ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 4, No. 6, 192-195 (2019)

www.astesj.com

ASTESJ

ISSN: 2415-6698

Integration of Third-Party Routing Stack to NetScaler CPX

Shreyas K K[*,1], Abhishek H P[1], Sneha M[1], Shobha G[1], Deepak Kumar[2]

[1]*Department of CSE, R. V. College of Engineering, 560059, India*

[2]*Network Professional, Citrix Systems, Bangalore, 560059, India*

ARTICLE INFO

Article history:
Received: 26 August, 2019
Accepted: 16 November, 2019
Online: 05 December, 2019

Keywords:
Application Delivery Controller
Quagga
Virtual IP
NetScaler CPX
Route Health Injection
Border Gateway Protocol

ABSTRACT

The Citrix NetScaler CPX, is one of the many Application Delivery Controllers (ADCs), which is used mainly for load balancing between the application servers in a datacenter network. ADCs are typically placed in the demilitarized zone in an organization's network, handling the connectivity of the external clients to the organization's servers. While performing load balancing between the servers, there is need for better routing between application servers. Route Health Injection (RHI) delegates the control of routing protocol announcement to a server based on the health of a service and the connectivity of the server to the network. This paper discusses on how the current version of NetScaler CPX is integrated with a third-party routing stack, Quagga, to enable the connectivity of the clients to the servers, with the additional feature of Route Health Injection (RHI) for better routing.

1. Introduction

Application delivery controllers (ADC) [1], which are widely deployed as a key fixture in the enterprise, help applications adapt to the networks and protocols that are in place today. ADCs can help ensure high availability of applications by providing seamless failover. This is done by balancing application workloads across a cluster of active servers in one or multiple sites.

One side of the ADC is the server farm capable of providing services and the other side is the Internet with clients and routers. A group of servers providing the similar service can be pooled and assigned a single Virtual Internet Protocol (VIP) address. This enables load balancing the incoming request among one of the servers in the pool.

NetScaler is available as a physical appliance or as software appliance. It supports a varied functionality including load balancing, SSL offloading, HTTP compression, content filtering, etc. NetScaler CPX [2] brings the power of advanced L4-7 application delivery services to containerized microservices infrastructures [3].

Route Health Injection (RHI)[4] allows a router to publish a host route to a globally distributed website. Normally, an IP address must exist on a single host in the Internet. However, with global server load balancing, the same IP address can exist on multiple machines using the Virtual IPs. Thus, a VIP is an IP address that will associate to real servers attached to the load balancer.

In a normal configuration, a router may have multiple paths to the VIP corresponding to a server site. But the router chooses the path with minimum cost. The router's behavior works well when all the servers are available. In case of a failure of the service at a particular site, the VIP must no longer be visible to the rest of the network.

However, if the gateway routers advertise the network routes rather than host routes, then even if the VIP is down, the router still advertises the network to which the VIP belongs, thus leading a path to an unavailable server.

To avoid this problem, only the host route of the servers can be advertised instead of the network route. As a result, paths to the website those are no longer available ages out of the routing tables and only the paths to the available sites exists.

Figure 1 shows an example topology wherein there are two sites serving the same website. When the site at location 1 fails, then the request is routed to the VIP of servers at location 2.

An ADC must be able to take smart real time decisions based on the status/availability of the backend servers. It must effectively

[*] Shreyas K. K., Bangalore, Contact: 6360441200, shreyaskk97@gmail.com

https://dx.doi.org/10.25046/aj040624

divert traffic from any non-responsive backend server and continue to send that traffic to an alternate load balanced server that is available. The decision of advertising and removing a VIP to/from the downstream routers is made based on the availability of a VIP.

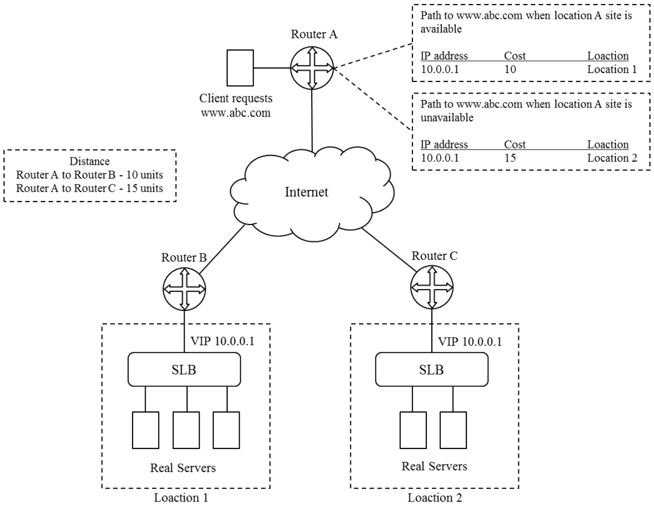


Figure 1: RHI example

To demonstrate the routing in our environment, we use a routing software package Quagga [5] that provides the features of TCP/IP based routing services. It supports routing protocols such as RIP, OSPF, BGP, IS-IS, etc. In addition to the IPv4 support, Quagga also provides support to the IPv6 routing protocols. Quagga is distributed under the GNU General Public License.

2. Existing System

Hogge, Jr. et al. [6] described the use of route health injection using the OSPF protocol. It explains the use of RHI in mobile SMS and MMS technology. SMS and MMS require the Short Message Service Center (SMSC). It explains the advantages of using redundant message servers. The primary message service center and the secondary message service center are typically distributed globally. When the primary service center faces a catastrophic failure, the secondary or the redundant service center is used to serve the SMS and MMS requests.

Naseh et al. [7] describes the technique of having active/standby data center to achieve high availability of applications. In normal conditions, the primary data center is functional and the secondary data center monitors the state of the primary data center. When the primary data center fails, the secondary data center injects the IP address to the Internet and makes the service. Now, any requests from the clients that arise are directed to the secondary data center. This illustration is an example of RHI used for availability in case of a disaster.

Clients request the VIP for a service. The request is then forwarded to one of the physical server based on a load balancing algorithm [8] that runs on the ADC (for example, the ADC might use Round Robin method of allocating user requests in a circular fashion). It is required to make an end to end connection from the client to the servers through the VIP. Thus, a routing stack has to be integrated in the NetScaler CPX, which makes this connection possible. Along with the integration of a routing stack, it will obviously be an added benefit to implement the Route Health Injection (RHI) functionality, wherein, only the active VIPs are advertised to the downstream routers and the inactive ones are removed, hence increasing the overall response time.

The objective is to develop a web application which does the following: monitor the health of NetScaler CPX VIPs, and inject the route using open-source routing engine.

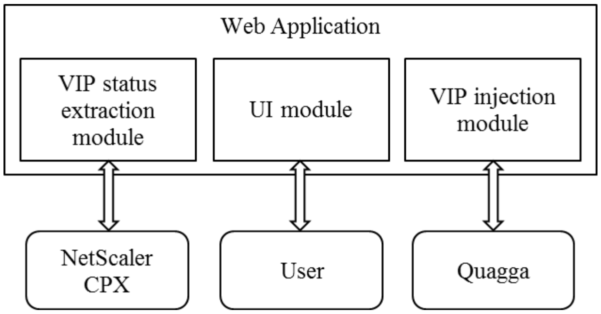


Figure 2: System block diagram

3. Proposed Methodology

To achieve the main objective of establishing an end to end connectivity of the clients to the VIP, the NetScaler CPX is integrated with a routing system as shown in Figure 2. The system is composed of the NetScaler CPX container, a routing engine called Quagga, and a web application. The web application is developed to communicate to NetScaler CPX on one side and the routing engine on the other. The VIP statuses are retrieved from the NetScaler through the VIP extraction module and are parsed to determine its state. The main function of active VIPs injection and inactive VIPs removal is handled next.

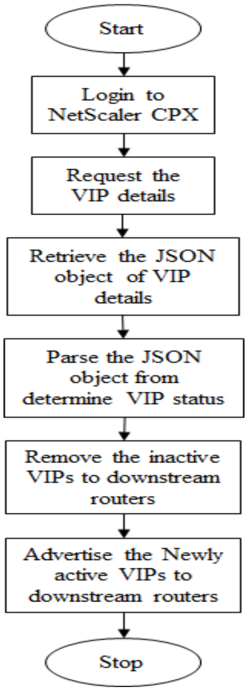

Start
Login to NetScaler CPX
Request the VIP details
Retrieve the JSON object of VIP details
Parse the JSON object from determine VIP status
Remove the inactive VIPs to downstream routers
Advertise the Newly active VIPs to downstream routers
Stop

Figure 3: Flowchart of the system

The VIP injection module handles the task of advertising/removing VIPs to/from the local Quagga routing engine. The active VIPs are advertised as host routes to the routing table if they aren't previously advertised. The inactive VIPs are removed from the routing table if they previously existed. The user can view the details of the active and inactive VIPs through the UI module.

The VIP details are to be retrieved from NetScaler CPX. A request from the web application is sent to NetScaler CPX asking for the VIP details. NetScaler CPX bundles the VIP details in an JSON object and delivers to the web application. The JSON object containing the VIP details are parsed to determine the VIP status. All the inactive VIPs are removes from the local routing table. All the recently active VIPs are advertised to the local routing table.

The BGP further handles the task of updating the routing tables of all neighboring routers and hence, assuring that only the active VIPs are visible throughout the network. The flow of the system is shown in Figure 3.

4. Experimental Analysis and Discussion of Results

The project is based on a system to make connectivity from the clients to the services. Hence, there is no specific evaluation metrics used. The only possible evaluation criteria used to verify the system is the connectivity of the client to the VIP. The system is checked with the existence of connectivity from the clients to the VIPs. It is also verified that only the active VIPs are visible and the inactive ones are not.

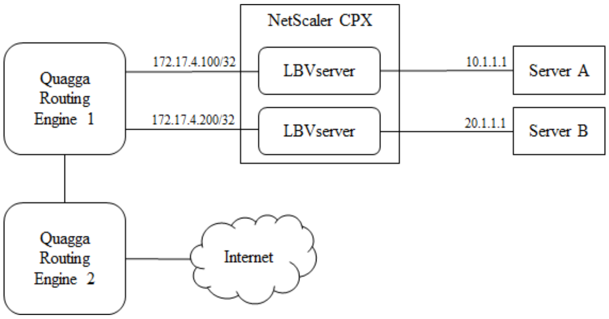
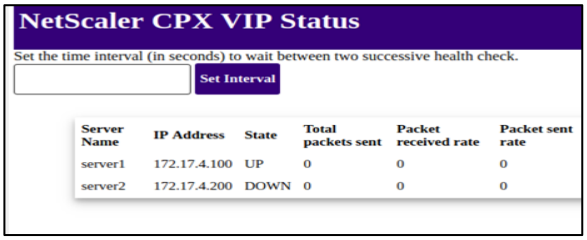


Figure 4: Sample Topology with NetScaler CPX

Consider a simple topology depicted in Figure 4 with NetScaler CPX connected to two servers.

The physical IP addresses of server A and server B are 10.1.1.1/24 and 20.1.1.1/24 respectively. The physical servers are logically grouped based on the service they provide. Hence, the clients have a logical view of the servers. In our case, server1 and server2 are the logical group of servers. The VIPs of server1 and server2 are 172.17.4.100/32 and 172.17.4.200/32 respectively.



NetScaler CPX VIP Status

Set the time interval (in seconds) to wait between two successive health check.

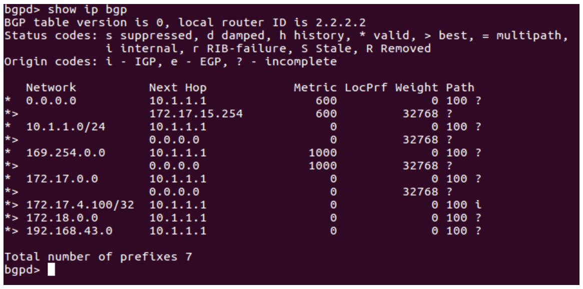
Server Name	IP Address	State	Total packets sent	Packet received rate	Packet sent rate
server1	172.17.4.100	UP	0	0	0
server2	172.17.4.200	DOWN	0	0	0

Figure 5: Web Application – VIP Published

The Quagga routing engine 1 is local to the NetScaler CPX and Quagga routing engine 2 is used to verify the route publishing.

The following two cases illustrate the routing table contents of Quagga Routing Engine 2 which demonstrates the route advertising and removal.

If VIP is UP - VIP 172.17.4.100 is UP and it is published to all neighboring routers' routing table. Any request to the VIP will be reachable. The web application is shown in Figure 5 and the routing table in Figure 6.



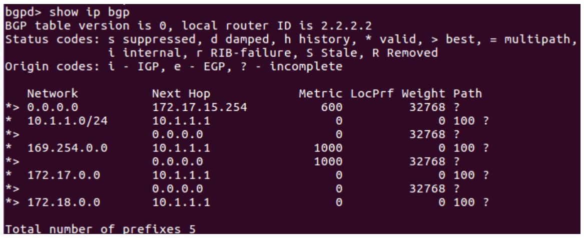
```
bgpd> show ip bgp
BGP table version is 0, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  0.0.0.0          10.1.1.1               600             0 100 ?
*>                  172.17.15.254          600         32768 ?
*  10.1.1.0/24      10.1.1.1                 0             0 100 ?
*>                  0.0.0.0                  0         32768 ?
*  169.254.0.0      10.1.1.1              1000             0 100 ?
*>                  0.0.0.0               1000         32768 ?
*  172.17.0.0       10.1.1.1                 0             0 100 ?
*>                  0.0.0.0                  0         32768 ?
*> 172.17.4.100/32  10.1.1.1                 0             0 100 i
*> 172.18.0.0       10.1.1.1                 0             0 100 ?
*> 192.168.43.0     10.1.1.1                 0             0 100 ?

Total number of prefixes 7
bgpd> █
```

Figure 6: Routing Table – VIP Published

If VIP is DOWN – VIP 172.17.4.100 is DOWN and will be removed from all neighboring routers' routing tables. The VIP is not accessible. The routing table is shown in Figure 7.



```
bgpd> show ip bgp
BGP table version is 0, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          172.17.15.254          600         32768 ?
*  10.1.1.0/24      10.1.1.1                 0             0 100 ?
*>                  0.0.0.0                  0         32768 ?
*  169.254.0.0      10.1.1.1              1000             0 100 ?
*>                  0.0.0.0               1000         32768 ?
*  172.17.0.0       10.1.1.1                 0             0 100 ?
*>                  0.0.0.0                  0         32768 ?
*> 172.18.0.0       10.1.1.1                 0             0 100 ?

Total number of prefixes 5
```

Figure 7: Routing Table – VIP Removed

5. Conclusion

The system is implemented to achieve the main objective of establishing an end to end connectivity of the clients to the VIP along with reducing the response time with RHI. The system, hence, integrated the third party routing stack to the NetScaler CPX. The NetScaler CPX is configured and the routing engine set up with the web application. The active VIPs are advertised while the inactive VIPs are removed from the routing tables.

However, the system is compatible only with NetScaler CPX deployed environment and in the Citrix enterprise network. It also is designed specifically to the Quagga routing engine. Thus, any necessity to work with different routing engines requires modification. The system also has limited configuration capability.

6. Future work

The future enhancements considered to the project tries to overcome the limitations. They are described as follows:

- Development of a Wrapper API to work with any kind of Routing Engine.
- Development of the system capable for many other configuration capabilities through the web interface.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all. We would like to thank Head of Department, Computer Science & Engineering, and the Principal, R V College of Engineering, for their moral support towards completing our project work. We also thank the entire team of Citrix for providing us an opportunity to carry out the project and for their continuous support and guidance.

References

[1] Citrix ADC data sheet, https://www.citrix.com/content/dam/citrix/en_us/documents/data-sheet/citrix-adc-data-sheet.pdf, Accessed Date: 01-12-2019

[2] Citrix Networking CPX Datasheet, https://www.citrix.com/en-in/products/citrix-adc/resources/netscaler-cpx-data-sheet.html, Accessed Date: 01-12-2019

[3] Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio, "An Updated Performance Comparison of Virtual Machines and Linux Containers", IEEE, 978-1-4799-1957-4, 171-172, 2015

[4] Netscaler - Dynamic routing and route health injection, https://www.virtualdesktopdevops.com/netscaler/netscaler-dynamic-routing.html, Accessed Date: 01-12-2019

[5] Quagga Routing Suite, www.quagga.net, Accessed Date: 01-12-2019

[6] Hogge, Jr., "OSPF Routing to Geographically Diverse Applications using OSPF and Route Health Injection", United States Patent, Patent No. US 7707308 B1

[7] Naseh, "Disaster Recovery for Active Standby Data Center using Route Health and BGP", United States Patent, Patent No. US 7710865 B2

[8] Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma, "Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology, 38, 2008 pp. 269- 272.