УДК 004.272:004.422.635.5 DOI: 10.31673/2412-9070.2024.031822

V. KUZMINYKH, PhD, assoc. professor, ORCID: 0000-0002-8258-0816,
B. XU, postgraduate, ORCID: 0000-0003-1430-5334,
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv

THE INFLUENCE OF CURRENT RESULTS IN A EVENT-ORIENTED DATA COLLECTION SYSTEM

The article discusses synchronous and asynchronous procedures in the implementation of the microservices management algorithm in an adaptive system for processing large data flows when collecting information on the main event-oriented approach in the implementation of the architecture of a software system that processes information in real-time. This approach is important when processing large volumes of data from heterogeneous information sources, especially when the task is to minimize the total processing time of large data streams and reduce the total number of calls to data sources. The proposed approach makes it possible, based on its adaptability, to manage the selection of the composition and the number of calls of microservices to the sources, by the events that occur during the collection of information. Thus, it is possible to determine the choice of sources based on the assessment of the effectiveness of obtaining relevant information from them. This is especially important when processing large data flows from heterogeneous information sources when the task is to minimize the total time of collection and processing of large data flows. In turn, this poses the task of minimizing the number of requests to information sources to obtain a sufficient number of data units that are relevant to the search query. The creation of effective big data processing systems requires constant development of approaches to the architecture of building software applications. The event-oriented microservice architecture of the system makes it possible to adapt the operation of the system to the loads on individual microservices and the efficiency of their work by forming and responding to relevant events based on the analysis of relevant events that occur during the collection and initial processing of the received data. Depending on the specific task, it is possible to use both a synchronous and an asynchronous microservice management algorithm. The article provides an analysis of the effectiveness of obtaining relevant data depending on the degree of consideration in the evaluation of results and the formation of events, both the depth of the previous history of the results of requests and the size of their quantitative impact. The use of event-oriented microservice architecture can be especially effective when developing various information and analytical systems that need, according to user requests, to access various sources of information, analyze their data for relevance according to the request, and process large data streams.

Keywords: microservices; adaptation; event-driven architecture; big data.

Introduction

Data collection based on the processing of a large number of information sources is considered one of the standard methods for collecting information in various spheres of activity of modern society, which is related to both scientific and social, military, and similar other studies. The use of electronic information carriers to a large extent determines the approaches and methods of directed search and increases the efficiency of both individual procedures and information search in general.

The amount of information is constantly growing from year to year, which leads to problems of orderly storage and purposeful search for the necessary data. In proportion to the number of sources and volumes of information, the complexity of the processing task increases significantly. With a very large number of electronic materials, soon it will simply be impossible to find the necessary information without using effective information processing methods [1].

Consolidation of information [2] based on the use of highly effective methods of searching for information relevant to the request using adaptive management procedures of the microservice architecture of the software system can help in solving these problems. In a broad sense, consolidation can be understood as the process of searching, selecting, analyzing, structuring, transforming, storing, cataloging, and providing the consumer with information on given topics. The task of information consolidation is one of the most important tasks of processing large volumes of data [3].

With the increase in the amount of information, the number of sources of information, scientific publications, and electronic libraries also increases, which has an impact on approaches to finding the necessary information. The best of such sources limit access to data repositories for monetization to support authors and support systems, this leads to significant costs during constant high-volume searches for information.

Different sources of information have different content, which makes it logical to give preference to those that contain a larger amount of sought-after information to increase the efficiency of the search and reduce the cost of paying for access to electronic sources of information.

In some cases, data consolidation is the initial stage of the implementation of an analytical task or a project of a software information and analytical system [4]. One of the main processes of consolidation is the collection and transformation of data for their further efficient processing in the system. Also, an important process is the evaluation of the quality of the data obtained from the sources.

The main results that data consolidation should provide for further comprehensive processing are:

- compact storage of large volumes of data;
- maintaining the integrity of the data structure;
- high-speed access to large volumes of data;
- control of data relevance.

In order to effectively search for data from a large number of information sources, it is necessary to consolidate data using specialized software tools that will provide:

- effective selection of the most relevant sources of information;
- the possibility of accumulating information about the correspondence of sources during the execution of the request;
- analysis of both the most promising from the point of view of relevance and less relevant sources;
- taking into account the possibility of changing the state of the sources upon repeated request;
- taking into account the information assessment of promising sources from the point of view of relevance for their arrangement.

The consolidation process is a set of methods and procedures aimed at extracting data from various sources, ensuring the necessary level of their informativeness and quality, transforming them into a single format in which they can be loaded into a data warehouse or analytical system.

Very often, when implementing various analytical tasks, consolidation is considered as the initial stage of implementation [5; 6]. The basis of consolidation is the process of collecting and organizing data storage in a form that is optimal from the point of view of their processing on a specific analytical platform or solving a specific analytical problem. Accompanying tasks of consolidation are the assessment of data quality and their enrichment, with the aim of reducing the amount of information to be processed in an information-search or information-analytical system.

The main results that data consolidation should provide for further in-depth processing are:

- high-speed access to large volumes of data;
- compact storage of large volumes of data;
- maintaining the integrity of the data structure;
- control of consistency and relevance of data.

A key concept of consolidation is a data source - a data store containing structured data that can be useful for solving an analytical problem. It is necessary that the used analytical platform can access the data from this object directly or after its conversion to another format [6].

Search plays a key role in the consolidation process. Consolidation systems are built around the task of aggregated search of query-relevant data from a large number of sources. Therefore, the quality of the consolidation system in general depends on the correct choice of search methods.

Main part

The tasks of searching and processing data from various sources very often require highly efficient extraction of data from selected information resources. Such a task has several important aspects for development and analysis, which affect the possibility of significantly improving the efficiency and cost of the collection.

This can be particularly important in terms of the cost of performing individual operations and the overall speed of data acquisition.

At the same time, the cost parameters refer to the cost:

- access time to resources (data sources);
- the number of requests to data sources that provided the necessary information relevant to the request;
- use of communication channels with data sources;
- use of data collection technical support resources.

In addition, there are many tasks based on information gathering, in which the main characteristics are related to the time of data acquisition and are related to the need to obtain information in real time [7].

Therefore, the task of building effective access methods and algorithms that make it possible to reduce the overall cost characteristics can be very important.

We can distinguish three main approaches to solving the problem of data collection from various distributed sources of information, which are built based on the use of microservice architecture. The general structural scheme of solving the problem of data collection from various extensive sources of information is shown in Fig. 1.

First, there are methods of sequential extraction of data from a group of information sources. At the same time, data extraction is carried out sequentially with each of the sources after the end of data extraction from the previous source [8]. In this way, a queue of sources is established, which is determined based on the processing of information search results, which are determined by evaluating the number of relevant data among the total number of retrieved records.

The second is methods of parallel extraction of data from a group of information sources. Parallel extraction methods are more efficient than sequential

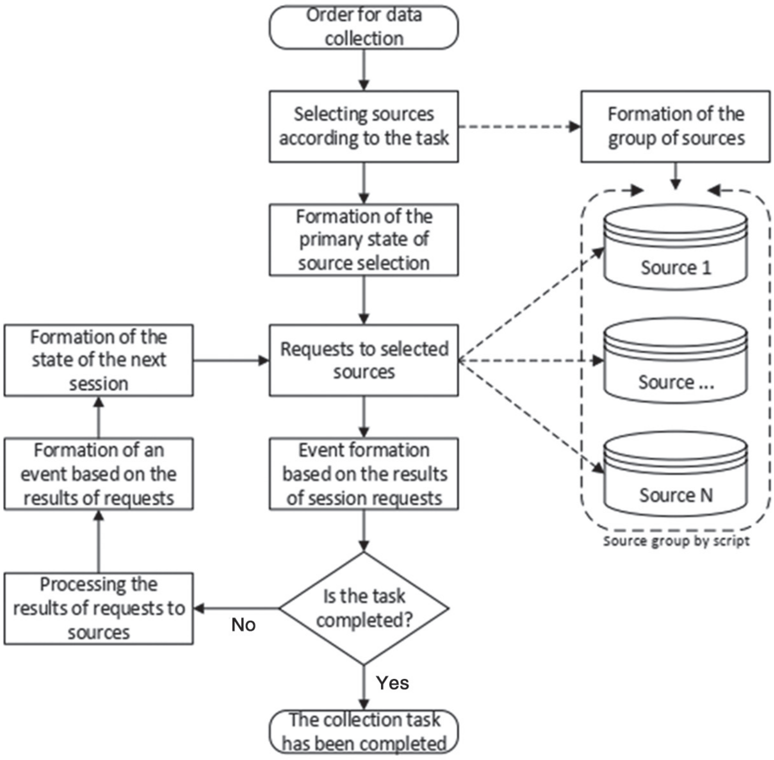
Order for data collection
Selecting sources according to the task
Formation of the group of sources
Formation of the primary state of source selection
Source 1
Formation of the state of the next session
Requests to selected sources
Source ...
Formation of an event based on the results of requests
Event formation based on the results of session requests
Source N
Source group by script
Processing the results of requests to sources
No
Is the task completed?
Yes
The collection task has been completed

Fig. 1. General structural scheme of solving the problem of data collection

data extraction but require more technical support and more powerful access channels to data sources. Among them, there are two main approaches to assessing the amount of available relevant information in the sources and redefining the volume of requests to each of the sources to ensure an increase in overall efficiency [9].

At the same time, the efficiency of the *i*-th source for the *S*-th session is defined as the ratio:

$$E_i^S = R_i^S / K_i^S \quad \text{for } i = 1,\ldots,N,$$

where *N* is the total number of sources used; *S* — current session number; R_i^S — the number of received data relevant to the request from the *i*-th source for the *S* session; K_i^S — the total amount of data received from the *i*-th source for *S* session.

The synchronous approach of parallel extraction of data from a group of information sources is characterized by the fact that the determination of efficiency estimates for each of the sources is carried out after receiving results from all sources, regardless of the moments of completion of collection by each of them. Then the source from which the information was received last determines the total time of each session.

The asynchronous approach of parallel extraction of data from a group of information sources is characterized by the fact that the determination of efficiency ratings for each of the sources is carried out after receiving the results from the first of the sources at the end of data selection. At the same time, it is not taken into account that the selection of data from other sources has not yet been completed. Thus, in this case, for other sources, performance evaluations for the current session are carried out on unfinished sessions. This approach, on the one hand, can significantly speed up the process of obtaining relevant data, and on the other hand, it can significantly slow down the adaptation process of determining the most relevant sources in terms of the number of relevant data.

Regardless of the type of methods described above, in the simplest case it is possible to determine that $D_i^S = (R_i^S - R_i^{S-1})/R_i^{S-1}$ — change in the amount of relevant data across *S* and $S - 1$ sessions.

An estimate of the amount of relevant data per session $S + 1$ that is expected can be defined as

$$\check{R}_i^{S+1} = R_i^S + D_s.$$

Then the value estimate for the next step can be determined as follows

$$K_i^{S+1} = F(\check{R}_i^{S+1}),$$

where *F* — the evaluation function, which is the implementation of the event of a change in the value of the amount of relevant data from session to session for each of the sources.

In most cases, it is possible to evaluate the data extraction performance for the current and previous sessions for all sources

$$E_i^{S-1} = \frac{R_i^{S-1}}{K_i^{S-1}} \quad \text{and} \quad E_i^S = \frac{R_i^S}{K_i^S},$$

where $i = 1,\ldots,N$.

$$K_i^0 = const = \frac{Q}{N}, \;\; R_i^0 = 0.$$

Based on the received evaluations of the efficiency of data extraction for the current and previous sessions for all sources, it is possible to evaluate the change in efficiency as a factor that can decisively influence the determination of the decrease or increase of K_i^{S+1} values — the total amount of data received from the i-th source for session S.

$$D_i^S = E_i^S - E_i^{S-1}.$$

The change in efficiency for each session should be affected by a certain coefficient, the value of which should decrease for more previous sessions, which can be defined as the effect of the time loss of the influence of previous sessions.

Thus, the total effect of influence on K_i^{S+1} can be defined as

$$Z_i^{S+1} = \sum_{j=1}^{S} D_i^j \upsilon^{j-S+1},$$

where υ is the influence factor of the results of previous sessions.

But in the general case, not the entire previous session history can be analyzed to determine the impact, but only a few previous ones

$$Z_i^{S+1} = \sum_{j=h}^{S} D_i^j \upsilon^{j-S+1},$$

where the value of h determines the number of previous sessions that are taken into account when taking into account the influence of previous sessions.

Then the total amount of data received from the i-th source for $S + 1$ session in a simpler fit can be defined as

$$K_i^{S+1} = K_i^S(1 + Z_i^{S+1}).$$

After that, the results are normalized according to the received sum of K_i^{S+1} grades for the total size of the session.

Although such a solution is quite simple, it reflects the general essence of the approach.

Important for this approach to determining the number of requests to sources based on the estimation of the number of relevant results is the study of the optimality of this process depending on the values: h — the number of previous sessions taken into account when taking into account the influence of previous sessions; υ — coefficient of influence of the results of previous sessions.

Analysis of results

For the analysis and comparison of the number of previous sessions and the coefficient of influence of the results of the previous sessions in the algorithm described above, testing was carried out on the generated test repository. A test database was built from 10 data sources of 10,000 records with a percentage of relevant records from 1% to 10%.

The analysis of the results shows that as the value of h increases, the process of adapting the software system to the characteristics of the sources increases significantly and attains its maximum effect at a value equal to 5. As additional studies have shown, at values greater than 5, the further increase in efficiency, that is, the increase in relevant results, practically was absent Similar results were obtained at different values of the coefficient of influence of the results of previous sessions υ.

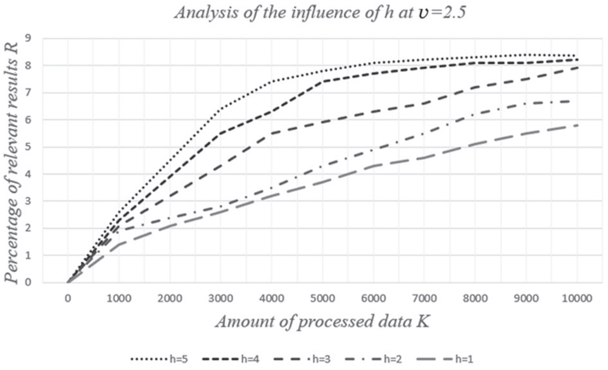

Fig. 2. Analysis of the influence of the number of previous sessions *h*

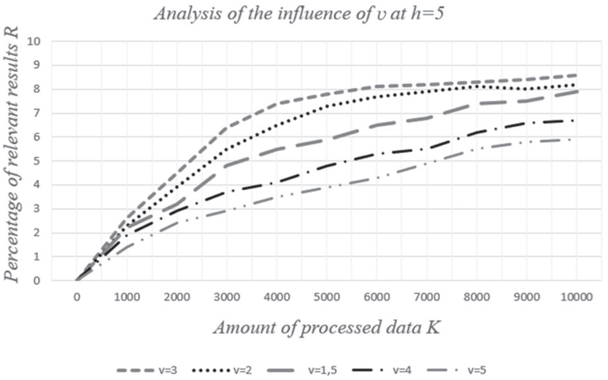

Fig. 3. Analysis of dependence on the influence coefficient *υ*

The conducted testing and analysis of the dependence on the influence coefficient υ makes it possible to determine that the most qualitative results corresponding to the task can be obtained at values of the influence coefficient of the results of previous sessions from 2 to 3. The analysis carried out at smaller values of υ showed similar, but less expressive results.

Conclusions

The article examines the general structural scheme for solving the problem of data collection and describes the conducted testing of the dependence of the adaptation efficiency of obtaining relevant results on the number of previous sessions that are taken into account when taking into account their influence, and the influence coefficient of the results of previous sessions, which determines the degree of their consideration when calculating possibility estimates obtaining relevant data in the next data collection session. Thus, a general increase in obtaining relevant data in subsequent data collection sessions is predicted based on the analysis of previously obtained information about the availability of such data in relevant sources.

This approach in the implementation of the data collection algorithm makes it possible to significantly reduce the number of requests to data sources and, thus, reduce the time and cost of obtaining the necessary data. Such an approach can significantly increase the efficiency of data collection for various information and analytical systems.

References

1. **Richardson C.** *Microservices. Development and refactoring patterns. S.-P.: Peter, 2019. 544 p.*

2. **Ford N., Parsons R., Kua P.** *Building Evolutionary Architectures: Support Constant Change. O'Reilly Media, 2017. 332 p.*

3. **Belnar A.** *Building Event-Driven Microservices: Leveraging Organizational Data at Scale. USA, O'Reilly Media, 2020, 324 p.*

4. **Improving** *the Efficiency of Typical Scenarios of Analytical Activities / O. V. Koval, V. O. Kuzminykh, I. I. Husyeva [et al.] // CEUR Workshop Proceedings, ISSN: 1613-0073. 2021. Vol. 3241. P. 123–132.*

5. **Wolff E.** *Microservices, Flexible Software Architecture. Boston: Addison-Wesley, 2016. 436 p.*

6. **Ashley D.** *Bootstrapping Microservices with Docker, Kubernetes, and Terraform: A project-based guide. Shelter: Manning Publications Co., 2021. 442 p.*

7. **Pethuru Raj, Jeeva S. Chelladhurai, Vinod Singh.** *Learning Docker. Packt Publishing, 2015. 240 p.*

8. **Hugo Filipe Oliveira Rocha.** *Practical Event-Driven Microservices Architecture: Building Sustainable and Highly Scalable Event-Driven Microservices. Ermesinde: Apress, 2021. 472 p.*

9. **Adaptive** *Software System for International Activity Level Assessment / O. V. Koval, V. O. Kuzminykh, I. I. Husyeva [et al.] // CEUR Workshop Proceedings. 2022. Vol. 3503. P. 52–61.*

В. О. Кузьміних, Б. Сюй

ВПЛИВ ПОТОЧНИХ РЕЗУЛЬТАТІВ У ПОДІЙНО-ОРІЄНТОВАНІЙ СИСТЕМІ ЗБОРУ ДАНИХ

У статті розглядаються варіанти реалізації алгоритму управління мікросервісами в системі збору та обробки великих потоків даних в реальному часі на основі адаптивного підходу при реалізації архітектури програмної системи. Адаптивність реалізації програмної системи досягається шляхом використання подійно-орієнтованої мікросервісною архітектури. Реалізація подійно-орієнтованої мікро-сервісною архітектури можлива з використанням, як синхронних, так і асинхронних процедур, що впливає на ефективність збору та обробки даних у цілому. Такий підхід важливий при обробці великих обсягів даних, які отримуються з різнорідних за повнотою, актуальністю та періодом збереження джерел інформації. При цьому, як правило, ставиться завдання мінімізувати загальний час обробки потоків даних. Запропонований підхід дає змогу керувати вибором складу та кількості звернень мікросервісів до джерел за подіями, що формуються під час збору інформації. Формування подій побудовано на основі аналізу результатів отримання даних з відповідних інформаційних джерел, що використовуються для збору даних. Це, у свою чергу, ставить завдання мінімізації кількості запитів до джерел інформації для отримання достатньої кількості одиниць даних, релевантних пошуковому запиту. Подійно-орієнтована мікросервісна архітектура системи дозволяє адаптувати роботу системи до навантажень на окремі мікросервіси на основі аналізу подій під час збору і первинної обробки отриманих даних. Залежно від конкретного завдання можливе використання як синхронного, так і асинхронного алгоритму управління мікросервісами. У статті проведено аналіз ефективності отримання релевантних даних залежно, як від ступеня врахування попередніх результатів при формуванні подій, так і величин врахування їх впливів. Використання подійно-орієнтованої мікросервісної архітектури може бути особливо ефективним при розробці різноманітних інформаційно-аналітичних систем, які аналізують великі об'єми даних з різноманітних інформаційних джерел у реальному часі.

Ключові слова: мікросервіси; адаптація; подійно-орієнтована архітектура; великі дані.