УДК 681.3

Л.О. Березко, А.І. Якимець
Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

ЕФЕКТИВНИЙ МЕТОД ОБРОБКИ ЗАПИТІВ ДО ВЕБ-СЕРВІСІВ

Розглянуто питання підвищення ефективності обробки потоку запитів до веб-сервісу. Проаналізовано можливі методи та запропоновано реалізацію на основі диспетчерського підходу.

Ключові слова: розподіл навантаження, балансування навантаження, веб-сервери.

In the article the issue of increasing the efficiency of processing flow requests to the Web server. Possible methods and implementation based on the proposed control approach.

Key words: balancing distribution, load balancing, web-servers.

Вступ

На сучасному етапі розвитку інформаційних технологій веб-сервери є чи не найважливішими елементами веб-систем, що функціонують як в мережі Internet, так і в мережах Intranet, а їх швидкодія є визначальним параметром. Зростання кількості веб-сервісів та користувачів цих сервісів вимагає значного підвищення швидкості реакції веб-серверів на запити користувачів. Як правило, один веб-сервер не спроможний опрацювати великий потік запитів за прийнятний період часу, тому актуальним є питання розподілу запитів до деякого веб-сервісу між декількома веб-серверами [1–4].

Стан проблеми

Особливістю інформаційних систем із клієнт-серверною архітектурою, що функціонують в сучасних мережах, є генерація великої кількості повідомлень, що передаються мережами. Довільний запит користувача на виконання конкретної задачі веб-сервісом може викликати створення та надсилання десятків запитів та відповідей до інших веб-сервісів. Одним із підходів до підвищення їх продуктивності є збільшення кількості веб-серверів, які опрацьовують запити, згенеровані користувачами, тобто клієнтами, до цього веб-сервісу [5–9].

Клієнт повинен бачити веб-сервіс, до якого звертається, як єдине ціле. Для цього на декількох веб-серверах встановлюється ідентичне програмне забезпечення, а маршрутизатор надсилає запит від клієнта до одного із серверів, який вибирають за певними ознаками. На однакові запити отримують однакові відповіді.

Розподіл запитів між веб-серверами допомагає уникнути ситуації, коли потік запитів викликає таке навантаження на окремий сервер, що можлива швидкість обробки запитів та наявні системні ресурси потребуватимуть побудови черги.

Виникає задача балансування навантаження, тобто визначення того веб-сервера з декількох, на який треба саме в цей момент відправити запит для найшвидшої реакції на нього. Розглянемо можливі варіанти її розв’язання.

1. Метод Round-robin. Суть цього методу у виборі варіантів за коловим циклом. Балансувальник навантаження відсилає запити до кожного з веб-серверів за коловою чергою, незалежно від його завантаженості в момент відсилання запиту, що є суттєвим недоліком такого методу.

2. Вибір сервера клієнтом. Клієнти можуть вибирати один із доступних веб-серверів або випадково, або використовуючи механізми інтелектуального вибору. Основним недоліком методу є великі часові затримки, потрібні для визначення стану веб-серверів під час кожного звертання.

3. Використання механізму дворівневої диспетчеризації. Спочатку DNS-сервер визначає, на який веб-сервер направлено запит клієнта. Після отримання запиту кожний такий веб-сервер може переспрямовувати запит на інший веб-сервер. Такий децентралізований метод балансування навантаження реалізовано у веб-сервері Apache. Основним недоліком є велика часова затримка під час перенаправлення запитів між веб-серверами. Є небезпека виникнення нескінченного циклу.

Постановка задачі

Пропонується підвищити швидкість обробки запитів до веб-сервісів, вдосконалюючи розподіл запитів між веб-серверами, ґрунтуючись на методі диспетчеризації. Диспетчеризація є централізованим методом балансування, який передбачає використання агента-диспетчера, якому повідомляються системні значення завантаженості веб-серверів та який ініціює надсилання до них запитів [3].

Розв'язок задачі

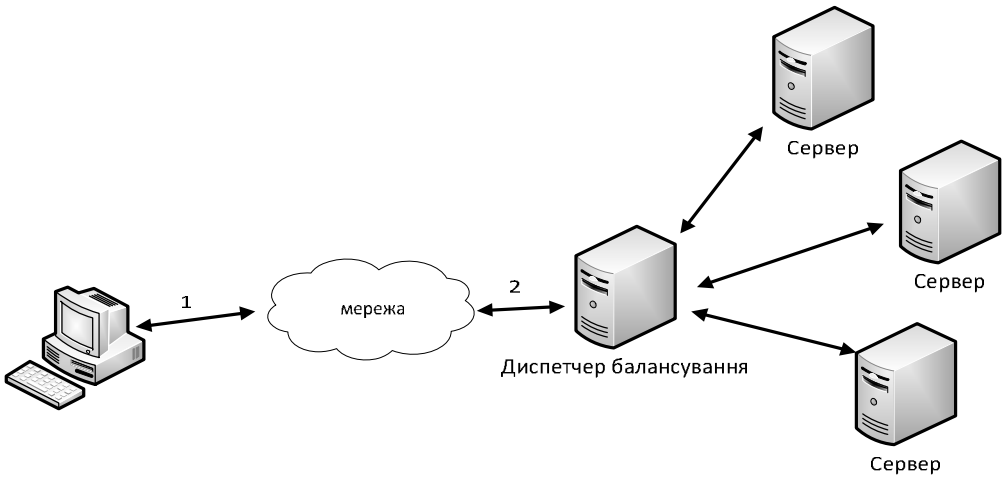
Диспетчер балансування навантажень отримує запити від клієнтів, надіслані до єдиної віртуальної IP-адреси, та виконує їх розподіл між реальними IP-адресами веб-серверів, для чого виконує такі дії.

1. Перевірка працездатності веб-серверів.
2. Періодичне визначення часу відповіді на запити.
3. Визначення ступеня використання перепускної здатності кожного веб-сервера.
4. Визначення кількості активних звертань на кожному сервері.

Диспетчер робить вибір, на який з веб-серверів буде передано запит користувача для здійснення авторизації, враховуючи кількість користувачів, які в цей час вже авторизовані, та час відповіді на запити. Кількість робочих процесів при цьому не враховується. Як тільки користувач авторизується на певному сервері, сесія користувача прив'язується до цього сервера. Таке балансування навантажень на веб-сервери може здійснювати або програма, що виконується на комп'ютері-диспетчері, або спеціальний пристрій. Використання комп'ютера зі стандартною операційною системою, такою як Windows або UNIX, дає переваги, оскільки в аварійних ситуаціях його можна замінити іншою машиною зі схожими характеристиками. Використання окремого пристрою дає перевагу в швидкості, оскільки в ньому використовується процесор, оптимізований під конкретне завдання.

Зауважимо, що єдина точка доступу до диспетчера балансування є тим місцем, аварійна ситуація в якому виведе з ладу всю побудовану систему. Тому доцільно використовувати "гарячий резерв" – комп'ютер, який конфігуровано як дзеркало комп'ютера-диспетчера і який не використовується до виникнення виняткових ситуацій. Активний та дублюючий диспетчери періодично надсилають повідомлення на IP-адресу 225.0.0.2 для контролю їх працездатності. Це UDP пакети у форматі VRRP (Virtual Router Redundancy Protocol), які попадуть на кінцевий пункт призначення за 200–300 мілісекунд. Дублюючий диспетчер переймає повноваження активного, коли не отримує повідомлення про працездатність від активного диспетчера.

На рисунку показано загальну схему диспетчерського методу. Диспетчер постійно опитує сервери, які мають з ним зв'язок. Опитування виконується протоколом UDP на чітко визначеному порті. Диспетчер надсилає запит до кожного сервера про його стан, на що сервер відповідає інформацією про своє завантаження. Зважаючи на це, диспетчер вирішує, якому із серверів передати на виконання запит користувача.



Диспетчерський метод балансування навантаження

Було створено веб-сервер, в якому програмно реалізовано запропонований метод обробки потоку запитів до веб-сервісу, з такими основними характеристиками:

– опрацьовує запити GET, POST в стандартах HTTP 1.0 та 1.1;

– працює з форматами html, htm, jpeg, png, gif, css, js;

– опрацьовує запити з необхідністю стискання даних перед відправкою;

– забезпечує передачу ідентифікатора сесії через cookies;

– зберігає дані в межах сесії;

– за відсутності можливості обробки запитів до сервісу пересилає їх на інші веб-сервери, що є в списку дочірніх.

Диспетчер балансування працює так:

– реалізується режим пересилання запитів до різних серверів, які містять ідентичний контент;

– пересилання здійснюється за стратегією: server1: x%; server2: y%; server3: 100-x%-y%;

– пересилання здійснюються у межах сесії .

Головною особливістю такої реалізації диспетчерського підходу є відсутність великих черг запитів на диспетчері балансування навантаження.

Висновки

Актуальність вибраної теми в наш час очевидна, оскільки постійне зростання кількості робочих станцій та веб-сервісів збільшує навантаження на мережі та програмно-апаратні засоби. Відповідно потрібно вдосконалювати наявні та створювати нові механізми забезпечення розподілу ресурсів мережі для ефективної роботи інформаційних систем. Запропонований підхід до створення веб-серверів актуальний для організацій, яким потрібно опрацьовувати велику кількість запитів для вирішення однотипних завдань. Для ефективного функціонування необхідна наявність мінімум двох серверів однотипної конфігурації. Реалізація можлива на комп'ютерах порівняно невеликої потужності.

1. Моисеев Т.Н., Распределение информационных потоков данных в распределенных многосерверных системах / Моисеев Т.Н. – Воронеж: Научная книга, 2005. – 145 с.2. Рогачко Е.С. Динамическое распределение нагрузки в сетях массового обслуживания / Рогачко Е.С. – Саратов, 2007. – 102 с. 3. Французов Д. Оценка производительности вычислительных систем. Открытые системы / Французов Д. – М., 1996. – С. 58–66. 4. Таненбаум Е. Сучасні комп'ютерні системи. – СПб.: Питер, 2008. 5. Coulouris G., Dollimore J., Kindberg T. Distributed Systems. Concepts and Design. - Addison-Wesley Publishing Company, 1995. 6. Leinberger W., Karypis G., Kumar V., Biswas R. Load balancing across nearhomogeneous multi-resource servers // 9-th Heterogeneous Computing Workshop. – 2000. – P. 60–71. 7. Tony Bourke: Server Load Balancing - O'Reilly, ISBN 0-596-00050-28. Heiss H.-U., Schmitz M. Decentralized dynamic load balancing: The particles approach// Information Sciences. – 1995. – № 84. – P. 115–128.9. Subrata R., Zomaya A.Y., Landfeldt B. Artificial life techniques for load balancing incomputational Grids // Journal of Computer and System Sciences. – 2007. –73, № 8. – P. 1176–1190.