НЕКОТОРЫЕ ЗАДАЧИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ FIFO-ОЧЕРЕДЯМИ*

Е. А. Аксенова, А. В. Драц, А. В. Соколов (Петрозаводск)

Во многих приложениях требуется работа с несколькими FIFO-очередями, расположенными в общем пространстве памяти. Для этого применяют различные программные или аппаратные решения[1–3]. В работах [5–10] предлагались математические модели для последовательного циклического и связанного способов представления очередей. На основе предложенных моделей решались задачи оптимального начального распределения памяти для очередей при различных критериях оптимальности. Если в качестве критерия оптимальности рассматривалась минимальная доля потерянных элементов при бесконечном времени работы очередей, то предполагалось, что при переполнении очереди все последующие элементы, поступающие в нее, отбрасываются до тех пор, пока не появится свободная память. Если в качестве критерия оптимальности рассматривалось максимальное среднее время до переполнения памяти, то предполагалось, что при переполнении очереди работа программы заканчивается. Такой метод работы с очередями необходим, когда FIFO-очереди используются в программных системах и потери элементов очередей недопустимы.

Управление двумя FIFO-очередями в случае их движения друг за другом по кругу

Исследуем новый метод представления нескольких последовательных циклических FIFO-очередей в общей памяти, предложенный в [5]. В этом методе очереди двигаются по кругу, друг за другом, начиная с некоторого начального места в памяти. Память заранее не делится между очередями. В случае если одна из очередей стала пустой, вторая может занимать всю доступную память, пока не догонит сама себя. Если же оживет вторая очередь, то ее можно пустить с середины свободного участка или с некоторой оптимальной точки (поиск этой точки является нашей задачей), которая зависит от вероятностных характеристик очередей. В качестве критериев оптимальности рассмотрены максимальное среднее время до переполнения памяти и минимальная доля потерянных элементов. Предложена и исследуется математическая модель для этого способа работы с двумя FIFO-очередями. Данный метод сравнивается с последовательным и связанным методами представления FIFO-очередей.

Постановка задачи

Рассмотрим две очереди, находящиеся в памяти размера m единиц. Время дискретно и на каждом шаге возможна одна из следующих операций:

- включение элемента в первую очередь с вероятностью p_1,
- исключение элемента из первой очереди с вероятностью q_2,
- включение элемента в первую очередь с вероятностью p_2,
- исключение элемента из первой очереди с вероятностью q_2,
- чтение элемента в одной из очередей без его исключения с вероятностью r,

$p_1 + q_1 + p_2 + q_2 + r = 1.$

Все элементы имеют одинаковый размер, завершение работы не происходит в случае попытки исключения элемента из пустой очереди. Работа начинается с пустых очередей, x и y – длины очередей, z – расстояние от головы первой очереди до хвоста второй (рис. 1). Если $x = 0$ или $y = 0$, т.е. хотя бы одна из очередей пустая, то z, вообще говоря, не определено, поэтому будем считать в этом случае $z = 0$. Тройка (x, y, z) одно-

* Работа выполнена при финансовой поддержке РФФИ, грант №09-01-00330.

значно задает положение очередей в памяти. Переполнение возникнет в том случае, если первая очередь догонит вторую (при $z = 0$) или вторая первую (при $x+y+z=m$).

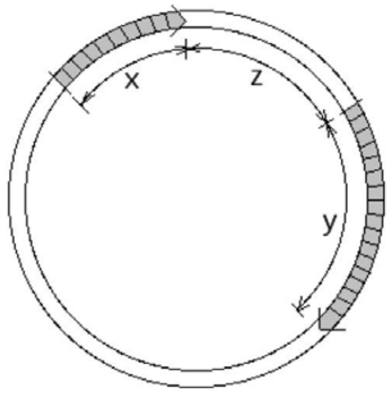

Рис. 1. Движение очередей по кругу

Среднее время работы до переполнения

Будем считать, что в случае переполнения происходит аварийное завершение работы. Критерием оптимальности является максимизация среднего времени работы до переполнения T. В качестве математической модели рассмотрим блуждание по целочисленной трехмерной пирамиде с вершиной (0; 0; 0), ребрами $x = 0$; $y = 0$; $z = 0$ и основанием $x + y + z = m$. Поглощающие экраны: x + y + z = m+ 1 и z = -1. Отражающие экраны: x = -1 и y = -1. Случайное блуждание рассматривается как однородная поглощающая цепь Маркова. Разработан алгоритм генерации матрицы переходов соответствующей цепи Маркова и алгоритм вычисления среднего времени работы до переполнения.

Управление на бесконечном времени

Переполнение одной из очередей не всегда является аварийной ситуацией. В том случае, если очередь заполнила всю отведенную ей память, то все последующие элементы отбрасываются до тех пор, пока не появится свободная память. Такое поведение очереди называется «сбросом хвоста» [2]. В этом случае критерий оптимальности – минимизация времени, которое система проводит в состоянии «сброса хвоста». В качестве математической модели рассмотрим блуждание по трехмерной пирамиде, как в предыдущем случае. Дополнительно для каждого состояния на плоскостях $z + y + z =$ m и $z = 0$ введем состояния «сброса хвоста». Время, которое система проводит в состоянии «сброса хвоста» P^*, можно определить с помощью аппарата регулярных цепей Маркова. Для этого необходимо найти предельный вектор α и просуммировать его компоненты, соответствующие состояниям «сброса хвоста».

Сравнение различных вариантов представлений

Были разработаны имитационные модели работы FIFO-очередей для разных методов представления. Результаты вычислений на имитационной модели совпали с результатами вычислений с использованием марковской модели. Из проведенных экспериментов можно сделать вывод, что для критерия максимизации среднего времени работы до переполнения предпочтительнее использовать представление очередей в виде движения друг за другом по кругу или связанное представление в том случае, если на указатели тратится незначительная часть памяти. Для критерия минимизации времени, которое система проводит в состоянии «сброса хвоста», предпочтительнее использовать последовательное представление. Связанное представление и представление в виде движения друг за другом по кругу предпочтительнее лишь в некоторых случаях, когда вероятности исключения элементов из очередей больше, чем вероятности включения.

Оптимальный метод перераспределения общей памяти для двух последовательных циклических FIFO-очередей

Постановка задачи и математическая модель

Рассмотрим две FIFO-очереди, расположенные в памяти размера m единиц. В каждый момент времени может произойти включение элемента (информации) в очередь, исключение элемента из очереди, чтение или отсутствие операции. В очередях хранятся элементы равной длины. Пусть заданы вероятностные характеристики очередей:

- p_i – вероятность включения элемента в i-ую очередь, i=1,2;
- q_i – вероятность исключения элемента из i-й очереди, i=1,2;
- r – вероятность того, что очереди не меняют свою длину (чтение или отсутствие операции).

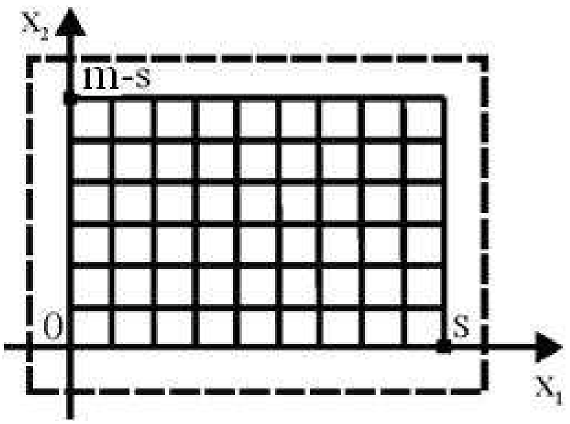

Рис. 2. Область блуждания

Выделим первой очереди s единиц памяти, тогда второй очереди останется m-s единиц. Обозначим текущие длины очередей x_1 и x_2. В качестве математической модели рассмотрим случайное блуждание по целочисленной решетке в прямоугольной области на плоскости $0 \le x_1 < s+1$, $0 \le x_2 < m-s+1$ (рис. 2). Блуждание начинается в точке $x_1=0$, $x_2=0$. Переполнению первой очереди соответствуют точки на прямой $x_1=s+1$, переполнению второй – точки на прямой $x_2=m-s+1$. Прямые $x_1=-1$ и $x_2=-1$ – отражающие экраны, т.е. при исключении элемента из пустой очереди работа не завершается.

Требуется оптимально перераспределить свободную память между очередями после переполнения одной из очередей. В данной задаче предполагаем, что при переполнении одной из очередей происходит перераспределение свободной памяти между очередями и работа с программной системой продолжается. Другими словами, при попытке включения элемента в заполненную очередь, когда x_1=s (или x_2=m-s), требуется определить новую область блуждания $0 \le x_1 < s^*+1$, $0 \le x_2 < m-s^*+1$, т.е. такое значение s^* , где $s^* > s$ (или $m-s^* > m-s$), чтобы время до следующего переполнения какой-либо очереди было максимальным (рис. 3). Этот процесс перераспределения можно продолжать до полного исчерпания свободной памяти.

Такой подход оправдан, если при переполнении одной из очередей в области памяти, выделенной для других очередей, есть достаточно свободной памяти. В классической работе Д. Кнута [1] для работы с n последовательными стеками и очередями рассматривался алгоритм Гарвика. В этом алгоритме при переполнении какой-либо структуры данных приблизительно 10% свободной памяти делится поровну между структурами данных, а оставшиеся 90% делятся пропорционально росту размеров структур данных с момента предыдущего распределения памяти.

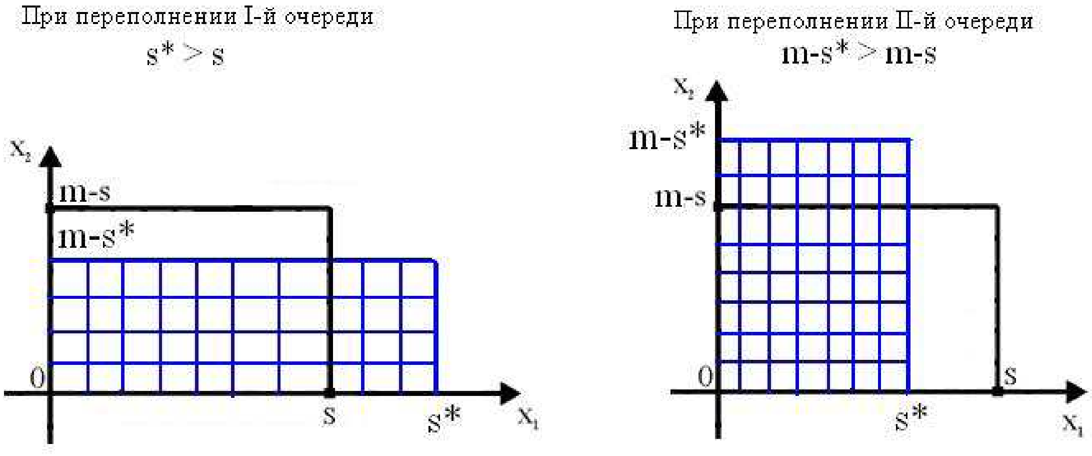

Рис. 3. Выбор новой области блуждания

Алгоритм решения

Математическая постановка задачи сводится к решению задачи нелинейного целочисленного программирования с критерием оптимальности, заданным алгоритмически. Случайное блуждание по целочисленной решетке будем рассматривать как конечную однородную поглощающую цепь Маркова с матрицей вероятностей переходов P [4]. Для решения задачи требуется матрица Q вероятностей переходов из невозвратных состояний в невозвратные (Q – подматрица матрицы P). Вводится нумерация состояний марковской цепи, это позволяет определить структуру матрицы Q для любого размера памяти m и любого значения s. Среднее время работы до переполнения будем искать с помощью фундаментальной матрицы $N=(E\text{-}Q)^{-1}$. Предполагаем, что в самом начале работы, когда обе очереди пустые ($x_1=0$, $x_2=0$), память между ними тоже нужно разделить оптимально, т.е. выбрать такое s, чтобы время работы с очередями до переполнения было максимальным.

Предложенный алгоритм реализован на языке C++, проведены численные эксперименты с марковской моделью. Для рассматриваемой цепи Маркова можно вычислить вероятности попадания в состояния поглощения, которые расположены на прямых $x_1=s+1$ и $x_2=m\text{-}s+1$, т. е. вероятности переполнения очередей. Было бы интересно построить и проанализировать весь процесс перераспределения до полного исчерпания памяти с учетом вероятностей переполнения и сравнить с имитационной моделью данного процесса.

Литература

1. **Кнут Д.** Искусство программирования для ЭВМ. Т. 1. М.: Вильямс, 2001.
2. **Боллапрагада В., Мэрфи К., Уайт У.** Структура операционной системы Cisco IOS. М.: Вильямс,2002. 208 с.
3. **Кормен Е., Лейзерсон Ч., Ривест Р.** Алгоритмы построение и анализ. М.: МЦНМО, 2000.
4. **Кемени Дж., Снелл Дж.** Конечные цепи Маркова. М.: Наука, 1970. 272 с.
5. **Соколов А. В.** Математические модели и алгоритмы оптимального управления динамическими структурами данных. Петрозаводск: ПетрГУ, 2002. 216 с.

6. **Драц А. В. Соколов А. В.** Оптимальное размещение в памяти одного уровня n стеков и/или очередей. //Стохастическая оптимизация в информатике, Вып.5. СПб.: Изд-во С.-Петербургского университета, 2009. С. 72–90.
7. **Аксенова Е. А. Драц А. В. Соколов А. В.** Оптимальное управление n FIFO-очередями на бесконечном времени. //Информационно-управляющие системы. СПб.: Политехника, 2009. С. 46–55.
8. **Аксенова Е. А., Соколов А. В.** Некоторые задачи оптимального управления FIFO-очередями//Труды II Всероссийской научной конференции «Методы и средства обработки информации». М.: Изд. отдел ВМК МГУ, 2005. С. 318–322.
9. **Аксенова Е. А.** Оптимальное управление FIFO-очередями на бесконечном времени// Стохастическая оптимизация в информатике. Межвуз. сб. СПб.: Изд-во СПбГУ, 2006. С. 71–76.
10. **Аксенова Е. А., Драц А. В., Соколов А. В.** Об оптимальном управлении FIFO-очередями на бесконечном времени// Обозрение прикладной и промышленной математики. 2009. Т. 16. Вып. 3. С. 401–415.