Численный метод диспетчеризации вычислений для систем промышленной автоматизации

А.Е. Дубовик (ИНЭУМ)

Рассматривается новый численный алгоритм диспетчеризации вычислений с динамически изменяющимися приоритетами, ориентированный для распределенных АСУТП.

Одной из основных задач различных автоматизированных систем является рациональное распределение (планирование) ресурсов ЭВМ между пользователями системы и выполняемыми вычислительными процессами (задачами) автоматизации. При построении управляющих программ (планировщиков, диспетчеров) используется целый ряд различных методов и алгоритмов диспетчеризации вычислений. Управляющие программы при любом из методов диспетчеризации вычислений решают задачу минимизации времени ожидания и сведения к минимуму возможных потерь информации. При этом учитывается и оценивается время ожидания вычислительных процессов (заявок, задач, сообщений), и то обстоятельство, что обработка информации, особенно в диалоговых, больших автоматизированных, коммуникационных и других вычислительных системах производится путем многократного выполнения циклов, состоящих из запроса задачи абонента-пользователя и/или реакции, ответа ЭВМ (узла сети, сервера и т.д.). Методы и алгоритмы, лежащие в основе управляющих программ для решения разнообразных задач диспетчеризации вычислений в АСУТП реализуют, в основном, традиционные вычислительные дисциплины обслуживания "первым пришел – первым обслужен" или "последним пришел – первым обслужен", а также дисциплины обслуживания на основе очередей приоритетов. Первые из вышеуказанных методов основываются на учете только одного параметра – времени поступления заявки на обслуживание вычислительного процесса (сообщения). Второй метод, учитывая особенности каждого конкретного процесса: размер процесса, приоритет процесса, содержащуюся в нем информацию и т.д., как правило, не учитывает время поступления запроса. В результате могут происходить значительные задержки в обслуживании других процессов и сообщений и, как следствие, отказы в работе систем управления (при предельных нагрузках системы и/или ограниченном времени ожидания). В ряде случаев при наличии значительного числа вычислительных процессов в системе (или поступающих сообщений), при анализе подобных вычислительных систем и выборе алгоритмов диспетчеризации вычислительных процессов пытаются использовать методы теории массового обслуживания, учитывая так или иначе известное число входных потоков задач (сообщений) и их вероятностные характеристики. Однако, практическое решение таких задач диспетчеризации вычислений (и др. подобных задач) является достаточно сложным и имеет неоднозначный ответ. Кроме того, при диспетчеризации вычислений следует учитывать, что в подобных приоритетных вычислительных системах и сетях ЭВМ для сложных задач автоматизации всегда имеется вероятность того, что низкоприоритетные процессы вовсе не получат вычислительных ресурсов. Подобное в ряде АСУТП особенно для задач ядерной энергетики просто не допустимо. Дать полную практическую, адекватную реальности оценку работы подобных вычислительных систем для автоматизированных систем методами теории массового обслуживания можно лишь приближенно, так как эти методы требуют учета многочисленных разнообразных вероятностных факторов, которые практически невозможно выразить количественно и тем более объединить в единый оценочный критерий. Поэтому в отличие от существующих методов в данной работе предлагается новый численный метод диспетчеризации вычислений, который имеет актуальную возможность эффективной приоритетной организации вычислений в распределенных вычислительных системах с динамически изменяющимися приоритетами. Причем численная диспетчеризация может быть легко осуществлена в каждой ЭВМ системы, при приоритетной коммутации и маршрутизации сообщений в сетях в составе распределенных сложных АСУТП [2, 3, 4].

В работах [3, 4] приведены и строго формализованы теоретические основы предлагаемого численного метода диспетчеризации вычислений. Поэтому в этой работе целесообразно рассмотреть практические аспекты и алгоритм реализации метода для эффективной организации вычислений на простых численных примерах установки приоритетов выполнения прикладных, сетевых и др. задач. Эта же методология организации вычислений может быть использована в широком классе задач в разноуровневых системах автоматизации.

В качестве конкретного примера рассмотрим реализацию численного алгоритма в реальном узле сети. В

качестве исходных данных рассмотрен небольшой узел сети распределенной АСУ с тремя потоками заявок (и/или сообщений) К1, К2, К3, которые заданы своими интервалами поступления, и/или уровнями приоритета $\Delta\tau_1 = 1{,}5$, $\Delta\tau_2 = 2{,}6$ и $\Delta\tau_3 = 4{,}0$. Для простоты определим величину интервала, соответствующего пропускной способности узла сети или системы, $\Delta t^* = 1$. Следуя приведенным в [2, 3] оптимальному решающему правилу, к каждому моменту времени обслуживания t_1 необходимо определить минимальное значение разности, т.е. вычесть из значений 1,5, 2,6, 4,0 по единице. Очевидно, минимум этой разности соответствует интервалу $\Delta\tau_1$, что определит приоритет выполнения (или прохождения) заявки К1. Суммируя значение $\Delta\tau_1$ и Δt^* (т.е. 1,5 + 1), осуществляем динамическое смещение начала отсчета заявки К1 в последнее зафиксированное значение. Вычисления продолжаются аналогично для последующих моментов времени t_2,..., t_i,..., $t_n = T$. Вычислительную процедуру диспетчеризации в соответствии с определяемыми приоритетами выполнения заявок К1, К2, К3 можно представить в виде таблиц. В этих таблицах в темных столбцах приведены результаты аналогичной последовательности вычислений, определяющие последовательность выбора каждого конкретного процесса при диспетчеризации вычислений с динамически изменяющимися приоритетами.

1.5 - 1	1	2.5 - 2	2	3.5 - 3		3.5 - 4	4	5.5 - 5		5.5 - 6	6	7.5 - 7	
2.6 - 1		2.6 - 2		2.6 - 3	3	5.6 - 4		5.6 - 5		5.6 - 6		5.6 - 7	7
1		4 - 2		4 - 3		4 - 4		4 - 5	5	9 - 6		9 - 7	

7.5 - 8	8	9.5 - 9		11.5 - 10	10	11.5 - 11		11.5 - 12	12	13.5 - 13	
9.6 - 8		9.6 - 9		9.6 - 10		13.6 - 11	11	13.6 - 12		13.6 - 13	
9 - 8		9 - 9	9	13 - 10		13 - 11		13 - 12		13 - 13	13

Упрощая вычисления, можно представить вычислительную процедуру в виде следующих таблиц

1.5 - 1	1	1.5 - 1	2	1.5 - 1		0.5 - 1	4	1.5 - 1		0.5 - 1	6	1.5 - 7	
2.6 - 1		1.6 - 1		0.6 - 1	3	2.6 - 1		1.6 - 1		0.6 - 1		0.6 - 7	7
		3 - 1		2 - 1		1 - 1		0 - 1	5	4 - 1		3 - 7	

0.5 - 1	8	1.5 - 1		0.5 - 1	10	1.5 - 1		1.5 - 1	12	1.5 - 1	
2.6 - 1		1.6 - 1		0.6 - 1		0.6 - 1	11	2.6 - 1		1.6 - 1	
2 - 1			9	4 - 1		3 -1		2 - 1		1 - 1	13

Процесс оптимальной диспетчеризации вычислений, минимизирующий время ожидания, проиллюстрирован на рисунке. Здесь, для приведенного примера – узла сети с тремя потоками заявок на обслуживание К1, К2, К3, точками показаны оптимальные моменты выбора задач (из трех потоков заявок).

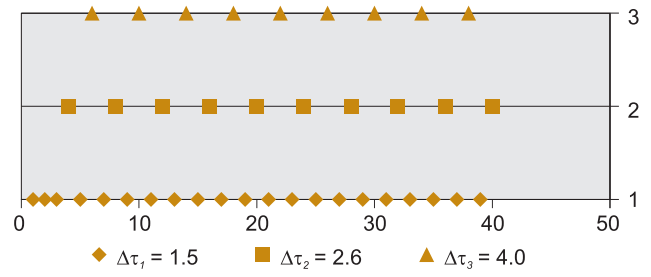

Рис. 1. График обслуживания заявок в зависимости от приоритета

Выписав из темных столбцов таблиц полученные результаты аналогичной вычислительной процедуры, получим оптимальный порядок выполнения заявок, минимизирующих время ожидания:

К1 – 1, 2, 4, 6, 8, 10, 12, 14 ... ;

К2 – 3, 7, 11, 15, ... ;

К3 – 5, 9, 13, ..., т.е. заявки с более высоким приоритетом К1 обслуживаются чаще в моменты t_1, t_2, t_4, заявки К2 – в t_3, t_7, t_{11}, а низкоприоритетные заявки К3 – в t_5, t_9 и т.д. Следует отметить, что подобных примеров можно привести сколь угодно много и аналогичные (рекуррентные) вычислительные процедуры могут продолжаться сколь угодно долго и не зависят от числа заявок, пропускной способности вычислительной системы и др. Процедура приведенных вычислений и полученные результаты показывают, что процесс диспетчеризации вычислений по приведенным правилам соответствует приоритетам (или интенсивности поступления) различных заявок, не зависит от их числа, динамически изменяет приоритеты и на каждом временном интервале минимизирует время ожидания. В этом достаточно просто убедиться, оценив среднее и максимальное значения полученных интервалов времени ожидания как каждого потока заявок, так и их суммы при заданной пропускной способности вычислителя [2, 3].

Как показывают результаты моделирования, предлагаемый оптимальный численный метод достаточно прост и универсален в программной реализации, не зависит от числа и вида заявок, а также от продолжительности времени функционирования системы. Реализация метода позволяет резко повысить эффективность диспетчеризации вычислений как в каждом узле сети, так и в каждом компьютере при самых минимальных вычислительных затратах. При этом имеется уникальная возможность сочетания различных дисциплин диспетчеризации от самых простых, типа "первым пришел – первым обслужен" или "последним пришел – первым обслужен", до различных многофункциональных дисциплин очередей с приоритетами. Приведенный алгоритм метода позволяет достаточно просто создавать различные управляющие программы диспетчеризации вычислений в различных системах промышленной автоматизации.

Особо отметим важное преимущество приведенного алгоритма и технологии диспетчеризации. Эта технология позволяет достаточно эффективно реализовать возможность динамически изменять приоритеты, учитывая как время поступления процессов и/или сообщений, так и особенности самих процессов, режима их обработки путем выбора функции $\Delta\tau_j = f(R_1, R_2, ..., R_n)$, где R_1, R_2,..., R_n параметры сообщения (длина, время обработки и др.).

В качестве важных и актуальных примеров практической реализации приведенного алгоритма отме-

чены некоторые наиболее типичные случаи. Это, прежде всего, диспетчеризация вычислений в ОС, системах разделения времени, диалоговых и коммутационных системах, используемых в промышленной автоматизации.

Эффективность указанных систем характеризуется, как правило, временем реакции вычислителя (т. е. интервалом времени между поступлением запроса и ответом). Исследования показывают, что наиболее эффективным является постоянное время реакции системы и, как отмечено, отклонения от этого времени нежелательны как в большую, так и в меньшую сторону. Известно, что основное влияние на время реакции оказывают характеристики распределения потоков заявок или сообщений и время их обработки. Время обработки зависит в основном от типа сообщения и может быть достаточно точно определено априори. Поэтому уровень приоритета сообщения может, например, выражаться соотношением $\Delta\tau_j = K - M_j$ где M_j – среднее время обработки сообщения типа $k = \max(M_j)$. Используя подобный выбор уровня приоритета при диспетчеризации, как показали исследования, можно получить характеристику времени реакции системы, приближающуюся к идеальной, т.е. можно создать операционную систему с идеальными возможностями в этом смысле.

Передача сообщений в больших распределенных системах АСУТП и сетях характеризуется тем, что каждый из узлов имеет, как правило, пропускную способность значительно большую, чем линии связи. Это приводит к отсутствию входных очередей сообщений, однако, как правило, имеются выходные очереди (из узла сети в сеть). В этом случае при диспетчеризации вычислений и определении порядка передачи сообщений следует учитывать длину сообщения, так как она меньше зависит от времени обработки, чем от времени передачи и, соответственно, короткие сообщения более полно загружают вычислительные ресурсы. Поэтому рекомендуется установить уровень приоритета сообщений пропорционально их длине, что обеспечит высокую эффективность тех систем, у которых показателем эффективности является доля заявок, выполненных за время, меньшее заданного (что даже при перегрузках системы, связанных с обработкой длинных сообщений, в меньшей степени увеличивает время ожидания).

Программа, реализующая приведенный численный алгоритм диспетчеризации вычислений с динамически изменяющимися приоритетами, внедрена в составе комплекса прикладных программ АСУТП Красногорской компрессорной станции в ООО "Уралтрансгаз".

Список литературы

1. *Кузминский М.* Операционные системы //Открытые системы 1997. №1.
2. *Дубовик А.Е., Дубовик Е.А.* Численный метод коммутации и маршрутизации сообщений // Сб. тр. IV Междун. конф. "Развитие и применение открытых систем". Н. Новгород. 1997.
3. *Дубовик А.Е., Дубовик Е.А.* Основы и методы многоканального измерения функций различного спектрального состава // Сб. тр. 1-й Междун. конф. "Цифровая обработка сигналов и ее применение" DSPA'98. т. 4. Москва. 1998.
4. *Dubovik A.,Ye.,Dubovik Ye.A.* The optimal numeric metod of calculation dispatching with dynamic priorities // Сборник тр. Междун. конф. EAST-WEST, EWITD'96. Москва.

Дубовик Александр Евгеньевич – аспирант Института электронных управляющих машин.
Контактный телефон (095) 455-55-71.