Journal of Science and Technology
ISSN: 2456-5660 Volume 8, Issue 04 (April-2023)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2023.v8.i04.pp118 - 123*

Design of Multi-priority Message Sending Method Based on Bandwidth State Control

[1*]Zhongjun Dong | [2]Peng Zhang | [3]Ning Hu
[1]PG SCHOLAR, [2] ASST. PROFESSOR,[3] ASSOSCIATE PROFESSOR & HOD
Aeronautical Industry 631 Institute, Xi 'an City, Shaanxi Province, China

To Cite this Article
Zhongjun Dong , Peng Zhang ,Ning Hu**, Design of Multi-priority Message Sending Method Based on Bandwidth State Control "** *Journal of Science and Technology, Vol. 08, Issue 07,-April 2023, pp118-123*

Article Info
Received: 02-03-2023 Revised: 08-04-2023 Accepted: 22-04-2023 Published: 29-04-2023

ABSTRACT: During the use and operation of the integrated centralized system, it relies on and generates a large amount of information, which is distributed on a large number of clients waiting to be processed. Due to the limited storage resources, the priority of the server to process these messages is different, resulting in the difficulty of allocating resources to the client messages with low priority. Therefore, different services need to be customized for different clients or data streams. In this paper, a message processing method using token bucket algorithm for bandwidth state control is proposed, which not only has less development complexity, but also allocates bandwidth resources more reasonably, and effectively reduces the performance jitter of the native system.

KEYWORDS— Bandwidth control, Token bucket, Message processing, Priority queue

1. INTRODUCTION

In recent years, with the further development of information system, more and more data need to be managed in network file system. In the traditional local storage mode and the NAS storage that features network file systems, priority queues are used to preferentially process high priority read and write tasks. In this way, limited computing resources are allocated to important tasks and the stability of the client system is improved. However, in this priority queue mode, it may be difficult for low-priority clients to allocate computing resources from

the server, causing the file system to wait forever, hard to serve, and fail to implement message transfer.

2. COVERAGE

In order to enable the file system to deal with low-priority tasks in time, we design a multi-priority message sending method based on bandwidth state control. By allocating bandwidth to tasks of different priorities, low-priority tasks can be processed slowly but effectively while high-priority tasks are processed quickly. This method improves the certainty of system behavior, enhances the control of system bandwidth, and ensures that client tasks can be handled efficiently.

The design idea of this method is as follows: Based on priority control, bandwidth control and machine control strategies are added for different clients. By adding a periodic token generator to the network file system server, tokens are issued for tasks of different bandwidth according to the processing efficiency of the file system itself. Only the tasks with tokens can be executed, while those without tokens wait for tokens in the buffer queue. So that tasks of different priority clients have a chance to process.

3. BANDWIDTH ALLOCATION

For example, if there are two clients in the system, you can allocate 75% of the total bandwidth to the client with a high priority and 25% to the client with a low priority in the ratio of 3:1 based on user requirements. You can also allocate 4 based on user requirements: 1 ratio, 80% of the total bandwidth is allocated to high-priority clients and 20% of the bandwidth is allocated to low-priority clients. For example, if there are more than two clients in the system, the system prioritizes the priorities of all clients and allocates the total bandwidth to each client based on user requirements to ensure that all clients could process their tasks.

4. GENERATE TOKEN

The server obtains the number and priority of clients in the system and allocates the total bandwidth to clients based on their priorities. The bandwidth allocated to clients with higher priorities is greater than that allocated to clients with lower priorities. Also, the type of dynamic token is defined according to the number of clients. Calculate the dynamic token generation rate corresponding to each client based on the bandwidth allocated by each client.

For example: If the average rate at which the server processes messages (also called packets) is r and the bandwidth of the client is 20% of the total bandwidth, a dynamic token corresponding to the client can be generated every 5/r seconds. The generation rate of dynamic tokens matches the bandwidth allocated by the server to the client.

5. TOKEN BUCKET

All tokens are stored in the token bucket. In this case, bitmap method or other methods should be used to distinguish different types of dynamic tokens. There can also be token buckets that match the number of clients, and the token buckets can be bound to the clients. Meanwhile, the sizes of different token buckets can be set according to the bandwidth ratio. In this case, the dynamic tokens generated by the token generator are placed in the corresponding token buckets. During dynamic token generation, if the token bucket is full, dynamic token generation is stopped or discarded. Dynamic tokens can be distinguished either by bitmap staining or by other methods. During dynamic token generation, the token bucket may be full due to various reasons. In this case, you can stop the generation of dynamic tokens or discard the generated dynamic tokens.

6. TOKEN USE AND ELIMINATION

Due to the client message is generated continuously, and the server according to a certain period of time to forward the message generated by the client. Therefore, messages generated by this client can be first stored in the request queue for forwarding. The server periodically compares the number of dynamic tokens to the number of messages while processing the task, and based on the comparison results decides whether to send or cache the message. The period in which the number of messages is compared to the number of dynamic tokens can be set by the period in which the message is sent by the client, or in other ways. If the number of dynamic tokens is greater than or equal to the number of messages, the message in the request queue is forwarded and the used dynamic tokens in the token bucket are deleted. If the number of dynamic tokens is less than the number of messages, the message is stored in the cache queue for that client. The specific process is shown in Figure 1.

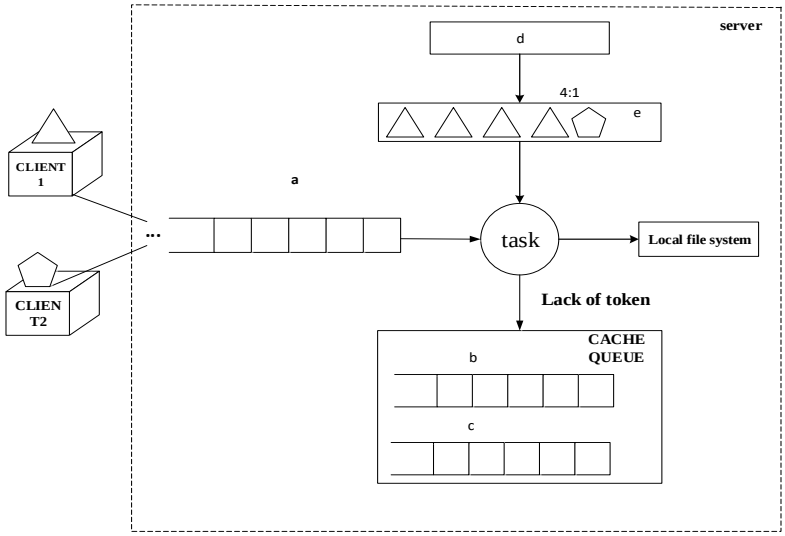

Figure 1. Dong

Figure guide:

(a): priority request control queues

(b): client 1 cache queue

(c): client 2 cache queue

(d): Periodic token generator

(e): token bucket

7. INFORMATION PROCESSING FOR CACHE QUEUE

When messages in a client are stored in a cache queue, periodically compare the number of messages in the cache queue with the number of dynamic tokens corresponding to the client. If the number of dynamic tokens is greater than or equal to the number of messages, the messages in the cache queue are forwarded and the used dynamic tokens in the token bucket are deleted. If the number of dynamic tokens is less than the number of messages, wait until the next time the number of dynamic tokens is compared to the number of messages. If the cache queue is full and the number of dynamic tokens is less than the number of messages, the cache queue messages are discarded and a message sending error signal is reported.

8. CONCLUSION

Compared with the prior art, the beneficial effects of this method are: Based on controller priority control, bandwidth control and machine control strategies are added for different clients, so that the tasks of clients with different priorities have the opportunity to be processed. In this way, the network file system is avoided due to the closed operating environment and application scenarios and limited computing resources (bandwidth resources). The tasks of low-priority clients cannot be handled in a timely manner. As a result, low-priority clients cannot respond to the problem. The preceding method can improve the certainty of system behavior and ensure that client tasks can be handled in a timely and effective manner.

REFERENCES

[1] Keshari, Surendra Kumar, Vineet Kansal, and Sumit Kumar. "A systematic review of quality of services (QoS) in software defined networking (SDN)." *Wireless Personal Communications* 116 (2021): 2593-2614.

[2] Jing, Weipeng, et al. "QoS-DPSO: QoS-aware task scheduling for cloud computing system." *Journal of Network and Systems Management* 29 (2021): 1-29.

[3] Ichwan, Muhammad Iqbal, Lipur Sugiyanta, and Prasetyo Wibowo Yunanto. "Analisis Manajemen Bandwidth Hierarchical Token Bucket (HTB) dengan Mikrotik pada Jaringan SMK Negeri 22." *PINTER: Jurnal Pendidikan Teknik Informatika dan Komputer* 3.2 (2019): 122-126..

[4] Putra, Ketut Gede Widia Pratama. *Penerapan Manajemen Bandwidth Menggunakan Metode Hierarchical Token Bucket Pada Layanan Hotspot Mikrotik Undiksha*. Diss. Universitas Pendidikan Ganesha, 2020.

[5] Armanto, Armanto, and Nelly Khairani Daulay. "Analisis Quality of Service (Qos) Pada Jaringan Internet Di Universitas Bina Insan Lubuklinggau Menggunakan Metode Hierarchical Token Bucket (Htb)." *Jurnal Digital Teknologi Informasi* 3.1 (2020): 8-13.

[6] Daulay, Oktavia Larassari. "Analisis Quality of Services (Qos) Pada Manajemen Bandwidth Menggunakan Metode Hirarchical Token Bucket (Htb) Pada Sistem Jaringan." *JISTech (Journal of Islamic Science and Technology)* 5.2 (2021).

[7] Merceedi, Karwan Jameel, and Nareen Abdulla Sabry. "A Comprehensive Survey for Hadoop Distributed File System." *Asian Journal of Research in Computer Science* (2021).

[8] Li, Xijun, Yunfan Zhou, and Ji Zhang. "PASCAL: A Learning-aided Cooperative Bandwidth Control Policy for Hierarchical Storage Systems." *arXiv preprint arXiv:2303.08066* (2023).

[9] Zhang, Xinchang, and Tianyi Wang. "Elastic and reliable bandwidth reservation based on distributed traffic monitoring and control." *IEEE Transactions on Parallel and Distributed Systems* 33.12 (2022): 4563-4580.

[10] Li, Lang, Xinming Tan, and Chao Deng. "An Improved Token Bucket Algorithm for Service Gateway Traffic Limiting." *Proceedings of the International Conference on Advances in Computer Technology, Information Science and Communications-CTISC*. 2019.

[11] Hakim, Lukmanul, et al. "Manajemen Bandwidth Menggunakan Metode Hierarchical Token Bucket Pada SMK Muhammadiyah Karangampel." (2021).

[12] Nawi, Nor Afifah Mat, and Robiah Yusof. "Simulation of Token Bucket Algorithm for Network Traffic Performance." *Journal of Advanced Computing Technology and Application (JACTA)* 2.1 (2020): 21-28.

[13] Fadilah, Ahmad Zen, Rd Rohmat Saedudin, and Umar Yunan Kurnia Septo Hediyanto. "Analisis Simulasi Manajemen Bandwidth Menggunakan Metode Hierarchical Token Bucket (htb) Untuk Meningkatkat Quality Of Service (qos)." *eProceedings of Engineering* 8.5 (2021).

[14] Pamungkas, Fatwahadi Ilham. "Perbandingan Manajemen Bandwidth Menggunakan Metode PCQ (Per Connection Queue) dan HTB (Hirarchical Token Bucket)." *Buletin Sistem Informasi dan Teknologi Islam ISSN* 2721 (2021): 0901.