

Журавель С.В.

МЕТОДИ ОПТИМІЗАЦІЇ РОЗПОДІЛУ НАВАНТАЖЕННЯ

sergey.zhuravel@gmail.com

Національний авіаційний університет

Розглянуто найбільш розповсюджені методи розподілу навантаження для WEB-серверів що підвищують їх швидкодію та продуктивність і забезпечують їх безперервну роботу

Ключові слова: *Web-сервер, load balancing, DNS, обчислювальна система, системи розподілу навантаження*

Вступ

WEB-сервери стали не тільки сховищем текстової і графічної інформації, але і місцем зберігання мультимедійних матеріалів. Пріоритетнішою стає задача обслуговування запитів за гарантований час, що неминуче потребує удосконалення технічних, алгоритмічних і програмних засобів побудови розподілених WEB-серверів.

Існує кілька способів підвищення швидкодії WEB-сервера: можна збільшити полосу пропускання, встановити високопродуктивне мережеве обладнання, розробити ефективні додатки для WEB, оптимізувати і модернізувати програмні і апаратні компоненти WEB-сервера, а також взяти до уваги технологію «кешування» в середовищі WEB.

Ще один спосіб підвищення продуктивності вузла WEB полягає в тому, щоб збільшити кількість серверів і розташовувати на них «зеркальні» копії матеріалів. Таким чином можна розподілити загальне навантаження по всім компонентам систем і скоротити час повернення інформації при виконанні сервером внутрішніх процедур оброблення клієнтських запитів. При цьому зберігаються і існуючі сервери, оскільки виводити їх із експлуатації і замінити новими не має потреби.

Також слід відзначити програмні продукти, які зменшують навантаження, розподіляючи її по декільком серверам. Окрім цього, вони підвищують відмовостійкість WEB-серверів: у випадку відмовлення одного із серверів пакети даних

направляються на інший. Таким чином, час очікування зменшується, а число необроблених запитів зводиться до мінімуму. Системи балансування навантаження можна використовувати як при наявності лише одного WEB-додатка, так і при роботі з цілим рядом вузлів. Отримавши представлення про те, що таке «системи балансування навантаження» і як вони працюють, можна визначити найбільш важливі їх характеристики, які слід враховувати при виборі засобу вирівнювання навантаження.

Існує декілька розповсюджених методів розподілу навантаження для WEB-серверів:

– круговий DNS (*Round Robin DNS*) – цей метод вважається одним із перших в основі якого знаходиться DNS-сервер. Його суть полягає в циклічному проходженні по списку IP-адрес серверів, що знаходяться у кластері;

– апаратний розподіл навантаження (*Hardware Load Balancing*) – даний метод полягає в направленні TCP/IP пакетів до різних серверів у кластері;

– програмний розподіл навантаження (*Software Load Balancing*) – процес балансування в цьому методі оснований на програмному забезпеченні. Зазвичай, це інтегровані компоненти у WEB-сервер і серверні додатки у вигляді програмних пакетів;

– змішані схеми, коли використовується комбінація апаратних і програмних засобів.

Круговий DNS

Круговий *DNS* – це найпростіший спосіб перенаправлення *HTTP*-запитів на кілька серверів. Як відомо, будь-який *DNS*-сервер зберігає пари «ім'я хоста/*IP*-адреса» для кожного комп'ютера в певному домені.

Специфікація стандарту *DNS* дозволяє вказати кілька різних *IN A* записів для одного імені.

Цей список виглядає приблизно так:

```
domain.com IN A 10.0.0.1
```

```
www.domain.com IN A 10.0.0.2
```

У цьому випадку *DNS* сервер у відповідь на запит дозволу імені в *IP*-адресу буде видавати не одну адресу, а весь список:

```
# host www.domain.com
```

```
www.domain.com has address 10.0.0.1
```

```
www.domain.com has address 10.0.0.2
```

При цьому з кожним новим запитом послідовність адрес у списку у відповіді буде змінюватися. Таким чином клієнтські запити будуть розподілятися між різними адресами і потраплятимуть на різні сервери.

Балансування. На перший погляд це рішення здається непоганим і дуже дешевим. Але балансування навантаження тут полягає лише в тому, що кожний із серверів отримає рівну кількість запитів.

У цьому рішенні немає чіткого критерію вибору *IP* зі списку клієнтом (може вибиратися перший елемент, може кешуватися останнє вибране значення, можливі інші варіанти) - все залежить від реалізації конкретного клієнта.

Відмовостійкість. Крім того, ця схема ніяк не допоможе у випадку виходу будь-якої машини з ладу. Немає механізмів визначення доступності вузлів і можливості завдання їх "ваг". У разі аварії на одному або декількох вузлах кластера навантаження буде продовжувати розподілятися між робочими вузлами і з тими, що вийшли з ладу.

Ще один з недоліків – кешування таблиць *DNS*. Це може привести до того, що схема кругового *DNS* взагалі втрачає

свою суть. У разі аварії або зміні тих чи інших записів в *DNS* потрібен деякий час (залежить від налаштувань *DNS*) для оновлення даних на всіх клієнтах.

Адміністрування. З погляду адміністратора – ця схема ідеальна. Для її роботи не потрібне додаткове обладнання й перебудова мережі й серверів. На внесення змін у таблицю *DNS* не потрібно багато часу й сил.

Апаратний розподіл навантаження

Даний метод не є кращою альтернативою круговому *DNS*.

У мережі використовується прилад, що називається апаратним розподільувачем навантаження (*Hardware Load Balancer*), який працює в складі обчислювального кластеру.

При отриманні запитів для кластера, розподільувач приймає рішення, якому із серверів кластера варто його направити, опитує всі сервера й додатки (тобто певний *TCP/IP* порт) – чи доступні вони. У деяких випадках перевіряє, чи повертає серверу коректні дані. Ця переробка називається "перетворенням мережної адреси" (*network address translation*);

Балансування. Для визначення якому із серверів направити запит розподільувач вирішує на основі статистичних даних. Він збирає величезну кількість інформації про активність у мережі. У нього закладено кілька алгоритмів, з яких адміністратор системи може вибирати один із них. Алгоритми містять у собі й кругове перемикання адрес і пропорційний перебір (круговий перебір адрес із коефіцієнтами). Завдяки цим алгоритмам розподільувач може прийняти "розумне" й ефективне рішення щодо розподілу навантаження. У такий спосіб отримуємо більш ефективний результат, чим при використанні кругового *DNS*.

Відмовостійкість. Якщо один з серверів виходить з ладу або не відповідає на запити, апаратний розподільувач навантаження не буде посилати йому дані, він видалить його зі свого списку серверів і на-

править весь трафік на інший сервер. У результаті стійкість вища, ніж при використанні кругового *DNS*.

Не можна випускати з уваги те, що сам розподільувач теж може вийти з ладу. Для рішення цієї проблеми звичайно встановлюють другий, дублюючий розподільувач, що працює як "завжди готовий резерв" (*hot spare*).

Ця конфігурація з такою надійністю як правило дуже стійка.

Адміністрування. В адмініструванні дана схема трохи складніша, ніж схема із круговим *DNS*. Встановити й налаштувати розподільувач не складніше, ніж звичайний маршрутизатор, і вмінь адміністратора мережі буде досить для цього. Налаштування виконуються тільки на самому розподільувачі й ніяк не впливає на самі серверні машини. Це велика перевага, тому що адміністраторам не треба займатися конкретно кожним сервером.

Програмний розподіл навантаження

Для розгляду роботи даного методу, візьмемо за основу програмний продукт, що базується на описі програмного розподільувача навантаження від *Microsoft (Windows Load Balancing Service – WLBS)*. На кожен сервер встановлюється програмне забезпечення, що поєднує сервера в єдиний кластер з єдиною *IP*-адресою. Ця *IP*-адреса й привласнюється доменному імені проекту. Кожній машині привласнюється свій ідентифікатор у кластері від 1 до 32. Тобто, максимум у кластері може бути 32 машини. Крім того призначається ваговий коефіцієнт, тому що машини можуть мати різний обсяг ресурсів, і встановлюється набір правил.

Балансування. Звісно, виникає запитання, якщо 32 машини в кластері будуть приймати пакети для одного і того ж самого *IP*-адреса, то клієнт повинен буде отримувати 32 відповіді на свій запит? Секрет полягає в тому, що при встановленні розподільувача, між *TCP/IP* стеком і драйвером мережної карти використовується фільтр, що визначає, який із

серверів повинен обробити запит. Всі сервера в кластері фільтрують трафік, і тільки певний сервер відповідає на запит.

Так як цей розподільувач є програмою, у нього більше засобів об'єктивно оцінювати здатності машини обробити запит. Йому доступний відсоток завантаження процесора, обсяг вільної пам'яті, і обсяг вільного місця на диску та інше. Тому що принцип роботи побудований на фільтрації пакетів, а не на перетворенні їхніх заголовків, програмний розподільник працює швидше, ніж апаратний.

Звичайно на кожную машину встановлюють ще одну мережну плату, через яку машини в кластері спілкуються між собою й з базою даних. Тоді весь подібний трафік іде через перший мережний інтерфейс (на якому знаходиться розподільувач), а весь міжмашинний трафік іде через другий інтерфейс, так що адміністрування машин та інші сервіси не заважають роботі *WEB*-серверів.

Відмовостійкість. Розподільувачі на всіх машинах періодично розсилають по мережі особливі повідомлення, за допомогою яких визначається стан усього кластера. Коли додається або видаляється машина, програмний розподільник починає процес, що називається "відомість" (*convergence*). Під час відомості машини, оцінюється новий стан кластера й відповідно міняється алгоритм. Зазвичай, на це витрачається 10-13 секунд.

Серед повідомлень, які генерує розподільувач є повідомлення яке називається "пульс" (*heartbeat*). У розсиланні таких повідомлень бере участь кожна машина. Частоту пульсу можна міняти, а за замовчуванням вона дорівнює 1 пульсу (повідомленню) в секунду. Машина вважається недоступною, якщо вона не змогла взяти участь в 5 "ударах пульсу" підряд. Після цього машини, що залишилися, починають процес "відомості".

Адміністрування. На відміну від попередніх розглянутих нами рішень,

програмний розподіл навантаження сто-сується кожного сервера. На кожену машину в кластері потрібно встановити програмне забезпечення й налаштувати його, встановити додаткові мережні карти, а значить буде потрібно ще один комутатор (*switch*). Крім цього, конфігурація кожної машини буде відрізнятися, тому що в кожній машині буде своє унікальне ім'я й свій ваговий коефіцієнт.

Позитивна сторона полягає в тому, що програмним розподілом навантаження можна управляти з будь-якого комп'ютера мережі. Як тільки нова машина налаштована й на ній установлений розподілювач, її можна включати в мережу й віддалено додати її в кластер чи видалити її звідти.

Змішані схеми

Існують також змішані схеми, коли працюють разом апаратні й програмні розподілювачі навантаження. У системі використовується один апаратний розподілювач навантаження (і ще один резервний) і два кластери серверів, у яких використовуються програмні розподілювачі навантаження. Апаратному розподілювачу невідомо про те, що за двома IP-адресами знаходиться безліч серверів. Так само, кожному із кластерів невідомо про існування свого сусіда.

Балансування. Використовуючи змішану схему можна перевищити межу в 32 машини, що встановлена програмним розподілювачем. По суті, якщо припустити, що апаратний розподілювач здатний працювати з 256 окремими вузлами (цілком реальне припущення), тоді можна розширювати потужність ресурсу аж до величезної кількості машин – 8192 серверів.

Відмовостійкість. Завдяки такій схемі можна будувати з'єднання серверів, що повністю ізольовані одне від одного не тільки через мережу, але й географічно. У такий спосіб можна отримати додатковий засіб захисту від різних примх природи (повеней, пожеж і т.д.) Якщо безпека якого-небудь із серверів буде порушена, його завжди можна від-

ключити від мережі, не вплинувши на роботу *WEB*-проекту.

Адміністрування. Як було підкреслено вище, апаратні розподілювачі не значно збільшують турботу адміністратора. Реалізація й використання змішаної схеми не сильно відрізняється від реалізації й використання схеми із програмним розподілом навантаження.

Висновки

Проаналізувавши основні методи розподілення навантаження для *Web*-серверів, та маючи представлення розгалуження мережі на сьогодні, можна чітко аргументувати, що самим ефективним методом є поєднання апаратних і програмних розподілювачів навантаження, або іншими словами – змішані схеми. Завдяки таким схемам можна будувати з'єднання серверів, що повністю ізольовані одне від одного не тільки через мережу, але й географічно. Тому, цей варіант дає змогу повноцінно масштабувати кластерну систему не тільки по відношенню до загального навантаження *WEB*-серверів, а й прийняти і впровадити концептуальні рішення, щодо розгалуження кластерної системи в тому чи іншому географічному регіоні, по мірі його завантаження.

Список літератури

1. *Fomenkov M.* Longitudinal study of Internet traffic in 1998-2003 / M. Fomenkov, K. Keys, D. Moore, K. Claffy // WISICT '04: Proceedings of the winter international symposium on Information and communication technologies (Cancun, Mexico). – Trinity College Dublin, 2004. – Vol. 58. – P. 1–6.
2. *Paxson V.* Wide-area Traffic: The Failure of Poisson Modeling / V. Paxson, S. Floyd. – IEEE/ACM Transactions on Networking, 2005. – Vol. 3. – No. 3. – P. 226–244.
3. *Gojmerac A.* Towards low-complexity Internet traffic engineering: The Adaptive Multi-Path algorithm / A. Gojmerac, P. Reichl, L. Jansen // Computer Networks: The International Journal of Computer and Telecommunications

Networking. – Elsevier North-Holland, Inc. New York, NY, USA, 2008. – Volume 52. – Issue 15. – P. 2894–2907.

4. *Киллелиа П.* Тюнинг веб-сервера / П. Киллелиа. – 2-е издание, 2003.–528 с.

5. *Либман Л.* Философия распределения нагрузки / Л. Либман // Журнал сетевых решений LAN. – 2000. – № 5. – С. 13.

6. *Олифер В.Г.* Компьютерные сети. Принципы, технологии, протоколы. Учебник для вузов 3-е издание / В.Г. Олифер, Н.А. Олифер. – Спб.: Питер, 2007. – 960 с.

7. *Столлинкс В.* Современные компьютерные сети. 2-е издание / В. Столлинкс. – Спб.: Питер, 2003. – 783 с.

8. *Таненбаум Э.* Компьютерные сети / Э. Таненбаум.–Спб.:Питер, 2008. – 992 с.

9. Microsoft Inc. MS Windows NT Load Balancing Service (WLBS) [Электронный ресурс]. – 2001.Режим доступа: <http://technet.microsoft.com/en-us/library/cc768150.aspx>

10. *Жуков І.А.* Метод побудови паралельних структур для пошуку альтернативних маршрутів у комп'ютерних мережах / І.А. Жуков, О.П. Мартинова // Вісник НАУ. – К.: НАУ, 2004. – №1. – С. 14–17.

11. *Cardellini V.* The State of the Art in Locally Distributed Web-server Systems / V. Cardellini, E. Casalicchio, M. Colajanni, P.S. Yu. – IBM Research Report, RC22209 (W0110-048), 2001, October. – P. 7–12.

12. Springer Berlin / Heidelberg. A Load Balancing Method Using Ring Network in the Grid Database. 2007.

13. *Чжоу Т.* Системы балансировки нагрузки Web-серверов / Тао Чжоу // Журнал "Windows 2000 Magazine". – 2000. – #03. – P. 17.

14. *Bashroush R.* A generic reference software architecture for load balancing over mirrored Web servers: NaSr case study / R. Bashroush, I. Spence, P. Kilpatrick, T.J. Brown // AICCSA '05: Proceedings of the ACS/IEEE 2005 International Conference on Computer Systems and Applications. – IEEE Computer Society Washington, DC, USA, 2005. – P. 132.

Статтю подано до редакції 10.06.2004