УДК 650.21

РЕВОТЮК М.П., КУЗНЕЦОВА Н.В.

ОЦЕНКА ЭФФЕКТА РАСПАРАЛЛЕЛИВАНИЯ ЖАДНЫХ АЛГОРИТМОВ

Распараллеливание задач повышенной вычислительной сложности по кооперативной схеме использования ресурсов вычислительной сети порождает волновой процесс, характерный и жадным алгоритмам. В случае дефицита времени возникает задача управления ресурсами сети хотя бы на приближенных моделях. Цель исследования – оценка сокращения времени решения задач на сети жадным алгоритмом выбора из конечных множеств вариантов в предположении их равномерного распределения.

Жадные алгоритмы [1] основаны на решении локальных задач выбора вида

$$Z = \langle V, P, S \rangle, \tag{1}$$

где *V* – множество вариантов, подлежащих оценке по заданному критерию; *P* – процедура получения оценки качества для отдельного варианта из множества *V*; *S* – процедура реализации вычислительной схемы решения *Z*, определяющая порядок применения *P* к элементам множества *V*.

Пусть Z – задача минимизации целевой функции *F*, а процедура *P* обладает возможностью получения в процессе работы нижних оценок значения *F*.

Процедура *S* в (1), по существу, определяет порядок перечисления вариантов в последовательности $V = \{v_i\},\ i = \overline{1,n}$. Если время анализа каждого отдельного варианта v_i будет w_i, то время *W* выбора рационального варианта при условии их независимого последовательного анализа на одиночной ЭВМ составит

$$W = \sum_{i=1}^{n} w_i . \tag{2}$$

Предположим, что процедура *S* допускает разделение вариантов *V* на подмножества, время работы процедуры *P* характеризуется аддитивной зависимостью от количества итераций оценки отдельного варианта, а значения оценок целевой функции *F* связаны на множестве номеров итераций зависимостями

$$(F_{i,j} \le F_{i,j+k}) \wedge (L_{i,j} \ge L_{i,j+k}),\ k > 0,\ i = \overline{1,n}. \tag{3}$$

Здесь $F_{i,j}$ - текущая оценка целевой функции *F*, $L_{i,j}$ – нижняя граница оценки такой функции для варианта v_i на итерации *j*, $j = \overline{1,k(i)}$; $k(i)$ - количество итераций исчерпывающего анализа варианта v_i, когда выполняется условие $F_{i,k(i)} = L_{i,k(i)} = F(i),\ i = \overline{1,n}$.

Характерный для многих задач минимизации вид изменения оценок варианта v_i позволяет организовать рациональную схему вычислений, используя значение рекордной оценки *R* лучшего из ранее просмотренных вариантов,

$$R = \min_j \{F_j\},\ j = \overline{1,i-1}. \tag{4}$$

Итерации анализа варианта v_i можно прервать на шаге $m(i)$, если установлена истинность условия $L_{i,m(i)} < R$. Так как $m(i) \le k(i), i = \overline{1,n}$, то полагая $w_i \approx k(i)$, можно заметить, что время последовательного анализа всех вариантов с учетом ранее полученных рекордных оценок составит

$$W^* = \sum_{i=1}^{n} m(i) = \sum_{i=1}^{n} \min_j \{k(j), j \le i\}. \qquad (5)$$

Сравнивая (2) и (5), легко обнаружить, что $W^* \le W$, причем эффект сокращения времени решения в конечном счете зависит от порядка рассмотрения вариантов, т.е. от того, насколько рано выбираются хорошие варианты.

Можно выделить следующие случаи законов упорядочения вариантов с однозначно прогнозируемой характеристикой качества:

- наилучший случай – $k(i) < k(i+1),\ \overline{i = 1, n-1}$;
- наихудший случай – $k(i) > k(i+1),\ \overline{i = 1, n-1}$;
- случай неприменимости накопления опыта – $k(i) = k(i+1),\ \overline{i = 1, n-1}$.

Однако упорядочение вариантов априорно неизвестно, иначе можно было бы просто исключить заведомо неперспективные из рассмотрения. Очевидно, что для характеристики задачи интерес представляет интервал $[k_{\min}, k_{\max}]$, где

$$k_{\min} = \min_i \{k(i), i = \overline{1, n}\},\ k_{\max} = \max_i \{k(i), i = \overline{1, n}\}.$$

Выражение (5) не предписывает обязательность последовательного просмотра вариантов. Если процедура *S*, определяющая порядок перечисления вариантов в последовательности *V*, допускает образование *m* подмножеств вариантов, то ускорение моментов установления бесперспективности может быть достигнуто путем распараллеливания процесса решения между *m* отдельными ЭВМ при незначительной модификации процедуры *P*.

Модификации процедуры *P* состоит в том, что в процессе анализа вариантов каждая ЭВМ имеет доступ к значению оценки *F* наилучшего варианта из числа просмотренных на всех ЭВМ. Если какая-то из ЭВМ улучшает значение *F*, то она немедленно передает сообщение об этом остальным ЭВМ. Таким образом, итерации анализа вариантов выполняются при известном текущем значении рекордной оценки *R* согласно (4), которое может неоднократно улучшено в любой момент времени.

Пусть каждая ЭВМ получает для анализа некоторое подмножество вариантов. Для оценки вычислительных затрат на анализ всех вариантов рассмотрим наихудший случай их перечисления, когда $k(i) > k(i+1),\ \overline{i = 1, n-1}$. Если $k(1) = k_{\max}$, а $k(n) = k_{\min} = 0$, то (5) в этом случае можно представить в виде

$$W^* = k_{\max} \cdot n/2. \qquad (6)$$

Пусть подмножества вариантов образованы так, что $|V_i| = n/m,\ i = \overline{1, m}$. Тогда при параллельной работе *m* ЭВМ и синхронным рассмотрением отдельных вариантов время решения задачи *Z* определяется продолжительностью рассмотрения последнего подмножества. Его оценка получается масштабированием (6) вида $k_{\max} \leftarrow k_{\max}/m$, $n \leftarrow n/m$. Отсюда следует, что время решения задачи на *m* ЭВМ сокращается в m^2 раз.

Последний вывод основан на наихудших предположениях. Вследствие отсутствия информации о виде закона распределения оценок поступающих на анализ вариантов, предположения, лежащие в основе полученных выше результатов являются достаточно грубыми. Например, здесь предполагается однородность вычислительной среды на всем интервале решения задачи *Z* и не учитывается возможность образования очередей сообщений. Имея в наличии методы или программные средства решения задачи Z, определяемой (1), часто представляет интерес сравнение альтернатив использования мощной ЭВМ или кооперации маломощных ЭВМ.

Предположим, что множество вариантов V отображается на отрезок $x \in [0, b]$, а значения вычислительной трудоемкости их анализа – на отрезок $y \in [0, a]$.

Накопление опыта и время решения зависит от исходного упорядочения последовательности анализируемых вариантов. Наилучший случай, как отмечалось ранее, соответствует некоторой неубывающей, а наихудший – убывающей функции $y(x)$. Реальный вид такой функции априорно неизвестен. Однако предполагая ее линейный вид, возможные случаи упорядочения вариантов можно рассматривать относительно значения dy/dx для прямой, проходящей через точку $(a/2,\ b/2)$. Фиксация точки отражает намерение нормировки времени решения.

Общее время анализа вариантов с учетом накопления опыта в системе из m ЭВМ при рассматриваемых предположениях характеризуется зависимостью [2]:

$$t(m) = \begin{cases} bx/m, & 0 < x \le a/2; \\ \left(a - x + (2x - a)/2m\right)b/m, & a/2 < x \le a. \end{cases}$$

Легко заметить, что $t(m) \in [0,\ ab/2m]$. После усреднения по всем значениям x для случая его равномерного распределения (для простоты формулировки выводов) на интервале $[0, a]$ получим среднее время решения задачи Z на m ЭВМ:

$$\overline{t(m)} = ab\left(2 + 1/m\right)/8m.$$

Среднее время решения задачи Z при независимом рассмотрении вариантов

$$\overline{t^*(m)} = ab/2m.$$

Из отношения $\overline{t^*(1)}/\overline{t(1)}$ следует, что накопление опыта на одиночной ЭВМ сокращает время решения в среднем в 4/3 раза.

Относительный эффект распределения процесса между m агентами с обменом рекордными оценками

$$\frac{\overline{t(1)}}{\overline{t(m)}} = \frac{3m^2}{2m+1}.$$

Очевидно, что для $m > 1$ время решения сокращается от 2.4 до $\sim 1.5 \cdot m$ раз.

Полученные оценки характеризуют полезность распараллеливания жадных алгоритмов и могут быть использованы для обоснования вычислительной схемы решения задачи в условиях дефицита времени.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ/Пер. с англ. – М.: МЦМНО, 2002. – 960 с.

2. Тихомирова Е.В. Характеристики потоков анализируемых вариантов в кооперативных схемах//Моделирование и информационные технологии проектирования: Сб. научн. тр., вып. 4. – Мн.: Объединенный институт проблем информатики НАН Беларуси, 2002. – С.110-116.

Ревотюк Михаил Павлович
Доцент кафедры информационных технологий автоматизированных систем, канд. техн. наук
Белорусский государственный университет информатики и радиоэлектроники, г. Минск
Тел.: +375(17)239-84-62
E-mail: rmp@bsuir.unibel.by
Кузнецова Наталья Владимировна
Аспирант кафедры информационных технологий автоматизированных систем
Белорусский государственный университет информатики и радиоэлектроники, г. Минск
Тел.: +375(17)239-88-23
E-mail: kafitas@bsuir.unibel.by