

Структура и производительность подсистем памяти современных вычислительных платформ

П. А. Гаврилушкин*

В статье описывается архитектура подсистем памяти современных вычислительных платформ. Рассматривается взаимосвязь эффективности выполнения программ и особенности организации работы с памятью.

1. Введение

Скорость работы процессора несравнимо выше скорости работы подсистемы памяти. Эта асимметричность обязывает разработчиков программного обеспечения очень тщательно подходить к налаживанию взаимодействия между процессором и памятью. На всех современных компьютерных платформах время ожидания данных из памяти на порядок больше времени исполнения операции, что не может не сказаться на времени исполнения программы в целом. Каким же образом можно повысить эффективность работы с подсистемой памяти? Во-первых, это возможность установки более мощной аппаратной базы: процессор с более быстрым и большим кэшем, материнская плата с более производительными каналами связи и контроллером памяти, высокоскоростные модули оперативной памяти и внешние источники данных. И во-вторых, это соответствие алгоритма архитектуре памяти.

2. Архитектура памяти

Запоминающие устройства могут быть разделены как по назначению, так и по физическим принципам построения. Если не брать в

*Научно-исследовательский вычислительный центр МГУ

рассмотрение специальную память, как то: постоянную (ROM), перепрограммируемую (Flash), энергонезависимую память, применяемую для хранения установок BIOS (CMOS RAM), видеопамять, то остальная память компьютера соответствует иерархической структуре (рис. 1). И для нее справедливо следующее правило: процессор и каждый из уровней иерархии может обращаться на чтение и запись только к ближайшему снизу уровню. Причем, более высокий уровень имеет меньший объём, бóльшую стоимость и обладает бóльшим быстродействием.

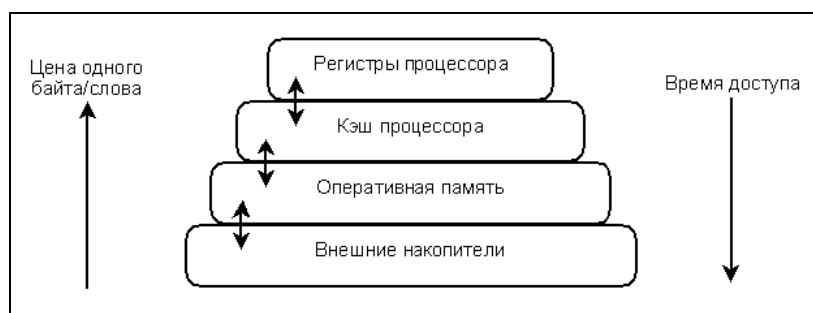


Рис. 1. Иерархия памяти

Рассмотрим основные параметры подсистемы памяти. У каждого запоминающего устройства (ЗУ) есть такие параметры, как:

- латентность — время ожидания (в мс. или тактах) реакции ЗУ на поступивший запрос;
- объём — максимальное количество одновременно хранимой информации (байт) в ЗУ;
- единица доступа — минимальное количество байт, которое можно получить за одно обращение к ЗУ;
- адресуемость — возможность обращения к данным в ЗУ по адресу;

- продолжительность сохранения — постоянная (ПЗУ) или временная (ОЗУ);
- энергопотребление — количество потребляемой ЗУ энергии на операции чтения, хранения и записи.

В процессе развития аппаратного обеспечения компьютеров латентность постепенно уменьшается, а объём увеличивается. С точки зрения программного обеспечения, необходимо отметить свойства локальности исполняемых команд и используемых данных. На практике, это означает что выборка очередной команды (порция данных) из всех команд (данных) происходит не равновероятно, а в зависимости от предыдущей. Балансировка соотношения стоимости/производительности и принцип локальности обращений привели к иерархии памяти (быстрая память маленького объёма, а более медленная — большого). Рассмотрим подробнее каждый из уровней иерархии, начиная с ближайшего к процессору.

3. Регистровая память

На вершине иерархии стоит регистровая память. Чтение и запись в нее происходит на частоте процессора и с нулевой латентностью. Каждому регистру соответствует уникальное имя, по которому можно обратиться как по обычному адресу. Количество регистров и их структура отличаются для разных архитектур процессоров: x86, x86-64, PowerPC и другие. Современные процессоры с архитектурой x86-64 обладают 64-битными регистрами данных, регистрами указателей, сегментными регистрами, регистрами флагов, а также служебными регистрами. Общий объём регистровой памяти исчисляется сотнями байт. Для повышения производительности программ используются расширенные наборы регистров, например восемь MMX-регистров размером по 64 бита.

4. Кэш-память

Следующий уровень в иерархии занимает кэш-память или просто кэш. Он может быть подразделён на уровни (L1, L2, L3), также

состоящие в иерархии. L1-кэш является неотъемлемой частью процессора, работает с ним на одной частоте (частоте ядра), обладает небольшим объёмом (до 128Кб) и латентностью в 2-4 такта. Обычно он состоит из кэша для данных (L1D) и кэша для инструкций (L1I). Их объёмы и латентности могут различаться между собой. Кэши L2 и L3 имеют бóльший размер (от мегабайта), доступ к ним характеризуется ещё бóльшей латентностью (десятки тактов) и, в отличие от L1, могут быть отчуждены от процессора (не присутствовать вообще, не присутствовать на кристалле, иметь возможность программного отключения). Кэш L3 используется, в основном, в серверном сегменте (процессоры серии Xeon, PowerPC, UltraSPARC T). Если в одном процессоре присутствует несколько ядер или в системе несколько процессоров, то L1 всегда свой для каждого ядра. Кэши L2 также могут присутствовать по одному на каждое ядро, а могут быть динамически разделяемыми между несколькими ядрами.

Кэш-память формируется из строк (lines). Каждая строка, с учетом локальности использования данных, соответствует группе соседних байт основной памяти. Это соответствие определяется адресными полями. Адресные поля хранятся для строк целиком, а не для отдельных байт, тем самым достигается экономия по занимаемому ими месту.

В отличие от остальных запоминающих устройств, кэш не является адресуемым. Обращение к данным кэша происходит неявно и не по внутреннему адресу, а по адресу в оперативной памяти. Для чтения из ячейки, имеющей некоторый адрес в оперативной памяти, необходимо осуществить поиск по соответствующему адресу и в кэше. Успешный поиск называется попаданием (cache hit), а неуспешный — промахом (cache miss). Латентность при чтении данных, уже находящихся в нём, отличается на порядок от латентности чтения данных, требующих предварительной подкачки из кэшей более низких уровней или основной памяти. Количество неуспешных поисков напрямую влияет на эффективность исполнения программы, так как за каждый промах приходится платить увеличением латентности (miss penalty).

Существует три типа [1] кэш-памяти, отличающихся по способу

размещения данных основной памяти:

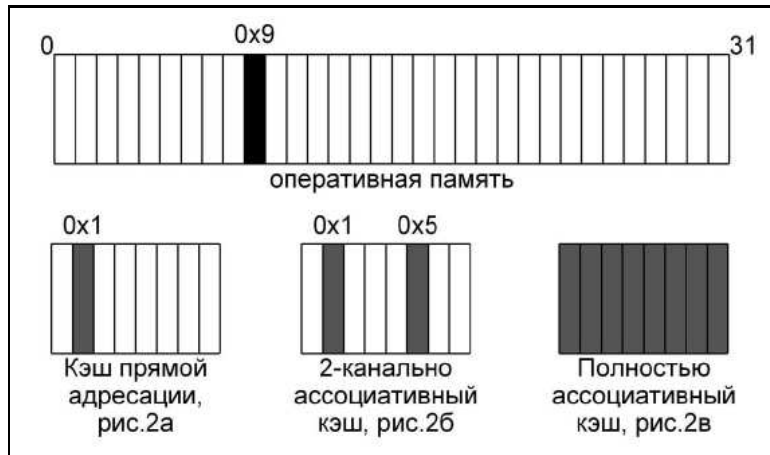


Рис. 2. Ассоциативность кэша. В верхней части рисунка изображен участок основной памяти размером в 32 байта. А в нижней части различные варианты организации кэш-памяти.

- Кэш прямой адресации (Direct Mapped Cache). Кэш, в котором каждому возможному значению поля виртуального адреса соответствует ровно одна строка, определяемая по значению этого поля (виртуальный адрес = младшие биты адреса в памяти). Такая организация неизбежно влечёт отображение различных ячеек памяти с одинаковыми младшими частями адресов в одну строку кэша, то есть коллизия (collision). Этот пример иллюстрирован на рис.2а: адреса 0x9 и 0x17 имеют три одинаковых младших бита, и поэтому оба отобразятся в виртуальный адрес 0x1.
- Полностью ассоциативный кэш (Fully Associative Cache). Этот тип кэш-памяти решает проблему коллизий, которая характерна для кэша прямой адресации. В случае полностью ассоциативного кэша блок данных из любого адреса оперативной

памяти может храниться в любой строке кэша, а сам адрес используется как тег кэша; при проверке на совпадение все теги должны одновременно сравниваться с адресом запроса, что требует дополнительного усложнения аппаратуры. Для устранения проблемы некогерентности используется ещё один тег — тег актуальности данных. На рис. 2в серым цветом отмечены ячейки, в которые может попасть ячейка с адресом 0x9 при занесении. Для полностью ассоциативного кэша — это все доступные ячейки.

- N-канально ассоциативный кэш (N-Way Set Associative Cache). Строки N-канально ассоциативного кэша делятся на секторы, или секции (sets). Информация по некоторому адресу оперативной памяти может храниться в N местах кэш-памяти. Этот способ адресации позволяет получить значительное преимущество по скорости перед кэшем прямой адресации, но имеет гораздо более простую аппаратную реализацию, нежели чем полностью ассоциативный кэш. На рис. 2б серым цветом показаны два возможных места хранения ячейки с адресом 0x9 при занесении. Это адреса 0x1 и 0x5, обладающие смещением 0x1 в каждом из каналов кэша.

Запись ячейки в кэш, по своей сути, никогда не требует подкачки данных из основной памяти или других уровней кэша, поэтому выполняется за гарантированное время. Но и здесь есть свои тонкости, а именно: необходимость поддержания актуальности (когерентности) данных как на различных уровнях кэша, так и в основной памяти. Обеспечение когерентности решается аппаратно на уровне контроллера кэша. Либо когерентность соблюдается всегда, либо при возникновении конфликта, всё зависит от политики записи при кэшировании.

Существуют две основные [2] политики записи при кэшировании:

- Сквозная запись (write-through). Подразумевает, что при изменении содержимого ячейки памяти, запись происходит синхронно и в кэш, и в основную память. Такая политика используется в L1 кэше процессоров Intel Pentium 4, Intel Pentium D.

- Отложенная запись (write-back). Подразумевает, что можно отложить момент записи цифрованных данных в основную память, записав их только в кэш. При этом модифицированные данные будут выгружены в оперативную память только в случае обращения к ним какого либо другого устройства (другое ядро ЦП, контроллер DMA) либо нехватки места в кэше для размещения других данных. Такая политика используется в L1 кэше процессоров семейства Intel Core, AMD Athlon64.

Современные алгоритмы вытеснения малоиспользуемых данных и предвыборки следующих команд позволяют достичь уровня попаданий порядка 90% при соблюдении локальности обращений. Примером снижения (сильного снижения) производительности при отсутствии свойств локальности обращений может служить последовательная выборка (выборка через интервал, на единицу большой длины строки) из памяти элементов массива, целиком не уместяющегося в кэш. Типичные характеристики кэшей процессоров серии Intel Core 2 Duo: 32+32КБ (кэш данных+кэш инструкций) 8-канально ассоциативного кэша первого уровня с размером строки равным 64 байта и политикой сквозной записи в менее скоростную память, 4096КБ 16-канально ассоциативного кэша второго уровня с размером строки равным 64 байта и политикой отложенной записи в менее скоростную память, кэш третьего уровня не поддерживается.

5. Оперативная память

Рассмотрим в хронологическом порядке технологии, применяемые в современной оперативной памяти DDR SDRAM (синхронная динамическая память с произвольным доступом и удвоенной скоростью передачи данных).

Память с произвольным доступом (RAM). Один из видов памяти, позволяющий обращаться к данным на чтение и запись, не учитывая порядок их расположения.

Динамическая память с произвольным доступом (DRAM) — это один из видов памяти с произвольным доступом. Динамическая память является энергозависимой и нуждается в периодической реге-

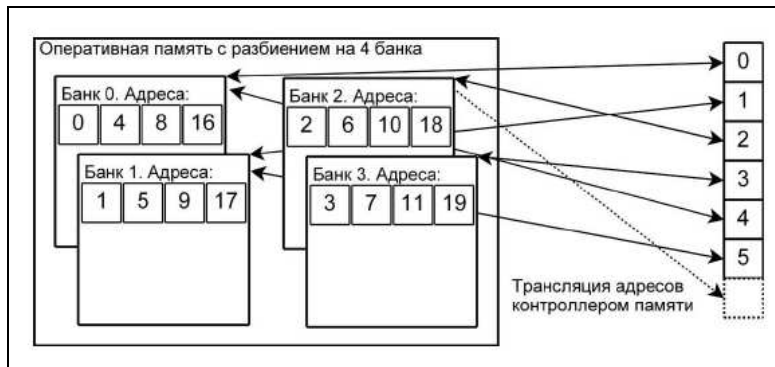


Рис. 3. Разбиение на банки

нерации данных.

Структура динамической памяти с произвольным доступом следующая: группы битовых ячеек образуют строки, группы строк образуют страницы, группы страниц образуют банки. Для ускорения доступа к данным запоминающего устройства ячейки с соседними адресами хранятся в различных банках, обеспечивая снижение влияния задержек регенерации.

Основными характеристиками памяти DRAM являются тайминги (timings) и рабочая частота. Для обращения к ячейке контроллер задаёт номер банка, номер страницы в нём, номер строки и номер столбца. Каждое из этих действий выполняется за некоторое время — тайминг. Основными таймингами DRAM [3] являются:

1. Задержка между подачей номера столбца и получением содержимого ячейки, называемая временем рабочего цикла (CAS).
2. Задержка между подачей номера строки и номера столбца, называемая временем полного доступа (RAS to CAS).
3. Задержка между чтением последней ячейки и подачей номера новой строки (RAS precharge).

№ сервера	Размер L1 (Кб)	Ассоциативность L1	Размер строки L1	Размер L2 (Мб)	Н-канальная ассоциативность L2	Размер строки L2	Частота памяти (МГц)	Частота шины памяти (МГц)
1	64+64	2	64	1+1	16	64	200	200
2	64+64	2	64	1+1	16	64	200	200
3	32+32	8	64	4	16	64	133	266
4	32+32	8	64	4	16	64	133	266
5	32+32	8	64	4	16	64	166	333
6	32+32	8	64	4+4	16	64	166	333
7	64+64	2	64	1	16	64	166	166
8	16+16	8	64	2	8	64	133	266

Таблица 1. Характеристики и структура памяти современных платформ

4. Задержка для регенерации строки памяти (RAS Active to Precharge Delay, t_{RAS}).

Эти величины измеряются в тактах, и чем они меньше, тем быстрее работает оперативная память. Записываются все четыре значения через дефис, например 3-4-4-8.

Синхронная динамическая память с произвольным доступом (SDRAM). В отличие от других типов DRAM, использовавших асинхронный обмен данными, ответ на поступивший в устройство управляющий сигнал возвращается не сразу, а лишь при получении следующего тактового сигнала. Тактовые сигналы позволяют организовать работу SDRAM в виде конечного автомата, исполняющего команды. При этом команды могут поступать в виде непрерывного потока, не дожидаясь, пока будет завершено выполнение преды-

№ сервера	Лагентность L1 (такты)	Лагентность L2 (такты)	Лагентность памяти(такты)	Тайминги памяти (CAS, RAS to CAS, RAS, tRAS)	Средняя пропускная способность памяти при чтении (ГБ/с)	Максимальная пропускная способность памяти при чтении (ГБ/с)
6	3	14	140	5-5-5-15	5,27	6,47
7	3	17	144	2,5-3-3-7	3,03	5,03
8	4	28,5	106	4-3-3-8	5,06	6,37

Таблица 2. Временные характеристики памяти и реальная пропускная способность по результатам тестов [4]

дущих инструкций (конвейерная обработка): сразу после команды записи может поступить следующая команда, не ожидая, когда данные окажутся записаны. Поступление команды чтения приведёт к тому, что на выходе данные появятся с некоторой задержкой.

Синхронная динамическая память с произвольным доступом и удвоенной скоростью передачи данных (Double Data Rate SDRAM). При использовании DDR SDRAM достигается большая полоса пропускания, нежели в обыкновенной SDRAM. Передача данных осуществляется по обоим фронтам синхросигнала, за счёт этого фактически удваивается производительность, не увеличивая при этом частоты шины памяти. Следующие поколения технологии DDR, такие как DDR2, DDR3, позволяют снизить энергопотребление наряду с увеличением рабочих частот, ценой увеличения таймингов. При этом никаких принципиально новых решений они не приносят. В связи с этим, нельзя однозначно показать превосходство чипов па-

мяти DDR3, работающих на высоких частотах — нужно в каждом конкретном случае производить тестовые замеры производительности, учитывая задержки и рабочие частоты.

6. Характеристики подсистем памяти современных серверов

Приведем характеристики подсистем памяти нескольких серверов, находящихся в процессорном полигоне НИВЦ МГУ:

1. Процессор: AMD Opteron 265 1.8 ГГц (два ядра, FSB 1000, 1МБ L2); Материнская плата: Thunder K8SD Pro на чипсете AMD 8131; Память: 2ГБ РС3200;
2. Процессор: AMD Opteron 280 2.4 ГГц (два ядра, FSB 1000, 1МБ L2); Материнская плата: Thunder K8SD Pro на чипсете AMD 8131; Память: 4ГБ РС3200;
3. Процессор: Intel Xeon 5050 3.0 ГГц (два ядра Dempsey, FSB 667, 4МБ L2); Материнская плата: Intel «Alcolu» 5000P; Память: 4ГБ РС2-4200;
4. Процессор: Intel Xeon 5150 2.66 ГГц (два ядра Woodcrest, FSB 1333, 4МБ L2); Материнская плата: Intel «Alcolu» 5000P; Память: 16ГБ РС2-4200;
5. Процессор: Intel Xeon 5355 2.66 ГГц (четыре ядра Clowertown, FSB 1333, 8МБ L2); Материнская плата: Intel «Alcolu» 5000P; Память: 6ГБ РС2-5300;

Также рассмотрим тесты пропускной способности подсистем памяти следующих серверов:

6. Процессор: Intel Xeon 2.0 ГГц (два ядра Woodcrest, FSB 1333, 4МБ L2); Материнская плата: Intel «Alcolu» 5000P; Память: 8ГБ РС2-5300;
7. Процессор: AMD Opteron 244 (FSB 1800, 1МБ L2); Материнская плата: ASUS SK8N на чипсете nForce3; Память: 2ГБ РС2700;

8. Процессор: Intel Pentium 4 3.6 ГГц (ядро Prescott, FSB 800, 2МБ L2); Материнская плата: Gigabyte 8AENXP-D на чипсете Intel 925XE; Память: 1ГБ PC2-4200.

Для тестирования использовалась программа RightMark Memory Analyzer [5], осуществляющая чтение данных в двух вариантах: измерение предельной пропускной способности памяти с использованием оптимизаций чтения типа предвыборки (последний столбец) и измерение средней реальной пропускной способности без оптимизаций (предпоследний столбец). Как видно из сравнительной таблицы, отличия по строкам в пропускной способности памяти при чтении обуславливаются различиями в аппаратных конфигурациях. Но, что более важно, отличия в двух последних столбцах является следствием программного улучшения, которое даёт сравнимый прирост в рамках одной конфигурации!

Список литературы

- [1] Cache mapping and associativity,
<http://www.pcguides.com/ref/mbsys/cache/funcMapping-c.html>.
- [2] Cache write policy,
<http://www.pcguides.com/ref/mbsys/cache/funcWrite-c.html>.
- [3] DDR SDRAM, http://en.wikipedia.org/wiki/DDR_SDRAM.
- [4] Детальное исследование платформ с помощью тестового пакета RightMark Memory Analyzer,
<http://www.ixbt.com/cpu/rmma-k7-k8.shtml>,
<http://www.ixbt.com/cpu/rmma-prescott2.shtml>.
- [5] RightMark Memory Analyzer,
http://cpu.rightmark.org/products/rmma_rus.shtml.