

Описаний тут підхід дозволяє страховій компанії визначати вплив різноманітних факторів на виникнення страхових випадків, і приймати рішення про надання послуг соціального страхування.

1. Айвазян С.А. и др. *Прикладная статистика: Основы моделирования и первичная обработка данных*. – М.: Финансы и статистика, 1983.
2. Иберла К. *Факторный анализ*. – М: Статистика, 1980.
3. *Advances in knowledge discovery and data mining*. Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. (editors), AAAI/MIT Press, 1996.
4. Bijak W., Podgórska M., Utkin J.M., Nowak E. *Matematyka I statystyka finansowa*. //Warszawa, FRR, 1994.
5. Cyrek M. *Komputerowy system ubezpieczeń*. //Bielsko-Biala, WZIIZ, 1998, (praca dyplomowa inżynierska).
6. Dobija M., Smaga E. *Podstawy matematyki finansowej I ubezpieczeniowej*. Warszawa-Kraków, WN PWN, 1996.
7. Han J., Kamber M. *Data mining: methods and technique*, Morgan Kaufman, 2000.
8. Jelińska J. *Komputerowa analiza ubezpieczeń kredytów konsumpcyjnych*. //Bielsko-Biala, WZIIZ, 1999, (praca dyplomowa inżynierska).
9. Klenart M. *Komputerowa analiza ubezpieczeń z funduszem inwestycyjnym*. //Bielsko-Biala, WZIIZ, 1999, (praca dyplomowa inżynierska).
10. Kohut I. *Komputerowa analiza ubezpieczeń kredytów hipotecznych*. //Bielsko-Biala, WZIIZ, 1999, (praca dyplomowa inżynierska).
11. Kózka A. *Komputerowa analiza ubezpieczeń emerytalnych*. //Bielsko-Biala, WZIIZ, 1999, (praca dyplomowa inżynierska).
12. *Matematyka finansowa*. Warszawa, Bizant, 1994.
13. Przeradzki K. *Komputerowa analiza ubezpieczeń posagowych*. //Bielsko-Biala, WZIIZ, 1998, (praca dyplomowa inżynierska).
14. Quinlan J.R. *Induction of decision trees*. *Machine Learning*, 1:81-106, 1986.

УДК 683.1

А.М. Пелешин

НУ “Львівська політехніка”,

кафедра “Інформаційні системи та мережі”

ОПТИМІЗАЦІЯ РОЗМІЩЕННЯ ІНФОРМАЦІЙНИХ РЕСУРСІВ НА КОМПОНЕНТАХ WEB-СИСТЕМИ

© Пелешин А.М., 2001

This paper includes the research of the Web-system coordination problem proposed. Typical example of Web-system coordination are investigated. This example is optimization of query support location of global Web-system.

У статті розглядається проблема координації складної Web-системи. Досліджено типовий приклад координації - задачу оптимізації розміщення інформаційних ресурсів на компонентах Web-системи

Задача побудови інформаційних систем оптимальної фізичної структури досліджувалася (зокрема, у роботах [4,9,10,13,16,17]), проте з розвитком глобальної мережі Інтернет та Web-технологій виникли принципово нові види цієї задачі. У нових умовах побудова систем оптимальної структури здійснюється з врахуванням глобального, багаторівневого характеру ІС, відсутності єдиної власності на них, стохастичної поведінки користувачів такої системи. Ефективним математичним апаратом розв'язання таких задач є (як показано в роботах [8,2,14]) теорія координації ([1,5,6]).

У статті розглядається типовий приклад координації Web-систем. Проаналізовано задачу координації та побудовано алгоритм координації Web-систем. Застосування побудованого алгоритму до початку функціонування Web-системи дає можливість знаходити скоординоване розміщення ресурсів наперед. Запропонований алгоритм має розподілений характер, деякі його етапи виконуються на компонентах Web-системи. Таким чином, для виконання усього алгоритму на кожному з етапів достатньо інформації.

Побудований алгоритм дає можливість знаходити скоординоване розташування ресурсів до початку функціонування Web-системи. Таким чином, усуваються затрати на спроби розміщення неефективних ресурсів на компонентах. Такий вииграш є досить значним, оскільки проміжки часу на кожній ітерації, необхідні для визначення прибутку від розміщення ресурсу на кожному вузлі в режимі реального часу, є досить тривалими (від кількох діб до кількох місяців – залежно від інтенсивності надходження запитів користувачів та оперативності функціонування механізмів отримання прибутків від запиту). На тривалість ітерації можуть впливати такі часові фактори як: пора доби, робочі, вихідні та святкові дні, сезонні коливання тощо.

Розглянемо задачу оптимізації розміщення інформаційних ресурсів між компонентами Web-системи. Головним критерієм якості розміщення інформаційних ресурсів є прибуток, що отримується при зверненні до запитів Web-системи (із врахуванням затрат на підтримку запиту). Сама Web-система є результатом операції паралельного з'єднання компонент [7]. Web-системи нижчого рівня (компоненти) є формально незалежні, і верхній рівень Web-системи здійснює лише консультуючий вплив на компоненти. Цей вплив містить прогноз поведінки клієнта компонентів (тобто Web-системи верхнього рівня). Самі запити можуть розміщатися у багатьох компонентах одночасно.

Нехай:

$\{W_{S_j}\}_{j=1}^{N_{Part}}$ - множина компонент Web-системи;

$\{Q_i\}_{i=1}^{N_Q}$ - множина запитів до Web-системи.

$X_j = \{x_{ij}\}_{i=1}^{N_Q}$ - керуючі впливи j-ї компоненти, що відображають, які саме ресурси будуть на ній розміщені. Вектор X_j продукується компонентою як результат розв'язання локальної оптимізаційної задачі. Тоді $X = \{X_j\}$ - схема розміщення запитів на компонентах Web-системи,

$\mu(Q_i)$ - частота звернень клієнта до Q_i .

$\mu(Q_i, W_{S_j}, X)$ - частота звернень до Q_i , якщо він буде розміщений на W_{S_j} при схемі розміщення X .

Замість $\mu(Q_i, W_{S_j}, X)$ можна розглядати $\Pr(W_{S_j}/Q_i, X)$ - умовну імовірність звернення до W_{S_j} із запитом Q_i при схемі розміщення X . Тоді

$$\mu(Q_i, W_{S_j}, X) = \mu(Q_i) \Pr(W_{S_j}/Q_i, X). \quad (1)$$

Локальні функції мети компонент Web-системи мають такий вигляд:

$$G_j = \sum_{i=1}^{N_Q} x_{ij} (\mu(Q_i, W_{S_j}, X) \text{Prof}(Q_i, W_{S_j}) - \text{Stor}(Q_i, W_{S_j})) \quad (2)$$

де:

$\text{Prof}(Q_i, Ws_j)$ - прибуток j -ї компоненти Web-системи від звернення до запиту Q_i ,

що на ній розміщується;

$\text{Stor}(Q_i, Ws_j)$ - витрати j -ї компоненти на апаратну та адміністративну підтримку

Q_i у себе.

Мета оптимізації локальної компоненти:

$$G_j \rightarrow \max. \quad (3)$$

Глобальна функція мети є наперед визначена як сума локальних компонент:

$$G = \sum_{j=1}^{N_{part}} G_j \rightarrow \max. \quad (4)$$

На Web-систему визначаються глобальні обмеження такого виду:

$$\sum_{j=1}^{N_{part}} x_{ij} \geq 1 \quad \forall j \in \{1 \dots N_Q\}. \quad (5)$$

Тобто кожен запит повинен підтримуватися принаймні однією компонентою.

Застосуємо методи координації до знаходження оптимального розташування запитів між компонентами Web-системи. Параметрами, що координують компоненти Web-системи, є величини $\mu(Q_i, Ws_j, X)$ (або $\text{Pr}(Ws_j/Q_i, X)$) у випадку, якщо компоненти Web-системи володіють інформацією про актуальне значення $\mu(Q_i)$.

Координацію здійснюватимемо ітеративно. На кожній ітерації Web-система верхнього рівня генерує прогноз своєї поведінки ($\mu(Q_i, Ws_j, X)$ або $\text{Pr}(Ws_j/Q_i, X)$), що базується на актуальному розміщенні запитів. Виходячи з цього прогнозу, власники компонент Web-системи модифікують їх структури з метою максимізації прибутку від звернень до цих компонент. Верхній рівень Web-системи генерує новий прогноз та порівнює його на близькість до попереднього прогнозу (використовуючи статистичний підхід). Прогноз будується незалежно для кожної компоненти. Він передбачає для j -ї компоненти такі значення частот звернення до Q_i , які будуть отримані, якщо на ній буде розміщено підтримку цього запиту. При цьому використовується інформація про поведінку клієнта компонент Web-системи (тобто її верхнього рівня). Така інформація може отримуватися кількома різними шляхами.

Дослідження поведінки клієнта Web-системою нижнього рівня. У такому разі між ітераціями координації проходить певний проміжок часу, протягом якого проводиться таке дослідження. Такий підхід є найменш ефективним, проте найпростішим для застосування у глобальних системах при відсутності єдиних стандартів та правил опису профілю користувача.

Генерація координаційних параметрів безпосередньо клієнтом на основі відомих йому правил поведінки та подальша передача їх компонентам Web-системи. Такий підхід вимагає наявності стандартів та правил передачі такої інформації і тому його застосування у глобальних системах на сьогодні (до розроблення таких узгоджень) обмежене. Проте в intranet-системах використання цього підходу є цілком реальним.

Визначення конкретних значень координаційних параметрів на основі відомої моделі верхнього рівня Web-системи. Такий підхід застосовується у випадку, коли клієнт компонент Web-системи є комп'ютерною системою з відомими правилами функціонування.

Ітеративний алгоритм координації системи, що розглядається, наведений на рис. 1.



Рис. 1. Ітеративний алгоритм координації корисності Web-системи та її компонент.

Розв'язки локальних задач мають такий вигляд:

$$x_{ij} = \begin{cases} 1, \text{Prof}(Q_i, Ws_j) \mu(Q_i, Ws_j, X) \geq \text{Stor}(Q_i, Ws_j) \\ 0, \text{Prof}(Q_i, Ws_j) \mu(Q_i, Ws_j, X) < \text{Stor}(Q_i, Ws_j) \end{cases} \quad (6)$$

Глобальні обмеження є узгодженими з локальними задачами і не перешкоджають процесу координації, якщо є істинною така умова:

$$\forall i \exists j : \mu(Q_i) \text{Prof}(Q_i, Ws_j) - \text{Stor}(Q_i, Ws_j) \geq 0. \quad (7)$$

Дійсно, тоді для будь-якого i знайдеться хоча б одна компонента W_{S_j} , де при $\Pr(W_{S_j}/Q_i, X)=1$ доцільно розмістити запит Q_i . Ця умова означає, що для кожного запиту повинна існувати хоча б одна компонента, для якої в ідеальних умовах запит був би незбитковим. Якщо вказана умова не виконується, то для досягнення координованості Web-системи потрібно усувати нерентабельні запити або зменшувати витрати на їхню підтримку.

У загальному випадку локальні та глобальна цільові функції можуть не мати властивості обмеженої узгодженості і, як наслідок, розглянута модель задачі координації може не мати розв'язку.

Зміст відсутності розв'язку задачі координації такий - координація системи не завжди можлива без наявності впливів, що мають характер розпоряджень верхнього рівня компонентам нижнього рівня. Вплив через прогнозування може бути недостатнім для координації Web-системи. Точніше, прогнозування є достатнім для координації Web-системи за умови обмеженої узгодженості, яка у розглянутому випадку може не існувати. Тому для досягнення координованості у Web-системі її верхній рівень повинен генерувати розпорядження для нижчого рівня. Модифікація Web-системи, у якій передбачено такий вплив, може реалізовуватися, якщо:

- Web-система має одного власника (наприклад, intranet-система);
- власники компонент Web-системи делегують певні повноваження верхньому рівню і, відповідно, обмежують свою свободу (наприклад, підписання договору про координацію дій в INTERNET).

У моделі Web-системи такі зміни схеми взаємодії відображаються як модифікація локальних функцій якості, яка передбачає розпорядчий вплив верхнього рівня.

У багатьох випадках задачу координації Web-системи, що розглядається, можна спростувати. У такому разі верхній рівень використовується лише для прогнозування поведінки. Власних інтересів (і, відповідно, окремої оптимізаційної задачі) він не має. Метою координації є досягнення стабільного стану, в якому усі компоненти Web-системи розміщали б у себе вигідні їм запити. Критерієм досягнення оптимального стану Web-системи залишається близькість прогнозних значень поведінки верхнього рівня до її фактичних значень. Спрощені задачі координації є характерні для сучасних Web-систем, що функціонують у глобальних інформаційних мережах. Верхній рівень Web-системи є користувачем, якого не цікавлять прибутки компонент Web-системи. Координаційні параметри отримуються компонентами шляхом моніторингу поведінки верхнього рівня.

1. Алиев Р.А., Либерзон М.И. Методы и алгоритмы координации в промышленных системах управления. - М.: Радио и связь, 1987.- 208с. 2. Буров Є.В., Пелецишин А.М. Аналіз та оптимізація інформаційних потоків у сучасних internet та intranet системах. // Вісник Державного Університету "Львівська Політехніка", Львів, 1998, №330.- С.27-34. 3. Буров Є.В., Пелецишин А.М. Оптимізація розміщення даних у Web-системах. // Вісник ДУ "Львівська Політехніка", Львів, 1998, №330.- С.17-27. 4. Йеркса Г. Механизмы, балансирующие нагрузку Web-узлов. // Сети и системы связи, 1999, №5.- С.120-126. 5. Месарович М., Такахара Я. Общая теория систем: Математические основы.- М.:Мир, 1978.- 310с. 6. Месарович М., Мако Д., Такахара Я. Теория многоуровневых иерархических систем. - М.:Мир, 1973. 7. Пелецишин А.М., Буров Є.В. Принципи формального моделювання інформаційних систем на базі Web-технологій. // Вісник ДУ "Львівська Політехніка", Львів, 1999, №383.- С.178-201. Пелецишин А.М. Методи та алгоритми оптимізації Web-систем.// Вісник ДУ "Львівська політехніка", Львів, 2000, 406.- С.199-211 8. Хэррингшоу К. Оптимизация сервера Web. //LAN/журнал сетевых решений, 1997, №8, <http://www.osp.ru/lan/1997/08/111.htm> 9. Apers P. Data allocation in distributed database systems. //ACM Trans. on Database Systems, September 1988, vol.13,№3, pp.263-304 10. Billsus D., Pazzani M. Learning probabilistic user models. In workshop notes of Machine Learning for User Modeling, Sixth International Conference on User Modeling, Chia Laguna, Sardinia, 2-5

June 1997, 6p 11. Gillet S., Kapor M. *Self-governing Internet: Coordination by Design*. Massachusetts Institute of Technology. Center for Coordination Science. Technical Report. 1997., 25p., <http://ccs.mit.edu/CCSWP197/CCSWP197.html> 12. March S., Rho S.. *Allocating Data and Operations to Nodes and Distributed Database Design// IEEE Transactions on Knowledge and Data Engineering*, Vol.77 No. 2, April 1995, pp.305-318 13. Miettinen K., Korhonen P. *Distributed Performance Analysis on the Internet Using a Centric Database*. //International Institute for Applied Systems Analysis. INTERIM REPORT IR-98-111. December 1998, 15p. <http://www.iiasa.ac.at/Publications/Documents/IR-98-111.pdf>. 14. Triantafillou P., Taylor D. *The Location-Based Paradigm for Replication: Achieving Efficiency and Availability in Distributed Systems*. On Software Engineering, Vol.21, No 1, January 1995, pp1-18 15. Wolfson O., Jajodia S. *An algorithm for dynamic data allocation in distributed systems*. //Information Processing Letters, 1995, Vol. 53, No. 2, pp. 113--119. 16. Woodside C., Tripathi S.. *Optimal allocations of File Servers in a Local Network Environment*. //IEEE Transactional Software Engineeing, 1986 17. Zaane O. *Resource and knowledge discovery from the internet and multimedia repositories*. //PhD Theses. Simon Fraser University, 1999 304 p

УДК 683.1

А.М. Пелещин, Т.Б. Гулка

НУ “Львівська політехніка”,
кафедра “Інформаційні системи та мережі”

ІНФОРМАЦІЙНА СИСТЕМА АНАЛІЗУ ДІЯЛЬНОСТІ WEB-ВУЗЛА

© Пелещин А.М., Гулка Т.Б., 2002

The article reads about the main approachesto the building of systems of analysis of Inter site usage based on access books

Розглянуто основні підходи до побудови системи аналізу використання Інтернет-сайту, базованої на журналах доступу.

Розвиток Інтернету постійно породжує нові галузі досліджень. Останнім часом актуальними стали дослідження, пов'язані із визначенням узагальнених характеристик відвідувачів web-сайту. Мета таких досліджень очевидна: знаючи вподобання та характеристики відвідувача, можна робити цілеспрямовану рекламну кампанію, обновляти чи модифікувати Web-вузол, проводити оптимізацію фізичної структури Web-системи. Для такого аналізу користуються декількома взаємодоповнюючими методами. Один з них застосовується компаніями, які при наданні своїх послуг в Інтернеті вимагають заповнити анкети відвідувача. В анкетах потрібно вказати свої дані, такі як вік, стать, професія, хобі, місце проживання і т.д. Недоліки такого методу очевидні:

- більшість сайтів не пропонують ніяких послуг в онлайні і тому не можуть вимагати заповнення анкети;
- перевірити, чи правильно відвідувач ввів інформацію про себе неможливо.

Про те є і переваги: якщо з певною ймовірністю вважати певний відсоток інформації достовірним, то можна дізнатися деякі дані, що іншим способом зробити не можливо (наприклад вік, стать, професія).

Інший метод базується на аналізі журналів доступу до сайту. Такі журнали ведуть всі популярні Web-сервери. Крім того більшість адміністраторів сайтів ведуть свої журнали. Журнали формуються шляхом читання значень змінних середовища CGI, за допомогою CGI-скриптів. При запуску на сервері CGI-скрипта сервер формує середовище оточення, в якому скрипт може знайти всю доступну інформацію про HTTP з'єднання і запит. Таким