

ИССЛЕДОВАНИЕ ПОДСИСТЕМЫ «КЭШ-ПАМЯТЬ – ОПЕРАТИВНАЯ ПАМЯТЬ» МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Н.В. Ефимушкина, М.М. Ефремов, С.П. Орлов

Самарский государственный технический университет
443100, г. Самара, ул. Молодогвардейская, 244
E-mail: orlovsp1946@gmail.com

Рассмотрены основные способы организации кэш-памяти современных вычислительных систем и принципы построения их моделей. С помощью имитационных моделей выполнен анализ наиболее важных временных характеристик подсистемы памяти и определены эффективные режимы ее работы.

Ключевые слова: *имитационная модель, оперативная память, кэш-память, многопроцессорная система.*

Введение. В настоящее время для согласования скоростей работы устройств вычислительных систем применяют иерархическую структуру памяти [1]. Для однопроцессорных систем достаточно подробно исследованы условия эффективной организации подсистемы «кэш-память – оперативная память» [2]. Однако для многоядерных процессоров и многопроцессорных систем довольно сложно получить аналитические решения, описывающие процессы обмена данными между уровнями памяти. В этом случае целесообразно использовать имитационное моделирование. В статье рассматривается объект моделирования – подсистема памяти компьютера и описаны результаты имитационного моделирования различных режимов работы.

Объект моделирования. Рассматривается многоуровневая структура памяти современного компьютера [3]. Она содержит четыре уровня: устройства кэш-памяти первого, второго и третьего уровня и оперативную память, находящуюся на нижнем уровне иерархии. При этом используется принцип локальности ссылок: программа и обрабатываемые ею данные обычно располагаются в последовательных ячейках памяти и образуют компактные области, а следующее обращение к памяти при выполнении программы с большой вероятностью происходит к тому же блоку данных, который находится в данный момент в памяти верхнего уровня. В процессе выполнения программы происходит активное взаимодействие памяти всех уровней. Результатом обращения к памяти верхнего уровня (кэш) может быть попадание или промах. Промах означает, что объект не найден в кэше, и его поиск быть продолжен на более низком уровне.

Поскольку повышение производительности является главной причиной иерархической организации памяти, частота попаданий и промахов является важной характеристикой. Время обращения при попадании есть время обращения к более высокому уровню иерархии. Оно включает в себя, в частности, и время, необходимое для определения результата (попадание или промах). Потери на промах – это время

Наталья Владимировна Ефимушкина (к.т.н.), доцент кафедры «Вычислительная техника».

Михаил Михайлович Ефремов, аспирант.

Сергей Павлович Орлов (д.т.н., профессор), заведующий кафедрой «Вычислительная техника».

для замещения блока более высокого уровня на блок из более низкого плюс время для пересылки этого блока в требуемое устройство (обычно в процессор). Описываемые потери включают в себя две компоненты: время доступа (время обращения к первому слову блока при промахе) и время пересылки оставшихся слов блока. Первая компонента связана с задержкой памяти более низкого уровня, в то время как время пересылки определяется полосой пропускания канала между устройствами памяти двух смежных уровней.

Память любого типа может иметь различные структуры и режимы функционирования. Определение оптимальных структур и режимов для различных классов программ и вычислительных систем является сложной задачей. В предлагаемой работе исследовалась наиболее важная подсистема «кэш – оперативная память». Физические эксперименты над этой подсистемой организовать весьма сложно, поэтому было принято целесообразным использовать метод имитационного моделирования. Метод является универсальным и позволяет исследовать системы любой сложности, с любыми режимами функционирования. При этом имеется возможность воспроизводить в модели наиболее важные особенности объекта и исключать второстепенные.

В современных вычислительных системах (ВС) используются три типа организации кэш-памяти [4]:

а) кэш с прямым отображением (каждая строка кэша может содержать строку основной памяти только из определенного подмножества адресов, причем эти подмножества не пересекаются);

б) полностью ассоциативный кэш (любая строка ОП может находиться в любой строке кэша);

в) множественно-ассоциативный кэш (кэш-память делится на непересекающиеся подмножества). Каждая строка основной памяти может попасть в любое место только одного подмножества кэша.

При выполнении очередной команды процессор запрашивает операнд по некоторому адресу. Если соответствующая строка имеется в кэше, то происходит кэш-попадание, иначе – кэш-промах. В последнем случае возникает проблема замены некоторой строки в кэше на новую строку. Для этого предлагаются различные дисциплины замещения строк:

1) замещение строки, к которой дольше всего не обращались (алгоритм LRU – Least Recently Used);

2) вытеснение строки, которая была загружена раньше всех (алгоритм FIFO – First Input First Output);

3) замена наименее часто запрашиваемой строки (алгоритм LFU – Least Frequently Used);

4) замена случайной строки (Random-алгоритм).

Наиболее часто применяются две стратегии: случайный выбор Random и LRU.

В структуре кэш-памяти выделяют два типа блоков данных:

а) отображения данных (собственно сами данные, дублированные из оперативной памяти);

б) теги, содержащие признаки, указывающие на расположение кэшированных данных в оперативной памяти.

Пространство памяти отображения данных в кэше разбивается на строки – блоки фиксированной длины (например, 32, 64 или 128 байт). Каждая строка кэша может содержать непрерывный выровненный блок байт из оперативной памяти. Какой блок ОП отображен на данную строку кэша, определяется тегом строки и алгорит-

мом отображения. Именно по алгоритмам отображения оперативной памяти в кэш и выделяют три типа кэш-памяти, которые перечислены выше.

Для полностью ассоциативного кэша характерно, что кэш-контроллер может поместить любой блок оперативной памяти в любую строку кэш-памяти. В этом случае физический адрес разбивается на две части: смещение в блоке (строке кэша) и номер блока. При помещении блока в кэш его номер сохраняется в теге соответствующей строки. Когда процессор обращается к кэшу, промах будет обнаружен только после сравнения тегов всех строк с номером блока.

Одно из основных достоинств рассматриваемого способа отображения – хорошая возможность размещения больших массивов данных, т. к. нет ограничений на то, какой блок может быть отображен на ту или иную строку кэш-памяти. При этом сокращается количество промахов. К недостаткам относят сложную аппаратную реализацию этого способа, требующую большого количества схем (в основном компараторов), что приводит к увеличению времени доступа к кэшу и увеличению его стоимости.

Альтернативный способ используется в кэше прямого отображения. В нем адрес памяти (номер блока) однозначно определяет строку кэша, в которую будет помещен данный блок. Физический адрес разбивается на три части: смещение в блоке (строке кэша), номер строки кэша и тег. Тот или иной блок будет всегда помещаться в строго определенную строку кэша, при необходимости заменяя собой хранящийся там другой блок. Если процессор обращается к кэшу за данными, достаточно проверить тег лишь одной строки.

Очевидными преимуществами описываемой структуры являются простота и дешевизна реализации. К недостаткам следует отнести низкую эффективность такого кэша из-за возможных частых перезагрузок строк.

Компромиссным вариантом между первыми двумя структурами является множественный ассоциативный или частично-ассоциативный кэш. При этом способе организации кэш-памяти строки объединяются в группы, в которые могут входить 2, 4, ... строк. В соответствии с количеством строк в таких группах различают 2-входный, 4-входный и т. п. ассоциативный кэш.

При обращении к памяти физический адрес разбивается на три части: смещение в блоке (строке кэша), номер группы (набора) и тег. Блок памяти, адрес которого соответствует определенной группе, может быть размещен в любой строке этой группы, и в теге строки размещается соответствующее значение. Очевидно, что в рамках выбранной группы соблюдается принцип ассоциативности. С другой стороны, тот или иной блок может попасть только в строго определенную группу, что перекликается с принципом организации кэша прямого отображения. Для того чтобы процессор смог идентифицировать кэш-промах, ему надо будет проверить теги лишь одной группы (2/4/8/ строк).

Описанный алгоритм отображения сочетает достоинства как полностью ассоциативного кэша (хорошее использование памяти и высокая скорость), так и кэша прямого отображения (простота и дешевизна), лишь незначительно уступая по перечисленным характеристикам исходным структурам. Именно поэтому множественный ассоциативный кэш получил наиболее широкое распространение.

Критерием эффективной работы кэша можно считать уменьшение среднего времени доступа к памяти по сравнению с системой без него. В таком случае среднее время доступа можно оценить следующим образом:

$$T_{cp} = (T_{hit} \times R_{hit}) + (T_{miss} \times (1 - R_{hit})),$$

где T_{hit} – время доступа к кэш-памяти в случае попадания (включает время на иден-

тификацию промаха или попадания);

T_{miss} – время, необходимое на загрузку блока из основной памяти в строку кэша в случае кэш-промаха и последующую доставку запрошенных данных в процессор;

R_{hit} – частота попаданий.

Очевидно, что чем ближе значение R_{hit} к 1, тем ближе значение T_{cp} к T_{hit} . Частота попаданий определяется в основном архитектурой кэш-памяти и ее объемом. Влияние этого фактора на рост производительности процессора иллюстрирует табл. 1.

Таблица 1

Размер и эффективность кэш-памяти

Размер кэш-памяти, Кб	Частота попаданий, %	Рост производительности, %
Нет кэш-памяти	–	0
16	81	35
32	86	38
64	88	39
128	89	39

Кэш первого уровня является составной частью всех современных CPU. Кэш второго уровня (L2) начали использовать на всех процессорах после выхода Pentium III. Современные процессоры оснащаются до 6 Мбайт кэш-памяти L2 на кристалле. Она разделяется между двумя ядрами, например процессора Intel Core 2 Duo.

Кэш третьего уровня стал применяться процессорах Intel Itanium 2, Pentium 4 Extreme и Xeon MP. Современные архитектуры используют L3 как большой общий буфер для обмена данными между ядрами в многоядерных процессорах. Современные четырехъядерные процессоры имеют выделенные кэши L1 и L2 для каждого ядра, а также большой кэш L3, являющийся общим для всех ядер.

Имитационное моделирование подсистемы «кэш-память – оперативная память». Типичным для памяти современных ВС является набор характеристик, приведенных в табл. 2. С помощью программ имитационного моделирования, разработанных авторами [5, 6, 7], проведены исследования структур подсистем всех описанных типов.

Таблица 2

Типичные характеристики современной кэш-памяти

Наименование	Значение
Размер блока (строки)	4-128 байт
Время попадания (hit time)	1-4 такта синхронизации (обычно 1 такт)
Потери при промахе (miss penalty): – время доступа (access time) – время пересылки (transfer time)	8-32 такта синхронизации: – 6-10 тактов синхронизации; – 2-22 такта синхронизации
Доля промахов (miss rate)	1 – 20 %
Размер кэш-памяти	4 Кбайт – 16 Мбайт

Графики зависимости между основными параметрами и характеристиками памяти для однопроцессорной ВС приведены на рис. 1, а, рис. 1, б и рис. 2. Длина

последовательности команд – 1000 простых команд, объем кэша измерялся в килобайтах. Время выполнения определяется количеством машинных тактов.

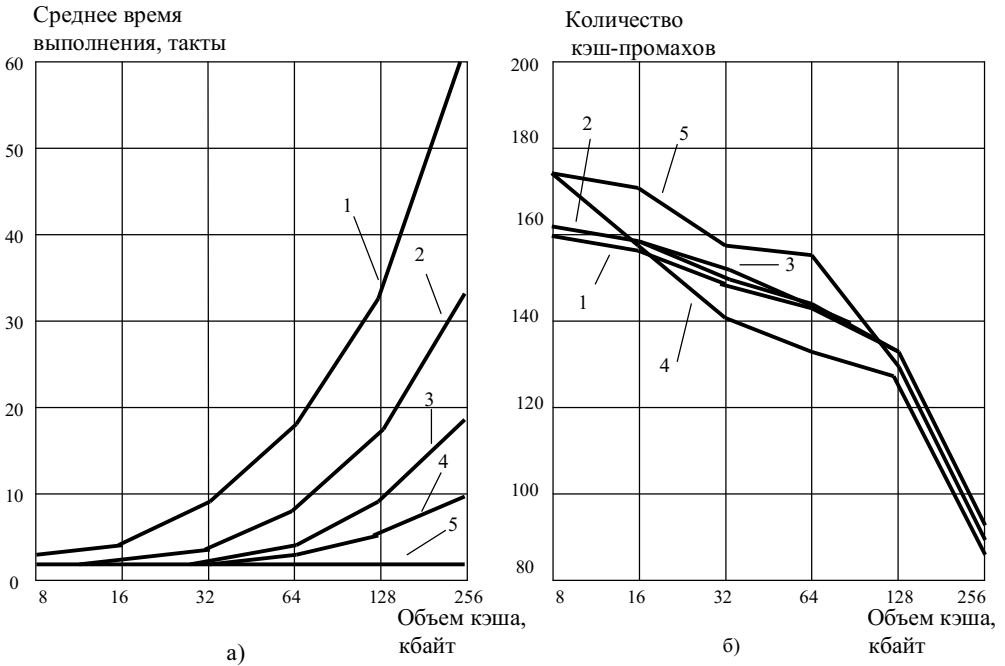


Рис. 1. Зависимость параметров обмена данными от объема кэш памяти:

а) среднее время выполнения команды; б) количество кэш-промахов;

1 – кэш с прямым отображением; 2 – полностью ассоциативный кэш; 3 – множественно-ассоциативный кэш с двумя подмножествами; 4 – множественно-ассоциативный кэш с четырьмя подмножествами; 5 – множественно-ассоциативный кэш с восемью подмножествами

Анализ графиков показывает, что модели адекватно воспроизводят процесс функционирования подсистемы «кэш – оперативная память». Наивысшую производительность и наилучшие временные характеристики обеспечивает полностью ассоциативный кэш, худшие – устройство с прямым отображением, а множественно-ассоциативный кэш занимает промежуточное положение. Методы замещения строк имеют близкие характеристики, за исключением алгоритма LFU, который является самым неэффективным.

Принцип локальности ссылок при однопрограммном режиме работы процессоров приводит к тому, что в кэшах любого типа массивы данных из оперативной памяти занимают соседние блоки. Поэтому число промахов зависит только от объема кэша. Исследования на модели подтвердили, что у всех трех типов кэш-памяти количества кэш-промахов имеют близкие значения. Размер обрабатываемого массива данных (параметр «длина цикла» на рис. 3, а) практически не влияет на среднее время выполнения команды и количество кэш-промахов до тех пор, пока он не превысит объем кэш-памяти. В последнем случае число промахов резко возрастает, так как при каждом повторении цикла обработки приходится подкачивать часть данных, которые не поместились в кэш.

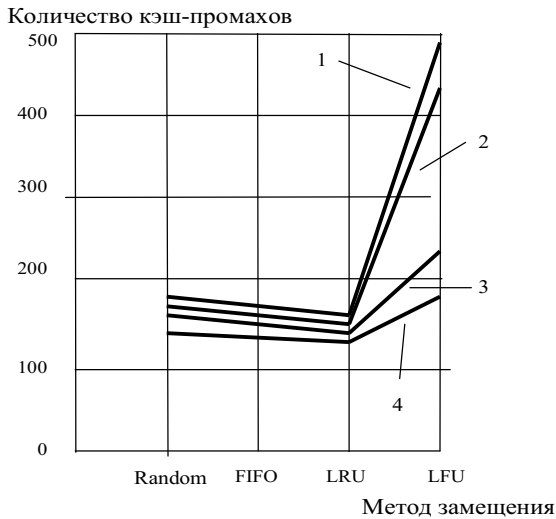


Рис. 2. Зависимость количества промахов от вида кэша и метода замещения строк:
 1 – полностью ассоциативный кэш; 2 – множественно-ассоциативный кэш с двумя подмножествами; 3 – множественно-ассоциативный кэш с четырьмя подмножествами; 4 – множественно-ассоциативный кэш с восемью подмножествами

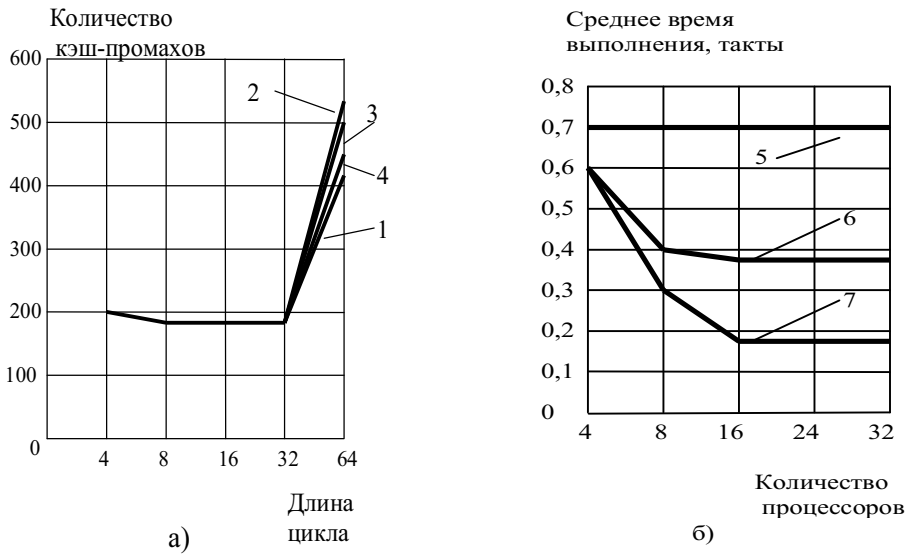


Рис. 3. График зависимости: а) количества кэш-промахов от длины цикла; б) среднего времени выполнения команды от количества процессоров;

1 – кэш с прямым отображением; 2 – полностью ассоциативный кэш; 3 – множественно-ассоциативный кэш с двумя подмножествами (двухканальный); 4 – множественно-ассоциативный кэш с четырьмя подмножествами (четырёхканальный); 5 – кэш и локальная память ЛП; 6 – только кэш-память; 7 – нет кэш-памяти

Графики зависимости между основными параметрами и характеристиками для многопроцессорной вычислительной системы приведены на рис. 3, б. Здесь через ЛП обозначена локальная память (кэш уровня L3). Анализ графиков показывает, что наиболее эффективной является трехуровневая организация памяти, включающая в себя кэш, локальную и общую память. Из рис. 3, б следует, что с ростом числа цен-

тральных процессоров от 4 до 16 увеличивается количество одновременно выполняемых команд, что приводит к уменьшению среднего времени выполнения одной команды. При числе процессоров от 16 до 32 и более узким местом становится системная шина. Поэтому дальнейший рост количества процессоров не приводит к увеличению производительности вычислительной системы.

Заключение. Методика имитационного моделирования многоуровневой памяти может быть использована при проектировании управляющих систем на основе компьютеров с многоядерными процессорами. Имитационные модели памяти также эффективны при изучении свойств кэш-памяти в специальных дисциплинах бакалавриата по направлению 230100 «Информатика и вычислительная техника».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Столлингс В.* Структурная организация и архитектура компьютерных систем: Пер. с англ. – Изд. 5-е. – М.: Вильямс, 2002. – 896 с.
2. *Цилькер Б.Я., Орлов С.А.* Организация ЭВМ и систем: учебник для вузов. – СПб.: Питер, 2004. – 668 с.
3. *Таненбаум Э.* Архитектура компьютера: пер. с англ. – Изд. 5-е. – СПб., 2010. – 848 с.
4. *Хамахер К., Вранешич З., Заки С.* Организация ЭВМ: пер. с англ. / Сер. Классика computer science. – Изд. 5-е. – СПб: Питер, 2003. – 845 с.
5. *Ефимушкина Н.В., Орлов С.П.* Вычислительные системы и комплексы: Учеб. пособие. – М.: Машиностроение-1, 2006. – 268 с.
6. *Ефремов М.М.* Имитационные модели для исследования подсистемы памяти современных вычислительных систем // Сборник тезисов докладов участников XXIV Всероссийской конференции обучающихся «Национальное достояние России». – Минобрнауки РФ, Рособразование, РОСКОС-МОС, РАО, НС «ИНТЕГРАЦИЯ», 2009. – 1428 с. – С. 363-364.
7. *Ефремов М.М.* Методы изучения подсистем памяти современных вычислительных систем // Приоритетные направления современной российской науки глазами молодых ученых: Всероссийская научно-практич. конф. молодых ученых и специалистов, 4-6 ноября 2009 г., г. Рязань / Отв. ред. А.Н. Козлов; Ряз. гос. ун-т им. Есенина. – Рязань, 2009. – 434 с. – С. 143-145.

Статья поступила в редакцию 24 октября 2011 г.

RESAERCH SUBSYSTEM «CACHE-MEMORY – RANDOM ACCESS MEMORY» IN MULTYPROCESSOR COMPUTER SYSTEMS

N.V. Efimushkina, M.M. Efremov, S.P. Orlov

Samara State Technical University
244, Molodogvardeyskaya st., Samara, 443100

The main ways of organizing memory cache of modern computational systems and the principles of their models are considered. Proposed simulation models to analyze the most important temporal characteristics of the memory subsystem and identifies effective modes of operation.

Keywords: *simulation modeling, random access memory, cache, multiprocessor computer systems.*

*Natalia V. Efimushkina (Ph.D. (Techn.)), Associate professor.
Michail M. Efremov, Postgraduate student.
Sergey P. Orlov (Dr. (Techn.)), Professor.*