

Кэш

Материал из Википедии — свободной энциклопедии

Кэ́ш^{[1][2][3][4]} или **ке́ш**^{[5][6][7]} (англ. *cache*, от фр. *cacher* — «прятать»; произносится [kæʃ] — «кэш») — промежуточный *буфер* с быстрым доступом к нему, содержащий информацию, которая может быть запрошена с наибольшей вероятностью. Доступ к данным в кэше осуществляется быстрее, чем выборка исходных данных из более медленной памяти или удалённого источника, однако её объём существенно ограничен по сравнению с хранилищем исходных данных.

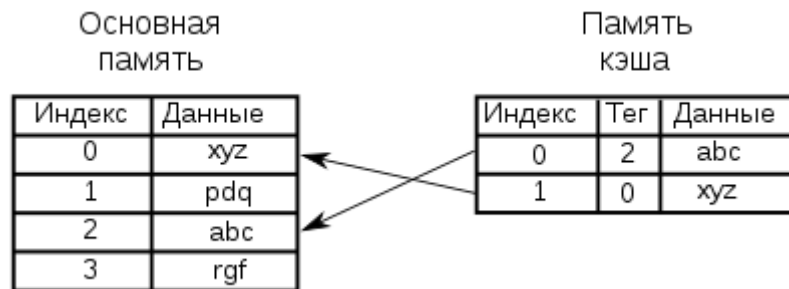
Содержание
История
Функционирование
Аппаратная реализация
Кэш центрального процессора
Уровни кэша процессора
Ассоциативность кэша
Кэширование внешних накопителей
Программная реализация
Политика записи при кэшировании
Алгоритм работы кэша с отложенной записью
Алгоритм вытеснения
Кэширование, выполняемое операционной системой
Кэширование интернет-страниц
Кэширование результатов работы
См. также
Примечания
Литература

История

Впервые слово «cache» в компьютерном контексте было использовано в 1967 году во время подготовки статьи для публикации в журнале «*IBM Systems Journal*». Статья касалась усовершенствования памяти в разрабатываемой модели 85 из серии *IBM System/360*. Редактор журнала Лайл Джонсон попросил придумать более описательный термин, нежели «высокоскоростной буфер», но из-за отсутствия идей сам предложил слово «cache». Статья была опубликована в начале 1968 года, авторы были премированы *IBM*, их работа получила распространение и впоследствии была улучшена, а слово «кэш» вскоре стало использоваться в компьютерной литературе как общепринятый термин^[8].

Функционирование

Кэш — это память с большей скоростью доступа, предназначенная для ускорения обращения к данным, содержащимся постоянно в памяти с меньшей скоростью доступа (далее «основная память»). Кэширование применяется ЦПУ, жёсткими дисками, браузерами, веб-серверами, службами DNS и WINS.



Отображение кэша памяти ЦПУ в основной памяти.

Кэш состоит из набора записей.

Каждая запись ассоциирована с элементом данных или блоком данных (небольшой части данных), которая является копией элемента данных в основной памяти. Каждая запись имеет идентификатор, часто называемый тегом, определяющий соответствие между элементами данных в кэше и их копиями в основной памяти.

Когда клиент кэша (ЦПУ, веб-браузер, операционная система) обращается к данным, прежде всего исследуется кэш. Если в кэше найдена запись с идентификатором, совпадающим с идентификатором затребованного элемента данных, то используются элементы данных в кэше. Такой случай называется *попаданием кэша*. Если в кэше не найдена запись, содержащая затребованный элемент данных, то он читается из основной памяти в кэш, и становится доступным для последующих обращений. Такой случай называется *промахом кэша*. Процент обращений к кэшу, когда в нём найден результат, называется *уровнем попаданий*, или *коэффициентом попаданий* в кэш.

Например, веб-браузер проверяет локальный кэш на диске на наличие локальной копии веб-страницы, соответствующей запрошенному URL. В этом примере URL — это идентификатор, а содержимое веб-страницы — это элементы данных.

Если кэш ограничен в объёме, то при промахе может быть принято решение отбросить некоторую запись для освобождения пространства. Для выбора отбрасываемой записи используются разные алгоритмы вытеснения.

При модификации элементов данных в кэше выполняется их обновление в основной памяти. Задержка во времени между модификацией данных в кэше и обновлением основной памяти управляется так называемой политикой записи.

В кэше с *немедленной записью* каждое изменение вызывает синхронное обновление данных в основной памяти.

В кэше с *отложенной записью* (или *обратной записью*) обновление происходит в случае вытеснения элемента данных, периодически или по запросу клиента. Для отслеживания модифицированных элементов данных записи кэша хранят признак модификации (*изменённый* или «грязный»). Промах в кэше с отложенной записью может потребовать два обращения к основной памяти: первое для записи заменяемых данных из кэша, второе для чтения необходимого элемента данных.

В случае, если данные в основной памяти могут быть изменены независимо от кэша, то запись кэша может стать *неактуальной*. Протоколы взаимодействия между кэшами, которые сохраняют согласованность данных, называют протоколами когерентности кэша.

Аппаратная реализация

Кэш центрального процессора

В связи с ростом частоты, на которой функционируют процессоры, и повышением производительности подсистемы оперативной памяти (ОЗУ), узким местом вычислительной системы стал интерфейс передачи данных.

Кэш-память может давать значительный выигрыш в производительности в случае, когда тактовая частота ОЗУ значительно меньше тактовой частоты процессора. Ряд моделей процессоров обладают собственным кэшем для минимизации времени доступа к оперативной памяти (ОЗУ), которая медленнее, чем регистры (эти регистры и буферы ввода-вывода могут считаться кэшем нулевого уровня). Тактовая частота для кэш-памяти обычно ненамного меньше частоты ЦП.

В процессорах с поддержкой виртуальной адресации часто вводят небольшой быстродействующий буфер трансляций адресов (TLB). Его скорость важна, так как он опрашивается на каждом обращении в память.

Проблема синхронизации между различными кэшами (как одного, так и множества процессоров) решается когерентностью кэша.

Существует три варианта обмена информацией между кэш-памятью различных уровней, или, как говорят, кэш-архитектуры: инклюзивная, эксклюзивная и неэксклюзивная.

Эксклюзивная кэш-память предполагает уникальность информации, находящейся в различных уровнях кэша (предпочитает фирма AMD).

В неэксклюзивной кэши могут вести себя как угодно.

Уровни кэша процессора

Кэш центрального процессора разделён на несколько уровней. Максимальное количество кэшей — четыре. В универсальном процессоре в настоящее время число уровней может достигать трёх. Кэш-память уровня N+1, как правило, больше по размеру и медленнее по скорости доступа и передаче данных, чем кэш-память уровня N.

- Самым быстрым является кэш первого уровня — L1 cache (level 1 cache). По сути, он является неотъемлемой частью процессора, поскольку расположен на одном с ним кристалле и входит в состав функциональных блоков. В современных процессорах обычно L1 разделён на два кэша — кэш команд (инструкций) и кэш данных (Гарвардская архитектура). Большинство процессоров без L1 не могут функционировать. L1 работает на частоте процессора, и, в общем случае, обращение к нему может производиться каждый такт. Зачастую является возможным выполнять несколько операций чтения/записи одновременно.
- Вторым по быстродействию является кэш второго уровня — L2 cache, который обычно, как и L1, расположен на одном кристалле с процессором. В ранних версиях процессоров L2 реализовывался в виде отдельного набора микросхем памяти на материнской плате. Объём L2 от 128 кбайт до 1–12 Мбайт. В современных многоядерных процессорах кэш второго уровня, находясь на том же кристалле, является памятью раздельного пользования — при общем объёме кэша в n Мбайт на каждое ядро приходится по n/c Мбайта, где c — количество ядер процессора.

- Кэш третьего уровня наименее быстродействующий, но он может быть очень большим — более 24 Мбайт. L3 медленнее предыдущих кэшей, но всё равно значительно быстрее, чем оперативная память. В многопроцессорных системах находится в общем пользовании и предназначен для синхронизации данных различных L2.
- Существует четвёртый уровень кэша, применение которого оправдано только для многопроцессорных высокопроизводительных серверов и мейнфреймов. Обычно он реализован отдельной микросхемой.

Ассоциативность кэша

Одна из фундаментальных характеристик кэш-памяти — уровень ассоциативности — отображает её логическую сегментацию, которая вызвана тем, что последовательный перебор всех строк кэша в поисках необходимых данных потребовал бы десятков тактов и свёл бы на нет весь выигрыш от использования встроенной в ЦП памяти. Поэтому ячейки ОЗУ жёстко привязываются к строкам кэш-памяти (в каждой строке могут быть данные из фиксированного набора адресов), что значительно сокращает время поиска.

При одинаковом объёме кэша схема с большей ассоциативностью будет наименее быстрой, но наиболее эффективной (после четырёхпоточковой реализации прирост «удельной эффективности» на один поток растёт мало).

Кэширование внешних накопителей

Многие периферийные устройства хранения данных используют внутренний кэш для ускорения работы, в частности, жёсткие диски используют кэш-память от 1 до 256 Мбайт (модели с поддержкой NCQ/TCQ используют её для хранения и обработки запросов), устройства чтения CD/DVD/BD-дисков также кэшируют прочитанную информацию для ускорения повторного обращения.

Операционная система также использует часть оперативной памяти в качестве кэша дисковых операций (например, для внешних устройств, не обладающих собственной кэш-памятью, в том числе жёстких дисков, flash-памяти и гибких дисков). Часто для кэширования жёстких дисков предоставляется вся свободная (не выделенная процессам) оперативная память.

Применение кэширования внешних накопителей обусловлено следующими факторами:

1. скорость доступа процессора к оперативной памяти в сотни и более раз больше, чем к памяти внешних накопителей;
2. производительность дисковых устройств хранения (жёсткие, гибкие, оптические диски) максимальна при чтении-записи нескольких последовательно расположенных блоков и значительно уменьшается при одиночных запросах в разные места диска, что связано с инерцией механического привода головки.
3. крайне неравномерная частота обращения к различным блокам памяти внешних накопителей:
 1. использование части блоков несколькими процессами одновременно, по чтению и записи (например, в базах данных)
 2. очень частое чтение части блоков (индексные файлы, каталоги в файловой системе)
 3. очень частая запись части блоков (файлы логов, журналов, баз данных; метаданные файловой системы).

При чтении кэш позволяет прочитать блок один раз, затем хранить одну копию блока в оперативной памяти для всех процессов и выдавать содержимое блока «мгновенно» (по сравнению с запросом к диску). Существует техника «предзапроса» — в фоновом режиме операционной системой считываются в кэш также несколько следующих блоков (после нужного).

При записи кэш позволяет сгруппировать короткие записи в более крупные, которые эффективнее обрабатываются накопителями, либо избежать записи промежуточных модификаций. При этом все промежуточные состояния блока видны процессам из оперативной памяти.

Кэширование внешних устройств хранения значительно увеличивает производительность системы за счёт оптимизации использования ввода-вывода. Преимуществом технологии является прозрачная (незаметная для программ) автоматическая оптимизация использования памяти-дисков при неизменности логики приложений, работающих с файлами.

Недостатком кэширования записи является промежуток времени между запросом на запись от программы и фактической записью блока на диск, а также изменение порядка выполнения записей, что может приводить к потерям информации или несогласованности структур при сбое питания или зависании системы. Данная проблема сглаживается принудительной периодической синхронизацией (записью изменённых строк кэша) и журналированием файловых систем.

Программная реализация

Политика записи при кэшировании

При чтении данных кэш-память даёт однозначный выигрыш в производительности. При записи данных выигрыш можно получить только ценой снижения надёжности. Поэтому в различных приложениях может быть выбрана та или иная политика записи кэш-памяти.

Существуют две основные политики записи кэш-памяти — сквозная запись (write-through) и отложенная запись (write-back):

1. Сквозная запись — запись производится непосредственно в основную память (и дублируется в кэш), то есть запись не кэшируется.
2. Отложенная запись — запись данных производится в кэш. Запись же в основную память производится позже (при вытеснении или по истечении времени), группируя в одной операции несколько операций записи в соседние ячейки. Технология обратной записи на некоторое время делает данные в основной памяти неактуальными, для самого ЦП эти неактуальности не заметны, но перед обращением к памяти другого ведущего системной шины (контроллера DMA, bus-master-устройства шины PCI) кэш должен быть записан в память принудительно. При использовании обратной записи в многопроцессорной системе кэши различных ЦП должны быть согласованы (или процессоры должны использовать одну кэш-память).

Алгоритм работы кэша с отложенной записью

Изначально все заголовки буферов помещаются в список свободных буферов. Если процесс намеревается прочитать или модифицировать блок, то он выполняет следующий алгоритм:

1. пытается найти в хеш-таблице заголовок буфера с заданным номером;
2. в случае, если полученный буфер занят, ждёт его освобождения;

3. в случае, если буфер не найден в хеш-таблице, берёт первый буфер из хвоста списка свободных;
4. в случае, если список свободных буферов пуст, то выполняется алгоритм вытеснения (см. ниже);
5. в случае, если полученный буфер помечен как «грязный», выполняет асинхронную запись содержимого буфера во внешнюю память.
6. удаляет буфер из хеш-таблицы, если он был помещён в неё;
7. помещает буфер в хеш-таблицу с новым номером.

Процесс читает данные в полученный буфер и освобождает его. В случае модификации процесс перед освобождением помечает буфер как «грязный». При освобождении буфер помещается в голову списка свободных буферов.

Таким образом:

1. если процесс прочитал некоторый блок в буфер, то велика вероятность, что другой процесс при чтении этого блока найдёт буфер в оперативной памяти;
2. запись данных во внешнюю память выполняется только тогда, когда не хватает «чистых» буферов, либо по запросу.

Алгоритм вытеснения

Если список свободных буферов пуст, то выполняется алгоритм вытеснения буфера. Алгоритм вытеснения существенно влияет на производительность кэша. Существуют следующие алгоритмы:

1. Реализуемые с помощью таймера:

1. LRU (англ. *Least Recently Used*) — вытесняется буфер, неиспользованный дольше всех;
2. MRU (англ. *Most Recently Used*) — вытесняется последний использованный буфер;

2. Реализуемые с помощью счётчика:

1. LFU (англ. *Least Frequently Used*) — вытесняется буфер, использованный реже всех;
2. ARC (англ. *Adaptive Replacement Cache*) — алгоритм вытеснения, комбинирующий LRU и LFU, запатентованный IBM.

Применение того или иного алгоритма зависит от стратегии кэширования данных. LRU наиболее эффективен, если данные гарантированно будут повторно использованы в ближайшее время. MRU наиболее эффективен, если данные гарантированно не будут повторно использованы в ближайшее время. В случае, если приложение явно указывает стратегию кэширования для некоторого набора данных, то кэш будет функционировать наиболее эффективно.

Кэширование, выполняемое операционной системой

Кэш оперативной памяти состоит из следующих элементов:

1. набор страниц оперативной памяти, разделённых на буферы, равные по длине блоку данных соответствующего устройства внешней памяти;
2. набор заголовков буферов, описывающих состояние соответствующего буфера;
3. хеш-таблицы, содержащей соответствие номера блока заголовку;

4. списки свободных буферов.

Кэширование интернет-страниц

В процессе передачи информации по сети может использоваться кэширование интернет-страниц — процесс сохранения часто запрашиваемых документов на (промежуточных) прокси-серверах или машине пользователя, с целью предотвращения их постоянной загрузки с сервера-источника и уменьшения трафика. Таким образом, информация перемещается ближе к пользователю. Управление кэшированием осуществляется при помощи HTTP-заголовков.

Как вариант, кэширование веб-страниц может осуществляться с помощью CMS конкретного сайта для снижения нагрузки на сервер при большой посещаемости. Кэширование может производиться как в память, так и в файловый кэш^[9]. Недостаток кэширования заключается в том, что изменения, внесённые на одном браузере, могут не сразу отражаться в другом браузере, в котором данные берутся из кэш-памяти.

Кэширование результатов работы

Многие программы записывают куда-либо промежуточные или вспомогательные результаты работы, чтобы не вычислять их каждый раз, когда они понадобятся. Это ускоряет работу, но требует дополнительной памяти (оперативной или дисковой). Примером такого кэширования является индексирование баз данных.

См. также

- Алгоритмы кэширования
- Когерентность кэша
- Кэш процессора

Примечания

1. Кэш // Большой орфографический словарь русского языка / под ред. С. Г. Бархударова, И. Ф. Протченко и Л. И. Скворцова. — 3-е изд. — М.: ОНИКС Мир и Образование, 2007. — С. 399. — ISBN 978-5-488-00924-0. — ISBN 978-5-94666-375-5.
2. Большой толковый словарь русского языка / Автор, сост. и гл. ред. С. А. Кузнецов. Институт лингвистических исследований РАН, 2000
3. Захаренко Е. Н., Комарова Л. Н., Нечаева И. В. Новый словарь иностранных слов. М.: 2003
4. Толковый словарь по вычислительной технике. Microsoft Press, из-во «Русская Редакция», 1995
5. Русский орфографический словарь: около 180 000 слов [Электронная версия] (<http://www.slovari.ru/default.aspx?p=242>) / О. Е. Иванова, В. В. Лопатин (отв. ред.), И. В. Нечаева, Л. К. Чельцова. — 2-е изд., испр. и доп. — М.: Российская академия наук. Институт русского языка имени В. В. Виноградова, 2004. — 960 с. — ISBN 5-88744-052-X.
6. Першиков В. И., Савинков В. М. Толковый словарь по информатике / Рецензенты: канд. физ.-мат. наук А. С. Марков и д-р физ.-мат. наук И. В. Поттосин. — М.: Финансы и статистика, 1991. — 543 с. — 50 000 экз. — ISBN 5-279-00367-0.
7. Борковский А. Б. Англо-русский словарь по программированию и информатике (с толкованиями). — М.: Русский язык, 1990. — 335 с. — 50 050 (доп.) экз. — ISBN 5-200-

01169-3.

8. G. C. Stierhoff, A. G. Davis. A History of the IBM Systems Journal // IEEE Annals of the History of Computing. — январь 1998. — Т. 20, № 1. — С. 29—35. — doi:10.1109/85.646206 (<https://doi.org/10.1109/85.646206>).
9. Распределенные ОС (<http://parallel.ru/krukov/lec5.html>)

Литература

- Бах М. Дж. Архитектура операционной системы UNIX
-

Источник — <https://ru.wikipedia.org/w/index.php?title=Кэш&oldid=113744895>

Эта страница в последний раз была отредактирована 21 апреля 2021 в 16:17.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.