Research on Load Balancing Technology for Product Data Management Server Cluster

Jing Li[✉], Peizhang Wang, Yi Sun, Qian Li, Lu Jia, Yongle He, Lijuan Duan,
and Yin Luo

Network Information Department, China North Vehicle Research Institute, Huaishuling 4#,
Fengtai, Beijing, China
30952197@qq.com

Abstract. The problem of load imbalance in Product Data Management (PDM) server clusters not only reduces the efficiency of PDM system applications, but also causes downtime in severe cases, affecting the development process of scientific research models. This article applies three server cluster load balancing technologies (IPVS, NGINX, DNS polling) to the comprehensive testing and analysis of the PDM server cluster in the research institute, solves critical network connection issues, achieving server load balancing.

Keywords: IPVS · NGINX · DNS round-robin · SSH

1 Introduction

The Product Data Management (PDM) server cluster in the research institute consists of three Linux servers, and the PDM database uses Oracle's RAC technology, which ensures normal access to PDM application data as long as the database on any one server is running normally. The PDM service Windchill on the three servers also uses PTC's clustering technology, so that any server can independently provide services to external users. A gateway isolates the servers from the clients to ensure PDM data security. Users can access any PDM server to collaborate on design work.

2 Overview of IPVS (IP Virtual Server) Technology

The working principle of IPVS (IP Virtual Server) is to realize a single image of the system at the IP layer. The load balancing scheduler is used as the only entrance to the server cluster, and clients access the server by accessing the virtual IP address (VIP: Virtual IP Address). IPVS currently has three main modes: VS/NAT, VS/TUN and VS/DR [1].

2.1 VS/NAT Mode (Virtual Server via Network Address Translation)

VS/NAT is the simplest meth, where all Real servers (RS) only need to point their gateway to the Load balancer (LB). Both request messages and response messages need to be address rewritten by LB, that is a large overhead. When the number of RS reaches a certain number, LB will become a bottleneck of the entire system.

F. Huang et al. (Eds.): ICAIE 2023, AHCS 15, pp. 310–317, 2024.
https://doi.org/10.2991/978-94-6463-242-2_38

2.2 VS/TUN Mode (Virtual Server via IP Tunneling)

The VS/TUN model's LB only processes request packets, while RS's response packets are returned directly to clients without going through the LB. By using VS/TUN, which is more efficient than VS/NAT, the maximum throughput of the cluster system can be increased by nearly 10 times.

2.3 VS/DR Mode (Virtual Server via Direct Routing)

VS/DR model (Virtual Server via Direct Routing) like the VS/TUN, the response to the customer does not go through the LB, greatly improving the scalability of the cluster system. It modifies the MAC address of the data frame to the MAC address of the selected backend RS and then sends it to the backend RS group [2]. This method uses direct routing technology to implement virtual servers without requiring that RS in the cluster must support IP tunneling protocols, reducing the overhead of encapsulating IP. It requires that LB and RS must be in the same subnet.

2.4 Configuration of Load Balancing Server in IPVS Mode

See Table 1.

3 Nginx Load Balancing Technology

Nginx is a high-performance HTTP and reverse proxy web server. Its multi-process event-driven architecture, fully asynchronous network processing mechanism, and minimal inter-process switching design enable it to support millions of TCP connections at the same time. At the same time, its built-in upstream module not only can interact with upstream servers without any blocking but also implements the function of forwarding the response packet of the upstream application layer protocol to the downstream client. These designs ensure that Nginx can serve as an excellent reverse proxy server, balance and coordinate services by load balancing and forward them to upstream servers such as Apache and Tomcat to smoothly complete high-concurrency HITP requests and responses [4]. Load balancing sketch map is shown as Fig. 1.

3.1 Nginx Load Balancing Algorithm

Nginx mainly configures the load balancing algorithm through its upstream module to achieve load balancing from the client IP to the backend RS. Currently, there are five types of scheduling algorithms supported by Nginx: round-robin and weighted round-robin, ip_hash, url_hash, least30nn, and fair [4].

3.2 Nginx Load Balancing Installation Configuration and Start-Stop

See Table 2.

Table 1. Configuration command in IPVS mode

Command annotation	Command
Install ipvsadm load balancing manager on the load balancing server	yum -y install ipvsadm
Set up a virtual IP (VIP) for tunnel mode to provide services	ip addr del 192.168.180.11/24 dev tunl0
Set up a polling [3] algorithm for load balancing servers	ipvsadm -A -t 192.168.180.11:80 -s wrr
Add three web servers to the load balancing server	ipvsadm -a -t 192.168.180.11:80 -r 192.168.180.43:80 -i ipvsadm -a -t 192.168.180.11:80 -r 192.168.180.44:80 -i ipvsadm -a -t 192.168.180.11:80 -r 192.168.180.45:80 -i
Start ipvsadm service	service ipvsadm start
Query load balancing server status	ipvsadm –Lcn
Set the tunnel mode to provide services for the virtual IP (VIP)	ip addr del 192.168.180.11/32 dev tunl0
Implement ARP restrictions on VIP to ensure that when accessing VIP on the network, the first access is to the load balancing server	echo “1”>/proc/sys/net/ipv4/conf/tunl0/arp_ignore echo “2”>/proc/sys/net/ipv4/conf/tunl0/arp_announce echo “1”>/proc/sys/net/ipv4/conf/all/arp_ignore echo “2” > /proc/sys/net/ipv4/conf/all/arp_announce

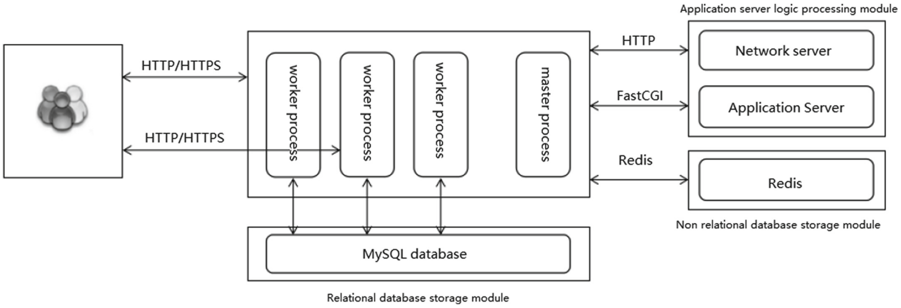


Fig. 1. Load balancing sketch map

Table 2. Installation, modify configuration and start-stop Command

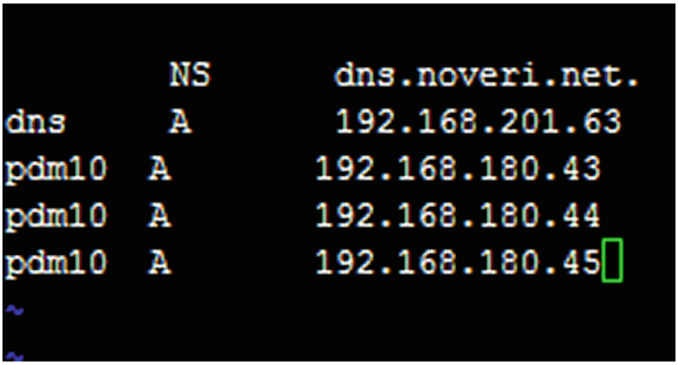
Command annotation	Command
Installation	tar -zxf nginx-1.19.7.tar.gz cd nginx-1.19.7 ./configure make make install

Modify configuration	vi /root/nginx-1.19.7/conf/nginx.conf
	http {
	upstream pdm10.noveri.net{
	http {ip_hash;
	consistent_hash $request_uri;
	server 192.168.180.43:80;
	server 192.168.180.44:80;
	server 192.168.180.45:80;}
	location / { proxy_pass http://pdm10.noveri.net;
	proxy_set_header Host $host;
	proxy_set_header X-Real-IP $remote_addr;
	proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;}}

Start Nginx	/nginx
Stop Nginx	/nginx -s quit

3.3 Nginx + Sticky Mode

Nginx's load balancing automatically switches to another server after each client accesses the server once, ensuring that the client's access to the server is evenly distributed across all servers [5]. But when a client submits data to one server and then goes to get the data processed by the server, Nginx redirects it to another server, the client cannot obtain the results processed by the server. Nginx's IP push algorithm ensures that the same client can access the same server for a period, which can relatively solve this problem [6]. But due to the particularity of the research institute's confidential network, all clients must access the server through the gateway, so all client IPs obtained by Nginx's load balancing server are the same IP (the gateway's IP). Using IP Push algorithm, all clients in the research institute can only access one server, which loses the effect of Nginx's load balancing [7].



```
        NS      dns.noveri.net.
dns     A        192.168.201.63
pdm10  A       192.168.180.43
pdm10  A       192.168.180.44
pdm10  A       192.168.180.45
~
~
```

Fig. 2. DNS server configuration diagram

Sticky is a cookie-based load balancing solution that uses cookies to maintain sessions between the client and backend servers. Under certain conditions it can ensure that the same backend RS is accessed by the same client. You need to download a Sticky installer, configure it on an already installed Nginx server, modify the algorithm mode of Nginx to sticky [8].

4 DNS Server Polling Mode DNS Server Polling Mode Principle

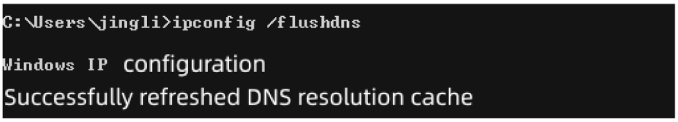
On the DNS server, the same domain name is assigned to several different server IPs. When the client accesses the server of the domain name, it will first obtain the real IP address of the server of the domain name from the DNS server [9]. When DNS receives a request, it will sequentially take out the IP address that is different from the IP address provided last time and matches the domain name and pass it to the client. In this way, clients can access several servers evenly [10].

4.1 DNS Server Polling Configuration

The DNS server polling configuration is as shown in Fig. 2: On the DNS server, a domain name PDM10 is assigned to three IP addresses at the same time. When a new user accesses the DNS server, DNS assigns a new IP to PDM10 each time [11]. The DNS server configuration diagram in research institute is shown as Fig. 2.

4.2 DNS Server Polling Verification

To test it on the client side, open the command line and enter ‘ipconfig /flushdns’. Each time the command is executed, the IP address of the pdm10 server obtained through the DNS server will change once [12], as shown in the Fig. 3.



```
C:\Users\jingli>ipconfig /flushdns

Windows IP configuration
Successfully refreshed DNS resolution cache
```

Fig. 3. ipconfig/flushdns to modify PDM server IP

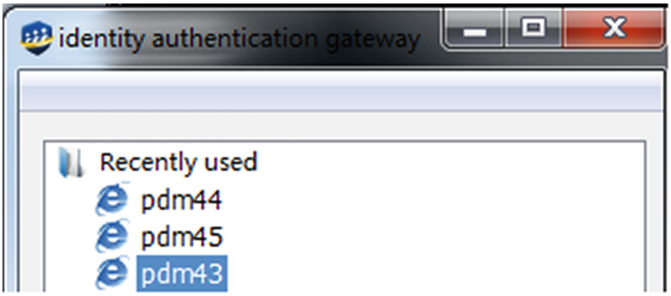


Fig. 4. Gateway configuration

4.3 Configuring the PDM Server in the Gateway

In conjunction with the settings of the DNS server, three PDM servers need to be configured in the gateway to ensure that users can penetrate the gateway and use three PDM servers at the same time. The gateway configuration is as shown in the Fig. 4.

5 Load Balancing Technology Ultimately Chosen by the Research Institute

The topology diagram of the end users accessing the PDM server in the research institute is shown in the Fig. 5: End users need to obtain PDM server information through terminal switches, firewalls, gateways, and confidential service switches. Switches and firewalls will not change the IP address of the user terminal accessing the PDM server, but after authentication through the gateway, all terminal IP addresses can only be recognized as the IP address of the gateway on the PDM server. Both IPVS and Nginx load balancing technologies need to obtain the IP address of the user terminal before they can allocate the corresponding PDM server to users for access [13]. Since only the IP address of the gateway can be provided to IPVS and Nginx servers after passing through the gateway, IPVS and Nginx servers can only allow all user terminals to access one PDM server, which cannot achieve load balancing [14]. However, DNS polling technology distinguishes three PDM servers during gateway authentication, ensuring that all three servers provide services to users evenly.

Therefore, DNS server polling technology was ultimately chosen as the load balancing technology by the research institute.

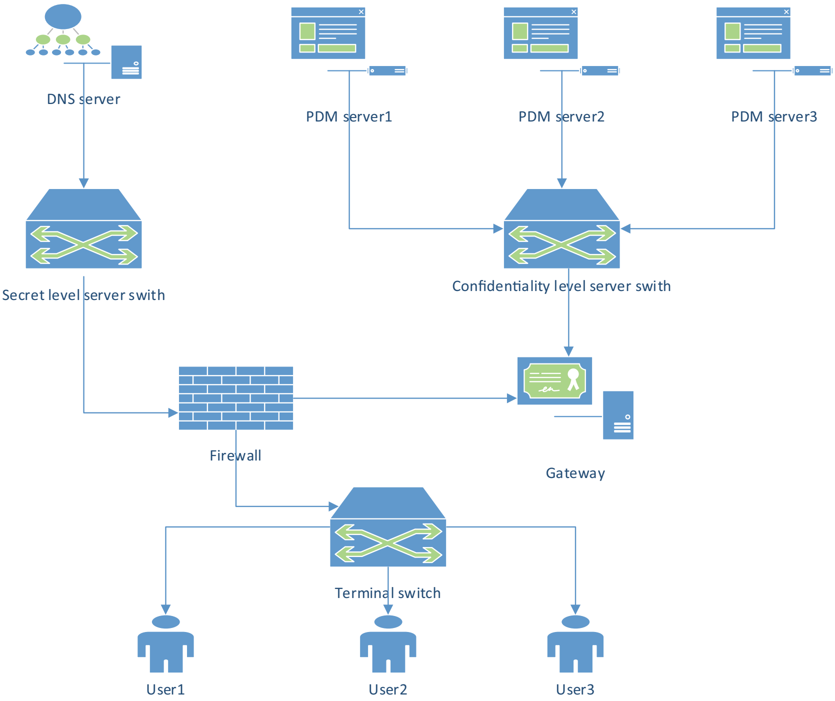


Fig. 5. User Access PDM Server Network Topology Map

6 Conclusion

This article mainly analyzes three load balancing schemes, taking the PDM server cluster and its network environment of the research institute as the background. The goal is to select a product data management system load balancing scheme that is most suitable for the research institute's network environment to achieve user access to the research institute's PDM server evenly distributed to three servers.

The main work and research contents of this article are as follows:

1. A detailed introduction of three load balancing technologies.
2. Analysis of the PDM access network diagram of the research institute's PDM server.
3. Finally, the server polling technology was selected as the load balancing scheme for the research institute.

References

1. Tian Gao. Research on Load Balancing Strategies based on Server Clustering [D]. Tianjin: Tianjin University of Technology, 2016.0
2. Qiao Sun, Buqiao Deng, Zhiqiang Wang, A Scalable Load Balancing Strategy Technology based on Distributed Server Clustering [J]. Telecommunication Science, 2017, 33.

3. Tingting Zhang, Fengkun Wang, Jing Ge. A Load Balancing Algorithm for SDN-WLAN based on Round-Robin Mechanism [J]. Journal of Jiamusi University (Natural Science Edition), 2016, 34 (3): 420-422.
4. Hong Lin. Nginx Research and Programming Implementation of Visual Management Solution for Reverse Proxy [J]. Journal of Guizhou Normal College, 2015, 31(12): 38-41.
5. Guilan Feng, Zhengnan Li. Nginx Application Research of Reverse Proxy in Visual Management Solution for Reverse Proxy in University Website System [J]. Network Security Technology and Applications, 2017, (6).
6. Dacai Chen. Research and Application of High-Concurrency Access Server based on Nginx [D]. Shenyang: Shenyang Institute of Computing Technology, Chinese Academy of Sciences, (2018).
7. Jianming Cui, Jiayi Liu, Chengyong Yang, [J]. Network Delay Cloud Computing Resource Scheduling Algorithm based on Priority Queueing Theory, Journal of Guilin University of Technology, 2017, 37(2): 360-365.
8. Pei Chen, Weidong Ma. Implementation of Load Balancing Algorithm based on Nginx [J]. International Electronic Elements | Int Electr Elem, 2017, 25(19):19-22.
9. Zhuo Liu, Xiangli Zhang. Design and Implementation of Load Balancing Cluster based on Nginx [J]. Journal of Guilin University of Electronic Technology, 2017(6):490-493.
10. Weisen Zhang, Tao Chen, Kang Li. Comparative Study on Nginx High-Concurrency Load Balancing Principles and Strategies [J]. Industrial Control Computer | Ind Control Comput, 2018(1):85-86, 89.
11. Chuan Qin. Improvement and Implementation of Web Server Load Balancing Strategy based on Nginx [D]. Chengdu: Southwest Jiaotong University, 2017.
12. Xudong Huang, Yuan Lin. Simple Deployment of Parallel Environment based on Linux Cluster II. Application Research of Computers. 2004, (11).
13. Yuyan Liu, Mingyu Shen. [J]. Application of LVS Load Balancing Technology in Network Service. Journal of Hefei University of Technology (Natural Science), 2007. 30 (12).
14. Shidong Yao, Research on Linux Server Load Balancing based on LVS Technology. Science & Technology Information, 2009, (23).