

Penerapan Metode Round Robin Pada Jaringan Multihoming Di Computer Cluster

by Itjrd UIR

Submission date: 16-Jun-2025 09:00PM (UTC+0700)

Submission ID: 2681223143

File name: yudhiarta,_Journal_editor,_3_Yudi.pdf.docx (470.68K)

Word count: 2282

Character count: 14390

Penerapan Metode Round Robin Pada Jaringan Multihoming Di Computer Cluster

Yudhi Arta¹

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Islam Riau
E-mail : yudhiarta@uir.ac.id

2 Abstract

The importance of an information currently required a system capable of providing the best service, especially information service with the websites media. Increasing the amount of access to the website resulting into increased workload webserver and can not be solved with a single server. This problem can be resolved by applying the cluster load balance method. Basically cluster load balance work by sharing webserver workload is distributed to multiple server nodes so that the websites becomes balanced. Weight round robin scheduling algorithm can balance the load by determining the amount of weight to each server node. With the load balance by using a scheduling algorithm is expected to prevent the overload requests, workload leveling webserver, and speed up the response time and throughput of the performance webserver.

Keyword : Load balance, Linux Virtual Service, Weight Round Robin, Respon Time, Throughput.

Abstrak

Pentingnya sebuah informasi saat ini menuntut sebuah sistem sanggup memberikan pelayanan yang terbaik, khususnya pelayanan informasi dengan media *website*. Meningkatnya jumlah akses terhadap *website* mengakibatkan beban kerja *webserver* menjadi meningkat dan tidak bisa diatasi dengan *single server*. Permasalahan tersebut dapat diatasi dengan menerapkan metode *cluster load balance*. Pada dasarnya *cluster load balance* bekerja dengan membagikan beban kerja *webserver* secara terdistribusi ke beberapa *node server* agar *website* menjadi seimbang. Algoritma penjadwalan *weighted round robin* dapat menyeimbangkan beban dengan menentukan jumlah bobot ke masing-masing *node server*. Dengan adanya *load balance* dengan menggunakan algoritma penjadwalan diharapkan mampu mencegah terjadinya *request* yang *overload*, meratakan

beban kerja *webserver*, dan mempercepat *respon time* serta *throughput* terhadap kinerja *webserver*.
Kata kunci : Load balance , Linux Virtual Service , NAT, Weighted Round Robin, Respon Time, Throughput.

1. PENDAHULUAN

Round Robin merupakan salah satu metode load balance yang sederhana dalam mengembangkan beban (load). Load balance dengan metode Round Robin yaitu menggunakan kedua gateway secara bersamaan dengan membagi beban secara berurutan dan bergiliran.

Pemrosesan paralel (*parallel processing*) adalah penggunaan lebih dari satu CPU untuk menjalankan sebuah program secara simultan. Idealnya, *parallel processing* membuat program berjalan lebih cepat karena semakin banyak CPU yang digunakan. Tetapi dalam praktek, seringkali sulit membagi program sehingga dapat dieksekusi oleh CPU yang berbeda-beda tanpa berkaitan di antaranya.

Pada saat ini perkembangan teknologi berkembang dengan sangat pesat, hal ini dibuktikan dengan cepatnya perkembangan sebuah informasi. Informasi sudah menjadi keperluan pokok oleh berbagai bidang, baik instansi pemerintahan, institusi pendidikan,

maupun perusahaan-perusahaan. Mengingat pentingnya informasi tersebut, maka dituntut agar dapat bisa diakses di mana saja dan kapan saja, yaitu dengan menggunakan jasa internet sebagai sarana penyedia informasi. Jasa internet yang umum digunakan yaitu World Wide Web (WWW) atau sering dikenali dengan web.

Kebutuhan internet yang semakin meningkat membuat pengguna internet membutuhkan bandwidth yang besar. Sehingga tidak jarang di perusahaan, intitusi pendidikan, warnet bahkan di perumahan menerapkan multihoming untuk memenuhi kebutuhannya. Jaringan multihoming merupakan suatu sistem jaringan yang memiliki jalur keluar lebih dari satu.

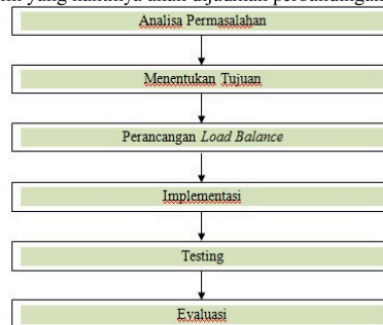
Pentingnya sebuah media informasi mengakibatkan meningkatnya jumlah pengunjung yang mengakses website setiap harinya, serta masih menggunakan single server dalam penyedia layanan tersebut maka faktor inilah menjadi faktor sistem menjadi *overload* dan *crash* terhadap *request*, serta performansi kinerja website menjadi berkurang. Dari dampak tersebut maka mengakibatkan beban pada server menjadi meningkat dan waktu respon pada layanan tersebut menjadi lambat. Sehingga dengan adanya permasalahan tersebut, maka perlu diperhatikan cara untuk mengatasi beban server tersebut serta peningkatan ketersediaan dan meminimalkan waktu tanggap dari webserver tersebut.

Clustering merupakan teknik dimana dua atau lebih *webserver* dikelompokkan bersama sebagai sebuah *cluster* yang mengakomodasi peningkatan beban. Dengan adanya permasalahan di atas, maka perlu adanya *clustering* untuk mengatasi masalah tersebut, yaitu dengan menerapkan metode *load balancing* pada *webserver*, maka beban server akan dibagi dengan rata sesuai dengan jumlah server yang telah tercluster.

2. METODE PENELITIAN

2.1 Metode Penelitian

Tahapan kerja penelitian ini akan diuraikan pada kerangka kerja, yang mana kerangka kerja tersebut akan digunakan sebagai acuan atau langkah dalam membangun load balance cluster ini. Pada awalnya yang akan dilakukan yaitu melakukan analisis terhadap kebutuhan membangun load balance ini. Selanjutnya akan dilakukan dengan merancang dan menguji dari load balance itu sendiri serta mengetahui hasil dari pengujian penelitian ini yang nantinya akan dijadikan perbandingan.

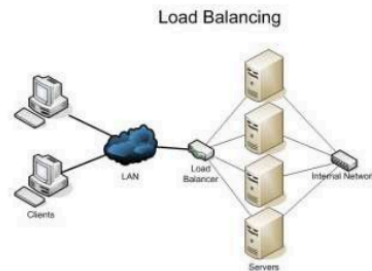


Gambar 2.1. Kerangka Kerja Penelitian

2.2 Load Balance Cluster

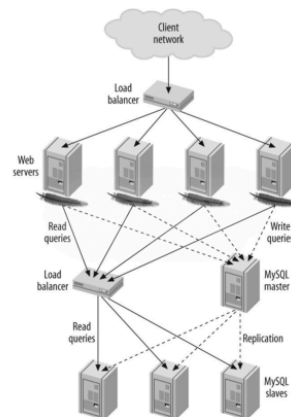
Load balance cluster bekerja dengan mengirimkan layanan-layanan di dalam sebuah jaringan ke *node-node* yang telah di-*cluster* untuk menyeimbangkan beban permintaan layanan di antara *node cluster*. Prinsip kerja *load balance cluster*, ketika *node* pada *cluster load balance* tidak bekerja maka aplikasi *load balance* akan mendeteksi kegagalan dan meneruskan permintaan ke *node cluster* lainnya.

Pada dasarnya mekanisme penyeimbang beban *load balance* yaitu setiap beban yang masuk ke *load balancer* akan dialihkan ke masing-masing *server*, berdasarkan bobot yang *server* dan koneksi yang masuk. Sehingga setiap *request* yang masuk akan dibagi secara merata atau seimbang kepada masing-masing *real server*.



Gambar 2.2. Load Balancing

Load balancing merupakan kemampuan untuk menyebarkan beban dari proses untuk sebuah aplikasi kepada beberapa sistem yang berbeda untuk meningkatkan kemampuan pemrosesan pada permintaan yang datang. Sehingga dengan *load balancing* akan mengirimkan beberapa porsi pemrosesan dari permintaan ke sebuah sistem kepada sistem independen lain yang akan ditangani secara bersamaan.



Gambar 2.3. Arsitektur Load balance

2.3 Linux Virtual Server

Linux Virtual Server yaitu suatu server yang mempunyai skalabilitas dan mempunyai high availability yang dibangun di atas sebuah cluster dengan beberapa real server yang berada dibelakangnya. Pada *Linux Virtual Server* terdapat 3 metode, yaitu *Linux Virtual Server via NAT (LVS-NAT)*, *Linux Virtual Server via Direct-Routing (LVS-Direct Routing)*, dan *Linux Virtual Server via Tunneling*.

2.4 Algoritma Penjadwalan

Algoritma Penjadwalan yaitu metode yang digunakan untuk membagikan beban kerja ke masing-masing real server yang berada dibelakang director. Jenis-jenis algoritma penjadwalan yang umum digunakan yaitu algoritma *round robin*, *weighted round robin*, *ratio*, *fastest*, dan *least-connection*.

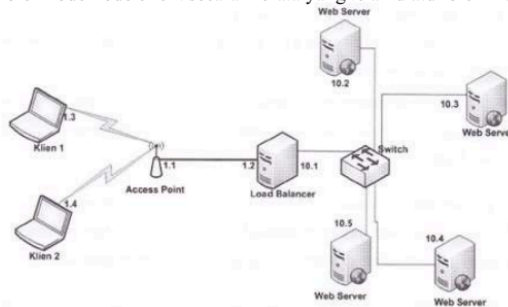
2.5 Iptables

Iptables merupakan sebuah aplikasi pada sebuah sistem operasi yang berfungsi sebagai alat *filtering* terhadap lalu lintas data. Pada *iptables* ini mengatur semua lalu lintas pada jaringan, baik yang masuk maupun keluar atau hanya sekedar melewati saja. *Iptables* dapat mengatur semua kegiatan dalam sistem komputer baik besar data yang boleh lewat, jenis paket data yang dapat diterima, mengatur trafik sesuai dengan asal dan tujuan data, *forwarding*, *NAT*, *redirecting*, pengelolaan *port*, dan *firewall*.

3. HASIL DAN PEMBAHASAN

3.1 Perancangan Topologi

Pada perancangan topologi dengan penerapan *load balance* ini, mempunyai sebuah node yang bertindak sebagai master / director dan beberapa node yang akan bertugas sebagai client. Dengan adanya penerapan *load balance* ini, semua beban kerja akan dikerjakan oleh node-node client secara merata yang telah diatur oleh master cluster.



Gambar 3.1. Topologi Webserver Load balance

3.2 Instalasi (Installation) dan Implementasi Sistem

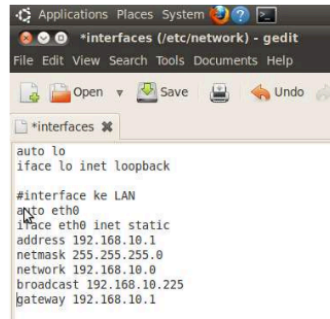
Pada tahap ini menjelaskan langkah-langkah perancangan Load Balance menggunakan router dengan metode *Round Robin*. Berdasarkan beberapa tahapan sebelumnya, didapat beberapa proses yang diperlukan untuk merancang *Load Balance* menggunakan router dengan metode *Round Robin*, yaitu :

1. Konfigurasi interface

Konfigurasi awal yang dilakukan dari sisi router *load balance* adalah mengkonfigurasi *interface*. Pada perancangan *router load balance* ini, server memiliki 3 interface yaitu *eth0*, *ppp0*, dan *ppp1*. Interface *eth0* akan diarahkan ke client sedangkan interface *ppp0* dan *ppp1* diarahkan ke modem. Perintah yang dijalankan untuk melakukan konfigurasi *interface* pada terminal Linux, yaitu :

gedit /etc/network/interfaces

Berikut merupakan konfigurasi *interface* yang akan digunakan sebagai router *load balance*:

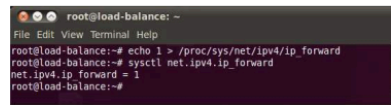


Gambar 3.2. Konfigurasi IP

Pada konfigurasi *interface* dilakukan konfigurasi secara manual pada *eth0* saja. Interface *ppp0* dan *ppp1* tidak dikonfigurasi secara manual karena *ppp0* dan *ppp1* menggunakan *ip dynamic* sehingga dengan otomatis mendapatkan IP ketika modem terkoneksi dengan *internet*.

2. Mengaktifkan IP forward

Langkah ini dilakukan dari sisi router, ini bertujuan agar komputer yang akan digunakan sebagai router dapat meneruskan traffic dari LAN ke client.

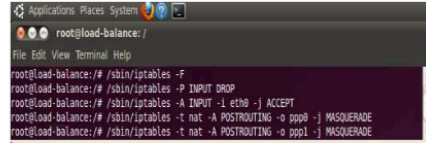


Gambar 3.3. IP Forward

IPTables merupakan tool yang sudah tersedia pada system operasi Linux ubuntu 10.04, fungsinya adalah men-setup, mengatur, dan memeriksa paket dari aturan pemfilteran IP. Pada perancangan *router load balance* proses *request* dari *client* akan diarahkan ke modem.

3. Membuat *table routing*

Langkah awal yang dilakukan adalah melakukan proses konfigurasi. Berikut merupakan langkah-langkah konfigurasi dari router (*load balance*), yaitu :



Gambar 3.4. IPTables Load Balance

Membuat *table routing* dengan cara :

```
#gedit/etc/iproute2/rt_tables
```

4. Membuat script *load balance round robin*

Didalam script *load balance round robin* terdapat :

```
if [ ! -z "$(route -n | grep "$DEV1")" ];  
then  
y=1  
fi  
if [ ! -z "$(route -n | grep "$DEV2")" ];  
then  
x=1  
fi
```

Script tersebut berfungsi untuk mengecek apakah di dalam *route -n* terdapat *dev1/dev2* atau tidak. Dimana *dev1* adalah *ppp0* dan *dev2* adalah *ppp1*. apabila tidak ditemukan *dev1* maka akan memberikan nilai *y=1* dan apabila tidak ditemukan interface *dev2* maka nilai *x* adalah 1.

3.3. Pengujian (Testing)

Tahap pengujian dilaksanakan pada sistem yang telah dirancang dan dikonfigurasi pada tahap sebelumnya. Pada tahap ini akan dilakukan pengujian apakah router (*load balance*) telah dapat berfungsi dengan baik dan dapat menjalankan tugasnya. Untuk pengujian akan dilakukan dalam beberapa cara, yaitu :

1. Pengujian Menggunakan *Web Browser*

Pengujian ini dilakukan untuk melihat apakah kedua gateway bisa digunakan secara bersama-sama. Pada pengujian ini kedua modem sudah aktif dan *load balance* sudah aktif. Pengujian ini menggunakan web browser www.ip-adress.com dan www.checkip.org. Pada pengujian ini menggunakan kartu GSM three (3) dan XL.

Ip address three (3) yang terhubung internet adalah 180.214.232.26 sedangkan ip address XL yang terhubung dengan internet adalah 202.152.243.2.

Tabel 3.1. Pengujian Menggunakan Web

Pengujian ke-	www. ip-adress.com	www. checkip.org
1	180.214.232.26	180.214.232.26
2	202.152.243.2	202.152.243.2
3	180.214.232.26	202.152.243.2
4	180.214.232.26	180.214.232.26
5	202.152.243.2	202.152.243.2
6	202.152.243.2	202.152.243.2
7	180.214.232.26	180.214.232.26
8	202.152.243.2	180.214.232.26
9	180.214.232.26	202.152.243.2
10	202.152.243.2	180.214.232.26

Dari tabel diatas dapat terlihat bahwa kedua gateway bisa digunakan secara bersama-sama. Ketika mengakses alamat www.ip-adress.com sebanyak 10 kali, jumlah ip 202.152.243.2 adalah 5 kali dan ip 180.214.232.26 berjumlah 5 kali. Ini menunjukkan bahwa kedua *gateway* digunakan secara seimbang.

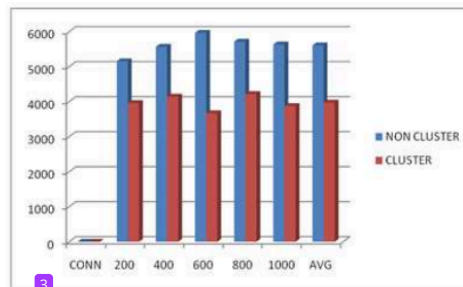
Penggunaan gateway tidak selamanya saling bergantian karena hal ini dipengaruhi oleh sinyal modem GSM, sehingga kecepatan akses internetnya pun berubah. Dengan demikian dapat disimpulkan bahwa load [balance](#) pada jaringan multihoming menggunakan router dengan *metode round robin* bisa [berjalan dengan baik](#).

2. Pengujian Hasil Load Balance

Pada bagian hasil dan pembahasan, penulis akan menampilkan ³ hasil perbandingan pengujian *load balance* dari respon time dan thoroughput pada masing-masing arsitektur, baik dengan *single server* maupun *load balance*.

3
Tabel 3.1. Hasil Perbandingan Respon Time Single Server dan Load balance

Total Connection	Single Server	Server Load balance
200	5155,6	3595,1
400	5566,3	4142,7
600	5967,1	3672
800	5714	4219,7
1000	5632,5	3876,4
Rata-rata	5607,1	3973,98



Gambar 3.6. Grafik Perbandingan Respon Time Single Server dan Load balance

4. KESIMPULAN

Kesimpulan dari pengujian ini adalah :

1. Load balance menghasilkan nilai respon time yang lebih sedikit dan nilai throughput yang lebih besar dibandingkan dengan arsitektur single server.
2. Sistem load balance dengan metode round robin bisa diterapkan pada jaringan multihoming. Dan Sistem ini berfungsi untuk menyeimbangkan traffic penggunaan internet pada jaringan multihoming.
3. Dengan algoritma penjadwalan weighted round robin, beban webserver dapat didistribusikan dengan baik kepada masing-masing server dengan ketentuan bobot yang diberikan.
4. Pada pengujian webserver harus diberikan waktu proses pengujian (timeout) untuk mendapatkan proses stress load yang tidak terlalu lama. Semakin lama proses pengujian maka hasil pengujian tidak menghasilkan nilai yang baik untuk dijadikan sebagai perbandingan.
5. Load balance dapat membantu dalam mempercepat kinerja proses terhadap penyediaan layanan webserver.
6. Load balance dapat memberikan ketersediaan informasi yang tinggi (high availability) terhadap website.

7. Dengan adanya *load balance server* kita dapat membagi beban sistem yang akan dikirimkan ke masing-masing node dengan menggunakan algoritma penjadwalan Weighted Round Robin.
8. Dengan adanya *load balance* dengan penerapan NAT, kita dapat menghemat jumlah IP yang diberikan oleh provider untuk dimanfaatkan semaksimal mungkin untuk perangkat yang lain.
9. *Load balance* dapat digunakan dalam berbagai hal yaitu pelayanan terhadap website, learning management system, Mail, pembagian beban bandwidth, dan lain-lain.

DAFTAR PUSTAKA

- [1] A. B. M. Moniruzzaman and Syed Akther Hossain (2014), "A Low Cost Two-Tier Architectur Model for High Availability Clusters Application Load Balancing", Volume 7, Number 1.
- [2] Ambia Rachman Haryadi (2010), " Load Balancing Menggunakan Linux Virtual Server (LVS) Via Network Address Translation (NAT) Pada PT. Bank Syariah".
- [3] Andy Purnama Nurhatta, Adian Fatchur Rochim and R.Rizal Isnanto (2012), " Sistem Penyeimbang Beban Web Server Dengan Iptables", Volume 1, Number 3.
- [4] Burhanuddin and Yusep Rosmansyah (2008),"Studi Mengenai Kinerja Webserver Berbasis Linux Menggunakan Teknologi Load Balancing".
- [5] B. K. Gupta, S.C. Sharma and Jyothi Sethi (2013), " Performance Evaluation of ATM Networks with Round Robin and Weighted Round Robin Algorithm", Volume 5, issue 5.
- [6] Desy Lukitasari and Ahmad Fali Oklilas (2010), Jurnal Generic, "Analisis Perbandingan Load Balancing Web Server Tunggal Dengan Web Server Custer Menggunakan Linux Virtual Server", Volume 5, Number 2, 31.
- [7] Imam Maghribi Mursal (2011), "Desain dan Implementasi Load Balancing Jaringan Lokal Pada CV. Sukses Makmur Mandiri Palembang".
- [8] Jefty Alvonsius Rabu, Joko Purwadi and Willy S. Raharjo (2012), "Implementasi Load Balancing Web Server Menggunakan Metode LVS-NAT", Volume 8, Number 2.
- [9] Jiani Guo and Laxmi Narayan Bhuyan (2006), "Load Balancing In a Cluster-Based Web Server for Multimedia Application", Volume 17, Number 1.
- [10] Mei Lu Chin, Chong Eng Tan and Mohammad Imran Bandan (2012), "Efficient DNS Based Load Balancing For Bursty Web Application Traffic", Volume 1, Number 1.
- [11] N. Khrisnamoorthy, R. Asokan, PhD and S. Sangeetha (2013)," Performance Evaluation of Weighted Round Robin Grid Schedulling", Volume 68, Number 13.
- [12] Seok-Pil Lee and Eui-Seok Nahm (2012), "A New Approach to Modelling of Linux Virtual Server based on Performance Metrics Using an Optimal Load Balancing Algorithm", Volume 6, Number 2.
- [13] Sumit Srivastava, Pankaj Dadheech and Mahender Kumar Beniwal (2011), "Load Balancing Using High Performance Computing Cluster Programming", Volume 8, Issue 1.
- [14] Theddy R Maitimu (2008), "Perancangan dan Implementasi Webserver Clustering dengan Skema Load balance Menggunakan Linux Virtual Server Via NAT, Volume 5, Number 1.

-
- [15] Yessy Asri (2010), "Rancang Bangun Aplikasi Setting Load Balancing Webserver Pada Freebsd", Volume 3, Number 1.
 - [16] Yogi Kurniawan, Sabriansyah R.A., ST., M.Eng, Eko Sakti P., S.Kom., M.Kom (2012), "Analisis Kinerja Algoritma *Load balancer* dan Implementasi Pada Layanan Web".
 - [16] Arta, Y., Kadir, E. A., & Suryani, D. (2016, May). KNOPPIX: Parallel computer design and results comparison speed analysis used AMDAHL theory. In Information and Communication Technology (ICoICT), 2016 4th International Conference on (pp. 1-5). IEEE.
-

Penerapan Metode Round Robin Pada Jaringan Multihoming Di Computer Cluster

ORIGINALITY REPORT

21%

SIMILARITY INDEX

21%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

pdfs.semanticscholar.org

Internet Source

8%

2

www.e-jurnal.com

Internet Source

6%

3

journal.uir.ac.id

Internet Source

6%

Exclude quotes Off

Exclude matches < 6%

Exclude bibliography On

Penerapan Metode Round Robin Pada Jaringan Multihoming Di Computer Cluster

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10