Cloud Load Balance

OPS Guidelines

Product Introduction

Contents

OPS Guidelines
Health Check Troubleshooting

Last updated：2017-12-15 16:24:30

1. Layer-4 Troubleshooting

Under TCP protocol, cloud load balancer uses SYN packet for the check, while under UDP protocol, it uses Ping command for the check.

When a CLB back-end CVM port is marked "Unhealthy" in the page, you should conduct troubleshooting using the following procedure:

- Check whether the CLB back-end CVM is configured with a firewall that affects the service. If yes, please disable it.
- Use Netstat command to check if there is a process listening on the back-end CVM port. If no such process is found, restart the service.

2. Layer-7 Troubleshooting

For Layer-7 services (HTTP protocol), when an exception is detected in the health check by a listener process, the troubleshooting can be performed in the following ways:

- CLB's Layer-7 health check service communicates with the back-end CVMs via private network, so you need to log in to the server to check whether the application server port is being listened on normally at the private network address; if not, you should move the listening of application server port to the private network address to ensure the normal communication between the CLB system and back-end CVM.

Assume that the CLB front-end port is 80, the CVM back-end port is 80, and CVM's private IP is 1.1.1.10

The server on Windows system uses the following command:

```
netstat -ano | findstr :80
```

The server on Linux system uses the following command:

```
netstat -anp | grep :80
```

If you can see the listening status at 1.1.1.10:80 or 0.0.0.0:80, the configuration is considered normal.

- Make sure that the relevant port has been opened on back-end CVM, which must be consistent with the back-end port you configured in the CLB listener configuration.

For a Layer-4 CLB, it is considered normal as long as the back-end port telnet gives a response. You can use `telnet 1.1.1.10 80` to test. For a Layer-7 CLB, such HTTP status codes as 200 indicate a normal state. The test is conducted as follows:

On Windows system, you can directly input private IP in a CVM browser to test whether it is normal. In this example, http://1.1.1.10 is input.
On Linux system, you can check whether the status is HTTP / 1.1 200 OK through the `curl-I` command. In this example, `curl -I 1.1.1.10` is used.

- Check whether there is a firewall or other security software inside the back-end CVM. Such software can easily block local IP address of the CLB system, causing the failure of the CLB system to communicate with the back-end CVM.

Check whether the firewall of private network on the CVM allows port 80. You can temporarily disable the firewall for the test.

For Windows system, run the `firewall.cpl` entry to disable the firewall
For Linux system, input `/etc/init.d/iptables stop` to disable the firewall

- Check whether the settings of the CLB health check parameters are correct.
- The recommended test file specified for health check is a simple page in html form and is only used for check return results. Dynamic scripting languages such as php are not recommended.
- Check whether there is a high load on the back-end CVM that leads to slow response of CVM to provide service.

Solutions to Overmuch Client Timewait

Last updated：2018-06-01 17:32:27

Background:

When performing stress test on LB, you may often encounter "connect" failures caused by too much client timewait (all ports are occupied in short time). Here are the reasons and solutions.

Introduction to Linux Parameters:

tcp_timestamps: Whether the tcp timestamps option is enabled. timestamps is negotiated during three-way handshake of TCP. If either party does not support it, the timestamps option will not be adopted by the connection.
tcp_tw_recycle: Whether reclaiming is enabled for tcp in time_wait status
tcp_tw_resuse: If enabled, you can directly reclaim connections which keeps being in time_wait status for more than 1s.

Reasons:

Too much clients in timewait status. The client actively breaks connections, and each connection breaking changes the status of the connection to timewait. And the reclaiming timeout is 60s by default. Generally, in this case, the user enables parameters `tcp_tw_recycle` and `tcp_tw_resuse` for reclaiming of connections in timewait status.
However, because the `tcp_timestamps` option is disabled in the CLB, neither `tcp_tw_recycle` nor `tcp_tw_resuse` that is enabled by the client can take effect, and connections in timewait status cannot be reclaimed quickly. Next we explain the meaning of several linux parameters and reasons why LB cannot enable `tcp_timestamps` .

1. tcp_tw_recycle and tcp_tw_resuse only take effect when tcp_timestamps is enabled.

2. tcp_timestamps and tcp_tw_recycle cannot be enabled simultaneously, because the public network client may fail to access the server through NAT gateway. The reasons are as follows:

If both tcp_tw_recycle and tcp_timestamps are enabled, the timestamp in the socket connect request of CVM with the same source IP must be incremental within 60s. Taking 2.6.32 kernel as an example, the

details are as follows:

```
if (tmp_opt.saw_tstamp &&
    tcp_death_row.sysctl_tw_recycle &&
    (dst = inet_csk_route_req(sk, req)) ! = NULL &&
    (peer = rt_get_peer((struct rtable *)dst)) ! = NULL &&
    peer- >v4daddr == saddr) {
        if (get_seconds() < peer- >tcp_ts_stamp + TCP_PAWS_MSL &&
            (s32)(peer- >tcp_ts -  req- >ts_recent) >
                                TCP_PAWS_WINDOW) {
                NET_INC_STATS_BH(sock_net(sk), LINUX_MIB_PAWSPASSIVEREJECTED);
                goto ↓drop_and_release;
        }
}
```

> tmp_opt.saw_tstamp: The socket supports tcp_timestamp
> sysctl_tw_recycle: tcp_tw_recycle option is enabled in the local system
> TCP_PAWS_MSL: 60s; it indicates that the last TCP communication of the source IP occurred within 60s.
> TCP_PAWS_WINDOW: 1; it indicates that the timestamp of the last TCP communication of the source IP is greater than that of this TCP communication.

1. LB (Layer-7) disables tcp_timestamps because public network clients may fail to access to the server through NAT gateway, as shown in the example below:

a) A quintet is still in the status of time_wait. For allocation policy for port by NAT gateway, the same quintet is re-used within 2MSL, with syn packet sent.

b) When tcp_timestamps is enabled and the following two conditions are met, the syn packet will be dropped (because the timestamp option is enabled, and it is considered as old packet).
i. Time stamp of last time > Time stamp of this time
ii. Packets are received within 24 days (timestamp field is 32-bit and timestamp is updated once per 1 ms by default in Linux. Timestamp echo will occur after 24 days).
Note: This problem is more obvious on the mobile, because all the clients share a limited public IP under the ISP's NAT gateway. The quintet may also be reused within 2MSL. The timestamps sent by different clients may not be incremental.
Taking 2.6.32 kernel as an example, the details are as follows:

```
static inline int tcp_paws_check(const struct tcp_options_received *rx_opt,
                                 int paws_win)
{
      if ((s32)(rx_opt->ts_recent - rx_opt->rcv_tsval) <= paws_win)
            return 1;
      if (unlikely(get_seconds() >= rx_opt->ts_recent_stamp + TCP_PAWS_24DAYS))
            return 1;

      return 0;
}
```

> rx_opt->ts_recent: Time stamp of last time
> rx_opt->rcv_tsval: Time stamp received this time
> get_seconds(): current time
> rx_opt->ts_recent_stamp: Time when the last packet was received

Solutions:

Solutions to the problem of excessive clients in Timewait status:

1. When HTTP uses a short connection (Connection: close), LB disables the connection actively, and the client will not generate timewait.

2. If it is required to use a persistent connection, enable SO_LINGER option of the socket and use rst to disable the connection to avoid the timewait status and achieve fast port reclaiming.

Load Balancer HTTPS Service Performance Test

Last updated：2017-12-04 11:40:50

1. About Cloud Load Balancer (CLB)'s HTTPS Capability

Tencent Cloud's CLB has achieved significant improvement in HTTPS performance based on the full optimization of protocol stack and servers. At the same time, our cooperation with world-leading certificate providers results in a considerable cost-saving in certificates. Tencent Cloud's CLB brings substantial benefits for your business in the following aspects:

1 ） The use of HTTPS does not affect the access speed of client.

2) A single CVM in a cluster features a fast SSL encryption and decryption capability, with the full handshakes reaching up to 65,000 cps. This is at least 3.5 times faster than that when high-performance CPU is relied on, which greatly reduces the server costs, enhances the service capacity at the time of business volume and traffic surges and achieves a stronger computing-based anti-attack capability.

3 ） Support the uninstallation and translation of a variety of protocols. Reduce the stress involved in the adaption to various protocols of the client. The business back-end just needs to support HTTP1.1 to use various versions of protocols such as HTTP2, SPDY, SSL3.0 and TLS1.2.

4 ） One-stop service covering SSL certificate application, monitoring and replacement. By working with world-leading certificate providers including comodo and symantec, we have significantly simplified the certificate application procedures and reduced relevant costs.

5) Anti-CC and WAF capabilities Effectively prevent attacks at application layer such as slow connection, high-frequency targeted attack, SQL injection, website malicious code.

2. Test Purpose

HTTPS service has such advantages as authentication, message encryption and integrity verification. However, achieving secure communication by adding SSL protocol will inevitably cause some performance loss, including a longer latency and consumption of CPU resources for encryption and

decryption. This document provides the test data on performance limits of Tencent Cloud's HTTPS service with SSL encryption and decryption, which can be used for reference and comparison with traditional performance data of HTTPS.

3. Test Environment

- Stress testing tool: wrk 4.0.2
- Tencent Cloud's underlying service environment: Nginx 1.1.6_1.9.9 + Openssl 1.0.2h
- OS of the machine on which Nginx is installed: Linux TENCENT64.site 3.10.94-1-tlinux2-0036.tl2 # 1 SMP Thu Jan 21 03:40:59 CST 2016 x86_64 x86_64 x86_64 GNU / Linux
- OS of other stress-source machines: Linux TENCENT64.site 2.6.32.43-tlinux-1.0.17-default # 1 SMP Tue Nov 17 18:03:12 CST 2015 x86_64 x86_64 x86_64 GNU / Linux

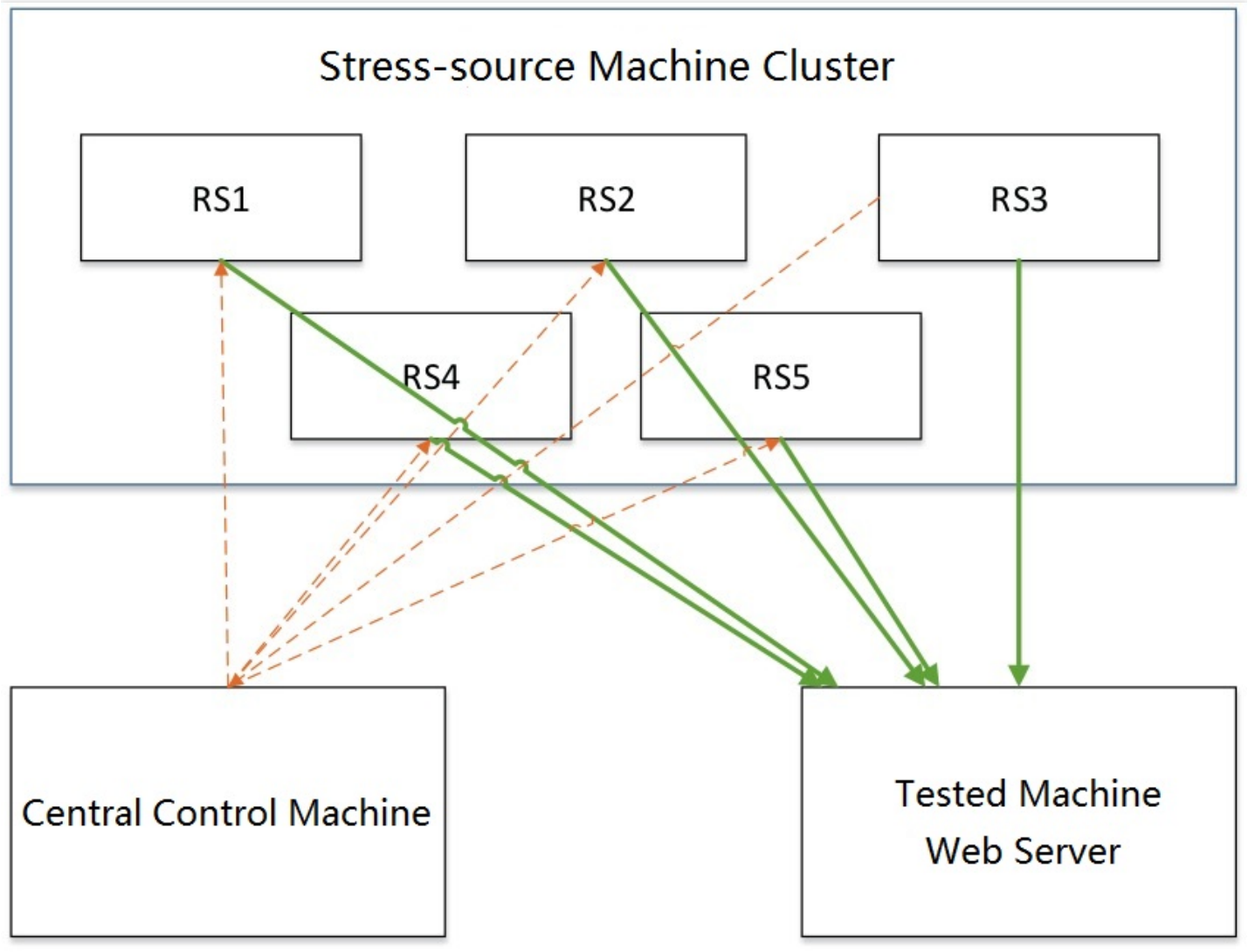
4. WebServer Cluster Test Scheme

The stress from a single stress-source machine is not enough to test the performance limit of Tencent Cloud's HTTPS service. Therefore, multiple stress-source machines are needed to send stress. The test involves three parts:

1) Stress-source machine cluster. Used to exert HTTP/HTTPS stress and output stress testing results of a single machine.

2) Central control machine, which synchronously controls the start and stop of the stress-source machine cluster, obtains the stress testing data from each machine, and aggregates and outputs the data.

3) Tested machine, the CVM hosting Tencent Cloud's HTTPS service. When the performance of webserver is tested, page is returned directly and there is no need to make upstream connection.
The connections are as follows:



5. Performance Testing Data of HTTPS WebServer

Connection Type	Session cache	Packet Size (bytes)	Encryption Suite	Performance (qps)
Persistent	On	230	ECDHE-RSA-AES128-GCM-SHA256	296241
Short	Off	230	ECDHE-RSA-AES128-GCM-SHA256	65630

6. Conclusion on CLB HTTPS Capability Test

According to the above table, Tencent Cloud's HTTPS service supports SSL encryption and decryption, and it has multiple CVM clusters at back-end. A single CVM in a cluster can achieve a performance of up to 65000 qps during full handshake and of about 300000 qps during a persistent connection.

In general, due to the use of SSL protocol, at least an additional full handshake is required for HTTPS protocol, making the latency increase by 2*RTT. In addition, SSL symmetric/asymmetric encryption can consume substantial CPU resource, and RSA's decryption capability is the main barrier to the HTTPS access.

With Tencent Cloud's HTTPS service, users need not to deploy services additionally for SSL encryption and decryption. With no additional cost is charged, the service allows users to easily enjoy powerful business-hosting capacity and anti-attack capability.

Stress Testing FAQ

Last updated：2018-06-01 17:31:46

Taking users' stress test experiences as reference, this document summarizes common performance issues of stress tests, and provides users with troubleshooting solutions and suggestions for stress test.

Stress Test FAQ

1. The public network traffic is not opened for backend CVMs

When you purchase a CVM, if public network traffic is not opened, the forwarding may fail when the public network cloud load balancer is mounted on the CVM.

2. Bandwidth of Backend CVM is Insufficient

If the bandwidth is set too low for the backend CVM, the backend CVM cannot return packets back to the LB when the bandwidth exceeds the threshold, and LB will return 504 and 502 to the client.

3. Insufficient Client Ports

If there are too few clients or the port range of the client is set too small, it may fail to establish the connection due to insufficient ports. Besides, if the keep_alive field is greater than 0 when a persistent connection is established, the connection will keep occupying the port, thus resulting in insufficient client ports.

4. Backend CVM-based Applications Become Performance Bottleneck

After the request reaches the backend CVM through cloud load balancer, the load on the backend CVM itself is balanced normally, but because all applications on the backend CVM depend on other applications such as databases, performance bottlenecks in the databases may also affect the performance of stress test.

5. The Health Status of the Backend CVM is Exceptional

The health status of the backend CVM tends to be ignored when performing stress test. If there is a health check failure or unstable health check status (sometimes good, sometimes bad and changing repeatedly) for backend CVM, it may lead to low performance of stress test.

6. Session Persistence Enabled for Cloud Load Balancer Results in Uneven Backend CVM Traffic Distribution

After session persistence is enabled for cloud load balancer, requests tend to fall on some backend CVMs. As a result, the traffic distribution is uneven, and the performance of the stress test is affected. It is recommended to disable session persistence when performing stress tests.

Suggestions for Stress Tests

Note: The following settings are only used for stress tests of cloud load balancer. It is not necessary to configure the exact same settings for the user's production environment.

- It is recommended to use short connection to perform stress tests on forwarding capability of cloud load balancer.

In general, except for verifying some features such as session persistence, the stress test is mainly designed to verify the forwarding capability of cloud load balancer. Therefore, the short connection can be used to test the processing capability of LBs and backend CVMs.

- It is recommended to use a persistent connection to perform the stress tests on throughput of cloud load balancer, such as the upper limit of bandwidth, long connection service.

In this case, it is recommended to adjust the timeout value of a stress test tool to a smaller threshold. If the timeout period is too long, the average response time will be longer, which will be detrimental to quickly determine whether the stress test level is reached.

- It is recommended to use a static web page provided by backend CVM for stress test to avoid loss caused by application logic itself, such as I/O, DB.

- Disable session persistence feature for listening. Otherwise, the pressure will be concentrated on individual backend CVMs. In addition, when the pressure performance is not satisfactory, you can determine whether the traffic is evenly distributed by checking the monitor data of backend CVM under cloud load balancer.

- Disable health check feature for listening to reduce access requests to backend CVMs from health check.

- Use multiple clients (> 5) for stress tests. Dispersed source IP is better for simulating the actual situation online.