

МОДИФІКОВАНИЙ МЕТОД УПРАВЛІННЯ ІНФРАСТРУКТУРОЮ МЕРЕЖІ ПРОМИСЛОВОГО ІНТЕРНЕТУ РЕЧЕЙ

Давидюк А.М., Курдеча В.В.

Інститут телекомунікаційних систем

КПІ ім. Ігоря Сікорського, Україна

E-mail: mr.wokawoka@gmail.com

Modified network infrastructure management method for industrial internet of things

Today there is a problem with automation and work simplification for data warehouses. This problem is also acute for the industrial Internet of Things networks, as these networks operate on a large amount of data. Our solution offers the creation of a cluster with non-relational databases and containerization of the databases themselves as a fast, cheap and reliable way to deploy the environment

Причина виникнення проблеми

На сьогоднішній день постає проблема з автоматизацією і спрощенням роботи з сховищами даних. Ця проблема також гостро постає для мереж промислового Інтернету речей, оскільки ці мережі оперують великою кількістю даних.

Результати показують, що кластеризація компонентів бази даних зменшить час обробки даних та забезпечить більшу цілісність даних. А контейнеризація допомагає спростити адміністрування та оптимізувати ресурси.

З розвиненим технологічним рівнем та відповідною доступністю обчислювальних пристроїв зменшується ризик відмови систем системи, а також збільшується надійність збереження даних.

Загалом, кластеризація в архітектурі сервера може бути універсальним варіантом розповсюдження, а також збереження цільових даних у разі відмови. Таким чином отримавши широкі доповнення до сучасних систем. На всіх рівнях трирівневої структури IoT. Контейнеризація ж дозволить зменшити кількість виділених ресурсів для кожного вузла кластеру, а налаштування алгоритму шаблонного розгорнення контейнерів дозволить зробити адміністрування та управління мережею автономними та зручними у налаштуваннях.

То ж наше рішення пропонує створення кластеру з нереляційними базами даних та контейнеризація самих баз даних, як швидкий, дешевий та надійний спосіб розгорнути середовище.

Кластеризація нереляційних баз даних

Найпопулярнішою робочою моделлю для врівноваження навантаження баз даних NoSQL є заточення (або фрагментація). При фрагментарному підході база даних поділяється на фрагменти (рис. 1), які записують і зчитують дані, що може значно підвищити швидкість роботи з базою даних.

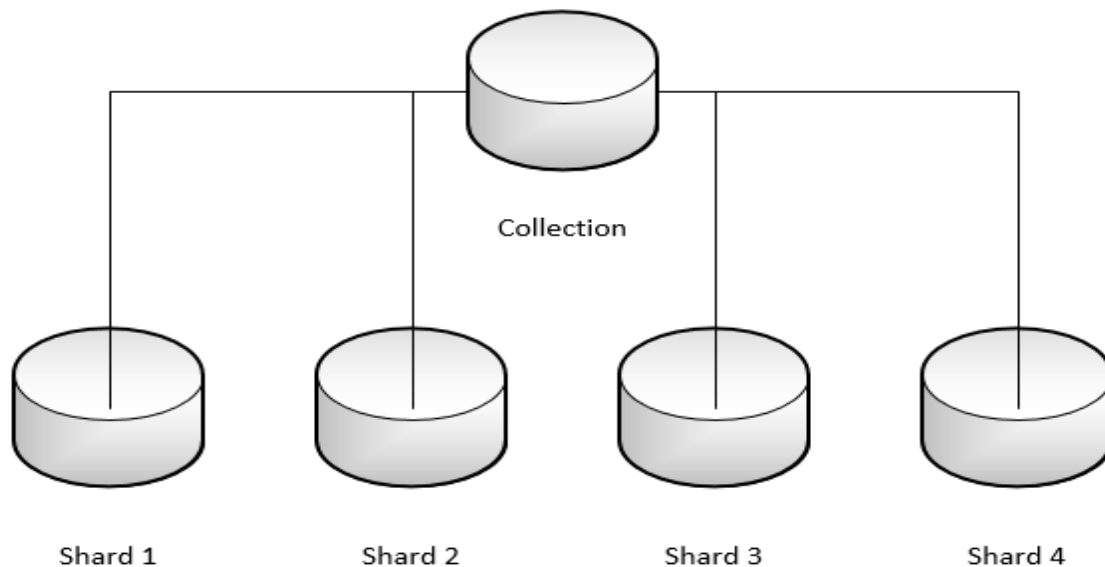


Рис. 1. Узагальнена структура кластерів бази даних, що не мають відношення, на основі дробового типу реплікації.

Алгоритм контейнеризації та управління

У звичайних операційних системах для персональних комп'ютерів комп'ютерна програма може бачити (навіть якщо вона не може отримати доступ) усі системні ресурси. Вони включають:

- Апаратні функції, які ви можете використовувати, такі як процесор та мережеве з'єднання;
- Дані, які можна читати або записувати, наприклад файли, папки та мережеві папки;
- Підключені периферійні пристрої, з якими він може взаємодіяти, наприклад веб-камера, принтер, сканер або факс.

За допомогою віртуалізації або контейнеризації операційної системи ви можете запускати програми в контейнерах, які виділяють лише частини цих ресурсів і не завантажують не потрібних компонентів операційної системи.

Контейнеризація аналогічна віртуалізації додатків: в останньому лише одна комп'ютерна програма розміщується в ізольованому контейнері, а ізоляція стосується лише файлової системи. Тому контейнеризація є перспективним підходом до нашого рішення, оскільки вона дозволить нам налаштувати контейнер найпростішим способом, а потім просто встановити оптимальну кількість контейнерів для збільшення або зменшення необхідних ресурсів.

Це дозволить нашому рішення імітувати фрагменти шардінгу NoSQL з динамічним створенням / видаленням вузлів баз даних. На основі результату, показаного на рис.2, найбільш перспективними базами даних для високонавантажених систем IoT може бути Couchbase.

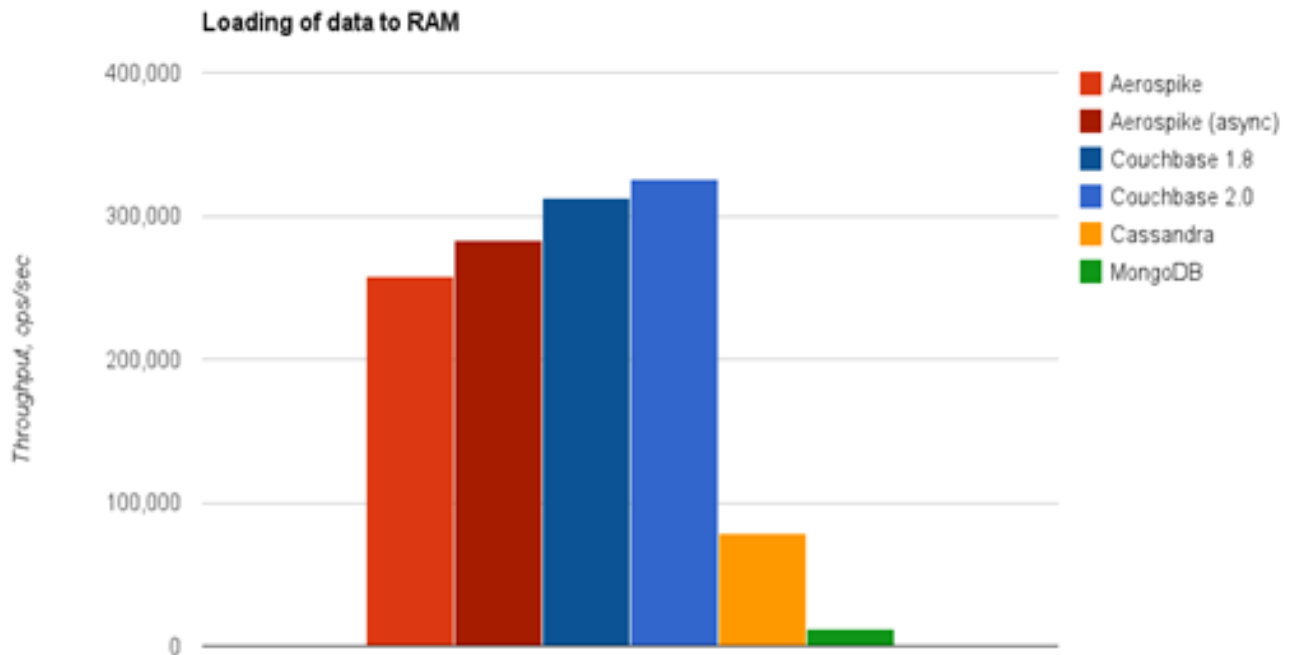


Рис. 2. Завантаження діаграм даних для деяких NoSQL DB.

В якості прикладу візьмемо сервер Couchbase, який може забезпечити 300 000 операцій запису в секунду і мережу, яка може генерувати, наприклад, 200 000 сигналів в секунду з певним граничним коефіцієнтом $\delta = 0,8$, і якщо кількість запитів зростатиме з часом і коли вона досягне до 280 000 запитів в секунду, система розгорне новий контейнер, який буде брати на себе частину навантаження.

А у випадку зменшення трафіку згорне останній і перенесе копію даних на резервну базу.

Висновки. Підсумовуючи вищеописане, для швидкості розгортки та реплікації серверів даних є кілька підходів, які можна об'єднати. Тому наше рішення дозволить розбити систему на автоматизовані контейнери, що будуть працювати по кластерній моделі, що дозволить швидко згорнути і розгорнути додаткові вузли і економити час та гроші замовника.

Література

1. X. Tao and C. Ji, "Clustering massive small data for IOT," The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014), Shanghai, 2014, pp. 974-978. doi: 10.1109/ICSAI.2014.7009427.
2. C. You and X. Wu, "Feature selection embedded subspace clustering with lowrank and locality constraints," 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 2018, pp. 1-8. doi: 10.1109/ISC2.2018.8656922.
3. H. Hromic et al., "Real time analysis of sensor data for the Internet of Things by means of clustering and event processing," 2015 IEEE International 11 Conference on Communications (ICC), London, 2015, pp. 685-691. doi: 10.1109/ICC.2015.7248401.
4. K. Lin, D. Wang, F. Xia and H. Ge, "Device Clustering Algorithm Based on Multimodal Data Correlation in Cognitive Internet of Things," in IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2263-2271, Aug. 2018. doi: 10.1109/JIOT.2017.2728705.