Оценки вероятности промаха при случайном тестировании кэша

А. С. Куцаев[1]

[1]ФГУ ФНЦ НИИСИ РАН, Москва, Россия, koutsaev@niisi.msk.ru

Аннотация. Вероятность промаха при случайном тестировании системы кэшей зависит в основном от распределения памяти. Оно позволяет ограничить число активных строк, создавая тем самым нагрузку на кэш. Перебор областей памяти в генераторе тестов позволяет сократить вспомогательные действия в тесте, не связанные непосредственно с тестированием. Оценки вероятности промаха в кэшах первого и второго уровня при различных условиях позволяют выбрать параметры для генерации эффективных тестов.

Ключевые слова: случайные тесты, вероятность промаха в кэше, перебор областей памяти

1. Введение

Тестирование системы кэшей и TLB с помощью генератора случайных тестов [1, 2] требует подбора управляющих параметров с учетом задачи, особенно в части случайного выбора адресов памяти. Случайный выбор адресов без дополнительных условий обычно приводит к неэффективным тестам. Причина в том, что размер кэшей достаточно велик, и случайные обращения к памяти вызывают в кэше в основном промахи без замещения данных.

Для тестирования интересен режим частых попаданий, когда данные целиком помещаются в кэш, и обмен с памятью происходит наиболее быстро. Также полезен режим частых промахов, создающий большую нагрузку на кэш. В случайном тесте эти режимы могут осложняться добавлением переходов, циклов и исключительных ситуаций. Переходный режим случайного заполнения кэша менее интересен, так как не связан с определенным видом нагрузки на кэш. В генераторе тестов есть средство быстрого случайного заполнения кэша.

Для создания эффективных тестов кэшей наиболее важен выбор распределения памяти. Он позволяет реализовать т.н. режим "горячих строк", в котором нагрузка приходится на небольшое подмножество строк кэша. Далее для получения "горячих строк" отводимая память состоит из небольших одинаковых областей, адреса начала которых выровнены по размеру множества кэша.

Режим частых попаданий проще, так как требует только, чтобы число областей памяти N не превышало числа множеств кэша A. Для режима частых промахов условия $N > A$ мало. При случайном выборе областей памяти вероятность промаха равна $1 - A/N$, так что число областей должно составлять несколько десятков. При тестировании кэша 1-го уровня области можно расположить так, что регистры базы адреса не будут требовать частой перезагрузки. Но для кэша 2-го уровня на каждую область требуется отдельный регистр, так что перезагрузки неизбежны. Еще больше областей памяти может потребоваться при тестировании TLB.

Генератор тестов предлагает режим перебора областей памяти, в котором случайный выбор адресов происходит определенное число раз в одной области или группе областей. После этого происходит переход к следующей области или их группе из списка. Список допустимых областей один и тот же для операций загрузки и сохранения. Режим перебора позволяет перезагружать регистры адреса группами по мере необходимости. В результате дополнительные действия, не связанные с целью тестирования, отделены от основных действий.

В главе 2 оценивается вероятность промаха в кэше первого уровня, если обращения к памяти ограничены ее чтением. В главе 3 делается то же самое, если чтение и запись равновероятны. В главе 4 оценивается промаха в кэше второго уровня, также в случае чтения и записи. В главе 5 приведены выводы.

2. Кэш L1, только чтение

Рассмотрим промахи в кэше первого уровня (L1) при переборе областей памяти, если выполняется только чтение данных. Схема оценки с некоторыми изменениями применима и для чтения-записи. Пусть память данных состоит из N одинаковых областей, адреса начала которых выровнены по размеру множества кэша, т.е. каждая область связана с одними и теми же строками кэша. Примем для простоты, что перебор областей происходит по одной. В цикле перебора N шагов, число циклов не ограничено, на каждом шаге выполняется ровно K обращений к памяти, число активных строк кэша L.

Далее используется определенная система

кэшей. В ней кэш L1 со сквозной записью, кэш 2-го уровня (L2) с обратной записью. В вычислительных примерах в кэше L1 4 множества (ways), число строк в множестве 256, длина строки 16 байтов. В кэше L2 4 множества, число строк в множестве 4096, длина строки 32 байта. Строка L2 здесь отвечает двум строкам L1, что может быть полезно в некоторых тестовых ситуациях.

Когда кэш заполнен, каждое чтение памяти приводит к попаданию в кэше либо к промаху с замещением данных, которые не использовались дольше всего. Тег кэша определен областью памяти, т.е. номером шага в цикле перебора, строка кэша определяется младшими битами случайного адреса.

Вероятность того, что случайный адрес будет выбран в определенной строке S кэша, равна $1/L$. При K выборах адреса на шаге вероятность того, что строка S будет выбрана на шаге хотя бы один раз, равна

$$P_1(L,K) = 1 - (1 - 1/L)^K \qquad (2.1)$$

Это дополнение до 1 вероятности, что на этом шаге строка S не будет выбрана ни разу. При чтении памяти тег адреса и данные копируются в кэш на этом шаге, если они не скопированы ранее. Повторный выбор строки S на этом же шаге приводит к попаданию в кэш и не влияет на наличие и очередность тегов в строке.

При длительном выполнении теста можно выбрать область памяти и принять, что цикл перебора начинается с нее. Обозначим T_1 тег адреса, отвечающий области на первом шаге цикла. Оценим вероятность промаха в кэше на первом шаге цикла. Пусть строка S выбрана на первом шаге цикла с номером C+1 (при нескольких выборах строки S на шаге берется первый). Попадание в кэше происходит, если тег T_1 уже есть в строке S, т. е. если в предыдущий раз тег T_1 был выбран в строке S на первом шаге цикла с номером B и с тех пор не был замещен в циклах B, B+1, ... C. Для этого нужно, чтобы в циклах с B по C замещение тега (любого) в строке S произошло менее A раз.

Поскольку кэш заполнен, в строке S все теги действительны. Замещение тега T_1 в ней не может происходить на первом шаге циклов с B по C по условию. Кроме того, замещение тегов в строке S возможно не на всех шагах с 2 по N. Если в начале цикла строка S содержит тег T_X, и этот тег не замещен до шага X, на котором выбирается данный тег, то на шаге X будет попадание в кэше, а не замещение. В каждом цикле число шагов, на которых возможно замещение тега, может принимать значения от N-A до N-1. Для оценок вероятности замещения снизу и сверху нужно использовать число шагов на цикл N-A и N-1, соответственно.

Вероятность выбора строки S на шаге перебора хотя бы один раз равна $P_1(L, K)$ (2.1). Вероятность того, что на m шагах перебора замещение тега произойдет ровно i раз, обозначим G_i. Вероятности G_i. (i = 0, 1,...m) являются членами разложения по степеням бинома $1 = ((1 - P_1) + P_1)^m$.

$$G_i = C_m^i (1 - P_1)^{m-i} P_1^i \qquad (2.2)$$

где C_m^i - биномиальные коэффициенты. Вероятность того, что последний выбранный тег в строке S не будет замещен на m шагах, обозначим $P_T(m)$. Она равна сумме первых A членов разложения бинома. Эта вероятность тем меньше, чем больше шагов участвует в замещении.

$$P_T(m) = \sum_{i=0}^{A-1} G_i \qquad (2.3)$$

До начала цикла C+1 предыдущий выбор тега T_1 в строке S может быть в цикле C, C-1, C-2,... 1, либо тег не выбран ни разу. Это полный набор событий с вероятностями P_1, $P_1(1\text{-}P_1)$, $P_1(1\text{-}P_1)^2$ и т.д.; вероятность невыбора равна $(1\text{-}P_1)^C$. Вероятность того, что предыдущий выбор тега T_1 был в цикле B и тег не замещен до начала цикла C+1 на m шагах, равна произведению $P_1(1\text{-}P_1)^{C\text{-}B}$ на $P_T(m)$. Суммируя для всех циклов от 1 до C и заменяя C-B+1 на i, получим оценки для вероятности незамещения тега T_1:

$$P_{SC}(N) = \sum_{i=1}^{C-B+1} P_1(1 - P_1)^{i-1} P_T(iM) \qquad (2.4)$$

где M=N-A для оценки сверху и M=N-1 для оценки снизу, согласно замечанию при выводе (2.3). С ростом C-B эти суммы сходятся как степенной ряд или быстрее, так что для анализа можно брать их предел $P_S(N)$. Его дополнение до 1 дает оценки для вероятности первого промаха в строке S на первом шаге цикла C+1, когда C достаточно велико.

$$P_F(N) = 1 - \sum_{i=1}^{\infty} P_1(1 - P_1)^{i-1} P_T(iM) \qquad (2.5)$$

Для оценки $P_F(N)$ снизу нужно брать в (2.5) M=N-A.

Можно сравнить $P_F(N)$ с вероятностью промаха при случайном выборе областей. Примем число обращений к памяти на шаге K=1, чтобы исключить влияние повторных выборов (оно рассмотрено далее). На Рис. 1 показаны результаты для случайного выбора областей (1-A/N) и оценки снизу для перебора при L=16 и L=32 в зависимости от N. Здесь видно, что при N>16 вероятность промаха при переборе областей выше, чем при их случайном выборе. Также можно заметить, что при N>16 вероятность промаха падает с ростом числа строк, поскольку замещение тегов на меньшем числе строк происходит более

интенсивно.

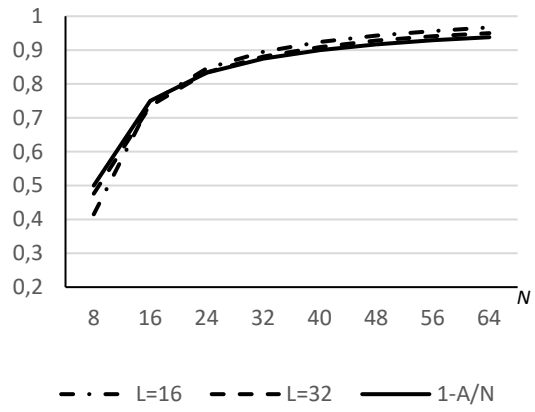

Рис. 1. Вероятности промаха в кэше L1 при *K*=1, только чтение.

При втором и следующих выборах адреса на шаге перебора вероятность промаха зависит от возможности повторного выбора строки. При первом выборе строки на шаге вероятность промаха для нее одна и та же, $P_F(N)$. При повторном выборе будет попадание в кэше.

Оценить вероятности двух и более промахов на шаге можно с помощью рекуррентных соотношений. Очередная новая строка выбирается на шаге с вероятностью 1 - $h{\cdot}p$, где $p = 1/L$, а h - число уже выбранных на шаге новых строк. Обозначим вероятность выбора h новых строк при K выборах через $P_{K-1}^{h}(p)$; это полином степени K-1 от p. Для K=2 имеем $P_1^1(p) = p,\ P_1^2(p) = 1 - p$.

Используя вероятность выбора очередной новой строки 1-$h{\cdot}p$, получим для K выборов:

$$P_K^{K+1}(p) = P_{K-1}^{K}(p) \cdot (1 - Kp)$$

$$...P_K^h(p) = P_{K-1}^{h-1}(p) \cdot (1 - (h-1)p) + + P_{K-1}^{h}(p) \cdot hp, \qquad h = 2, \dots, K$$

$$P_K^1(p) = P_{K-1}^1(p) \cdot p$$

Среднее число новых строк, нормированное на K, дается выражением:

$$M_K(p) = \frac{1}{K}\sum_{h=1}^{K+1} i \cdot P_K^h(p) \qquad (2.6)$$

Так как в случаях, представляющих интерес, значение $p = 1/L$ невелико, для вычисления $M_K(p)$ в них достаточно нулевой и первой степени p:

$$M_K(p) = 1 - \frac{K}{2}p + \cdots \qquad (2.7)$$

Так, для L=16 и L=32 зависимость $M_K(p)$ от K выглядит практически линейной. Коэффициент при p^2 зависит от K более сложно. Среднее число промахов на шаге перебора дается выражением $K \cdot M_K(p) \cdot P_F(N)$. В принятых условиях увеличение K до 6 - 8 не приводит к заметному снижению эффективности, хотя и не дает роста вероятности промаха.

3. Кэш L1, чтение и запись

Оценки для чтения и записи памяти сложнее, так как обе операции влияют на порядок тегов в строке, но только чтение обеспечивает наличие тега и приводит к замещению тега. Для оценки вероятности замещения вместо $P_1(L,K)$ нужно использовать $P_2(L,K)$. Это вероятность того, что на шаге перебора будет хотя бы одно чтение памяти. Если чтение и запись памяти происходят равновероятно, выражение для $P_2(L,K)$ аналогично (2.1):

$$P_2(L,K) = 1 - (1 - 1/2L)^K \qquad (3.1)$$

Вероятность незамещения тега T_1 на m шагах перебора $P_{TW}(m)$ принимает вид

$$P_{TW}(m) = \sum_{i=0}^{A-1} G_i, \qquad (3.2)$$

где $G_i = C_K^i (1 - P_2)^{K-i} {P_2}^i$. При большом числе циклов вероятность незамещения тега T_1 к началу любого цикла будет близка к предельной, обозначаемой $P_{SW}(N)$. Она находится аналогично вероятности для только чтения $P_S(N)$, с заменой $P_T(m)$ на $P_{TW}(m)$, следующим образом.

Вероятность незамещения тега T_1 до начала цикла C+1 складывается, как и выше (2.4), из вероятностей предыдущего выбора тега T_1 на циклах C, C-1,..., B , умноженных на вероятности незамещения до цикла C+1. Добавляется учет того, что перед записью в память тег T_1 в строке S необходим. Составляющие $P_{TW}(m)$ можно свести в таблицу:

Номер цикла	Вероятность выбора строки	Вероятность незамещения
C	P1	PTW(M)
C-1	(1-P1)P1	PTW(2M)
C-2	(1-P1)2P1	PTW(3M)
...	...	...
B	(1-P1)C-BP1	PTW((C-B+1)•M)

В отличие от случая только чтения появляется еще один сомножитель: вероятность чтения или, при наличии тега T1, записи, при условии выбора строки S. Она одинакова для всех циклов и равна . P_2/P_1 + (1 - P_2/P_1)·$P_{SW}(N)$. Здесь условная вероятность хотя бы одного чтения P_2 делится на вероятность условия, что строка выбрана, т.е. P_1. Вероятность отсутствия чтения (т.е. записи) умножается на вероятность наличия тега T_1 в строке перед началом цикла. Она уже представляет собой предел при C-$B \to \infty$. Пере-

множая и суммируя вероятности, получим в пределе равенство с $P_{SW}(N)$ в левой и правой части, из которого можно выразить искомое:

$$P_{SW}(N) = \frac{P_2 Z}{1-(P_1-P_2)Z}, \qquad (3.3)$$

где $Z = \sum_{i=1}^{\infty}(1-P_1)^{i-1}P_{TW}(iM)$

Вероятность первого промаха при чтении-записи $P_{FW}(N) = 1 - P_{SW}(N)$. Оценки снизу вероятности первого промаха при чтении и записи показаны на Рис. 2. По сравнения со случаем только чтения эти оценки ниже. Причина в том, что при чтении и записи замещение тегов происходит примерно вдвое реже.

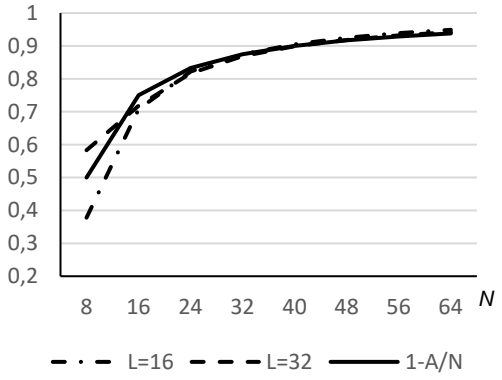

Рис. 2. Вероятности промаха в кэше L1 при K=1, чтение и запись.

При двух и более выборах адреса на шаге перебора вероятность промаха зависит, как и выше, от возможности повторного выбора строки на шаге. Отличие в том, что после промаха при записи повторный выбор строки также приводит к промаху. Здесь вероятность промаха определяет не число строк, выбранных заново, а число строк, выбранных для чтения. Если на шаге уже выбрано для чтения H строк, то вероятность выбора такой строки равна $H{\cdot}p$ (где p=1/L), при этом происходит попадание в кэш. Вероятность выбора строки, где чтения не было, равна 1-$H{\cdot}p$. При этом рост числа промахов M происходит при чтении и записи, а рост H только при чтении. Эти правила можно представить в виде аналога разностной схемы, где по вероятностям промаха $P_K^{H,M}$ для индексов H, M на уровне K вычисляются вероятности $P_{K+1}^{H,M}$ на уровне K+1. Исходно $P_1^{0,1} = P_1^{1,1} = 1/2$, остальные нули. Предварительно $P_{K+1}^{H,M}$ обнуляется, затем для всех индексов:

$$P_{K+1}^{H,M} = P_{K+1}^{H,M} + P_K^{H,M}\, pH$$

$$P_{K+1}^{H,M+1} = P_{K+1}^{H,M+1} + P_K^{H,M}\,(1-pH)/2$$

$$P_{K+1}^{H+1,M+1} = P_{K+1}^{H+1,M+1} + P_K^{H,M}\,(1-pH)/2$$

После получения данных на уровне K, чтобы получить P_K^M, нужно для каждого M просуммировать вероятности $P_K^{H,M}$, по H от 0 до K. Среднее число новых строк, нормированное на K, дается выражением (2.6).

4. Промахи в кэше L2

Для системы кэшей 1-го и 2-го уровня можно использовать набор областей памяти, аналогичный предыдущему. Выравнивание областей памяти должно отвечать размеру множества кэша L2, который значительно больше, чем в L1. Другие возможности организации памяти будут рассмотрены далее. Будем рассматривать чтение и запись памяти с равными вероятностями.

Чтобы в кэше L2 произошел промах, необходимо отсутствие в кэше тега и копии данных для адреса памяти. При записи в память этого достаточно, но при чтении необходим также промах в кэше L1, иначе обращения к L2 не будет. Зависимость условия в L1 и L2 для случая чтения выглядит следующим образом.

Пусть строке S2 кэша L2 соответствует пара строк SA и SB кэша L1, и пусть в строке SA происходит промах при чтении по адресу с тегом T1, т.е. на первом шаге цикла. Перед этим происходит следующая последовательность событий.

- Последнее перед промахом копирование данных по адресу с тегом T1 в строку SA при чтении либо обновление в строке SA при записи в память;
- Одновременно с этим в строке S2 при чтении и записи происходит копирование или обновление соответствующего тега T2 и данных;
- Замещение тега T1 в строке SA в одном или нескольких циклах перебора;
- Замещение тега T2 и данных в строке S2, не позже, чем в L1 (см. ниже).

Теги T_1 и T_2 отвечают адресам одной и той же области памяти, т.е. действуют на одном и том же шаге перебора. Случаев замещения в строке S_2 примерно вчетверо больше, чем в строке S_A, за счет (a) операций записи в память в строке S_A и (b) операций чтения и записи в строке S_B. Таким образом, после замещения в строке S_2 тег T_2 может быть снова прочитан, если позволяют условия (два и более цикла перебора для замещения).

Тогда при промахе в строке S_A возможно попадание в строке S_2. Этот случай маловероятен; в основном промаху в строке S_A отвечает замещение тега T_2 в строке S_2, т.е. промах в L2.

Если же в строке S_A происходит попадание, то тег T_1 и данные не были замещены со времени предыдущего чтения/обновления (как минимум за один цикл перебора). Но так как в кэше L2 случаев замещения в строке S_2 вчетверо больше, то возможно как наличие тега T_2, так и его замещение.

Таким образом, при чтении памяти промах в кэше L1 имеет некоторую положительную корреляцию с замещением соответствующего тега в кэше L2. В случае полной зависимости этих событий вероятность промаха в L2 равна вероятности первого промаха в L1, $P_{FW}(N)$. В случае полной независимости $P_{FW}(N)$ нужно умножить на вероятность замещения соответствующего тега в L2. Между этими крайними случаями данное произведение вероятностей можно рассматривать как оценку снизу для вероятности промаха в L2 при чтении памяти. При записи в память состояние L1 не влияет на вероятность промаха в L2.

Вероятность незамещения соответствующего тега в L2, $P_{S2}(N)$, находится по тем же правилам, что $P_S(N)$ в L1 в случае только чтения (2.4), так как в L2 все операции отражаются в кэше. Число строк L при этом нужно уменьшить вдвое по сравнению с L1.

Оценка снизу вероятности промаха в кэше L2 в начале шага перебора

$$P_{F2(}N) = (1 - P_{S2}(N))\cdot(1 + P_{FW}(N)) /2 \quad (4.1)$$

Зависимость вероятности промаха от N позволяет выбрать приемлемое число областей, на Рис. 2 видно, что это N=32 и выше. Зависимость вероятности промаха от числа строк L кэша позволяет выбрать приемлемый размер области. На Рис. 3 показаны зависимости вероятности промаха $P_{FW}(L)$ для кэша L1 и $P_{F2}(L)$ для кэша L2 при числе областей N=32 и N=64 (число строк везде отвечает размеру строки L1). Здесь видно, что вероятность промаха мало меняется, за исключением очень небольших областей (L=16, размер 128 байтов).

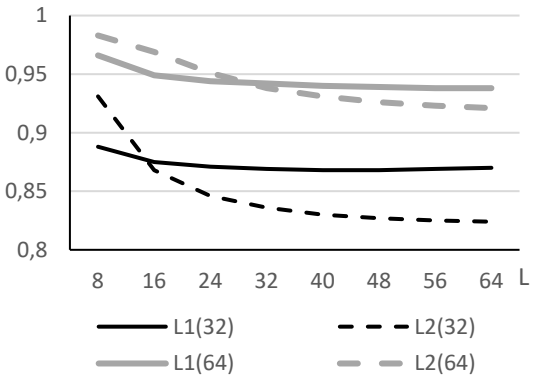

Рис. 3. Вероятности промаха в кэшах L1 и L2 при N=32 и N=64.

Можно также рассмотреть организацию памяти, в которой адреса начала соседних областей вдвое меньше размера множества кэша в L2. Тогда в кэше L1 все остается по-прежнему, а в кэше L2 будет два отрезка активных строк, для областей с четными и нечетными номерами, соответственно. Работа с кэшем L2 для этих групп областей происходит независимо, поэтому в расчетах вероятности промаха нужно брать число областей в шаблоне N, деленное на 2. Уменьшая дальше смещение областей, можно в итоге получить ситуацию, в которой память целиком помещается в кэш L2, и промахов в нем не будет. Выбирая смещение, можно регулировать вероятность промахов в L2, тогда как в L1 она не будет меняться.

5. Заключение

Использование перебора областей памяти при случайном тестировании позволяет использовать большое количество областей памяти без снижения эффективности при тестировании кэшей и TLB. Разработанные способы оценки вероятности промаха сверху и снизу для кэшей первого и второго уровня при переборе областей позволяют выбрать параметры для генерации эффективных случайных тестов.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН «Проведение фундаментальных научных исследований (47 ГП)» по теме № FNEF-2022-0004 «Разработка архитектуры, системных решений и методов для создания микропроцессорных ядер и коммуникационных средств семейства систем на кристалле двойного назначения. 0580-2022-0004», Рег № 122041100063-6.

Miss Probability Estimates for Random Testing of the Cache

A. S. Koutsaev

Abstract. The probability of a miss in a random test of the cache system depends mainly on memory layout. It makes it possible to limit the number of active cache lines, thereby forcing the load on the cache. Enumerating memory areas in the test generator allows one to reduce auxiliary actions in the test that are not directly related to the target of testing. Estimates of the probability of a miss in caches of the first and second levels under various conditions serve to select parameters for generating effective tests.

Keywords: random tests, probability of cache miss, enumeration of memory areas.

Литература

1. А.С. Куцаев. Развитие генератора случайных тестов tergen. Труды НИИСИ РАН 2018, Том 8, № 1, 19-26.

2. А.С. Куцаев, Случайные тесты с перебором классов инструкций. Труды НИИСИ РАН 2022, Том 12, № 1-2, С. 32-37.