УДК 004.942(045)

В.М. Боровик, к.т.н., с.н.с.

ОПТИМІЗАЦІЯ КЕРУВАННЯ ЧЕРГАМИ ЗАПИТІВ У СИСТЕМАХ «КЛІЄНТ–СЕРВЕР»

Національний авіаційний університет
E-mail: vborovik@ukr.net

Розроблено математичну модель керування чергами запитів у системах «клієнт–сервер». Розглянуто алгоритм визначення ситуаційних пріоритетів прийняття в чергу запитів. Зосереджено увагу на можливості витісненням запитів у разі заповненої черги. Показано практичну реалізацію отриманих результатів.

A mathematical model for managing queues of requests in the systems of "client-server". An algorithm determining the situational setting priorities requests in the queue. Particular attention is paid to the possibility of displacement of ochereli queries when filling the queue. Explains the practical realization of obtained data.

Разработана математическая модель управления очередями запросов в системах «клиент–сервер». Рассмотрен алгоритм определения ситуационных приоритетов постановки запросов в очередь. Уделено внимание возможности вытеснения из очереди запросов при заполненной очереди. Показана практическая реализация полученных результатов.

Вступ

У роботі [1] детально розглядається поведінка запитів у черзі для систем «клієнт–сервер». Ускладнена модель структури черги враховує можливість вилучення запитів із черги, що більше відповідає реальним практичним ситуаціям.

Головна концепція – побудова ефективної системи розподіленого оброблення інформації для систем баз даних на основі структури мережі «клієнт–сервер» [2; 3]. Важлива складова досліджуваної моделі – керування чергою запитів у разі появи нових запитів [4; 5].

Під час розроблення математичної моделі як системи масового обслуговування отримуємо основу прикладної задачі керування системою «клієнт–сервер», з використанням такої термінології:

система – комп'ютерна мережа;

прилад – комп'ютер, сервер (серверний процес);

вхідний потік заяв – запитів;

процес обслуговування заяви – оброблення запитів;

черга – база даних запитів;

типи вхідних потоків – типи запитів, кожний з яких має однаковий середній час обслуговування;

дисципліна обслуговування – пріоритет запитів у разі постановки в чергу чи обслуговуванні приладом.

Мета роботи – оптимізувати керування запитами, зосередивши увагу на практичній реалізації – алгоритмі пріоритетного оброблення інформації.

Постановка проблеми

Усі заяви (надалі – запити) надходять на прилад (сервер чи серверний процес) у деякі випадкові моменти часу

$$t_0 < t_1 < t_2 \ldots < t_k \ldots < \ldots .$$

Інтервал між їх появою

$$\tau_k = t_k - t_{k-1}$$

є незалежним випадковим значенням з експоненціальним законом розподілу:

$$F(t) = P\{\tau_k \leq \ \} = 1 - \ ^{-\lambda} ,$$

Середній час між послідовними моментами появи двох однотипних запитів $1/\lambda$.

© В.М. Боровик, 2011

Далі будемо розрізняти типи запитів від відповідного клієнта, а саме, $\lambda_i(\overline{1,n})$ – інтенсивність появи заяв *i*-го типу (від *i*-го клі-єнта).

Таким чином, на сервер надходять *n* пуассонівських вхідних потоків різнотипних запитів.

Усі запити, які надходять на сервер, можуть бути виконані на будь-якому з них.

Час їх виконання буде випадковим значенням із густиною ймовірності

$\varphi(t) = \mu e^{-\mu t}$.

Середній час виконання $1/\mu$ залежить від типу заяви.

У загальному випадку в разі багатоплатформеної архітектури час виконання залежить від типу запиту та номера приладу (комп'ютера, процесора) $_{ij}(i = \overline{1,n}; j = \overline{1,m})$.

У нашому випадку за однотипних серверів час виконання запиту не залежить від номера приладу $\mu_i(i = \overline{1,n})$.

Розглянемо поведінку запитів у черзі.

Якщо прилад вільний і до нього нема черги, то запит, що з'явився в системі, відразу переходить на обслуговування. Це відповідає неальтернативній ситуації керування.

Питання також легко вирішується, якщо в черзі є запити одного типу. Якщо $n > 1$, потрібно вирішити альтернативні ситуації призначення запитів на обслуговування.

Складніша ситуація у разі обмеження кількості місць у черзі та відсутності місць у черзі. У цьому випадку всі інші запити, які надходять у цей момент у систему, отримують відмову, що не дозволяє їм враховуватися у разі постановки на оброблення.

Таке обмеження для багатьох систем масового обслуговування є достатньо важким обмеженням для роботи комп'ютерних мереж.

В умовах конкретної прикладної задачі може виникнути ситуація, коли вигідніше вивести з системи запит деякого типу, щоб ввести більш цінний.

Для можливості керувати такими ситуаціями вводиться пріоритетний параметр $\vartheta_x^i(\vec{\mathbf{i}}\big|_{i_t>0})$, $s,t = \overline{1,n}$, який дозволяє оцінити ймовірність вилучення запитів *s*-го типу запитом *t*-го типу, який знаходиться в повністю заповненій черзі. Оскільки процес витіснення запитів із черги може проходити тільки в моделях із обмеженням на загальну довжину черги, врахуємо умову

$\sum_{s=1}^{n} i_s = R$, де $R > 0$.

Розглянемо ситуаційний пріоритет для керування запитами в черзі. За умови відсутності в черзі місць отримаємо два випадки:

– новий запит з'явився в черзі з ймовірністю $\vartheta_i^+(\vec{\mathbf{i}}), (s = \overline{1,n})$;

– запит *s*-того типу не потрапив у чергу, що відповідає ймовірності невилучення інших запитів $\vartheta_i^-(\vec{\mathbf{i}}), (s = \overline{1,n})$.

Зазначені ймовірності переходів становлять повний набір для цієї ситуації:

$\vartheta_s^+(\vec{\mathbf{i}}) + \vartheta_s^-(\vec{\mathbf{i}}) = 1, (s = \overline{1,n})$.

Імовірність витіснення інших запитів $\vartheta_s^-(\vec{\mathbf{i}})$ може розглядатися для ситуацій, коли в черзі є запити *s*-го типу чи їх нема.

Імовірність витіснення *s*-го запиту запитом того ж типу відповідає його втраті, як і у разі відсутності в черзі. Для цього введемо пріоритетний параметр $\vartheta_i^0(\vec{\mathbf{i}}), (s = \overline{1,n})$ з умови:

$\vartheta_s^0(\vec{\mathbf{i}}) = \vartheta_s^s(\vec{\mathbf{i}}) = \vartheta_s^-(\vec{\mathbf{i}}), (s = \overline{1,n})$.

Витіснення запитів із черги визначається так:

$$\vartheta_s^+(\vec{\mathbf{i}}) = \sum_{\substack{t=1 \\ t \neq s}}^{n} u(i_t)\vartheta_s^t(\vec{\mathbf{i}}), (s = \overline{1,n}).$$

У кінцевому вигляді маємо:

$$\vartheta_s^0(\vec{\mathbf{i}}) = \sum_{\substack{t=1 \\ t \neq s}}^{n} u(i_t)\vartheta_s^t(\vec{\mathbf{i}}), (s = \overline{1,n}).$$

У момент закінчення обслуговування неальтернативна ситуація відповідає існуванню в черзі заяв тільки одного типу. Для цієї ситуації, яка відноситься до тривіального випадку моделей «розмноження та загибелі», стан системи являє собою лінійний ланцюг. Закон вибору заяв на обслуговування в цьому випадку може бути довільним, що не змінює розподіл стаціонарних ймовірностей станів системи.

Для черги з заявами одного типу буде виконуватися дисципліна обслуговування *FCFS* – «перший прийшов – перший обслугований».

За допомогою уведених параметрів отримуємо набір параметрів, які треба усунути (рис. 1).

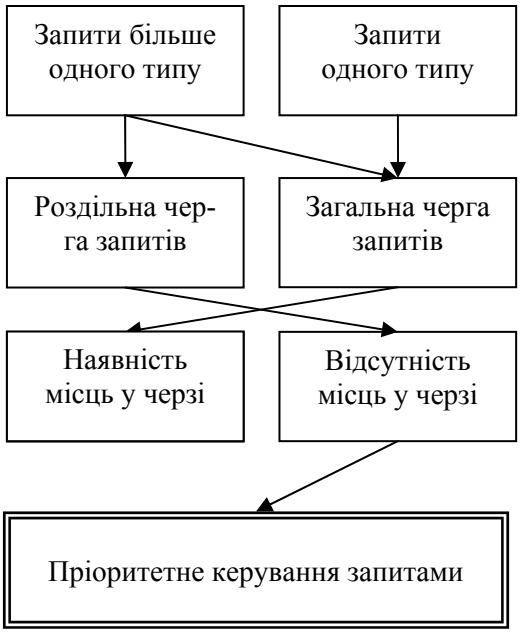

Рис. 1. Класифікація моделей керування чергами запитів

Динаміка переходів для спрощеної моделі з одним приладом (сервером) та одним місцем у черзі за кожним типом показана на рис. 2, де кругами зображено всі можливі стани процесу, а стрілками – можливі перехідні інтенсивновсті (умовні та безумовні).

Для рівноважного випадку складаються рівняння всіх стаціонарних імовірностей станів.

$\pi(k,\vec{1})$:

1) $\pi(0,00)(\lambda_1+\lambda_2)=\pi(1,00)\mu_1+\pi(2,00)\mu_2$;
2) $\pi(1,00)(\lambda_1+\lambda_{2+}\mu_1)= \pi(0,00)\lambda_1+$
$+ \pi(1,10)\mu_1+\pi(2,10)\mu_2$;
3) $\pi(2,00)(\lambda_1+\lambda_2+\mu_2)= \pi(0,00)\lambda_2+$
$+ \pi(2,01)\mu_2+\pi(1,01)\mu_1$;
4) $\pi(1,10)(\lambda_1+\lambda_2+\mu_1)= \pi(1,00)\lambda_1+$
$+ \pi(1,20)\ \mu_1+\pi(2,20)\ \mu_2$;
5) $\pi(1,01)(\lambda_1+\lambda_2+\mu_1)= \pi(1,00)\lambda_2+$
$+ \pi(1,11)\mu_1\delta^1(11)+\pi(2,11)\ \mu_2\delta^1(11)$;
6) $\pi(2,10)(\lambda_1+\lambda_2+\mu_2)= \pi(2,00)\lambda_1+$
$+ \pi(1,11)\mu_1\delta^2(11)+\pi(2,11)\mu_2\delta^2(11)$;
7) $\pi(2,01)(\lambda_1+\lambda_2+\mu_2)=\pi(2,00)\lambda_2+$
$+ \pi(2,02)\mu_2+\pi(1,02)\ \mu_1$;
8) $\pi(2,20)(\lambda_2(\vartheta_2^1(20)-\vartheta_2^0(20))+\mu_2)=$
$= \pi(2,10)\lambda_1+\pi(2,11)\lambda_1\vartheta_1^2(11)$;
9) $\pi(2,02)(\lambda_1(\vartheta_1^2(02)-{}_2^0(20))+\mu_2=$
$= \pi(2,01)\lambda_2+\pi(2,11)\lambda_2\vartheta_2^1(11)$;
10) $\pi(2,11)(\lambda_1(\vartheta_1^2(11)-\vartheta_1^0(11))+$
$+\lambda_2(\vartheta_2^1(11)-\vartheta_2^0(11))+\mu_2=$
$=\pi(2,01)\lambda_1+\pi(2,10)\lambda_2+\pi(2,02)\lambda_1\vartheta_1^2(02)+$
$+\pi(2,20)\lambda_2\vartheta_2^1(20)$;
11) $\pi(1,02)(\lambda_1(\vartheta_1^2(02)-\vartheta_1^0(02))+\mu_1)=$
$=\pi(1,01)\lambda_2+\pi(1,11)\lambda_2\vartheta_2^1(02)$;
12) $\pi(1,20)(\lambda_2(\vartheta_2^1(20)-\vartheta_2^0(20))+\mu_1)=$
$=\pi(1,10)\lambda_1+\pi(1,11)\lambda_1+\pi(1,11)\lambda_1\vartheta_1^2(11)$;
13) $\pi(1,11)(\lambda_1(\vartheta_1^2(11)-\vartheta_1^0(11))+$
$+\lambda_2(\vartheta_2^1(11)-\vartheta_2^0(11))+\mu_1=$
$=\pi(1,01)\lambda_1+\pi(1,10)\lambda_2+\pi(1,02)\lambda_1\vartheta_1^2(02)+$
$+\pi(1,20)\lambda_2\vartheta_2^1(20)$.

Нормуюча умова має такий вигляд:

$$\pi(0,\vec{\mathbf{0}})+\sum_{k=1}^{n}\sum_{\vec{}}\pi(k,\vec{\mathbf{i}})=1.$$

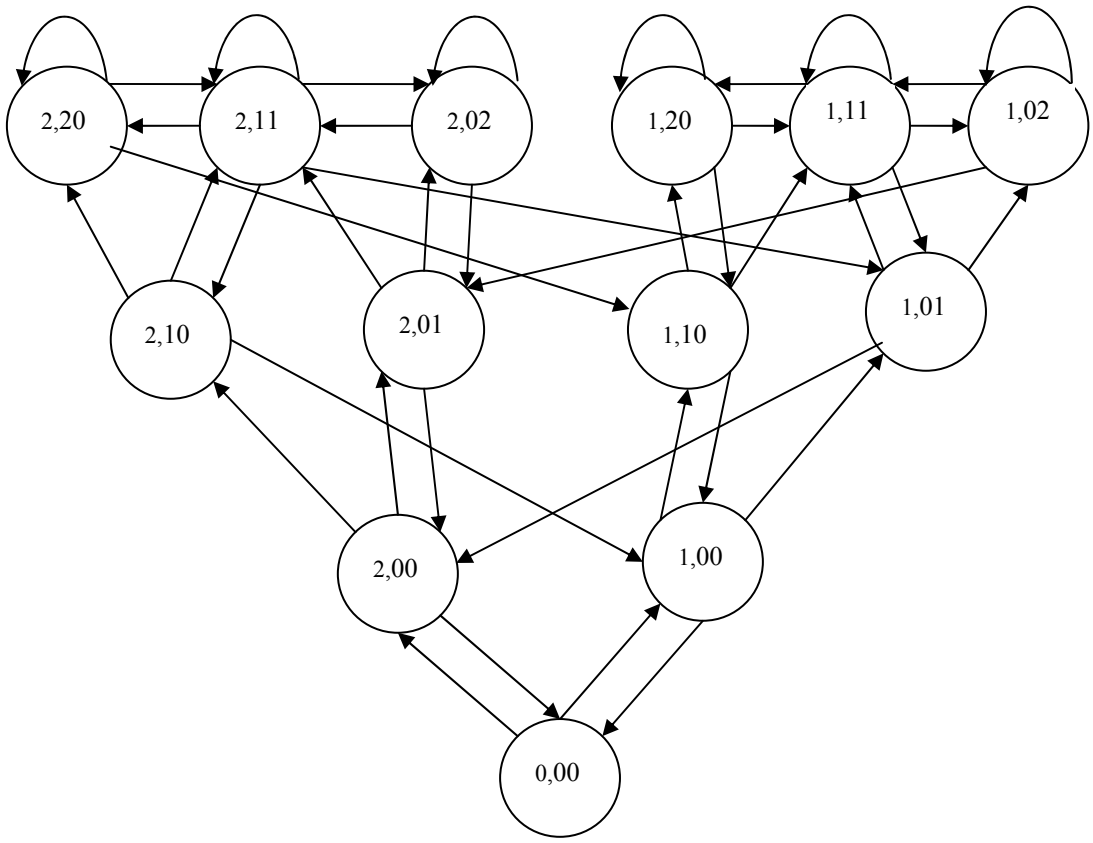

Рис. 2. Граф можливих переходів моделі

Ситуаційний пріоритетний параметр $\vartheta_s^t(\vec{\mathbf{i}})$, $s,t=\overline{1,n}$; $s \neq t$ уводиться тільки для таких станів, де $\mathbf{i}_s > 0$, $s=\overline{1,n}$.

Стани, де пріоритетний параметр $\vartheta_s^t(\vec{\mathbf{i}})$, $s,t=\overline{1,n}$ визначає ймовірність неможливості включення *i*-го запиту в чергу, що можливо для всіх станів, крім тих, де $R=i_s$, призводять до неальтернативної ситуації.

У разі надання інформації «прилад вільний» може з'явитися ситуація, коли у черзі знаходяться запити більше одного типу.

Для вирішення таких конфліктних ситуацій використовується пріоритетний параметр $\delta^s(\vec{\mathbf{i}})$, $s=\overline{1,n}$, який визначає можливість вибору запиту *s*-го типу для обслуговування на сервері. Звичайно, що якась заява обов'язково буде вибрана, тому:

$$\sum_{s=1}^{n} \delta^s(i_1, i_2, \ldots, i_n)\, u(i_s) = 1,$$

де $u(x) = \begin{cases} 1, & \text{якщо } x > 0, \\ 0, & \text{якщо } x \le 0. \end{cases}$

Модель мультісерверної системи

Застосовуючи такий підхід до загальної моделі, де кількість серверів (приладів) більше одного, переходим до векторного запису всіх стаціонарних імовірностей станів.

Порівняно з попереднім результатом зміниться лише одне рівняння:

$$\pi(\vec{\mathbf{k}}, \vec{\mathbf{i}}) \sum_{s=1}^{n} \lambda_s u(R - i_s)[u(\Omega) + [1 - u(\Omega)] \times$$

$$\times \sum_{\substack{t=1 \\ t \neq s}}^{n} \vartheta_s^t(\vec{\mathbf{i}}) u(i_t)] + \sum_{j=1}^{m} \mu_{k_j,j}) =$$

$$= \sum_{s=1}^{n} \lambda_s \{u(i_s)\pi(\vec{\mathbf{k}}, \vec{\mathbf{i}}\big|_{i_s = i_s - 1}) + [1 - u(\Omega)] \times$$

$$\times [\sum_{\substack{t=1\\t\neq s}}^{n} u(i_s)\pi(\vec{\mathbf{k}}, \vec{\mathbf{i}}\Big|_{\substack{i_s=i_s-1,\\ i_t=i_t+1}})\vartheta_s^t(\vec{\mathbf{i}}\Big|_{\substack{i_s=i_s-1,\\ i_t=i_t+1}}) +$$

$$+ u(R - i_s)\vartheta_s^0(\vec{\mathbf{i}})\pi(\vec{\mathbf{k}}, \vec{\mathbf{i}})]\} +$$

$$+ \sum_{j=1}^{m} u(\Omega)[u(\sum_{\substack{t=1\\t\neq k_j}}^{n} i_t)\delta_j^{k_j}(\vec{\mathbf{k}}, \vec{\mathbf{i}}\Big|_{i_{k_j}=i_{k_j}+1}) +$$

$$+ [1 - u(\sum_{\substack{t=1\\t\neq k_j}}^{n} i_t)]]\sum_{s=1}^{n} \mu_{sj}\pi(\vec{\mathbf{k}}\Big|_{k_j=s}, \vec{\mathbf{i}}\Big|_{k_j=k_j+1});$$

$$k_j = \overline{1,n};\ \mu_{ij}, (\overline{1,n};\overline{1,m});$$

$$\sum_{s=1}^{n} i_s > 0;\ \Omega = R - \sum_{s=1}^{n} i_s.$$

Таким чином отримаємо систему з двома керуючими параметрами:

$\vartheta(\cdot)$ – параметр для керування чергами запитів;

$\delta(\cdot)$ – параметр для керування направленням запитів на сервера (прилади).

Із теорії скінченних ланцюгів Маркова відомо, що всі стаціонарні ймовірності станів більше нуля $\pi(\cdot) > 0$. Оптимальне значення керуючих елементів – $\delta(\cdot)$ та $\vartheta(\cdot)$ можуть дорівнювати лише нулю чи одиницю.

Оптимізаційна задача зводиться до пошуку екстремуму функції в задачі лінійного програмування.

Оскільки розрахунки керуючих параметрів ведуться не в реальному часі, їх отримає суто технічний аспект.

На рис. 3 програма «Диспетчер-1» реалізує $\vartheta(\cdot)$, а «Диспетчер-2» реалізує $\delta(\cdot)$.

Висновки

Головним практичним застосуванням отриманих результатів є системи «клієнт-сервер» комп'ютерних мереж.

Результати роботи системи, у тому числі попередньо розраховані керуючі параметри, зберігаються в базі даних.

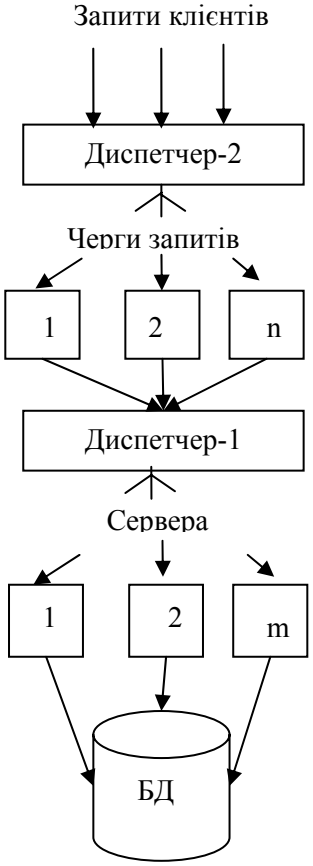

Рис. 3. Структура досліджуваної системи «клієнт–сервер»: БД – база даних

Література

1. *Боровик В.М.* Моделі ситуаційного управління запитами в системах «клієнт-сервер» / В.М. Боровик // Вісник НАУ. – 2010. – № 4. – С. 52–57.

2. *Дейт К. Дж.* Введение в системы баз данных / К. Дж. Дейт. – К.: Вильямс, 2006. – 1328 с.

3. *Карпова Т.* Базы данных: модели, разработка, реализация / Т.Карпова. – СПб.: Питер, 2003. – 304 с.

4. *Меликов А.З.* Телетрафик: Модели, методы, оптимизация / А.З. Меликов, Л.А. Пономаренко, В. В. Паладюк. – К.: Політехніка, 2007. – 256 с.

5. *Меликов А.З.* Математические модели многопотоковых систем обслуживания / А.З. Меликов, Л.А. Пономаренко, П.А. Рюмшин. – К.: Техніка, 1991. – 265 с.

Стаття надійшла до редакції 07.12.2010.