АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

УДК 004.8/681.5 DOI: https://doi.org/10.32515/2414-3820.2021.51.188-194

Р.М. Минайленко, доц., канд. техн. наук, **В.А. Резніченко,** викл., **О.К. Конопліцька-Слободенюк,** викл., **Л.І. Поліщук**, ст. викл.
Центральноукраїнський національний технічний університет, Кропивницький, Україна
e-mail: aron70@ukr.net, upsbilli@ukr.net

Огляд методів балансування навантаження в хмарних системах

В статті проведено огляд методів балансування навантаження в хмарних системах. Показано, що існуючі методи балансування навантаження хмарних систем мають обмежене використання і на даний час універсальної системи балансування навантаження не існує. Крім того, ні один із розглянутих методів не враховує такі важливі складові систем як мережа і дискова підсистема. Методи балансування навантаження хмарних систем вимагають вдосконалення, метою якого повинна бути можливість повного моніторингу системи для задоволення вимог користувачів і розробників
хмарні системи, обчислювальні ресурси, балансування навантаження, продуктивність

Постановка проблеми. Аналіз і розподіл навантаження в хмарних системах є доволі актуальним завданням, оскільки більшість хмарних систем з відкритим доступом використовують прості планувальники навантаження своїх фізичних серверів[1-3].

Проблема балансування навантаження вимагає вирішення не тоді, коли сервер неочікувано вийшов з ладу в процесі роботи над поставленим завданням, що у користувачів відбиває бажання використовувати такий продукт, а на самому початку створення проекту. На ранніх стадіях проектування цілком прийнятно нарощувати потужність з допомогою підключення нових серверів або застосовувати алгоритми оптимізації коду. Але при досягненні певної граничної межі цих заходів стає недостатньо[1,2].

Аналіз останніх досліджень і публікацій. Хмарні системи є на сьогоднішній день найбільш популярною концепцією інформаційних систем і є результатом еволюції цілого ланцюга методів їх побудови. Основним завданням хмарних технологій є створення віртуальної хмарної системи яка складається із віртуальних розподілених ресурсів. Ці ресурси забезпечують віддалене надання послуг доступу до хмарної системи з потрібним рівнем обслуговування користувача. Хмарні технології вирішують наступні завдання[3-5]:

– виконання додатків у хмарі;

– віртуалізація обладнання і обчислювальних ресурсів;

– забезпечення одночасної роботи великої кількості користувачів, причому кількість користувачів може змінюватись.

Коли йде мова про послуги (особливо платні) виникає завдання забезпечення високого рівня обслуговування користувачів . У випадку інформаційних технологій це означає, що система повинна забезпечувати швидкий обмін даними і доступ повинен

бути зрозумілим та зручним. Важливим фактором також є економічні аспекти, пов'язані з витратами на обчислювальні ресурси. Тому в хмарних системах часто існує проблема вирішення завдання пошуку оптимального співвідношення між потрібним рівнем обслуговування та економічними витратами на обчислювальні ресурси. Також існує проблема вибору методів балансування навантаження обчислювальних ресурсів, які б дозволяли гнучко змінювати об'єми реальних ресурсів в хмарній обчислювальній системі і значно збільшувати її продуктивність[1, 2-5].

Постановка завдання. Важливим завданням хмарних технологій є балансування навантаження обчислювальних ресурсів. Методи балансування навантаження переслідують наступне:

– рентабельність, тобто покращення продуктивності системи за розумну ціну;

– універсальність, тобто система має бути гнучкою, з можливістю масштабування і змінювати свої розміри та топологію. Причому алгоритм повинен залишатись працездатним при збільшенні навантаження;

– зменшення часу відгуку на запит і часу виконання запиту;

– забезпечення продуктивності, тобто всі сервери системи працюють з однаковою завантаженістю;

– розуміння алгоритму, тобто потрібно знати всі переваги та недоліки виконуваного алгоритму і розуміти як він працює.

Мета балансування навантаження:

– покращення роботи системи;

– підтримка стабільності системи;

– наявність резервного плану при виникненні критичних ситуацій в системі; можливість адаптування майбутніх модифікації системи.

Виклад основного матеріалу. На даний час існує декілька рівнів балансування навантаження: [6-9].

1. *Метод кластеризації.*

Даний метод дозволяє керувати декількома незалежними серверами як однією системою, що дозволяє значно спростити керування системою і зробити її універсальною. Схему балансування навантаження методом кластеризації показано на рис. 1:

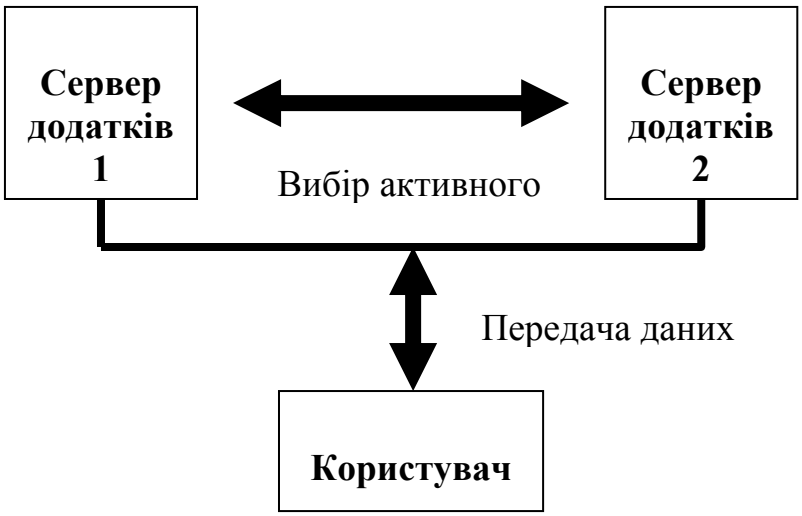

Рисунок 1 – Схема балансування навантаження методом кластеризації

Джерело: [6]

Даний метод забезпечує декілька способів балансування навантаження системи:

1. Адміністративний – адміністратор вирішує як розподілити навантаження між серверами.

2. За замовчуванням – навантаження серверів розподілене порівну між ними.

Всі запити користувачів розподіляються між вузлами, тим самим балансуючи навантаження під час видалення вузла кластера або додавання нового. При зміні навантаження на апаратну частину серверів розподілене навантаження не змінюється. Такий механізм розподілу навантаження показує найбільшу ефективність при використанні web–сервісів з великою кількістю клієнтів які продукують велику кількість коротких запитів при цьому забезпечується швидка реакція на зміну складу вузлів кластера.

Сервери кластера, якими керує механізм балансування навантаженням, через деякі проміжки часу обмінюються інформацією. При виявленні збоїв, коли один із серверів вийшов з ладу, вузли, що залишились розподіляють навантаження гарантуючи стабільну роботу системи і доступність до серверів, а сам збій відображується на роботі системи як невелика затримка відгуку на запит.

– Недоліки даного методу:

– Сервери повинні знаходитись в одному сегменті мережі.

– Кожний вузол потребує спеціального програмного забезпечення для функціонування кластера.

– Неможливість побудови мультиплатформеного кластера.

2. *Метод балансування навантаження через один пристрій*

В даному методі звертання користувача до хмари відбувається через визначений пристрій, котрий розподіляє навантаження за визначеними наперед правилами або орієнтується на час обробки запиту серверами, змінюючи встановлені параметри сесій. Причому зміни відбуваються і в заголовках сесій пакетів, що пересилаються. Схему балансування навантаження через один пристрій показано на рис.2:

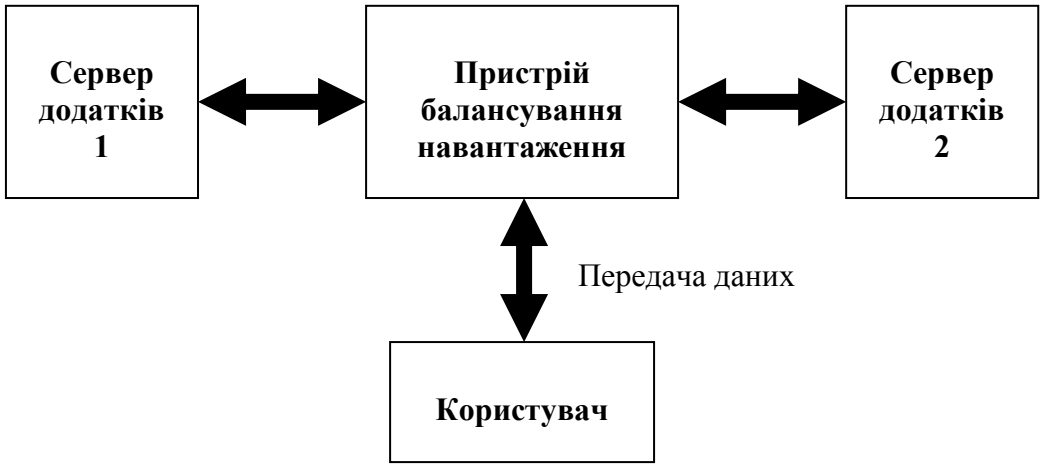

Рисунок 2 – Схема балансування навантаження через один пристрій.

Джерело: [6, 7]

Даний метод застосовується при відсутності потреби у створенні довготривалої сесії між сервером і користувачем.

3. *Метод балансування навантаження з використанням проксі–сервера.*

В даному методі балансування навантаження відбувається з допомогою змін заголовків на рівні вище транспортного. Запит користувача надходить на web–сервер,

де відбувається його обробка і відповідь надходить до користувача. Для збільшення продуктивності такої схеми обміну інформацією створюється проксі–сервер між користувачем і web–сервером. Схему балансування навантаження з використанням проксі–сервера представлено на рис.3:

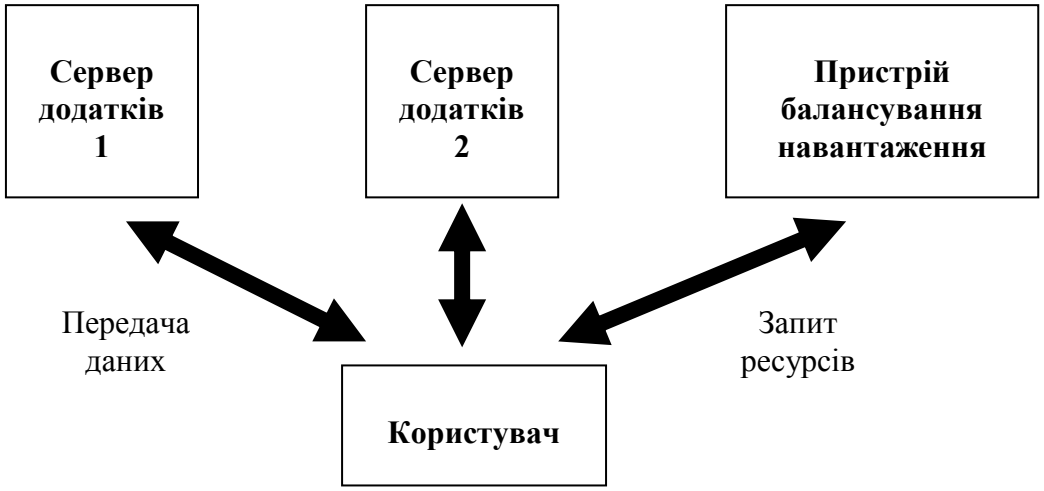

Рисунок 3 – Схема балансування навантаження з використанням проксі–сервера
Джерело: [6]

Розглянемо суть даного методу більш детально. Від користувача надходить запит на проксі –сервер, який пересилає запит на web–сервер. Web–сервер формує відповідь на запит користувача і пересилає її на проксі–сервер, де відбувається кешування відповіді від сервера. Збережена інформація передається у відповідь на всі інші запити користувача. Схему роботи проксі–сервера представлено на рис.4:

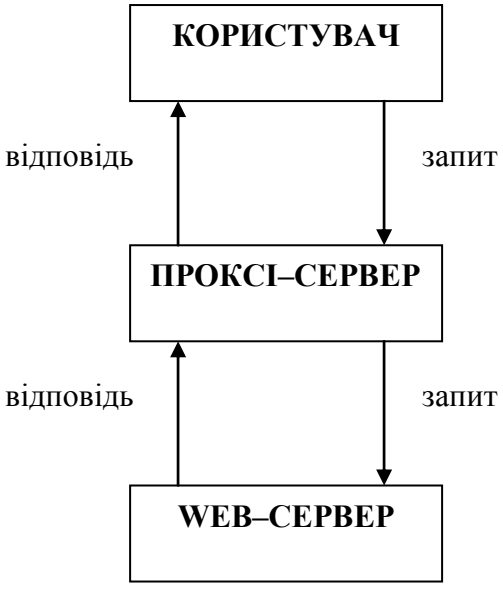

Рисунок 4 – Схема роботи проксі–сервера

Джерело: [9]

Крім того проксі–сервер балансує навантаження між web–серверами які працюють з різними областями додатків. В межах даного методу можна задавати наступні параметри:

– граничний час очікування відгуку сервера на запит;

– граничну кількість неуспішних запитів;

– можливість балансування додатків.

Недоліками даного методу балансування навантаження є:

– висока вартість апаратно–програмного комплексу який вирішує проблему балансування навантаження;

– наявність тільки одного пункту пропуску трафіка;

- використання універсальних апаратних платформ для балансування зменшує функціональні можливості і можливість масштабування системи;

- вузька галузь застосування при високих вимогах до продуктивності і надійності.

Інші методи, що не використовують пропускання трафіку через один пристрій ділять рівномірно або пропорційно запити між платформами, що входять в склад розподіленого сервера. І у випадку коли всі платформи мають різні обчислювальні потужності та різні встановлені додатки, розподіл додатків може призвести до перенавантаження одних платформ і недозавантаження інших. Навіть у випадку однорідності обчислювальних ресурсів не гарантовано, що кожний сервер буде виконувати запит однаково. В такому випадку звичайне зависання системи буде сприйматись користувачем негативно і він буде робити спробу відновити роботу з системою, що приводитиме до збільшення навантаження на сервер[1,2,8-14].

Крім того у хмарному середовищі на процес розподілу ресурсів можуть впливати наступні чинники:

– нерівномірна завантаженість системи;

– відсутність призначення ресурсів користувачем;

– використовувані ресурси не відповідають тим, на які створювався запит;

– відсутність інформації про реальне використання додатка в ресурсі;

– відмінність між ресурсів і класу обладнання;

– додатки використовують різні ресурси;

З метою урахування цих факторів в хмарних обчислювальних середовищах виникло декілька способів виділення і розподілення ресурсів:

- планування ресурсів диспетчером;

- кластеризація;

- призначення ресурсів вручну.

Але універсальної системи балансування навантаження все ж не існує і для кожної моделі обслуговування застосовується свій метод або множина методів[14].

Висновки. Проведений огляд показав, що існуючі методи балансування навантаження хмарних систем мають обмежене використання і на даний час універсальної системи балансування навантаження не існує. Крім того ні один із розглянутих методів не враховує такі важливі складові систем як мережа і дискова підсистема.

Методи балансування навантаження хмарних систем вимагають вдосконалення, метою якого повинна бути можливість повного моніторингу системи для задоволення вимог користувачів і розробників.

Список літератури

1. Карр Н. Великий переход: что готовит революция облачных технологий. / перевод. Андрей Баранов. М.: Манн, Иванов и Фербер, 2014. URL: http://loveread.ec/view_global.php?id=66055/
2. Андреевский И.Л. Технологии облачных вычислений. СПб.: Санкт-Петербургский государственный экономический университет, 2018. 79 с.
3. Caballer M., Blanquer I., Moltó G., de Alfonso C. Dynamic management of virtual infrastructures . *Journal of Grid Computing*. 2015. Vol. 13. No. 1. P. 53–70. Doi: 10.1007/s10723-014-9296-5
4. Giannakopoulos I., Konstantinou I., Tsoumakos D., Koziris N. Cloud application deployment with transient failure recovery . *Journal of Cloud Computing*. 2018. Vol. 7. No. 1. Art. no. 11. Doi: 10.1186/s13677-018-0112-9
5. Spanaki P., Sklavos N. Cloud Computing: Security Issues and Establish-ing Virtual Cloud Environment via Vagrant to Secure Cloud Hosts . Computer and Network Security Essentials. Springer, 2018. P. 539–553. Doi: 10.1007/978-3-319-58424-9_31
6. Hashimoto M. Vagrant: Up and Running: Create and Manage Virtualized Development Environments . O'Reilly Media Inc, 2013.
7. Mouat A. Using Docker: Developing and Deploying Software with Con-tainers . O'Reilly Media Inc, 2016.
8. Sammons G. Learning Vagrant: Fast programming guide – CreateSpace Independent Publishing Platform, 2016.
9. Peacock, M. Creating Development Environments with Vagrant . Packt Publishing Ltd, 2015.
10. Iuhasz G., Pop D., Dragan I. Architecture of a scalable platform for monitoring multiple big data frameworks . *Scalable Computing: Prac-tice and Experience*. 2016. V. 17. No. 4. P. 313-321. Doi: 10.12694/scpe.v17i4.1203
11. Nikulchev E., Ilin D., Kolyasnikov P., Belov V., Zakharov I., Malykh S. Programming Technologies for the Development of Web-Based Platform for Digital Psychological Tools . *International journal of advanced computer science and applications*. 2018. Vol. 9. No. 8. P. 34-45. Doi:10.14569/IJACSA.2018.090806
12. Kashyap S., Min C., Kim T. Opportunistic spinlocks: Achieving virtual machine scalability in the clouds . *ACM SIGOPS Operating Systems Review*. 2016. Vol. 50. No. 1. P. 9-16. Doi: 10.1145/2903267.2903271.
13. Saikrishna P. S., Pasumarthy R., Bhatt N. P. Identification and multivari-able gain-scheduling control for cloud computing systems . *IEEE Trans. Control Sys. Technol*., 2016, Vol.25, no.3, pp.792-807.
14. Савельев А.О. Введение в облачные решения Microsoft. Курс лекций. 2-е издание, исправленное. М.: НОУ Интуит, 2016.

References

1. Karr, N. (2014). Velikij perehod: Chto govorit revoliusia oblashnyh tehnologij Elektron. dan. M.: Mann, Ivanov i Ferber [in Russian].
2. Andreevskij, I.L. (2018). Tehnologii ooblachnyh vychislenij . Spb.: Sankt-Peterburjskij gosudarstvennyj ekonomicheskij universitet [in Russian].
3. Caballer, M., Blanquer, I., Moltó, G. & de Alfonso C. (2015). Dynamic management of virtual infrastructures . *Journal of Grid Computing. V. 13. No. 1*. P. 53–70. Doi: 10.1007/s10723-014-9296-5 [in English].
4. Giannakopoulos I., Konstantinou I., Tsoumakos D. & Koziris N. (2018). Cloud application deployment with transient failure recovery . *Journal of Cloud Computing. V. 7. No. 1. Art. no. 11*. Doi: 10.1186/s13677-018-0112-9 [in English].
5. Spanaki, P. & Sklavos, N. (2018). Cloud Computing: Security Issues and Establish-ing Virtual Cloud Environment via Vagrant to Secure Cloud Hosts . Computer and Network Security Essentials. Springer, P. 539– 553. Doi: 10.1007/978-3-319-58424-9_31 [in English].
6. Hashimoto, M. (2013). Vagrant: Up and Running: Create and Manage Virtualized Development Environments – O'Reilly Media Inc. [in English].
7. Mouat A. (2016). Using Docker: Developing and Deploying Software with Con-tainers .O'Reilly Media Inc. [in English].
8. Sammons G. (2016). Learning Vagrant: Fast programming guide . CreateSpace Independent Publishing Platform. [in English].
9. Peacock, M. (2015). Creating Development Environments with Vagrant . Packt Publishing Ltd. [in English].

10. Iuhasz, G., Pop, D. & Dragan, I. (2016). Architecture of a scalable platform for monitoring multiple big data frameworks . *Scalable Computing: Prac-tice and Experience*. *V. 17. No. 4*. P. 313-321. Doi: 10.12694/scpe.v17i4.1203 [in English].
11. Nikulchev, E., Ilin, D., Kolyasnikov, P., Belov, V., Zakharov, I. & Malykh, S. (2018). Programming Technologies for the Development of Web-Based Platform for Digital Psychological Tools . *International journal of advanced computer science and applications*. *V. 9. No. 8.* P. 34-45. Doi:10.14569/IJACSA.2018.090806 [in English].
12. Kashyap. S., Min. C. & Kim. T. (2016). Opportunistic spinlocks: Achieving virtual machine scalability in the clouds . *ACM SIGOPS Operating Systems Review*. *V. 50. No. 1.* P. 9-16. Doi: 10.1145/2903267.2903271 [in English].
13. Saikrishna, P. S., Pasumarthy, R. & Bhatt, N. P. (2016). Identification and multivari-able gain-scheduling control for cloud computing systems *IEEE Trans. Control Sys. Technol*., *Vol.25, no.3*, pp.792-807.
14. Saveljev A.O. (2016). *Introduction to Microsoft Cloud Solutions*.. (2d ed.). Moskow :NOU Intuit [in Russian].

Roman Minailenko, Assoc. Prof., PhD tech. sci., **Vitalii Reznichenko,** lecturer**, Oksana Konoplitska-Slobodenyuk,** lecturer, **Liudmyla Polishchuk,** Senior Lecturer
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine

Overview of Load Balancing Methods in Cloud Systems

Cloud systems are currently the most popular concept of information systems and are the result of the evolution of a chain of methods for their construction. The main task of cloud technologies is to create a virtual cloud system consisting of virtual distributed resources. These resources provide remote provisioning of cloud access services with the required level of customer service

Analysis and load balancing in cloud systems is quite an urgent task, as most open access cloud systems use simple load schedulers for their physical servers.

The problem of load balancing requires a solution not when the server unexpectedly failed in the process of working on the task, which discourages users from using such a product, but at the very beginning of the project. In the early stages of design, it is acceptable to increase capacity by connecting new servers or using code optimization algorithms. But when a certain limit is reached, these measures become insufficient.

The article reviews the methods of load balancing in cloud systems. It is shown that the existing methods of load balancing of cloud systems have limited use and currently there is no universal load balancing system. In addition, none of the considered methods takes into account such important components of systems as network and disk subsystem. Load balancing methods for cloud systems require improvement, the purpose of which should be the ability to fully monitor the system to meet the requirements of users and developers.

cloud systems, computing resources, load balancing, performance

Одержано (Received) 27.10.2021 *Прорецензовано (Reviewed) 04.11.2021*
Прийнято до друку (Approved) 29.11.2021