

RANCANG BANGUN SISTEM LOAD BALANCING MYSQL MENGUNAKAN LINUX VIRTUAL SERVER DENGAN ALGORITMA ROUND ROBIN

Chandra Efendi Lumban Tobing^{*)}, Adian Fatchur Rochim, and Maman Somantri

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail : fendi_carter@yahoo.co.id

Abstrak

Permintaan basis data akan meningkat drastis sesuai kebutuhan musiman atau popularitas suatu aplikasi informasi. Permintaan yang besar dalam waktu hampir atau bersamaan akan membuat server MySQL tunggal mengalami overload hingga kegagalan. Penelitian ini melakukan perancangan sistem dan pengujian sistem Load Balancing sebagai solusi mengatasi permintaan MySQL beban berat. Penelitian ini menggunakan pendekatan sistem Linux Virtual Server sebagai Load Balancer. Penjadwalan pembagian beban menggunakan algoritma round robin. Linux Virtual Server juga dirancang sebagai antarmuka MySQL Cluster. Sistem Load Balancing yang dibentuk dibandingkan unjuk kerjanya dengan sistem MySQL tunggal. Hasil pengujian menunjukkan Linux Virtual Server mampu menjadi antarmuka MySQL Cluster dan mengatasi kemungkinan kegagalan satu titik. Pengujian unjuk kerja menunjukkan Linux Virtual Server memiliki waktu tanggap lebih besar daripada sistem MySQL tunggal. Pengujian unjuk kerja menunjukkan Linux Virtual Server mampu menangani beban berat karena permintaan multiuser.

Kata Kunci : MySQL, Load Balancing, Linux Virtual Server

Abstract

Database request will increase drastically as seasonal needs or popularity of an information application. High request on the same time will make single MySQL server experience overload then system failure. This research perform system design and system testing of Load Balancing system as a solution to overcome MySQL heavy request. This research use Linux Virtual Server system approach as Load Balancer. Load Balancing scheduling use round robin algorithm. Linux Virtual Server also designed to become interface for MySQL Cluster. Load Balancing system performance will be compared with single MySQL system. The the tests result show Linux Virtual Server capable of being MySQL Cluster interface and overcome the probability of single point of failure. Performance test shows Linux Virtual Server has higher response time than the single MySQL system's. Performance test shows Linux Virtual Server capable to handle heavy load from multiuser request.

Keywords : MySQL, Load Balancing, Linux Virtual Server

1. Pendahuluan

Load Balancing merupakan kemampuan untuk menyebarkan beban dari pemrosesan sebuah aplikasi ke sejumlah sistem terpisah untuk meningkatkan unjuk kerja (performance) secara umum dalam memproses permintaan masuk. *Load Balancing* menyebarkan sebagian dari proses ke sistem atau perangkat lain yang independen dan menangani penyebaran secara concurrent. Mekanisme *Load Balancing* memiliki keunggulan dalam mengurangi banyaknya proses yang dikerjakan oleh satu atau server utama. Mekanisme tersebut menjadikan sistem

dapat menangani lebih banyak permintaan dan meningkatkan performa[8].

Penelitian sebelumnya [9][12] telah berhasil membentuk sistem *High Availability* MySQL Cluster. MySQL Cluster mampu menangani kegagalan *server* MySQL dan sekaligus replikasi secara cepat. Penelitian tersebut berada hanya dalam lingkup sistem MySQL Cluster. Penerapan MySQL Cluster dengan server jamak sebagai penyedia basis data, belum dilakukan dalam penelitian tersebut. *Load Balancing* merupakan solusi untuk melayani *client* yang terhubung ke MySQL Cluster yang memiliki server jamak.

Linux Virtual Server (LVS) merupakan *highly scalable* dan *highly available server* yang dibangun di atas sebuah cluster dari *real server*. LVS menjalankan *load balancer* yang bekerja di atas sistem operasi Linux. Arsitektur *cluster* LVS bekerja *fully transparent* (transparan secara penuh) untuk *end user*. *End user* berinteraksi dengan sistem *cluster* LVS seolah-olah terdapat hanya satu *server* berunjuk kerja tinggi [22].

LVS memiliki tiga teknik *Load Balancing IP* tersedia di dalam *Load Balancer*, salah satunya adalah metode *direct routing*. Pendekatan via *direct routing* memproses setengah koneksi *client-to-server client-to-server*. Paket yang diterima LVS langsung diteruskan ke *real server*. *Real server* meneruskan tanggapan paket tersebut menggunakan jalur atau jaringan yang berbeda tanpa melalui LVS[23]. Penjadwalan paket LVS salah satunya menggunakan algoritma *round robin*. Algoritma *round robin* meneruskan setiap permintaan ke server yang ditunjuk dengan metode sekuensial berulang: A, B, C, A, B, C, dst[19].

Unjuk kerja merujuk pada cara sebuah komputer/sistem merespon pada beban kerja (*workload*) tertentu[20]. Terdapat beberapa parameter utama dalam mengukur unjuk kerja, salah satunya adalah waktu tanggap (*response time*). Waktu tanggap adalah banyaknya waktu yang digunakan untuk menyelesaikan suatu tugas[19].

Makalah ini memaparkan desain dan pengujian unjuk kerja LVS sebagai solusi beban berat. Unjuk kerja sistem LVS dibandingkan dengan sistem MySQL tunggal. Parameter utama yang digunakan adalah waktu tanggap. Waktu tanggap diperoleh dengan menguji sistem terhadap sejumlah besar *query* MySQL yang dibentuk oleh sejumlah *user* yang mengakses dalam waktu bersamaan.

2. Metode

2.1. Perancangan Sistem Secara Umum

Perancangan sistem ini secara umum membentuk dua jenis sistem MySQL. Sistem pertama disebut “MySQL Tanpa Load Balancing”, untuk melayani permintaan MySQL server tunggal. Sistem kedua disebut “MySQL Dengan Load Balancing”, untuk melayani permintaan MySQL dengan fitur *Load Balancing*. Perancangan sistem menggunakan **10 unit** komputer. Satu unit digunakan sebagai server MySQL sistem “MySQL Tanpa Load Balancing”. Lima unit digunakan dalam sistem MySQL Cluster, penyusun “MySQL Dengan Load Balancing”. Dua unit komputer digunakan sebagai web server. Web server dengan MySQL server dirancang terpisah oleh jaringan. Dua unit komputer digunakan sebagai Load Balancer. Komputer pertama menjadi Load Balancer utama. Komputer kedua menjadi Load Balancer cadangan.

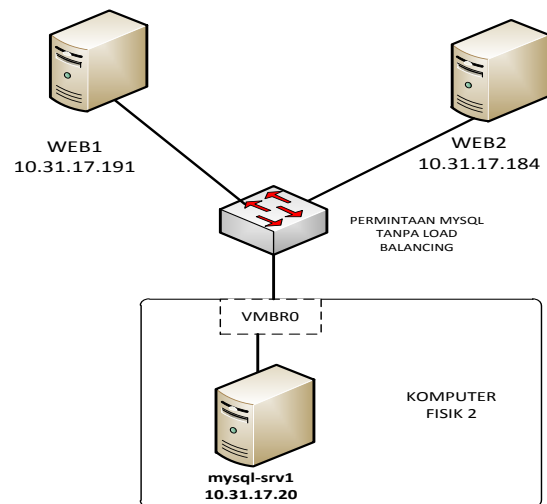
Penelitian ini merancang nama komputer yang digunakan dalam tahap perancangan maupun implementasi dan pengujian. Nama komputer ditunjukkan pada tabel 1.

Tabel 1. Rancangan nama server

No.	Nama	Fungsi
1	Web1	Web Server
2	Web2	Web Server
3	LB1	Load Balancer Aktif
4	LB2	Load Balancer Pasif
5	mysql-srv1	Server “MySQL Tanpa Load Balancing” Management Node MySQL Cluster
6	MGM1	SQL Node1 + NDB Node1 / Real Server
7	slave1	SQL Node2 + NDB Node2 / Real Server
8	slave2	SQL Node3 + NDB Node3 / Real Server
9	slave3	SQL Node4 + NDB Node4 / Real Server
10	slave4	

2.2. Perancangan Sistem “MySQL Tanpa Load Balancing”

Perancangan server “MySQL Tanpa Load Balancing” merupakan konfigurasi umum sistem web server dan MySQL server. Konfigurasi ini menerapkan pemisahan web server dengan MySQL. Konfigurasi ini bertujuan agar satu komputer server MySQL dapat melayani beberapa komputer web server.



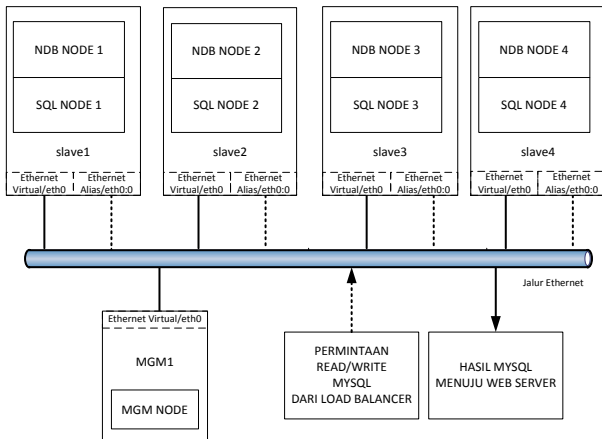
Gambar 1. Rancangan konfigurasi “MySQL Tanpa Load Balancing”

Gambar 1 memperlihatkan tiga buah web server yang terhubung ke satu MySQL server.

2.2. Perancangan Sistem MySQL Cluster

Perancangan MySQL Cluster menggunakan lima buah komputer. Satu buah untuk management node (*MGM node*). Empat buah untuk *NDB node* dan *SQL node*

(mysqld). Penggunaan empat buah komputer ditujukan untuk mengatasi kemungkinan terjadi kegagalan pada satu buah komputer. Jika terjadi kegagalan pada satu buah komputer, tiga buah komputer sisanya tetap dapat digunakan sebagai penyedia MySQL untuk proses Load Balancing. Gambar 2. menunjukkan skema rancangan sistem MySQL Cluster.



Gambar 2. Perancangan MySQL Cluster

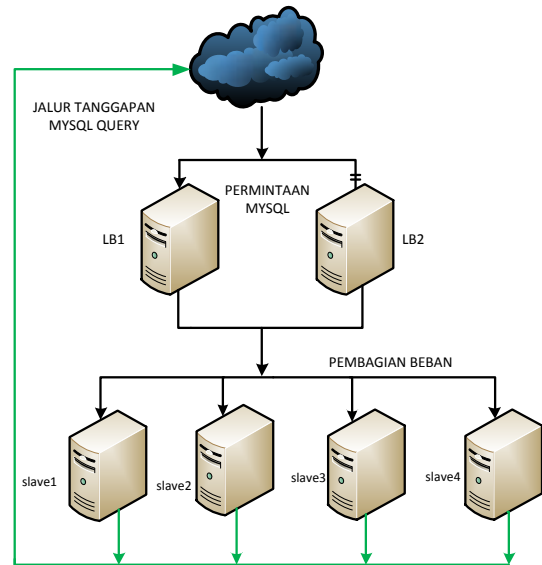
Empat SQL Node bekerja sebagai *Real Server* pada sistem “MySQL Dengan Load Balancing”. Empat NDB Node bekerja sebagai storage sistem baik MySQL Cluster maupun sistem “MySQL Dengan Load Balancing”.

2.3. Perancangan Load Balancer

Perancangan *Load Balancer* menggunakan dua buah komputer fisik. Komputer pertama (LB1) bekerja sebagai *Load Balancer* utama atau *Load Balancer aktif*. Komputer kedua (LB2) bekerja sebagai cadangan komputer pertama. Kedua komputer juga dirancang saling mem-backup, sehingga dapat mengantisipasi kemungkinan kegagalan satu titik.

Dua komputer *Load Balancer* menggunakan aplikasi Piranha. Piranha memiliki dua aplikasi penyusun utama. Aplikasi pertama adalah *Linux Virtual Server (LVS)* atau dikenal dengan sebutan *Linux Director*. LVS bekerja sebagai *Load Balancer*. Aplikasi kedua adalah *nanny*. *Nanny* bekerja sebagai monitor untuk menjaga ketersediaan real server maupun *Load Balancer* cadangan.

Perancangan ini menggunakan metode *packet forwarding* (penerusan paket) tipe *Direct Routing* yang disediakan oleh LVS. LVS menerapkan VIP sebagai alias dari alamat IP nyata milik komputer *Load Balancer*. Perancangan Real Server menggunakan VIP sebagai alias dari alamat IP tiap SQL node. Perancangan ini memerlukan pencegahan penyebaran alamat Media Access Control (MAC) milik *Real Server*. Pencegahan penyebaran MAC menggunakan aplikasi “arptables_jf”.



Gambar 3. Perancangan Load Balancer

Gambar 3 menunjukkan diagram sistem “MySQL Dengan Load Balancing”. Empat slave merupakan komputer hasil perancangan MySQL Cluster. Perancangan sistem “MySQL Dengan Load Balancing” menggunakan satu jaringan lokal yaitu 10.31.17.0.

2.4. Perancangan Pengujian Proses Load Balancing

Pengujian proses *Load Balancing* menggunakan senarai PHP sederhana. Senarai tersebut menggunakan proses *query* pada MySQL. Proses *query* meliputi *open connection*, *select query*, dan *close connection*. Rangkaian proses yang dikenal dengan satu *query session* dilakukan secara berulang. Perancangan ini membatasi pengulangan 10000 query. Berkas PHP tersebut diakses oleh situs yang telah dipasang. Hasil tanggapan MySQL dianalisa menggunakan perangkat lunak Wireshark. Wireshark ditempatkan di Web1. Wireshark menangkap semua paket yang berkaitan dengan aktifitas MySQL. Pengujian menggunakan Wireshark menganalisa algoritma *Round Robin* bekerja dengan baik.

2.5. Perancangan Pengujian Unjuk Kerja LVS

Pengujian unjuk kerja menggunakan aplikasi SysBench. Aplikasi SysBench mampu mensimulasikan pengujian *multiuser*. *Multiuser* dibentuk dengan menggunakan opsi *multithread* pada pengujian Online Transaction Processing (OLTP) SysBench. Pengujian OLTP SysBench menguji kehandalan dan kemampuan dari sebuah sistem MySQL. SysBench dirancang untuk menguji kemampuan “MySQL Tanpa Load Balancing” dan “MySQL Dengan Load Balancing”. SysBench menguji kemampuan MySQL mengatasi *multiple user* dalam waktu bersamaan. Banyaknya *thread* mewakili

banyaknya *Multiple user*. Pengujian dirancang progresif dengan memulai jumlah thread sejumlah 10 hingga batas kemampuan sistem.

3. Hasil dan Analisa

3.1. Persiapan Sistem Server

Tahap awal implementasi adalah mempersiapkan sistem dan konfigurasi server. Setiap server menggunakan sistem operasi Linux Centos 6.3. Semua server MySQL berada dalam satu buah komputer fisik. Server-server MySQL tersebut merupakan server virtual.

Tahap selanjutnya adalah menetapkan alamat IP untuk setiap server. Tabel 2 menunjukkan daftar konfigurasi alamat IP untuk setiap server.

Tabel 2. Pengalamatan IP semua server

Nama Server	Alamat IP
Web1	10.31.17.191
Web2	10.31.17.184
mysql-srv1	10.31.17.20
MGM1	10.31.17.202
slave1	10.31.17.203
slave2	10.31.17.204
slave3	10.31.17.205
slave4	10.31.17.206
LB1	10.31.17.183
LB2	10.31.17.184

3.2. MySQL Tanpa Load Balancing

Sistem “MySQL Tanpa Load Balancing” menggunakan MySQL versi 5.5.29. *Library* PERL perlu dipasang sebelum menginstal MySQL. MySQL versi 5.5.29 diunduh dari situs resmi MySQL. MySQL yang telah berhasil diunduh diinstal menggunakan proses instalasi RPM.

Basis data untuk situs *Web Server* diletakan di mysql-srv1. Nama basis data tersebut adalah WP1 dan WP2.

3.3. MySQL Dengan Load Balancing

Sistem “MySQL Dengan Load Balancing” menggunakan MySQL Cluster versi NDB 7.2.9. Instalasi MySQL Cluster juga membutuhkan *library* PERL. MySQL Cluster diunduh situs resmi MySQL. “MySQL Cluster server” di instal pada semua slave dan MGM1. “MySQL Cluster client” hanya diinstal pada keempat slave.

Konfigurasi utama MySQL Cluster terletak pada MGM1. Konfigurasi utama tersebut terdapat pada `/var/lib/mysql-cluster/config.ini`. Konfigurasi pendukung terletak pada setiap slave. Fungsi konfigurasi pendukung adalah, memperkenalkan diri pada Management Node bahwa slave tersebut adalah anggota MySQL Cluster. MGM1 akan menggunakan slave tersebut sesuai tugas yang diatur dalam berkas konfigurasi utama.

Proses selanjutnya adalah menjalankan MySQL Cluster. MySQL Cluster dijalankan dengan urutan mulai dari MGM *node*, semua NDB *node* dan semua SQL *node*.

Pengujian proses kerja dan unjuk kerja “MySQL Dengan Load Balancing” membutuhkan basis data buatan. Basis data “test” dengan tabel “t2” digunakan pada pengujian proses kerja. Basis data “sptest” digunakan pada pengujian unjuk kerja. Setiap tabel yang direplikasi mewajibkan penggunaan opsi `engine = ndbcluster`.

3.4. Implementasi Web Server

Aplikasi *web server* dalam penelitian ini menggunakan Apache2 untuk Centos. *Package* tambahan yang wajib diinstal antara lain, php5, php-mysql, mysql-client, mysql-odbc, php5-cli, php5-common.

Kedua *web server* menggunakan *Content Management System* (CMS) Wordpress. Tampilan awal Wordpress ditunjukkan pada gambar 4.



Gambar 4. Tampilan awal Wordpress

3.5. Implementasi Load Balancing

Penelitian ini menggunakan LVS sebagai Load Balancer. Aplikasi LVS tersedia dalam kelompok *package* Piranha bentukan Red Hat. Selain LVS, Piranha menyediakan “pulse” sebagai manajemen proses seluruh LVS dan “nanny” sebagai monitor *Real Server*. Setelah proses instalasi LVS menggunakan beberapa langkah konfigurasi dasar agar dapat bekerja dengan baik. Langkah tersebut adalah sebagai berikut.

- Mematikan SELinux (Security Enhanced Linux)
- Mengatur *forwarding* Centos
- Mematikan servis dan *startup* IPTables
- Mengatur *startup* LVS

3.5.1. Konfigurasi Sisi Load Balancer

Piranha menyediakan manajemen LVS via *web*. Tahap awal konfigurasi LVS adalah menentukan *primary* IP (LB1) dan metode *forwarding*. Gambar 5 menunjukkan

Global Setting LVS. Primary IP yang digunakan adalah 10.31.17.183. Metode forwarding yang digunakan adalah "direct routing".



Gambar 5. Tampilan Global Setting Piranha

Fitur Redundancy LVS mengantisipasi kemungkinan kegagalan satu titik. Implementasi Redundancy menggunakan 10.31.17.185 sebagai Redundant IP. Gambar 6 menunjukkan halaman pengaturan Redundancy.



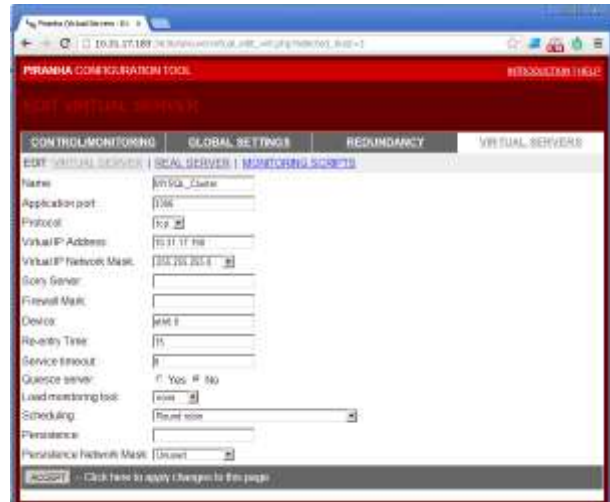
Gambar 6. Halaman REDUNDANCY

Komponen utama LVS adalah pengaturan Virtual Server. Gambar 7 menunjukkan proses pembentukan Virtual Server.



Gambar 7. Pembentukan Virtual Server

Virtual Server yang dibentuk di "EDIT" dan diaktifkan. Alamat Virtual IP (VIP) yang digunakan adalah 10.31.17.190. Algoritma penjadwalan menggunakan "Round Robin". Gambar 8 menunjukkan pengaturan Virtual Server.



Gambar 8. Halaman EDIT VIRTUAL SERVER

Anggota Real Server diatur pada halaman EDIT REAL SERVER yang ditunjukkan pada gambar 9.



Gambar 9. Halaman "EDIT REAL SERVER"

Konfigurasi monitoring scripts memungkinkan LVS memenejemen status kesehatan MySQL. MySQL yang tidak bisa digunakan (down) akan di deteksi oleh nanny dan dibuang dari tabel routing. Gambar 10 menunjukkan konfigurasi "Monitoring Scripts".



Gambar 10. Halaman EDIT MONITORING SCRIPTS

3.5.2. Konfigurasi Sisi Real Server

Setiap Real Server perlu mengijinkan Monitoring Scripts mengakses MySQL. Akses MySQL ditetapkan dengan memberikan grant usage pada user yang ditetapkan berkas /root/mysql_mon.sh milik LB1.

VIP 10.31.17.190 diterapkan pada setiap server slave. Penetapan VIP menggunakan teknik IP alias milik Centos 6. Pencegahan penyebaran MAC Real Server menggunakan aplikasi "arpables_jf". Aturan "arpables_jf" adalah sebagai berikut.

```
arpables -A IN -d 10.31.17.190 -j DROP
arpables -A OUT -d 10.31.17.190 -j
mangle --mangle-ip-s 10.31.17.203
arpables -A OUT -s 10.31.17.190 -o
eth0 -j mangle --mangle-ip-s
10.31.17.203
```

3.6. Pengujian Proses Kerja

Pengujian Proses Kerja menguji LVS menerapkan algoritma round robin. Pengujian proses kerja menggunakan program PHP buatan. Program PHP tersebut melakukan perulangan 10000 kali proses query MySQL. Query MySQL ditujukan membaca basis data "test" dengan tabel "t2". Alamat IP tujuan adalah 10.31.17.190. Program PHP diletakan dan dijalankan di Web2. Hasil pengujian berupa data penangkapan paket oleh Wireshark. Wireshark diinstal dan dijalankan pada Web1. Rangkaian data hasil tangkapan ditunjukkan pada tabel 3.

Tabel 3. Rangkaian data hasil tangkapan Wireshark

MAC Sumber	MAC Tujuan	Info
36:13:ec:34:f7:de	6c:62:6d:52:f8:f0	Request Query
12:66:de:2d:20:53	36:13:ec:34:f7:de	Response
36:13:ec:34:f7:de	6c:62:6d:52:f8:f0	Request Query

66:e2:8e:32:8e:e1	36:13:ec:34:f7:de	Response
36:13:ec:34:f7:de	6c:62:6d:52:f8:f0	Request Query
76:e1:4a:4a:13:f1	36:13:ec:34:f7:de	Response
36:13:ec:34:f7:de	6c:62:6d:52:f8:f0	Request Query
4e:4d:91:6a:05:de	36:13:ec:34:f7:de	Response
36:13:ec:34:f7:de	6c:62:6d:52:f8:f0	Request Query
12:66:de:2d:20:53	36:13:ec:34:f7:de	Response

Tabel 3 LVS menunjukkan proses akan berulang pada sesi ke-5 yang ditandai samanya MAC asal sesi pertama dan sesi ke-5. Hasil pengujian ini menunjukkan LVS mampu membagi beban menggunakan algoritma Round Robin.

3.7. Pengujian Unjuk Kerja

Pengujian unjuk kerja menggunakan perangkat lunak SysBench. Pengujian unjuk kerja melakukan uji ketahanan sistem MySQL terhadap permintaan Multiuser. SysBench mensimulasikan Multiuser dengan melakukan pengujian Multithread. Parameter yang digunakan adalah waktu tanggap (response time) dan jumlah thread. Jumlah query dibatasi pada 50000 request.

SysBench memiliki dua proses utama dalam pengujian. Proses pertama adalah mempersiapkan isi basis data bahan pengujian. Persiapan basis data menggunakan perintah sebagai berikut.

```
[root@LB2 ~]# sysbench --test=oltp --
mysql-table-engine=myisam --oltp-
table-size=700000 --mysql-
host=10.31.17.20 --mysql-port=3306 -
-mysql-user=user --mysql-
password=user123 prepare
```

Proses kedua adalah menguji sistem MySQL menggunakan basis data yang sudah dibentuk. Pengujian SysBench menggunakan perintah sebagai berikut.

```
[root@LB2 ~]# sysbench --num-threads=10 -
-max-requests=50000 --max-time=0 --
test=oltp --oltp-test-mode=simple --
oltp-read-only=on --oltp-auto-
inc=off --oltp-table-size=700000 --
oltp-dist-type=uniform --db-
driver=mysql --mysql-
host=10.31.17.190 --mysql-db=sbtest
--mysql-user=user --mysql-
password=user123 --mysql-table-
engine=ndbcluster run
```

Pengujian dilakukan secara bertahap dimulai dengan thread = 10. Tabel 4.5 menunjukkan hasil pengujian "MySQL Tanpa Load Balancing".

Tabel 4. Hasil pengujian unjuk kerja “MySQL Tanpa Load Balancing”

Jumlah Thread	Transaction/s	Total Time (s)	Keterangan
10	25480.12	1.9623	
11	21564.54	2.3186	
12	21422.63	2.3340	
13	-	-	MySQL Down
350	-	-	MySQL Down
500	-	-	MySQL Down
1000	-	-	MySQL Down

Tabel 5 menunjukkan hasil pengujian “MySQL Dengan Load Balancing”. Gambar 11 menunjukkan grafik unjuk kerja “Sistem MySQL Tanpa Load Balancing”. “Sistem MySQL Tanpa Load Balancing akan bekerja semakin lambat seiring dengan meningkatnya jumlah thread.



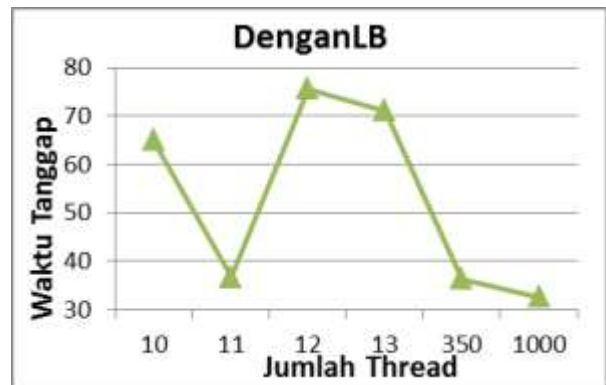
Gambar 11. Grafik unjuk kerja “Sistem MySQL Tanpa Load Balancing”

Tabel 5. Hasil pengujian unjuk kerja “MySQL Dengan Load Balancing”

Jumlah Thread	Transaction/s	Total Time (s)	Keterangan
10	768.45	65.0662	
11	1359.88	36.7678	
12	660.86	75.6586	
13	703.79	71.0443	Tetap Mampu
350	1375.94	36.3416	Tetap Mampu
1000	1529.47	32.6930	Tetap Mampu

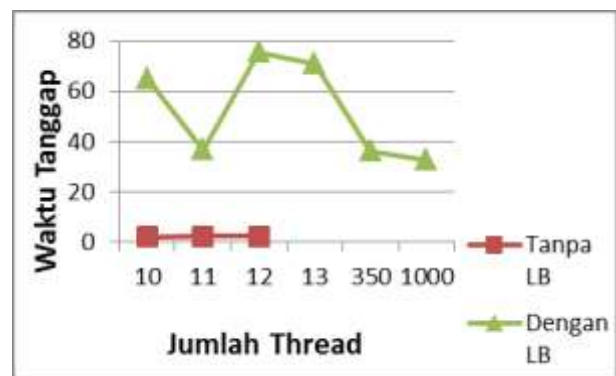
Tabel 5 menunjukkan sistem “MySQL Dengan Load Balancing” tetap mampu mengatasi jumlah thread lebih dari “13”. Hasil tersebut menunjukkan sistem *Load Balancing* LVS mampu mengatasi user dalam jumlah besar yang mengakses sistem secara bersamaan. Gambar 12 menunjukkan unjuk kerja “Sistem MySQL Dengan Load Balancing”. “Sistem MySQL Dengan Load

Balancing akan bekerja semakin baik jika jumlah thread semakin besar.



Gambar 12. Grafik unjuk kerja “Sistem MySQL Dengan Load Balancing”

Grafik perbandingan unjuk kerja kedua sistem ditunjukkan pada gambar 13.



Gambar 13. Perbandingan hasil pengujian unjuk kerja

Gambar 13 menunjukkan perbedaan yang besar antara waktu tanggap “MySQL Tanpa Load Balancing” dengan “MySQL Dengan Load Balancing”.

Faktor-faktor penyebab sistem LVS memiliki waktu tanggap lebih besar antara lain. Latency meningkat sesuai dengan jumlah komputer yang dilewati semakin banyak. Caching MySQL Cluster belum terbentuk sempurna dalam jangka waktu yang singkat. Perangkat keras yang digunakan diterapkan dalam sistem virtual. Satu server dipaksa melayani 6 komputer virtual.

4. Kesimpulan

Dari hasil penelitian dan pembahasan pada bab sebelumnya dapat ditarik beberapa kesimpulan antara lain, Linux Virtual Server mampu menjadi solusi antarmuka sistem MySQL Cluster. Linux Virtual Server mampu melakukan kegiatan *Load Balancing* (membagi beban) paket MySQL ke empat server MySQL. Pengujian

menggunakan proses kerja *Load Balancing* memperlihatkan Linux Virtual Server mampu meneruskan paket berdasarkan algoritma Round Robin. Pengujian unjuk kerja menunjukkan Linux Virtual Server memiliki waktu tanggap lebih besar terhadap permintaan MySQL daripada sistem “MySQL Tanpa Load Balancing” dengan nilai 65 detik : 2 detik pada pengujian 12 *thread*. Pengujian unjuk kerja menunjukkan Linux Virtual Server mampu menangani jumlah thread lebih besar daripada sistem “MySQL Tanpa Load Balancing”, yang ditunjukkan dengan kegagalan sistem pada thread ke 13 untuk “MySQL Tanpa Load Balancing”. Kemampuan menangani multithread mewakili kemampuan Linux Virtual Server menangani jumlah user lebih, dalam waktu bersamaan daripada “MySQL Tanpa Load Balancing”.

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah pengujian sistem *Load Balancing* perlu dilakukan pada perangkat keras fisik tidak pada sistem virtual. Pengujian tersebut perlu dibuat skema yang lebih terarah dengan parameter yang mewakili kebutuhan dunia nyata. Dan pengujian unjuk kerja *multiuser* perlu menggunakan user yang nyata untuk mendapat hasil yang lebih valid.

Referensi

- [1]. Bell, C., *Expert MySQL*, Second Edition. New York City: Apress Media LLC., 2013
- [2]. Gilmore, W. J., *Beginning PHP and MySQL from Novice to Professional*, Fourth Edition, New York City: Apress Media LLC., 2008.
- [3]. Kpytov, A., *SysBench Manual*, Cupertino, California: MySQL AB., 2009, Tersedia pada: <http://sysbench.sourceforge.net/docs/>
- [4]. Linux Virtual Server Project, *IPVS*, Linux Virtual Server Project, 2012, <http://kb.linuxvirtualserver.org/wiki/IPVS>
- [5]. Membrey, P., D. Hows, and E. Plugge, *Practical Load Balancing*. New York City: Apress Media LLC., 2012.
- [6]. Mulyono, S. J., *Perancangan Clustering Pada Basis Data Dengan Menggunakan MySQL Cluster*, Semarang: Universitas Diponegoro, Laporan Tugas Akhir S-1, 2011.
- [7]. Oracle, *MySQL 5.5 Reference Manual*, Redwood City, California: Oracle Inc., 2012, Tersedia pada: <http://downloads.mysql.com/docs/refman-5.5-en.a4.pdf>
- [8]. Prabowo, A., *Perancangan MySQL Cluster Untuk Mengatasi Kegagalan Sistem Basis Data Pada Sisi Server*, Semarang: Universitas Diponegoro, Laporan Tugas Akhir S-1, 2010.
- [9]. Red Hat, *Red Hat Enterprise Linux AS 2.1: The Official Red Hat Enterprise Linux AS Installation Guide*, Raleigh, North Carolina: Red Hat Inc., 2002, <http://www.centos.org/docs/2/rhl-ig-as-x86-en-2.1/s1-lvs-block-diagram.html>
- [10]. Red Hat, *Red Hat Enterprise Linux 6 Load Balancer Administration*, Raleigh, North Carolina: Red Hat Inc., 2010, Tersedia pada: https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Load_Balancer_A
- [11]. Schwartz, B., and P. Zaitsev, *A Brief Introduction To Goal-Driven Performance Optimization*. Pleasanton, California: Percona Inc., 2010.
- [12]. Schwartz, B., P. Zaitsev, and V. Tkachenko, *High Performance MySQL*, Third Edition, Sebastopol, California: O'Reilly Media Inc. 2012
- [13]. Wangsmo, M., *Piranha - load-balanced web and ftp clusters*, Red Hat Inc., <http://www.redhat.com/support/wpapers/piranha>
- [14]. Zhang, W., *What Is Virtual Server?*, Linux Virtual Server Project, 2011, <http://www.linuxvirtualserver.org/whatis.html>
- [15]. Zhang, W., *How virtual server works?*, Linux Virtual Server Project, 2011, <http://www.linuxvirtualserver.org/how.html>
- [16]. Zhang, W., *General Architecture of LVS Clusters*, Linux Virtual Server Project, 2011, <http://www.linuxvirtualserver.org/architecture.html>