© А.Н. Докучаев, 2011

УДК 004.031.43:004.272.43:004.451.42

А.Н. Докучаев

МЕТОД ГЛОБАЛЬНОЙ МУЛЬТИПРОЦЕССОРНОЙ ДИСПЕТЧЕРИЗАЦИИ ДЛЯ ВСТРАИВАЕМЫХ СИСТЕМ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ И СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Рассмотрен пример реализации механизмов планирования во встраиваемой системе, основанный на принудительных периодических вытеснениях.
Ключевые слова: системы реального времени, встраиваемые системы специального назначения, параллелизм, мультипроцессорная диспетчеризация.

Методы глобального статического планирования (ГСП) в многоядерных и мультипроцессорных вычислительных системах реального времени (СРВ) обладают рядом неоспоримых преимуществ перед динамической диспетчеризацией. В частности, им требуется гораздо меньше системных ресурсов на выполнение алгоритма диспетчера и отсутствует необходимость детектирования событий изменения приоритетов и перепланирования.

Большинство встраиваемых систем специального назначения, успешно применяющихся в нефтегазовой и горнодобывающей промышленности, машиностроении, энергетике и металлургии являются также системами реального времени. Таким образом, требования надежности и своевременной реакции на события, предъявляемые к СРВ, распространяются и на них.

Среди методов ГСП в СРВ наиболее примечательным является SM-US [1], поскольку среди дисциплин данного класса обладает наивысшей эффективностью при соблюдении своевременного выполнения объектов диспетчеризации реального времени. Дисциплина подразумевает разделение задач на тяжелые и легкие. Легким назначаются приоритеты в соответствии с частотно-монотонным планированием, тяжелым же выделяется диапазон высших приоритетов, правило выбора котроых автором метода не регламентируется. Под критерием классификации легких и тяжелых задач подразумевается выражение:

$$u_i = {C_i}/{T_i} \le \frac{2}{3+\sqrt{5}} \ . \qquad (1)$$

Метод характеризуется успешным планированием любого набора задач с нагрузкой, не превышающей $U \le 38\%$.

С целью повышения эффективности диспетчеризации легких задач в мультипроцессорных СРВ методом SM-US предлагается модифицировать алгоритм следующим образом:

1. Выделить среди подмножества легких так называемые сверхлегкие задачи τ_j, удовлетворяющие условию:

$$\sum_{k \in hp(\tau_j)} f_{si}\left(\frac{T_j - C_k}{C_j}\right) \geq m, \qquad (2)$$

где $f_{si}(x) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$ – мера влияния высокоприоритетной задачи на недиспетчеризируемость сверхлегкой.

2. Для любой из сверхлегких задач при достижении числом более приоритетных задач количества процессоров ($hp(\tau_i) = m$) требуется изменить принцип потребления вычислительных ресурсов тяжелыми задачами в наборе.

3. Набор параметров задач из тяжелого подмножества дополняется периодом T_i^{BS} и длительностью B_i^S принудительного блокирования, а также величиной времени непрерывного выполнения $C_i^{'} = T_i^{BS} - B_i^S$.

4. В соответствии с определенными дополнительными характеристиками требуется периодически производить принудительные вытеснения и блокирования тяжелых задач. Максимальное число тяжелых объектов, выполняющихся подобным образом не должно превышать количества имеющихся в СРВ процессоров m.

Для сведения к минимуму числа вытеснений легких (1) и сверхлегких задач (2), обладающих сравнительно невысокими уровнями приоритета, целесообразно воспользоваться следующим критерием определения характера периодических блокирований:

$$\frac{T_i^{BS} - B_i^S}{C_i} = 0.5 . \qquad (3)$$

При помощи выражения (3) легко могут быть вычислены дополнительные параметры диспетчеризации для всех задач из тяжелого подмножества.

Эффективность нового метода по отношению к эталонному представлена на рис. 1. Величина C' характеризует относительный прирост (в процентах) числа успешно диспетчируемых наборов задач при исследовании выборки с равномерно распределенными параметрами C_i и $T_i = D_i$, (в рамках эксперимента для генерируемых наборов задач реального времени $T_i \in [50;1000]$ и $C_i \geq 10$) для различного числа задач n и разной вычислительной нагрузке на процессоры U (число процессоров $m = 4$). Оценка эффективности метода производилась в соответствии с принципами теории анализа времени отклика (от англ. RTA – Response Time Analysis).

Рассмотрим метод оценки времени отклика для представленного в данной работе алгоритма диспетчеризации. Согласно основному положению теории RTA произвольный набор задач будет успешно выполняться с соблюдением

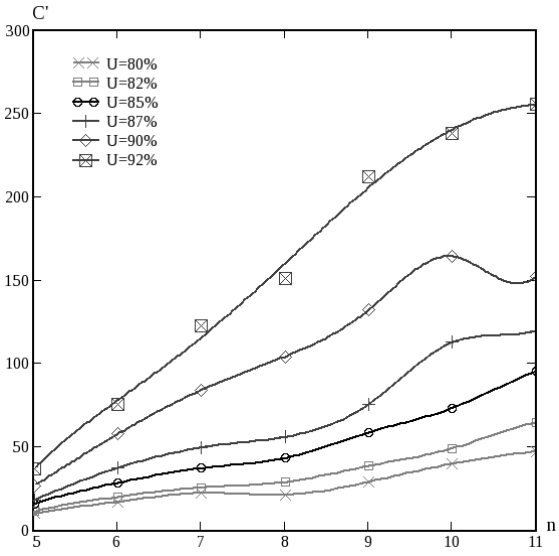

Рис. 1. Зависимость эффективности диспетчеризации от числа задач и вычислительной нагрузки в 4 процессорной СРВ

всех крайних сроков завершения при использовании исследуемого метода ГСП, если для каждой задачи время отклика не превышает абсолютного значения крайнего срока завершения в каждом периоде. Величина времени отклика характеризует абсолютное значение требуемого задаче времени исполнения в произвольном периоде при наихудшем фазировании объектов диспетчеризации из подмножества $hp(\tau_i)$ с учетом всех вытеснений. Очевидно, вычисление этого параметра достаточно тудоемкая задача, поскольку требуется проведение математического моделирования всего процесса диспетчеризации высокоприоритетных задач. Тем не менее, в случае алгоритма представленного в работе допустимо применение методики оценки времени отклика для ГСП согласно [2]. Для этого достаточно рассматривать вместо всех тяжелых задач модель периодически вытесняемой псевдозадачи, обладающей следующим набором параметров: $C_{i,j} = T_i^{BS} - B_i^S$, $T_{i,j} = T_i^{BS} = D_{i,j}$. При учете (3) можем определить число экземпляров псевдозадачи $\tau_{i,j}$ (максимальное значение параметра j), существующих в произвольном периоде выполнения объекта τ_i из выражения:

$$\forall \tau_{i,j} : j \in \left[1; \frac{C_i}{T_i^{BS} - B_i^S}\right].$$

При учете всего вышесказанного оценка времени отклика [2] для любой задачи, классифицируемой в качестве легкой или сверхлегкой, может быть определена при помощи выражения [3]:

$$R_i = C_i + \left\lfloor \frac{I_i + I_i^h}{m} \right\rfloor. \qquad (4)$$

Здесь величина I_i есть интерференция от выполнения высокоприоритетных легких задач, а I_i^h, соответственно, тяжелых. Обе интерференции вводятся следующим образом:

$$I_i = \sum_k I_k^N(R_i, C_i) + \sum_k I_k^{DF}(R_i, C_i),$$

$$I_i^h = \sum_k^h I_k^N(R_i, C_i) + \sum_k^h I_k^{DF}(R_i, C_i),$$

где $I_k^N(R_i, C_i)$ суммируется по всем легким/тяжелым высокоприоритетным задачам, а $I_k^{DF}(R_i, C_i)$ лишь по $m-1$ экземпляру высокоприоритетных задач с наибольшими $I_k^{DF}(R_i, C_i)$. Максимум интерференции, от задачи τ_k, определяется выражением:

$$I_k^R(R, C) = \min(W_k^R(R), R - C + 1). \quad (5)$$

Для легких задач верхняя граница нагрузки на процессор, обусловленная выполнением задачи τ_k есть:

$$W_k^R(L) = N_k^R(L)C_k +$$

$$\min(C_k, L + R_k - C_k - N_k^R(L)T_k),$$

где $N_k^R(L) = \left\lfloor \frac{L + R_k - C_k}{T_k} \right\rfloor$.

Для тяжелых задач данная величина вычисляется по формуле:

$$W_k^R(L) = N_k^R(L)C_k' + \min(C_k', L + R_k - C_k' - N_k^R(L)T_k^{BS}),$$

где $N_k^R(L) = \left\lfloor \frac{L + R_k - C_k'}{T_k^{BS}} \right\rfloor$.

Вклад в общую интерференцию экземпляров τ_k, начавших выполнение до момента порождения экземпляра τ_i, задан следующим образом:

$$I_k^N(R,C) = \min(W_k^N(R), R - C + 1). \qquad (6)$$

По аналогии рассчитаем верхние границы нагрузки легких и тяжелых экземпляров задач для $I_k^N(R,C)$:

$$W_k^N(L) = N_k^N(L)C_k +$$

$$+ \min(C_k, L - N_k^N(L)T_k), \text{ при } N_k^N(L) = \left\lfloor \frac{L}{T_k} \right\rfloor;$$

для легких и для тяжелых объектов:

$$W_k^N(L) = N_k^N(L)C_k' +$$

$$+ \min(C_k', L - N_k^N(L)T_k^{BS}), \text{ при } N_k^N(L) = \left\lfloor \frac{L}{T_k^{BS}} \right\rfloor \text{ соответственно.}$$

Разность интерференций вида (5) и (6) определяется выражением:

$$I_k^{DF}(R,C) = I_k^R(R,C) - I_k^N(R,C).$$

Рекуррентное выражение (2) имеет решение при начальном приближении $R_i^0 = C_i$. Для тяжелых задач время отклика полагается равным: $R_i = C_i'$

Сравнительно низкая эффективность метода (рис. 1) при малом числе задач объясняется редкостью возникновения сверхлегких объектов диспетчеризации (2). Ее возрастание с увеличением вычислительной нагрузки также можно объяснить "утяжелением" набора задач при фиксированном числе объектов. Анализ представленных результатов позволяет утверждать, что максимальная эффективность метода достигается на разнородных наборах задач (в смысле отнесения к разным подмножествам) с высокими суммарными требованиями к вычислительным ресурсам процессора. Исходя из специфики предметной области, можно придти к выводу, что применяемые встраиваемые системы специального назначения в большинстве случаев соответствуют указанному описанию.

Большое влияние при оценке эффективности метода диспетчеризации оказывает учет накладных расходов, обусловленных не идеальностью вычислительной архитектуры. Речь в данном случае идет о неатомарности процессов вытеснения объектов и перепланирования, сопровождающихся переключениями контекстов активной и восстанавливаемой задач. Для случая, соответствующего рис. 1, подобные затраты вычислительных мощностей приняты равными нулю.

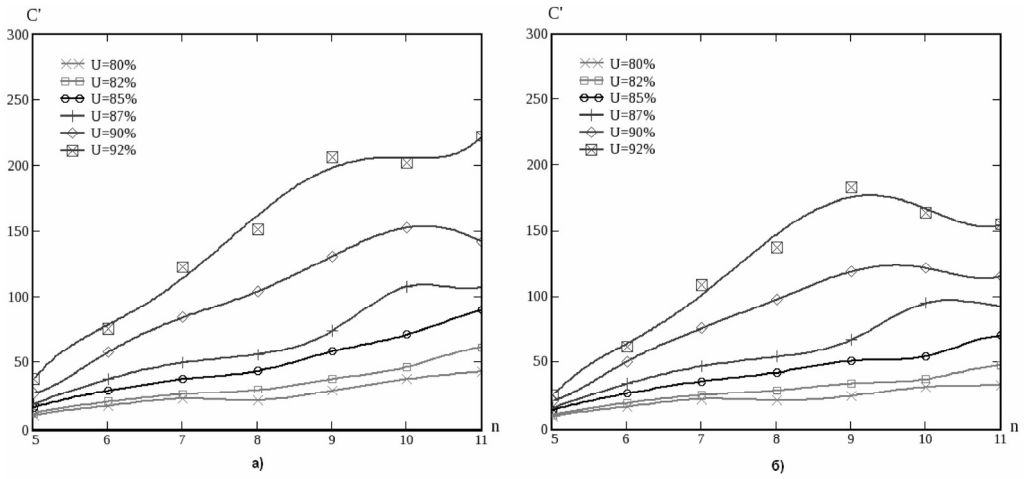

Рис. 2. Зависимость эффективности диспетчеризации от числа задач и вычислительной нагрузки в 4 процессорной СРВ при наличии вытеснений

Динамика изменения относительного прироста эффективности для рассматриваемого метода (автор именует предложенный алгоритм SMMS-FP) в зависимости от количества затрат на перепланирование представлена на рис. 2. В случае "а" время единичного переключения контекста принято равным 1% от времени выполнения тяжелой задачи с максимальным периодом и 50% нагрузкой на процессор. Для случая "б" эта величина принята равной 5%.

Необходимо также отметить, что представленная математическая модель может применяться не только для диспетчеризации задач во встраиваемых системах специального назначения и СРВ, но также и при системном анализе подобных программно – аппаратных комплексов в различных отраслях промышленности. По сравнению с эталонным методом эффективность предложенного алгоритма очевидна. При суммарной вычислительной нагрузке набора из $2m-1$ задач равной 92% в четырех процессорной СРВ наблюдается двукратное увеличение числа успешных экспериментов. Для систем с ограниченным запасом вычислительных ресурсов этот факт особенно актуален.

СПИСОК ЛИТЕРАТУРЫ

1. *Andersson B.* Global Static-Priority Preemptive Multiprocessor Scheduling with Utilization Bound 38% // Proc. of the 12th International Conference on Principles of Distributed Systems. – December, 2008. – P. 73–88.

2. *Nan Guan, Martin Stigge, Wang Yi.* New Response Time Bounds for Fixed Priority Multiprocessor Scheduling // Proc. of 30th IEEE Real-Time Systems Symposium (RTSS 2009). – December, 2009. – P. 387–397.

3. *Davis R.I., Burns A.* Improved Priority Assignment for Global Fixed Priority Pre-emptive Scheduling in Multiprocessor Real-Time Systems // Real-Time Systems. – 2010. – Vol. 47, N. 1. – P. 1–40. ИАЗ

КОРОТКО ОБ АВТОРЕ

Докучаев А.Н. – аспирант БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова, кафедра «Системы обработки информации и управления», Санкт-Петербург, a.n.dokuchaev@gmail.com