

DOI 10.31891/2307-5732-2019-275-4-62-67

УДК 004.932.7

Ю.П. КЛЮЦЬ, К.Ю. СТЕФАНОВИЧ, Є.С. ШАХОВАЛ, В.І. ДЕМЕСЬКО

Хмельницький національний університет

ДИНАМІЧНЕ БАЛАНСУВАННЯ ТРАФІКУ МІЖ ДЕКІЛЬКОМА ПРОВАЙДЕРАМИ

В роботі проводиться аналіз методів балансування трафіку між декількома провайдерами. Досліджується проблема навантаження на мережу та її вирішення шляхом балансування трафіку. Розкриваються та досліджуються методи, алгоритми та моделі балансування трафіку між декількома провайдерами. Описано метод та функції балансування трафіку.

Ключові слова: балансування мережі, розподіл трафіку, метод, алгоритм, провайдер, інтернет, мережа, процесор, навантаження, мережі.

Y.P. KLOTS, K.Y. STEFANOVYCH, Y.S. SHAKHOVAL, V.I. DEMESKO

Khmelnitskyi National University

DYNAMIC TRAFFIC BALANCE BETWEEN SEVERAL PROVIDERS

In the work, the research and disclosure of the concept of balancing traffic between several providers. The problem of the load on the network and its solution by balancing the traffic is studied. Methods, algorithms and models for balancing traffic between several providers are revealed and explored. Describes the functions and capabilities of each algorithm and network allocation method. Balancing Internet traffic Optimizes network performance. The paper analyses the main load balancing algorithms for different performance indicators where each algorithm has its own parameter. It's also mentioned the main features and scope of each algorithm and determined the advantages and disadvantages of each algorithm. There is also described the method and functions of balancing on the basis of StoneSoft and Barracuda Networks products with their pros and cons. StoneSoft's solutions have a greater emphasis on security and business continuity, while Barracuda is a good solution for medium-sized organizations. The article describes the mathematical model of the load balancing system where the balancing machine uses a mass maintenance system.

Keywords: network balancing, traffic distribution, method, algorithm, provider, internet, network, processor, load, network.

Вступ

Балансування навантаження – одна з основних концепцій, необхідних для побудови надійних розподілених систем.

У комп'ютерних системах балансування навантаження розподіляє навантаження між декількома обчислювальними ресурсами, такими як комп'ютери, комп'ютерні кластери, мережі, центральні процесори або диски. Мета балансування навантаження – оптимізація використання ресурсів, максимізація пропускної здатності, зменшення часу відгуку і запобігання перевантаження будь-якого одного ресурсу.

У термінології комп'ютерних мереж балансування навантаження або вирівнювання навантаження чи розподілення навантаження – це метод розподілу завдань між декількома мережевими пристроями, наприклад, це можуть бути сервери, з метою оптимізації використання ресурсів, скорочення часу обслуговування запитів, горизонтального масштабування кластера, мається на увазі динамічне додавання або видалення пристроїв, а також забезпечення відмовостійкості, тобто резервування.

Основна мета балансування навантаження – це оптимізація використання ресурсів, максимізація пропускної здатності, зменшення часу відгуку і запобігання перевантаження будь-якого одного ресурсу. Використання декількох компонентів балансування навантаження, замість одного компонента, може значно підвищити надійність і доступність за рахунок резервування. Саме балансування навантаження передбачає, зазвичай, наявність спеціального програмного забезпечення або апаратних засобів, таких як багаторівневий комутатор або система доменних імен, таких як серверний процес.

Різниця між балансуванням навантаження і фізичного з'єднання в тому, що балансування навантаження ділить трафік між мережевими інтерфейсами на мережевий сонет основі, в той час як з'єднання каналу передбачає поділ трафіку між фізичними інтерфейсами на більш низькому рівні, і робить це або в пакет, або за допомогою з'єднання.

Підходи та алгоритми балансування мережі

Процедура балансування здійснюється за допомогою цілого комплексу алгоритмів і методів, відповідним таким рівнями моделі OSI:

мережевому;

транспортному;

прикладному.

Динамічний підхід балансування навантаження підрозділяється на два типи:

розподілений;

нерозподілений або централізований.

Вони визначаються наступним чином: В нерозподіленому підході один вузол або група вузлів

відповідають за управління та розподіл по всій системі. Інші вузли не розподіляють завдання і не виконують керуючі функції, отже цей тип алгоритмів не є відмовостійким, може статися перевантаження центрального вузла прийняття рішень. Такий підхід є корисним в невеликих мережах з низьким рівнем навантаженням. В розподіленому підході кожен вузол незалежно будує свій вектор навантаження. Всі вузли в мережі відповідають за розподіл навантаження і наповнення власної локальної бази даних для прийняття ефективних рішень балансування. Це призводить до великих комунікаційних витрат і складності алгоритмів. Такі алгоритми є корисними у великих і гетерогенних мережах.

При розподіленому підході балансування навантаження може мати дві форми: кооперативну і некооперативну. У кооперативній вузли працюють спільно для досягнення спільної мети, наприклад, поліпшення загального часу відгуку. При некооперативному підході балансування вузол працює незалежно від мети, наприклад, поліпшення часу виконання завдання. Вузли постійно взаємодіють один з одним і при розподіленому підході генерують більше повідомлень, ніж при нерозподіленому. Передача повідомлень між вузлами для обміну інформацією про оновлення системи може привести до зниження продуктивності системи. Один вузол або група вузлів вирішує завдання балансування навантаження при нерозподіленому підході, він може приймати дві форми: централізовану та напіврозподілену [1].

У централізованих алгоритмах один вузол одноосібно відповідає за балансування навантаження всієї системи і називається центральним вузлом. Використовується в невеликих мережах. У напіврозподілених алгоритмах кластер формується групою вузлів системи, балансування навантаження відбувається в кожному кластері централізованого типу. Серед вузлів в кластері центральний вузол ініціює балансування навантаження в цій групі. Напіврозподілені алгоритми обмінюються більшою кількістю повідомлень в порівнянні з централізованими. Зазвичай динамічний алгоритм балансування містить чотири елементи: політику балансування (transfer policy); алгоритм вибору партнера (location policy); алгоритм вибору завдання для передачі (selection policy); механізм збору необхідної інформації про стан системи (information policy).

Політика балансування визначає, чи є вузол об'єктом балансування. Різні види політик балансування використовують порогові значення завантаженості вузлів, відхилення величини навантаження вузла від середнього значення по системі та інші методи. Алгоритм вибору завдання визначає, яке завдання необхідно передати. При виборі завдання для відправки алгоритм може враховувати, що накладні витрати, пов'язані з пересилкою завдання, повинні бути мінімальні, складність виконання завдання повинна бути великою, число зв'язків в завданні з локальними ресурсами повинно бути мінімальним. Алгоритм вибору партнера відповідає за вибір відповідного вузла для операції балансування. Поширеною технікою в розподілених алгоритмах є опитування (polling) вузлів. Опитування може бути послідовним або паралельним, використовувати результати попередніх опитувань. У централізованих алгоритмах вузол звертається до спеціалізованого координатора для визначення відповідного партнера для балансування. Координатор збирає і підтримує в актуальному стані інформацію про завантаженість вузлів системи, а алгоритм пошуку партнера використовує ці дані.

Механізм збору інформації про завантаженість системи визначає джерело інформації, час збору даних про завантаженість, місце зберігання інформації. Існує кілька класів механізмів збору інформації.

1. Збір даних по необхідності. В даному класі використовуються розподілені алгоритми, які збирають інформацію про завантаженість, коли вузол потребує балансування навантаження.

2. Періодичний збір даних. Алгоритми даного класу можуть бути як централізованими, так і розподіленими. Залежно від зібраних даних, алгоритм ініціює балансування навантаження.

3. Збір даних щодо зміни стану.

У системах, що реалізують алгоритми даного класу, вузли самі поширюють інформацію про зміну завантаженості при зміні внутрішнього стану. У разі розподілених алгоритмів дані направляються сусіднім вузлам, в разі централізованих алгоритмів дані направляються координатору.

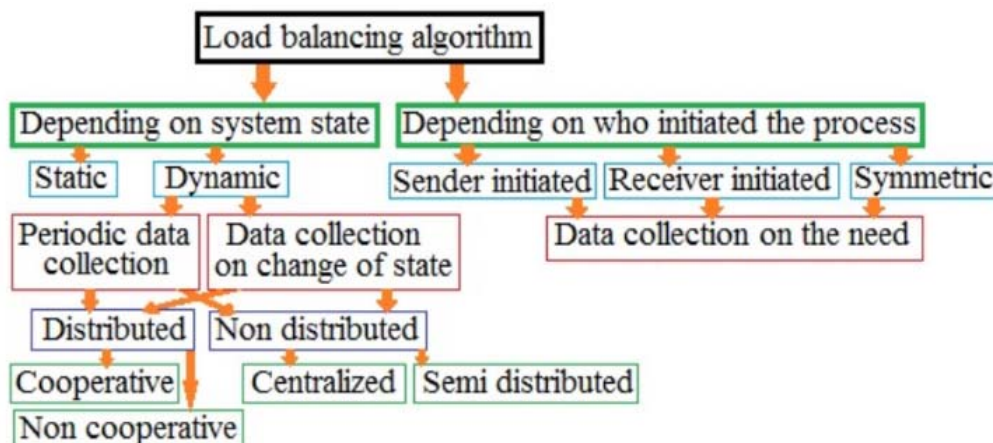


Рис. 1. Класифікація алгоритмів балансування

Завдяки алгоритмам балансування досягається більш висока пропускна здатність і поліпшується час відгуку в розподілених системах. Однак кожен з алгоритмів має як переваги, так і недоліки [2].

Система балансування

Представлена система балансування навантаження складається з групи серверів і балансувальника навантаження. Ця система балансування навантаження побудована на основі підсистеми балансування навантаження і підсистеми управління і моніторингу, які тісно взаємодіють один з одним: підсистема балансування навантаження: алгоритм балансування навантаження, інформація про поточний стан системи, гнучкі настройки QoS, динамічний розподіл трафіку по різних каналах зв'язку і вузлів залежно від їх поточного стану, ступеня завантаження, адміністративних політик балансування навантаження. Підсистема управління та моніторингу: збір та аналіз статистики про поточний стан системи, визначення мультифрактальних властивостей вхідного потоку даних, розрахунок розподілу потоків по вузлах мережі з урахуванням класифікації трафіку і завантаженості серверів і каналів зв'язку.



Рис. 2. Система балансування навантаження

де $h(q)$ – вибіркові значення функції загального значення Херста, $H=h(2)$ – значення параметра Херста, $\Delta h = h(q_{\min}) - h(q_{\max})$ – діапазон загального значення Херста для відрізка трафіку, μ – складність запиту. Складність запиту визначається як вектор необхідних ресурсів $u = (CPU, Net, RAM)$ для виконання запиту. Кожному q -му класу обслуговування відповідає набір векторів необхідних ресурсів $u = (CPU, Net, RAM), r = 1, 2, \dots$

Балансувальник навантаження і сервера з'єднані між собою двосторонніми мережевими зв'язками з максимальною пропускною здатністю $Link_i = \{L_i\}, i = 1, 2, \dots, N$, які мають доступну пропускну здатність $Net_i(t) = \{Net_i\}$ в момент часу t . Кожний сервер $Serv_k, k = 1, 2, \dots$ характеризується параметрами:

$CPU_i(t)$ – обсяг вільного ЦПУ i -го серверу в момент часу t , який обчислюється як:

$$CPU_i(t) = 1 - \sum_j Load_M / j,$$

де $Load_M$ – завантаженість кожного ядра $j = \{1, 2, \dots\}$ – в кожному процесорі $M = \{1, 2, \dots\}$ в сервері в момент часу t . $RAM_i(t)$ – обсяг вільної оперативної пам'яті i -го сервера в момент часу t .

На вхід балансувальника навантаження LB надходять кілька незалежних потоків пакетів з різною інтенсивністю $\lambda_1, \lambda_2, \dots, \lambda_N$ кожен з яких відправляється в чергу Q обмеженої ємності. Час обслуговування заявок залежить від класу обслуговування q , тобто враховується пріоритетність заявок, найвищий пріоритет буде завжди першим. Поки всі пріоритетні запити на обслуговування не будуть оброблені, пакети інших типів залишаються в черзі до закінчення їх часу існування. Знову надійшовші пріоритетні запити переривають обробку непріоритетних і з ймовірністю дорівнює одиниці, витісняють їх в накопичувач, якщо є вільні місця очікування, або за межі системи, якщо вільних місць немає. Витісненні з обслуговування пакети приєднуються до черги непріоритетних вимог і можуть бути дооброблені після всіх пріоритетних. Накопичувачі є роздільними для кожного вхідного потоку, вільні місця очікування цілком доступні для будь-якого запиту, який надійшов знову.

На відміну від типових пріоритетних СМО розглянута система забезпечена імовірним виштовхуючим механізмом. Пріоритетний пакет, що застав всі місця очікування зайнятими в момент обробки іншого пріоритетного пакета, із заданою вірогідністю витісняє з накопичувача один з менш пріоритетних пакетів і займає його місце. Витіснений пакет втрачається або відправляється назад в чергу. Підсистема балансування навантаження LB , відповідно до закладеного в неї алгоритму, здійснює видалення завдань з черг Q і призначає їх на вільні обчислювальні ядра відповідних серверів.

Для опису механізму вивільнення зайнятих трафіком мережеских ресурсів при закінченні передачі трафіку q -го класу обслуговування, це відбувається на основі даних, що надходять від протоколу маршрутизації, що підтримує повідомлення про доступну смугу пропускання і доступних ресурсах на сервері, наприклад, CSPF, SNMP, введемо змінну $LB \in \{0, 1\}$ вказує, що в момент t на сервер перестав надходити трафік класу q ($\varepsilon = 1$) який був прийнятий на обслуговування в момент t_0 і повинен був передаватися по шляху $Net_i(t)$ на сервер $Serv_k$. Дана змінна містить всі необхідні дані для визначення мережеских ресурсів, що підлягають вивільненню. Балансувальник LB в t -й момент характеризується коефіцієнтом втрат $X_{LB}^q(t) \in X$ середнім часом очікування пакета в черзі $T_{LB}^q(t) \in T$.

Моніторинг стану серверів і вільної пропускної здатності можна здійснити трьома способами: після кожного запиту, що надійшов;

в фіксовані проміжки часу, які визначаються статичним алгоритмом;
в нефіксовані проміжки часу, які визначаються динамічним алгоритмом.

Інформація, отримана першим способом, є найбільшою за обсягом, тому що вимірювання проводяться після кожного запиту, що надійшов. При другому способі кількість інформації постійна, але необхідно визначити інтервал знімання інформації, щоб обсяг інформації не був надмірним або недостатнім. При третьому способі кількість інформації залежить від частоти інтервалів контролю, який повинен пристосовуватися до структури вхідного трафіку, з огляду на його самоподібну структуру [3].

Балансування системи

Систему балансування навантаження пропонується будувати на основі наступних підсистем:

1. Підсистема балансування навантаження;
2. Підсистема централізованого управління та моніторингу.

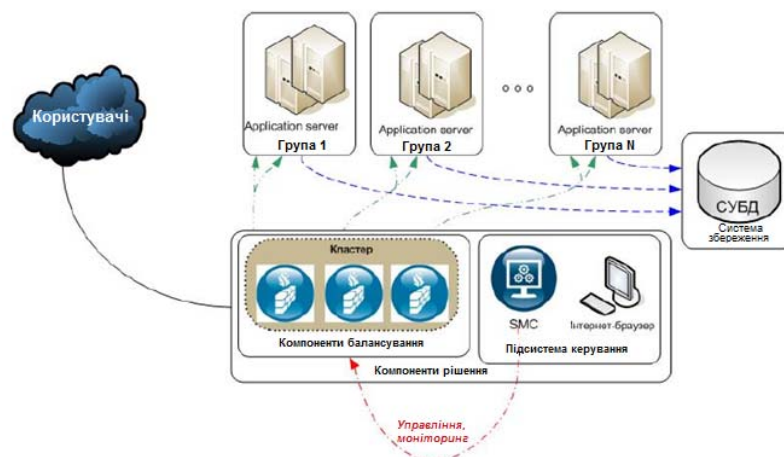


Рис. 3. Система балансування на основі продукту StoneSoft

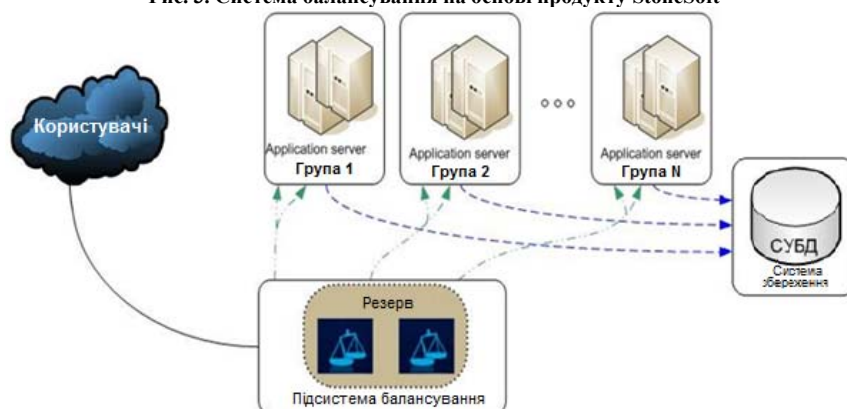


Рис. 4. Схема балансування на основі продукту Barracuda Networks

Рішення з балансування навантаження включає в себе сукупність зазначених підсистем, тісно взаємодіючих одна між одною для забезпечення відповідності заявленому функціоналу.

Рішення побудоване на основі використання продукту компанії StoneSoft вигідно виділяються на тлі конкурентів, оскільки розроблялися з урахуванням потреб великих організацій, для яких безпека бізнес-процесів, в тому числі завдання безперервності бізнесу знаходяться на першому плані. Стандартні апаратні платформи можуть підтримувати швидкості більше 10 Гб/с на пристрій, а з урахуванням можливостей кластеризації ці величини зростають більше 100 Гб/с на кластер. Характерною особливістю є те, що система балансування навантаження тісно інтегрована з фаєрволом і поставляється як комбіноване рішення і робить дану пропозицію оптимальною по співвідношенню ціна / якість і мінімальною з точки зору загальної вартості користування. StoneGate підтримує ряд технологій, що дозволяють реалізувати механізми балансування навантаження пулів серверів шляхом застосування запатентованої технології MultiLink [4].

MultiLink дозволяє без застосування складних протоколів динамічної маршрутизації реалізувати динамічне балансування трафіку за каналами зв'язку залежно від їх завантаження і реальної пропускної здатності в конкретний момент часу, динамічне балансування навантаження по пулу ресурсів (серверів) залежно від їх поточного завантаження, відстежувати стан серверів, коефіцієнт використання процесора, пам'яті, диска і приймати рішення на підставі цієї інформації.

Алгоритм роботи полягає в підтримці однакового середнього значення числа відкритих з'єднань на всіх серверах з урахуванням необхідності коректної «прив'язки» вже відкритих з'єднань до вибраного серверного ресурсу. При цьому можливо зробити інтеграцію з DNS-серверами і відправляти їм DDNS-поновлення при виході, наприклад, з ладу одного з каналів зв'язку, ефективно перенаправляючи

користувачів на інший канал / ресурс / сервер. Для моніторингу стану сервісів можливо використовувати програмні агенти, що працюють на вузлі локально і надають на підставі зібраних даних балансувальнику інформацію про те, наскільки завантажений сервер [5].

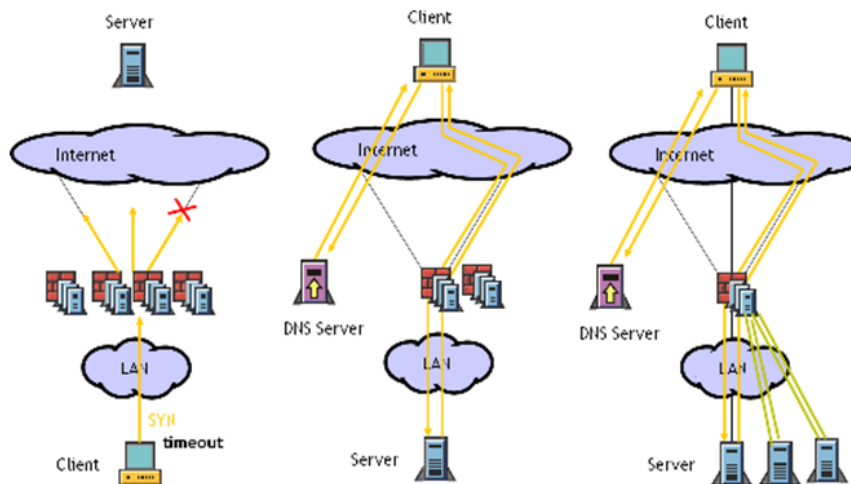


Рис. 5. Балансування навантаження за каналами та ресурсами

Підсистема балансування навантаження на базі рішень компанії Barracuda Networks будується на базі пари пристроїв Load Balancer 340, розраховані на швидкості до 1 Гб/с і обслуговування до 35 серверів. Використання пари пристроїв дозволяє забезпечити резервування системи на випадок виходу з ладу основного пристрою. За нормальних умов другий балансувальник лише синхронізує стан сесій і включається в роботу при виході з ладу основного, коли втрачається канал синхронізації. Час перемикання з основного на резервний становить від 5 до 30 секунд залежно від моделі впровадження. Load Balancer підтримує 3 методи установки: Route-path; Bridge-path; Direct server return.

Найпоширенішим є перший варіант – route-path. В цьому випадку сервери виділяються в окрему підмережу, а пристрої балансування розподіляють трафік, що приходить на віртуальні адреси, які маршрутизуються на зовнішні інтерфейси, що розташовуються в окремому мережевому сегменті. Обмеженням в цьому випадку є необхідність безпосереднього підключення серверів до сегменту, доступному через LAN- інтерфейси Barracuda Networks. При методі впровадження Bridge-path, або Direct Server Return збільшується час перемикання на резервний пристрій (до 30 секунд), а також виникають додаткові вимоги до конфігурації мережевого і серверного устаткування. Bridge-path режим передбачає фізичний поділ сегмента мережі, де розташовуються сервери, на дві частини, що вимагає залучення додаткового мережевого обладнання. У режимі Direct Server Return сервери і балансувальник також знаходяться в одному логічному мережевому сегменті, однак фізичного поділу середовища можна уникнути. Особливістю цього впровадження є те, що запит на пул серверів проходить через балансувальник, проте у відповідь трафік вже йде, минаючи його, безпосередньо до клієнта, який ініціював сесію, що зменшує час відгуку і зменшує завантаження пристроїв балансування. Однак в цьому випадку неможливе використання cookie або трафіку SSL.

Залежно від ваги сервера, яка була йому присвоєна під час конфігурування сервісу, Barracuda може по-різному розподіляти навантаження. Алгоритм вибирається залежно від необхідного результату: в разі round- robin один і той же сервер може отримувати велику кількість тривалих відкритих з'єднань, що може привести до його перевантаження. У другому випадку система намагається рівномірно розподілити кількість відкритих з'єднань за серверами з урахуванням їх ваги [6,7].

Так, підсистема балансування навантаження на базі StoneSoft не лімітує кількість груп серверів і сервісів, для Barracuda це ліміт в 35 серверів, що недостатньо для великих розподілених проектів. Доступ до цих сервісів контролюється і розмежовується за рахунок тісної інтеграції підсистем міжмережевого екранування і балансування навантаження між собою у вирішенні StoneSoft, а в рішенні Barracuda блокуються окремі атаки за рахунок вбудованої підсистеми IPS.

При розподілі трафіку за ресурсами в пулі у StoneSoft можливо контролювати не просто працездатність мережевих інтерфейсів сервера, але і контролювати параметри самої ОС, аж до завантаження процесора, оперативної пам'яті, відгуків служб. У Barracuda стандартне опитування йде по ICMP, але можна також вказати на необхідність моніторингу шляхом ініціювання пробних TCP з'єднань, а також SNMP-опитування завантаження ЦП. У StoneSoft в разі необхідності можливо організувати пріоритизацію трафіку або управління смугою пропускання для різних сервісів або навіть категорій користувачів. У Barracuda для цього необхідно встановлювати зовнішній пристрій. Однак, якщо інфраструктура в частині мережевого обладнання вже розгорнута і працює, це не є критичним фактором [8].

Таблиця 1

Порівняльна таблиця балансування та функцій на базі StoneSoft та Barracuda

Функції	StoneSoft	Barracuda
Кластеризація	Так, до 16 одночасно працюючих пристроїв	Ні, моментальне резервування з 2-х
Балансування навантаження по серверам	Так	Так
Алгоритми балансування	Round robin with session Persistence	Weighted round robin, Weighted Least connection
Прив'язка сесій до серверу	На рівні L4	На рівні L4, HTTP cookie, Session Directory Integration
Перевірка працездатності серверів	ICMP або Агенти на серверах	ICMP, TCP з'єднання, завантаження ЦП
Наявність резервного серверу	Ні	Так
Централізований моніторинг стану системи	Так	Ні, тільки працюючого пристрою
Обмеження доступу до серверів	Так	Ні
Вбудований механізм безпеки	FW, IPS	IPS

Висновки

В роботі проведено аналіз основних алгоритмів балансування навантаження за різними показниками продуктивності, для кожного алгоритму вказані параметри, що в ньому використовуються. Вказані основні особливості і сфера застосування кожного алгоритму, визначені переваги і недоліки кожного алгоритму. Представлено рекомендації вибору алгоритму балансування навантаження, виходячи із специфіки виконуваного завдання, проекту та мети. В роботі запропонована математична модель системи балансування навантаження, в якій балансувальник навантаження описується за допомогою системи масового обслуговування. Стан серверів описуються об'ємом вільного ЦПУ і обсягом вільної оперативної пам'яті. Всі значення параметрів моделі мають залежність від часу. Така модель дозволяє описувати поведінку розподіленої мережі в часі для різних класів обслуговування.

Описано метод та функції балансування на базі продуктів StoneSoft і Barracuda Networks, що мають свої переваги і недоліки. Рішення StoneSoft роблять більший акцент на безпеку і забезпечення безперервності бізнесу, тоді як Barracuda – це хороше рішення для середніх організацій.

Література

1. Кортунів В.І., Воробйов А.В. Вирішення задачі розподілення навантаження на основі динамічної моделі маршрутизатора // Проблеми телекомунікацій. 2015. 1 жовтня (№ 20). С. 4.
2. Емельянов А.О. Балансування навантаження: основні методи та алгоритми // ІТ-інфраструктура для бізнесу. 2017. 1 листопа. (№ 50). С. 15.
3. Камеда Ш., Поуртейлер О. Парадокси в розподілених рішеннях щодо оптимального балансування навантаження для мереж однорідних комп'ютерів. Журнал ACM (JACM), в. 49 н. 3, 2002.
4. Балансування навантаження. URL : <https://habr.com/ru/post/192274/> (дата звернення : 14.03.2019).
5. Балансування навантаження за допомогою Round Robin DNS. URL : <https://www.8host.com/blog/balansirovka-nagruzki-s-pomoshhyu-round-robin-dns/> (дата звернення : 15.03.2019).
6. Порівняльний аналіз методів балансування трафіку. URL : <https://habr.com/ru/company/oleg-bunin/blog/319262/> (дата звернення : 18.03.2019).
7. Відео конференція Олега Буніна. URL : <https://player.vimeo.com/video/151817241> (дата звернення : 18.03.2019).
8. Введення в сучасну мережу балансування навантаження і проксі. URL : <https://medium.com/southbridge/introduction-to-modern-network-load-balancing-and-proxying-52e8ca36adde> (дата звернення : 20.03.2019).

References

1. Kortunov V.I., Vorobyov A.V. Solving the load distribution task based on the dynamic router model. Problems of telecommunications, October 1, 2015. (№ 20). P. 4.
2. Emelyanov A.O. Load Balancing: Basic Methods and Algorithms. IT infrastructure for business. 2017. 1 November (№ 50) P. 15.
3. Kameda H., Pourtallier O. Paradoxes in distributed decisions on optimal load balancing for networks of homogeneous computers. Journal of the ACM (JACM), v. 49 n. 3, 2002.
4. Load balancing. URL: <https://habr.com/en/post/192274/> (view date: 14.03.2019).
5. Load balancing with Round Robin DNS. URL: <https://www.8host.com/blog/balansirovka-nagruzki-s-pomoshhyu-round-robin-dns/> (view date: 15.03.2019).
6. Comparative analysis of methods of balancing traffic. URL: <https://habr.com/ru/company/oleg-bunin/blog/319262/> (view date: 18.03.2019).
7. Video conference by Oleg Bunin. URL: <https://player.vimeo.com/video/151817241> (view date: 18.03.2019).
8. Introduction to the modern network of load balancing and proxies. URL: <https://medium.com/southbridge/introduction-to-modern-network-load-balancing-and-proxying-52e8ca36adde> (view date: 20.03.2019).

Рецензія/Peer review : 14.5.2019 р.

Надрукована/Printed : 17.7.2019 р.

Рецензент: д.т.н., проф. Говорущенко Т.О.