УДК 004.75

КОШЕЛЬ М.О.,
СИМОНЕНКО А.В.

БАЛАНСУВАННЯ ЗАВАНТАЖЕННЯ ВЕБ-СЕРВІСІВ, З ВИКОРИСТАННЯМ МОБІЛЬНИХ АГЕНТІВ

В даній роботі описані методи балансування завантаження розподілених веб-сервісів. Розглянуті переваги та недоліки таких методів, та запропонований метод з використанням технології мобільних агентів.

In this work methods of load balancing of the distributed web-services are described. All advantages and lacks of these methods are considered and the optimal method with use of mobile agents is offered.

Вступ

На даний час веб-сервісам знаходять все більш широке застосування. Вони використовуються в різних випадках і ситуаціях в Інтернеті [1, 2, 3, 4]. Але швидке збільшення кількості веб-сервісів і користувачів цих сервісів потребує підвищення продуктивності веб-серверів, для зменшення часу відгуку на запити, які, в свою чергу можуть з'явитися в будь-який момент часу. При цьому необхідно забезпечити ще й надійність таких веб-серверів.

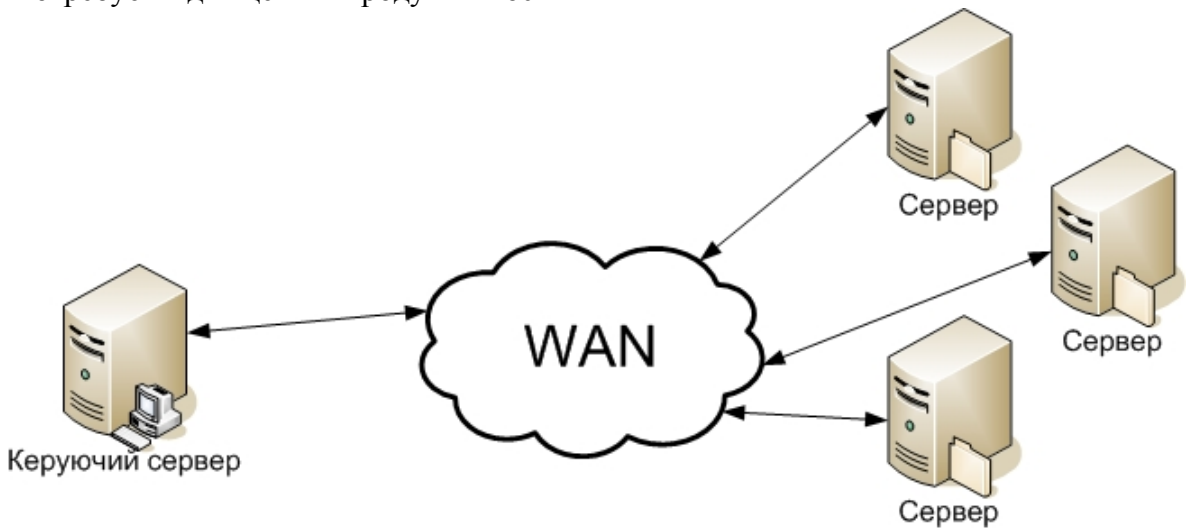

Рис.1. Структура розподілених веб-сервісів

Для реалізації високопродуктивних і надійних веб-серверів використовують розподілені веб-сервери. Розподілені веб-сервери представляють собою набір (N-ну кількість серверів) веб-серверів. Це продубльовані ресурси для одночасного надання послуг багатьом користувачам. Вхідні запити можуть бути розподілені між серверами згідно певних стратегій розподілення завантаження, і тому ці запити можуть бути опрацьовані в певних часових межах (час відгуку). Розподілені веб-сервери (мал. 1) можуть бути організовані різними способами:

- вони можуть бути інтегровані в кластер веб-серверів, з'єднаних через локальну обчислювальну мережу, щоб працювати як один потужний сервер;
- вони можуть використовуватись в різних географічних місцях через глобальну обчислювальну мережу;

Розподілені веб-сервери можуть легко розширюватися, та мають високу степінь масштабованості. Кількість їх може бути збільшено простим додаванням нового сервера в локальну мережу.

Для забезпечення доброї масштабованості високопродуктивних веб-серверів потрібно проведення балансування завантаження усіх веб-серверів. Вхідні запити від користувачів повинні бути розподілені відповідно до стратегії навантаження між серверами, щоб ко-

ристувач отримував відповідь на запит в певних часових межах. Як і у випадках з розподіленими обчисленнями та розподіленим моделюванням, роботу з перевантажених серверів необхідно перемістити на не завантажені, що сприяє підвищенню пропускної здатності системи. В іншому випадку може бути ситуація, коли запит користувача буде знаходитись в черзі безкінечно довго. В таких випадках сервер може відхилити запит користувача. В цій роботі описані методи, що допоможуть уникнути таких ситуацій.

Методи балансування завантаження, що базуються на технології клієнт-сервер

Виділяють наступні категорії методів балансування навантаження [1,2]:

- Клієнтські
- Основані на DNS
- Диспетчерські
- Серверні

Рис.2. Клієнтський підхід балансування завантаження

Клієнтський підхід (мал. 2) реалізує вибір сервера на стороні клієнта. Клієнти можуть вибрати один з доступних веб-серверів випадковим чином, або вибрати сервер, що найбільш їм підходить, використовуючи механізми інтелектуального вибору.

Рис. 3. Підхід, що базується на DNS

Наприклад, браузер Netscape Navigator, використовує клієнтський підхід для доступу до своїх сайтів. Коли користувач переглядає домашню сторінку Netscape, браузер випадковим чином вибирає один із серверів і направляє йому запит користувача. Однак випадковий вибір не може гарантувати балансування навантаження всіх серверів та доступність вибраного сервера. Інтелектуальний вибір сервера може бути реалізований з використанням Java апплетів, що запущені на стороні клієнт для визначення стану серверів і затримок мережі. В такому випадку може бути вибраний сервер, що найбільше підходить, і запит користувача буде направлений саме йому. *Недолік такого методу полягає в великій часовій затримці, що викликана визначенням станів серверів.*

Підхід с використанням DNS – це прийняття рішення на стороні DNS-сервера, що обробляє запити по трансляції імен (мал. 3). Щоб перетворити ім’я, клієнт відправляє запит на DNS-сервер для перетворення імені веб-сервіса на адресу веб-сервера. Він, згідно

з стратегією балансування, вибирає адресу сервера, та відправляє його клієнту. Найпростіша стратегія – RoundRobin, а більш складна включає моніторинг завантаження, облік адміністративних особливостей мережі і т.д.

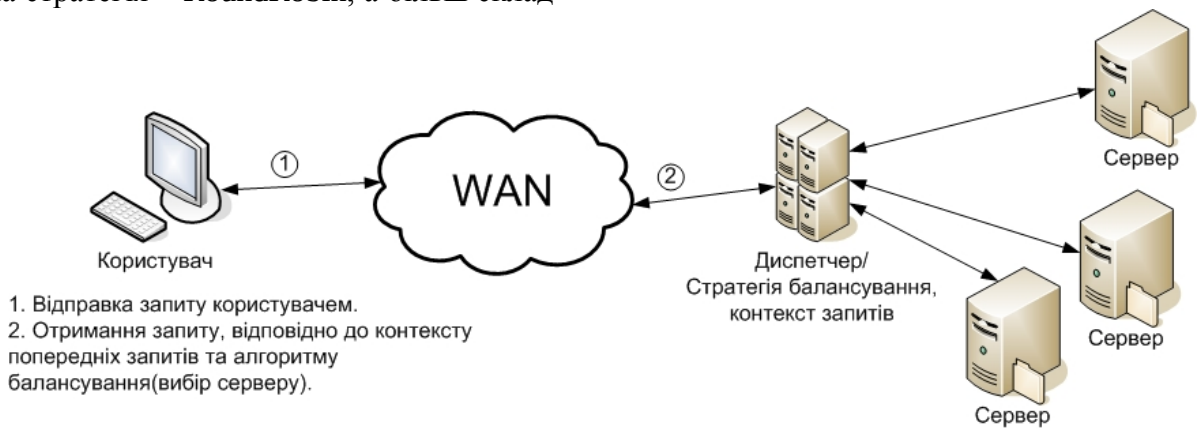

Рис. 4. Диспетчерський підхід балансування завантаження

Однак, DNS-сервер може стати вузьким місцем в процесі маршрутизації. Існують програмні продукти для розподілення навантаження між багатьма географічно розподіленими серверами, в яких DNS-сервер визначає і доступність серверів, і часову затримку в мережі для вибору найбільш оптимального сервера, що використовує технологію клієнт-сервер.

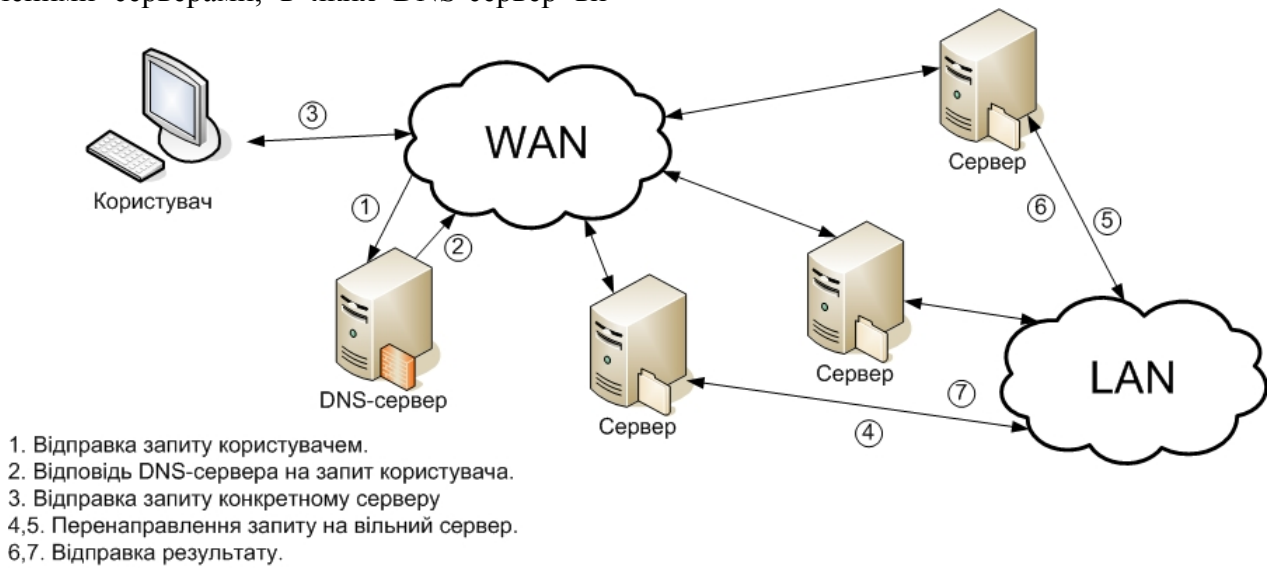

Рис. 5. Серверний підхід балансування завантаження

Диспетчерський підхід (мал. 4) виконує відображення адрес на рівні IP. Це потрібно для того, щоб отримати повний контроль над маршрутизацією клієнтських запитів. Він визначає групу веб-серверів, що має однотипну віртуальну IP адресу, яка є IP адресою диспетчера. Диспетчер в свою чергу діє в якості центрального планувальника і має повний контроль над маршрутизацією запитів. Диспетчер ідентифікує кожний сервер з частковою адресою і, якщо потрібно, то перенаправляє клієнт-серверний пакет, переписавши IP адресу. В основі цього підходу лежить механізм на основі TCP-маршрутизатора. TCP-маршрутизатор в цьому підході, виконує роль диспетчера. Основним недоліком цього підходу є накладні витрати на зміну адреси. Однак існує інший спосіб перенаправлення пакетів – перенаправлення пакетів на рівні HTTP. Тут диспетчер розподіляє вхідні запити між серверами через механізми перенаправлення, що надає протокол HTTP. Диспетчер може перенаправити запит через відповідь клієнта, при цьому в заголовку відповіді вказана адреса сервера. В цьому випадку, клієнт перенаправляє запит, але вже напряму на сервер.

Серверний підхід (мал. 5) використовує механізм дворівневої диспетчеризації. DNS-сервер спочатку, визначає сервер, на який йде клієнтський запит. Після цього, кожен сервер може перенаправити запит на інший сервер. Це децентралізована стратегія перерозподілення завантаження, в якій всім сер-

верам дозволена участь в процесі балансування завантаження. Існує реалізація такого підходу на основі веб-сервера Apache.

Таблиця 1. Переваги та недоліки методів балансування завантаження, що використовують технологію клієнт-сервер.

Підходи	Переваги	Недоліки
Клієнтський	Простота розроблення програмного забезпечення.	Велика часова затримка, викликана визначенням станів серверів.
DNS		Велика часова затримка, викликана великою чергою запитів на DNS-сервері, коли в один момент часу приходить велика кількість запитів.
Диспетчерський		Збільшення накладних витрат на зміну адреси.
Серверний		Велика часова затримка, при пере направленні сервером запитів, на інший. Може виникнути безкінечний цикл.

Ця реалізація дозволяє виконати однорідне пере розподілення HTTP запитів від серверів до мало використовуваних. Запит передається по ланцюжку функцій прийняття рішень, що називаються функціями-кандидатами. Кожна функція-кандидат пере впорядковує набір доступних серверів і вибирає метод пере розподілення запиту (HTTP-пере направлення або через HTTP-проксі), базуючись на інформації про ресурси.

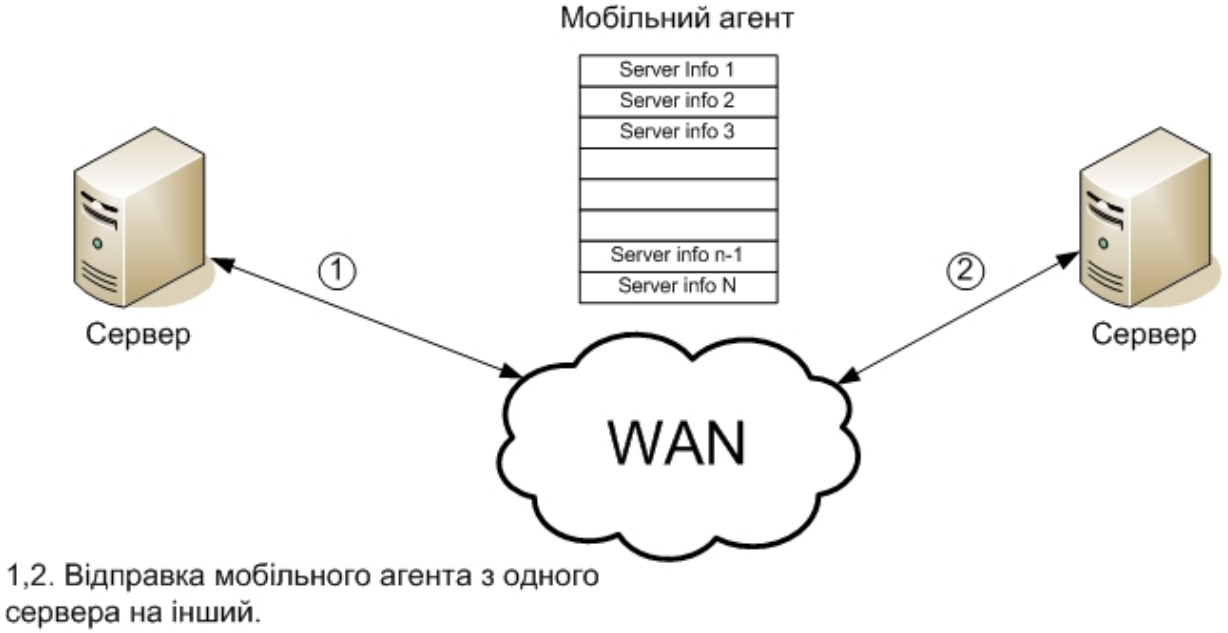

Рис. 6. Балансування завантаження з використанням мобільних агентів

В таблиці 1 представлені переваги та недоліки методів балансування завантаження, що базуються на технології клієнт сервер. Запропонований в статті метод балансування навантаження, з використанням технології мобільних агентів не має вказаних недоліків.

Метод балансування завантаження з використанням мобільних агентів

Мобільних агент – це програмний компонент, який може автоматично пересуватись з одного вузла мережі на інший, разом зі своїм станом і виконавчим кодом і виконувати різноманітні операції на цих вузлах.

Агенти можуть розділити функціональність при проектуванні веб-систем. В традиційному підході балансування завантаження, який базується на відправці повідомлень, модулі серверних сервісів змішують основну функціональність веб-сервіса з підтримкою таких функцій, як балансування завантаження. Коли вводиться нова політика балансування завантаження, може знадобитись переписати серверний модуль. З іншої сторони

при використанні мобільних агентів функції підтримки можуть бути відділені від сервісних модулів і реалізовані порізно в мобільних агентах. А це свідчить про те, що спосіб балансування завантаження, що базується на мобільних агентах, являється більш гнучким при підключенні нових політик балансування завантаження для різноманітних веб-серверних систем.

Агенти використовують мало трафіку. В методах, які базуються на передачі повідомлень, веб-сервери повинні періодично обмінюватись повідомлення з інформацією про завантаження серверів, для того щоб приймати рішення балансуванню завантаженням. Обмін повідомленнями хоча і знижує пропускну здатність веб-системи, але передається лише службова інформація, тому це незначно впливає на продуктивність мережі. Взаємодія на місці використання усуває прямий обмін повідомленнями, що відбувається між серверами в обох напрямках. З цього слідує, що можуть бути значно зменшені мережевий трафік і затрати на комунікацію.

Мобільні агенти підтримують асинхронні і автономні операції. Сервери можуть запускати мобільні агенти, які незалежно мандрують між серверами (мал. 6), і виконують різного роду операції. Мобільний агент може виконувати алгоритми, що містять різноманітні стратегії балансування завантажень і приймають рішення по розподіленню завантаження на льоту, згідно поточному стану сервера.

Використання мобільних агентів підвищує надійність веб-систем, так як вони можуть переносити запроси клієнтів від несправних серверів до активних(ті що працюють).

Список посилань

1. Таненбаум Е. Комп’ютерні мережі. –СПб.:Пітер. – 2002.
2. Таненбаум Е. Сучасні комп’ютерні системи. –СПб.:Пітер. – 2008.
3. Таненбаум Е., Ван Стеєн М. Розподілені системи. – СПб.:Пітер, 2003.
4. Coulouris G., Dollimore J., Kindberg T. Distributed Systems. Concepts and Design. - Addison-Wesley Publishing Company, 1995.

Поступила в редакцию 18.12.2009