UDC 004.415

IMPROVED CLUSTER MANAGEMENT METHOD FOR INDUSTRIAL “INTERNET OF THINGS” NETWORKS

Andriy M. Davydiuk, Andrii A. Astrakhantsev

Institute of Telecommunication Systems
Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine

Background. With growing technological level and the corresponding availability of computational devices, sharply appears the problem of providing the high availability of high-loaded systems, as well as the reliability of the data storage and integrity. Therefore, it is not surprising that a large number of solutions have put high demands to the data manipulation.

Objective. The aim of the paper is to create a generic algorithm for industrial IoT networks that will be used for managing containerization components in database clusters.

Methods. During the work process most of existing solutions for providing automatic scalability of database cluster have been analyzed. Based on this analysis a special algorithm that manipulates the images of database cluster nodes based on predefined input parameters was proposed.

Results. Algorithm for managing database clusters in industrial IoT networks that will allow clients to optimize the resource usage with saving availability and incoming throughput characteristics of the cluster.

Conclusions. On the first stage, using this approach in production solutions requires more preparation and administration expenses, compared with existing solutions, but with time, it will give an advantage because the system will fully support the cluster to keep it in optimal condition without human interruption.

Keywords: clusterization; containerization; industrial IoT networks.

Introduction

With growing technological level and the corresponding availability of computational devices, sharply appears the problem of providing the high availability of high-loaded systems, as well as the reliability of the data storage and integrity. Therefore, it is not surprising that a large number of solutions have providing high demands with the data manipulation.

In general, clustering in the server architecture can be a universal option for distribution, as well as a powerful tool of saving data from losses. Moreover, on modern market we can see a new solution that propose us a better space optimization, scalability and security – containerization. In addition, it seems that those solutions are good enough for integration on server layer of IoT systems.

Perhaps one of the main parts of this system is a database containing data collected from all devices in the network, so it needs to pay sufficient attention onto the development and maintenance of the system. Generally, data can be stored in a different way, but today we have a two popular types of databases – relational and non-relational (hereinafter SQL and NoSQL respectively). Which of them to use and combine is all depends on the type of data that will be stored.

For example, to store arrays of large data with a homogeneous structure without taking some operations on them, it is better to use NoSQL, but for constructing the logical structure, and for using data manipulation or even transfer a part of solution logic into the database layer the classic SQL will be the best solution [1].

Our solution focuses on storing data from end devices and based on described above the main purpose of this work is combine clusterization with containerization. That will help to reach improvements in scalability, availability and security and to create an algorithm that can be configured to provide optimal cluster node management for concrete industrial IoT network.

I. Existing solution overview.
I.1 Clustering relational databases

Depending on the type of database, which were selected, or their combinations, for example, store unprocessed data from devices on non-relational databases, and processed and structured information on relational ones, need to choose own clustering model and fault-tolerance databases [2]. For example, for Microsoft SQL Relational databases, there are two most commonly used approaches [3]. The first one is scalability of the usual Clustered SQL Server (Fig. 1).

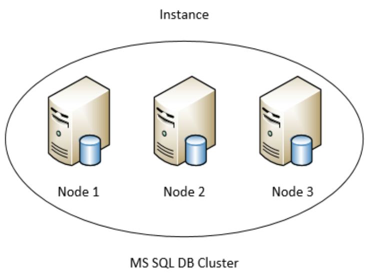


Fig.1 Generalized cluster structure of the relational database

The second one is Always On SQL Server, which, depending on the configuration settings were chosen, can improve efficiency and speed of the data throughput. The pros of this solution are a good level of scalability and high availability.

I.2 Clustering of non-relational databases

The most popular working model for load balancing of NoSQL databases is a sharding (or fragmentation) [4]. In a fragmentation approach, the database is divided into fragments (Fig. 2), which records and reads data, which can significantly increase the speed of working with the database.

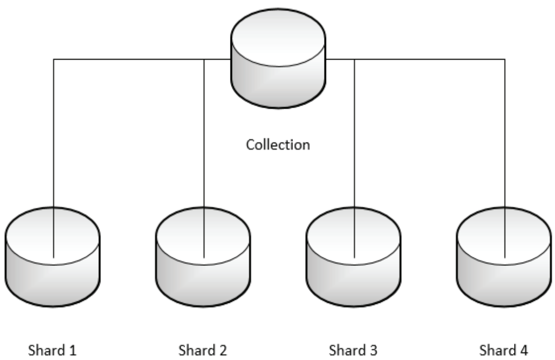


Fig.2 A generalized cluster structure of a nonrelation database based on a fractional type of replication

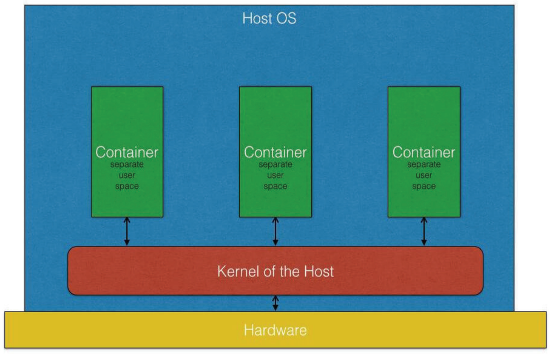
I.3 Containerization and control algorithm

In normal operating systems for personal computers, a computer program can see (even if it cannot access) all system resources. They include:

- Hardware features that used, such as a processor and a network connection
- Data that can be read or written, such as files, folders, and network folders
- Connected peripherals with which it can interact, such as a webcam, printer, scanner, or fax

With the operating system virtualization or containerization, someone can run apps within the containers, which allocate only parts of these resources. A program that expects to see the entire computer, when it launches inside the container, can only see selected resources and find them available. Several containers can be created on each operating system, each with a subset of computer resources. Each container may contain any number of computer programs. These programs can run simultaneously or separately, even interact with each other.

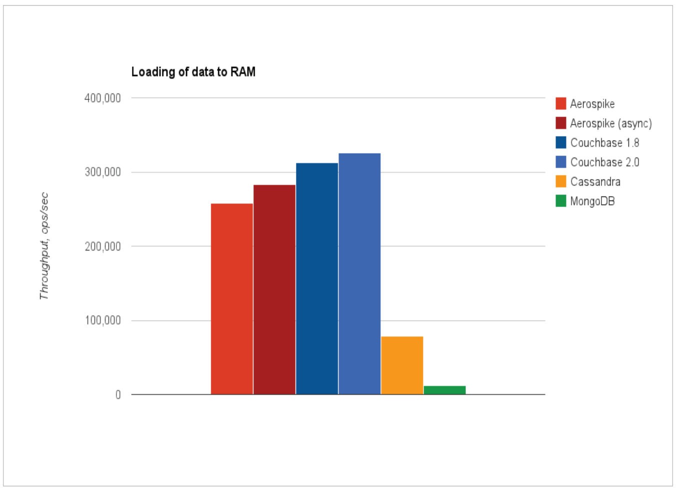
Based on the result shown on Fig.4 the most perspective databases for high loaded systems could be the Couchbase. In addition, we can a see a dynamic how more productive non-relational database can be, as less SQL features it supports, comparing Mongo and Aerospike.



Operating System/Container Virtualization

Fig.3 A generalized cluster structure of a non-relational database constructed on a regular replication

NoSQL databases have optimal throughput characteristics and they are suitable for using in industrial IoT with great number of end node devices. In addition, many vendors have built in features for containerization and clusterization.



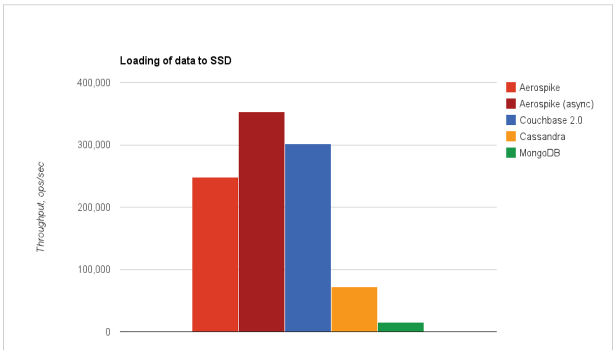


Fig. 4. Loading data diagrams for some NoSQL Db [6]

Let introduce a short list of common solutions:

1. *Sharding from Mongo.* Mongo used a replica as a guarantor of fault tolerance. A replica set is a group of at least three MongoDB instances that maintain the same data. One node of the set is deemed primary and is responsible for all write operations. It records all changes in the oplog so that the remaining nodes (secondaries) can accurately reflect the primary's data. If the primary becomes unavailable, a new one will be automatically elected from the active secondaries after a short delay.

2. *Couchbase Clustering*. The Couchbase cluster size is automatically adjusted based on incoming load by changing the DB servers number (up to 10 instances per layer) according to the following conditions:

+1 node if CPU/RAM usage is >70% for at least 5 minutes

-1 node if CPU/RAM usage is <40% for at least 5 minutes

When a node is added to or removed from the cluster, the process of data rebalancing is automatically handled. It is aimed to evenly re-distribute all the information, stored within a cluster, across the available nodes. Herewith, the cluster remains up and continues to serve and handle client requests.

3. *Master-Slave*. Cluster that uses a master-slave model has one main node and a set of slave nodes that are usually used to save the data copy and in case of fall of master node the one of slave nodes will take all master's tasks.

Proposed solution

For automatization of the cluster was implemented a special algorithm for analyzing the incoming data flow, which will manage cluster nodes automatically without interference from administrators, in this solution it is proposed to use a formula that will control the number of database nodes in the cluster.

$$k = \lceil \frac{E}{e} \rceil + \lambda \qquad (1)$$

where: E – input data stream at a time; e – the amount of data that can be processed by the node, λ – coefficient for dynamic change of the number of cluster nodes. Therefore, based on (1), the maximum bandwidth of the system will be equal to:

$$E_{max} = e * k \qquad (2)$$

To determine the final dependence of λ must be first determined the average change in incoming messages over a period:

$$\Delta E_{mid} = \sum_{i}^{n} (E_{max} - E_{mom\,i}) / n \qquad (3)$$

where $E_{mid\,i}$ data flow at a certain point in time, n – the number of time segments for which the measurement was performed.

It is also necessary for each system to determine the approximate time spent on a full system deploy, it will be the sum of time spent on creating / deleting a container (t_c), time spent on joining the node to the cluster (t_d) and the time spent on balancing the cluster after creating / deleting nodes (t_{cr}).

$$t_f = t_c + t_d + t_{cr} \qquad (4)$$

Based on (2) – (4) determination of the coefficient λ, at moment E_{mom} can be expressed:

$$\lambda = \{ \frac{(E_{max} - E_{mom})}{\frac{|\Delta E_{mid}|}{t_f}} \leq \delta \text{ та } \Delta E_{mid} > 0$$
$$= 1, \frac{(E_{max} - E_{mom})}{\frac{|\Delta E_{mid}|}{t_f}} \geq \delta \text{ та } \Delta E_{mid}$$
$$< 0 = -1, \quad 0$$

The limit level is a natural whole δ >1, it is proposed to choose as the optimal ratio of the predicted scan time to the real scan time.

For example, if the server want to be deployed as soon as possible before the maximum bandwidth is reached, or on the contrary, to be minimized when the bandwidth drop to $E_{max} - e$ then δ = 1.

Conclusion

In this paper proposed an algorithm for managing cluster industrial IoT networks that will allow clients to optimize the resource usage with saving availability and incoming throughput characteristics of the cluster. On the first stage, developed solution requires a more preparation and administration expenses, but with time, it will give an advantage because the system will fully support cluster to keep it in optimal condition without human interruption.

References

1. Hinai S. A.and Singh A. V., 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. Handoko H., Isa S.M. High Availability Analysis with Database Cluster, Load Balancer and Virtual Router Redundancy Protocol, 2018 3rd International Conference on Computer and Communication Systems.
3. Database cluster and load balancing – Link to resource: https://stackoverflow.com/questions/1163216/database-cluster-and-load-balancing
4. Sharded Cluster Components– Link to resource: https://docs.mongodb.com/manual/core/sharded-cluster-components/
5. Failover Clustering and Always On Availability Groups (SQL Server) – Link to resource: 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
6. Testing the productivity of NoSQL DBs – Link to resource: https://xakep.ru/2014/01/11/nosql-bd-test/
7. Xuan Z. Load Balancing Technology Based On Consistent Hashing For Database Cluster Systems, 2012, DOI: 10.2991/citcs.2012.157

Давидюк А.М., Астраханцев А.А.
Вдосконалений кластеризований метод управління для мереж промислового "Інтернету речей"

Проблематика. Зі зростанням технологічного рівня та відповідної доступності обчислювальних пристроїв різко постає проблема забезпечення високої доступності високонавантажених систем, а також надійності зберігання та цілісності даних. Тому не дивно, що велика кількість рішень поставила високі вимоги до обробки даних.

Мета дослідження. Створення загального алгоритму для промислових мереж IoT, який буде використовуватися для управління компонентами контейнеризації в кластерах баз даних.

Методика реалізації. В процесі роботи було проаналізовано більшість існуючих рішень для забезпечення автоматичної масштабованості кластера баз даних. На основі цього аналізу було запропоновано спеціальний алгоритм, який маніпулює зображеннями вузлів кластера баз даних базуючись на заздалегідь визначених вхідних параметрів системи.

Результати досліджень. Алгоритм управління кластерами баз даних у промислових мережах IoT, що дозволить клієнтам оптимізувати використання ресурсів із збереженням доступності та вхідної пропускної здатності кластера.

Висновки. На першому етапі використання цього підходу у виробничих рішеннях вимагає більших витрат, ніж існуючі рішення, на підготовку та адміністрування, але з часом це дасть перевагу, оскільки система буде повністю підтримувати кластер, щоб підтримувати його в оптимальному стані без втручань з боку людини.

Ключові слова: кластеризація; контейнеризація; промислова мережа Інтернету речей.

Давидюк А.М., Астраханцев А.А.
Усовершенствованный кластеризованный метод управления для сетей промышленного "Интернета вещей"

Проблематика. С ростом технологического уровня и соответствующей доступности вычислительных устройств резко встает проблема обеспечения высокой доступности высоконагруженных систем, а также надежности хранения и целостности данных. Поэтому неудивительно, что большое количество решений поставила высокие требования к обработке данных.

Цель исследований. Создание общего алгоритма для промышленных сетей IoT, который будет использоваться для управления компонентами контейнеризации в кластерах баз данных.

Методика реализации. В процессе работы было проанализировано большинство существующих решений для обеспечения автоматической масштабируемости кластера баз данных. На основе этого анализа было предложено специальный алгоритм, который манипулирует изображениями узлов кластера баз данных основываясь на заранее определенных входных параметров системы.

Результаты исследований. Алгоритм управления кластерами баз данных в промышленных сетях IoT, что позволит клиентам оптимизировать использование ресурсов с сохранением доступности и входящей пропускной способности кластера.

Выводы. На первом этапе использования этого подхода в производственных решениях требует больших затрат, чем существующие решения, на подготовку и администрирования, но со временем это даст преимущество, поскольку система будет полностью поддерживать кластер, чтобы поддерживать его в оптимальном состоянии без вмешательств со стороны человека.

Ключевые слова: кластеризация; контейнеризация; промышленная сеть Интернета вещей.