

МОДЕЛИРОВАНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ ЭФФЕКТИВНОЙ ОРГАНИЗАЦИИ ПРОЦЕССА ОБРАБОТКИ ИЗОБРАЖЕНИЙ НА МВС

Аннотация. Предложена модель обработки изображений, позволяющая заранее оценить эффективность выполнения на многопроцессорной вычислительной системе и изучить влияние на время обработки таких параметров конфигурации системы и организации процесса обработки, как число мультипроцессоров и размер фрагмента данных.

Ключевые слова: моделирование, сеть Петри, мультипроцессор.

Введение. Задачи обработки аэрокосмических изображений, намного превышающих объем оперативной памяти, требуют высокой скорости вычислений и выполняются на многопроцессорных вычислительных системах (МВС) типа одна команда - много данных (ОКМД) [1]. Время обработки в мультипроцессоре ОКМД-типа зависит от числа процессорных элементов (ПЭ) и размещения данных в памяти. Существуют МВС содержащие несколько мультипроцессоров, а также реконфигурируемые системы, у которых может изменяться число мультипроцессоров и количество ПЭ в них. Это делает актуальным разработку организации процесса обработки изображений на МВС с различными конфигурациях системы.

Постановка задачи. Необходимо разработать модель обработки изображений с учетом пространственного параллелизма и параллелизма задач для эффективного выполнения на МВС.

Основная часть. Для МВС, содержащей мультипроцессоры ОКМД-типа, естественно использовать двухуровневую систему программирования, где на нижнем уровне на специализированных языках программируются алгоритмы обработки, а на верхнем уровне организуется общий процесс вычислений на языке управления заданиями.

Процесс обработки изображений, являющихся большими объектами данных, предполагает разбиение их на фрагменты и пофрагментную обработку, которая позволяет применить иерархическую ор-

ганизацию вычислительного процесса: выполнение процедуры - выполнение задачи - выполнение задания [2].

Задание состоит из ряда задач, каждая из которых обрабатывает объект данных согласно одному алгоритму. Задача содержит прикладную процедуру, выполняющую алгоритм обработки над фрагментом данных в памяти мультипроцессора, и вспомогательные процедуры, обеспечивающие выполнение этого алгоритма над всем множеством фрагментов входного объекта данных. Таким образом, в задании выполняется множество процедур над множеством фрагментов изображения, причем возможно их одновременное выполнение.

Процесс обработки представлен следующим образом:

$$Y = \bigcup_{k=1}^L \left(\bigcup_{i=1}^{N_k} \left(\bigcup_{j=1}^{M_k} P_{k,j}(X_i) \right) \right),$$

где \bigcup - представляет объединение, X_i - фрагмент входного объекта данных, Y - выходной объект данных, L - число задач в задании, N_k - количество фрагментов для задачи k , M_k - число процедур в задаче k , $P_{k,j}$ - процедура j задачи k .

Управление выполнением задания базируется на методе асинхронного программирования. Каждой процедуре ставится в соответствие условие инициирования – спусковая функция, представляющая собой логическую функцию от управляющих переменных. В качестве управляющих переменных используются логические переменные процедур, наборов данных и параметров, а так же специальные переменные.

Предлагается модель задания обработки изображения, в которой определяются условия запуска процедур. В качестве основы модели задания выбрана модель параллельных вычислений на базе расширенной сети Петри РСМ [3]. В предлагаемой модели, названной параметризированной сетью Петри, явно добавлено множество параметров конфигурации, задающее параметризацию переходов, мест, а также соответствующих им фишек. Правила срабатывания переходов модифицируются с учетом параметризации.

Параметризованная сеть Петри определяется как набор

$$PN = (T, P, F, M_0, C, S, R, D, K),$$

где T, P, F, M_0 имеют тот же смысл, что и в обычной сети Петри [4],

C - множество управляющих переменных,

S- множество спусковых функций,

R- множество процедур модификации управляющих переменных,

D- множество времен срабатывания переходов.

K- множество параметров конфигурации.

С каждым переходом t_i связано множество управляющих переменных s_i , спусковая функция s_i , процедура перехода r_i , время срабатывания перехода d_i и параметр конфигурации k_i . Спусковая функция s_i определяется на управляющих переменных и используется для определения готовности перехода к срабатыванию по правилам срабатывания. Процедуры переходов так же оперируют управляющими переменными и используются для их модификации. Время срабатывания определяет задержку, связанную с переходом, и может представлять функцию от управляющих переменных.

Любой переход (или место) может быть параметризован соответствующим ему параметром конфигурации. Параметризованные места содержат фишки, пронумерованные значениями из множества этого параметра конфигурации. Одним из параметров конфигурации выступает число мультипроцессоров в МВС.

Определим правила срабатывания переходов модели. Переход может находиться в трех состояниях:

- готовности,
- неготовности,
- срабатывания.

Переход готов к срабатыванию, если все его входные места содержат хотя бы по одной фишке (ненумерованной или с номером из множества параметра перехода), а спусковая функция, соответствующая переходу, является истинной.

Готовый к срабатыванию переход входит в состояние срабатывания и из каждого его входного места изымается по одной фишке.

Переход остается в состоянии срабатывания промежутки времени, определенный его временем задержки. В конце этого периода в каждое его выходное место добавляется по одной фишке и выполняется соответствующая процедура модификации. Для параметризованного перехода фишка имеет тот же номер, что и входная фишка перехода. После этого переход попадает в неготовое состояние.

Состояние в любой момент определяется текущей разметкой сети, вектором управляющих переменных и вектором «оставшегося времени срабатывания» переходов. Разметка сети определяет состояние управления, а вектор управляющих переменных – состояние данных.

Построим модель задания в терминах параметризированной сети Петри следующим образом.

1. Модель представляет собой сеть $PN=(T, P, F, M_0, C, S, R, D, K)$.

2. Каждой процедуре задания соответствует переход t_i в сети PN , $t_i \in T$.

3. Логической переменной набора данных, параметра, процедуры или управляющей логической переменной, используемой в спусковой функции, соответствует место p_j , $p_j \in P$.

4. Если логическая переменная, соответствующая месту p_j , используется в спусковой функции процедуры, соответствующей переходу t_i , то существует дуга $F(p_j, t_i)$, исходящая из p_j и входящая в t_i .

5. Если в выходе процедуры, соответствующей переходу t_i , указана логическая переменная, соответствующая месту p_j , то существует дуга $F(t_i, p_j)$, исходящая из t_i и входящая в p_j .

6. Начальная разметка сети M_0 представляет собой вектор $M_0=\{m_1, m_2, \dots, m_p\}$. Количество фишек m_j в месте p_j равно значению управляющей переменной c_j для не параметризованного места и $c_j \cdot (k_{j2}-k_{j1}+1)$ (т.е. значению управляющей переменной c_j умноженному на $k_{j2}-k_{j1}+1$ значений параметра конфигурации), если место параметризовано со значением параметра конфигурации $k_j = \overline{k_{j1}, k_{j2}}$.

7. Множество параметров конфигурации $K=\{k_1, k_y, \dots, k_x\}$, определяется областями значений параметров конфигурации $k_j = \overline{k_{j1}, k_{j2}}$. Параметризация переходов и мест задается двумя множествами: $K_t=\{k_{t1}, k_{t2}, \dots, k_{tt}\}$ и $K_p=\{k_{p1}, k_{p2}, \dots, k_{pp}\}$, где в позиции $k_{ti}(k_{pi})$, соответствующей переходу t_i (месту p_j), стоит 0 – если переход (место) не параметризован или параметр $k_{ti}(k_{pi})$ – для параметризованного перехода (места).

8. Число фрагментов набора данных является одной из специальных управляющих переменных. В спусковой функции перехода, соответствующей прикладной процедуре, присутствует проверка на

неравенство нулю этой специальной переменной. После обработки одного фрагмента она уменьшается на единицу в процедуре модификации управляющих переменных перехода.

9. Время выполнения процедуры, соответствующей переходу t_i , задается как функция от размера фрагмента данных и числа ПЭ.

Пример параметризованной сети Петри для задания трансформирования изображения по многочленам приведен на рис. 1.

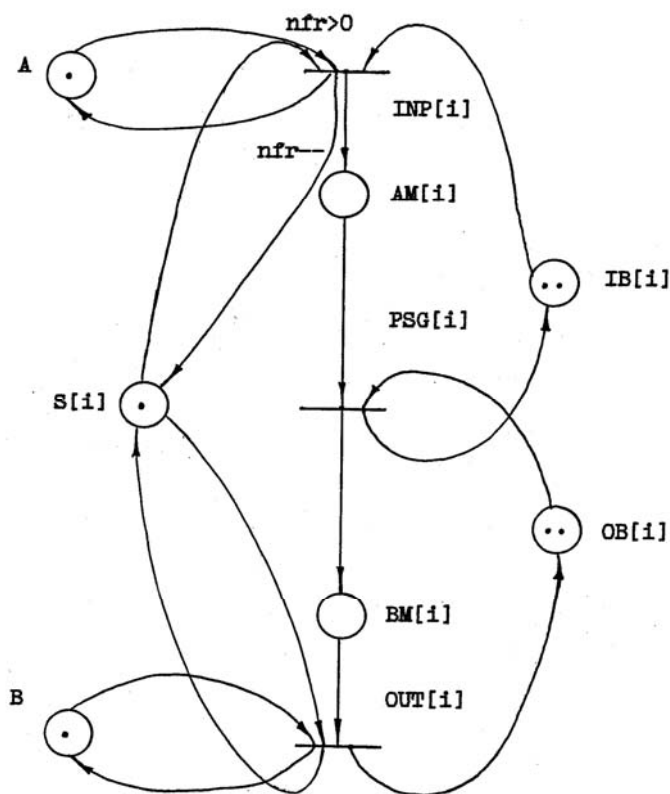


Рисунок 1 - Пример параметризованной сети Петри

Для выбора эффективной организации процесса обработки изображений на МВС проводится моделирование вычислений предложенной параметризованной сетью Петри, поддерживающей временные атрибуты переходов, а также зависимости их срабатывания от данных и от конфигурации МВС. С ее помощью изучается влияние размеров фрагментов изображения на время выполнения задания при разном числе ПЭ в мультипроцессоре и разном количестве мультипроцессоров в МВС.

Для задач предварительной обработки изображений проведенное моделирование позволило выбрать оптимальные размеры фрагментов, при которых время выполнения задания минимальное. Сокращение времени обработки в среднем составляет 15-18%.

Вывод. Предложена модель задания, предоставляющая возможность заранее оценить эффективность его выполнения системой определенной архитектуры и изучить влияние на длительность обработки таких параметров конфигурации системы и организации процесса обработки, как число мультипроцессоров и размер фрагмента данных.

ЛИТЕРАТУРА

1. Головкин Б.А. Параллельные вычислительные системы.- М.: Наука, 1980.- 459 с.
2. Пономарев И.В., Федотов А.Г. Структурные свойства данных и методы построения программного обеспечения реконфигурируемых многопроцессорных систем с общим управлением.- М.: Вопросы кибернетики, 170, 1991, с. 132-158.
3. Adiga A.K., Browne J.C. A graph model for parallel computations expressed in the Computation Structure Language// Proc. of the 1986 Intern. Conf. on Parallel Processing, August 19-22. 1986. P. 880-886.
4. Питерсон Дж. Теория сетей Петри и моделирование систем.- М.: Мир, 1984.- 264 с.

Получено 19.01.2011г.