Adaptive replica consistency policy for Kafka

Zonghuai Guo [1,2,*], *Shiwang Ding*[1,2]

[1]Chongqing University of Posts and Telecommunications, 400065, Nan'an District, Chongqing, P.R.China
[2]Chongqing Mobile Internet Application Engineering Technology Research Center, 400065, Nan'an District, Chongqing, P.R.China

Abstract. With the rapid development of the Internet, such as storm, s4, sparkstreaming and other large data real-time computing framework, is widely used in real-time monitoring, real-time recommendation, real-time transaction analysis and other systems for real-time consumption of data streams, Kafka messaging system has been widely deployed. Aiming at the problem that the Kafka cluster needs a lot of network overhead, disk overhead and memory consumption to ensure the reliability of the message, the clustering load is increased, and a replica adaptive synchronization strategy based on the message heat and replica update frequency is proposed. It is proved that the Kafka cluster can guarantee the reliability of the message, and it can significantly reduce the overhead of the resource and improve the throughput of the cluster by using the method of dynamically adjusting the replica synchronization to reduce the system resource consumption while ensuring the reliability of the message. to ensure the system availability and high performance.

1 Introduction

In recent years, with the rapid development of the Internet and mobile Internet, application services based on big data,Such as real-time stock trading analysis, real-time recommendations and other electricity providers to promote the deployment of applications storm, s4, sparkstreaming real-time computing framework.Most of these real-time computing frameworks are based on Kafka messaging system construction. However,the Kafka messaging system may cause serious problems such as prolonged messages, lost messages, out of sync of the primary and secondary data， because of the data update, network delay, server downtime and other reasons,those affect the reliability of the message system.At present, the Kafka message cluster ensures the reliability of data by using the copy mechanism. For this reason, the Kafka cluster brings more additional network overhead, disk overhead, and memory consumption, which increases the cluster load and affects the overall performance of the message system.

The main contribution of this paper is to adopt a dynamically adjusting the consistency policy of replica synchronization to reduce the consumption of network bandwidth and other resources during replica synchronization. A replica adaptive consistency synchronization strategy based on topic heat and replica update frequency is proposed.

2 Related work

At present, more and more open source distributed processing systems support the integration with Kafka. In order to ensure the reliability of Kafka cluster message, Wang Yan and others Proposed a design of Consumer based on Kafka's reliability [2]. It is based on Kafka's lowLevel API to ensure that consumers ensure that each message can be consumed at least once during consumption. Achieve data reliability on the consumer side. However, this does not guarantee that the message can not be lost on the server. Therefore, in order to ensure the reliability of the message on the server, Yang Donghui and others propose a method for guaranteeing message reliability of distributed message queuing [1] Reliable transmission of messages. Due to the additional network and resource overhead associated with replica synchronization to ensure data reliability, balancing the additional overhead and system performance required to maintain data reliability is of paramount importance. Wang Ximei and others proposed an adaptive copy consistency maintenance mechanism based on application and user requirements [5], The system adjusts the consistency policy in real time according to the formalized requirements of consistency, so as to achieve the dynamic balance between copy consistency, availability and system performance.

[*] Corresponding author: 992828405@qq.com

The above algorithms are not flexible enough to guarantee the data reliability in the Kafka cluster and the overhead of the cluster is large. Since the Kafka cluster dynamically maintains the ISR set, this paper combines the reliability of Kafka message queue with the adaptive synchronization strategy in the cloud storage system, In this paper, an adaptive strategy for synchronizing Kafka cluster replica data is proposed to dynamically adjust the ISR set to ensure the reliability of data and to reduce the extra overhead of network overhead, memory and CPU in the cluster.

3 Kafka message queue reliability

In this section, we mainly introduce the native policies, cluster structures, and reliability definitions of clusters that Kafka Cluster maintains for data reliability.

3.1Kafka cluster system structure

The Kafka message system, including message producer, message consumers, broker and managers (zookeeper) four parts,As shown in Figure 1:

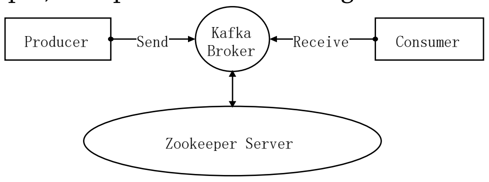

Fig.1. producer consumer and zookeeper structure relationship

3.2 Kafka reliability

Kafka message queue In cluster mode, by copying the data in each partition to multiple Kafka server to achieve a copy mechanism, which stores the data source Server node called Leader, and used to store the data replication node called Follower, Follower by backing up the Leader data to ensure the reliability of the message, which requires Follower node and leader node to keep pace. The Partition copy is highly consistent with the Leader by copying the Leader data. Follower pulls messages from the leader and saves them in their own log messages just like an ordinary consumer. Therefore, there will be the following problems:

1)Take up a lot of network bandwidth (for example, the cluster was originally large and a large amount of data, and later add a Broker node need to migrate data), and even plug the network.

2)Because there can be many copies of Kafka, creating a lot of unnecessary overhead if the copy of each partition is updated each time;

Therefore, in order to reduce the additional system overhead caused by replica data synchronization and ensure the consistency of messages, this paper proposes a replica synchronization adaptive consistency strategy based on message heat.

4 Kafka replica synchronization adaptive consistency strategy

In this section, we mainly introduce the Partition thermal model, the message heat prediction model, and Kafka cluster adaptive consistency strategy.

4.1 Partition thermal model

During a period of time, if the Partition's access frequency is higher, the access popularity of this Partition is greater; therefore, this article uses the access frequency to represent the message access heat in the ith period of time,As shown in Formula (1):

$$S_i = E_i / T_i \tag{1}$$

Where Ei represents the weighted average number of accesses of the Partition in the i th time period (Ti), as shown in Formula (2); Partition may also present different access situations even during an access time period, for example, The beginning of a large amount of traffic, and then gradually reduced. It may also be that the traffic is small initially and then gradually increased.

Therefore, if it is apparently inaccurate to judge the heat of a message based solely on the total number of time periods, we use the LRFU (Least Recently Frequently Used) algorithm to calculate the heat of the Partition during the time interval. The LRFU algorithm combines LRU (Least Recently Used), and least recently used (Least Frequently Used) algorithm [6], taking into account the access time and the access frequency. The principle is that the weight of the last access is large and the weight of the previous access is smaller.

Since the function of Partition's access times is not continuous within the time Ti, we use Formula (2) to get the weighted average number of access within the time Ti.

$$E_i = \sum_{1}^{n} X_j P_j \qquad (j = 1..n) \tag{2}$$

Where Xj is the offset of the jth time point in Ti; Pj is the weight function of the jth time point in Ti:

$$P_j = \left(\frac{1}{\psi + 1}\right)^{\omega\,(T_i - \sigma)} \qquad (\psi > 0,\ 0 < \omega < 1) \tag{3}$$

One of the time periods T contains n time points, $\sigma \in \{t1, t2,\ldots tj,\ldots tn\}$,j=1..n.When $\omega = 0$, it is the LFU algorithm; when $\omega = 1$, it is the LRU algorithm; when $0 < \omega < 1$, it combines both the LFU and LRU algorithms [6].

4.2 Heat prediction model

In the Kafka cluster, the heat of the message changes with the data stream processing needs. Therefore, the current heat of the partition is calculated the every time Ti , and the corresponding replica consistency policy is selected according to the predicted message popularity value Ensure the consistency of the copy. This article further combined with the history of the partition heat set different weights, re-calculate the partition estimated heat value as Formula (4) [6]:

$$H_{(i)} = \varepsilon S_i + (1 - \varepsilon\ )H_{(i-1)} \quad (4)$$

H(i) represents the historical heat of the message in the i-th time interval, H(i–1) represents the historical heat of the message in the i-1 time interval(Ti–1), Si represents the current heat of the message in Ti, and ε represents the weight of the current heat of the partition , The greater the value of ε, the greater the impact of heat value, and vice versa. The heat of the algorithm is based on the historical heat of the partition, so that the partition heat can maintain a certain degree of stability, reducing the impact of the current heat volatility on the partition heat calculation; on the other hand, by reducing the history for the greater the current heat weight, Value of the heat of the interference of the message, which more accurately reflect the current actual access to the message.

4.3 Copy of the partion update frequency

In the Kafka cluster, the frequency of copy updates varies for several reasons, resulting in lag in some copy updates, affecting data consistency. Therefore, the copy update frequency will also affect the Kafka cluster data consistency; copy update frequency as shown in Formula (5):

$$Fi = W/Ti \quad (5)$$

W represents partition increase message offset within the time Ti, Fi represents copy update frequency within the time Ti.

4.4 Copy adaptive consistency Strategy

In order to improve the system performance as much as possible while guaranteeing the consistency of the message, this paper uses the replica adaptive consistency strategy， forecast the future access behavior of the message according to the access of partition within the time interval Ti ,and according to news estimated heat H(i) to measure it Combined with a copy of the partition in a period of time to update the frequency Fi, select a more appropriate update strategy. in case:

$$H(i)/Fi > 1$$

It indicates that the consumption rate of the topic is greater than the update rate of the copy. In this case, put a copy of the partition that meets the condition in the ISR, and immediately Synchronous the copy. in case:

$$H(i)/Fi < 1$$

topic consumption rate is less than the copy of the update rate, At this point, the partition copy of the topic under this condition is removed from the ISR, and the replica data is not synchronized.When the network overhead in a cluster is not large, the partition is put into the ISR collection for data synchronization.

5 Experimental results and analysis

This paper experiment by building a Kafka cluster, the cluster operating environment is as follows:Intel(R) Xeon(R) CPU E5620 @ 2.40GHz RAM:32G,hard disk SATA-500G/7200 18T,The number of 3.

5.1 Analyze the total update amount of news in the cluster

Experiment set up three consumers with different consumption rates, set the time period of 1s, compared to the same amount of time to update the cluster data; experiment also by setting Ten different size time period. The experimental results are shown in Figure 2 and Figure 3. As shown in Figure 2, the total amount of updates of cluster message in this paper is less than that of Kafka's original copy update policy in any period of time.From Figure 3, we can see that with the increase of the time period, the gap between the native synchronization strategy and the adaptive synchronization strategy is increasing. Both of them basically increase linearly. This is due to the smaller number of partitions in the ISR collection. The number of partitions that need to be updated is reduced, resulting in a reduction in the total number of message updates for the entire cluster. But for themselves, read and write rates are relatively stable, causing them to essentially all grow linearly.

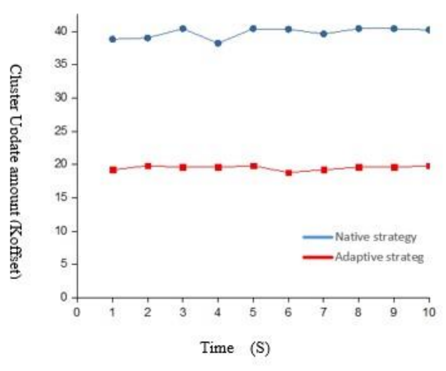

Fig.2. The same time period message update comparison chart

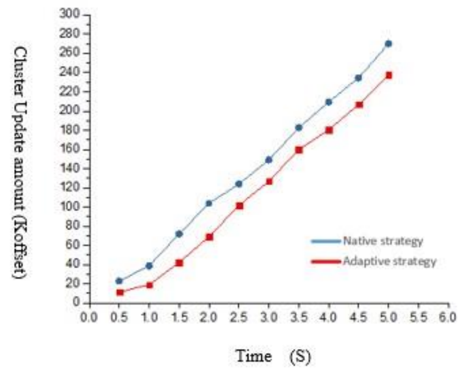

Fig.3. Different time period message update comparison chart

5.2 Cluster throughput rate analysis

In order to compare the efficiency of clustering under two strategies, we can verify the different clustering throughput rates and set different consumption speeds to influence the different heats of messages. Here, we set the time period as 1s and set the algorithm according to the cluster Fewer copies than native clusters. The experimental results are shown in Figure 4. As can be seen from Figure 4, the throughput rate of Kafka clusters implementing adaptive policies is significantly higher than that of native policy clusters because the total number of replicas required for the entire cluster decreases, so the network overhead and resource overhead used to synchronize replicas in the cluster Reduce, so the efficiency of the entire cluster will increase, the throughput rate will increase.

The paper further contrasts the update of cluster messages under different consumption rates, as shown in Figure 5, when the rate difference between consumers increases, the throughput rate will decrease because the speed of a consumer updates than the copy When the speed is still high, the copy of the partion needs to be synchronized immediately, resulting in an increase in the number of copies to be updated, thus adding extra overhead and reducing the throughput rate.

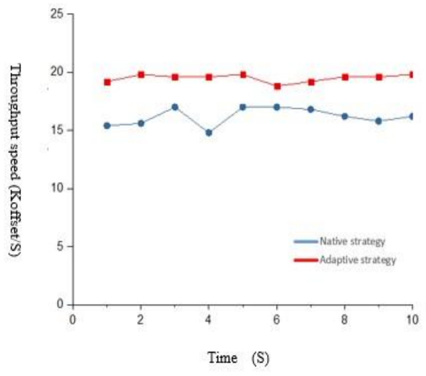

Fig.4. Cluster throughput rate comparison chart

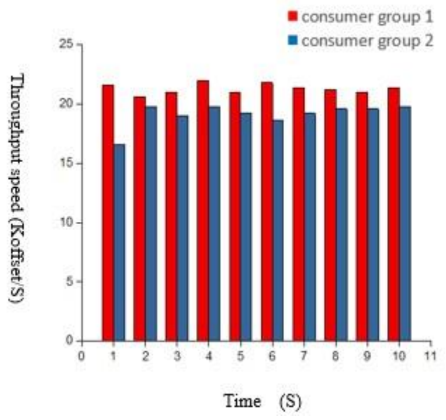

Fig.5. Different consumer groups

6 Conclusion

Kafka ensures the reliability of message queues and its consistency through a replication mechanism .This paper analyzes the reliability mechanism of Kafka message queue and its replica data synchronization strategy.Aiming at the difference of applications and users' data consistency requirements in Kafka cluster, this paper proposes an adaptive replica consistency strategy based on application characteristics, which allows the application to choose replicas to update the consistency policy according to the statistical characteristics at run time to maximize the consistency of messages Sexual and usability needs.Experiments and analyzes show that the proposed adaptive replica consistency strategy can improve the availability of replicas and reduce the system overhead caused by the update synchronization while meeting the reliability requirements of applications and users.As a result, it is more suitable than the existing Kafka replica data synchronization strategy for today's diverse application environments in big data clusters.The future work will focus on improving the accuracy of the news forecasting model to reduce the unstable impact of copy synchronization due to the large difference between the partition's consumption heat and the partition's update speed, in order to obtain the copy consistency, A better balance between usability and system performance.

References

1. Yang Donghui.Computer Knowledge and Techn-ology.11(21), pp.75-79,(2015)
2. Wang Yan,Wang Chun..COMPUTER ENGINE-ERING & SOFTWARE. 37(1),pp. 61-66.(2016)
3. Kraska T, Hentschel M, Alonso G,et al. Proceedings of the VLDB Endowment,2(1),pp.253-264,(2009)
4. Yu H F,Amin V.ACM Transactions on Computer Systems,20(3), pp. 239 -282(2002)
5. WANG Xi-Mei. Journal of Graduate University of Chinese Academy of Sciences. Vol.30 No.1 January pp.91-97(2013)
6. REN Tao.Computer Engineering and Applications. 51(4),pp.83-86(2015)
7. Wang Ximei，Yang Shoubao，Wang Shuling，et al.Proc of the 9th International Conference on Gridand Cloud Computing,16(1),pp.13-17(2016)
8. CUI Wei.Computer Engineering and Design. Vol. 28 No. 10,pp.2259-2261(2007)
9. S. Maddineni, J. Kim, et al,Journal of grid comput-ing, vol. 10, no. 4, pp. 647-664, (2012)