ОБОЗРЕНИЕ
ПРИКЛАДНОЙ И ПРОМЫШЛЕННОЙ
Том 19 МАТЕМАТИКИ Выпуск 3
2012

С. О. Беззубцев (Москва, ООО «Стартлинк»). **Об одной задаче оценки времени выполнения программ на специализированном оборудовании.**

Рассматривается задача обработки данных на узле общего назначения, оборудованном модулем ускорителя PBM6 (далее, модуль) [1]. Обработка данных сводится к последовательному выполнению следующих операций: предобработка данных; программирование модуля; считывание результатов вычислений с модуля; постобработка данных. При этом ко всей последовательности операций предъявляются требования ко времени выполнения: суммарное время выполнения указанной последовательности не должно превышать наперед заданного порогового значения. Таким образом, для рассматриваемого аппаратно-программного комплекса возникает задача оценки наихудшего времени выполнения программы взаимодействующей с модулем.

Операции пред- и постобработки представляют собой последовательные программы и выполняются только на процессоре общего назначения (далее, исполнитель), а, следовательно, для оценки времени выполнения этих операций могут применяться любые существующие методы, например, [2–5].

Операции программирования модуля и считывания результатов вычислений с модуля осуществляются программой путем выполнения на исполнителе последовательностей инструкций, реализующих доступ к ячейкам модуля. Такие операции требуют отдельной обработки, и оценка времени их выполнения не может быть вычислена с применением существующих методов ввиду наличия асинхронно изменяющихся (по отношению к программе и исполнителю) данных — значений ячеек модуля, а, следовательно, возникающего недетерменизма поведения этой программы.

Рассмотрим данную проблему на примере. Пусть программа такова, что переходы (направленные дуги, которым сопоставлено описание условия перехода по ним) из некоторой вершины V графа потока управления зависят от значения ячейки X исполнителя (здесь и далее под ячейкой исполнителя понимается ячейка оперативной памяти или регистра исполнителя, под ячейкой модуля понимается регистр модуля, доступный исполнителю на чтение и запись через шину PCI-express; в тексте ячейки исполнителя обозначены литерой X, ячейки модуля — литерой A). При этом само значение ячейки X является результатом операции доступа к ячейке A модуля. Из этого, очевидно, следует, что выполнимость каждого из путей, содержащих обозначенную вершину V, не может быть определена без учета возможных значений A.

Таким образом, если:

— задана программа P на подмножестве языка C, без циклов, рекурсий, без использования адресной арифметики, содержащая только операции «$=, \neq, >, <$» и любые их комбинации в условиях ветвления;

— задана модель исполнителя I программы P парой функций BCET(bb), WCET(bb), задающих отображение номера линейного участка программы P в оценки наилучшего и наихудшего времен выполнения, соответственно;

— задана модель модуля M функцией $\mathrm{Hm}(T1, T2, A)$, задающей диапазоны возможных значений ячейки A модуля при выполнении операций на ней в интервале времени $[T1, T2]$,

то для заданного описания аппаратно-программного комплекса может быть вычислена оценка времени выполнения программы P, выполняющейся на исполнителе I, взаимодействующей с модулем M, при этом точность оценки будет зависеть только от точности заданных моделей I и M.

Очевидно, что точная оценка может быть вычислена только при условии вычисления оценки для всех путей программы P (за вычетом невыполнимых путей). Полный перебор всех исполняемых путей реализуется как вычисление оценки времени выполнения каждого из путей программы, вычисление достижимости каждого из путей программы и последующего выбора исполняемого пути с максимальным временем выполнения.

Вычисление времени выполнения сводится к последовательному вычислению времен выполнения всех линейных участков пути, исполняемость пути сводится к поиску целочисленного решения для системы неравенств, составленной из условий переходов, встречающихся в данном пути.

Сокращение времени вычисления оценки может быть достигнуто за счет применения итеративно уточняющего алгоритма на базе метода BMC [6]. На каждой итерации данного алгоритма выдвигается проверяемое решение t. Для проверки решения t применяется метод BMC [7], [8]. С помошью BMC реализуется поиск контрпримера, т. е. такого пути в программе, время выполнения которого превосходит t. В случае, если контрпример найден, то выбирается следующее приближение решения $(t + s)$, где $s > 0$, и поиск контрпримера повторяется. В противном случае найденное решение — искомая оценка времени выполнения программы. Очевидно, что для получения точной оценки потребуется не более, чем N итераций, где N — число выполнимых путей в программе P. При этом число итераций может быть уменьшено до $\log_2(N)$ по методу бисекции.

Экспериментальные исследования показывают значительный выигрыш по времени вычисления оценки с применением итеративно уточняющего метода по сравнению с полным перебором. Примеры приведены в таблице.

Таблица. Зависимость времени анализа от числа условных операторов в программе

Количество условных операторов	20	30	40	50
Время анализа методом полного перебора (сек)	160	2000	10000	25000
Время анализа итеративно-уточняющим методом (сек)	3	4	6	10

Несмотря на то, что время анализа, достигаемое при реализации итеративно уточняющего метода, растёт экспоненциально, как и в случае реализации полного перебора, его небольшие значения позволяют использовать данный метод на практике для анализа программ, взаимодействующих с модулем.

Дальнейшего внимания и изучения заслуживают: адаптация методов свертки графов потока управления и их применимости для решения задачи оценки времени выполнения программ, взаимодействующих с внешними, асинхронными источниками данных; применимость методов оценки частотных характеристик поведения программ для оценки времени выполнения на многоитерационных сценариях; снятие или смягчение наложенных выше ограничений.

СПИСОК ЛИТЕРАТУРЫ

1. *Беззубцев С. О.* Особенности управления операциями ввода–вывода для ускорителя ресурсоемких вычислений. — В сб.: Материалы Второй Всероссийской научно-технической конференции «Суперкомпьютерные технологии» (СКТ-2012). Ростов-на-Дону: Изд-во ЮФУ, 2012, с. 93–98.
2. *Wilhelm R., Engblom J.* The Worst-Case Execution Time Problem. — In: Overview of Methods and Survey of Tools, 2008.
3. *Heckmann R., Ferdinand C.* Worst-Case Execution Time Prediction by Static Program Analysis, 2004.
4. *Tan L.* The Worst Case Execution Time Tool Challenge 2006, 2006.
5. *Ющенко Н. В.* Оценка времени выполнения программ статико-динамическим методом. — Программные системы и инструменты, № 2. ВМиК МГУ, 2001, с. 157–167.
6. *Biere A., Cimatti A., Clarke E. M.* Bounded Model Checking. — Adv. Comput., 2003, v. 58, p. 118–149.
7. *Kim S., Patel H. D., Edwards S. A.* Using a Model Checker to Determine Worst-case Execution Time. — Computer Science Technical Report CUCS-038-09. Columbia University, 2009.
8. *Kroening D.* CBMC: Bounded Model Checking for ANSI-C. http://www.cprover.org/cbmc/.
9. *Вдовин П. М.* Методы и средства оценки наихудшего времени выполнения распределенной программы. Дипломная работа. М.: ВМиК МГУ, 2012.