УДК 004.074.32

Олександр Марковський,
Аббасі Шагін,
Вікторія Максимук

ЕФЕКТИВНА ОРГАНІЗАЦІЯ БАГАТОРІВНЕВОЇ ХЕШ-ПАМ'ЯТІ

У статті пропонується підхід до підвищення ефективності організації багаторівневої хеш-пам'яті для пошуку за ключем. Верхній рівень такої хеш-пам'яті утворює пошуковий кеш-буфер, а нижній – основна пам'ять. Особливістю організації хеш-пошуку, що досліджується в статті, є квазістатичний характер ключів. Це означає, що інтенсивність операцій зміни ключів на порядки менша за інтенсивність операцій пошуку за ключем. Розроблена ймовірнісна математична модель хеш-пошуку у квазипостійних масивах ключів, які зберігаються в дворівневій хеш-пам'яті. На основі цієї моделі запропоновано підхід до оптимізації параметрів хеш-пам'яті. Теоретично та експериментально доведено, що запропонована організація потребує не більше одного звернення до повільної пам'яті нижнього рівня.

Ключові слова: хеш-пам'ять, хеш-пошук, колізії адресації.
Бібл.: 2.

In paper an approach for increasing effectiveness of organization of multi-level hash-memory for rapid retrieval by key. Upper level of such hash-memory is retrieval buffer in cash memory unit and low level is in main memory. The peculiarities of researched hash-memory organization cocsist of near-permanent characters of key array. It means that intensively of retrieval operation by few order is height in compare to operations connected with key change. The probabilistic model of hash-searching in near-permanent keys arrays which are storages in two-level memory has been developed. On the base of this model the effectiveness organization of hash-access to information has been proposed. It has been shown that proposed hash-access organization requires ensure no more then one access to slow low-level memory.

Key words: hash-memory, hash-retrieval, addressing collign.
Bibl.: 2

Актуальність теми дослідження. Пошук за ключем традиційно є однією з базових процедур обробки інформації. Для багатьох практично важливих застосувань питома вага операцій пошуку складає до 30-40% [1]. В останні роки багаторазово виросли обсяги інформаційних масивів, у яких виконується пошук, значно більш жорсткими стали вимоги щодо оперативності пошуку. Вказані чинники диктують необхідність створення нових підходів до організації пошуку інформації в сучасних та перспективних комп'ютерних системах.

Потенційно найбільшу швидкодію пошуку забезпечує хеш-адресація. До останнього часу суттєвою перешкодою на шляху широкого застосування хеш-пошуку було те, що ця технологія вимагає використання надлишкового об'єму пам'яті. Проте вражаючі успіхи інтегральної технології виготовлення мікросхем пам'яті призвели до їх суттєвого здешевлення, що створює технологічні передумови розширення використання хеш-пам'яті.

Таким чином, наукова задача вдосконалення та підвищення ефективності технології хеш-пошуку є актуальною на сучасному етапі розвитку інформаційних технологій.

Постановка проблеми. Сутність хеш-адресації полягає в тому, що ключі і пов'язана з ними інформація записуються за адресою, яка однозначно залежить від ключа. Суттєвим недоліком хеш-адресації є наявність колізій. Вони виникають коли два і більше ключів претендують на одну адресу. Технологічно проблема колізій вирішується за допомогою лінійного пробінгу. Його реалізація полягає в перевірці на вільність адреси, до якої звертається ключ: якщо ця комірка пам'яті зайнята, по порядку перевіряються наступні, і запис виконується в першу вільну комірку. Пошук здійснюється шляхом перевірки заданого ключа зі значеннями, записаними в хеш-памяті, за обчисленою хеш-адресою: коли вони співпадають, пошук завершено. В іншому випадку потрібно послідовно переглянути наступні комірки; знаходження порожньої вказує на те, що такого ключа не існує. Відомим недоліком використання лінійного пробінгу є вторинне групування записів, що призводить до сповільнення пошуку [1].

Аналіз останніх досліджень та публікацій. Одним з найважливіших підходів до формування хеш-перетворень, що не породжують колізій для заданого масиву ключів, є сегментування. Сегментування передбачає розбиття заданого масиву ключів на безліч піднаборів і формування хеш-перетворень, що не породжують колізії для кожного з піднаборів окремо.

Рекурсивний метод формування хеш-перетворення, не породжуючого колізій, запропонований в [2], заснований на використанні сімейств хеш-функцій, що не породжують колізій.

Ефективність рекурсивного формування функцій хеш-перетворення значною мірою залежить від ступеня заповнення хеш-пам'яті: при коефіцієнтах заповнення хеш-пам'яті більших за 0.75 час пошуку хеш-перетворення може стати критичним.

Виділення недосліджених частин загальної проблеми. Основними недоліками існуючих методів побудови хеш-перетворень для квазістатичних масивів ключів є те, що вони потребують дуже багато часу при коефіцієнті заповнення більше 0.75, а також те, що вони не враховують особливостей організації реальної пам'яті комп'ютерних систем, зокрема її багаторівневий характер.

Постановка завдання. Ціллю досліджень є підвищення ефективності хеш-адресації за рахунок оптимізації параметрів хеш-пам'яті при її багаторівневій організації.

Викладення основного матеріалу. Модель дворівневої хеш-пам'яті. При хеш-пошуку в сучасних обчислювальних системах, пам'ять яких має багаторівневу організацію, є ряд особливостей, які можуть принципово змінити підхід до швидкого пошуку даних.

Розглядається дворівнева хеш-пам'ять, що складається з повільнодіючої спільної пам'яті і швидкодіючої кеш-пам'яті, об'єм якої обмежений.

В основу розроблюваної моделі покладена концепція отримання хеш-перетворення $H(X)$, що забезпечує відображення заданої множини Ω із m ключів на s сторінках хеш-пам'яті таким чином, щоб кількість ключів, адресованих у кожну зі сторінок, не перевищувала $(\alpha+\delta)\cdot w$, де α- коефіцієнт завантаження хеш-пам'яті, δ - допустима варіація завантаження сторінки хеш-пам'яті,

$$\alpha+\delta\leq 1.$$

Коефіцієнт α завантаження хеш-пам'яті визначається відношенням кількості m записів, що зберігаються в ній до максимально можливої їх кількості $M = s\cdot w$, виходячи з обсягу хеш-пам'яті:

$$\alpha = \frac{m}{M} = \frac{m}{s \cdot w}. \quad (1)$$

Отримання хеш-перетворення $H(X)$, що задовольняє вказаній вище умові, може бути отримане шляхом підбору. В якості механізму підбору хеш-перетворення доцільним представляється використання стандартизованого шифроблоку типу DES або Rijndael, який здійснює криптографічне однозначне шифрування з використанням ключа K даних D у код C: $C = HK(D)$ [2]. Ключова умова для пошуку інформації X у такому варіанті використовується в якості вхідних даних шифроблоку, ключ K шифроблоку виступає в ролі настроювального коду і є, по суті, номером хеш-перетворення. Вихідний код C шифроблоку поділяється на дві частини: h-розрядний фрагмент використовується як хеш-адреса $A_K(X)$ сторінки, а розряди, що залишилися, представляють собою хеш-згортку $SK(X)$ ключа X пошуку. Відповідно, підбір хеш-перетворення $HK(X)$ виконується шляхом зміни ключа K шифроблоку.

Вважаючи, що на сторінці може розміститися w ключів, хеш-адреса $AK(X)$ сторінки є $h = log_2 s$ - розрядний код. Хеш-функція розподіляє m ключів по s групах, які містять $\eta_1,\eta_2,\ldots,\eta_s$ ключів, причому $\sum_{j=1}^{s} \eta_j = m$. Для того, щоб хеш-перетворення розподіляло ключі сторінками хеш-пам'яті, необхідно, щоб кількість хеш-адрес кожної з сторінок не перевищувала максимально допустимого числа $u=(\alpha+\delta)\cdot w$ записів на сторінці: $\forall j \in \{1,\ldots,s\}$: $\eta_j \le u$. При цьому в кожній зі сторінок хеш-пам'яті забезпечується наявність вільного об'єму пам'яті, достатнього для розміщення $w\cdot(1-\alpha-\delta)$ записів.

Якщо виходити з того, що хеш-перетворення $H_K(X)$ формує рівномірно розподілені хеш-адреси, то в одну сторінку, в середньому, потрапляє m/s хеш-адрес. В якості теоретичної моделі розподілу хеш-адрес коректно використовувати ймовірнісну модель Бернуллі. Відповідно до цієї моделі, попадання хеш-адрес m ключів у межі фіксованої сторінки пам'яті можна розглядати як m дослідів, у кожному з яких з імовірністю $1/s$ відбувається подія: потрапляння хеш-адреси в адресний простір цієї сторінки. Тоді згідно з властивостями даної моделі, можна вважати, що математичне очікування числа хеш-адрес, що потрапляють у фіксовану сторінку, дорівнює m / s з дисперсією $m \cdot \frac{1}{s} \cdot (1 - \frac{1}{s})$. Тоді, відповідно до теореми Муавра-Лапласа, кількість хеш-адрес, що потрапляють у рамки сторінки, підпорядкована розподілу Гауса з математичним очікуванням m/s і дисперсією $m \cdot \frac{1}{s} \cdot (1 - \frac{1}{s})$. Якщо врахувати, що число s сторінок хеш-пам'яті достатньо велике, то дисперсію кількості хеш-адрес, що потрапляють у фіксовану сторінку, можна наближено вважати рівною m / s.

Імовірність P_{os} переповнення сторінки, тобто ймовірність того, що число хеш-адрес, що потрапляють у фіксовану сторінку хеш-пам'яті, перевищить u, для нормального розподілу з урахуванням (1) визначається наступним виразом:

$$P_{os} = 0.5 - \Phi(\frac{u - m/s}{\sqrt{m/s}}) = 0.5 - \Phi(\frac{w\cdot(\alpha+\delta) - m/s}{\sqrt{m/s}}) = 0.5 - \Phi(\delta \cdot \sqrt{\frac{w}{\alpha}}) \quad (2)$$

Для постійного масиву ключів, тобто в ситуації, коли в сторінці немає необхідності мати запас вільної пам'яті для $w\cdot(1-\alpha-\delta)$ записів, $\delta = 1- \alpha$ і ймовірність P_{osw} переповнення сторінки, тобто ймовірність того, що число хеш-адрес, що потрапляють у фіксовану сторінку хеш-пам'яті, перевищить w, визначається наступним виразом:

$$P_{osw} = 0.5 - \Phi(\sqrt{w} \cdot \frac{1-\alpha}{\sqrt{\alpha}}) \quad (3)$$

З виразу (3) випливає, що ймовірність переповнення сторінки визначальною мірою залежить від значення коефіцієнту δ запасу вільної пам'яті в сторінці, від коефіцієнта α заповнення хеш-пам'яті, а також від об'єму сторінки і пам'яті, що потрібна для зберігання одного запису .

Для того, щоб виключити переповнення всіх s сторінок при заповненні фіксованою множиною Ω з m ключів, необхідно підібрати відповідне хеш-перетворення. Ймовірність P_0 того, що за одну пробу хеш-перетворення вдасться виключити переповнення всіх сторінок хеш-пам'яті визначається добутком ймовірностей того, що кожна з s сторінок не буде переповнена:

$$P_0 = (1 - P_{OS})^S = \left[0.5 + \Phi(\delta \cdot \sqrt{\frac{w}{\alpha}}) \right]^S \quad (4)$$

Середня кількість g проб, які необхідно виконати для підбору хеш-перетворення, при якому виключається переповнення сторінок хеш-пам'яті, визначається наступним виразом:

$$g = \sum_{j=1}^{\infty} j \cdot P_0 \cdot (1 - P_0)^{j-1} = \frac{1}{P_0} \quad (5)$$

Проведені експериментальні дослідження на основі статистичного моделювання довели адекватність викладеної математичної моделі хеш-адресації квазістатичного масиву ключів в хеш-пам'яті, розділеної на сторінки. Запропоновану модель можна використовувати для оптимізації параметрів хеш-пам'яті.

Висновки. Розроблено модель хеш-пошуку в квазістатичних масивах ідентифікаційних кодів користувачів, що враховує багаторівневу організацію пам'яті. На основі розробленої моделі запропонована організація хеш-пошуку в квазіпостійних масивах ключів. Доведено, що час доступу до інформації по ключу визначається зверненням не більше, ніж до однієї сторінки пам'яті нижнього рівня. Запропонована організація хеш-пам'яті може бути ефективно використана для підвищення швидкості пошуку в сучасних комп'ютерних системах для широкого кола практично важливих застосувань.

Список використаних джерел

1. Марковский А.П. Хеш-память с ограниченным временем поиска по ключу/ А.П. Марковский, Е.В. Порхун, А.В. Мнацаканов // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка.- 2008,- № 49,- С.156-162.

2. Марковський О.П., Максимук В.Р. Організація мультиадресації в хеш-пам'яті // Збірник тез доповідей 9-ї наукової конференції Прикладна математика та комп'ютинг ПМК-2017. Київ 19-21 квітня 2017. – К.:Просвіта, 2017 – С.195-199.

ДОВІДКА ПРО АВТОРІВ

Марковський Олександр Петрович – кандидат технічних наук, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Oleksandr Markovskyi – Associate Professor, *PhD,* Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Аббасі Шагін – студент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Abbasi Shagin –student, Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Максимук Вікторія Романівна – студентка, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Viktoriia Maksymuk –student, Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

MarkovskyiO.P.,
AbbasiShagin,
MaksymukV.R.

EFFECTIVENESS MULTI-LEVEL HASH-MEMORY ORGANIZATION

Acute problem. The Hash function is considered to be the traditional technique for the distribution of keys in an unstructured memory, providing direct access. While the ideal condition would be that one and only one key corresponds to a certain address, in practice, during the use of common algorithms of Hashing, this condition is not fulfilled. There are occasions where different keys correspond to the same address. This situation is called a *collision*. Collisions can be deal with using a variety of methods. Common characteristic for all these methods is the reduction of the speed at which information can be searched in the memory, along with an increase in memory requirements.

Target setting. The ability to search quickly for information is considered a major problem in a wide area of applications, like searching information systems. Hashing as a method of searching for information has existed for many years, but in the above mentioned method the problem of collisions is an inhibitory factor in practical applications.

One of the most effective methods for resolving the problems of collisions, for constant number of keys is the function "Hash transformation, free of collisions", which is called Perfect Hashing Boolean Function.

From the above table we can conclude that in the field of searching, a Hash algorithm based method proves the relation between the Hash Function and the forming of the Hash address bits is a linear function of the key bits.

Actual scientific researches and issues analysis. The negative aspects of the Perfect Hash Function are considered to be the following:

a) The use only of constant (not varied) number of keys.

b) The fact that the process for searching the Perfect Hash function for a given number of keys demands useful time. Recently several of algorithms for searching Perfect Hash Functions have been proposed [9-21]. All these algorithms use, more or less, the force attack searching method to define the subset G (which includes all the Perfect Hash Functions), from the set of all the possible Hash transformations. The effectiveness of these algorithms is determined from the rate of settlement of accommodation in fulfilling the following conditions:

- The algorithm should obtain the Perfect Hash Function (or the Minimum Hash Function) for any number of keys. That means requires force attack searching in the determinate subset G, in which are included all the Perfect Hash Functions.
- The number of functions of the subset G, in which force attack searching is executed, should be the smallest one.
- The time required to calculate the Hash Function should be the least possible.
- We consider in advance that we must have the highest possibility that the Hash Function does not collide for a given set of keys.

Uninvestigated parts of general matters defining. The algorithm that is mentioned in study and which is considered to be one of the most effective, ensures the searching time of the Minimum Perfect Hash Function is proportional to m, where m=number of keys which must be distributed in memory.

The research objective. The fractional use of Hash Function for the forming of bits, in practice, does not cause any changes in the expected results. Having as initial state the aforementioned theoretical research we can form a practical method for the design of both a Minimum and not a Minimum Perfect Hash Function, while being able at any time to create the software in Assembly code.

The statement of basic materials. In paper an approach for increasing effectiveness of organization of multi-level hash-memory for rapid retrieval by key. Upper level of such hash-memory is retrieval buffer in cash memory unit and low level is in main memory. The peculiarities of researched hash-memory organization cocsist of near-permanent characters of key array. It means that intensively of retrieval operation by few order is height in compare to operations connected with key change. The probabilistic model of hash-searching in near-permanent keys arrays which are storages in two-level memory has been developed. On the base of this model the effectiveness organization of hash-access to information has been proposed. It has been shown that proposed hash-access organization requires ensure no more then one access to slow low-level memory.

Conclusions. In paper the new hash-searching organization has been proposed. The essence of this organization consist of using multi hash addressing for several key. It is allowed to remove the grouping keys in hash memory. The procedures of recording keys into the hash-memory and key searching been developed. The experimental evaluation of average of memory access for key recording and key searching has been obtained.

Key words: hash-memory, hash-retrieval, addressing collign.