

МОДЕЛЬ И АНАЛИЗ ЭФФЕКТИВНОСТИ РАБОТЫ ДВУХ ПАРАЛЛЕЛЬНЫХ FIFO-ОЧЕРЕДЕЙ В ОБЩЕЙ ПАМЯТИ

А.В. Соколов, Е.А. Барковский

ИПМИ КарНЦ РАН, ПензГУ

Во многих приложениях требуется работа с несколькими FIFO-очередями, расположенными в общем пространстве памяти. Эффективные алгоритмы такой работы 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. Для представления FIFO-очередей применяют различные программные или аппаратные решения [1, 2, 3].

Отметим, так же, что среди архитектур многоядерных процессоров есть и такие, где отсутствует кэш-память. Например, в архитектуре AsAP-II каждое ядро имеет два FIFO-буфера, а в архитектуре SEAForth – два стека для хранения данных и адресов возвратов [4]. В этих архитектурах очереди и стеки реализованы циклически и раздельно с возможностью потери элементов при переполнении. Мы же исследуем ситуации, когда для хранения нескольких структур данных используется общая память, что в ряде случаев позволяет снизить потери элементов при переполнении.

В [5, 6, 7, 8] 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 [1] 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 [1] 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 [9].

В [10,11] мы предложили математическую модель и решали 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. Решалась 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). 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 [2] 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 данной работе мы предлагаем математическую и имитационную модели процесса работы с двумя параллельными очередями, когда они двигаются по кругу друг за другом. Этот метод работы с FIFO-очередями предложен в [12].

Пусть в памяти размера m единиц мы работаем с двумя циклическими параллельными FIFO-очередями, которые двигаются друг за другом по кругу. Возобновление движения пустой очереди начинается с середины промежутка между очередями.

Операции, производимые с очередями, выполняются по следующей схеме: на нечетном шаге происходит операция включения в одну из очередей, на четном шаге – операция исключения 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 являться регулярной и однородной, два последовательных шага объединяем в один, а также вводим следующие вероятности операций, не изменяющих длины очередей (например, чтение): r_1 – на нечетном шаге и r_2 – на четном шаге, при этом $r_1 \neq 0, r_2 \neq 0$.

Соответственно, $p_1 + p_2 + r_1 = 1, q_1 + q_2 + r_2 = 1$.

Тогда состояние 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 $p_1q_2 + p_2q_1 + p_1r_2 + p_{12}q_2 + p_2r_2 + p_{12}q_1 + p_{12}r_2 + q_1r_1 + p_2q_{12} + q_2r_1 + p_1q_{12} + q_{12}r_1 + r_1r_2 + p_1q_1 + p_2q_2 + p_{12}q_{12} = 1$.

Предполагается, что в очередях хранятся данные фиксированного размера. При исключении информации из пустой очереди не происходит завершения работы. Схема движения очередей показана на Рис. 1.

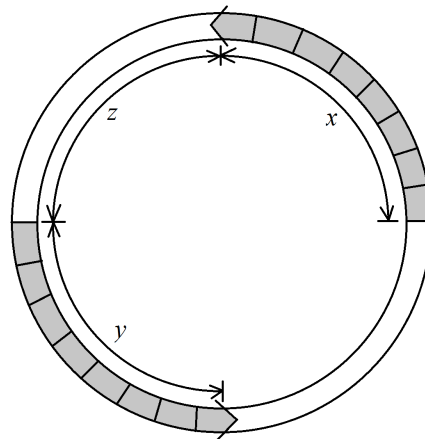


Рис. 1

Целью исследования является определение средней доли потерянных элементов для сравнения со средней долей потерянных элементов при последовательном циклическом способе организации параллельных очередей в случае оптимального разбиения общей памяти [10].

Обозначим через x и y текущие длины очередей, через z – расстояние между концом первой очереди и началом второй. В качестве математической модели рассматриваем блуждание по целочисленной трехмерной пирамиде с вершиной $(0; 0; 0)$ и основанием $x + y + z = m$.

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.

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

ЛИТЕРАТУРА:

1. Р. Седжвик. Фундаментальные алгоритмы на С++. К.: Диасофт, 2001. 688 с.
2. В. Боллапрагада, К. Мэрфи, У. Расс. Структура операционной системы Cisco IOS. М.: Вильямс, 2002. 208 с.
3. Д. Кнут. Искусство программирования для ЭВМ. Т. 1. М.: Вильямс, 2001. 736 с.

4. А.В. Калачев. Многоядерные процессоры. Интернет-Университет Информационных технологий: БИНОМ. Лаборатория знаний. 2010. 247 с.
5. 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.
6. 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.
7. A.V. Sokolov, A.V. Drac. The linked list representation of n LIFO-stacks and/or FIFO-queues in the single-level memory // Information Processing Letters 2013, Vol.13, Iss.19-21, pp. 832-835
8. А.В. Соколов, А.В. Драц. Моделирование некоторых методов представления n FIFO-очереди в памяти одного уровня // Эвристические алгоритмы и распределенные вычисления. 2014. Т.1, №1, С. 40-52
9. 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.
10. А.В. Соколов, Е.А. Барковский. Анализ эффективности параллельной работы с двумя FIFO-очередями в общей памяти // Труды международной суперкомпьютерной конференции “Научный сервис в сети интернет: все грани параллелизма”. Российская академия наук. Суперкомпьютерный консорциум университетов России. Изд-во МГУ, 2013. С. 51-53.
11. А.В. Соколов, Е.А. Барковский. Оптимальное разбиение общей памяти для двух параллельных FIFO-очереди // Стохастическая оптимизация в информатике. 2013. Т.9, №2, С. 124-137.
12. А.В. Соколов. Математические модели и алгоритмы оптимального управления динамическими структурами данных. Петрозаводск: ПетрГУ, 2002. 216 с.