АНАЛИЗ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНОЙ РАБОТЫ С ДВУМЯ FIFO-ОЧЕРЕДЯМИ В ОБЩЕЙ ПАМЯТИ

А.В. Соколов, Е.А. Барковский

В случае многоядерных архитектур возможна параллельная работа нескольких потоков со структурой данных (стек, очередь, список, хеш-таблица и т.д.). Для синхронизации такой работы были предложены разные механизмы [1]. Существует ряд программных систем (Intel TBB, Microsoft TPL и др.), где реализована параллельная работа с некоторыми базовыми структурами данных.

В данной работе мы рассматриваем возможность параллельной работы сразу с несколькими структурами данных в общей памяти. При реализации на практике работы с несколькими структурами данных будут использованы какие-то общепринятые механизмы синхронизации, которые не влияют на динамику изменения длин структур данных и, следовательно, на наши модели.

Во многих приложениях требуется работа с несколькими FIFO-очередями, расположенными в общем пространстве памяти. Эффективные алгоритмы работы с несколькими FIFO-очередями в общем пространстве памяти необходимы при разработке различных сетевых устройств и встроенных операционных систем, управляющих потоками пакетов Internet, таких, например, как Cisco IOS, где требования на время обработки пакетов маршрутизатором очень жесткие. Механизм страничной виртуальной памяти здесь не используется, и вся работа происходит в нескольких пулах оперативной памяти. Количество очередей в таких устройствах может достигать нескольких сотен и тысяч, а в будущем, по экспертным оценкам, может достигнуть нескольких миллионов [2]. Для представления FIFO-очередей применяют различные программные или аппаратные решения [3, 4, 5].

В [6, 7] предлагались модели для последовательного, связанного и страничного способов представления нескольких FIFO-очередей в памяти одного уровня. В этих моделях предполагается, что на каждом шаге дискретного времени с заданными вероятностями происходят некоторые операции со структурами данных. Время выполнения операций это не случайная величина, а константа, поэтому фиксированным является и шаг времени. В работе [3] приведены результаты имитационных экспериментов и поставлена задача построить математическую модель процесса работы с несколькими FIFO-очередями в общей памяти, когда операции с очередями выполняются по несколько другому принципу. В данной схеме работы на нечетном шаге допускаются операции включения элементов в одну из n очередей с равными вероятностями, а на четном шаге – операции исключения элементов из очередей с равными вероятностями. Исключение из пустой очереди не приводит к завершению работы. В [3] ставилась задача определить вероятность (как функцию от n и j) того, что очередь, выбранная для операции на j-том шаге, будет пустой, а также вычислить математическое ожидание количества элементов в очередях после j операций. В данной задаче не рассматривался конкретный способ представления очередей в памяти, то есть предполагалось, что очереди могут быть неограниченной длины, что на практике невыполнимо. Эта задача была решена в [8].

В [9] предложена математическая модель этого процесса для числа очередей $n = 2$, и решается задача оптимального разбиения общей памяти для двух FIFO-очередей в случае их последовательного циклического представления в предположении, что операции с очередями выполняются по этому принципу, но с неравными вероятностями. В качестве критерия оптимальности рассмотрена минимальная доля потерянных элементов при бесконечном времени работы очередей. Эту величину разумно минимизировать, когда переполнение очереди является не аварийной, а стандартной ситуацией (здесь мы подчеркиваем, что в некоторых приложениях при переполнении очереди работа программы заканчивается, и тогда в качестве критерия оптимальности надо рассматривать максимальное среднее время до переполнения памяти). Так, если очередь занимает всю предоставленную ей память, то все последующие элементы, поступающие в нее, отбрасываются до тех пор, пока не появится свободная память (т.е. до тех пор, пока не произойдет исключение элемента из очереди). Такая схема работы применяется, например, в работе сетевых маршрутизаторов [4] в том случае, когда по мере увеличения трафика очередь на исходящем интерфейсе маршрутизатора заполняется пакетами. Такое поведение маршрутизатора называется "сбросом хвоста". Потери пакетов приводят к нежелательному результату, поэтому число таких ситуаций необходимо свести к минимуму.

В [10] построены математическая и имитационная модели процесса работы с двумя очередями, когда они двигаются по кругу друг за другом. Этот метод работы с FIFO-очередями предложен в [11].

В данной работе мы предлагаем математическую модель и решаем задачу оптимального разбиения общей памяти для двух FIFO-очередей в случае последовательного циклического представления очередей, в предположении, что операции с очередями выполняются по этому принципу, но возможно, наряду с последовательным, и параллельное выполнение операций с очередями с заданными вероятностями.

Пусть в памяти размером в m единиц мы работаем с двумя последовательными циклическими FIFO-очередями с элементами фиксированного размера в одну условную единицу.

Для последовательного представления каждой очереди выделим некоторое количество единиц памяти из общего объема равного m единиц. Пусть s – количество единиц памяти, выделенных первой очереди, тогда $(m - s)$ – количество единиц памяти, выделенных второй очереди.

Операции, производимые с очередями, выполняются по следующей схеме: на нечетном шаге происходит операция включения элемента в одну из очередей, на четном шаге – операция исключения элемента из какой-либо очереди, причем известны некоторые вероятностные характеристики операций, производимых с очередями. Пусть p_1 и p_2 – вероятности включения элемента в первую и вторую очереди, соответственно, p_{12} – вероятность одновременного включения в обе очереди. q_1 и q_2 – вероятности исключения элемента из первой и второй очередей, соответственно, q_{12} – вероятность одновременного исключения из обеих очередей.

Поскольку построенная на основе такой постановки задачи марковская цепь не будет регулярной и однородной, два последовательных шага объединяем в один, а также вводим в рассмотрение вероятности выполнения операций, не изменяющих длины очередей, например, операции чтения: r_1 – вероятность выполнения операции на нечетном шаге и r_2 – на четном, при этом $r_1 \neq 0$, $r_2 \neq 0$. Соответственно, $p_1 + p_2 + p_{12} + r_1 = 1$, $q_1 + q_2 + q_{12} + r_2 = 1$.

Тогда состояния на каждом шаге определяется наступлением одной из следующих комбинаций событий:

1. включение в первую, исключение из второй очереди с вероятностью p_1q_2;
2. включение во вторую, исключение из первой очереди с вероятностью p_2q_1;
3. включение в первую очередь с вероятностью $p_1r_2 + p_{12}q_2$;
4. включение во вторую очередь с вероятностью $p_2r_2 + p_{12}q_1$;
5. включение параллельно в обе очереди с вероятностью $p_{12}r_2$;
6. исключение из первой очереди с вероятностью $q_1r_1 + p_2q_{12}$;
7. исключение из второй очереди с вероятностью $q_2r_1 + p_1q_{12}$;
8. исключение параллельно из обеих очередей с вероятностью $q_{12}r_1$;
9. выполнение над очередями сохраняющих их состояние противоположных операций с вероятностью $r_1r_2 + p_1q_1 + p_2q_2 + p_{12}q_{12}$, где $p_1q_2 + p_2q_1 + p_1r_2 + p_{12}q_2 + p_2r_2 + p_{12}q_1 + p_{12}r_2 + q_1r_1 + p_2q_{12} + q_2r_1 + p_1q_{12} + q_{12}r_1 + r_1r_2 + p_1q_1 + p_2q_2 + p_{12}q_{12} = 1$.

Целью исследования является определение оптимального распределения памяти между очередями, когда в качестве критерия оптимальности рассматривается минимальная средняя доля потерянных при переполнении элементов очередей. По закону больших чисел для регулярных цепей Маркова [12] это эквивалентно нахождению решения, доставляющего минимум значению вероятности переполнения памяти на бесконечном промежутке времени.

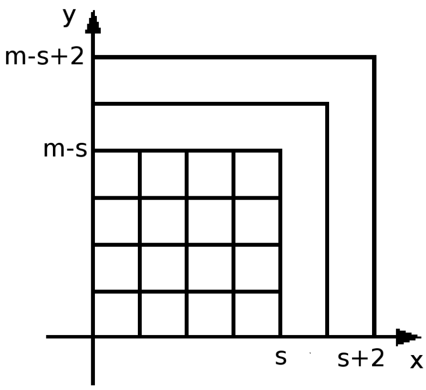

Рис. 1

Обозначим через x и y текущие длины первой и второй очередей соответственно. В качестве математической модели рассматриваем случайное блуждание в двумерном пространстве по целочисленной решетке в области $0 \leq x \leq s + 2$, $0 \leq y \leq m - s + 2$ (рис. 1), где прямые $x = s + 1$, $y = m - s + 1$ образуют первый отражающий экран – попадая на эти прямые, мы будем находиться на них до тех пор, пока не произойдет исключение элемента из очереди; прямые $x = s + 2$, $y = m - s + 2$ образуют второй отражающий экран, который определен для случаев включения элемента в заполненную очередь и немедленного исключения элемента из этой же очереди. Введением данного экрана учитывается произошедшая потеря элемента, очередь формально переходит на экран, а фактически в область $0 \leq x < s$, $0 \leq y < m - s$, конкретно на прямые $x = s - 1$ или $y = m - s - 1$.

Для решения поставленной задачи использовался аппарат управляемых случайных блужданий, регулярных цепей Маркова, система Intel® Math Kernel Library PARDISO. Были проанализированы результаты численных экспериментов, и сделаны выводы. Вычисления производились с помощью кластера КарНЦ РАН.

Перехват работы (work stealing) становится основным способом балансировки нагрузки динамических многопоточных вычислений на многоядерной аппаратуре [13]. Поэтому в будущем мы планируем рассмотреть математические модели работы с такими структурами данных как дек с ограниченным входом [5], который используется для реализации этого механизма [14, 15], и кактус-стек и его параллельную реализацию [13]. Заметим, что в [14] предложенный ими способ работы с динамическими структурами данных представлен как новый, хотя в наших работах такой метод был описан и проанализирован для представления стеков и очередей, как страничный метод, раньше [16].

Работа выполнена при финансовой поддержке гранта РФФИ (проект 12-01-00253) и Программы стратегического развития ПетрГУ в рамках реализации комплекса мероприятий по развитию научно-исследовательской деятельности.

ЛИТЕРАТУРА:

1. M. Herlihy, N. Shavit. The Art of Multiprocessor Programming. Morgan Kaufmann, 2008. 528 p.
2. A. Nikologiannis, M. Katevenis. Multi-Queue Management for Advanced QoS in High-Speed Communication Systems Computer Architecture and VLSI Systems Lab, Institute of Computer Science (ICS) Head, http://archvlsi.ics.forth.gr/muqpro/queueMgt.html
3. Р. Седжвик. Фундаментальные алгоритмы на С++. К.: Диасофт, 2001. 688 с.
4. В. Боллапрагада, К. Мэрфи, У. Расс. Структура операционной системы Cisco IOS. М.: Вильямс, 2002. 208 с.
5. Д. Кнут. Искусство программирования для ЭВМ. Т. 1. М.: Вильямс, 2001. 736 с.
6. Е.А. Аксенова, А.В. Драц, А.В. Соколов. Оптимальное управление *n* FIFO-очередями на бесконечном времени // Информационно-управляющие системы. 2009. № 1. С. 46-54.
7. E.A. Aksenova, A.V. Sokolov. The optimal implementation of two FIFO-queues in single-level memory // Applied Mathematics. 2011. Vol. 2, № 10. P. 1297-1302.
8. А.В. Драц, А.В. Соколов. Математический анализ процесса работы с M FIFO-очередями // Стохастическая оптимизация в информатике. 2012. Т. 8, № 2. С. 75-82.
9. Н.В. Каблукова, А.В. Соколов. Оптимальное разбиение общей памяти для двух последовательных циклических FIFO-очередей // Прикладная информатика. 2012. № 40. С. 113-125.
10. Н.В. Каблукова, А.В. Соколов. Математический анализ одного способа представления двух FIFO-очередей в общей памяти // Труды Карельского научного центра РАН. 2013. № 1. С. 46-54.
11. А.В. Соколов. Математические модели и алгоритмы оптимального управления динамическими структурами данных. Петрозаводск: ПетрГУ, 2002. 216 с.
12. Дж. Кемени, Дж. Снелл. Конечные цепи Маркова. М.: Наука, 1960. 272 с.
13. I.A. Lee. Memory Abstractions for Parallel Programming . Massachusetts Institute of Technology , 2012. 163 p.
14. D. Hendler, Y. Lev, M. Moir, N. Shavit. A Dynamic-Sized Nonblocking Work Stealing Deque // Distributed Computing - Special issue: DISC 04, 2006. V. 18, № 3. P. 189-207.
15. D. Chase, Y. Lev. Dynamic Circular Work-Stealing Deque // SPAA '05. Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures, 2005. P. 21-28.
16. Е.А. Аксенова, А.А. Лазутина, А.В. Соколов. Об оптимальных методах представления динамических структур данных // Обозрение прикладной и промышленной математики. 2003. Т. 10, № 2. С. 375376.