

IMPLEMENTASI HIGH AVAILABILITY SERVER MENGGUNAKAN PLATFORM HAPROXY (STUDI KASUS: APLIKASI ZAMMAD UNTUK ONLINE HELP DESK)

IMPLEMENTATION OF HIGH AVAILABILITY SERVER USING THE HAPROXY PLATFORM (CASE STUDY: ZAMMAD APPLICATION FOR ONLINE HELPDESK)

Umar Ali Ahmad¹, Randy Erfa Saputra², Rahmat Muda Harahap³

^{1,2,3} Universitas Telkom, Bandung

¹umarali@telkomuniversity.ac.id, ²randyerfa@telkomuniversity.ac.id,

³cintawaniat@students.telkomuniversity.ac.id

Abstrak

Cloud Computing berkisar pada akuisisi berbasis internet dan pelepasan sumber daya dari pusat data. Menjadi komputasi dinamis berbasis internet, *cloud computing* juga dapat mengalami kelebihan permintaan. Penyeimbangan muatan adalah aspek penting yang berkaitan dengan distribusi sumber daya sedemikian rupa sehingga tidak ada kelebihan beban yang terjadi di setiap mesin dan sumber daya yang dimanfaatkan secara optimal.

Load balancing merupakan salah satu mekanisme untuk membagi beban pada *cloud computing* ke beberapa server. *Haproxy* adalah produk opensource yang mendukung keperluan *Load balancing* dan failover web server. *Haproxy* sangat diperlukan apabila aplikasi menuntut SLA yang cukup ketat dan tidak mentolerir adanya downtime. Selain itu *Haproxy* memungkinkan adanya backup dari load balancer yang digunakan sehingga apabila terjadi down pada load balancer utama maka load balancer yang menjadi backup dalam waktu singkat secara otomatis akan menggantikan load balancer yang mati tersebut.

Algoritma Round Robin merupakan salah satu algoritma penjadwalan proses yang digunakan secara luas didalam penjadwalan CPU. Algoritma Round Robin menggunakan sistem time sharing dengan static quantum time untuk setiap proses yang akan dieksekusi CPU. Algoritma ini tergantung pada ukuran quantum time yang diberikan.

Kata Kunci : Cloud Computing, Load balancing, Haproxy, Round Robin

Abstract

The concept of cloud computing revolves around the acquisition and release of resources from data centers via the internet. Cloud computing, as an internet-based dynamic computing system, can experience surges in demand. Load balancing is an important aspect of resource distribution that ensures that no machine is overloaded and that resources are used to their full potential.

Load balancing is a technique for distributing cloud computing workload over many servers. Haproxy is an open source load balancing and failover web server that supports load balancing and failover. If the application has a reasonably strict SLA and does not tolerate downtime, Haproxy is a must-have. Furthermore, Haproxy provides for backups of the load balancer in use, so that if the main load balancer goes down, the load balancer that becomes a backup in a short period of time will instantly replace the dead load balancer.

The Round Robin algorithm is a widely used process scheduling mechanism in CPU scheduling. For each task to be executed by the CPU, the Round Robin algorithm uses a time sharing system with static quantum time. The size of the specified quantum time determines the algorithm.

Keywords: Cloud Computing, Load balancing, Haproxy, Round Robin

1. Pendahuluan

Penerapan teknologi informasi dalam suatu perusahaan ataupun organisasi merupakan hal yang sangat dibutuhkan untuk saat ini. Pada tahun 2017 penggunaan internet di Indonesia menduduki peringkat ke-6 di dunia. Menurut Asosiasi Penyelenggara Internet Indonesia (APJII) dengan 143,26 juta pengguna internet atau setara dengan 54,68% dari total jumlah penduduk Indonesia. Dari tingginya jumlah penggunaan internet ini tentunya membuka peluang bisnis yang dapat berhubungan dengan teknologi informasi. Saat ini sebuah perusahaan atau organisasi yang menggunakan teknologi yang mengaplikasikan layanan sistem informasi baik itu digunakan untuk kepentingan internal perusahaan atau organisasi tersebut atau untuk hubungan dengan eksternal perusahaan atau organisasi pasti memiliki server baik yang memiliki secara fisik atau pun dengan menyewa server cloud.

Kebutuhan dan kemampuan dari server yang dimiliki baik itu server fisik maupun dengan menyewa cloud computing tentu juga berpengaruh dalam usaha perusahaan. Berbagai cara digunakan untuk mengoptimalkan server yang dimiliki. Saat ini sebuah perusahaan atau organisasi yang menggunakan teknologi yang mengaplikasikan layanan sistem informasi baik itu digunakan untuk kepentingan internal perusahaan atau organisasi tersebut atau untuk hubungan dengan eksternal perusahaan atau organisasi pasti memiliki server baik yang memiliki secara fisik atau pun dengan menyewa server cloud. Kebutuhan dan kemampuan dari server yang dimiliki baik itu server fisik maupun dengan menyewa cloud computing tentu juga berpengaruh dalam usaha perusahaan. Berbagai cara digunakan untuk mengoptimalkan server yang dimiliki. Berbagai cara dilakukan agar pelayanan yang dimiliki oleh sistem yang dimiliki berjalan dengan optimal. Sistem yang handal tentunya akan dapat menangani masalah yang ada. Sebagai contoh saat sistem berjalan, meningkatnya request / permintaan pengaksesan menyebabkan server tidak dapat menangani permintaan dari pengguna sehingga menyebabkan overload.

2. Cloud Computing

Cloud computing adalah gabungan pemanfaatan teknologi komputer dan pengembangan berbasis

Internet. Suatu cara atau metode yang memudahkan pengguna untuk mengakses informasi tanpa mengetahui apa yang ada didalamnya, ahli dengannya, atau memiliki kendali terhadap infrastruktur teknologi yang membantunya. Cloud computing juga dapat diartikan sebagai sebuah model yang memungkinkan pengguna untuk dapat mengakses sumber daya seperti processor, storage (media penyimpan), network, software (perangkat lunak) secara abstrak dan diberikan sebagai layanan di jaringan/internet[1].

2.1 Hypervisor

Hypervisor (disebut juga *Virtual Machine Monitor*) adalah platform atau aplikasi untuk menjalankan teknik virtualisasi, yang dapat menjalankan beberapa *guest OS* di dalam *host OS*. Secara sederhana, proses virtualisasi dilakukan oleh firmware ini mulai dari berbagi resource yang dimiliki oleh *si host*, hingga mengelola akses antara *hardware* dengan sistem operasi yang berjalan di atasnya.

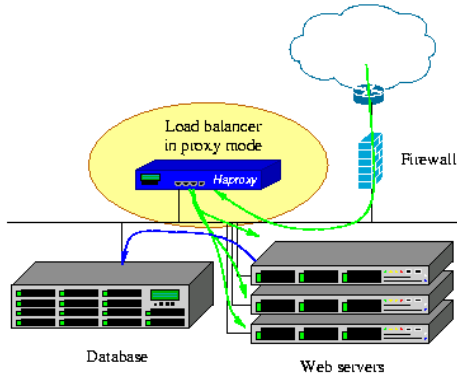
2.2 Zammad

Zammad didirikan oleh Martin Edenhofer, yang sebelumnya terlibat dalam pengembangan OTRS. Proyek ini meminta partisipasi aktif dalam pembangunan.^[4] Kode sumber bersifat open-source menurut Lisensi Publik Umum GNU AFFERO (GNU AGPLv3) dan tersedia melalui git. Untuk tujuan ini, Yayasan Zammad didirikan untuk memastikan kebebasan perangkat lunak. Inspirasi untuk Zammad Foundation adalah WordPress Foundation, Free Software Foundation, dan Mozilla Foundation.

2.3 Haproxy

High Availability Proxy adalah kepanjangan dari HAProxy sebuah perangkat lunak open source dibawah GPLv2 license. HAProxy digunakan untuk membagi beban request atau load balancer TCP/HTTP dan solusi proxy yang dapat dijalankan di sistem operasi Linux, Solaris, dan FreeBSD.

Pembagian bebannya pun beragam sesuai dengan algoritma yang ada. HAProxy sudah umum digunakan untuk meningkatkan kinerja dan kehandalan sebuah server dengan mendistribusikan beban kerja dari beberapa server lain seperti web server, database server, smtp server dan yang lainnya.



2.4 Round Robin

Algoritma Round Robin merupakan salah satu algoritma penjadwalan proses yang digunakan secara luas didalam penjadwalan CPU. Algoritma Round Robin menggunakan sistem time sharing dengan static quantum time untuk setiap proses yang akan dieksekusi CPU. waktu yang sama dari tiap-tiap prosesnya dan telah diatur dalam task scheduler [11]. Round robin dibuat sedemikian rupa untuk memenuhi kebutuhan time-sharing, time sharing berguna untuk berguna untuk meningkatkan waktu respon, dalam proses pembaruan. Waiting time pada Round robin tidak optimal, namun waktu respon pembaruan rata-rata lebih baik [12].

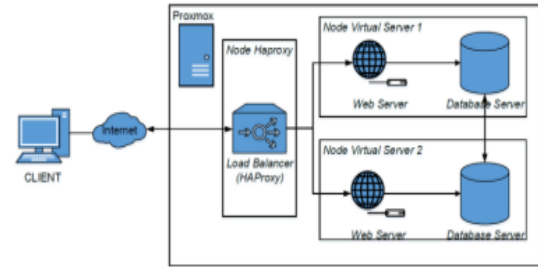
2.5 Availability

Berdasarkan dokumen ISO 2382-14, 1997 availability dapat didefinisikan sebagai: Kemampuan sebuah alat untuk berada dalam kondisi siap pakai sesuai fungsi yang diinginkan pada waktu tertentu atau kapanpun dalam interval waktu tertentu, diasumsikan bahwa sumber eksternalnya bila diperlukan, adalah tersedia. Secara garis besar availability merupakan nilai persentase jumlah waktu suatu jaringan mampu memberikan layanan dibandingkan dengan jumlah waktu yang diharapkan. Sedangkan rata-rata waktu suatu sistem atau jaringan dalam

kondisi down atau tidak mampu memberikan layanan disebut downtime. Sedangkan uptime didefinisikan sebagai waktu rata-rata suatu sistem atau jaringan dalam keadaan operasional[14]. Nilai Availability dapat dihitung dengan menggunakan persamaan 2.1 [15].

$$Availability = \frac{up\ time}{(uptime + downtime)} \times 100\%$$

2.6 Implementasi Sistem



Gambar 4, merupakan gambaran dari desain ataupun tologi jaringan yang akan dibuat dalam penelitian ini, *request* dari *client*. Sistem ini menggunakan haproxy sebagai load balancing. Agar meningkatkan layanan pada sistem, dilakukan konfigurasi failover pada sisi haproxy dengan mode aktif/pasif dengan menggunakan keepalived dan kemudian dilihat downtime saat terjadi kegagalan pada server. Dan sebagai pembuat lingkungan virtual serta sistem clustering disisi web server.

Table 1 IP Konfigurasi

Perangkat	Interface
Client	192.168.8.11/24
Haproxy	192.168.9.11/25
Server	192.168.137.117:8006

Tabel 1, merupakan pengalaman pada perangkat yang kita gunakan untuk melaukan pengujian, dengan menggunakan pengalaman ini untuk memper mudah komunikasi antar perangkat. Dengan digunakannya pengalaman ini penulis juga

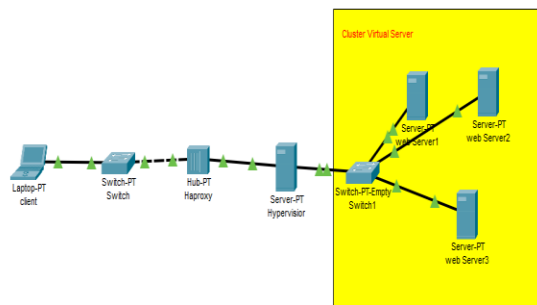
bisa dengan mudah mengetahui perangkat apa saja yang digunakan dan tujuannya.

3. Hasil dan Pembahasan [10 pts/Bold]

Pengujian dengan menggunakan monitoring yang didasari dengan pengecekan detail perancangan, dan analisis terhadap hasil uji untuk melihat kinerja dari sistem yang telah dibangun sebelumnya dengan instalisasi dan konfigurasi yang sudah berhasil dilakukan.

3.1 Pengujian Haproxy

Pada pengujian ini dilakukan menggunakan 3 buah virtual web server yang telah dikonfigurasi sebelumnya, beban yang

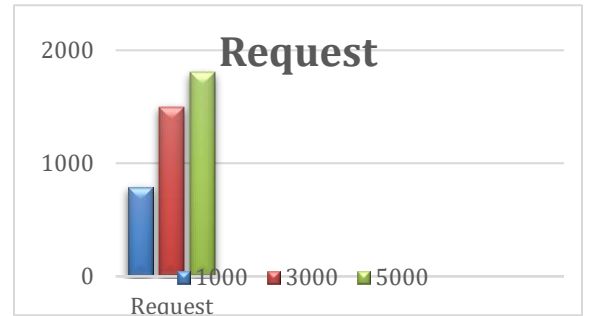


diberikan sama rata disetiap server oleh haproxy. Berikut rata – rata dari seppuluh kali pengamatan data yang dilakukan pada haproxy algoritma round robin.

User	Request/sec	Thougput(MBps)	CPU Utilization (%)		
			WS 1	WS 2	WS 3
1000	786.14	0.786	8.78	7.98	8.50
3000	1498.26	1.988	21.08	22.19	20.67
5000	1804.22	2.498	35.43	35.6	38.02

3.2 Pengujian Request

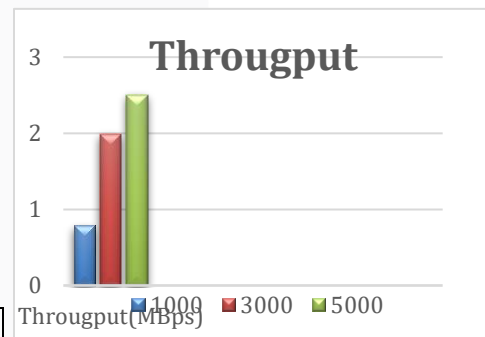
Jumlah request yang telah diamati dituangkan kedalam chart. Pengukuran ini dilakukan untuk mengetahui peningkatan request dapat dilayani oleh server. Untuk hasil pengukuran dapat diamati dan dituangkan kedalam chart dibawah ini:



Dapat kita lihat request yang diberikan saat 1000, 3000, 5000 request menghasilkan data memiliki nilai yang tidak terlalu jauh perbedaannya, ini dikarnakan semakin banyaknya web server yang dibuat semakin bagus nilai yang akan didapatkan didalam monitoring.

3.3 Pengujian Throughtput

Pengukuran ini sangat penting sekali dilakukan untuk mengetahui kemampuan sistem dalam memberikan layanan kepada client terhadap permintaan layanan yang dating secara bersamaan.

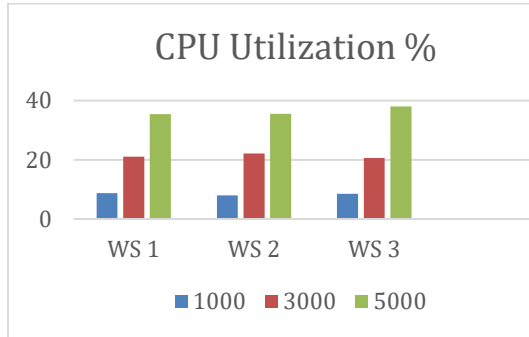


Pada gambar diatas menunjukkan keseluruhan nilai maksimal thorugput sebesar 2,4 MB/s saat server menangani 5000 request, ini disebabkan terdapat lebih dari satu server yang melayani permintaan.

3.4 Pengujian CPU Utilization

Pengujian ini dilakukan diserver menggunakan htop. Nilai CPU pada server menampilkan sejauh mana server dapat memproses permintaan dari client. Instruksi per siklus (insns per siklus: IPC), yang menunjukkan rata-rata berapa banyak instruksi yang kami selesaikan untuk setiap siklus clock CPU. Semakin tinggi, semakin

baik. Berikut hasil pengamatan dari CPU dengan menggunakan Haproxy:



Dengan meningkatnya permintaan client makan akan berdampak dengan meningkatnya nilai dari CPU. Namun dengan menggunakan Haproxy maka akan dibagi rata beban yang diberikan oleh client ke dalam tiga web server yang telah dibuat.

3.5 Pengujian Failover

Pada skema pengujian ini akan dilakukan failover kepada haproxy yang telah dikonfigurasi dengan mode aktif/pasif, dengan menggunakan keepalived. Pengujian ini akan diketahui berapa lama downtime yang terjadi saat haproxy mengalami down. Pada skema ini dilakukan pengukuran menggunakan hping dengan mengirim paket ICMP dimana diberikan delay 5ms untuk setiap pake yang dikirim. Dimana didapatkan nilai rata – rata pengamat sebesar 1956.7 ms. Niali tersebut terbilang baik dibandingkan jika tidak ada failover pada sistem. Waktu yang dibutuhkan sebuah komputer untuk rebbot bisa mencapai 300 detik. Berikut perhitungan tingkat ketersediaan server dalam jangka waktu satu tahun menurut sekala the nines, dengan persamaan 2:1.

$$Availability = \frac{up\ time}{(uptime + downtime)} \times 100\%$$

Nilai rata- rata downtime = 1956.7 ms

Up time = 31535272.628 detik/tahun

$$Availability = \frac{31535272.628}{(31535272.628 + 1956.7)} \times 100\% = 99,9938 \%$$

Menurut skala The Nines, nilai availability pada level tujuh dimana nilai downtime kurang dari 3,2 detik/tahun.

4. Kesimpulan

Dari hasil implementasi dan pengujian yang dilakukan dapat diambil kesimpulan, sebagai berikut:

- a. Kemampuan haproxy dengan semakin banyaknya web server akan dapat melayani request dengan baik, dikarnakan web server saling berbagia rata bebas yang diberikan oleh client, dan dengan hal ini bisa menurunkan CPU utilization.
- b. Dalam meningkatkan ketersediaan server, haproxy berhasil melakukan failover dengan nilai rata – rata downtime sebesar 1956.7 ms.

5. Referensi

- [1] PURBO, O. W. (2011). Petunjuk Praktis Cloud Computing Menggunakan Open Source, 1–48.
- [2] David Marshall, Wade A. Reynolds, and Dave McCrory, Advance Server Virtualization, Auerbach Publications, 2006
- [3] H. J. Younis. 2015. Efficient Load Balancing Algorithm in Cloud Computing
- [4] Kenneth Majors and friends, The Best Damn Server Virtualization, Syngress Publishing, Inc., Elsevier, Inc., 2007
- [5] Obasuyi, G. C., Sari A., “Security Challenges of Virtualization Hypervisors in Virtualized Hardware Environment”. Int. J. Communications, Network and System Sciences, 2015, 8, 260-273.
- [6] Asterina, D., “IMPLEMENTASI DAN ANALISIS METODE FAILOVER PADA SISTEM REDUNDANT DEDICATED SERVER DAN CLOUD SERVER UNTUK LAYANAN VOIP”. Fakultas Teknik Elektro, Bandung, 2015.
- [7] Mawarnblum, M. adan Garnkel ,T. (2005) Virtual Majurang monitor: Current Technologika dan Future Trends. Komprahim, 38 , 39 - 47
- [8] Zhao ,X. , Batas ,K dan Prakash ,A . (2009) Virtual Mkeamanan keamanany System. Rayuan di Computer Sainssekali dan Mesinering , 1 , 339 - 365.

- [9] Rakesh kumar katare and mamta kumari " A comparative study of various load balancing algorithm in parallel and distributed multiprocessor system" International Journal of Computer Applications (0975 – 8887) Volume 169 – No.10, July 2017
- [10] Zubair khan, Ravidra singh, Jahagir alam , Shailesh Saxena "classification of load balancing condition for parrel and distributed system" IJCSI vol 8 Issue 5 september 2011
- [11] K. Elissa, "Title of paper if known," unpublished.
- [12] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
- [13] Nischol Mishra and Nitin kumar mishra " Load Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing- A Systematic Review " International Journal of Computer Applications (0975 – 8887) Volume 131 – No.18, December 2015
- [14] P. Yogie Ariyanto, Soetam Rizky Wicaksono, Jurusan Sistem Informasi, Fakultas Sains, Universitas Ma Chung, Jl Villa, and Puncak Tidar. 2017. "Pengemudi Dengan Menggunakan Algoritma Round Robin (Studi Kasus : Zena Travel)." Sanis Dan Teknologi 6 (1): 189–98. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
- Diarjo, Andi Anto, and Dadang Iskandar Mulyana. 2017. "Penerapan Algoritma Round Robin Dan Modulo Pada Load Balancing." Penerapan Algoritma Round Robin Dan Modulo Pada Load Balancing Web Server 10 (1): 21–34.