

IMPLEMENTASI DAN ANALISIS KINERJA LOAD BALANCING PADA VIRTUAL SERVER MENGGUNAKAN ZEN LOAD BALANCER

IMPLEMENTATION AND PERFORMANCE ANALYSIS OF LOAD BALANCING ON VIRTUAL SERVERS USING ZEN LOAD BALANCER

Radiv Herdian¹, Dr. Rendy Munadi, Ir.², Tengku A. Riza, ST., MT.³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung

¹radivherdian@students.telkomuniversity.ac.id, ²rendymunadi@telkomuniversity.ac.id,

³tengkuriza@telkomuniversity.ac.id

Abstrak

Zen Load Balancer digunakan sebagai *load balancer* untuk mengatur distribusi pengolahan data ke beberapa *server*. Skenario pertama perbandingan antara *single server* dengan menggunakan *load balancer*, skenario kedua menggunakan pembagian beban dengan bobot yang berbeda, skenario ketiga melakukan *failover* pada Zen Load Balancer. Untuk pengujian terakhir dilakukan perbandingan antara Zen Load Balancer dengan LVS. Pada penelitian ini telah diimplementasikan *load balancing* pada *virtual server*. Didapatkan nilai *throughput* sebesar 13202 Kbps, dan jumlah layanan per-detik dengan nilai 1592 *request per-second*. Terjadi penurunan 65,527 % penggunaan CPU *virtual server* pada skenario *round robin* dan terjadi penurunan 20,7124 % penggunaan CPU *real server* pada skenario *weighted*. Pada skenario *failover* didapatkan nilai rata-rata *downtime* sebesar 9622 ms. Nilai *throughput* LVS mencapai 11916 Kbps, dan LVS dapat melayani maksimal 1321 *request per-second*.

Kata Kunci : *load balancing*, zen load balancer

Abstract

Zen Load Balancer is used as a load balancer for managing distributed data processing to multiple servers. The first scenario comparison between single server and using the load balancer system, the second scenario using load sharing with different weights, third scenario perform a failover scenario in Zen Load Balancer. Fourth scenario compared Zen Load Balancer and LVS performance. In this research, load balancing has been implemented on virtual servers. Obtained value of 13202 Kbps throughput, and request per-second with a value 1592. There was a decrease 65.527% CPU usage of virtual servers on a round robin scenario and decrease 20.7124% CPU usage on real server with weighted scenarios. In a failover scenario obtained an average value of 9622 ms downtime. LVS throughput value reached 11916 Kbps, and LVS can serve a maximum of 1321 requests per second.

Keywords : *load balancing*, zen load balancer

1. Pendahuluan

Penanganan dan perawatan sebuah jaringan komputer dilingkungan suatu perusahaan ataupun instansi sering menghadapi masalah. Masalah-masalah tersebut biasanya terletak pada masalah keamanan, masalah kehandalan, ketersediaan dan skalabilitas jaringan komputer tersebut.

Berkembangnya teknologi dan meningkatnya kebutuhan akan informasi yang semakin tinggi, mempengaruhi jumlah penggunaan layanan data. Semakin banyaknya *user* yang mengakses *server*, akan memberikan beban kerja yang berat pada *server* tersebut. Suatu *single server* bisa mengalami kegagalan, hal ini dapat disebabkan oleh *client* dengan jumlah yang mencapai ribuan bahkan jutaan yang mengakses *server* tersebut dalam waktu yang bersamaan, biasa disebut dengan *overload request*.

Load balancing dapat memberikan distribusi pengolahan data secara merata ke beberapa *server* dan menciptakan ketersediaan yang tinggi pada *server*. Pada penelitian ini *server* terinstall di dalam lingkungan *virtual* menggunakan Proxmox. Zen Load Balancer digunakan sebagai *load balancer* untuk mengatur distribusi pengolahan data ke beberapa *server*.

2. Dasar Teori

2.1 Load Balancing

Load balancing adalah suatu metode yang digunakan untuk mendistribusikan beban kepada beberapa perangkat, sehingga beban kerja tidak dititikberatkan kepada satu perangkat. Tujuan utama dari penggunaan *load balancing* adalah untuk meningkatkan kinerja sistem, dan memberikan ketersediaan yang tinggi pada pengguna. *Load balancer* dapat berupa *software*, *hardware*, maupun gabungan dari keduanya^[6]. Berikut ini beberapa penjadwalan yang umum digunakan pada *load balancing*^[13], yaitu :

- *Round robin*, pembagian dilakukan secara *cyclic* dan merata
- *Weighted*, pembagian beban dilakukan sesuai dengan bobot yang diberikan kepada perangkat
- *Least connection*, aliran paket data dilewatkan pada jumlah koneksi yang paling kecil

3. Perancangan dan Implementasi Sistem

3.1 Deskripsi Sistem

Sistem ini akan bekerja sebagai sistem yang akan melakukan *load balancing* pada *client* yang memberikan *request* kepada *server*. Zen Load Balancer digunakan sebagai media *load balancing* dan Proxmox sebagai pembuat lingkungan virtual di sisi *server*. Pada sistem ini akan diuji sejauh mana peningkatan kinerja *server* saat dilakukan tanpa *load balancing*, *load balancing* menggunakan penjadwalan *round robin*, penjadwalan *weighted*, dan pengukuran *downtime* saat terjadi kegagalan pada Zen Load Balancer. Serta perbandingan kinerja antara Zen Load Balancer dengan LVS

3.2 Kebutuhan Infrastruktur

Kebutuhan untuk implementasi sistem ini akan dikelompokkan menjadi 2 bagian, yaitu kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*).

3.2.1 Perangkat Keras

Berikut ini merupakan perangkat-perangkat keras yang digunakan dalam tugas akhir, yaitu :

1. Sebuah *client*, dengan spesifikasi sebagai berikut :
 - a. Intel Core i3, 2.3 GHz
 - b. 2 GB RAM
 - c. Sebuah *Ethernet Card*
 - d. 250 GB *Harddisk Drive*
2. Dua buah komputer sebagai *hypervisor*, dengan spesifikasi sebagai berikut :
 - a. Intel Core i3, 3.3 GHz
 - b. 2 GB RAM
 - c. Sebuah *Ethernet Card*
 - d. 500 GB *Harddisk Drive*
3. Sebuah komputer *server*, dengan spesifikasi sebagai berikut :
 - a. Intel Core 2-Duo, 2.4 GHz
 - b. 1 GB RAM
 - c. Sebuah *Ethernet Card*
 - d. 250 GB *Harddisk Drive*
4. Dua buah komputer *load balancer*, dengan spesifikasi sebagai berikut :
 - a. Intel Core 2-Duo, 2.4 GHz
 - b. 1 GB RAM
 - c. Dua buah *Ethernet Card*
 - d. 250 GB *Harddisk Drive*
5. Dua buah switch 5-port
6. Kabel UTP secukupnya untuk menghubungkan perangkat sesuai topologi yang digunakan pada setiap scenario pengujian

3.2.2 Perangkat Lunak

Berikut ini merupakan perangkat-perangkat lunak yang digunakan dalam tugas akhir, yaitu :

1. Ubuntu 13.04 sebagai sistem operasi pada *client*
2. Proxmox 2.2 sebagai *hypervisor*
3. Ubuntu 13.04 *server* sebagai sistem operasi pada *virtual server*
4. Apache2, perangkat lunak ini akan diinstalasi pada *virtual server* yang berfungsi sebagai *web server*.
5. Siege, perangkat lunak yang digunakan untuk memberikan *request* dengan jumlah yang besar pada *web server* dan memberikan laporan kinerja *server*
6. Hping, perangkat lunak yang digunakan untuk mengukur waktu *downtime* pada *failover* Zen Load Balancer
7. Htop, perangkat lunak yang digunakan untuk melihat kinerja suatu *server*

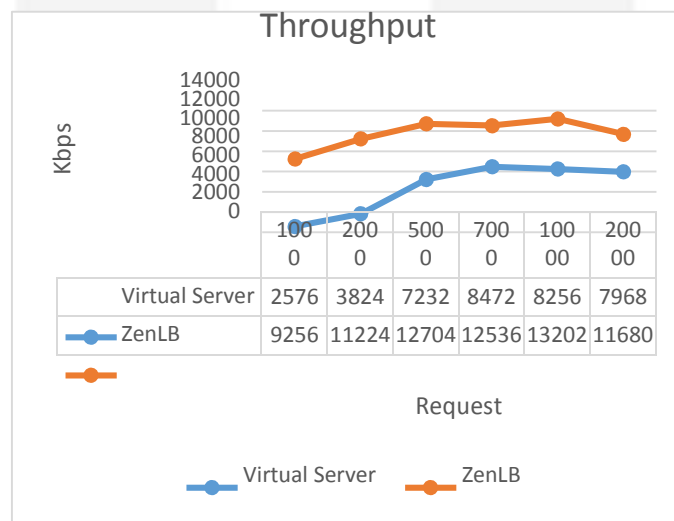
3.3 Skenario Pengujian

Parameter yang diukur pada tugas akhir ini adalah jumlah layanan per-detik, waktu respon, *throughput*, *CPU utilization*, dan *request loss*. Pada pengujian tugas akhir ini, tiap *user* mengakses berkas sebesar 1118 bytes. Berikut skenario yang telah dilakukan

- a. Skenario *round robin*, pada skenario ini diukur kinerja pada empat *virtual server*
- b. Skenario *failover*, pada skenario ini diukur nilai *downtime* yang terjadi pada Zen Load Balancer
- c. Skenario *weighted*, diukur kinerja CPU pada *virtual* dan *real server*
- d. Perbandingan kinerja Zen Load Balancer dengan LVS

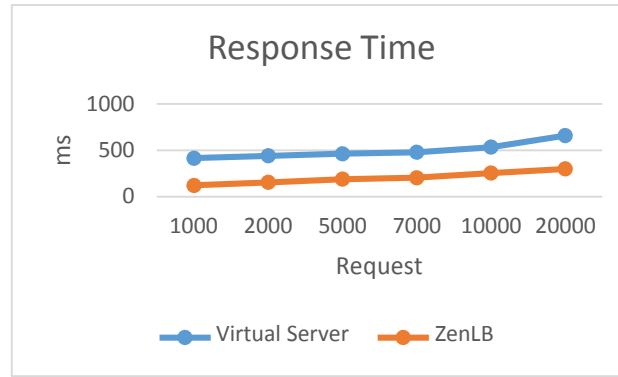
4. Pengukuran dan Analisis Kinerja Sistem

4.1 Perbandingan *Single Server* dengan Sistem Zen Load Balancer



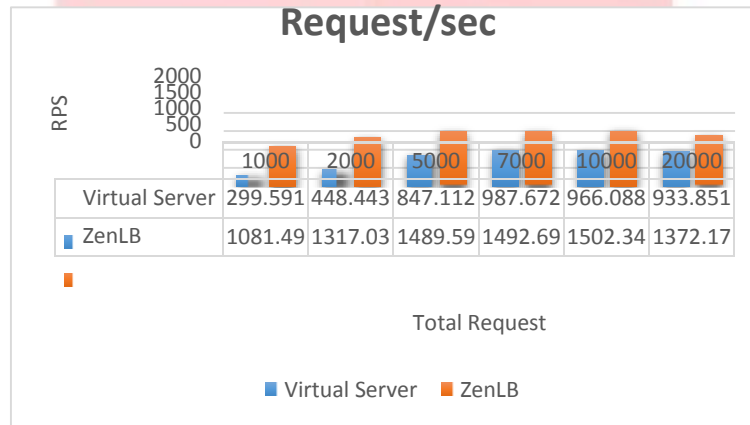
Gambar 1. Perbandingan *Throughput Single Server* dengan Zen Load Balancer

Nilai *throughput* lebih besar saat dilakukan *load balancing* menggunakan Zen Load Balancer dibandingkan dengan nilai *throughput* pada *single server*. Hal ini dikarenakan *request* yang datang dikerjakan lebih dari satu *server*, sehingga dapat mengurangi *bottleneck* disisi *server*. Didapatkan nilai maksimal *throughput* sebesar 14090 Kbps setelah penggunaan *load balancing*.



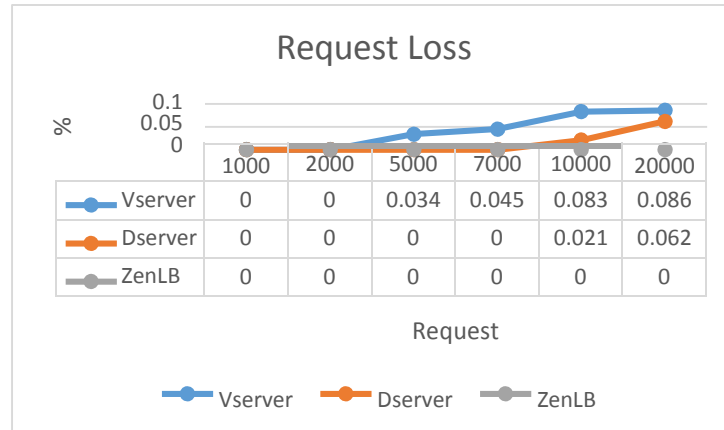
Gambar 2. Perbandingan Waktu Respon Single Server dengan Zen Load Balancer

Nilai waktu respon saat digunakan *load balancing* lebih kecil dibandingkan *single server* yang bekerja. Dengan adanya *load balancing*, *request* yang datang tidak dititikberatkan pada satu *server*, hal ini dapat menghindari adanya antrian yang datang pada *server* sehingga memberikan nilai yang lebih kecil pada waktu respon. Didapatkan nilai waktu respon sebesar 186 ms setelah penggunaan *load balancing*.



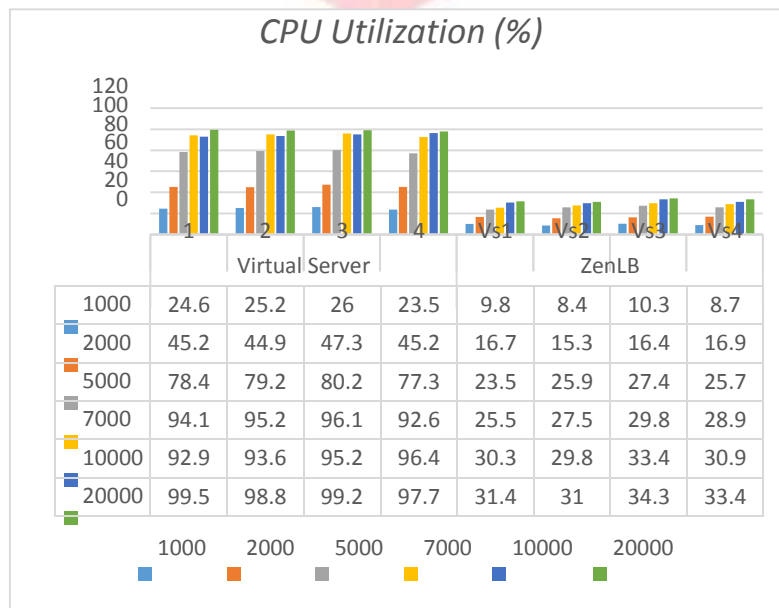
Gambar 3. Perbandingan Request Per-detik Single Server dengan Zen Round Robin

Jumlah *request* per-detik pada *single server* jauh lebih rendah dibandingkan dengan sistem Zen Load Balancer. Hal ini dikarenakan pada Zen Load Balancer menggunakan *virtual server* yang saling bekerja sama dalam melayani *request* yang datang, sehingga jumlah *request* yang dapat dilayani meningkat. Didapatkan nilai maksimal jumlah *request* per-detik sebesar 1592.314 *request* per-detik.



Gambar 4. Perbandingan Request Loss Single Server dengan Zen Round Robin

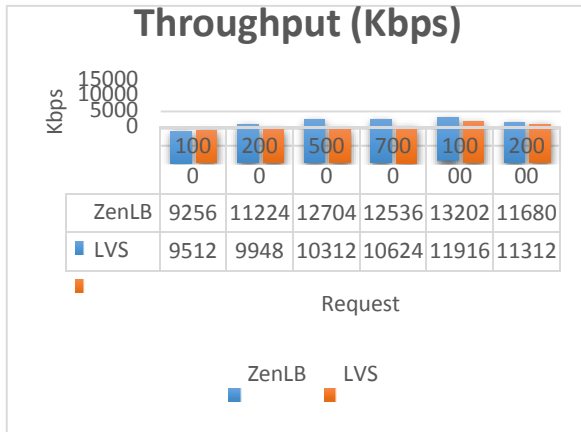
Terlihat bahwa pada *single server* memiliki nilai *request loss*, dikarenakan *server* tidak mampu melayani permintaan saat jumlah permintaan terus meningkat. Pada sistem Zen Load Balancer tidak terjadi *request loss*, ini membuktikan sistem bekerja lebih baik dalam melayani permintaan *user*. Jumlah permintaan tidak dibebankan kepada satu *server* sehingga tidak ada permintaan yang hilang.



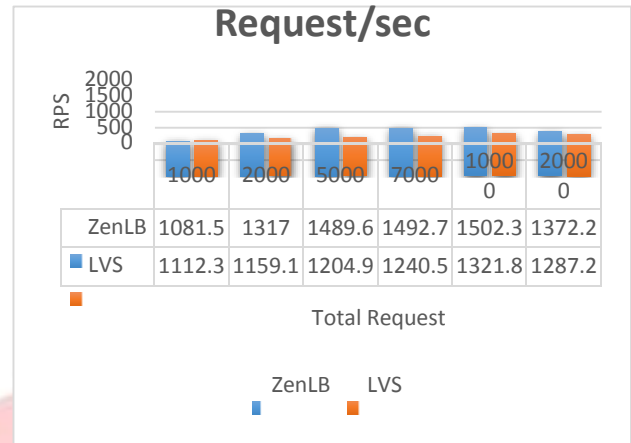
Gambar 5. Perbandingan CPU Single Server dengan Zen Round Robin

Beban kerja pada sistem Zen Load Balancer lebih kecil dibandingkan pada *single server*. Pada Zen Load Balancer *round robin* pembebanan dibagi kedalam empat buah *virtual server* dengan perbandingan yang sama. Terjadi penurunan 65,527 % pada pengujian menggunakan *round robin*.

4.2 Perbandingan Zen Load Balancer dengan LVS



Gambar 5. Perbandingan Throughput Zen dengan LVS



Gambar 6. Perbandingan Jumlah Request per-detik Zen dengan LVS

LVS merupakan *load balancer* yang bekerja pada *layer 4*^[13], sedangkan Zen Load Balancer merupakan *load balancer* yang bekerja pada *layer 7*, khususnya pada layanan *http*^[12]. Hal ini mempengaruhi proses kerja pada *load balancer*. LVS akan mengidentifikasi paket yang datang berdasarkan TCP dan *port* yang digunakan, sedangkan Zen Load Balancer akan langsung mengidentifikasi dari jenis layanan yang digunakan, sehingga akan mempengaruhi waktu kerja pada *load balancer*. Nilai *request* per-detik memberikan informasi bahwa Zen Load Balancer memiliki nilai *concurrent connection* yang lebih besar dari LVS. *Concurrent connection* merupakan jumlah *request* yang dapat dilayani oleh suatu *server* secara bersamaan dalam waktu yang sama. Semakin meningkatnya nilai *request* per-detik akan mempengaruhi nilai dari *throughput*.

4.3 Skenario Failover

Pengukuran dilakukan menggunakan Hping, dimana pengiriman paket diberikan jeda sebesar 5 ms. Didapatkan nilai *downtime* sebesar 9622 ms, nilai tersebut didapatkan dari nilai rata-rata tiga puluh kali hasil pengamatan. Semua konfigurasi *load balancing* dan *clustering server* terdapat pada *master*, Zen Load Balancer *slave* hanya memiliki konfigurasi *IP Address* dan konfigurasi sebagai *slave* untuk Zen Load Balancer *master*. Saat adanya kegagalan pada *master*, Zen Load Balancer *slave* akan langsung melakukan penyalinan semua proses dan konfigurasi yang sedang berlangsung pada Zen Load Balancer *master*.

5. Penutup

5.1 Kesimpulan

- 1 Zen Load Balancer dapat melayani hingga 1592 *request* per-detik, dan *single server* dapat melayani 987 *request* per-detik
- 2 Sistem Zen Load Balancer dapat melakukan *failover*, dengan *downtime* 9622 ms. Hal ini disebabkan adanya penyalinan proses dan konfigurasi dari *master* ke *slave*.
- 3 Waktu respon sistem Zen Load Balancer lebih baik dengan nilai 124 ms, hal ini dapat mengurangi adanya antrian *request* sehingga dapat menaikkan nilai *throughput* dan *request* per-detik
- 4 Sistem Zen Load Balancer dapat meminimalisir jumlah *request loss*. Dimana dalam pengujian ini nilai Zen Load Balancer adalah 0, untuk *request loss*
- 5 Nilai *throughput* pada Zen Load Balancer lebih besar dari LVS, dimana Zen Load Balancer memiliki nilai 13202 Kbps dan LVS memiliki nilai 11916 Kbps
- 6 Nilai *request* per-detik dari Zen Load Balancer lebih besar dari LVS, dimana Zen Load Balancer memiliki nilai 1502 *request* per-detik, dan LVS memiliki nilai 1312 Kbps. Nilai ini menunjukkan bahwa jumlah *concurrent connection* pada Zen Load Balancer lebih besar dari LVS

- 7 Batas kerja maksimal dari kedua *load balancer* mencapai 10000 *request*, dimana nilai akan cenderung stabil setelah melewati 10000 *request*

5.2 Saran

- 1 Perlu dilakukan implementasi dan konfigurasi lebih lanjut pada IPv6
- 2 Penelitian dilakukan dengan menggunakan gabungan antara *hardware load balancer* dengan *software load balancer*
- 3 Menggunakan konfigurasi jaringan VLAN pada sistem *load balancer*
- 4 Menggunakan *shared storage* sebagai tempat penyimpanan berkas-berkas *server*

6. Daftar Pustaka

- [1]. NI Real-Time Hypervisor Architecture and Performance Details Whitepaper (Dec 10, 2014).
- [2]. (2015, January 18). Retrieved from <http://cellstreamblog.blogspot.com/2011/10/defining-virtualization-what-is-it.html>.
- [3]. Aversa, L., & Bestavros, A. (n.d.). *Load Balancing A Cluster Web Servers Using Distributed Packet Rewriting*. Computer Science Department Boston University.
- [4]. Elsayed, E. A. (2012). *Reliability Engineering, 2nd Engineering*. Willey.
- [5]. Forouzan, B. A. (2006). *Data Communications & Networking fourth edition*.
- [6]. Kahanwal, B., & Singh, T. P. (2012). The Distributed Computing Paradigms : P2P, Grid, Cluster, Cloud, and Jungle. *International Journal of Latest Research in Science and Technology*, 183 - 187.
- [7]. Kaur, K., & Rai, A. K. (2014). A Comparative Analysis : Grid, Cluster and Cloud Computing. *International Journal of Advanced Research in Computer and Communication Engineering*.
- [8]. Menasce, D. A. (n.d.). *Virtualization : Concepts, Applications, and Performance Modeling. Virtualization : Concepts, Applications, and Performance Modeling*.
- [9]. Sofana, I. (2012). *Cloud Computing Teori dan Praktik (OpenNebula, VMware, dan Amazon AWS)*. Informatika Bandung.
- [10]. Stallings, W. (2000). *Data & Computer Communication sixth edition*. Prentice Halls.
- [11]. Teo, Y. M., & Ayani, R. (2001). Comparison of Load Balancing Strategies on Cluster-based Web Servers.
- [12]. *Zen Load Balancer*. (2015, January). Retrieved from www.zenloadbalancer.com.
- [13]. *Linux Virtual Server Project*. (2015, January). Retrieved from www.linuxvirtualserver.org