

IMPLEMENTASI LOAD BALANCING DENGAN MENGUNAKAN ALGORITMA ROUND ROBIN PADA KASUS PENDAFTARAN SISWA BARU SEKOLAH MENENGAH PERTAMA LABSCHOOL UNESA SURABAYA

Gaguk Triono
Teknologi Informasi
Sekolah Tinggi Teknik Surabaya
gagukt@gmail.com

ABSTRAK

Saat ini internet berkembang dengan sangat pesat, seiring dengan semakin banyaknya user yang terhubung pada jaringan internet. Ketika server tunggal mendapatkan request dari banyak user, besar kemungkinan akan terjadi overload dan crash sehingga request dari banyak user tidak dapat dilayani oleh server tunggal. Arsitektur cluster yang diterapkan sebagai server dengan performa tinggi adalah salah satu solusi yang efektif dan efisien untuk mengatasi masalah tersebut. Arsitektur cluster ini dapat dibangun dengan menggunakan konsep network load balancing dengan high-availability yang memungkinkan proses pengolahan data dibagi secara terdistribusi ke beberapa komputer, salah satu caranya menggunakan teknologi Linux virtual server. Pada Linux virtual server terdapat algoritma penjadwalan yang dapat mempengaruhi sistem kerja LVS, salah satunya algoritma round robin. Dengan menggunakan algoritma round robin maka beban kerja akan dibagi secara merata ke seluruh server. Pengujian algoritma round robin dilakukan dengan menggunakan parameter perbandingan jumlah user dan waktu respon, sehingga didapatkan waktu tanggap yang stabil antara 31 ms sampai 76,3 ms dengan jumlah user antara 10.000 sampai 50000 pada implementasi load balancing menggunakan algoritma round robin.

Kata kunci: *Load balancing, Linux Virtual Server, Algoritma Round Robin.*

ABSTRACT

Currently the Internet is growing very rapidly, as more and more users are connected to the Internet network. When a single server to get requests from many users are likely to overload and crash so that requests from many users will not be served by a single server.

Cluster architecture implemented as a server with high performance is one of effective and efficient solutions resolve the issue. The architecture of this cluster can be built using the concept of network load balancing with high-availability that enables the distributed data processing is divided into several computers, one way to use Linux virtual server technology.

On virtual Linux server there is a scheduling algorithm that can affect the LVS working system, one round robin algorithm. With the round robin algorithm to use your workload will be divided evenly throughput the server. Round robin testing of the

algorithm is done using the parameters in the number of users and response time, so we get the best results on the implementation of load balancing use round robin algorithm.

Keywords: Load balancing, Linux Virtual Server, Round Robin Algorithm

I. PENDAHULUAN

Banyak cara yang dilakukan sebuah instansi pendidikan untuk dapat melakukan proses penerimaan siswa baru, salah satu nya adalah penerimaan siswa baru secara online. Hal ini sangat membantu sekali dalam memperlancar registrasi atau penerimaan siswa baru sekolah tertentu.

Pengembangan teknologi komputasi berbasis internet sekarang ini lebih diarahkan kepada proses pengaplikasian sistem yang mudah dan tidak memerlukan banyak waktu atau tenaga. Dengan konsep tersebut, maka semakin banyak siswa yang bisa memiliki akses dan memanfaatkan sumber daya tersebut, karena tidak harus melakukan registrasi atau pendaftaran siswa baru secara manual.

Bila semakin banyak siswa yang mengakses, maka semakin padat pula traffic internet nya. Hal ini akan berpengaruh secara langsung pada web server yang menyediakan layanan registrasi siswa baru secara online, dengan kata lain beban sebuah server akan semakin berat, apalagi bila menggunakan server tunggal.

Salah satu cara yang dapat digunakan untuk menyelesaikan masalah diatas adalah dengan cara membagi beban kerja seara bergantian. Hal ini dapat dilakukan dengan menambahkan jumlah server yang menyediakan layanan registrasi atau pendaftaran siswa baru secara online.

Penerapan algoritma round robin pada komputer load balancer sangat dibutuhkan, karena dengan menggunakan algoritma tersebut setiap request dari client akan diarahkan kebeberapa real server, sehingga komputer real server akan menerima beban secara merata. Dengan adanya server tambahan dan komputer load balancer yang menggunakan algoritma round robin maka secara tidak langsung beban server akan berkurang dan hal ini pula akan meminimalisir terjadinya overload. Dengan demikian akses pada registrasi atau pendaftaran online akan semakin lebih efisien.

II. RANCANGAN SISTEM

Dalam perancangan pembangunan sistem implementasi Load Balancing dengan menggunakan Algoritma Round Robin, penulis ingin menerapkan penggunaan load balancer. Dimana setiap client akan ditangani oleh beberapa server. Dengan adanya load balancer maka beban akan diberikan kepada server yang berbeda dan secara otomatis maka beban tiap server akan berkurang.

Selain itu dengan load balancer kita dapat mengarahkan client ke web server yang lain jika terjadi web server overload atau web server sedang down, hal ini sangat bermanfaat, baik dari sisi kecepatan akses maupun efisiensi waktu pada saat user mengakses sebuah web.

Untuk menganalisis algoritma scheduling di linux virtual server (LVS), dilakukan perancangan, implementasi dan pengukuran waktu respon.

Pada rancangan model sistem LVS (Linux Virtual server) dilakukan pengujian dengan tahap sebagai berikut:

1. Client melakukan requests ke server melalui load balancer dengan tool httpperf dengan jumlah koneksi user antara 10.000/s – 50.000/s.
2. Komputer load balancer menerima requests dari client kemudian melakukan scheduling dan rewriting packets. Pada proses ini load balancer mengirimkan request ke real server yang sedang aktif dengan menggunakan algoritma round robin.
3. Real server menerima request dari client melalui komputer load balancer dan mengirimkan reply ke load balancer.
4. Komputer load balancer menerima reply dari real server yang sedang aktif, kemudian meneruskan reply ke komputer client yang meminta request. Client menerima replies dari load balancer. Pada proses ini komputer client dapat mengetahui respon time dari sebuah web server.

2.1 SISTEM LOAD BALANCING

Seperti telah dijelaskan di atas, sistem load balancing sebenarnya dapat dibuat dengan banyak cara. Pembuatannya tidak terikat oleh sebuah operating system saja, atau hanya dapat dibuat oleh sebuah perangkat saja. Namun secara garis besar cara pembuatan sistem load balancing terbagi menjadi tiga kategori besar yaitu:

1. DNS round robin
2. Integrated load balancing
3. Dedicated load balancing

Ketiga jenis ini memiliki cara kerja yang unik dan berbeda satu sama lain, tetapi tetap menuju suatu hasil akhir yang sama, yaitu menciptakan sebuah sistem yang lebih menjamin kelangsungan hidup jaringan di belakangnya dan membuatnya lebih scalability.

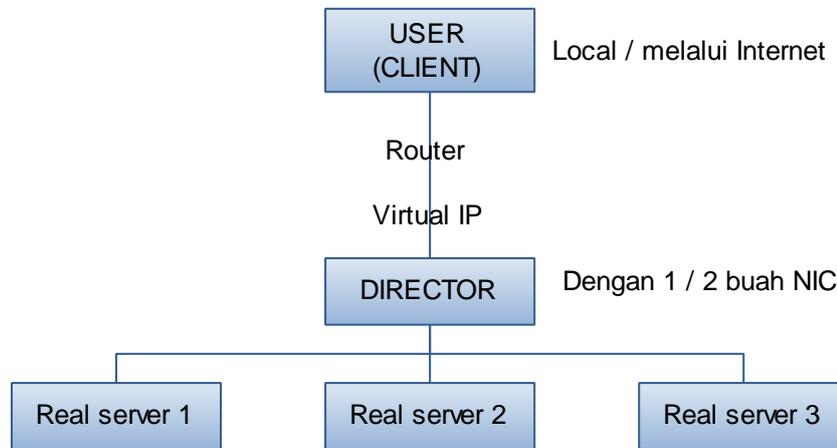
Load adalah suatu hal yang sangat penting sekali pada sistem yang diharapkan dapat menangani beban simultan yang besar. Load balancing adalah suatu proses untuk memindahkan proses dari host yang memiliki beban kerja tinggi ke host yang memiliki beban kerja kecil. Ini bertujuan agar waktu rata-rata mengerjakan tugas akan rendah dan menaikkan utilitas prosesor.

2.2 LINUX VIRTUAL SERVER

Linux Virtual Server atau disingkat LVS merupakan suatu teknologi clustering yang dapat digunakan untuk membangun suatu server dengan menggunakan kumpulan dari beberapa buah realserver. LVS merupakan implementasi dari komputer cluster dengan metoda High Availability. LVS mengimbangi berbagai bentuk dari service jaringan pada banyak mesin dengan memanipulasi paket sebagaimana diproses TCP/IP stack. Satu dari banyak peran yang paling umum dari Linux Virtual Server adalah bertindak sebagai server yang berada pada garis terdepan dari kelompok server web. Seperti ditunjukkan pada Gambar 1 Linux Virtual Server atau LVS ini terdiri dari sebuah Director dan beberapa realserver yang bekerja bersama dan memberikan servis terhadap permintaan user. Permintaan User diterima oleh Director yang seolah olah berfungsi sebagai IP Router yang akan meneruskan paket permintaan user tersebut pada real server yang siap memberikan servis yang diminta.

Dengan demikian virtual server akan terdiri dari beberapa komputer Yang mempunyai image yang sama tetapi ditempatkan pada IP yang berbeda. User dapat

mengakses virtual server tersebut dengan bantuan suatu Director, yang bertugas untuk melakukan pemetaan IP dari server dan komputer lainnya yang berperan sebagai virtual server.



Gambar 1. Struktur Dasar Linux Virtual Server dengan 3 buah real server.

LVS Director adalah modifikasi dari sistem Linux yang bertanggung jawab untuk mendistribusikan permintaan user/client terhadap realserver pada kelompok server. Realserver melakukan pekerjaan untuk memenuhi permintaan serta memberikan atau membuat laporan balik kepada user/client.

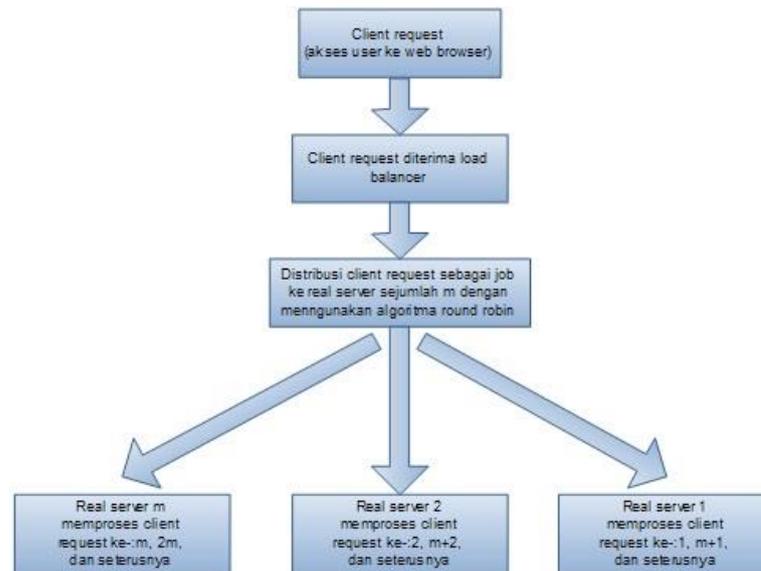
LVS Director memelihara rekaman daripada sejumlah permintaan yang telah ditangani oleh masing-masing realserver dan menggunakan informasi ini ketika memutuskan server mana yang akan ditugaskan untuk menangani suatu permintaan berikutnya.

LVS juga dapat memiliki Director cadangan yang akan menggantikan bilamana suatu saat Director utama mengalami suatu kegagalan sehingga membentuk suatu LVS failover.

Masing-masing realserver dapat bekerja dengan menggunakan berbagai sistem operasi dan aplikasi yang mendukung TCP/IP dan Ethernet. Pembatasan dan pemilihan sistem operasi pada realserver serta jenis servis yang didukung oleh realserver dilakukan pada saat proses konfigurasi LVS dijalankan.

2.3 ALGORITMA PENJADWALAN (SCHEDULING) ROUND ROBIN

Mekanisme penjadwalan pada LVS dikerjakan oleh sebuah patch kernel yang disebut modul IP Virtual Server atau IPVS modules. Modul ini mengaktifkan layer 4 yaitu transport layer switching yang dirancang dapat bekerja dengan baik pada multi server dalam IP address tunggal (virtual IP address). IPVS membuat IPVS table pada kernel untuk menelusuri dan merutekan paket ke real server secara efisien. Tabel ini digunakan oleh load balancer yang sedang aktif (yang pasif adalah backup-nya) untuk meneruskan client request dari virtual IP address ke real server. IPVS table secara rutin diperbarui menggunakan software ipvsadm.



Gambar 2. Algoritma penjadwalan round-robin

Pada penjadwalan tipe round-robin, manager mendistribusikan client request sama rata ke seluruh real server tanpa memperdulikan kapasitas server ataupun beban request. Jika ada tiga real server (A,B,C), maka request 1 akan diberikan manager kepada server A, request 2 ke server B, request 3 ke server C dan request 4 kembali ke server A. Mekanisme ini dapat dilakukan jika seluruh real server menggunakan spesifikasi komputer yang sama.

Konsep dasar dari algoritma ini adalah dengan menggunakan time-sharing. Pada dasarnya algoritma ini sama dengan FCFS, hanya saja bersifat preemptive. Setiap proses mendapatkan waktu CPU yang disebut dengan waktu quantum (quantum time) untuk membatasi waktu proses, biasanya 1-100 milidetik. Setelah waktu habis, proses ditunda dan ditambahkan pada ready queue.

Algoritma Round Robin merupakan algoritma yang paling sederhana dan banyak digunakan oleh perangkat load balancing . Algoritma ini membagi beban secara bergiliran dan berurutan dari satu server ke server lain sehingga membentuk putaran. Penjadwalan ini merupakan:

1. Penjadwalan preemptive, bukan di-preempt oleh proses lain, tapi terutama oleh penjadwal berdasarkan lama waktu berjalannya proses, disebut preempt by time.
2. Penjadwal tanpa prioritas.

Semua proses dianggap penting dan diberi sejumlah waktu proses yang disebut kwanta (quantum) atau time slice dimana proses itu berjalan.

Ketentuan algoritma round robin adalah sebagai berikut:

1. Jika kwanta dan proses belum selesai maka proses menjadi runnable dan pemroses dialihkan ke proses lain.
2. Jika kwanta belum habis dan proses menunggu suatu kejadian (selesainya operasi I/O), maka proses menjadi blocked dan pemroses dialihkan ke proses lain.
3. Jika kwanta belum habis tapi proses telah selesai, maka proses diakhiri dan pemroses dialihkan ke proses lain.

Algoritma penjadwalan ini dapat diimplementasi sebagai berikut:

- Mengelola senarai proses read (runnable) sesuai urutan kedatangan.

- Ambil proses yang berada di ujung depan antrian menjadi running.
- Bila kwanta belum habis dan proses selesai maka ambil proses di ujung depan antrian proses ready.
- Jika kwanta habis dan proses belum selesai maka tempatkan proses running ke ekor antrian proses ready dan ambil proses di ujung depan antrian proses ready.

Algoritma round robin menggilir proses yang ada di antrian. Proses akan mendapat jatah sebesar time quantum. Jika time quantum-nya habis atau proses sudah selesai, CPU akan dialokasikan ke proses berikutnya. Tentu proses ini cukup adil karena tak ada proses yang diprioritaskan, semua proses mendapat jatah waktu yang sama dari CPU yaitu $(1/n)$, dan tak akan menunggu lebih lama dari $(n-1)q$ dengan q adalah lama 1 quantum.

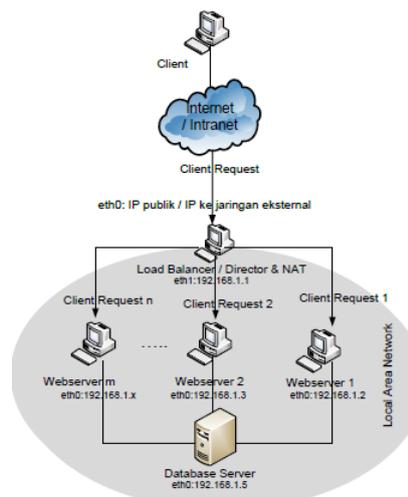
Algoritma ini sepenuhnya bergantung besarnya time quantum. Jika terlalu besar, algoritma ini akan sama saja dengan algoritma first come first served. Jika terlalu kecil, akan semakin banyak peralihan proses sehingga banyak waktu terbuang. Permasalahan utama pada Round Robin adalah menentukan besarnya time quantum. Jika time quantum yang ditentukan terlalu kecil, maka sebagian besar proses tidak akan selesai dalam 1 quantum. Hal ini tidak baik karena akan terjadi banyak switch, padahal CPU memerlukan waktu untuk beralih dari suatu proses ke proses lain (disebut dengan context switches time). Sebaliknya, jika time quantum terlalu besar, algoritma Round Robin akan berjalan seperti algoritma first come first served. Time quantum yang ideal adalah jika 80% dari total proses memiliki CPU burst time yang lebih kecil dari 1 time quantum.

2.4 Metode Forwarding Pada LVS

Ada tiga metode forwarding yang dapat digunakan pada cluster berbasis LVS.

a. LVS-NAT

Pada metode LVS-NAT, load balancer berfungsi pula sebagai Network Address Translation (NAT). Komputer yang berfungsi sebagai load balancer minimal memiliki dua NIC (Network Interface Card) yaitu eth0 dan eth1.



Gambar 3. LVS Via NAT

b. LVS-TUN

LVS-TUN menggunakan metode IP Tunneling agar paket dapat dienkapsulasi oleh director dan diteruskan menuju real server. Setelah diproses oleh real server, client request langsung dikembalikan ke client tanpa melalui director. LVS-TUN cocok untuk director dan real server yang tidak berada dalam jaringan yang sama atau dihubungkan via internet.

c. LVS-DR

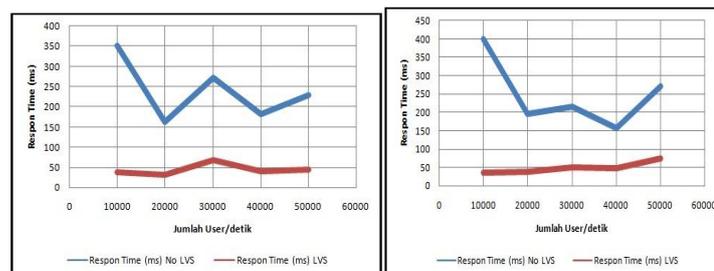
Pada LVS-DR (LVS-Direct Routing), real server mengembalikan client request yang telah diprosesnya langsung ke client yang memintanya. Perbedaannya dengan topologi lainnya adalah LVS-DR tidak memerlukan tunneling server, namun director dan real server harus berada di dalam jaringan yang sama.

III. HASIL DAN PEMBAHASAN

Untuk menganalisa respon time dari sebuah web server menggunakan algoritma round robin dilakukan perbandingan pengujian terhadap peningkatan performasi dari respon time sebuah web server yang menggunakan sistem dengan metode linux virtual server (LVS) dan tanpa menggunakan metode linux virtual server (LVS). Suatu pengujian perlu dilakukan terhadap sistem ini untuk menunjukkan peningkatan performa dari respon time sebuah web server. Dimana pada saat ini masih banyak perusahaan yang menggunakan server tunggal untuk melayani banyak client. Pengujian dilakukan dengan software network monitoring tool atau network analyzer untuk mengukur response time. Dalam pengujian dilakukan percobaan pengiriman client request dengan cara membuka halaman situs pendaftaran siswa baru secara online dalam jumlah yang bervariasi dari 10.000 hingga 50.000 kali akses secara bersamaan kemudian mengamati hasil respon time.

Tabel 1. Perbandingan Respon time

Koneksi	Respon Time		Koneksi	Respon Time	
	Non LVS (ms)	LVS (ms)		Non LVS (ms)	LVS (ms)
10000	401	36,3	10000	352,1	37,4
20000	195,1	39	20000	163,1	31
30000	213,9	51,4	30000	273,2	68,7
40000	156	47,4	40000	183,5	39,3
50000	271,4	75	50000	229	43,5



Gambar 4. Grafik perbandingan respon time.

Pada pengujian didapatkan hasil respon time dari web server dalam memproses client request secara umum meningkat sesuai dengan penambahan jumlah web server yang ada dalam cluster. Sehingga tujuan dalam pembuatan sistem load balancing yang dirancang untuk mempercepat proses pelayanan suatu web server terhadap client request yang meminta (membuka) halaman situs pendaftaran siswa baru secara online sudah tercapai pada pengujian.

Dengan menggunakan metode LVS maka waktu respon yang dihasilkan berbeda jauh dengan sistem yang tidak menggunakan sistem LVS, sistem yang menggunakan LVS memiliki waktu respon yang lebih cepat bila dibandingkan dengan sistem yang tidak menggunakan LVS. Sistem yang menggunakan LVS memiliki waktu respon yang cukup stabil.

IV. PENUTUP

Berdasarkan hasil uji coba dan analisis yang telah dilakukan maka dapat disimpulkan bahwa dengan menggunakan penerapan Load Balancing pada cluster computer maka down-time server dapat diatasi oleh komputer server lainnya.

Selain itu dengan menggunakan sistem LVS (Linux Virtual Server) maka respon time dari sebuah web server lebih cepat dan stabil bila dibandingkan dengan sistem yang tidak menggunakan LVS (Linux Virtual Server) yaitu 36,3ms dengan jumlah koneksi user antara 10.000/s – 50.000/s.

Untuk pengembangan penelitian selanjutnya dan untuk mendapatkan hasil performansi yang lebih baik maka dalam sistem ini perlu penambahan backup director yang digunakan untuk meminimalisasi down-time dari computer load balancer. Serta penggunaan komputer server dengan spesifikasi yang sama, agar didapatkan hasil analisis yang akurat.

V. DAFTAR PUSTAKA

- [1] Arbie. *Manajemen Database dengan MySql*. Andi. Yogyakarta. 2003.
- [2] Buku-jarkom2.pdf. dokumen online. dari URL: <http://toko.baliwae.com/>. diakses: Desember 2011.
- [3] Haris Nasution Abdul. *Komparasi Algoritma Penjadwalan Pada Layanan Terdistribusi Load Balancing Lvs Via Nat*. Institut Teknologi Sepuluh Nopember. Surabaya. 2011.
- [4] Husni. *Solusi Web Service Berbasis Linux*. Graha Ilmu. Yogyakarta. 2004.
- [5] MySql Master-Master Replication (Linux). dokumen online. dari URL: <http://ndalemk6p4a.wordpress.com/>, diakses: Januari 2012.
- [6] Rafiudin Rahmat. *Panduan Menjadi Administrator Sistem Unix*. Andi. Yogyakarta. 2004.
- [7] Suprajitno R. Djoko, Ari B. Pranata, Sirojuddin Ahmad, Tri W. Endri, dan Affandi Achmad. *Desain Server Cluster Learning Management System (LMS) Berbasis Linux Virtual Server (LVS) Dengan Algoritma Penjadwalan Round Robin*. Institut Teknologi Sepuluh Nopember (ITS). Surabaya.