

УДК 004.658.3(045)

І.А. ЖУКОВ, І.М. КРАВЕЦЬ

*Інститут комп'ютерних технологій Національного авіаційного університету, Україна***ОРГАНІЗАЦІЯ РОЗПОДІЛЕННЯ НАВАНТАЖЕННЯ БАЗ ДАНИХ
В ІНФОРМАЦІЙНО-АНАЛІТИЧНІЙ СИСТЕМІ**

Розроблено розподілену базу даних для інформаційно-аналітичної системи Міністерства транспорту і зв'язку України. Для покращення керування системою розподілених баз даних інформаційно-аналітичної системи було використано агентів та мультиагентні системи. Також, проведено тестування стандартної Web-системи та розподіленої бази даних інформаційно-аналітичної системи. Експериментальні результати розподіленої бази даних інформаційно-аналітичної системи довели ефективність застосування «еволюційного алгоритму фрагментації й розподілу».

Ключові слова: база даних, розподілення навантаження, мультиагенти, «еволюційний алгоритм фрагментації й розподілу».

Вступ

В інформаційно-аналітичній системі (ІАС) зв'язок між складовими розподіленої бази даних (РБД) є функціональним. РБД забезпечує збереження і виконання функцій управління даними в декількох вузлах і передачу даних між цими вузлами в процесі виконання запитів. Розбиття даних в РБД досягається через збереження різних таблиць або навіть різних частин і фрагментів однієї таблиці на різних комп'ютерах. Для користувача (або прикладної програми) не має значення, яким чином розподілені дані між комп'ютерами.

Не дивлячись на те, що РБД складається з декількох локальних БД, у користувачів зберігається ілюзія роботи з централізованою БД. Ця вимога прозорості роботи з БД вимагає вироблення узагальненого представлення про дані – глобальної концептуальної схеми. Визначення даних в такій схемі є аналогічним визначенню в централізованій БД. Відмінності починаються, коли вимагається зберігати дані в декількох вузлах. Щоб провести розбивку даних, було виконано секціювання таблиці глобальної схеми на фрагменти. Існує два типи секціювання: горизонтальне і вертикальне. При секціюванні таблиці по рядках виконується горизонтальне секціювання, при розбитті по стовпцях – вертикальне.

РБД ІАС відповідає таким властивостям або якостям ідеальної РБД:

1. *Локальна автономія.* Вона означає, що управління даними в кожному вузлі виконується локально і незалежно від інших вузлів системи.

2. *Незалежність вузлів.* Вважається, що в ідеальній системі всі вузли рівноправні і незалежні, а БД є рівноправними постачальниками інформації в

загальний інформаційний простір.

3. *Неперервність операцій.* Дані доступні завжди, а операції над ними проводяться неперервно.

4. *Прозорість розміщення даних.* Користувач не мусить знати де розміщені дані. Під час роботи створюється враження, що дані знаходяться саме на його комп'ютері.

5. *Прозора фрагментація.* Ця властивість трактується, як можливість створення фізично розподілених даних, які логічно утворюють єдине ціле. Допускається горизонтальна та вертикальна фрагментація.

6. *Прозорість тиражування.* Забезпечує тиражування (перенос змін) об'єктів первинної БД в усі вузли її розміщення внутрішньо-системними засобами.

7. *Обробка розподілених запитів.* Означає виконання операцій, сформованих, в рамках звичайного запиту на мові SQL.

8. *Обробка розподілених транзакцій.* Забезпечує виконання операцій з одночасним забезпеченням цілісності і узгодженості даних, шляхом використання двофазового протоколу фіксації транзакцій.

9. *Незалежність від обладнання.* Для оснащення вузла можуть використовуватися комп'ютери різних марок і виробників.

10. *Незалежність від операційних систем.* Передбачає допустимість взаємодії різноманітних операційних систем у різних вузлах розміщення розподіленої бази даних.

11. *Прозорість мережі.* Забезпечує будь-які протоколи в локальній обчислювальній мережі, яка обслуговує розподілену базу даних.

12. *Незалежність від типу баз даних.* Допускає співіснування різних систем керування базами даних.

Таким властивостям відповідає РБД ІАС Міністерства транспорту і зв'язку України [1]. За ос-

нову розподілення навантаження було взято еволюційний алгоритм фрагментації й розподілу БД.

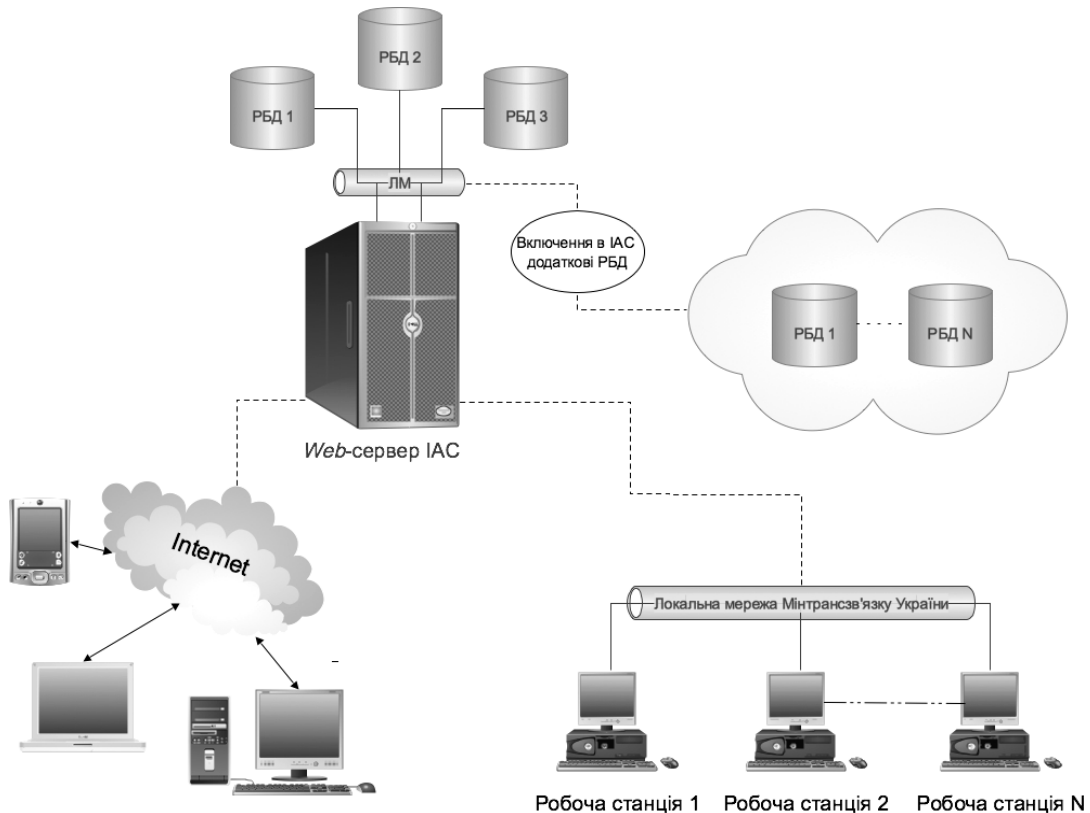


Рис. 1. Схема організації ІАС

1. Принципи функціонування ІАС

В ІАС реалізуються наступні основні принципи функціонування:

- наскрізні запити (пошук інформації по одному запиту у всіх БД, підключених до системи);
- пакетне оброблення запитів: на відміну від режиму виконання запитів online (інтерактивної обробки, безпосередньої взаємодії з користувачем), при пакетній обробці запити що ставляться в чергу на виконання й виконуються відповідно до їх пріоритетів;
- збереження результатів виконання запитів;
- контроль прав доступу користувачів до програмних компонентів ІАС, БД і результатам виконання запитів, захист від несанкціонованого доступу;
- автоматична генерація інтерфейсу користувача залежно від прав на отримання інформації з БД.

Підключення до ІАС нових БД, розробка запитів і звітів (форм подання результатів виконання запитів) здійснюється без написання програмного коду шляхом параметричного налаштування програмних компонентів комплексу. ІАС дозволяє використовувати в якості джерела даних РБД різних

СУБД (Oracle, Microsoft SQL, DB2, Informix й ін.). В основі реалізації виконання запитів до БД лежить автоматичне формування й виконання виражень SELECT мови SQL [2].

До складу комплексу ІАС входять програмні компоненти, що забезпечують виконання різних функцій по адмініструванню інформаційно-пошукових систем:

- керування доступністю БД для обробки запитів;
- керування користувачами, їхніми повноваженнями на одержання інформації із прикладних БД;
- моніторинг виконання запитів;
- формування звітів по статистиці запитів у розрізі відомств, підрозділів, співробітників, баз даних.

Комплекс ІАС реалізований в архітектурі клієнт-сервер із застосуванням Web-технологій й СУБД Oracle. Як Web-сервер, використовується Apache HTTP Server + nginx. Для доступу до програмних компонентів комплексу використовується Інтернет-браузер. На рис. 1 представлена схема організації ІАС, а на рис. 2 – функціональна схема ІАС.

Розглянута архітектура ІАС забезпечує високу масштабованість системи й оптимізацію її роботи.

2. Агенти і МАС

Для покращення керування системою РБД ІАС використовуються агенти і мультиагентні системи (МАС). Істотними в понятті агента є поняття делегування й автономії. Це означає обмежене безпосереднє керування й значний плюралізм (розмаїтість) у системах, у яких працюють відразу кілька агентів. У МАС агенти мають свої власні цілі, які можуть вступати в протиріччя із цілями інших агентів. Проте, група логічно децентралізованих агентів вирішує працювати разом для досягнення загальної мети. Це все було досягнуто із розподілом завдання на частини між агентами. Також, у комплексі ІАС, агенти мають можливість вирішувати вихідне завдання своїм способом. У випадку пошуку наближених рішень потім може бути обране найкраще. Або може застосовуватися процедура голосування агентів за те або інше рішення.

Оброблення запитів в ІАС відбувається на основі передачі і прийому повідомлень. Користувач відправляє запит через *Web*-сервер, далі агенти трансляції і пошуку інформації [1] відправляють запити декільком серверам об'єктів (БД). А уже сервер об'єктів посилає клієнтові відповідь на запит.

Парадигма мультиагентного проектування й реалізації систем полягає в тому, що програмні агенти для досягнення мети (виконання деякої роботи) переміщуються з одного сервера на інший. Агенти виконують свою роботу локально на тому сервері, на якому вони в цей момент перебувають. Обмін повідомленнями (між серверами) по мережі агенти, як правило, не використовують [3 – 5].

Модель агентів важлива, наприклад, для обробки транзакцій в ІАС. Як відомо, транзакція – це послідовність взаємозалежних дій, які повинні бути виконані на декількох РБД [6 – 8].

3. Експериментальні результати

Для тестування ІАС було взято два різних комплекси: перший (К1) – це один *Web*-сервер на якому знаходився програмний комплекс і БД ІАС, а другий (К2) – це *Web*-сервер з програмним комплексом та три РБД, де дані були розподілені "еволюційним алгоритмом" [1]. Для аналізу швидкодії і визначення стабільної роботи обох *Web*-серверів використовувався програмний комплекс "*ab - Apache HTTP server benchmarking tool*". А для побудови графіків на основі вихідних даних від "*ab*" – *Gnuplot*. Кількість запитів до *Web*-сервера при тестуванні (*-n requests*) за один сеанс – 1000, а кількість одночасних (паралельних) запитів в один і той же момент часу (*-c concurrency*) – 100. Тобто, нас цікавить як будуть поводити себе сервера, коли в один і той

же момент часу запити будуть виконувати 100 користувачів системи ІАС і наскільки ефективним було розподілення БД.

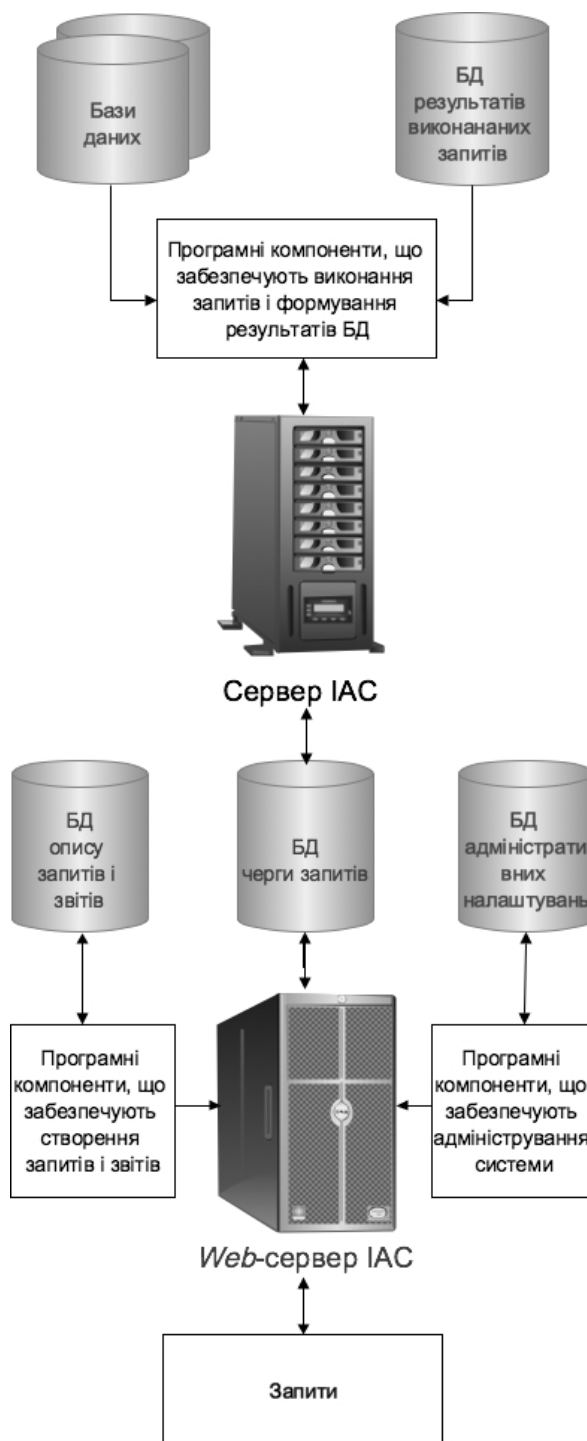


Рис. 2. Функціональна схема ІАС

3.1 Тестування К1

Запуск "*ab*" відбувся із командою: "*ab -c 100 -n 1000 -g k1.dat http://k1.ias.local/index.html*". Після завершення 1000 запитів отримали статистичні дані (табл. 1).

Таблиця 1
Статистичні дані при тестуванні K1

Параметр	Значення
К-ть одночасних запитів	100
Загальний час виконання тесту, сек.	15.714629
К-ть запитів на сервер за один сеанс	1000
К-ть помилкових відповідей від сервера	13
К-ть успішних відповідей від сервера	987
Середня к-ть запитів за 1 сек.	63.63
Середній затрачений час на один запит, мс.	1571.463
Середній затрачений час при всіх одночасних, мс.	15.715

На основі даних результатів зроблено висновки, що всі 1000 запитів за один сеанс виконались

успішно. На виході із програмного комплексу тестувань "ab" отримали *k1.dat* файл із більш детальними показниками. За допомогою командної строки та утиліти "Gnuplot" побудуємо графік результатів (рис. 3):

```
x-man:bin xman_apple$ gnuplot
set output "http_k1_benchmark.png"
set xlabel "Request to IAS K1"
set ylabel "ms"
plot "k1.dat" using 7 with lines title "ctime", \
"k1.dat" using 8 with lines title "dtime", \
"k1.dat" using 9 with lines title "ttime", \
"k1.dat" using 10 with lines title "wait"
```

3.2 Тестування K2

Запуск "ab" відбувся із командою: "ab -c 100 -n 1000 -g k2.dat http://k2.ias.local/index.html". Після завершення 1000 запитів отримали статистичні дані (табл. 2).

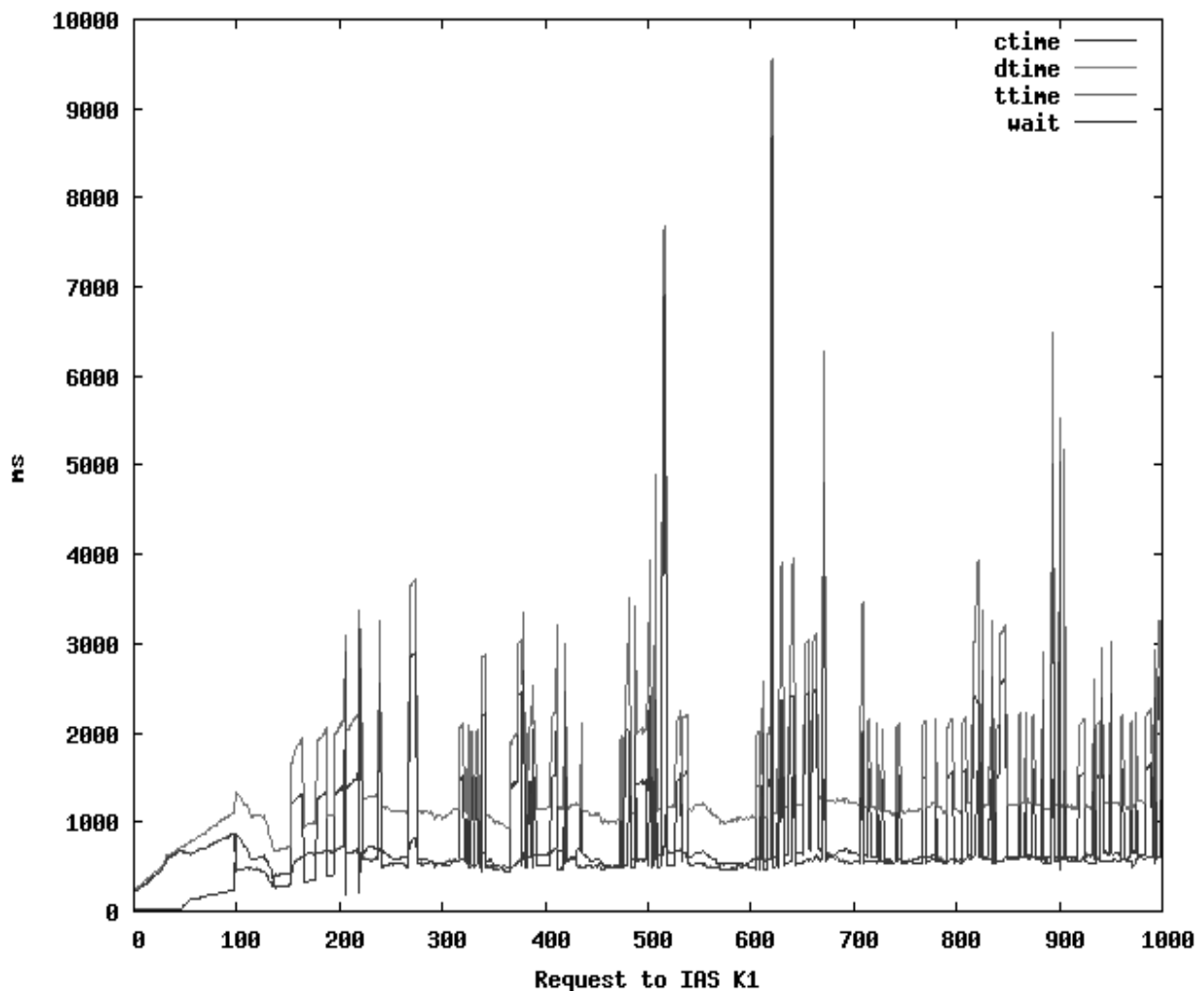


Рис. 3. Результати оброблення запитів комплексом K1 для ІАС

Таблиця 2
Статистичні дані при тестуванні K2

Параметр	Значення
К-ть одночасних запитів	100
Загальний час виконання тесту, сек.	0.364127
К-ть запитів на сервер за один сеанс	1000
К-ть помилкових відповідей від сервера	0
К-ть успішних відповідей від сервера	1000
Середня к-ть запитів за 1 сек.	2746.29
Середній затрачений час на один запит, мс.	36.413
Середній затрачений час при всіх одночасних, мс.	0.364

На основі даних результатів зроблено висновки, що всі 1000 запитів за один сеанс виконались успішно.

На виході із програмного комплексу тестувань "ab" отримали *k2.dat* файл із більш детальними показниками.

За допомогою командної строки та утиліти "Gnuplot" побудуємо графік результатів (рис. 4):

```
x-man:bin xman_apple$ gnuplot
set output "http_k2_benchmark.png"
set xlabel "Request to IAS K2"
set ylabel "ms"
plot "k2.dat" using 7 with lines title "ctime", \
"k2.dat" using 8 with lines title "dtime", \
"k2.dat" using 9 with lines title "ttime", \
"k2.dat" using 10 with lines title "wait"
```

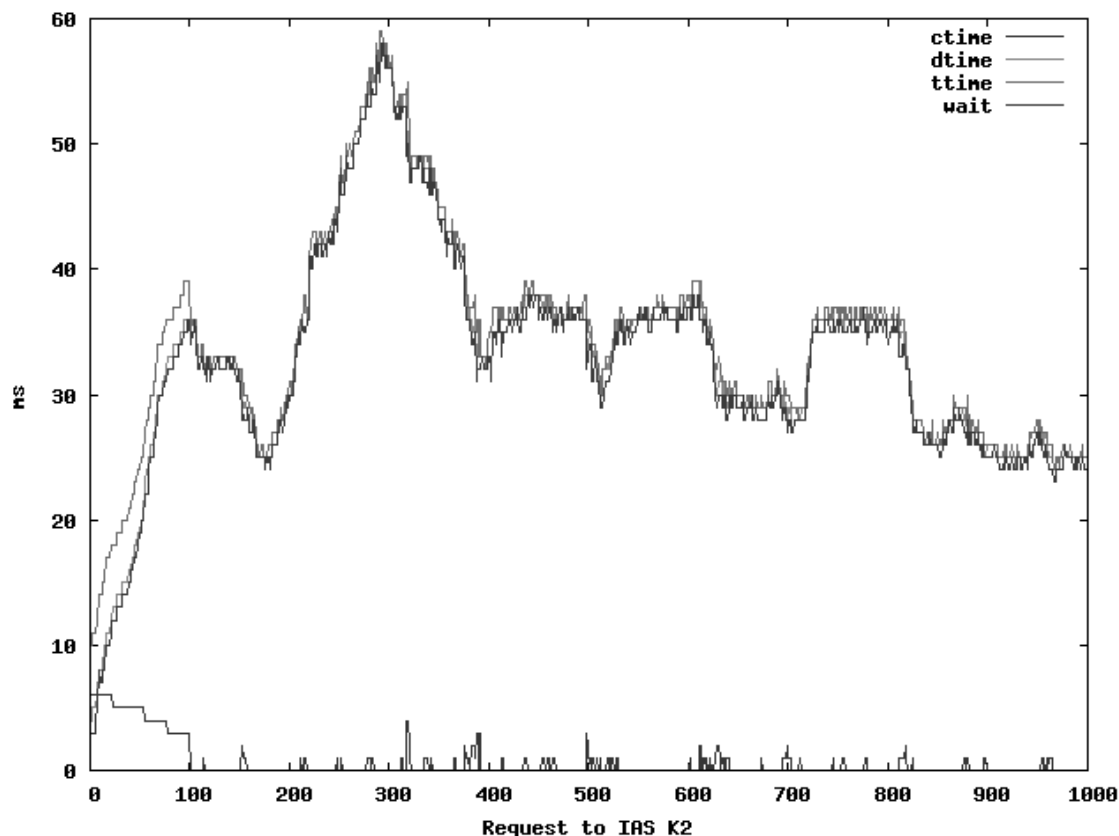


Рис. 4. Результати оброблення запитів комплексом K2 для ІАС

Висновки

Впровадження в ІАС Мінтрансв'язку України "еволюційного алгоритму фрагментації й розподілу даних" дало змогу зменшити навантаження на БД, а також розподілити дані по тим БД і частинам, які використовуються найчастіше. Комплекс K2 виявився не тільки стабільнішим, а й швидшим майже у 44 рази за K1, що дає йому змогу виконати на 2683 запитів більше за одну секунду.

Також слід відзначити пікові значення на обох графіках (рис. 3, 4). У K1 спостерігаємо у більшості випадків однакові показники виконаних запитів, а ось у K2 – з подальшою кількістю запитів час виконання зменшується. Це було досягнуто завдяки кешуванню і збереженню результатів в окрему БД.

Мобільні агенти підтримують асинхронні й автономні операції. Сервери можуть запускати мобільних агентів, які незалежно подорожують між серверами, попутно виконуючи різні операції.

Мобільний агент може виконувати алгоритми, що містять різні політики балансування завантаження й ухвалювати рішення щодо розподілу завантаження на “лету”, відповідно до поточних станів серверів.

Література

1. Жуков І.А. Розподілення навантаження баз даних в інформаційно-аналітичній системі / І.А. Жуков, І.М. Кравець // Проблеми інформатизації та управління: Збірник наукових праць.– К.: НАУ, 2007. – Вип. 4 (22). – С. 56-61.

2. Жуков І.А. Методи балансування навантаження для Web-серверів / І.А. Жуков, І.М. Кравець // Проблеми інформатизації та управління: Збірник наукових праць.– К.: НАУ, 2007. – Вип. 3 (21). – С. 46-54.

3. Танненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Танненбаум, М. Ван Стеен. – СПб.: Питер, 2003. – 877 с.

4. Tel G. Introduction to Distributed Algorithms / G. Tel. – Cambridge University Press, 2004. – 534 p.

5. Coulouris G. Distributed Systems. Concepts and Design / G. Coulouris, J. Dollimore, T. Kindberg. – Addison-Wesley Publishing Company, 1995. – 644 p.

6. Kwok Yu-Kwong. A new fuzzy-decision based load balancing system for distributed object computing / Yu-Kwong Kwo, Lap-Sun Cheung // Journal of Parallel and Distributed Computing. – 2004. – 64. – P. 238-253.

7. Baala H. A self-stabilizing distributed algorithm for spanning tree construction in wireless ad hoc networks / H. Baala, O. Flauzac, J. Gaber, M. Bui, T. El-Ghazawi // Journal of Parallel and Distributed Computing. – 2003. – 63. – P. 97-104.

8. Cao Jiannong. Das Scalable load balancing on distributed web servers using mobile agents / Jiannong Cao, Yudong Sun, Xianbin Wang, K. Sajal // Journal of Parallel and Distributed Computing. – 2003. – 63. – P. 996-1005.

Надійшла до редакції 23.02.2009

Рецензент: д-р техн. наук, проф., проф. кафедри спеціалізованих комп'ютерних систем В.Г. Зайцев, Національний технічний університет України «КПІ», Київ, Україна.

ОРГАНИЗАЦИЯ РАСПРЕДЕЛЕНИЯ НАГРУЗКИ БАЗ ДАННЫХ В ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ СИСТЕМЕ

И.А. Жуков, И.М. Кравец

Разработана распределенная база данных для информационно-аналитической системы Министерства транспорта и связи Украины. Для улучшения управления системой распределенных баз данных информационно-аналитической системы были использованы агенты и мультиагентные системы. Также проведено тестирование стандартной Web-системы и распределенной базы данных информационно-аналитической системы. Экспериментальные результаты распределенной базы данных информационно-аналитической системы доказали эффективность применения «эволюционного алгоритма фрагментации и распределения»

Ключевые слова: база данных, распределение нагрузки, мультиагенты, «эволюционный алгоритм фрагментации и распределения».

ORGANIZATION OF DISTRIBUTION LOAD DATABASE IN THE ANALYSIS AND INFORMATION SYSTEM

I.A. Zhukov, I.M. Kravets

A distributed database for data-processing system of the Ministry of Transport and Communications of Ukraine was developed. To improve management of distributed databases of information and analytical system agents and multiagents systems were used. Also a standard Web- system and a distributed database of information and analytical system testing were performed. The experimental results of distributed databases of information and analytical system proved effective to use «evolutionary algorithm of fragmentation and distribution».

Keywords: database, distribution of load, multiagents, «evolutionary algorithm of fragmentation and distribution»

Жуков Ігор Анатолійович – д-р техн. наук, проф., директор інституту комп'ютерних технологій Національного авіаційного університету, Київ, Україна, e-mail: zhukov@nau.edu.ua.

Кравець Іван Михайлович – аспірант кафедри комп'ютерних систем та мереж Національного авіаційного університету, Київ, Україна, e-mail: me@ikravets.com.