РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ИНФОРМАЦИИ
(Вычислительные системы)
1984 год Выпуск 105

УДК 681.325

О РАСПРЕДЕЛЕНИИ ПАКЕТА ЗАДАЧ ПО МАШИНАМ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Ю.Н.Потапова

1. Введение

Цель работы - исследование алгоритмов децентрализованного распределения пакета простых задач по машинам вычислительной системы (ВС) с программируемой структурой [1,2]. Под пакетом простых задач [1] будем понимать конечное множество равноприоритетных задач, каждая из которых решается на одной машине системы.

Вычислительные системы с программируемой структурой состоят из элементарных вычислительных машин (ЭМ), объединяемых программно-управляемой сетью связи [3]. Задачи поступают в систему через подключенные к машинам внешние устройства. Распределение внешних устройств по системе определяется на основе максимизации вероятности доступа к ним и/или минимизации времени обращения к ним [4]. Одной из основных проблем управления ВС является проблема упорядочения происходящих в ней процессов и, в частности, диспетчирование задач. Диспетчирование заключается в составлении расписания решаемых на системе задач. Под расписанием понимается закрепление задач за машинами ВС и задание очередности обработки задач в пределах каждой машины. Составление расписания может преследовать различные цели: предотвращение конфликтов при распределении ресурсов, рациональное использование памяти, уменьшение числа машин системы для решения заданных задач, сокращение времени решения задач на системе определенного размера и т.д.

В том случае, когда в системе одна из машин выделена для целей диспетчеризации, имеем систему с централизованным управлением. Если же в системе функции диспетчеризации распределены между несколькими (в пределе - между всеми) машинами, имеем систему с децентрализованным управлением.

При централизованном управлении для составления и реализации расписания управляющей машине необходимо собрать сведения о всех задачах, находящихся в системе. Затем обработать эти сведения в соответствии с принятым алгоритмом получения оптимального (минимального по времени его реализации) расписания. На основании результата работы этого алгоритма принимается решение о закреплении задач за определенными машинами. Далее на этапе реализации расписания выполняется рассылка задач по системе в соответствии с этим решением. Отсутствие общих ресурсов (общей памяти, общего устройства управления) в вычислительных системах с программируемой структурой приводит к тому, что в ВС с централизованным управлением в работу по сбору сведений о задачах и перемещению задач по ВС вовлекаются все машины. Централизованное управление накладывает ограничение на производительность ВС вследствие конечной производительности управляющей машины и ограниченной пропускной способности сети связи системы. Отмеченные недостатки централизованного управления вычислительными системами с программируемой структурой приводят к необходимости поиска средств децентрализованного управления.

Оптимальное расписание может быть получено точными методами математического программирования [I]. Однако в связи с трудоемкостью этих методов время составления расписания может превысить выигрыш от его реализации. Как правило, эти методы используются при статическом диспетчировании, когда расписание составляется до начала счета задач. Они практически не могут быть использованы при оперативном управлении системой, когда планирование осуществляется в процессе счета задач, а также в условиях сбоев и отказов машин системы. В этом случае необходимо динамическое диспетчирование.

Поэтому на практике используются эвристики [I,5-7], которые в большинстве случаев дают приемлемые решения, близкие к оптимальным. Они построены с учетом времени решения [I,5,6] задач, штрафа за задержку решения задач [I], директивных сроков окончания решения задач [6]. Эвристические алгоритмы [I,5,6] основаны на предпосылках: I) время пересылки задач между машинами системы пренебрежимо мало в сравнении со временем счета задач и не учитывается при распределении задач; 2) параметры всех задач считаются известными до начала их распределения.

В [7] рассматривается децентрализованный эвристический алгоритм диспетчеризация пакета простых задач, в котором каждая задача характеризуется временем решения и временем ее пересылки между двумя соседними машинами. Алгоритмом учитываются затраты времени на распределение и рассылку задач по машинам системы. Алгоритм не предполагает перераспределения задач, назначенных на счет к текущему моменту времени. Это свойственно динамическим алгоритмам. Цель алгоритма - сокращение времени решения пакета на системе, состоящей из заданного числа машин. Как показывают результаты моделирования алгоритма, отклонение времени решения пакета от минимального времени (для систем с рассмотренными параметрами) составляет приблизительно 10%. Минимальное время определяется суммой времен решения всех задач пакета, отнесенной к числу машин системы. Сокращение времени решения пакета осуществляется выравниванием времени работы машин ВС. Время занятости машины определено как ее нагрузка. Суть этого алгоритма заключается в том, что при распределении каждой задачи нагрузка по машинам выравнивается в пределах некоторой области, т.е. локально. Каждая ЭМ может самостоятельно принимать решение относительно размещения по системе находящихся в ней задач. Машина - распределитель задачи сравнивает нагрузку машин области, центром которой она (ЭМ) является. Задача пересылается в наименее нагруженную машину области. Решается задача в наименее нагруженной машине, которая является центром области. Размер области задается радиусом r, который определяется как минимально возможное количество межмашинных связей, соединяющих ЭМ - центр области с наиболее удаленной машиной области. В дальнейшем по отношению к этому алгоритму будем использовать термин HOLE-лунка. Значение нагрузки машин в [7] определяется вычислительными затратами конкретных машин - их абсолютной нагрузкой - K_a. Абсолютная нагрузка ЭМ определяется как сумма времен решения задач, считаемых ею, плюс время на обзор окрестности радиуса r по каждой из распределяемых ею задач.

Алгоритмом HOLE выполняется принцип близкодействия: лучшие результаты (для рассмотренных в [7] вариантов) по времени решения пакета алгоритм дает при выравнивании нагрузки по области радиусом $r = 1$. Это значит, что для принятия оптимального решения каждой ЭМ достаточно сведений о состоянии загруженности ее непосредственных соседей. Задачи обрабатываются в порядке расположения в пакете. Предварительная подготовка пакета не требуется, поэтому сведения о временных параметрах всех задач к началу распределения пакета не нужны.

Сочетание принципов близкодействия, децентрализации и динамичности по управлению позволяет при распределении задач "обходить" неисправные ЭМ. Для этого достаточно идентифицировать неисправность бесконечно большой нагрузкой ($K_a = \infty$) . Алгоритм HOLE не будет пересылать задачи в направлении неисправной ЭМ в соответствии со своей функциональной организацией.

Недостаток данного алгоритма заключается в том, что при $r = 1$ не учитывается тенденция изменения нагрузки по системе. Поэтому среднее число пунктов пересылки каждой задачи сравнительно велико - для исследованных вариантов порядка 3,5.

В настоящей работе исследуется алгоритм HOLE, в котором нагрузка системы в каждой ее машине представляется некоторым усредненным по области r значением. Это позволяет при распределении задач учесть тенденцию изменения нагрузки и сократить число "блужданий" задач по системе.

2. Постановка задачи

Пусть ВС представлена графом $G = (N, M)$ межмашинных связей, вершинам N которого сопоставлены машины, а ребрам M - линии связи между машинами. По определению ВС количество и пропускная способность линий связи у всех машин одинаковы.

Имеется конечное множество $I = \{I_i\}$, $i = \overline{1,L}$, простых задач; значение $L \gg N$. Известны значения $T_1 = \{t_i\}$, $i = \overline{1,L}$, времен решения задач из $\{I\}$, а также значения времен пересылки задач между двумя соседними машинами, задаваемые множеством $T_2 = \{\tau_i\}$, $i = \overline{1,L}$.

Перечень внешних устройств, через которые пакет поступает в систему, задан множеством номеров машин $H = \{h_i\}$, $h_i \in \{1, 2, \ldots, N\}$, к которым они подключены.

Быстродействие системы оцениваем показателем θ: $\theta = (T - T_{min})/T_{min}$, где T - время решения пакета, $T_{min} = (\sum_{j=1}^{L} t_j)/N$.

Распределение задач пакета по системе осуществляется алгоритмом HOLE . Задачи обрабатываются (и распределяются, и считаются) в порядке расположения в пакете.

С целью сокращения пути от машины-распределителя пакета к машине, считающей задачу, предлагается характеризовать загружен -

ность системы в каждой ее точке, кроме ЭМ - центра окрестности, усредненным значением нагрузки $K_{r_j}^{(l)}$ по области r, центром которой данная $ЭМ_j$ является; l - глубина усреднения:

$$K_{r_j}^{(l)} = [\sum_{i=1}^{n_r} (K_r^{(l-1)})_i]/n_r,$$

где n_r - количество ЭМ в области r; l и r - независимые величины. Фиксируя r и увеличивая l, получаем сведения о загруженности области бо́льших размеров, чем размеры, заданные r. При $l = I$ значение $K_{r_j}^{(l-1)} = K_{a_j}$, т.е. равно абсолютному значению загруженности машины. Значение $l = 0$ соответствует алгоритму выравнивания нагрузки по абсолютному значению K_a загруженности машин. При распределении задач нагрузка ЭМ-центра окрестности при любых l определяется значением K_a.

Алгоритм выравнивания нагрузки по параметру $K_r^{(l)}$ обозначим через $HOLE_l$, а соответствующие ему параметры через $X^{(l)}$.

В конкретном случае считаем

$$K_{a_i} = \sum_{j=1}^{n_1} t_j + n_2 \cdot T_{0r},$$

где n_1, n_2 - количество задач, считаемых и распределяемых машиной $ЭМ_i$ соответственно, T_{0r} - время анализа состояния загруженности машин окрестности радиуса r.

Качество алгоритма $HOLE_l$ (в функции от l) оцениваем I) быстродействием системы, которое определяем через θ; 2) средним числом S_t межмашинных пересылок задач по системе. Число пересылок S_t определяем как $S_t = S/L$, где S - суммарное количество межмашинных пересылок задач пакета, осуществляемое алгоритмом при распределении задач по ВС.

Исследование алгоритма проводится методом имитационного моделирования. Моделирующий алгоритм построен по принципу Δt [8]. При этом процесс функционирования системы рассматривается как последовательная смена состояний. Временной интервал анализа состояния системы равен Δt.

3. Описание модели

Модель строится с учетом следующих предположений:

Времена решения и пересылки задач из $\{T_1\}$, $\{T_2\}$ задаются случайным образом, закон распределения показательный. Плотности обозначим соответственно через λ_1, λ_2.

Время, затрачиваемое машиной на обзор окрестности радиусом r глубиной l, определяется как $T_{0r}^{(l)} = T_{0r} + T_{01} \cdot l$, где $T_{0r} = T_0(n_1+2n_2+\ldots+qn_q+\ldots+rn_r)$, здесь T_0 – время передачи между двумя машинами показателя загруженности одной из них, n_q – количество машин, находящихся на расстоянии q межмашинных связей от центра окрестности, следовательно, $T_{01} = T_0 \cdot n_1$.

Одновременно каждая машина может или считать, или распределять только одну задачу.

Пересылка задачи I_j совмещается во времени со счетом задачи I_k $(k \neq j)$ или обзором окрестности для определения машины, принимающей задачу I_i $(i \neq j, k)$, т.е. время пересылки задач не включено в общее время работы системы.

Пересылка задач между машинами осуществляется по кратчайшему пути, связывающему эти машины.

Приоритетность машин по размещению задач в системе определяется порядковым номером машины.

Состояние системы по нагрузке при рассылке задач изменяется по мере перераспределения задач, при решении задач – дискретно через такт Δt работы модели.

По каждой из N машин системы имеем две очереди: Q – распределяемых и R – решаемых задач. Каждая задача, стоящая в очереди Q, характеризуется тремя параметрами: t_j, τ_j, t_{pj}; t_j, τ_j определены выше, t_{pj} – время, отмечающее момент готовности j-й задачи к обработке. Очередью R решаемых задач параметр τ_j игнорируется.

Очереди Q и R просматриваются последовательно по машинам от первой до N-й. Очередь Q распределяемых задач имеет старший приоритет. Признаком окончания решения набора задач является нулевое значение длин очередей решаемых и распределяемых задач по всем машинам системы.

4. Результаты моделирования

В результате моделирования определяются параметры θ и S_t. Последний параметр позволяет выделить и оценить затраты алгоритма на выдачу (прием) задач машинами системы.

Моделирующая программа реализована на ФОРТРАНе. Программа моделирования системы, содержащей максимум 100 машин, решающей максимум 1000 задач, требует для своей реализации (с использованием личной библиотеки ОС ЕС) 182К оперативной памяти машины.

Численные результаты получены для двумерной системы, состоящей из 30 машин. Отношение числа L распределяемых задач к числу N машин системы принято равным 10. Отношение λ_2/λ_1 выбирается из среднего отношения количества циклов в программе к длине программы, равного одному из значений $\{50, 75, 100, 200\}$. Модель считалась для $\lambda_1 = 10$. Соотношения интенсивности пересылки задач λ_2 и времени T_0 обзора состояния загруженности одной из соседних машин принято равным $T_0 = 0{,}1\ \lambda_2^{-1}$, что соответствует длинам задач в несколько десятков и сотен команд. Значение Δt принято равным $3/4\ T_{01}$.

В связи со сложностью практической реализации модели (один эксперимент для $l = \overline{0,4}$ длится на ЕС 1050 40–80 мин) качество предлагаемого алгоритма целесообразно оценивать по некоторому числу экспериментов.

Результаты моделирования для соотношения $\lambda_2/\lambda_1 = 200$ сведены в таблицу. Значения $\theta^{(0)}$ и $S_t^{(0)}$ представлены для случая, когда выравнивание нагрузки ведется по абсолютному показателю загруженности машин – алгоритм $HOLE_0$. Зависимость быстродействия алгоритма $HOLE_l$ от l $(l = 1,\ldots,4)$ представлена разницей $\theta^{(l)}-\theta^{(0)}$. В таблице приведено соотношение количества пересылок задач $S_t^{(l)}/S_t^{(0)}$ для тех же значений l. Результаты являются средними по приведенному в последнем столбце числу опытов. Увеличение числа опытов от 10 до 40 уточняет результаты моделирования. Характер соотношения параметров θ и S_t при этом сохраняется.

Для исследованных вариантов использование алгоритмом выравнивания нагрузки усредненного показателя загруженности ЭМ – $K_r^{(l)}$ – в ряде случаев значительно сокращает число пересылок задач по сравнению с алгоритмом выравнивания нагрузки по абсолютному показателю загруженности K_a. Вывод справедлив для сравнительно большого порядка $N/2$, числа U точек ввода пакета в систему. Лучшие результаты по времени решения пакета и количеству пересы-

Т а б л и ц а

Число точек U – ввода пакета в ВС (номера ЭМ, к которым они подсоединены)	$\theta^{(0)}$	$\theta^{(1)} - \theta^{(0)}$				$s_*^{(0)}$	$s_*^{(1)}/s_*^{(0)}$				Число опытов
	Глубина усреднения 1										
	0	I	2	3	4	0	I	2	3	4	
I(любая)	0,329	0,204	0,056	0,049	0,058	3,92	7,62	I,I8	I,24	I,30	I0
2(I,I4)	0,305	0,II2	0,025	0,042	-	3,32	I0,39	I,26	I,06	-	I0
I0(I,4,...,28)	0,3I4	0,III	0,0I2	0,026	-	3,38	6,2	0,48	0,50	-	I0
I5(через I)	0,274	0,I2I	0,088	0,094	0,003	3,37	3,4I	0,38	0,44	0,43	I0
30(все)	0,3I4	-0,005	-0,04I	0,0I9	0,034	3,24	I,90	0,22	0,24	0,23	I0
-"-	0,328	-0,006	-0,022	0,003	0,004	3,I2	2,4I	0,32	0,25	0,26	I9
-"-	0,326	-	-0,0II	-0,009	0,005	3,I3	-	0,34	0,24	0,24	40

лок задач получаем при $l = 2$. Число пересылок задач при этом уменьшается по сравнению с алгоритмом $HOLE_0$ в 2-4 раза (столбец 9). Время решения пакета по алгоритму $HOLE_2$ сравнимо, а в ряде случаев меньше времени решения по алгоритму $HOLE_0$ (столбец 4).

Напомним, что одно из допущений модели заключается в совмещении счета или распределения задач с вводом/выводом задач в/из машины. В реальных условиях, когда нельзя пренебрегать затратами времени на ввод/вывод задач в/из машины, сокращение числа пересылок задач (при $l \geq 2$) даст выигрыш во времени (по сравнению с $l = 0$) пропорциональный сокращению числа пересылок задач по системе. В конкретном случае затраты на пересылку задач сократятся в 2-4 раза. При этом упростится ряд вопросов, связанных с хранением и обработкой транзитной информации: сократится объем буферной памяти, длина очередей, число конфликтов в сети связи и т.д.

5. Заключение

Таким образом, если число внешних устройств, через которые пакет задач поступает в ВС с рассматриваемыми параметрами, сравнительно велико, порядка половины от числа машин системы, то при распределении задач по системе целесообразно использовать алгоритм локального выравнивания усредненной нагрузки машин с глубиной усреднения равной двум. Для систем с рассмотренными параметрами число пересылок задач при этом сокращается в 2-4 раза по сравнению со случаем, когда распределение выполняется алгоритмом локального выравнивания абсолютной нагрузки. При малом числе точек ввода пакета в ВС, порядка десятой части от числа машин системы, целесообразно использовать алгоритм выравнивания абсолютной нагрузки машин.

В обоих случаях при распределении задач выполняется принцип близкодействия, т.е. каждой машине для принятия оптимального решения достаточно сведений о состоянии загруженности ее непосредственных соседей. Распределение задач осуществляется в динамическом режиме. Предварительная подготовка пакета не требуется. Алгоритм позволяет "обходить" неисправные машины системы.

Литература

1. ЕВРЕИНОВ Э.В., ХОРОШЕВСКИЙ В.Г. Однородные вычислительные системы. - Новосибирск: Наука, 1978. - 319 с.

2. КОРНЕЕВ В.В., ХОРОШЕВСКИЙ В.Г. Архитектура вычислительных систем с программируемой структурой. - Новосибирск, 1979. - 48с. (Препринт: ИМ СО АН СССР, ОВС-10).

3. ЕВРЕИНОВ Э.В., КОСАРЕВ Ю.Г. Однородные универсальные системы высокой производительности. - Новосибирск: Наука, 1966. - 308с.

4. КОРНЕЕВ В.В. О подключении внешних устройств к однородным вычислительным системам. -В кн.: Вычислительные системы. Вып. 63. Теория однородных вычислительных систем. Новосибирск, 1975, с. 103-108.

5. БАРСКИЙ А.Б. Планирование параллельных вычислительных процессов. -М.: Машиностроение, 1980. - 190 с.

6. ВАЙРАДЯН А.С., КОРОВИН А.В., УДАЛОВ В.Н. Эффективное функционирование управляющих мультипроцессорных систем. -М.: Радио и связь, 1984. - 326 с.

7. ПОТАПОВА Ю.Н. Сопоставление способов распределения пакета задач по машинам вычислительной системы. - В кн.: Вычислительные системы с программируемой структурой (Вычислительные системы, вып. 94). Новосибирск, 1982, с. 59-69.

8. БУСЛЕНКО И.П. Моделирование сложных систем. -М.: Наука, 1968. - 355 с.

Поступила в ред.-изд.отд.
3 июля 1984 года