

## Rancang Bangun Mirror Repository Linux Menggunakan Load-balanced Mirror Service System (LMSS) Pada Private Cloud

Ibnu Haris Fawaid<sup>\*1</sup>, Diah Risqiwati<sup>2</sup>, Denar Regata Akbi<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika/Universitas Muhammadiyah Malang

ibnuharisfawaid@webmail.umm.ac.id<sup>\*1</sup>, risqiwati@umm.ac.id<sup>2</sup>, dnarregata@umm.ac.id<sup>3</sup>

### Abstrak

Meningkatnya permintaan pembaharuan aplikasi berbasis open source karena seiring meningkatnya jumlah pengguna sistem operasi Linux, oleh karenanya ketika banyak pengguna linux yang melakukan permintaan pembaharuan aplikasi menyebabkan lambatnya kecepatan unduh dan membutuhkan waktu lama. Salah satu solusinya yaitu dengan membuat mirror repository linux agar memudahkan dan menawarkan koneksi pengguna yang lebih baik. Namun demikian, suatu saat ketika semakin bertambah jumlah pengguna mirror tersebut akan mengalami penurunan kualitas download hingga dapat mengalami down, sehingga tidak dapat melayani permintaan pembaharuan pengguna dengan baik. Penelitian ini bertujuan untuk merancang dan membangun Mirror Repository Linux dengan menggunakan Load-balanced Mirror Service System (LMSS) dengan mengintegrasikan beberapa Mirror Server yang dibangun secara generate melalui Manage Large Network (MLN), dengan pembagian beban menggunakan Linux Virtual Server (LVS) pada metode Network Address Translation (NAT). Sehingga beban permintaan dari client dapat ditangani Mirror Server secara bersama – sama. Dengan demikian, rata – rata waktu download dapat berkurang sebanyak 17,2 % daripada menggunakan Single Server.

**Kata Kunci:** Linux, Load-balancing, Linux Virtual Server

### Abstract

The increasing demand for open source-based application updates is due to the increasing number of users of Linux operating systems, therefore when many Linux users who request an application update cause slow download speed and takes a long time. One solution is to create Mirror Repository Linux in order to facilitate and offer better user connections. However, a time when the number of mirror users will decrease the quality of download until it can down, so it can not serve user renewal request well. This research aims to design and build linux mirror repositories by using Load-balanced Mirror Service System (LMSS) by integrating multiple Mirror Server generated via Manage Large Network (MLN), with load sharing using Linux Virtual Server (LVS) Network Address Translation (NAT). LMSS there is a Director Server as a counter / load manager that passes on the server, which will be forwarded to the destination server. So that the request load from the client can be handled Mirror Server together. Thus, the average download time can be reduced by 17.2% rather than using a single server.

**Keywords:** Linux, Cloud Computing, Load-balancing, Linux Virtual Server

### 1. Pendahuluan

Linux merupakan salah satu sistem operasi yang populer, diciptakan oleh Linus Torvalds pada tahun 1991, sistem operasi ini berhasil menyaingi sistem operasi lain dengan keunggulan yang banyak. Diantaranya banyak keunggulan tersebut, salah satunya Linux merupakan sistem operasi yang gratis, bebas dan mudah diunduh, disebarluaskan, hingga dikembangkan baik profesional maupun secara personal[1]. Sehingga tidak sedikit perusahaan yang menggunakan sistem operasi ini karena dapat memotong *budget* perusahaan tersebut. Dengan tawaran–tawaran tersebut, Linux mengalami peningkatan jumlah pengguna, begitu pula dengan permintaan pembaharuan paket aplikasi agar pengguna memiliki aplikasi terbaru agar memaksimalkan kinerja sistem operasi.

Ketika pengguna melakukan request update secara bersama–sama, tentu repository server mengalami peningkatan beban yang signifikan. Peningkatan beban tersebut dapat

mengakibatkan server dapat terlambat melayani atau bahkan down [2]. Salah satu solusi untuk menangani masalah demikian adalah meningkatkan hardware dari server tersebut. Akan tetapi dengan solusi tersebut, akan membutuhkan biaya yang mahal [3] dan suatu saat jika request update dari pengguna kembali meningkat, server akan mengalami down kembali pula. Dengan demikian, telah dilakukan penelitian untuk menangani masalah tersebut dengan cara membuat komputasi paralel, yaitu dengan cara mengintegrasikan banyak komputer server agar beban permintaan tersebut dapat ditangani secara bersama-sama [4].

Metode komputasi paralel juga telah dikembangkan dengan memecah bagian-bagian server, diantaranya dengan cluster server. Cluster server bertujuan agar sekelompok komputer server (mirror server) dapat ditangani dengan sebuah komputer cluster. Dengan demikian server-server tersebut akan lebih mudah diatur dengan cara mengendalikan cluster server tersebut.

Dalam membangun mirror server, tentunya membutuhkan banyak komputer fisik. Dengan demikian budget yang dikeluarkan juga lebih banyak [5]. Salah satu teknologi untuk meminimalisir budget tersebut adalah dengan cara membuat virtualisasi. Virtualisasi inilah sebagai mirror server secara virtual, dimana dalam keberadaannya tidak harus berbentuk server fisik. Cara inilah yang akan meminimalisir pengeluaran sehingga lebih menghemat biaya. Disamping itu, virtualisasi akan memudahkan seorang administrator untuk mengatur masing-masing server tersebut.

Dalam penelitian ini dilakukan membangun mirror repository linux dengan menggunakan Linux Mirror Service System (LMSS) yaitu dengan mengintegrasikan virtualisasi server dengan menggunakan Manage Large Network (MLN) dan membagi beban trafik dengan menggunakan Linux Virtual Server (LVS). Dari penelitian sebelumnya, LMSS dengan menggunakan LVS-Tunneling mampu mengurangi rata-rata waktu download sebesar 20% hingga 25% [2]. Pada penelitian kali ini, peneliti melakukan penelitian performansi Mirror Repository Linux yang dibangun dengan LMSS menggunakan LVS-NAT. Dari penelitian yang telah dilakukan, telah menghasilkan kesimpulan bahwa LMSS dengan menggunakan LVS – NAT dapat mengurangi rata – rata waktu download sebesar 17,2 %.

## **2. Tinjauan Pustaka**

### **2.1 Cloud Computing**

Cloud computing berarti komputasi awan, yaitu sebuah teknologi dimana sumber daya komputer pada umumnya seperti processor/central processing unit, media penyimpanan dan aplikasi-aplikasi di dalamnya menjadi abstrak yang berfungsi sebagai penyedia layanan jaringan yang menggunakan pola akses remote. Pengguna dapat menikmati layanan dari cloud computing dengan tanpa membutuhkan komponen secara fisik dan dapat dipakai kapanpun dan dimanapun.

Cloud computing dapat dibangun berdasarkan besarnya pengguna layanan, yaitu Public Cloud, Private Cloud dan Hybrid Cloud. Private cloud memiliki fungsi cloud untuk internal/private. Sedangkan Public Cloud dapat memberikan layanan cloud dalam skala besar dan dapat diakses oleh siapapun. Sedangkan Hybrd cloud merupakan gabungan dari keduanya [6] .

Dalam membangun Private Cloud, dapat menggunakan tools bermacam-macam, diantaranya yang populer adalah Proxmox, UEC, OpenNebula. Namun masih memiliki kelemahan yaitu tidak efisiensi dalam hal penggunaan memory, sehingga membutuhkan memory yang sangat besar. Salah satu tools untuk membangun private cloud adalah Manage Large Network (MLN), aplikasi ini mudah, simpel dan efisiensi penggunaan memory [2].

### **2.2 Virtualisasi Server**

Virtualisasi ini memungkinkan beberapa sistem operasi server dapat berjalan di satu mesin fisik yang sama. Tujuannya agar memaksimalkan fungsi infrastruktur/hardware yang dapat diandalkan dari sebuah server tersebut. Karena sering kali terjadi penggunaan sebuah server hanya 15-20% dari kemampuan sepenuhnya [7]. Fungsi virtualisasi tersebut juga dapat menghemat pemakaian sumber daya fisik maupun non fisik, sehingga dapat lebih efisien.

Setiap sistem operasi virtual pasti menggunakan Hypervisor. Hypervisor berfungsi sebagai Virtual Machine Manager (VMM) yaitu bagian yang melakukan abstraksi dari perangkat keras/fisik menjadi perangkat lunak yang dapat mendistribusikan beban kerja beberapa mesin virtual kedalam satu perangkat keras fisik.

### 2.3 Mesin Virtual Komputasi Awan

Manage Large Network (MLN) merupakan sebuah tool administrasi komputasi awan yang didesain untuk membangun dan menjalankan jaringan mesin virtual yang berbasis Xen Hypervisor, VMWare Server, dan Usermode Linux. MLN sangat cocok digunakan untuk membuat laboratorium jaringan virtual untuk pendidikan, penelitian, hosting, atau hanya sekedar bermain dengan mesin virtual [2].

MLN mengizinkan administrator untuk mendesain, membuat dan mengatur seluruh jaringan pada mesin virtual dan konfigurasi internalnya dalam basis satu template [8]. MLN juga mendukung dua platform virtualisasi open source yaitu XEN dan User-Mode Linux. Sebuah MLN Daemon memungkinkan jaringan virtual untuk melakukan penyebaran dan mengatur seluruh server fisik.

### 2.4 Load Balancing

Load balancing merupakan salah satu metode untuk membagikan beban kepada host cluster sehingga beban yang diterima menjadi lebih ringan [9]. Metode ini terdiri dari dua komponen server yaitu director server dan real server. Director server merupakan server yang bertugas membagi beban pada semua real server. Sedangkan real server merupakan komputer yang benar-benar melayani request dari client. Pembagian tersebut bertujuan untuk memanfaatkan utilitas prosesor serta mengerjakan tugas menjadi lebih singkat.

### 2.5 Linux Virtual Server

Linux Virtual Server (LVS) melanjutkan TCP session dan UDP session untuk membagi dan menyeimbangkan beban melalui beberapa real server, sehingga proses ini dapat menerapkan pemilihan pada layer 4 (transport) dari kernel Linux.

LVS dapat diimplementasikan melalui tiga cara, diantaranya dengan melalui Network Address Translation (NAT) yaitu dengan cara menghubungkan jaringan lokal dengan jaringan luar dengan memanfaatkan satu IP Public yang dapat memanipulasikan alamat IP dengan nomor port baik sumber maupun tujuannya. Sedangkan Direct Routing yaitu dengan meneruskan paket secara langsung kepada real server, dan IP Tunneling melewati alamat paket yang datang dari client melalui balancer kemudian diteruskan langsung ke masing-masing real server, merespon dan mengirim langsung ke alamat IP client yang melakukan request [10].

## 3. Perancangan Sistem

### 3.1 Metode Penelitian

Dalam menggunakan LVS, terdapat sebuah director aktif, yang bertugas untuk mengatur beban pada masing-masing server. Di sisi lain terdapat dua virtual server yang memuat real server di dalamnya serta memberikan layanan yang sama yaitu repository linux melalui protokol http.

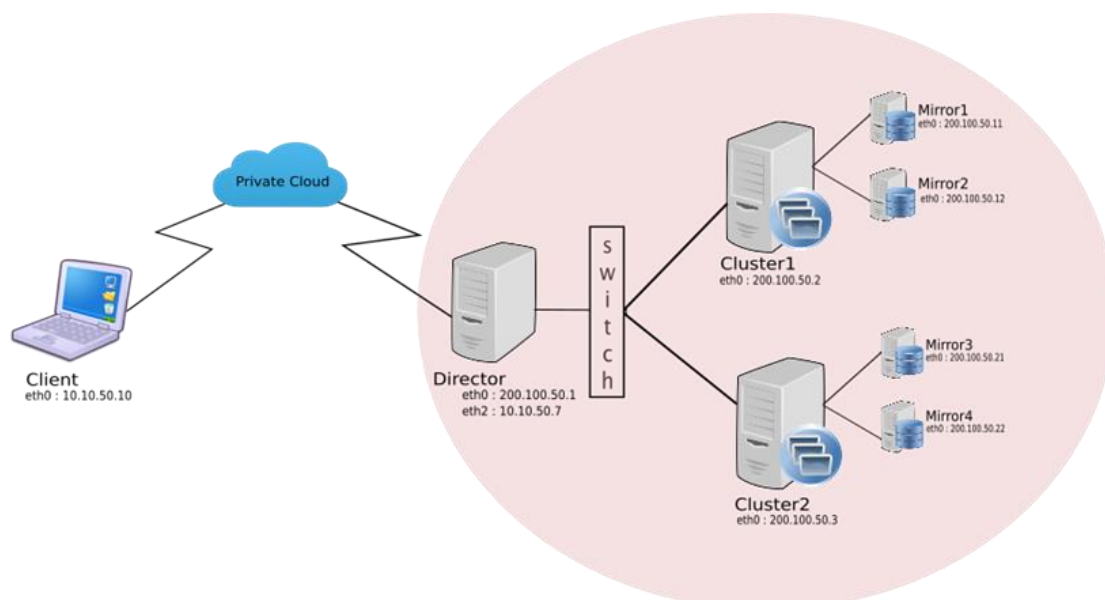
Dalam penelitian ini penulis menggunakan LVS via NAT untuk menyeimbangkan beban pada semua server dalam satu subnet yang sama, sehingga hasilnya lebih baik daripada LVS via lainnya. Oleh karena itu dalam membangun sistem tersebut terdapat komponen penyusun utama diantaranya :

1. Director server, sebagai sebuah server yang bertugas untuk meneruskan paket dari client kepada real server dan sebagai mesin yang menjalankan algoritma penjadwalan.
2. Cluster server, sebagai server fisik yang memiliki kapasitas besar dan di dalamnya memuat banyak mesin virtual. Jumlah cluster server pada penelitian ini sebanyak dua buah dengan spesifikasi yang sama.
3. Mirror server, sebagai server yang melayani permintaan client secara langsung. Dalam hal ini layanan linux repository berprotokol http.
4. Client, sebagai generator pembangkit request layanan update manager untuk menguji load-balancing dan sistem cluster. Pada perancangan ini hanya menggunakan lima client untuk merequest layanan update manager secara bersamaan.

### 3.2 Arsitektur Sistem

Pada penelitian ini terdiri dari tiga buah server yang masing-masing berfungsi sebagai penyedia layanan repository server, satu berfungsi sebagai director yaitu bertugas untuk membagi beban yang berasal dari client dan dua diantaranya sebagai Cluster server untuk menyimpan Mirror server. Mirror server dibangun melalui MLN yang dapat digenerate dari Diector

server. Server–server tersebut dapat terhubung dalam satu jaringan lokal. Client mengakses layanan mirror repository pada jaringan intranet melalui gateway Director Server, kemudian Director Server membagi ke masing–masing Mirror server sebagaimana dengan karakteristik algoritma Round Robin. Director server juga memiliki akses internet yang berfungsi untuk mengupdate repository dari mirror server yang telah dipilih. Arsitektur sistem yang digunakan seperti pada Gambar 1.



Gambar 1. Arsitektur Sistem LMSS

### 3.2 Pengujian Sistem

Pengujian yang akan dibagi menjadi dua skenario, diantaranya skenario 1 dalam kondisi single server dan skenario 2 dalam kondisi multi server menggunakan LMSS. Masing–masing skenario dilakukan dengan tiga tahap pengujian, diantaranya pengujian fungsionalitas sistem, pengujian Quality of Services (QoS) dan pengujian rata–rata waktu download.

1. Tahap pertama dilakukan pengujian fungsionalitas sistem. Pengujian ini bertujuan untuk memastikan bahwa layanan–layanan server (sistem) berfungsi dengan baik. Layanan–layanan tersebut diantaranya SSH Server, MLN Daemon–Master, FTP Server, Web Server, Mirror Repository Server, Rsync dan Crontab.
2. Tahap kedua dilakukan pengujian Quality of Services (QoS). Tujuan pengujian ini adalah untuk mengetahui performa dari LMSS menggunakan MLN dan LVS NAT dengan algoritma Round-Robin. Pengujian juga dilakukan untuk mengetahui hasil dari nilai rata – rata tiap parameter yang diberikan, diantaranya Throughput, Latency (delay), dan Jitter.
3. Tahap ketiga dilakukan pengujian Average Download Time (ADT). Parameter pengujian ini merupakan acuan dari jurnal referensi yang dipakai oleh penulis. Tujuannya agar mengetahui hasil kecepatan rata–rata waktu download ketika sejumlah client sedang melakukan pengunduhan paket repository mirror server secara bersama–sama.

### 4. Hasil dan Analisa

Tabel 1 menunjukkan hasil setelah melakukan pengujian sistem, telah dianalisa bahwa membangun Mirror Repository Linux dengan menggunakan LMSS dapat mengurangi penggunaan waktu dan sumber daya. Karena dengan menggunakan LMSS, Mirror server pada Cluster server dapat membuat Mirror server secara generate sehingga tidak memerlukan waktu yang lama dan tidak membutuhkan spesifikasi hardware khusus yang harus mengeluarkan banyak biaya dalam memasang aplikasi – aplikasi yang digunakan.

Begitu pula pada *Quality of Services (QoS)* dengan paramater *Throughput*, *Packet Loss*, *Delay*, dan *Jitter*, rancang bangun *Mirror Repository Linux* menggunakan LMSS terlihat lebih baik. Hasil dapat disimpulkan melalui Tabel 2.

Tabel 1. Hasil Pengujian Fungsional Sistem

Fungsional	Skenario		Keterangan
	1	2	
Ping server	√	√	Berhasil
Remote SSH server	√	√	Berhasil
Mirror Web Server	√	√	Berhasil
Instalasi Paket	√	√	Berhasil
Status Cluster Server		√	Berhasil
Status Mirror Server		√	Berhasil
Mengaktifkan Mirror Server		√	Berhasil
Melakukan Sinkronisasi File		√	Berhasil

Tabel 2. Hasil Pengujian QoS

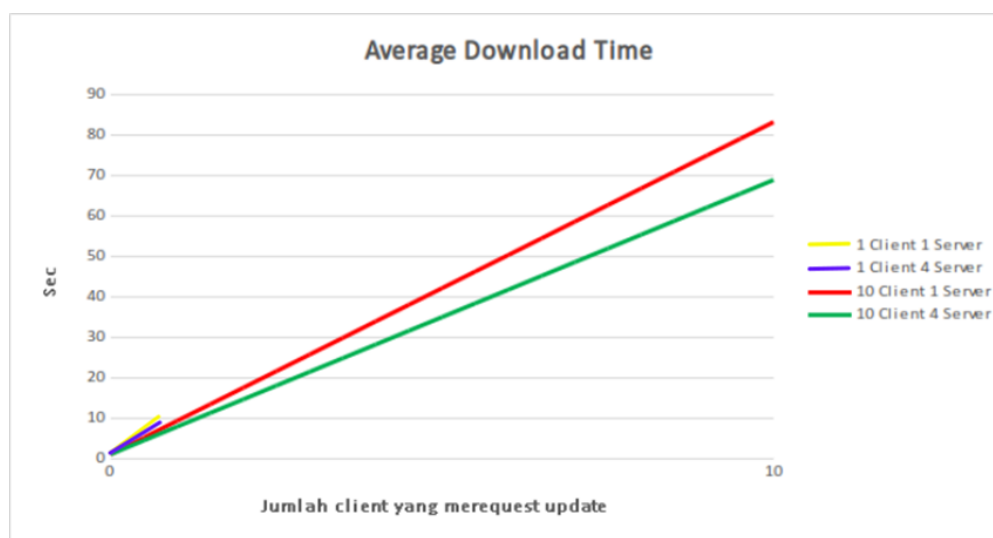
Skenario	Throughput (bps)	Packet Loss (%)	Delay (ms)	Jitter (ms)
1	12292456	0,000171887 %	0,000122985 ms	0,456 ms
2	12300901	0,000171889 %	0,000122901 ms	0,021 ms

Sedangkan *Average download Time* didapatkan hasil seperti pada Tabel 3.

Tabel 3. Hasil Pengujian Average Download Time

Kondisi	Average Download Time (s)
1	10 second
2	9,55 second
3	83,1 second
4	68,8 second

Dari hasil pengujian di atas, rata-rata waktu download oleh sebuah client pada LMSS (garis warna biru) berkurang 0,45 detik sedangkan dengan 10 client pada LMSS (garis hijau) berkurang 14,3 detik dari pada menggunakan single server (garis kuning dan merah) yang dapat dilihat Gambar 2.



Gambar 2. Grafik Average Download Time

#### 4. Kesimpulan

Dengan semua skenario dan kondisi pengujian yang telah dilakukan, dapat disimpulkan bahwa Mirror Repository Linux menggunakan LMSS dengan LVS–NAT dapat mengurangi rata–rata waktu download sebanyak 17,2 %. Sehingga penelitian ini dapat dijadikan perbandingan tipe penyeimbangan beban dengan menggunakan LVS-NAT dan LVS-Tunneling yang berdasar pada penelitian sebelumnya.

#### Referensi

- [1] D. Sudjiman, “Komputer Berbasis GNU/Linux dan Peluang Bisnisnya,” 2003.
- [2] T. C. Yen and P. Y. Tang, “Provision of load-balanced Linux packages mirror service in private cloud,” in *Proceeding - 2012 IEEE International Conference on Computational Intelligence and Cybernetics, CyberneticsCom 2012*, 2012.
- [3] S. W. Raharjo, J. A. Rabu, and J. Purwadi, “Implementasi Load Balancing Web Server Menggunakan Metode LVS-NAT,” *Informatika*, vol. 8, no. 2, pp. 169–180, 2012.
- [4] S. M. Pahlevi, “Komputasi Grid Dan Paralel,” *Agustus*, vol. 2008, pp. 6–715, 2008.
- [5] W. S. Raharjo and J. Purwadi, “Implementasi Load Menggunakan Balancing Web Server Metode LVS-NAT,” *INFORMATIKA*, vol. 8, no. 2, 2012.
- [6] Y. L. Oktavianus, “Membangun Sistem Cloud Computing Dengan Implementasi Load Balancing Dan Pengujian Algoritma Penjadwalan Linux Virtual Server Pada Ftp Server,” vol. 2, no. 1, 2013.
- [7] A. Arfriandi, “Perancangan, Implementasi, dan Analisis Kinerja Virtualisasi Server Menggunakan Proxmox, VMware ESX, dan Openstack,” *J. Teknol.*, vol. 5, no. 2, pp. 182–191, 2012.
- [8] K. Begnum, “Simplified cloud-oriented virtual machine management with MLN,” in *Journal of Supercomputing*, 2012.
- [9] S. K. Rachman, Ambia Haryadi, “LoadBalancing menggunakan LVS via NAT pada PT. Bank Mega Syariah.”
- [10] W. Zhang, “Linux virtual server for scalable network services,” *Ottawa Linux Symp.*, 2000.