Э.И. Ватутин

ОЦЕНКА СТЕПЕНИ ПАРАЛЛЕЛИЗМА ПАРАЛЛЕЛЬНОЙ ГРАФ-СХЕМЫ АЛГОРИТМА

Курск, Курский государственный технический университет

Одним из возможных способов построения систем логического управления (СЛУ) является их синтез в базисе логических мультиконтроллеров [1]. При этом подобные СЛУ, основанные на принципах однородности и модульности, способны быть настроены на произвольный параллельный алгоритм логического управления теоретически неограниченной сложности (выражаемой обычно числом вершин N в его представлении в виде графа) путем его разбиения на множество последовательных блоков в соответствии с рядом ограничений функционального и структурного характера. Одной из важных числовых оценок, значение которой вызывает интерес как в ходе построения разбиения или анализа его качества, так и при решении смежных задач (например, распараллеливания последовательного алгоритма), является степень параллелизма $\omega_{\max} = \omega\left(G^0\right)$ граф-схемы параллельного алгоритма $G^0 = \left\langle A^0, V^0 \right\rangle$, где A^0 – множество вершин, V^0 – множество дуг. Оценка $\omega_{\max}$ определяет нижнюю границу числа блоков в разбиении и, следовательно, представляет собой теоретический минимум аппаратной сложности СЛУ, выраженный числом модулей (контроллеров) в ее составе.

Степень параллелизма алгоритма может быть определена как мощность максимального по включению подмножества попарно-параллельных вершин $\tilde{A} = \left\{a_{i_1}, a_{i_2}, ..., a_{i_L}\right\} \subseteq A^0$, причем $\forall a_{i_j} \in \tilde{A}, a_{i_k} \in \tilde{A}, j \neq k: \ a_{i_j} \omega a_{j_k}$, где ω – символ бинарного отношения параллельности вершин [1]. Способ вычисления степени параллелизма исходя из приведенного определения $\omega_{\max} = \left|\tilde{A}\right|$ вычислительно сложен и применим на практике только для алгоритмов малой размерности $\left(N < 15 - 20\right)$ [2], т.к. требует реализации комбинаторного перебора значительного числа подмножеств вершин $\tilde{A}_j$ (фактически данная задача сводится к задаче выделения максимальной по включению клики на множестве отношения параллельности вершин, которая, как известно, является *NP*-трудной). Еще одним способом вычисления искомой числовой оценки является ее определение как ω-мощности базового сечения $\Omega_{\max}$ алгоритма G^0: $\omega_{\max} = \left|\Omega_{\max}\right|^{\omega}$ [1]. Алгоритм вычисления ω-мощности является полиномиальным, однако он требует наличия синтезированного базового сечения $\Omega_{\max}$, для отыскания которого, в свою очередь, требуется построение и редукция системы *R*-выражения Ξ. Несмотря на то, что алгоритмы построения и редукции системы *R*-выражения Ξ также являются полиномиальными, реализация операций над *R*-выражениями, представленными в виде деревьев, весьма трудоемка (более 2400 строк (88 КБ) исходного кода при программной реализации, порядка 60 млн. эквивалентных вентилей при аппаратной). Поэтому вызывает интерес разработка новых алгоритмов вычисления оценки $\omega_{\max}$, отличительными особенностями которых будут низкая вычислительная сложность и простота реализации.

Рассмотрим алгоритм вычисления оценки $\omega_{\max}$ с использованием матрицы отношений вершин. Прототипом для него послужил метод построения разбиений С.И. Баранова с последовательным формированием блоков разбиения [3], дающий минимально возможное число блоков в разбиениях в условиях отсутствия технологических ограничений (что было выявлено эмпирически в ходе проведенных вычислительных экспериментов).

```
Sep(A⁰) ≔ {A₁}; A₁ ≔ {aнач, aкон}; n ≔ 0;
repeat  // выполняется ωmax раз
  n ≔ n + 1;
  for aᵢ ∈ A⁰\Sep(A⁰)  // выполняется не более чем N раз
    if (∀ aⱼ∈Aₙ: ¬(aᵢ ω aⱼ))  // выполняется за время не более чем O(N)
      Aₙ ≔ Aₙ ∪ aᵢ;
until Sep(A⁰) ≠ A⁰;
ωmax ≔ n;
```

Здесь $Sep\left(A^0\right)$ – множество блоков разбиения, формируемое без учета ограничений и значений весовой функции; $A^0 \setminus Sep\left(A^0\right)$ – множество еще не рассмотренных вершин. Асимптотическая временная сложность рассмотренного выше алгоритма составляет $O\left(\omega_{\max} N^2\right)$, емкостная – $O\left(N\right)$ (для хранения множества блоков разбиения требуется не более чем N ячеек памяти).

В сравнении с алгоритмом [2] предлагаемый алгоритм использует тот же состав структур данных и предварительных преобразований (ациклическая граф-схема алгоритма без «пустых» дуг, матрица отношений), алгоритму на основе вычисления ω-мощности базового сечения [1] помимо ациклической граф-схемы алгоритма без «пустых» дуг необходимы система R-выражений и базовое сечение (но не требуется матрица отношений). Для вычисления степени параллелизма алгоритма управления с $N = 100$ вершинами программная реализация алгоритма с использованием ω-мощности затрачивает в среднем 87 мс, предложенный алгоритм – 79 мс (70 мс из которых уходит на предварительные преобразования и построение матрицы отношений), т.е. имеет место выигрыш во времени обработки на 9,2%. При $N = 1000$ среднее время работы алгоритма с использованием ω-мощности составляет 20,2 с, в то время как предлагаемого – 1,7 с (из них 1,4 с – предварительные преобразования и построение матрицы отношений), выигрыш в 11,9 раза.

Таким образом, предложенный алгоритм обладает меньшей вычислительной сложностью и более прост в реализации (60 строк (2,1 КБ) исходного кода при программной реализации). Для графов общего вида, система бинарных отношений которых не подчиняется свойствам, рассмотренным в [1], алгоритм неприменим, т.е. свести решение NP-полной задачи выделения максимальной по включению клики к задаче выделения подмножества $\tilde{A}$, решаемой за полиномиальное время, не представляется возможным: сферой применения предлагаемого алгоритма являются граф-схемы корректных параллельных алгоритмов.

Литература

1. Организация и синтез микропрограммных мультимикроконтроллеров / И.В. Зотов, В.А. Колосков, В.С. Титов [и др.]. – Курск: КурскГТУ, 1999. 368 с.
2. Ватутин Э.И. Оценка степени параллелизма алгоритма с использованием матрицы отношений // Современные инструментальные системы, информационные технологии и инновации. Ч. 2. – Курск: КурскГТУ, 2006. С. 135–139.
3. Баранов С.И., Журавина Л.Н., Песчанский В.А. Обобщенный метод декомпозиции граф-схем алгоритмов // А и ВТ. 1982. № 5. С. 43–51.