК (например, с использованием кольцевой или матричной топологии) и быстродействие СЛУ в целом. Путем взвешенного суммирования и нормировки частные показатели сводятся в интегральный показатель Z_j [9].

Задача поиска разбиения не может быть решена точно за приемлемое время ввиду того, что асимптотика числа решений характеризуется числом Белла B_N , где $N = |A|$ – число вершин в составе граф-схемы G , поэтому для ее решения на практике применяются различные эвристические методы. В настоящее время в составе разработанной авторами программной среды PAE [10] для автоматизированного построения разбиений имеются программные реализации следующих методов: метода С.И. Баранова [11], реализующего классический жадный подход (сокр. В); его модификация с ограничением на смежную окрестность [12] (сокр. АВ); метода параллельно-последовательной декомпозиции [13], основанного на множестве эквивалентных преобразований исходной граф-схемы G (сокр. Р); и метода случайного перебора [14] (сокр. RS). Первые три метода являются последовательными и достаточно быстро синтезируют единственное решение, метод случайного перебора относится к итерационным и обеспечивает перебор заданного количества решений C с выбором наилучшего из них, что требует существенно больших затрат вычислительного времени.

Вычислительные эксперименты [15–16] с последовательными методами показали, что при различных значениях силы технологических ограничений и размерности решаемой задачи методы демонстрируют существенно различную степень минимизации показателей качества, поэтому для каждого конкретного случая их практического применения можно сформулировать рекомендации о том, эффективно или нет применять данный метод в данных условиях. Определение областей преимущественного использования методов является вычислительно сложной задачей ввиду того, что необходим анализ степени минимизации частных показателей качества, усредненных на выборках граф-схем параллельных алгоритмов логического управления с псевдослучайной структурой [17] объемом $K = 5000$, с расчетом их средневыборочных значений g_x , $x \in \{H, X, Y, a, d, J\}$ и вероятностей получения суб- или квазиоптимального решения g_x для множества точек, образующих двумерный срез пространства параметров $(N, X_{\max}, Y_{\max}, W_{\max})$. Ввиду слабой связности указанной задачи она может быть эффективно решена с использованием грид систем на добровольной основе (в настоящее время для этого используется проект Gerasim@Home на платформе BOINC [18]).



Ввиду ограничений на время вычислительных экспериментов, составившее до 80 часов на расчет одного задания (англ. Work Unit, сокр. WU), для метода случайного перебора было выбрано число итераций $C = 10\,000$ для $N \cdot J \leq 100$ и $C = 1000$ для $100 < N \cdot J \leq 200$. В результате расчетов, выполненных на грид в период с июня 2014 по январь 2015 гг., и постобработки полученных результатов были получены результаты, использованные для анализа качества решений, получаемых методом случайного перебора. Как уже было отмечено ранее, наиболее показательным с позиции сравнения методов является сопоставление вероятностей g_x для решений, полученных с использованием различных методов. При этом возможно определение метода, для которого указанная величина максимальна, а соответствующая точка пространства параметров будет принадлежать к области преимущественного использования данного метода.

В результате обработки экспериментальных данных были получены следующие области преимущественного использования метода случайного перебора, соответствующие срезам (N, W_{\max}) , $1 < N \cdot J \leq 600$, $4 \leq W_{\max} \leq 200$ (рис. 1) и (N, X_{\max}) , $1 < N \cdot J \leq 700$, $3 \leq X_{\max} \leq 150$ (рис. 2) пространства параметров.

Анализ полученных результатов позволяет сделать вывод о том, что метод случайного перебора эффективен по всем параметрам качества, за исключением интенсивности межблочных взаимодействий Z_d , лишь при малой размерности задачи ($N < 60$ и 70). С ее ростом, несмотря на существенно меньшие затраты вычислительного времени, эффективными оказываются последовательные методы. В области сильных ($X_{\max}, W_{\max} < 50$) ограничений метод случайного перебора демонстрирует рекордно низкое число блоков разбиения. В перспективе данная особенность может быть использована при совместном использовании последовательных и итерационных методов в соответствии со стратегией ветвей и границ [19]: решение последовательного метода выступает в качестве начального значения рекорда и обеспечивает отсеечение неперспективных решений итерационного метода и, как следствие, экономию затрат вычислительного времени.

Работа выполнена в рамках выполнения государственного задания для Юго-Западного государственного университета на 2014–2017 гг., номер НИР 2246, а также при поддержке гранта Президента РФ для молодых ученых – кандидатов наук МК-9445.2016.8.

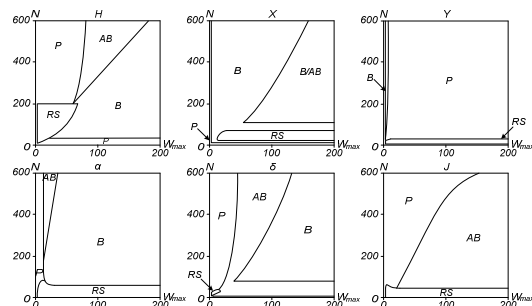


Рис. 1. Области преимущественного использования метода случайного перебора для среза (N, W_{\max})

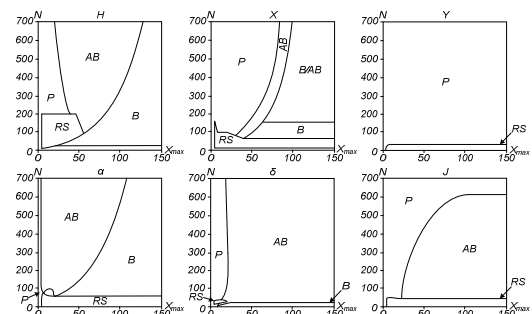


Рис. 2. Области преимущественного использования метода случайного перебора для среза (N, X_{\max})

Литература

1. Ватутин Э.И., Зотов И.В., Титов В.С. Акселератор для быстрого преобразования конструктивных подмножеств вершин параллельных алгоритмов // Параллельные вычисления и задачи управления (РАСО'10). М.: ИПУ РАН, 2010. С. 1301–1366.
2. Чернецкая И.Е. Теория автоматов. Курск, ЮЗГУ, 2011. 142 с.
3. Зотов И.В., Титов В.С. и др. Организация и синтез микропрограммных мультимикроконтроллеров. Курск: изд-во «Курск», 1999. 368 с.
4. Ватутин Э.И., Титов В.С. и др. Комбинаторно-логические задачи синтеза разбиений параллельных алгоритмов логического управления при проектировании логических мультимикроконтроллеров. Курск: КурскГТУ, 2010. 200 с.
5. Ватутин Э.И. Проектирование логических мультимикроконтроллеров. Синтез разбиений параллельных граф-схем алгоритмов. Saarbrücken: LAP, 2011. 292 с.



6. Ватутин Э.И., Титов В.С., Емельянов С.Г. Основы дискретной комбинаторной оптимизации. М.: АРГАМАК-МЕДИА, 2016. 270 с.
7. Ватутин Э.И., Титов В.С. Структурно-параметрическая оптимизация систем логического управления с использованием добровольных распределенных вычислений // Известия ЮЗГУ. Серия «Управление, вычислительная техника, информатика. Медицинское приборостроение». 2012. № 2–1. С. 12–17.
8. Ватутин Э.И., Зотов И.В. Построение матрицы отношений в задаче оптимального разбиения параллельных управляющих алгоритмов // Известия КурскГТУ. Курск, 2004. № 2. С. 85–89.
9. Ватутин Э.И. Оценка качества разбиений параллельных управляющих алгоритмов на последовательные подалгоритмы с использованием весовой функции // Интеллект – 2005. Тула, 2005. С. 29–30.
10. Ватутин Э.И., Зотов И.В. Программная система для построения разбиений параллельных управляющих алгоритмов // Идентификация систем и задачи управления (SICPRO'06). М.: ИПУ РАН, 2006. С. 2239–2250.
11. Баранов С.И., Журавина Л.Н., Песчанский В.А. Обобщенный метод декомпозиции граф-схем алгоритмов // А и ВТ. 1982. № 5. С. 43–51.
12. Ватутин Э.И., Леонов М.Е. Использование смежной окрестности при жадном последовательном формировании блоков разбиения граф-схем параллельных алгоритмов // Известия высших учебных заведений. Приборостроение. 2013. Т. 56. № 6. С. 30–35.
13. Ватутин Э.И., Зотов И.В. Метод формирования субоптимальных разбиений параллельных управляющих алгоритмов // Параллельные вычисления и задачи управления (РАСО'04). М.: ИПУ РАН, 2004. С. 884–917.
14. Ватутин Э.И., Колясников Д.В., Мартынов И.А., Титов В.С. Метод случайного перебора в задаче построения разбиений граф-схем параллельных алгоритмов // МППОС'14. Барнаул, 2014. С. 115–125.
15. Ватутин Э.И., Титов В.С. Анализ областей качественного превосходства последовательных эвристических методов синтеза разбиений при проектировании логических мультимикроконтроллеров // Известия высших учебных заведений. Приборостроение. 2015. Т. 58. № 2. С. 115–122. DOI: 10.17586/0021-3454-2015-58-2-115-122.
16. Vatutin E.I., Valyaev S.Yu., Titov V.S. Comparison of Sequential Methods for Getting Separations of Parallel Logic Control Algorithms Using Volunteer Computing // CEUR Workshop Proceedings. Vol. 1502. Aachen, 2015. P. 37–51.
17. Vatuin E.I. Constructing Random Sample Parallel Logic Control Algorithms // Baltic Olympiad on Automatic Control (BOAC'06). SPb, 2006. P. 162–166.
18. <http://gerasim.boinc.ru>
19. Land A.H., Doig A.G. An Automatic Method of Solving Discrete Programming Problems // Econometrica. Vol. 28. 1960. pp. 497–520. DOI: 10.2307/1910129.