

Поиск критических данных в графической модели параллельного алгоритма

А.Н. Коварцев, В.В. Жидченко

В Самарском государственном аэрокосмическом университете сотрудниками кафедры программных систем создана и успешно развивается технология графо-символического программирования (ГСП-технология) [1, 2]. ГСП-технология предназначена для разработки последовательных и параллельных программ с использованием графической модели описания алгоритмов (граф-модели). Граф-модель ГСП-технологии является разновидностью диаграмм потоков управления: алгоритм представляется в виде графа, вершины которого описывают действия над данными, а дуги определяют передачу управления между вершинами. На основании граф-модели алгоритма программы синтезируются автоматически. Разработаны средства анализа корректности модели и тестирования создаваемых программных модулей. ГСП-технология позволяет существенно сократить трудозатраты на разработку и отладку программ со сложной логической структурой, выполняющих большие объемы вычислений. Основная область применения разрабатываемых программ – решение вычислительных задач и моделирование систем.

Параллельные программы, создаваемые в ГСП-технологии, используют общую память. Для предотвращения ошибок несогласованного доступа к данным общей памяти в ГСП-технологии создан ряд методов автоматического анализа модели алгоритма. Одним из них является метод поиска критических данных. Данное предметной области (переменная программы) называется критическим, если оно используется несколькими вершинами граф-модели, которые в процессе вычислений могут исполняться одновременно. При этом хотя бы одна из вершин использует это данное для записи. Наличие критических данных является потенциальным источником ошибок в параллельной программе.

Метод поиска критических данных базируется на понятии способа использования данных предметной области - функции $H(Ald)$, использующей два аргумента: A - объект граф-модели, в качестве которого может выступать отдельная вершина, подграф или граф-модель в целом; d – данное предметной области. Способ использования определяет возможные действия над данным d в вершине A : чтение ($H(Ald)=1$), запись ($H(Ald)=2$), данное не используется ($H(Ald)=0$). Для критических данных $H(Ald)=3$.

Над способами использования данных вводятся двуместные операции следования (обозначается символом Δ), ветвления (∇) и параллельного исполнения ($\#$), определяющие способ использования данного d подмножеством из двух вершин. Указанные операции определены для трех возможных способов взаимного расположения вершин в граф-модели алгоритма.

Введенные операции позволяют определить алгебру над способами использования данных в граф-модели алгоритма. Для граф-моделей с произвольной структурой, не содержащих иерархии и кратных дуг, алгебра над способами использования данных позволяет строить формулу для вычисления способа использования каждого данного граф-моделью в целом. Если для какого-то данного результат вычисления его способа использования граф-моделью равен 3, то в рассматриваемой граф-модели возможен конфликт совместного использования этого данного. Следовательно, разработчику граф-модели необходимо позаботиться об устранении конфликта путем ее модификации, переименования данных или применения механизмов синхронизации.

Литература

1. Коварцев А.Н. Автоматизация разработки и тестирования программных средств. - Самарский государственный аэрокосмический университет, Самара, 1999. - 150 с.
2. Коварцев А.Н., Жидченко В.В. Программный комплекс разработки алгоритмов и программ параллельных вычислений // Высокопроизводительные параллельные вычисления на кластерных системах: Материалы Девятой международной конференции-семинара / Владимир: Владимирский государственный университет, 2009. С. 225-229.