

УДК 004.021

ПЛАНИРОВАНИЕ ЗАДАЧ В РАСПРЕДЕЛЕННЫХ ГЕТЕРОГЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Богданова О. В., Ступина А. А., Корпачева Л. Н., Юронен Е. А., Федорова А. В.,
Ежеманская С. Н.

ФГАОУ ВПО «Сибирский федеральный университет», Красноярск, Россия (660041, г. Красноярск, пр. Свободный, 79), e-mail: saa55@rambler.ru

Рассматривается проблема планирования задач, возникающих при обработке информации и управлении в распределенных гетерогенных информационных системах. Гетерогенная распределенная система управления процессами обработки данных строится на базе синтеза разнородных аппаратно-вычислительных и программных компонентов, а также средств управления. Гетерогенные информационные системы облегчают интеграцию разнородных информационных источников структурированных, слабоструктурированных и неструктурированных данных. В современных системах для реализации управления процессами обработки информации требуется организовать их распределенное включение в общую информационно-вычислительную среду через сети. В работе предложен алгоритм формирования плана задач обработки данных с минимальным числом процессоров. Рассмотрены эвристические алгоритмы планирования задач обработки данных и проведено их относительное сравнение при использовании в гетерогенных системах обработки данных. Раскрыты преимущества наличия в информационной системе механизмов конвейерного выполнения задач.

Ключевые слова: гетерогенная система, распределенная обработка данных, алгоритмы планирования, конвейерный план.

TASKS PLANNING IN THE DISTRIBUTED HETEROGENEOUS INFORMATION SYSTEMS

Bogdanova O. V., Stupina A. A., Korpacheva L. N., Yuronen E. A., Fedorova A. V.,
Ezhemanskaya S. N.

Siberian federal university, Krasnoyarsk, Russia (660041, Krasnoyarsk, Svobodnii av. 79), e-mail: saa55@rambler.ru

The problem of the tasks planning arising at information processing and management in distributed heterogeneous information systems is considered. The heterogeneous distributed control system of processes of data handling is under construction on the basis of synthesis of diverse hardware and computing and program components, and also control facilities. Heterogeneous information systems facilitate integration of diverse information sources of the structured, semistructured and unstructured data. In modern systems for realization of management by processes of information handling it is required to organize their distributed inclusion in the general information environment through networks. In work the algorithm of formation of the plan of problems of data processing with the minimum number of processors is offered. Heuristic algorithms of planning of problems of data processing are considered and relative comparison when using in heterogeneous systems of data processing is carried out them. Advantages of existence in information system of mechanisms of conveyor performance of tasks are opened.

Keywords: heterogeneous system, the distributed data processing, algorithms of planning, the conveyor plan.

Введение

Современный подход к созданию распределенных программно-информационных технологий для корпоративных и производственных структур основан на концепции гетерогенных систем, состоящей в объединении в единую систему или сеть множества разнородных обрабатывающих средств (процессоров), средств хранения, обработки информации и средств управления (разноплатформенные СУБД, различные учетные системы и т. д.). При этом воз-

никает необходимость сформировать необходимый набор инструментов и политик, который позволил бы минимизировать ключевые риски используемых гетерогенных систем на всех стадиях и фазах жизненного цикла, в частности, при планировании задач обработки информации. При этом необходимо учитывать, что в распределенных системах режим реального времени предполагает лимитирование времени ответа системы управления на запрос объекта. Ограничение на время реакции связывается в этом случае с выполнением периодических действий [5, 6].

Таким образом, при реализации периодичных задач формирование алгоритмов распределенной обработки информации и управления должно осуществляться с учетом ограничений, представленных в форме классов ресурсов, жесткого регламента задач и временных пределов реализации задач.

Методы и материал исследования

Рассмотрим существующие ограничения на классы ресурсов. Решения, предложенные ранее в [1,3,4], были связаны, прежде всего, с распределением процессоров. Вычислительные ограничения выражались в терминах времени выполнения и отношений предшествования.

Предлагаемая модель расширяет понятие стандартной модели, состоящей из множества r задач неравной длительности, связанных отношением предшествования и выполняемых на неприоритетной основе набором из n идентичных процессоров. В [6] отмечается, что проблема планирования зависимых задач очень сложна, нахождение ее оптимального решения требует больших вычислительных ресурсов, сравнимых с теми, которые требуются для собственно выполнения задач управления. Отметим, что при подходе, рассматриваемом в [6], планирование приближается к статическому варианту.

В нашем случае дополнительно предполагается наличие множества ресурсов $R = \{R_1, \dots, R_S\}$. Если задаче T_j необходим ресурс R_i , то это требование принимается во внимание в течение всего периода выполнения задачи. Потребность задачи T_j в ресурсе R_i обозначается через p_{ij} ($0 \leq p_{ij} \leq 1$).

Пусть $r_i(t)$ обозначает общее количество ресурсов R_i , которое используется в момент времени t . Тогда $r_i(t) = \text{Sum}(p_{ij})$ для всех T_j , выполняемых в момент времени t . Основная проблема заключается в обосновании критерия, отражающего степень влияния различных списков планов для этой модели на время завершения задачи (w).

Предположим, что для двух произвольных списков L и L' расширенная система из n процессоров выполняет набор из r задач с результирующими временами завершения w и w'

соответственно. Для такой среды предлагается ряд решений, которые дают следующие результаты:

- 1) $R = \{R_1\}$ (в системе существует только один вид ресурсов, отличных от процессора) – $w / w' \leq n$;
- 2) $R = \{R_1\}$ и независимости всех задач – $w / w' \leq (3 - 1 / n)$;
- 3) $R = (R_1, R_2, \dots, R_S)$, независимости задач и $n \geq r - w / w' \leq S + 1$.

В ходе исследования предложенной модели в гетерогенных системах были получены ограничения на количество задач, число процессоров и правила формирования списка используемых статических планов. Также было показано, что предложенные алгоритмы снижают эффективность, когда устраняются ограничения на ресурсы.

Классическим примером планирования независимых задач для жестких систем реального времени с одним процессором является алгоритм, разработанный Лью и Лейландом [9]. Этот алгоритм является динамическим, он использует вытесняющую многозадачность и основан на относительных статических (неизменяемых в течение жизни задачи) приоритетах. Целью является минимизация числа процессоров, требуемых для выполнения ряда задач при временных ограничениях на начало / конец выполнения заданий.

Пусть E_i – максимальное время выполнения одной итерации задачи J_i , а f_i – частота выполнения. Таким образом, каждой задаче J_i соответствуют два параметра $J_i: (f_i, E_i)$, $1 \leq i \leq n$, где n – количество включаемых в план задач.

Период повторения равен T_i , величине обратной f_i . Если n задач с J_1 по J_n распределены так, что $f_i > f_{i+1}$, то предполагается, что $f_i = 2f_{i+1}$. При этом к реализации допускаются задачи с любой частотой.

Рассматривая ограничения на время реакции системы, проанализируем периодические задачи первого класса (с бинарным частотным распределением). Множество задач, удовлетворяющих бинарному частотному распределению, приведено в табл. 1. На рис. 1 показаны задачи J_1 и J_2 , спланированные на разных процессорах. На рис. 2 показан план, уменьшающий количество процессоров с двух до одного. Проблемой является определение минимального количества процессоров без перебора всех возможных альтернатив.

Таблица 1

Характеристики для множества задач с бинарным частотным распределением

Задача	Частота	Период	Время выполнения
J_1	1/4	4	1

J_2	1/8	8	2
J_3	1/16	16	1½
J_4	1/32	32	5
J_5	1/64	64	3

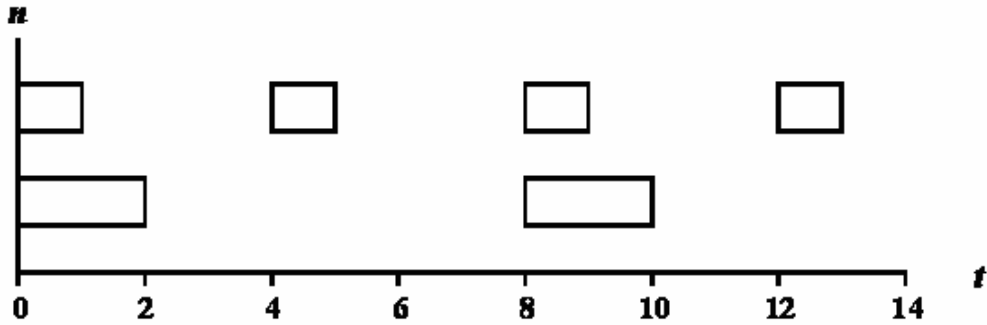


Рис. 1. Временная диаграмма для первых двух задач из табл. 1

Заметим, что слияние двух задач, приведенное на рис. 2, создает новую периодическую задачу с периодом t_1 (равным $2T_1$) и временем выполнения e_1 (равным $T_1 + E_1$). Кроме того, имеются два промежутка с простоями: I_1 – периодичный простой с длительностью $t_1 - e_1$ и Δ_2^1 – принудительное время простоя длиной $I_1 - E_2$ (обозначение Δ_j^i указывает, что принудительное время простоя получается, когда J_j объединяется с J_i).

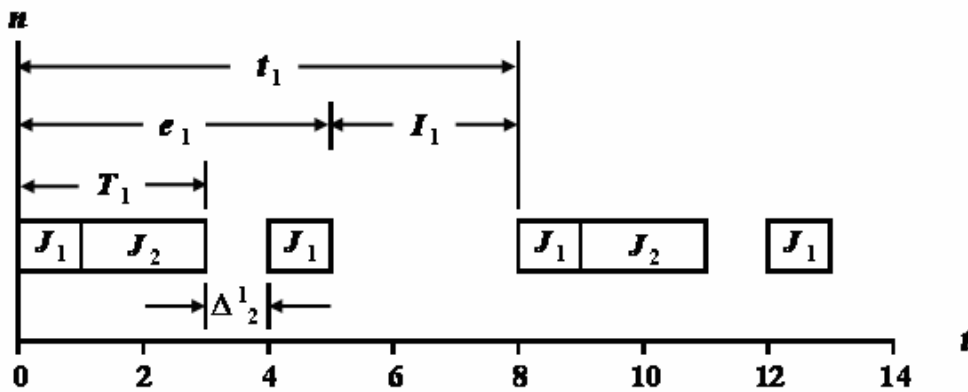


Рис. 2. Ослабление ограничений на число процессоров путем слияния задач

В процессе объединения остальных задач плана нет необходимости рассматривать размещение задач в интервале принудительного простоя. Вместо этого для такой среды план с минимальным числом процессоров формируется в соответствии со следующим алгоритмом:

1) Пусть J_1^*, J_2^*, \dots – подмножества задач, назначаемых процессорам P_1, P_2, \dots . Сначала $J_1^* = J_2^* = \dots = \emptyset$, а $I_1 = I_2 = \dots = \infty$. Всякий раз, когда задача J_j добавляется в некоторое пустое подмножество J_l^* , $I_l = T_j - E_j$.

2) Чтобы назначить очередную задачу J_i , необходимо найти наименьшее I_i такое, чтобы выполнялось условие $E_i \leq I_i$. Добавить J_i в подмножество J_1^* .

Оптимальный план для набора задач из табл. 1 показан на рис. 3. Этот результат может быть обобщен и для случая, когда $f_i = k f_{i+1}$, k – положительное целое число больше единицы.

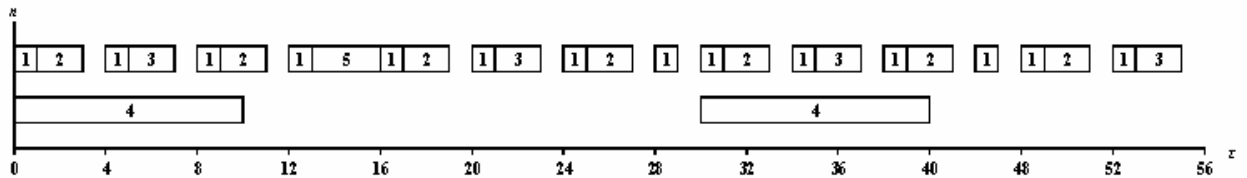


Рис. 3. Оптимальный план для задач из табл. 1.

В случае, когда устранена бинарная частотная связь между задачами, принятая ранее, т. е. в режиме реального времени реализуются периодичные задачи с независимым распределением частот, то задача становится более сложной, а оптимальное решение не может быть найдено точными методами. Были разработаны эвристические алгоритмы и проведено их относительное сравнение с применением моделирования. Предлагаемые подходы делятся на три группы.

1. В порядке уменьшения частоты. Задачи располагаются в порядке уменьшения частоты, и их назначение также должно проходить в этом порядке.
2. В порядке уменьшения критерия загрузки. Критерий загрузки задачи J_i , обозначаемый L_i , определяется следующим образом: $L_i = E_i / T_i$.
3. Сохранение минимальной длины критического интервала. Критический интервал между двумя задачами определяется как минимальный интервал между временем завершения первой задачи и временем начала выполнения второй задачи в некоторой точке плана. Определение этого интервала не включает первую итерацию обеих задач, где по определению начало выполнения второй задачи немедленно следует за завершением первой задачи.

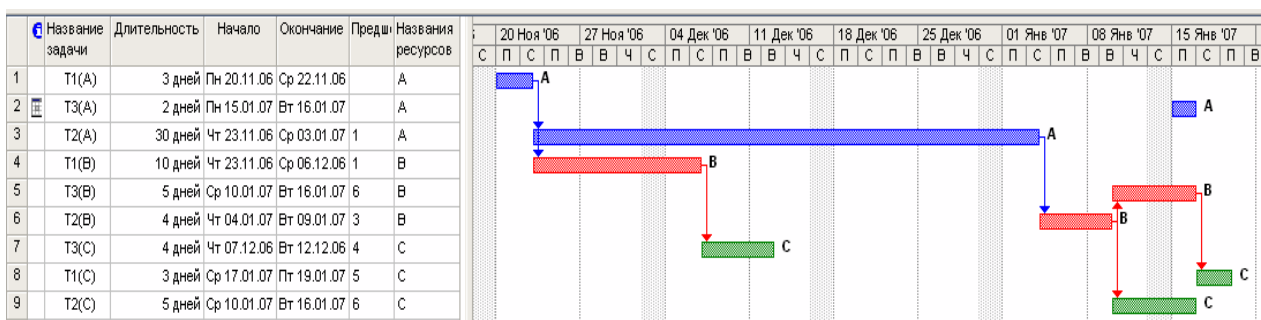
При тестировании данных методов, задачи разделялись на два класса. В 1-м классе частоты задач кратны более чем двум базовым частотам, а во 2-м – не более чем двум базовым частотам. Ни один из алгоритмов не показал значительного превосходства над другими. Однако подход 2 исключительно хорошо показал себя на задачах первого класса. Подход 3 лучше решает некоторые задачи второго класса, а оба подхода 1 и 2 неплохо решают задачи, которые оказались трудными для подхода 3. Во многих случаях уменьшение частот задач или времен их выполнения может привести к увеличению количества требуемых процессоров. И наоборот, требуемое количество процессоров может быть уменьшено за счет увели-

выполнить, должна быть обслужена одним из процессоров, а потом другими. Это чередование должно соблюдаться для всех задач, входящих в план, но требования идентичности процессоров нами не вводится.

Алгоритм Джонсона упорядочивает задачи, одновременно доступные на двухмашинном конвейерном плане, таким образом, чтобы минимизировать максимальное время потока. По алгоритму Джонсона задача T_i предшествует T_j , если $A_i + B_i \geq A_j + B_j$ и $A_i \leq A_j$, где A_i и B_i представляют требования T_i для процессора класса A и класса B соответственно. Процедуры предыдущего алгоритма обобщены для ситуации, когда более чем один процессор может существовать в каждом из двух классов – классе A и классе B . Для модифицированного алгоритма Джонсона (МЮ) в конвейерной среде с m процессорами класса A и n процессорами класса B задача T_i предшествует T_j , в соответствии с (МЮ), если $\min(A_i/m, B_j/n) < \min(A_j/m, B_i/n)$.

Эта модификация, фактически ослабляет действие нескольких ограничений, используемых в общем алгоритме Джонсона. Во-первых, допускается наличие более двух машин, а во-вторых, предполагается, что объем доступной промежуточной памяти равен нулю. Поэтому также предлагается процедура сравнительного анализа трехмашинных конвейерных планов: FSIS-план с бесконечной промежуточной памятью, FSNIS-план без промежуточной памяти, FSFIS-план, с ограниченной промежуточной памятью.

Иллюстрация программной реализации трехмашинного конвейерного FSNIS-плана без



промежуточной памяти приведена на рис. 5.

Рис. 5. Трехмашинный конвейерный FSNIS-план без промежуточной памяти

Заключение

Предложенный подход решения задач обработки данных в гетерогенных распределенных системах дает следующие преимущества: увеличение пропускной способности ИС кон-

вейерного типа; обеспечение однородности функций конвейерной ИС, что позволяет уменьшить требования к ИС; сокращение времени и повышение эффективности коммуникаций между процессорами; возможность мониторинга загруженности ИС конвейерного типа в режиме реального времени. Чтобы пользователю не пришлось производить мониторинг всех конвейерных планов, вводятся пороговые допустимые значения текущих задач, и контрольные устройства системы будут извещать о превышении порога.

Таким образом, технология, базирующаяся на конвейерной обработке данных, предоставляет возможности формирования показателей эффективности работы отдельных процессоров, реализующих конвейерные планы для неоднородных систем. При этом значительно повышается эффективность процессов обработки информации за счет обоснования факторов планирования задач обработки данных в информационных системах.

Список литературы

1. Барский А. Б. Параллельные процессы в вычислительных системах / А. Б. Барский. – М.: Радио и связь, 1990.
2. Джин Бэкон, Тим Харрис Операционные системы. Параллельные и распределенные системы Operating Systems. Concurrent and Distributed Software Design И.: Питер, ВНУ, 2004.
3. Воеводин В. В. Математические модели и методы в параллельных процессах / В. В. Воеводин. – М.: Наука, 1986.
4. Ковалев И. В. Моделирование и оптимизация параллельных процессов в информационно-управляющих системах / И. В. Ковалев, Р. Ю. Царев. – Красноярск: ИПЦ КГТУ, 2003.
5. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – СПб.: Питер, 1999.
6. Олифер В. Г. Сетевые операционные системы / В. Г. Олифер, Н. А. Олифер. – СПб.: Питер, 2001.
7. Эндрю Таненбаум, Маартен ван Стеен Распределенные системы. Принципы и парадигмы. – Изд-во: Питер, 2003.
8. Neumann K. Stochastic Project Networks. Temporal Analysis, Scheduling and Cost Minimization / K. Neumann // Lecture Notes in Economics and Mathematical Systems, No. 34, Springer-Verlag, 1990.
9. Phillips D. T. Fundamentals of network analysis / D. T. Phillips, A. Garsia-Diaz. Prentice-Hall, Inc. Englewood Cliffs. – New Jersey, 1981.

Рецензенты:

Лаптенко В.Д., д.т.н., профессор кафедры информационно-управляющих систем ФГБОУ ВПО «Сибирский государственный аэрокосмический университет им. академика М. Ф. Решетнева», г. Красноярск.

Лапко В.А., д.т.н., профессор кафедры космических средств и технологий ФГБОУ ВПО «Сибирский государственный аэрокосмический университет им. академика М. Ф. Решетнева», г. Красноярск.