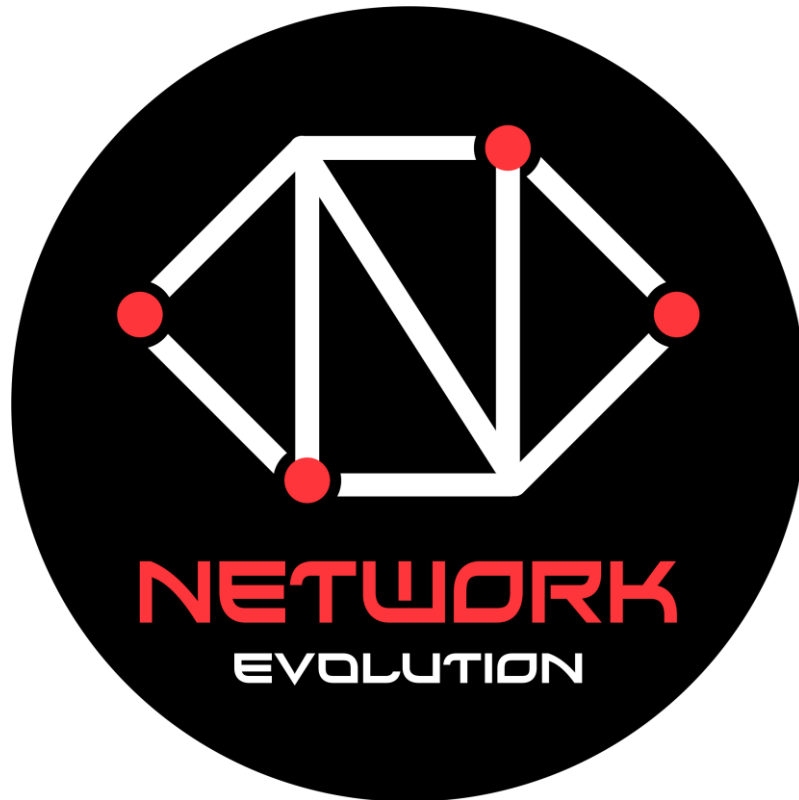


LOAD BALANCING WEB SERVER NGINX WITH HAPROXY ON CENTOS 7



Vassa Metayasha

S1 TT 05D – 17101161 – Divisi NFV

**PROGRAM STUDI S1 TEKNIK TELEKOMUNIKASI
FAKULTAS TEKNIK TELEKOMUNIKASI DAN ELEKTRO
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2019

BAB I

DASAR TEORI

1. Load Balancing

Load Balancing adalah sebuah teknik untuk melakukan distribusi beban trafik pada dua atau lebih jalur koneksi secara seimbang sehingga setiap jalur koneksi akan menerima beban trafik yang sama. Load balancing ini sangat banyak digunakan oleh perusahaan - perusahaan skala besar. Load balancing biasanya sering digunakan untuk kebutuhan trafik web yang sangat tinggi seperti web e-commerce. Load balancing biasanya digunakan pada saat sebuah server jika user yang melakukan akses server tersebut telah melebihi jumlah maksimal. Ketika user yang mengakses melebihi kapasitas maka dengan menggunakan teknik load balancing, beban trafik tersebut akan dialihkan ke server yang lain.

Berikut beberapa teknik algoritma yang bisa digunakan oleh haproxy pada load balancing :

- a. Round-Robin. Algoritma round-robin mendistribusikan beban kepada semua server anggota cluster sehingga masing - masing server mendapat beban yang sama dalam waktu yang sama. Round-robin cocok saat server anggota cluster memiliki kemampuan proccessing yang sama, jika tidak beberapa server bisa jadi menerima request lebih dari kemampuan proccessing server itu sendiri sedang yang lainnya hanya mendapat beban lebih sedikit dari resource yang dimiliki.
- b. Weighted round-robin. Algoritma weighted round-robin melakukan perhitungan perbedaan kemampuan proccessing dari masing masing server anggota cluster. Administrator memasukan secara manual parameter beban yang akan ditangani oleh masing masing server anggotacuster, kemudian scheduling sequence secara otomatis dilakukan berdasarkan beban server. Request kemudian diarahkan ke server yang berbeda sesuai dengan round-robin scheduling sequence.
- c. Least-connection. Algoritma Least-connection melakukan pengiriman request pada server anggota cluster, berdasarkan pada server mana yang memiliki fewest connections (koneksi paling sedikit).

- d. Load-based. Algoritma Load-based mengirimkan paket request ke server anggota cluster berdasarkan server mana yang memiliki beban terkecil.

Pada load balancing kali ini akan menggunakan teknik algoritma round robin agar beban pada setiap server sama. ^[1]

2. Web Server

Server atau Web server adalah sebuah software yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada klien yang dikenal dan biasanya kita kenal dengan nama web browser (Mozilla Firefox, Google Chrome) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web dan pada umumnya akan berbentuk dokumen HTML. Fungsi utama Server atau Web server adalah untuk melakukan atau akan mentransfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. halaman web yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi. pemanfaatan web server berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web termasuk yang di dalam berupa teks, video, gambar dan banyak lagi.

Salah satu web server yang digunakan kali ini adalah nginx. NGINX adalah software web server yang open source. Ketika pertama kali dirilis, NGINX hanya berfungsi sebagai HTTP web serving saja. Namun sekarang, software tersebut juga berperan sebagai reverse proxy, HTTP load balancer, dan email proxy untuk IMAP, POP3, dan SMTP. ^[2]

3. Virtualisasi Server (Cloud Computing)

Cloud Computing adalah sistem komputerisasi berbasis jaringan/internet, dimana suatu sumber daya, software, informasi dan aplikasi disediakan untuk digunakan oleh komputer lain yang membutuhkan. Mengapa konsep ini bernama komputasi awan atau cloud computing? Ini karena internet sendiri bisa dianggap sebagai sebuah awan besar (biasanya dalam skema network, internet dilambangkan sebagai awan) yang berisi sekumpulan besar komputer yang saling terhubung, jadi cloud computing bisa diartikan sebagai komputerisasi berbasis sekumpulan komputer yang saling terhubung.

Cloud computing bisa dianggap sebagai perluasan dari virtualisasi. Perusahaan bisa menempatkan aplikasi atau sistem yang digunakan di internet, tidak mengelolanya secara internal. Contoh cloud computing untuk versi public adalah layanan-layanan

milik Google seperti Google Docs dan Google Spreadsheet. Adanya kedua layanan tersebut meniadakan kebutuhan suatu aplikasi office untuk pengolah kata dan aplikasi spreadsheet di internal perusahaan. Contoh cloud computing untuk keperluan non public adalah Amazon EC2 (Amazon Elastic Compute Cloud). Amazon menyediakan komputer induk, kita bisa mengirim dan menggunakan sistem virtual dan menggunakannya dalam jangka waktu dan biaya sewa tertentu. ^[3]

BAB II

MANFAAT DAN TUJUAN

A. MANFAAT

- 1) Dapat membangun server secara virtual dengan menerapkan teknik load balancing
- 2) Dapat menerapkan sistem load balancing untuk mengarahkan trafik ke beberapa server
- 3) Dapat membedakan fungsi antara nginx (web server) dengan Haproxy (load balancer)

B. TUJUAN

- 1) Mengetahui Kinerja Load Balancing menggunakan metode algoritma round robin.
- 2) Melakukan pengukuran throughput, request loss, delay dan jitter pada nginx dengan haproxy.
- 3) Mengetahui perbedaan penggunaan antara haproxy dengan nginx.

BAB III

KONFIGURASI PERANGKAT

1. Siapkan 3 VM yang sudah terinstall OS Centos 7, dan berikan masing – masing VM RAM sebesar 512 MB dan harddisk 10gb:

link install Centos 7 : https://www.server-world.info/en/note?os=CentOS_7&p=install
link download Centos 7 : https://www.server-world.info/en/note?os=CentOS_7&p=download

Matikan Windows Firewall terlebih dahulu pada Laptop/PC anda dengan cara :

- Buka Control panel
- Pilih system and security
- Pilih windows firewall
- Kemudian pilih Turn Windwos Firewall on or off
- Kemudian pilih turn off seluruhnya.

2. VM 1 = Loadbalancer (NAT dan Adapter hanya host)

Login dengan Root Privileges

Install terlebih dahulu nano dengan cara :

```
yum -y install nano
```

cek port yang tersedia :

```
nmcli d
```

maka akan muncul enp0s3 (DHCP) dan enp0s8 (static)

kemudian setting IP adapter hanya host (static) :

```
nano /etc/sysconfig/network-scripts/ifcfg-enp0s8
```

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp0s8
UUID=819d07d6-9e6a-4204-a138-bb2b36d39992
DEVICE=enp0s8
ONBOOT=yes
IPADDR=192.168.1.102
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
```

Jika sudah save dengan cara tekan CTRL + X, kemudian Y.

Lalu setting IP NAT (DHCP):

```
nano /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

ubah pada bagian paling bawah menjadi ONBOOT="yes".

Selanjutnya restart network dengan cara :

`systemctl restart network`

3. VM 2 = nginx1

Login dengan Root Privileges

Install terlebih dahulu nano dengan cara :

`yum -y install nano`

cek port yang tersedia :

`nmcli d`

maka akan muncul enp0s3 (DHCP) dan enp0s8 (static)

kemudian setting IP adapter hanya host (static) :

`nano /etc/sysconfig/network-scripts/ifcfg-enp0s8`

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp0s8
UUID=819d07d6-9e6a-4204-a138-bb2b36d39992
DEVICE=enp0s8
ONBOOT=yes
IPADDR=192.168.1.104
NETMASK=255.255.255.0
```

Jika sudah save dengan cara tekan CTRL + X, kemudian Y.

Lalu setting IP NAT (DHCP):

`nano /etc/sysconfig/network-scripts/ifcfg-enp0s3`

ubah pada bagian paling bawah menjadi ONBOOT="yes".

Selanjutnya restart network dengan cara :

`systemctl restart network`

4. VM 3 = nginx2

Login dengan Root Privileges

Install terlebih dahulu nano dengan cara :

`yum -y install nano`

Cek port yang tersedia :

`nmcli d`

Maka akan muncul enp0s3 (DHCP) dan enp0s8 (static)

Kemudian setting IP adapter hanya host (static) :

`nano /etc/sysconfig/network-scripts/ifcfg-enp0s8`

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6_INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp0s8
UUID=819d07d6-9e6a-4204-a138-bb2b36d39992
DEVICE=enp0s8
ONBOOT=yes
IPADDR=192.168.1.105
NETMASK=255.255.255.0
```

Jika sudah save dengan cara tekan CTRL + X, kemudian Y.

Lalu setting IP NAT (DHCP):

`nano /etc/sysconfig/network-scripts/ifcfg-enp0s3`

ubah pada bagian paling bawah menjadi ONBOOT="yes".

Selanjutnya restart network dengan cara :

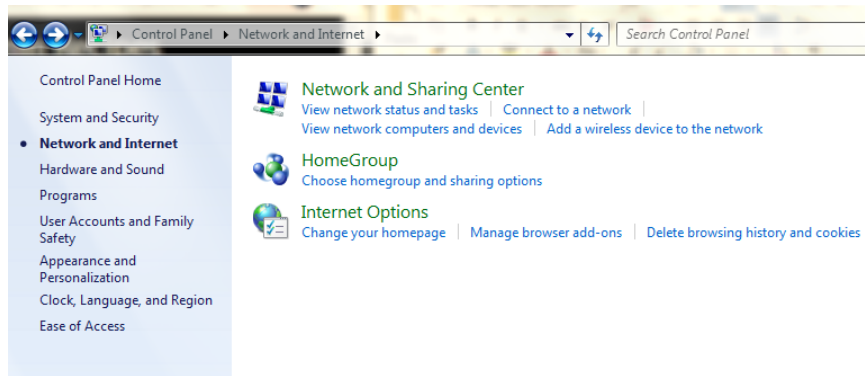
`systemctl restart network`

KEMUDIAN SETTING PADA VIRTUAL BOX ADAPTER PADA LAPTOP/PC ANDA:

1. Buka control panel
2. Pilih Network dan Internet



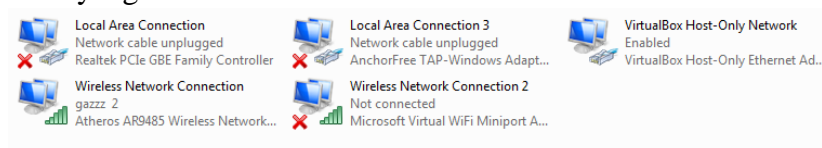
3. Pilih network and sharing



4. Pilih change adapter settings



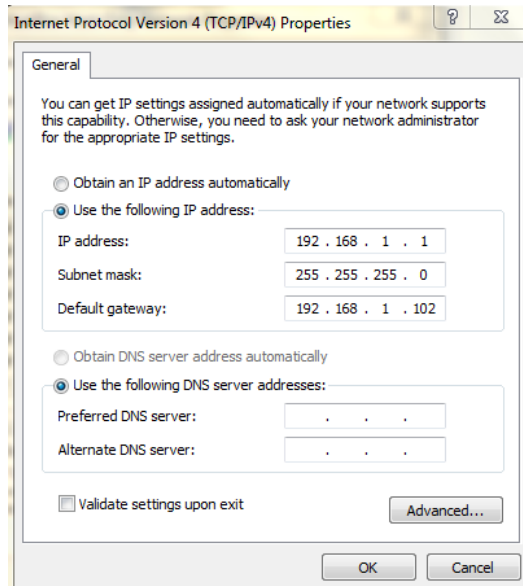
5. Pilih yang virtual box klik 2x



6. Pilih properti

7. Kemudian cari "Internet Protocol Version 4 (TCP/Ipv4), klik 2x

8. Setting seperti dibawah :



9. Kemudian OK

MULAI SETTING LOAD BALANCER

STEP 1 :

- Pada semua VM lakukan ini (loadbalancer, nginx1, dan nginx2)

1. Nano /etc/hosts

2. Tambahkan seperti dibawah :

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.1.102 loadbalancer
192.168.1.104 nginx1.loadbalance.me nginx1
192.168.1.105 nginx2.loadbalance.me nginx2
```

3. CTRL + X, kemudian Y (untuk keluar dan save)

STEP 2 :

- Pada @192.168.1.102 (loadbalancer)

1. yum -y update

2. kemudian install haproxy :

yum -y install haproxy

3. lakukan konfigurasi pada haproxy

4. pertama backup file haproxy.cfg :

mv /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.orig

5. kemudian buka file haproxy.cfg :

nano /etc/haproxy/haproxy.cfg

6. Paste Konfigurasi dibawah, cara pastenya pastikan kalian konfigurasinya tidak di virtual box, melainkan menggunakan aplikasi Putty :

```
#-----
# Global settings
#-----
```

```

global
    log      127.0.0.1 local2    #Log configuration

    chroot   /var/lib/haproxy
    pidfile  /var/run/haproxy.pid
    maxconn  4000
    user     haproxy             #Haproxy running under user and group "haproxy"
    group    haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats

#-----
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#-----
defaults
    mode                http
    log                 global
    option               httplog
    option               dontlognull
    option http-server-close
    option forwardfor    except 127.0.0.0/8
    option               redispatch
    retries              3
    timeout http-request 10s
    timeout queue        1m
    timeout connect      10s
    timeout client       1m
    timeout server       1m
    timeout http-keep-alive 10s
    timeout check        10s
    maxconn              3000

#-----
#HAProxy Monitoring Config
#-----
listen haproxy3-monitoring *:8080          #Haproxy Monitoring run on port
8080
    mode http
    option forwardfor
    option httpclose
    stats enable
    stats show-legends

```

```

stats refresh 5s
stats uri /stats                #URL for HAProxy monitoring
stats realm Haproxy\ Statistics
stats auth howtoforge:howtoforge #User and Password for login to the
monitoring dashboard
stats admin if TRUE
default_backend app-main        #This is optionally for monitoring backend

```

```
#-----
```

```
# FrontEnd Configuration
```

```
#-----
```

```
frontend main
```

```

bind *:80
option http-server-close
option forwardfor
default_backend app-main

```

```
#-----
```

```
# BackEnd roundrobin as balance algorithm
```

```
#-----
```

```
backend app-main
```

```

balance roundrobin                #Balance algorithm
option httpchk HEAD / HTTP/1.1\r\nHost:\ localhost #Check the server
application is up and healthy - 200 status code
server nginx1 192.168.1.104:80 check      #Nginx1
server nginx2 192.168.1.105:80 check      #Nginx2

```

7. kemudian keluar dari konfigurasi dan save (CTRL + X, kemudian Y)
8. Selanjutnya lakukan konfigurasi pada rsyslog untuk haproxy
Nano /etc/rsyslog.conf
9. Kemudian hilangkan tanda pagar # (komentar) pada bagian dibawah :
\$ModLoad imudp
\$UDPServerRun 514
10. Dan tambahkan dibawah UDPServerRun 514, jika ingin menggunakan ip yang lebih spesifik :
\$UDPServerAddress 127.0.0.1
11. Save dan keluar dari konfigurasi (CTRL + X, kemudian Y)
12. Kemudian buat file konfigurasi haproxy baru :
Nano /etc/rsyslog.d/haproxy.conf
13. Copy paste konfigurasi dibawah :
local2.=info /var/log/haproxy-access.log #For Access Log
local2.notice /var/log/haproxy-info.log #For Service Info - Backend,
loadbalancer
14. Save dan keluar dari konfigurasi (CTRL + X, kemudian Y)
15. Sekarang restart rsyslog dan start haproxy :

```
Systemctl restart rsyslog
Systemctl start haproxy
Systemctl enable haproxy
```

STEP 3 :

Pada step ini proses install nginx dari epel repository. **Lakukan ini pada VM nginx1 dan VM nginx2 :**

1. Install epel repository :
`yum -y install epel-release`
2. Kemudian install nginx :
`yum -y install nginx`
3. Setelah selesai proses instalasi, kemudian ubah isi yang ada pada File index.html :
`Cd /usr/share/nginx/html/`
4. Kemudian ketik pada server nginx1 :
`echo "<h1>nginx1.loadbalance.me</h1>" > index.html`
5. Pada server nginx2 :
`echo "<h1>nginx2.loadbalance.me</h1>" > index.html`
6. Selanjutnya start nginx dengan cara :
`systemctl enable nginx`
`systemctl start nginx`
7. Kemudian setting firewall agar zone menjadi public (**LAKUKAN INI PADA SEMUA VM**):
Lakukan pada VM loadbalancer, VM nginx1, VM nginx2.

```
[root@loadbalancer ~]# firewall-cmd --permanent --add-service=http
success
[root@loadbalancer ~]# firewall-cmd --reload
success
```

STEP 4 (boleh dilakukan, boleh tidak dilakukan) rekomendasi dilakukan :

Pada step kali ini, matikan jaringan NAT yang terhubung pada Server nginx1 dan server nginx2 saja dan biarkan NAT pada loadbalancer menyala.

LAKUKAN INI PADA VM nginx1 dan VM nginx2 :

1. untuk keluar dari seluruh folder
`cd` (ketik saja cd kemudian enter)
2. nmcli d
3. kali ini enp0s3 akan dimatikan (diputus) koneksinya
4. buka :
`nano /etc/sysconfig/network-scripts/ifcfg-enp0s3`
5. kemudian ubah ONBOOT="yes" menjadi ONBOOT="no"
6. save dan keluar dari konfigurasi (CTRL + X, kemudian Y)
7. kemudian restart network
`systemctl restart network`

STEP 5 : melakukan test

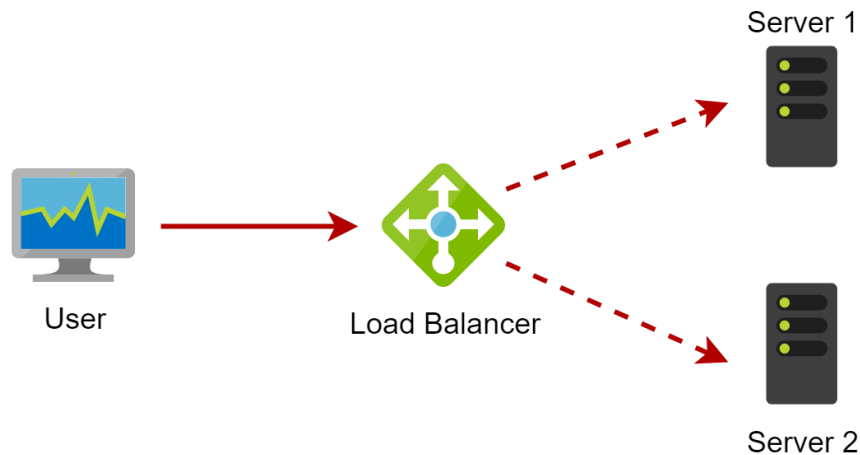
1. Buka Google chrome (Rekomendasi)
2. Ketik 192.168.1.102 (sebagai IP Loadbalancer)
3. Kemudian enter
4. Maka akan muncul Tulisan yang ada pada file index.html tadi
5. Kemudian refresh menggunakan F5

Maka akan berubah tampilan dari server nginx1 menjadi tampilan server nginx2.

BAB IV

HASIL DATA DAN ANALISIS

1. Gambar diagram alur proses dan penjelasan.



Gambar 1.1 Topologi Load balancing ^[4]

Load balancing kali ini menggunakan 1 client, 1 load balancer (untuk mengatur pembagian beban server), 2 web server (menggunakan Nginx). Dimana client disini adalah laptop pribadi dan loadbalancer serta web server disini adalah berbentuk virtual (diinstall dengan Virtual Box). Dimana pada load balancer diinstall packet service yaitu Haproxy yang dimana haproxy ini diberikan algoritma round robin untuk mengatur pemberian beban serta membagi beban kepada server. Dimana client akan mengakses server dengan menggunakan alamat ip yang nantinya load balancer disini akan bertugas untuk mengatur jalur akses dari client menuju server yang diinginkan.

2. Service yang dianalisis dan penjelasan.

Pada analisa Quality of Service kali ini menggunakan aplikasi wireshark dimana aplikasi ini bisa menghitung jumlah throughput, packet loss, delay, dan jitter. Dimana kali ini akan membandingkan QoS antara Load Balancing server menggunakan haproxy dengan algoritma round robin dengan Web Server Nginx (Single VM).

a. Throughput

Throughput atau bisa dikatakan sebagai bandwidth yang sebenarnya adalah kemampuan sebenarnya suatu jaringan mengirimkan data, yang dinamis sesuai keadaan dari trafik jaringan.

b. Packet Loss

Adalah perbandingan seluruh paket data yang hilang dengan total paket data dikirimkan antara sumber dan tujuan pengiriman.

c. Delay

Adalah waktu tunda sebuah paket data yang disebabkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuan.

d. Jitter

Adalah kumpulan dari semua delay terjadi dalam jaringan pada saat waktu tertentu.

3. Hasil Data dan penjelasan.

First packet:	2019-08-29 12:58:09
Last packet:	2019-08-29 12:58:11
Elapsed:	00:00:01
Capture	
Hardware:	Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz (with SSE4.2)
OS:	64-bit Windows 7 Service Pack 1, build 7601
Application:	Dumpcap (Wireshark) 3.0.3 (v3.0.3-0-g6130b92b0ec6)
Interfaces	
<u>Interface</u>	<u>Dropped packets</u> <u>Capture filter</u> <u>Link type</u> <u>Packet size limit</u>
VirtualBox Host-Only Network	0 (0 %) none Ethernet 262144 bytes
Statistics	
<u>Measurement</u>	<u>Captured</u> <u>Displayed</u> <u>Marked</u>
Packets	25 25 (100.0%) —
Time span, s	1.782 1.782 —
Average pps	14.0 14.0 —
Average packet size, B	332 332 —
Bytes	8304 8304 (100.0%) 0
Average bytes/s	4660 4660 —
Average bits/s	37 k 37 k —

Gambar 3.1 QoS nginx (single VM)

First packet:	2019-08-29 12:56:16
Last packet:	2019-08-29 12:56:18
Elapsed:	00:00:01
Capture	
Hardware:	Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz (with SSE4.2)
OS:	64-bit Windows 7 Service Pack 1, build 7601
Application:	Dumpcap (Wireshark) 3.0.3 (v3.0.3-0-g6130b92b0ec6)
Interfaces	
<u>Interface</u>	<u>Dropped packets</u> <u>Capture filter</u> <u>Link type</u> <u>Packet size limit</u>
VirtualBox Host-Only Network	0 (0 %) none Ethernet 262144 bytes
Statistics	
<u>Measurement</u>	<u>Captured</u> <u>Displayed</u> <u>Marked</u>
Packets	24 24 (100.0%) —
Time span, s	1.700 1.700 —
Average pps	14.1 14.1 —
Average packet size, B	373 373 —
Bytes	8950 8950 (100.0%) 0
Average bytes/s	5265 5265 —
Average bits/s	42 k 42 k —

Gambar 3.2 QoS Load Balancing Haproxy

Kali ini saya akan membandingkan throughput, packet loss, delay, dan jitter dengan menjalankan load balancing server dan web server (single VM) dalam waktu 1,7 detik. Bisa dilihat pada gambar diatas, keduanya mendapatkan packet captured hampir sama dalam waktu 1,7 detik tetapi mendapatkan jumlah throughput, delay dan jitter yang berbeda.

e. Throughput

Throughput pada gambar diatas terletak pada bagian Average bits/s. Pada gambar pertama pada bagian load balancing, throughput yang didapatkan sebanyak 42k bit/s berbeda dengan throughput yang didapatkan pada web server (single VM), pada bagian single VM mendapatkan lebih banyak jumlah throughput sebanyak 37k bit/s.

f. Packet Loss

Pada pengukuran keduanya tidak terdapat packet loss dikarenakan waktu yang dilakukan terlalu singkat serta jumlah file yang diberikan sangat kecil sehingga tidak terjadi packet yang hilang.

g. Delay

Pengukuran delay secara manual bisa dilakukan dengan menyimpan data TCP kemudian waktu yang didapatkan dihitung dengan cara “time 2 – time 1” maka akan mendapatkan hasil delay pada setiap captured. Kemudian total delay bisa dilihat dalam gambar diatas yaitu pada load balancing dilakukan selama 1,700 detik serta 1,782 detik pada nginx (single VM). Selanjutnya jika ingin mengukur rata – rata delay dengan cara “total delay / (jumlah captured – 1)” diukur pada Ms.Excel agar lebih mudah. Seperti gambar dibawah :

[illegible]

	No.	"Time"	"Source"	"Destination"	"Protocol"	"Length"	"Info"	time 2	time 1	delay
1	1,	"0.000000","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	0,00159	0	0,00159
2	2,"0.001586","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	0,06878	0,00159	0,06719	
3	3,"0.068777","	"0a:00:27:00:00:13","	"Broadcast","	"ARP","	"42","	"Who has 192.168.1.10? Tell 192.168.1.1"	0,20167	0,06878	0,1329	
4	4,"0.201673","	"192.168.1.1","	"192.168.1.104","	"TCP","	"54","	"51977 > 80 [ACK] Seq=521 Ack=180 Win=254 Len=0"	0,21412	0,20167	0,01245	
5	5,"0.214123","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	0,21543	0,21412	0,0013	
6	6,"0.215425","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	0,41678	0,21543	0,20135	
7	7,"0.416779","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	0,41734	0,41678	0,00056	
8	8,"0.417340","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	0,54868	0,41734	0,13134	
9	9,"0.548680","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	0,54974	0,54868	0,00106	
10	10,"0.549744","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	0,67468	0,54974	0,12494	
11	11,"0.674679","	"0a:00:27:00:00:13","	"Broadcast","	"ARP","	"42","	"Who has 192.168.1.10? Tell 192.168.1.1"	0,7013	0,67468	0,02662	
12	12,"0.701295","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	0,70181	0,7013	0,00052	
13	13,"0.701813","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	0,87861	0,70181	0,1768	
14	14,"0.878614","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	0,87961	0,87861	0,001	
15	15,"0.879614","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	1,06087	0,87961	0,18125	
16	16,"1.060865","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	1,06186	1,06087	0,00099	
17	17,"1.061856","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	1,2325	1,06186	0,17065	
18	18,"1.232501","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	1,23307	1,2325	0,00057	
19	19,"1.233074","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	1,41526	1,23307	0,18218	
20	20,"1.415256","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	1,41616	1,41526	0,0009	
21	21,"1.416160","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	1,57577	1,41616	0,15961	
22	22,"1.575769","	"192.168.1.1","	"192.168.1.104","	"HTTP","	"574","	"GET / HTTP/1.1 "	1,5769	1,57577	0,00113	
23	23,"1.576902","	"192.168.1.104","	"192.168.1.1","	"HTTP","	"233","	"HTTP/1.1 304 Not Modified "	1,67472	1,5769	0,09782	
24	24,"1.674719","	"0a:00:27:00:00:13","	"Broadcast","	"ARP","	"42","	"Who has 192.168.1.10? Tell 192.168.1.1"	1,78174	1,67472	0,10702	
25	25,"1.781743","	"192.168.1.1","	"192.168.1.104","	"TCP","	"54","	"51977 > 80 [ACK] Seq=5201 Ack=1791 Win=254 Len=0"		1,78174		
27										
28									total del	1,78174
29									rata2 del	0,07424

h. Jitter

delay 2	delay 1	jitter	
0,000141	0,00145	-0,00131	
0,008206	0,000141	0,008065	
0,005609	0,008206	-0,0026	
0,145461	0,005609	0,139852	
0,004979	0,145461	-0,14048	
0,174378	0,004979	0,169399	
0,006644	0,174378	-0,16773	
0,166978	0,006644	0,160334	
0,00862	0,166978	-0,15836	
0,172983	0,00862	0,164363	
0,005151	0,172983	-0,16783	
0,164667	0,005151	0,159516	
0,006818	0,164667	-0,15785	
0,161054	0,006818	0,154236	
0,004844	0,161054	-0,15621	
0,156181	0,004844	0,151337	
0,003352	0,156181	-0,15283	
0,172663	0,003352	0,169311	
0,005958	0,172663	-0,16671	
0,123189	0,005958	0,117231	
0,002621	0,123189	-0,12057	
0,19781	0,002621	0,195189	
	0,19781		
	total jitte	0,19636	
	rata2 jitte	0,008537	

Gambar 3.5 Pengukuran Jitter Load Balancing

delay 2	delay 1	jitter	
0,06719	0,00159	0,06561	
0,1329	0,06719	0,06571	
0,01245	0,1329	-0,12045	
0,0013	0,01245	-0,01115	
0,20135	0,0013	0,20005	
0,00056	0,20135	-0,20079	
0,13134	0,00056	0,13078	
0,00106	0,13134	-0,13028	
0,12494	0,00106	0,12387	
0,02662	0,12494	-0,09832	
0,00052	0,02662	-0,0261	
0,1768	0,00052	0,17628	
0,001	0,1768	-0,1758	
0,18125	0,001	0,18025	
0,00099	0,18125	-0,18026	
0,17065	0,00099	0,16965	
0,00057	0,17065	-0,17007	
0,18218	0,00057	0,18161	
0,0009	0,18218	-0,18128	
0,15961	0,0009	0,15871	
0,00113	0,15961	-0,15848	
0,09782	0,00113	0,09668	
0,10702	0,09782	0,00921	
	0,10702		
	total jitte	0,10544	
	rata2 jitte	0,00439	

Gambar 3.6 Pengukuran Jitter Nginx

BAB IV

KESIMPULAN

Berdasarkan data yang telah dikumpulkan dapat dilihat bahwa dalam waktu 1,7 detik throughput yang didapatkan oleh load balancing haproxy lebih banyak dibandingkan dengan nginx (single VM). Dapat dikatakan bahwa sistem load balancing menggunakan 3 buah Mesin VM lebih baik dibandingkan dengan menggunakan single VM, karena dengan menggunakan sistem load balancing server tidak akan kesulitan dalam melayani banyaknya request dari client dikarenakan terdapat load balancer yang akan mengatur kemana request dari client akan dilanjutkan.

DAFTAR PUSTAKA

- [1] R. Mufrizal, "Belajar Load Balancing Dengan Haproxy, Docker, dan Spring Boot," [Online]. Available: <https://rizkimufrizal.github.io/belajar-load-balancing-dengan-haproxy-docker-dan-spring-boot/>. [Accessed 28 Agustus 2019].
- [2] IDCcloudHost, "Pengertian Web Server dan Fungsinya," [Online]. Available: <https://idcloudhost.com/pengertian-web-server-dan-fungsinya/>. [Accessed 28 Agustus 2019].
- [3] M. V. Sugianto, "Keuntungan Teknologi Virtualisasi Cloud Computing," [Online]. Available: <https://www.excellent.co.id/product-services/vmware/keuntungan-teknologi-virtualisasi-cloud-computing/>. [Accessed 28 Agustus 2019].
- [4] R. Ariyolo, "Konfigurasi Dasar Server Load Balancer di Alibaba Cloud," [Online]. Available: <https://www.ariyolo.id/konfigurasi-dasar-server-load-balancer-di-alibaba-cloud>. [Accessed 29 Agustus 2019].

