УДК 004.021

В. С. Дужин, Г. С. Евсеев, Е. М. Линский

# ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМА УПРАВЛЕНИЯ ОБЪЕМОМ РАЗДЕЛОВ КЭША СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ

Рассматривается модель кэша системы хранения данных, состоящего из двух разделов. Построен алгоритм управления объемом разделов. Исследовано влияние ошибок определения текущего и последующего состояния потока на качество работы алгоритма.

***Ключевые слова:*** *системы хранения данных, кэш, оценка эффективности.*

**Введение.** Использование кэша в системе хранения данных позволяет существенно уменьшить время доступа к информации. К такой системе обращаются приложения, отправляя запросы на получение данных. Каждому приложению в кэше выделен отдельный раздел. Будем считать, что каждое приложение характеризуется требованием на время обработки запроса, выражающимся в среднем значении HR (Hit Rate, отношение числа запросов, найденных в кэше, к общему числу запросов за определенный временной интервал для его раздела).

В настоящей статье рассматривается алгоритм, динамически перераспределяющий границы раздела.

В отличие от работ [1, 2], в которых рассматриваются эвристические алгоритмы управления кэшем, в настоящей статье предложена модель кэша, состоящего из двух разделов, позволяющая проводить анализ алгоритмов управления.

**Модель системы.** Рассмотрим LRU-кэш, с которым работают два приложения, каждому из них ставится в соответствие свой раздел в кэше, свой поток входных запросов и требуемое значение HR.

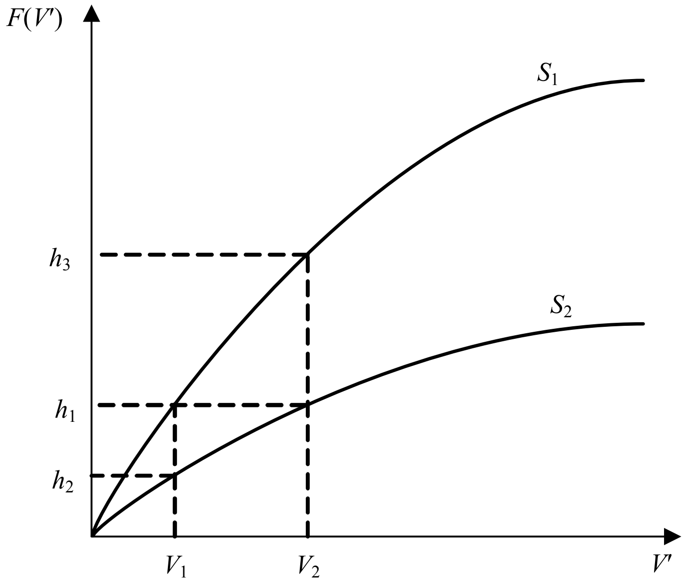
Входные потоки характеризуются интенсивностью запросов и распределением стекового расстояния, под которым будем понимать количество различных адресов между двумя обращениями к одному и тому же адресу [3—4].

Пусть входные потоки к каждому разделу могут находиться в двух состояниях: $S_1$ и $S_2$. Состояния различаются видом гистограммы стековых расстояний. Считается, что любой раздел, входной поток к которому находится в состоянии $S_1$, удовлетворяет требованию по HR, если его размер (объем) равен $V_1$. Раздел, входной поток к которому находится в состоянии $S_2$, удовлетворяет требованию, если его объем равен $V_2$. Будем считать, что $V_2 = 2V_1$, $V_{\text{cache}} = 3V_1$, где $V_{\text{cache}}$ — суммарный объем кэша.

Рассмотрим дискретную временную модель: пусть $N_1 = N_2 = N$ — интенсивность запросов к разделам в определенный интервал времени, а $\xi_1$, $\xi_2$ — число запросов к адресам, находящимся в кэше (за то же время). Вероятности переходов между состояниями *i*-го потока описываются матрицей переходов марковской цепи:

$$\begin{bmatrix} P_i(S_1 \mid S_1) & P_i(S_2 \mid S_1) \\ P_i(S_1 \mid S_2) & P_i(S_2 \mid S_2) \end{bmatrix}. \tag{1}$$

На рисунке приведены функции распределения вероятностей стековых расстояний для потоков в состояниях $S_1$ и $S_2$.




Заметим, что при заданном размере раздела $V'$ величина $F(V')$ может рассматриваться как средний HR на данном разделе, т.к. $F(V')$ – вероятность того, что стековое расстояние не превысит $V'$ (запрашиваемый адрес будет найден в кэше).

Будем считать, что требование к размеру раздела состоит в том, чтобы средний HR был не меньше $h_1$. Как видно из рисунка, это условие не выполняется, когда входной поток находится в состоянии $S_2$, а объем раздела равен $V_1$.

**Алгоритм управления кэшем.** Рассмотрим алгоритм управления размерами разделов в кэше, обеспечивающий улучшение качества работы системы за счет ее динамической адаптации к изменениям входного потока. В течение каждого временного интервала собирается статистика по входным потокам, на основании которой оценивается состояние потоков в данном интервале. Прогноз состояния входных потоков в последующем временном интервале осуществляется на основе этих оценок и матриц (1). В конце каждого интервала происходит перераспределение объемов разделов в соответствии с этим прогнозом. Например, если состояние потока в предыдущем интервале оценено как $\hat{S}_1$, а $P(S_2 \mid S_1) > 0{,}5$, то прогнозируется, что в следующем интервале поток перейдет в состояние $S_2$.

Ниже представлен способ получения оценки состояния входного потока. Пусть текущий объем раздела равен $V_1$, за последний интервал пришло $N$ запросов, при этом $\xi$ из них были найдены в кэше. В этом случае вероятность того, что входной поток к данному разделу находился в состоянии $S_1$, определяется следующим образом:

$$P(S_1 \mid V_1, N, \xi) = \frac{P(S_1, \xi \mid V_1, N)}{P(\xi \mid V_1, N)}. \tag{2}$$

Выражение (2) получено из формулы для вероятностей зависимых событий [5] (в данном случае случайными величинами являются $S_1$ и $\xi$, а $V_1$ и $N$ — постоянными). Аналогичным образом определяется вероятность нахождения входного потока к разделу в состоянии $S_2$. Обозначим через $R(V_1, N, \xi)$ отношение этих вероятностей и преобразуем его следующим образом:

$$R(V_1,N,\xi)=\frac{P(S_1,\xi\,|\,V_1,N)}{P(S_2,\xi\,|\,V_1,N)}=\frac{P(\xi\,|\,S_1,V_1,N)P(S_1\,|\,V_1,N)}{P(\xi\,|\,S_2,V_1,N)P(S_2\,|\,V_1,N)}. \tag{3}$$

Вероятности того, что из $N$ пришедших заявок $\xi$ будут найдены в кэше, описываются биномиальным распределением:

$$\begin{cases}P(\xi\,|\,S_1,V_1,N)=C_N^{\xi}h_1^{\xi}(1-h_1)^{N-\xi},\\ P(\xi\,|\,S_2,V_1,N)=C_N^{\xi}h_2^{\xi}(1-h_2)^{N-\xi}.\end{cases} \tag{4}$$

Вероятность того, что заявка найдена в кэше, равна $h_1$ и $h_2$ для состояний входного потока $S_1$ и $S_2$ соответственно (см. рисунок). Заметим, что $P(S_1\,|\,V_1,N)$ и $P(S_2\,|\,V_1,N)$ не зависят от случайной величины $\xi$ и определяются исключительно из матрицы переходов марковской цепи (1):

$$\begin{cases}P(S_1\,|\,V_1,N)=P(S_1),\\ P(S_2\,|\,V_1,N)=P(S_2).\end{cases} \tag{5}$$

Подставив (4) и (5) в выражение (3), после необходимых преобразований получим:

$$R(V_1,N,\xi)=\left(\frac{h_1(1-h_2)}{h_2(1-h_1)}\right)^{\xi}\left(\frac{1-h_1}{1-h_2}\right)^{N}\frac{P(S_1)}{P(S_2)}, \tag{6}$$

таким же образом получим отношение для потока к разделу размера $V_2$:

$$R(V_2,N,\xi)=\left(\frac{h_3(1-h_1)}{h_1(1-h_3)}\right)^{\xi}\left(\frac{1-h_3}{1-h_1}\right)^{N}\frac{P(S_1)}{P(S_2)}. \tag{7}$$

По формулам (6), (7) можно оценить состояние входного потока за временной интервал. Будем считать, что при $R>1$ входной поток в состоянии $S_1$, а при $R\le 1$ — в $S_2$:

$$\hat{S}(V,N,\xi)=\left[\begin{array}{l}\hat{S}_1,\ \text{если } V=V_1, R(V_1,N,\xi)>1,\\ \hat{S}_2,\ \text{если } V=V_1, R(V_1,N,\xi)\le 1,\\ \hat{S}_1,\ \text{если } V=V_2, R(V_2,N,\xi)>1,\\ \hat{S}_2,\ \text{если } V=V_2, R(V_2,N,\xi)\le 1.\end{array}\right. \tag{8}$$

Заметим, что достаточно определить оценку (8) лишь для одного потока: при оценке состояния первого потока $\hat{S}_1$ ему назначается объем $V_1$, а второму — $V_2$ (это связано с тем, что $V_{\text{cache}}=3V_1$, а распределять не весь объем кэша нецелесообразно). Если оценка состояния первого потока $\hat{S}_2$, ему назначается объем $V_2$, а второму — $V_1$.

Кэш выделяется, в первую очередь, тому потоку, оценка состояния которого более достоверна. Достоверность определяется тем, насколько значение $R$ отличается от единицы: при $R=1$ оценки становятся равновероятными, а значит их достоверность минимальна.

Таким образом, вначале вычисляются значения $R$ для каждого раздела, затем устанавливается размер того раздела, для которого $R$ больше (если $R<1$, то в сравнении будет участвовать величина $\frac{1}{R}$). Другому разделу назначается оставшийся объем кэша.

**Оценка работы алгоритма.** Рассмотрим методику оценки качества работы алгоритма управления кэшем, которое определим как вероятность того, что на каком-либо временном интервале требования на HR будут удовлетворены для обоих разделов. На качество работы алгоритма влияют ошибка оценки текущих состояний потоков, вызванная низкой интенсивностью входных потоков, а также ошибка определения последующего состояния, связанная со случайным характером переключения состояний входных потоков. Сначала рассмотрим идеальный случай, когда обе ошибки отсутствуют (построим верхнюю границу качества), затем — когда присутствует лишь ошибка определения последующего состояния и, наконец, когда присутствуют обе ошибки.

Вычислим верхнюю границу качества описанного алгоритма. Будем считать, что состояния потоков на каждом временном интервале известны заранее. Соответственно выбираются оптимальные размеры разделов во всех случаях, когда это возможно. Требования по HR не выполняются лишь тогда, когда оба входных потока находятся в состоянии $S_2$, поскольку в этом случае потребуется объем $2V_2 > V_{\text{cache}}$. Верхняя граница качества алгоритма определяется следующим выражением:

$$Q_1 = 1 - P_1(S_2)P_2(S_2), \tag{9}$$

где $P_1(S_2)$ и $P_2(S_2)$ — вероятность нахождения в состоянии $S_2$ потоков запросов к 1-му и 2-му разделам, которая определяется из формулы полной вероятности [5]:

$$P_i(S_2) = P_i(S_1)P_i(S_2 \mid S_1) + P_i(S_2)P_i(S_2 \mid S_2), \tag{10}$$

где $i = 1, 2$. Поскольку поток может быть в состоянии $S_1$ либо в состоянии $S_2$,

$$P_i(S_1) = 1 - P_i(S_2). \tag{11}$$

Подставим (11) в (10) и после необходимых преобразований получим:

$$P_i(S_2) = \frac{P_i(S_2 \mid S_1)}{P_i(S_1 \mid S_2) + P_i(S_2 \mid S_1)}. \tag{12}$$

Рассуждая аналогичным образом, можно вычислить вероятность нахождения потока в состоянии $S_1$:

$$P_i(S_1) = \frac{P_i(S_1 \mid S_2)}{P_i(S_1 \mid S_2) + P_i(S_2 \mid S_1)}. \tag{13}$$

Перепишем (9) с учетом (12):

$$Q_1 = 1 - \frac{P_1(S_2 \mid S_1)P_2(S_2 \mid S_1)}{\big(P_1(S_1 \mid S_2) + P_1(S_2 \mid S_1)\big)\big(P_2(S_1 \mid S_2) + P_2(S_2 \mid S_1)\big)}. \tag{14}$$

Вычислим оценку качества алгоритма при условии, что известны состояния входных потоков за прошедший временной интервал (интенсивность потоков $N \to \infty$), при этом прогноз следующего состояния осуществляется на основе матрицы переходов (1).

В таблице перечислены состояния системы, состоящей из двух разделов. Каждое состояние характеризуется объемом разделов и состояниями их входных потоков. Заметим: когда система находится в состояниях 1, 2, 5, 7, разделам хватает выделенных объемов для выполнения требований. В состояниях 4, 8 требования не могут быть удовлетворены в принципе, поскольку входные потоки к обоим разделам находятся в состоянии $S_2$. В состояниях 3, 6 требования могут быть выполнены, если объем разделов выбран правильно. Таким образом, оценка качества работы данного алгоритма вычисляется с помощью следующего выражения:

$$Q_2 = Q_1 - (P(3) + P(6)), \tag{15}$$

где $P(3)$, $P(6)$ — вероятности перехода системы в соответствующие состояния. Выражение (15) справедливо, поскольку $P(3)$ и $P(6)$ — вероятности несовместных событий.

Введем обозначение $V^{(i)} = \{V^{(1)}, V^{(2)}\}$. Пусть при $V^{(1)}$ первый раздел получает объем $V_1$, второй $V_2$, а при $V^{(2)}$ — наоборот.

Пусть $P_{\text{ош}}(S^{(1)},S^{(2)},V^{(1)})$ — вероятность того, что распределение разделов $V^{(1)}$ даст ошибку, если на предыдущем интервале состояние первого раздела было $S^{(1)}$, а второго — $S^{(2)}$. Аналогичным образом определяется $P_{\text{ош}}(S^{(1)},S^{(2)},V^{(2)})$. Тогда минимальная вероятность ошибки для ситуации $S^{(1)}=S_1$, $S^{(2)}=S_1$ определяется следующим образом:

$$P_{\text{ош}}(S_1,S_1)=\min\left\{P_{\text{ош}}(S_1,S_1,V^{(1)}),P_{\text{ош}}(S_1,S_1,V^{(2)})\right\}.$$

| № состояния | Объем 1-го раздела | Объем 2-го раздела | Состояние 1-го потока | Состояние 2-го потока | Требования по HR |
|---|---|---|---|---|---|
| 1 | $V_1$ | $V_2$ | $S_1$ | $S_1$ | Выполнены |
| 2 | $V_1$ | $V_2$ | $S_1$ | $S_2$ | Выполнены |
| 3 | $V_1$ | $V_2$ | $S_2$ | $S_1$ | Не выполнены из-за ошибки |
| 4 | $V_1$ | $V_2$ | $S_2$ | $S_2$ | Невыполнимы |
| 5 | $V_2$ | $V_1$ | $S_1$ | $S_1$ | Выполнены |
| 6 | $V_2$ | $V_1$ | $S_1$ | $S_2$ | Не выполнены из-за ошибки |
| 7 | $V_2$ | $V_1$ | $S_2$ | $S_1$ | Выполнены |
| 8 | $V_2$ | $V_1$ | $S_2$ | $S_2$ | Невыполнимы |

Вероятность ошибки определяется как сумма таких вероятностей для всех комбинаций $S^{(1)}$, $S^{(2)}$:

$$P_{\text{ош}}=P(3)+P(6)=\sum_{i=1}^{2}\sum_{j=1}^{2}\min\left\{P_{\text{ош}}(S_i,S_j,V^{(1)}),P_{\text{ош}}(S_i,S_j,V^{(2)})\right\}. \quad (16)$$

Перепишем (15) с учетом (16):

$$Q_2=Q_1-\sum_{i=1}^{2}\sum_{j=1}^{2}\min\left\{P_{\text{ош}}(S_i,S_j,V^{(1)}),P_{\text{ош}}(S_i,S_j,V^{(2)})\right\}.$$

Очевидно, что данный алгоритм работает лучше, когда состояния входных потоков изменяются редко: при стремлении элементов главной диагонали матрицы (1) к единице $Q_2$ стремится к $Q_1$ ($P(S_1\,|\,S_2)$ и $P(S_2\,|\,S_1)$ стремятся к нулю, вероятности $P(3)$ и $P(6)$ стремятся к нулю и выражение (15) вырождается в (14)).

Теперь рассмотрим ситуацию, когда оценка состояний входных потоков производится при конечной выборке. В этом случае к ошибке оценки следующего состояния добавляется ошибка оценки текущего состояния.

Определим зависимость качества алгоритма от интенсивности $N$. Для этого выпишем матрицу переходов системы из двух разделов:

$$\begin{cases} P(1)=\sum_{i=1}^{8}P(i)P(1\,|\,i),\\ P(2)=\sum_{i=1}^{8}P(i)P(2\,|\,i),\\ \quad\vdots\\ P(8)=\sum_{i=1}^{8}P(i)P(8\,|\,i).\end{cases} \quad (17)$$

Пусть $V(i)$ — распределение объемов разделов в *i*-м состоянии системы, например, $V(6)=V^{(2)}$.

Поскольку изменение объема раздела и изменение состояний входных потоков являются независимыми событиями, вероятность перехода из *j*-го в *i*-е состояние системы может быть представлена в виде произведения вероятности изменения объема раздела с $V_{t-1}$ на $V_t$ и вероятностей переходов состояний потоков 1-го и 2-го раздела соответственно: $P_1(S_t \mid S_{t-1})$ и $P_2(S_t \mid S_{t-1})$:

$$P(i\mid j)=P(V_t\ ,S_t^{(1)},S_t^{(2)}\mid V_{t-1},S_{t-1}^{(1)},S_{t-1}^{(2)})=P(V_t\ \mid V_{t-1},S_{t-1}^{(1)},S_{t-1}^{(2)})P_1(S_t\ \mid S_{t-1})P_2(S_t\ \mid S_{t-1}).$$

Вычислим $P(V_t\ \mid V_{t-1},S_{t-1}^{(1)},S_{t-1}^{(2)})$ для случая $V_t=V^{(2)},V_{t-1}=V^{(1)}$. Пусть оценка состояния первого потока оказалась более достоверной. Для того чтобы объем 1-го раздела стал равен $V_2$, необходимо, чтобы в предыдущем интервале была получена оценка состояния потока $\hat{S}^{(2)}=S_2$. При этом объем данного раздела в предыдущем интервале должен быть равен $V_1$. Из (8) следует, что для этого должно быть выполнено условие $R(V_1,N,\xi)\le 1$. Соответственно $P(V_t\ \mid V_{t-1},S_{t-1}^{(1)},S_{t-1}^{(2)})=P(R(V_1,N,\xi)\le 1)$.

После необходимых преобразований с учетом (6), (12), (13) получим:

$$P(V_t\ \mid V_{t-1},S_{t-1}^{(1)},S_{t-1}^{(2)})=P\left(\xi<\ln\left(\left(\frac{1-h_2}{1-h_1}\right)^N\frac{P_1(S_2\mid S_1)}{P_1(S_1\mid S_2)}\right)\ln\left(\frac{h_1(1-h_2)}{h_2(1-h_1)}\right)\right). \quad (18)$$

Обозначим правую часть неравенства (18) через $\Xi$ и перепишем его с учетом того, что $\xi$ имеет биномиальное распределение:

$$P(V_t\ \mid V_{t-1},S_{t-1}^{(1)},S_{t-1}^{(2)})=\sum_{m=0}^{m<\Xi}C_N^m h_2^m(1-h_2)^{N-m}. \quad (19)$$

Вычислив аналогичным образом все условные вероятности системы (17), получим систему из 8 уравнений с 8 неизвестными. Решив эту систему, получим значения *P*(4), *P*(8), *P*(3) и P(6). Качество работы данного алгоритма оценивается как

$$Q_3(N)=1-(P(4)+P(8)+P(3)+P(6)).$$

**Заключение.** В работе построена модель кэша системы хранения данных, состоящего из двух разделов, размеры которых адаптивно изменяются. Исследовано влияние ошибки определения текущего состояния потока и ошибки выбора следующего состояния потока на качество работы алгоритма. В дальнейшем предполагается обобщить данные результаты для ситуации, когда кэш содержит более двух разделов.

## СПИСОК ЛИТЕРАТУРЫ

1. *Qureshi M. K., Patt Y. N.* Utility-Based Cache Partitioning: A Low-Overhead, High-Performance, Runtime Mechanism to Partition Shared Caches // Proc. of the 39th Annual IEEE/ACM Intern. Symp. on Microarchitecture (MICRO 39). IEEE Comp. Soci. Washington, DC, USA, 2006. P. 423—432.

2. *Goyal P., Jadav D., Modha D. S., Tewari R.* CacheCOW: QoS for storage system caches // Proc. of the 11th Intern. Conf. on Quality of Service (IWQoS'03). Berlin: Springer-Verlag, 2003. P. 498—515.

3. *Feitelson D. G.*, Workload Modeling for Computer Systems Performance Evaluation [Электронный ресурс]: <http://www.cs.huji.ac.il/~feit/wlmod/>.

4. *Марковский С. Г., Тюрликов А. М.* Использование адресов абонентов для разрешения конфликтов в канале с шумом // ИУС. 2006. № 2(21). С. 27—37.

5. *Венцель Е. С.* Теория вероятностей. М.: Наука, 1964. 575 с.

**Сведения об авторах**

***Василий Сергеевич Дужин*** — аспирант; Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра безопасности информационных систем; E-mail: vduzhin@gmail.com

***Григорий Сергеевич Евсеев*** — канд. техн. наук, доцент; Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра моделирования вычислительных и электронных систем; E-mail: egs@vu.spb.ru

***Евгений Михайлович Линский*** — канд. техн. наук, доцент; Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра безопасности информационных систем; E-mail: evlinsky@vu.spb.ru



Рекомендована кафедрой
№ 51 безопасности информационных систем

Поступила в редакцию
01.02.13 г.