

ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ДАНИХ В МЕРЕЖАХ ПРОМИСЛОВОГО «ІНТЕРНЕТУ РЕЧЕЙ»

Давидюк А.М., Курдеча В.В.

Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна

E-mail: mr.wokawoka@gmail.com

Providing data integrity in the industrial “Internet of things” networks

Nowadays, for data speed and replication, there are two types of databases that have their own advantages and disadvantages like SQL and NoSQL databases. However, with the development of the systems itself, there were solutions that allow to qualitatively improve the performance of the data by distributing the load, separating the roles of the components of databases or replicating data from one node to another.

The results show that clusterization of the database components will increase the time of data processing and provides the data integrity.

Проблематика роботи з даними. З розвитком технологічного рівня та відповідно збільшенням доступності обчислюваних приладів виникла проблема забезпечення швидкого відклику високонавантажених систем, а також збільшення надійності в збереженні і цілісності даних зі збільшенням їх кількості. Тому, не дивно, що з'явилась велика кількість рішень що дозволяють забезпечити вищеприписані вимоги за допомогою маніпулювання роботи з даними.

Взагалі, кластеризацію в серверній архітектурі можна вважати універсальним варіантом для розподілення навантаження та збереження цілісності даних у випадку відмови системи. Тому вона набула широкого поширення у сучасних системах не залежно від типу використовуваних баз даних. Кластеризацію застосовують у мережевому рівні тривірневої архітектури IoT.

Мабуть однією з головних частин цього рівня системи являється база даних, що містить в собі дані зібраних з усіх пристроїв мережі, тому їй потрібно приділяти достатню увагу при розробці і супроводі системи. Взагалі дані можуть зберігатися по різному, але в наш час набули популярності два типи баз даних – реляційні та нереляційні (далі SQL та NoSQL відповідно). Який з них використовувати і поєднувати все залежить від типу даних що буду зберігатися і роботою з ними, наприклад для зберігання масивів великих даних з однорідною структурою без здійснення операцій над ними краще використовувати NoSQL, а для побудови логічної структури, здійснення різних родів операцій і перенесення частини логіки на саму базу даних краще використовувати SQL.

Кластеризація реляційних баз даних. В залежності від вибраного типу бази даних, або їх комбінацій, до прикладу зберігати необроблені дані з пристроїв на нереляційних базах, а оброблену і структуровану інформацію на реляційних, потрібно вибрати свою модель кластеризації та відмовостійкості баз даних. Наприклад сьогодні для реляційних баз Microsoft SQL існують два найпопулярніші підходи це розгортка звичайного Clustered SQL (рис.1) Server або ж Always On SQL Server, що в залежності від вибраних налаштувань конфігурації дозволять покращувати, або ж ефективність та швидкість роботи з даними, а бо ж їх надійне збереження, у випадку забезпечення високого рівня цих обох показників розгортаються хмарні сервіси, але мінусом таких рішень являється їх ціна.

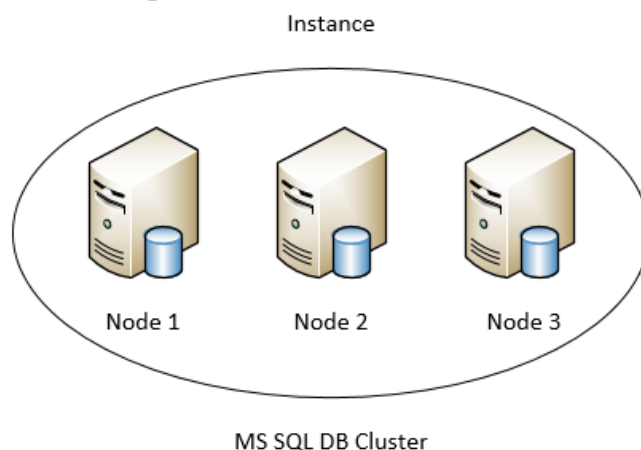


Рис. 1. Узагальнена структура кластера реляційної бази даних.

Кластеризація нереляційних баз даних. У випадку використання нереляційних баз даних, на прикладі Mongo DB також існують два підходи, один з них має більшу продуктивність, інший кращу реплікацію даних, це так звані осколковий підхід і підхід реплікації. При осколковому підході база даних ділиться на осколки (Shards рис.2) на якій йде запис та зчитування даних, що дозволяє значно підвищити швидкість роботи з базою даних. Для більшої ж надійності використовуються підхід репліки або реплікації рис.3. Суть підходу лежить в тому, що для забезпечення цілісності бази даних всі дані дублюються на кожний вузол. При цьому для збільшення швидкості зчитування не тільки головний вузол може бути використаний для зчитування даних.

Висновки. Отже, нині для забезпечення швидкості роботи з даними та їх реплікації існує два типи баз даних, які мають свої переваги та недоліки. Проте з розвитком самих систем з'явилися рішення що дозволяють якісно покращити характеристики роботи з даними шляхом розподілення навантаження, розмежування ролей компонентів баз даних або ж реплікації даних з одного вузла на інший.

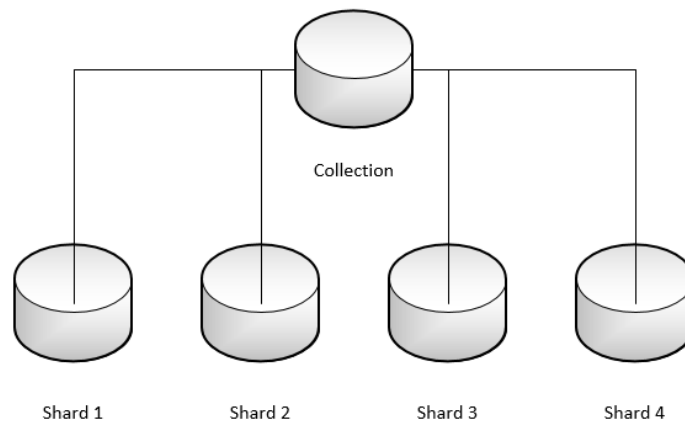


Рис. 2. Узагальнена структура кластера нереляційної бази даних побудованого на осколковому типу реплікації.

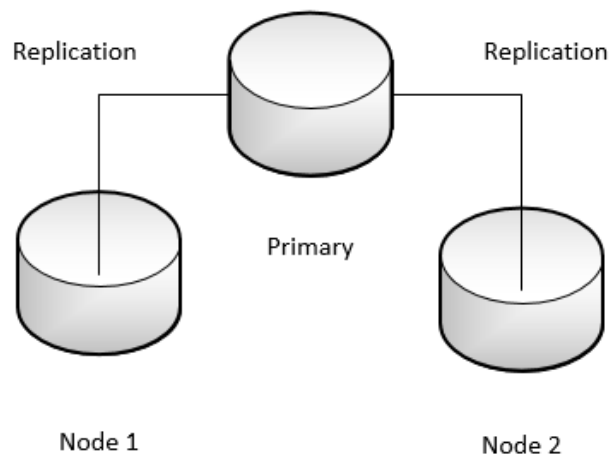


Рис. 3. Узагальнена структура кластера нереляційної бази даних побудованого на звичайній реплікації.

Література

1. S. A. Hinai and A. V. Singh, "Internet of things: Architecture, security challenges and solutions," 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), Dubai, 2017, pp. 1-4.
2. Database cluster and load balancing [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.com/questions/1163216/database-cluster-and-load-balancing>.
3. Sharded Cluster Components[Електронний ресурс] – Режим доступу до ресурсу: <https://docs.mongodb.com/manual/core/sharded-cluster-components/>.
4. Failover Clustering and Always On Availability Groups (SQL Server) [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/sql/database-engine/availability-groups/windows/failover-clustering-and-always-on-availability-groups-sql-server?view=sql-server-2017>.