

SISTEM PENYEIMBANG BEBAN WEB SERVER DENGAN IPTABLES

Andy Purnama Nurhatta^{*)}, Adian Fatchur Rochim, and R Rizal Isnanto

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

^{*)}E-mail: and1delc@gmail.com

Abstrak

Saat ini Internet berkembang dengan sangat pesat, seiring dengan semakin banyaknya pengguna yang terhubung pada jaringan internet. Ketika sebuah server tunggal mendapatkan request dari banyak pengguna, besar kemungkinan akan terjadi kelebihan beban dan crash sehingga request tidak dapat dilayani oleh single server. Arsitektur cluster yang diterapkan sebagai server dengan kinerja tinggi adalah salah satu solusi yang efektif dan efisien untuk mengatasi masalah tersebut. Tujuan dari penelitian ini adalah untuk arsitektur cluster menggunakan konsep jaringan penyeimbang beban yang memungkinkan proses pengolahan data dibagi secara terdistribusi ke beberapa server. IPTables adalah sebuah firewall pada mesin server Linux. Pada penelitian ini, firewall dimanfaatkan untuk meneruskan koneksi jaringan server web ke banyak titik, yang dapat digunakan untuk membangun layanan yang skalabilitas tinggi. Skalabilitas dicapai dengan secara transparan menambah atau mengurangi node pada cluster. Round robin merupakan salah satu metode penyeimbangan beban yang telah diketahui. Algoritma round robin mendistribusikan beban secara seimbang dengan menggunakan sistem antrian. Penggunaan penyeimbang beban pada web server dapat berjalan baik pada situs <http://hits4.undip.ac.id> untuk info pengumuman SNMPTN. Penyeimbang beban meneruskan koneksi berdasarkan algoritma Round Robin dan membuat layanan paralel pada setiap Web Server. Penelitian ini diuji berdasarkan kemampuannya dari segi permintaan http, kemampuan lebar data, penggunaan CPU, maupun penggunaan memori serta waktu respon dan throughput. Penelitian ini juga menggunakan web control panel Kloxo yang memudahkan webmaster mengatur penggunaan web server.

Kata kunci: Skalabilitas tinggi, Penyeimbang Beban, Round Robin, Kloxo

Abstract

Currently the Internet is growing very rapidly, as more and more users are connected to the Internet network. When a single server get requests from many users, most likely will happen is overload and crash, so the request can not be served by a single server. Cluster architecture is implemented as a server with high performance is one of effective and efficient solutions to resolve the issue. The purpose of this research is cluster architecture uses the concept of network load balancer that allows the distributed data processing is divided into multiple servers. At the end of this task, the authors use this firewall to forward web server network connections to multiple nodes, which can be used to build high scalable of services. Scalability is achieved by transparently adding or subtracting nodes in the cluster. Round robin is a load balancing methods are known. Round robin algorithm to distribute the load is balanced by using a queuing system. The use of the load balancer can run on the site for SNMPTN announcement in <http://hits4.undip.ac.id>. Load balancer forward connections based on the algorithm Round Robin and makes parallel services on each Web Server. This study tested based on their ability in terms of http requests, bandwidth capability, CPU usage, and memory usage. This study also uses the web control panel Kloxo that allows webmasters use of the web server easier.

Keywords: High Availability, Load Balancing, IPTables, Kloxo

1. Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi komputer dewasa ini sangat pesat, terlebih lagi ketika ditemukannya teknologi jaringan komputer dan Internet. Seiring dengan perkembangan Internet, *web server* bermunculan untuk

mendukung kebutuhan akan informasi yang bersifat *real-time* dan *online*. Serta, banyak perusahaan yang membutuhkan *website* untuk berbagai tujuan.

Teknologi *web server* dengan menggunakan *single server* ini ternyata masih belum bisa memenuhi kebutuhan jaringan dengan maksimal. Ketika permintaan dari klien meningkat secara drastis, *single server* tersebut tidak

mampu menangani permintaan dari klien, sehingga yang terjadi *web server* tersebut menjadi *overload* dan *crash*. Masalah seperti inilah yang akhirnya menimbulkan suatu ide bagaimana membuat suatu jaringan yang *scalable* dan *high-available* dan akhirnya ditemukan teknologi *load balancing*.

Teknologi penyeimbang beban ini merupakan sebuah teknologi untuk menangani kebutuhan beban jaringan yang terus meningkat drastis. Salah satu jenis teknologi penyeimbang beban yang ada adalah dengan IPTables.

IPTables adalah sebuah *firewall* pada mesin *server* Linux. Pada penelitian ini, digunakan *firewall* untuk meneruskan koneksi jaringan *server web* ke banyak *node*, yang digunakan untuk membangun layanan yang *high-scalability* dan *high-availability*. *Scalability* dicapai dengan secara transparan menambah atau mengurangi *node* pada *cluster*. *High availability* ditunjukkan dengan mendeteksi *node* yang gagal dan mengkonfigurasi ulang sistem dengan tepat.

1.2 Tujuan

1. Mendesain arsitektur *cluster* dengan menggunakan perangkat lunak IPTables.
2. Memahami kinerja dari *web server* dan konsumsi memori yang dipakai.
3. Mempelajari mekanisme kerja dari IPTables sebagai *firewall* untuk melakukan pemetaan alamat dan *port* yang ada ke jaringan yang lain dan memahami mekanisme *round robin* yang bisa dilakukan oleh IPTables.
4. Memahami manfaat penggunaan Kloxoxo sebagai *web control panel*.

1.3 Batasan Masalah

Penulisan Penelitian ini memiliki batasan pada permasalahan berikut :

1. Arsitektur *cluster* diimplementasi pada jaringan dengan pengalamatan IPv4.
2. Mesin penyeimbang beban menggunakan sistem operasi Ubuntu 10.04 LTS dan translasi maupun penerusan paket dilakukan oleh IPTables.
3. *Web server* yang digunakan adalah mesin Apache dan *web control panel*. *web control panel* menggunakan perangkat lunak Kloxoxo.
4. Tidak membahas mengenai basisdata.
5. Tidak membahas keamanan jaringan.

2. Metode

Perancangan ini mendesain 2 model arsitektur sistem. Perbedaan antara kedua arsitektur adalah kemudahan perawatan *web* pada arsitektur kedua. Arsitektur pertama

menggunakan Apache sebagai *web server*, *mysql* sebagai *server* basis data, dan NFS(Network Files System) untuk sinkronisasi kedua *web server*. Arsitektur kedua menggunakan Kloxoxo *web control panel*.

Dibutuhkan empat buah *server* pada arsitektur pertama, yaitu *server gateway* dengan IPTables, dua *web server*, dan sebuah NFS(Network Files System) *server*. *Server gateway* merupakan *server* yang berfungsi sebagai objek yang di-*hit* dan penyeimbang beban. *Server* NFS(Network Files System) untuk mensinkronisasi kedua file *web server*. Alasan pemisahan NFS(Network Files System) *server* pada *server* yang berbeda adalah beratnya beban pada NFS(Network Files System). Perancangan arsitektur ini ada pada gambar 1 berikut.



Gambar 1. Perancangan penyeimbang beban *web server* dengan *cluster* NFS

Perancangan penyeimbang beban *web server* pada Gambar 1. terdiri dari komponen sebagai berikut:

- a. *Gateway/Balancer*, berfungsi sebagai *gateway* dari jaringan LAN (*Local Area Network*) maupun WAN(*Wide Area Network*) sekaligus sebagai sebuah *server* yang meneruskan paket dari klien ke *real server* dengan teknologi NAT(*Network Address Translation*). Klien akan menganggap *Balancer* ini seolah-olah merupakan *server tunggal*.
- b. *HTTP Accelerator*, berfungsi sebagai *reverse proxy*, yang mana digunakan sebagai *cache* data statis *web* di depan *web server* sehingga mempercepat koneksi dari klien yang *request* ke *web server*.
- c. *Web Server*, berfungsi sebagai *server* yang melayani permintaan klien, dalam hal ini memberi layanan http melalui *port* 80.
- d. NFS (*Network File System*) *Server*, berfungsi sebagai sinkronisasi data antara *web server* 1 dan *web server* 2, sehingga saat kita meng-*update* salah satu *web server*, *web server* lain akan ikut ter-*update*.

Digunakan Kloxoxo sebagai *web control panel* pada arsitektur kedua. Kelebihan menggunakan Kloxoxo adalah kemampuannya mengorganisir banyak *web* dalam satu *server* dan tampilannya yang sudah menggunakan GUI(*Graphical User Interface*) yang memudahkan pengguna dalam manajemen *web* yang ada. Digunakan

tiga server dengan sebuah server untuk gateway dan dua buah server untuk web yang berisi Kloxox. Perancangan arsitektur kedua ada pada gambar 2. berikut.



Gambar 2. Perancangan penyeimbang beban web server dengan cluster kloxox

Gambar 2. menjelaskan ada dua buah komponen pada arsitektur kedua. Pertama, gateway yang difungsikan sama dengan arsitektur sebelumnya. Kedua, Kloxox digunakan sebagai web control panel pada 2 back-end web server.

$$f_2(P_m) = \begin{cases} 1, & P_m \leq \frac{P_{amb}}{2} \\ \frac{2}{P_{amb}} \sqrt{P_m P_{amb} - P_m^2}, & \frac{P_{amb}}{2} \leq P_m \leq P_{amb} \\ -\frac{2}{P_m} \sqrt{P_m P_{amb} - P_m^2}, & P_{amb} \leq P_m \leq 2P_{amb} \\ -1, & 2P_m \geq 2P_{amb} \end{cases}$$

P_m = manifold pressure (bar)

P_{amb} = ambient (atmospheric) pressure (bar), 1 bar

3. Hasil dan Analisa

Implementasi dilakukan dengan membuat layanan-layanan seperti pada perancangan. Dimulai dengan penanaman sistem operasi pada masing-masing server sampai dengan implementasi aplikasi-aplikasi yang diperlukan oleh masing-masing server. Setelah semua paket telah terpasang, maka hal terpenting yang harus dilakukan adalah melakukan konfigurasi agar semua proses proses dapat berjalan dengan performa yang baik.

3.1 Implementasi dan Konfigurasi IPTables

IPTables mengizinkan administrator untuk mengontrol sepenuhnya jaringan melalui paket IP dengan sistem operasi Linux. Sebuah kebijakan atau policy dapat dibuat dengan IPTables sebagai polisi lalu lintas jaringan. Sebuah policy pada IPTables dibuat berdasarkan sekumpulan

peraturan untuk mengatur setiap paket yang datang. Pada IPTables ada istilah ipchain. Ipchain merupakan daftar aturan bawaan dalam IPTables. Ketiga chain tersebut adalah INPUT, OUTPUT dan FORWARD.

3.2 Implementasi dan Konfigurasi Web Server

Perlu diinstall Apache web server, php5, php5-dev, dan Mysql server. Apache web server bertugas menerima dan membalas permintaan situs yang datang dari klien di web server. Php5 berfungsi mendukung bahasa pemrograman php pada web server. Php5-dev digunakan sebagai syarat implementasi PHP Accelerator dengan perangkat lunak eAccelerator. Mysql server sebagai basis data berfungsi sebagai syarat mengimplementasi Joomla dan Wordpress.

3.3 Implementasi dan Konfigurasi Kloxox sebagai Web Server

Alasan penggunaan Kloxox sebagai web server adalah kemampuannya mengatur web server secara GUI(Graphical User Interface) yang dapat dilakukan melalui browser yang secara default ada pada port 7777 untuk https dan 7778 untuk http. Kloxox ini stabil pada sistem operasi CentOS 5.x dan dalam hal ini menggunakan CentOS 5.8.

Kemudian setelah selesai, kita membuka browser dan membuka Kloxox dengan menggunakan port 7777 atau port 7778 dengan login sebagai 'admin' dan password 'admin'.

Setelah masuk ke halaman utama, kita akan diminta untuk mengganti password default kloxox tadi untuk alasan keamanan.

Kemudian kita akan diminta mengatur sistem keamanan lain bawaan dari kloxox yaitu lsguard. Lsguard ini melindungi server dari serangan brute force dengan memonitor pesan log ssh dan ftp dan memblokir alamat IP yang berkali-kali gagal login ke server. Lsguard secara default aktif dan tidak dapat dimatikan. Kita mengatur lsguard dengan mengatur berapa banyak kita boleh gagal dan alamat IP whitelist. Jika alamat IP ditemukan di whitelist, alamat tersebut tidak akan pernah diblok sebanyak apapun gagal mencoba login.

Kemampuan lain dari kloxox adalah mengoptimasi performa dari web server dengan mengatur beberapa aspek. Aspek yang pertama adalah tipe PHP bekerja, optimasi penggunaan memori, konversi MySQL, dan pengaturan chmod untuk setiap folder dan file pada direktori home yang akan diakses user.

Selain itu, kita dapat menambahkan alamat IP pada server sesuka kita dengan menambahkan subinterface pada interface yang ada.

Tujuan dari penambahan IP di atas adalah untuk penambahan domain jadi kita dapat menambah domain sebanyak yang kita mau dan sesuai alamat IP yang tersedia. Kemudian kita mengatur konfigurasi DNS sehingga domain maupun subdomain dari web server ini dapat terbaca dengan baik oleh user.

Kita dapat mengatur nama template DNS, alamat web, alamat mail dan alamat DNS utama. Setelah menambah alamat IP dan DNS barulah kita dapat membuat domain atau subdomain yang kita mau. Setiap domain dan subdomain ini dibedakan pada folder yang berbeda pada masing-masing login pengguna, kemudian dialiaskan oleh apache sehingga dalam satu server kita dapat mengatur berbagai situs. Jadi tidak mungkin suatu domain dengan domain yang lain bertumpukan. Untuk menambahkan domain kita mengatur nama domain dan dokumen root masing-masing situs.

Setelah menambahkan domain barulah kita mengatur di mana domain tersebut akan ditempatkan pada alamat IP yang kita tambahkan. Kelebihan lain dari kloxo adalah bersifat gratis dan dapat dijalankan pada server lokal tanpa harus terhubung ke internet. Barulah server kita siap dibaca oleh user.

3.4 Hasil pengujian dari penyeimbang beban web server

Dengan arsitektur IPTables, LAMP, dan NFS dilakukan pengujian dengan server SNMPTN Universitas Diponegoro 2012 untuk pengumuman hasil ujian SNMPTN dengan hostname <http://hits4.undip.ac.id/> dengan tampilan halaman web pada gambar 3.



Gambar 3. Tampilan <http://hits4.undip.ac.id/>

Dari hasil pengujian, penggunaan lebar jalur data sudah mencapai maksimal, karena pada jaringan web server digunakan lebar jalur data maksimal sebesar 60 Mbps. Walaupun permintaan dari pengguna cukup besar, yang diakumulasikan pada besarnya penggunaan lebar jalur data, kedua web server tersebut masih berjalan stabil dan

dapat diakses pengguna dengan lancar. Dengan demikian, penggunaan penyeimbang beban web server pada arsitektur pertama dapat berjalan dengan baik.

Sedangkan arsitektur kedua memiliki kelebihan pengaturan web, basis data melalui browser. Arsitektur kedua dapat memanajemen web dengan lebih mudah tanpa harus mengakses terminal server web.

3.5 Hasil pengujian dari penyeimbang beban web server menggunakan httpperf dan webbench

Pengujian dilakukan ke dua arsitektur, yaitu arsitektur server tunggal dan sistem penyeimbang beban menggunakan IPTables. Perangkat pengujian yang digunakan adalah httpperf dan webbench. Httpperf dan webbench digunakan untuk workload generator pada sistem penyeimbang beban maupun server tunggal. Httpperf melakukan koneksi ke server tunggal dan sistem penyeimbang beban membentuk 40000 koneksi dengan 4000 koneksi per detik. Webench melakukan koneksi ke server tunggal dan sistem penyeimbang beban membentuk 2000 dalam waktu 30 detik. Dari pengujian httpperf dan webench didapatkan hasil sebagai berikut.

Tabel 1 Hasil pengujian httpperf pada mesin tunggal dan sistem penyeimbang beban IPTables

Arsitektur	Rata Koneksi	Rata permintaan	Waktu Respon (ms)	Through put
Tunggal Arsitektur Pertama	128.8 conn/s	58.6 req/s	2757.8	22.6 KB/s
Tunggal Arsitektur Kedua	148.1 conn/s	81.1 req/s	2354.1	153.3 KB/s
Sistem Penyeimbang Beban Arsitektur Pertama	173.4 conn/s	100.1 req/s	2600.8	36.1 KB/s
Sistem Penyeimbang Beban Arsitektur Kedua	221.4 conn/s	112.9 req/s	2243.9	321.1 KB/s
Sistem Penyeimbang Beban Gabungan	261.3 conn/s	168.1 req/s	1549.2	291.8 KB/s

Tabel 1 menunjukkan peningkatan kinerja yang lebih baik dengan penambahan jumlah server yang ada dilihat dari koneksi dan permintaan yang dilayani serta throughput yang lebih besar. Dengan peningkatan kemampuan yang ada, sistem dapat melayani pengguna lebih baik dan lebih cepat. Throughput di sini berdasarkan penjumlahan kemampuan dari beberapa web server yang ada. Perbandingan kemampuan web server arsitektur pertama dan kedua mengalami perbedaan karena pada arsitektur kedua, kloxo mampu mengoptimalkan kemampuan web server sehingga kinerjanya meningkat.

Tabel 2 Hasil pengujian webbench pada mesin tunggal dan sistem penyeimbang beban IPTables

Arsitektur	Pages/min	Throughput
Tunggal Arsitektur Pertama	12158	200241 bytes/sec
Tunggal Arsitektur Kedua	9192	205638 bytes/sec
Sistem Penyeimbang Beban Arsitektur Pertama	16640	400563 bytes/sec
Sistem Penyeimbang Beban Arsitektur Kedua	17610	450580 bytes/sec
Sistem Penyeimbang Beban Gabungan	25946	814267 bytes/sec

Tabel 2 menunjukkan dua variabel yang ditampilkan. Pages/min menunjukkan kemampuan sistem penyeimbang beban dan server tunggal melayani jumlah halaman tiap menit. Halaman yang dapat dilayani dapat meningkat cukup signifikan. *Throughput* mengalami peningkatan pada sistem penyeimbang beban dibandingkan server tunggal.

4. Kesimpulan

1. Penggunaan penyeimbang beban pada *web server* pada arsitektur pertama dapat berjalan baik pada situs <http://hits4.undip.ac.id> untuk info pengumuman SNMPTN
2. Penggunaan memori dan CPU pada arsitektur pertama tidak terlalu membebani *server* walaupun permintaan cukup besar dapat dilihat pada trafik yang ada pada *web server*
3. Mekanisme kerja dari IPTables pada penelitian ini bekerja berdasarkan algoritma *round robin* yang membagi beban berdasarkan nomor kedatangan permintaan http.
4. Konfigurasi situs *web* dapat dengan mudah menggunakan GUI (*Graphical User Interface*) sehingga dapat memberikan kesempatan bagi yang kurang mengerti penggunaan situs secara rinci untuk mengatur situs secara lebih mudah dengan menggunakan Kloxo.
5. Terjadi peningkatan kinerja pada sistem penyeimbang beban dengan arsitektur kloxo dibandingkan server tunggal. Hal ini terlihat pada parameter throughput dan waktu respon

Referensi

[1]. Kusuma, P.H., *Implementasi Penyedia Layanan Terdistribusi Berbasis Linux dengan Konsep Penyeimbang Beban Jaringan*. Skripsi S-1, Institut Teknologi Telkom, Bandung, 2009.

[2]. Mack, J. *LVS-mini-HOWTO*, <http://www.austintek.com/LVS/LVS-HOWTO/>, 2006.

[3]. Nasution, A.H., *Komparasi Algoritma Penjadwalan Pada Layanan Terdistribusi Load Balancing LVS Via NAT*, Institut Teknologi Surabaya, 2011.

[4]. O'Rourke, P and Mike Keefe, *Performance Evaluation of Linux Virtual Server*, Mission Critical Linux, Inc, Massachusetts, 2001.

[5]. Putera, R, AF Rochim, dan Y Christiyono. *Sistem Otentikasi Terpusat Berbasis Lightweight Directory Access Protocol*. Universitas Diponegoro Semarang, 2011.

[6]. Sagala, Muhfi Asbin, *Implementasi Load Balancing Pada Web Server*. Skripsi S-1, Universitas Sumatera Utara, Medan, 2010.

[7]. Teo, Y.M., dan Rassul Ayani, *Comparison of Load balancing Strategies on Cluster-based Web Servers*, Fujitsu Computer (Pte) Ltd & National University of Singapore, Singapore, 2001.

[8]. Yulyanti, *Reverse Proxy*, <http://cheatlinknote.blogspot.com/2011/12/reverse-proxy.html>, 17 April 2012.

[9]. Zhang, W. *Linux Virtual Server for Scalable Network Services*. National Laboratory for Parallel & Distributed Processing, Changsa, Hunan, 2000.

[10]. ---, Kloxo, <http://lxcenter.org/software/kloxo>, 17 April 2012.

[11]. ---, *Lebih Lanjut Mengenai Reverse Proxy*, <http://teknologi.kompasiana.com/internet/2010/01/18/lebih-lanjut-mengenai-reverse-proxy/>, 17 April 2012.