PROYEK AKHIR

KOMPARASI ALGORITMA PENJADWALAN PADA LAYANAN TERDISTRIBUSI LOAD BALANCING LVS VIA NAT

ABDUL HARIS NASUTION
NRP. 7406 040 036

Dosen Pembimbing :
ISBAT UZZIN N. S.Kom
NIP. 197405052003121002

JURUSAN TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
S U R A B A Y A 2011

PROYEK AKHIR

KOMPARASI ALGORITMA PENJADWALAN PADA LAYANAN TERDISTRIBUSI LOAD BALANCING LVS VIA NAT

Abdul Haris Nasution
NRP. 7406.040.036

Dosen Pembimbing:
Isbat Uzzin Nadlori, S.Kom, MT
NIP. 197405052003121002

JURUSAN TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2011

KOMPARASI ALGORITMA PENJADWALAN PADA LAYANAN TERDISTRIBUSI *LOAD BALANCING* LVS *VIA* NAT

Oleh :
Abdul Haris Nasution
7406.040.036

Proyek Akhir ini Digunakan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Sain Terapan (S.ST)
di
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember Surabaya
Januari 2011
Disetujui oleh :

Tim Penguji Proyek Akhir :	Dosen Pembimbing :
1.	1.
Nur Rosyid Mubtada'i, S.Kom	**Isbat Uzzin Nadlori, S.Kom, MT**
NIP. 197407182001121001	**NIP. 197405052003121002**
2.	
Ahmad Syauqi Ahsan, S.Kom	
NIP. 197505302003121001	
3.	

Setiawardhana, ST, MT
NIP. 197708242005011001

Megetahui,
Ketua Jurusan Teknik Informatika

Arna Fariza, S.Kom, M.Kom
NIP. 197107081999032001

ABSTRAK

Saat ini internet berkembang dengan sangat pesat, seiring dengan semakin banyaknya *user* yang terhubung pada jaringan internet. Ketika sebuah *single server* mendapatkan *request* dari banyak *user*, besar kemungkinan akan terjadi *overload* dan *crash* sehingga *request* tidak dapat dilayani oleh *single server*.

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* dan *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 beberapa algoritma penjadwalan yang dapat mempengaruhi kinerja sistem LVS, performansi tiap algoritma tersebut dapat diamati dengan membandingakan antar algoritma terhadap parameter meliputi *throughput, request loss, CPU utilization*, dan waktu respon sehingga didapatkan algoritma penjadwalan terbaik pada implementasi *load balancing* LVS via NAT.

Kata Kunci: ***Load Balancing, Linux Virtual Server, NAT.***

ABSTRACT

Nowadays, the Internet growth is very fast, we can see that fact from many user that connected to the network. When a single server getting request from many user that is likely to occur overload and crash, so the request can not be served by a single server.

One of effective and efficient solutions to resolve that problem is system clustering. System cluster can be built using the concept of network load balancing and high-availability that enables data processing distributed to several computers, we can use *linux virtual server*.

Linux virtual server has several scheduling algorithms that can affect the performance of LVS system, performance of each algorithm can be observed by comparing between algorithms with some parameters such as *throughput*, *request loss*, CPU Utilization and *response time* to obtain the best scheduling algorithm in the implementation of load balancing LVS via NAT.

Keywords: ***Load Balancing, Linux Virtual Server, NAT*****.**

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya yang sangat besar sehingga penulis dapat menyelesaikan tugas akhir dengan judul :

KOMPARASI ALGORITMA PENJADWALAN PADA LAYANAN TERDISTRIBUSI LOAD BALANCING LVS VIA NAT

Buku tugas akhir ini disusun sebagai salah satu syarat akademik menyelesaikan studi pada program pendidikan diploma empat (D IV) pada Jurusan Teknik Informatika, Fakultas Politeknik Elektronika Negri Surabaya, Institut Teknologi Sepuluh Nopember Surabaya.

Pada Proyek Akhir ini, penulis membahas tentang sistem *cluster,* khususnya *load balancing* LVS *via* NAT, dimana pada LVS terdapat beberapa algoritma yang dapat digunakan dan memiliki performansi yang berbeda pula. Untuk itu penulis mencoba memandingkan antar algoritma yang ada guna mendapatkan algoritma terbaik berdasarkan parameter yang telah ditentukan sebelumnya yaitu; *throughput*, *request loss*, *CPU utilization* dan *time response*.

Dalam penyusunan buku tugas akhir ini penulis mengambil referensi dari beberapa sumber seperti teori - teori yang yang telah penulis peroleh dari perkuliahan, membaca literatur, serta bimbingan dari dosen pembimbing.

Akhirnya penulis menyadari bahwa masih banyak kekurangan dan kelemahan dalam buku ini. Oleh karena itu, penulis mengharapkan saran, kritik yang membangun, dan koreksi yang konstruktif untuk perkembangan lebih lanjut.

Semoga buku tugas akhir ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan dan bagi semua pihak pada umumnya serta bagi penulis sendiri pada khususnya.

Surabaya, 20 Januari 2011

Penulis

UCAPAN TERIMAKASIH

Segala puji dan syukur atas limpahan rahmat dan hidayah Allah SWT sehingga penulis dapat menyelesaikan penyusunan buku tugas akhir ini. Penulis menyadari dalam penyelesaian buku tugas akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu dengan kerendahan hati penulis mengucapkan terima kasih dan mempersembahkan buku tugas akhir ini kepada:

1. **ALLAH SWT**. Atas segala limpahan rahmat, hidayah, serta karunia NYA sehingga penulis dapat menyelesaikan proyek akhir ini.
2. Kedua orang tua tercinta, Ayahanda **Alm. J. Nasution** dan Ibunda **M. Hasibuan**, Kedua Kakanda, **Nurhamso Nasution** dan **Nurhaini Nasution**, yang selalu memberikan kepercayaan, motivasi dan dukungan penuh kepada penulis untuk menyelesaikan studi dan tugas akhir ini.
3. **Bapak Ir. Dadet Pramadihanto, M.Eng, Ph.D**, selaku Direktur Politeknik Elektronika Negeri Surabaya - ITS.
4. **Ibu Arna Fariza, S.Kom, M.Kom,** selaku Ketua Jurusan Teknologi Informasi Politeknik Elektronika Negeri Surabaya - ITS.
5. **Bapak Isbat Uzzin N, S.Kom, MT**, selaku dosen pembimbing yang telah memberikan bimbingan dan arahannya selama pengerjaan proyek akhir ini.
6. **Annisa Nur Muslimah**, atas semua dukungan yang pernah diberikan, *thank you*.. ☺
7. Teman-teman seperjuangan, *Amik, Agenk, Hakim, Yanur, Pram, Ifan, Aconk, Heri, Ipul, let's keep move on guys*.. ☺ . Teman-teman kost, *Ngawi, Rahmat, Arifin* dan lainya, *pokoke suwun ae yo rek*.. Dan teman-teman dari dunia maya, *mas bos Arif Fayantory, Ai* and *Mia* who make me laughing all the night.. lol.. ☺
8. Semua pihak yang tidak bisa penulis tuliskan di halaman ini satu per-satu, yang telah membantu dalam menyelesaikan proyek akhir ini.

Semoga Allah SWT membalas semua kabaikan–kebaikan yang telah kalian berikan. Amiin..

DAFTAR ISI

DAFTAR GAMBAR

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB I
PENDAHULUAN

1.1 LATAR BELAKANG

World Wide Web (WWW) telah menjadi standar *de facto* sebagai infrastruktur dasar untuk beragam layanan internet. Banyak sekali perusahaan dan organisasi yang membutuhkan website untuk beberapa tujuan seperi promosi produk, bisnis e-commerce, forum online dan layanan aplikasi web lainnya.

Single web server yang mayoritas digunakan oleh perusahaan tersebut untuk layanan berbasis web memiliki beberapa kekurangan yang tergolong fatal. Ketika website mendapatkan *request* yang banyak dimana *request* tersebut tidak bisa ditangani oleh web server, maka dampak terburuk adalah terjadi *overload* dan *crash*.

High-end Server seperti IBM Mainframe, BlueGene/L maupun Sun Fire yang harganya sangat mahal adalah salah satu solusi untuk menjawab masalah *overload*, namun hanya perusahaan-perusahan besar biasanya yang dapat dan mampu mengeluarkan budget yang besar untuk penyediaan *High-end* server. Dan bagaimana dengan perusahaan kecil dan organisasi yang tidak memiliki dana besar?. Masalah seperti inilah yang menjadi alasan penting untuk diterapkan suatu teknologi *network load balancing*.

Network Load Balancing merupakan sistem yang diharapkan dapat menangani beban simultan yang besar dengan kemungkinan kegagalan yang sangat kecil. Salah satu jenis sistem *network load balancing* yang bekerja pada layer empat yaitu Linux Virtual Server (LVS).

1.2 PERUMUSAN MASALAH

Untuk performansi suatu sistem cluster yang ingin dibangun, pemilihan alogritma penjadwalan yang tepat merupakan salah satu faktor penting yang perlu untuk diperhatikan. Dari sekian banyak algoritma yang ada tersebut, maka pada tugas akhir ini akan dicoba mengkomparasi performa sistem cluster saat menggunakan algoritma penjadwalan yang berbeda, berdasarkan parameter yang telah ditentukan, untuk menemukan mana algoritma yang mempunyai performa paling

baik.

Beberapa jenis algoritma penjadwalan yang dapat diterapkan pada sistem linux virtual server pada proses distribusi request kepada real server, antara lain yaitu : *Round Robin (rr), Weighted Round Robin (wrr), Least Connection (lc), Weighted Least Connection (wlc), Locality Based Least Connection (lblc), Locality Based Least Connection with Replication (lblcr), Destination Hashing (dh), Source Hashing (sh), Shortest Expected Delay Scheduling (sed), Never Queue Scheduling (nq).*

1.3 BATASAN MASALAH

- Test performansi algoritma penjadwalan hanya pada protokol http (80) / apache webserver.
- Diimplementasikan pada jaringan dengan pengalamatan Ipv4
- Tidak membahas keamanan jaringan
- Mengkomparasi ke 10 Algoritma penjadwalan
- Mengabaikan proses sinkronisasi data antar real server.
- Mesin load balancing digunakan pada PC, di LAN via NAT

1.4 TUJUAN

Menemukan algoritma penjadwalan terbaik pada sistem cluster menggunakan LVS load balancing via NAT, berdasarkan parameter tertentu.

1.5 METODOLOGI

Adapun metodologi yang dilakukan dalam pembuatan proyek akhir ini sebagai berikut:

- **Studi pustaka dan literatur**
  Pada tahap ini penulis akan mempelajari cara peng-*compile*-an kernel linux, administrasi paket load balancing (ipvsadm), LVS literatur dan lainnya, yang bersumber pada buku-buku dan dari internet.
- **Pembangunan sistem dan implementasi LVS**
  Pada tahap ini akan dilakukan pembangunan dan implementasi mesin *load balancing* dengan menggunakan LVS via NAT pada sistem operasi linux.

- **Mengkomparasi LVS dengan menggunakan bergabagai algoritma yang ada**
  Mengkomparasi dan mengevaluasi berbagai algoritma LVS berdasarkan parameter-parameter yang telah ditentukan untuk menemukan algoritma terbaik pada LVS via NAT
- **Penulisan proyek akhir**

1.6 SISTEMATIKA PENULISAN

Sistematika pembahasan dari penyusunan proyek akhir ni direncanakan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang pendahuluan yang terdiri dari latar belakang, permasalahan, batasan masalah, maksud dan tujuan, metodologi pada pembuatan proyek akhir serta sistematika pembahasan dari proyek akhir ini.

BAB II TEORI PENUNJANG

Bab ini membahas mengenai teori-teori yang berkaitan dan menunjang dalam penyelesaian proyek akhir ini.

BAB III PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi perancangan dan pembuatan sistem Linux Virtual Server (LVS).

BAB IV PENGUJIAN DAN ANALISA

Bab ini berisi tentang pengujian dan analisa dari penggunaan berbagai algoritma yang ada terhadap sistem LVS yang telah dibangun berdasarkan parameter-parameter yang telah ditentukan.

.

BAB V PENUTUP

Bab ini berisi kesimpulan dari pembuatan proyek akhir ini. Untuk lebih meningkatkan hasil akhir yang lebih baik maka diberikan juga saran-saran untuk perbaikan serta penyempurnaan proyek akhir ini.

Halaman ini sengaja dikosongkan

BAB II
TEORI PENUNJANG

Dalam bab ini akan dibahas mengenai teori-teori penting yang dapat menunjang dan menjadi acuan dalam pembuatan proyek akhir. Bagian tersebut meliputi penjelasan mengenai pengenalan sistem operasi linux, dasar sistem cluster, linux virtual server, network address translation dan algoritma penjadwalan.

2.1 Pengenalan Sistem Operasi Linux

Linux diciptakan oleh Linus Trovalds pada tahun 1991, kemudian dikembangkan bersama oleh jutaan orang di seluruh dunia sehingga menjadi sistem operasi yang berkembang sangat pesat sampai saat ini. Sebagai sistem operasi yang notabene berakar dari *Unix* maka *Linux* sangat stabil sebagai *server*. Dengan ketersediaan *source code linux* maupun program-program yang berlisensi GPL (*General Public Lisence*), setiap orang memiliki keleluasaan tinggi untuk melakukan optimisasi terhadap *server linux* yang dibuat sehingga tidak ada ketergantungan kepada suatu *vendor* tertentu.

Linux mampu berjalan pada berbagai jenis arsitektur komputer seperti *x86*, *PowerPC*, *Sun SPARC*, *MIPS*, dan lain sebagainya. Berbagai aplikasi seperti *web server*, *firewall*, dan *database server* juga bisa berjalan dengan baik pada *platform linux*. Kehandalan *linux* pada beberapa waktu terakhir juga ditambah dengan kemampuan untuk menjadi *node* pada sistem *cluster*. Sistem *high-availability* (HA) yang hampir menjadi keharusan untuk perusahaan-perusahaan besar juga telah tersedia perangkat lunaknya.

2.2 Dasar Sistem Cluster

2.2.1 Pengertian Sistem Cluster

Dalam dunia komputasi, *cluster*ing adalah gabungan dari beberapa komputer yang bekerja sama untuk menghasilkan *output* dari masalah tertentu yang diberikan (*input*) dimana *developer* sistem tersebut diharuskan membagi kerja ke setiap anggota gabungan komputer secara manual.

Secara umum, salah satu karakteristik utama komputer *cluster* adalah konsep *single entity* dimana kumpulan banyak komputer yang menjadi komputer *cluster* dipandang sebagai satu kesatuan sistem tunggal. Hal ini berarti bahwa *developer* tidak perlu lagi mengatur pembagian proses yang diberikan untuk didistribusikan ke masing-masing mesin dalam sistem *cluster* tersebut, tetapi *developer* cukup mendapatkan gambaran bahwa hasil dari proses yang diberikan oleh *user* dikerjakan oleh *single* sistem. Bagaimana pun juga, *developer* perlu mengetahui informasi pengolahan data oleh sistem *cluster* tersebut.
Dari definisi umum di atas, banyak sekali macam konfigurasi untuk sistem *cluster*. Secara garis besar terdapat dua kelompok yaitu :

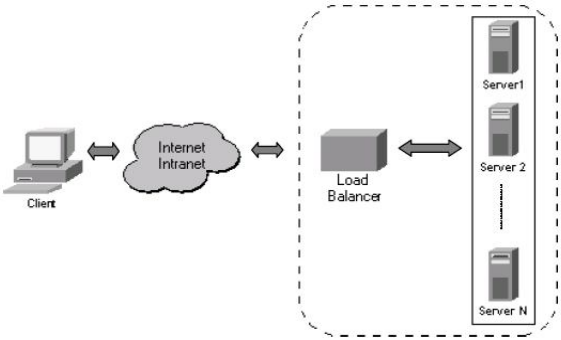
a. *Load balancing*
b. *Parallel computing*

2.2.2 Load Balancing

Load balancing adalah suatu metode untuk mendistribusikan beban kepada beberapa *host* sehingga beban kerja menjadi lebih ringan. Ini bertujuan agar waktu rata-rata mengerjakan tugas menjadi singkat dan dapat menaikkan utilitas prosesor.

Load balancing dapat diimplementasikan dengan *hardware* khusus, *software* maupun gabungan keduanya. Konfigurasi standar yang ada memberi gambaran bahwa satu mesin ditempatkan diantara *client* dan *server*, mesin ini disebut sebagai *director* karena tugasnya adalah memberikan balancing pada *request* dari *client* ke *server*.

Sebuah *load balancer* adalah perangkat jaringan yang dipasang diantara *client* dan *server*, bekerja sebagai saklar untuk *request* dari *client*. *Load balancer* mengimplementasikan beberapa metode penjadwalan yang akan menentukan ke arah *server* mana *request* dari *client* akan diteruskan.



Gambar 2.1 Skema Sistem *Load balancing*

Beberapa keuntungan yang diperoleh dari teknik *load balancing* sebagai berikut:

a. ***Flexibility :*** *Server* tidak lagi menjadi inti sistem dan resource utama, tetapi menjadi bagian dari banyak *server* yang membentuk *cluster*. Hal ini berarti bahwa performa per unit dari *cluster* tidak terlalu diperhitungkan, tetapi performa *cluster* secara keseluruhan. Sedangkan untuk meningkatkan performa dari *cluster*, *server* atau unit baru dapat ditambahkan tanpa mengganti unit yang lama.
b. ***Scalability :*** Sistem tidak memerlukan desain ulang seluruh arsitektur sistem untuk mengadaptasikan sistem tersebut ketika terjadi perubahan pada komponen sistem.
c. ***Security :*** Untuk semua trafik yang melewati *load balancer*, aturan keamanan dapat dimplementasikan dengan mudah. Dengan *private network* digunakan untuk *real servers*, alamat *IP* nya tidak akan diakses secara langsung dari luar sistem *cluster*.
d. ***High-availability :*** *Load balancer* dapat mengetahui kondisi *real server* dalam sistem secara otomatis, jika terdapat *real server* yang mati maka akan dihapus dari daftar *real server*, dan jika *real server* tersebut kembali aktif maka akan dimasukkan ke dalam daftar *real server*. *Load balancer* juga dapat dikonfigurasi *redundant* dengan *load balancer* yang lain.

2.2.3 Komputasi Paralel

Ide dibalik metode pendekatan komputasi paralel adalah untuk membuat sebuah mesin *single* yang yang sangat hebat dari kumpulan banyak komputer. Cara yang ditempuh untuk mewujudkannya adalah dengan membagi pekerjaan menjadi beberapa bagian pekerjaan kecil sehingga waktu yang dibutuhkan untuk komputasi bisa dipersingkat. Tipe *cluster* ini khususnya dibuat untuk pemecahan masalah komputasi data besar dan rumit yang memerlukan daya komputasi super cepat. Salah satu implementasi dari konsep *parallel computing* yang bersifat *open source* adalah "*Beowulf*".

Perbedaan mendasar antara kelompok *Load balancing* (A) dan *Parallel computing* (B) cukup jelas. Jika anggota di kelompok *node* B tidak peduli dengan anggota lainnya dalam satu *cluster* dan menganggap *request* yang diterimanya adalah langsung dari *client*, maka masing-masing anggota *node* A memiliki keterikatan yang kuat satu sama lain untuk mengerjakan job dalam *cluster*. Data yang diproses oleh *node* A memiliki kekurangan jika hasilnya hanya didapat dari salah satu anggota *node* A tersebut, hasil yang didapat harus kumpulan dari hasil per anggota yang dikumpulkan. Sedangkan data hasil proses salah satu anggota *node* B sudah bisa langsung dikirimkan ke *user* tanpa rekonstruksi ulang.

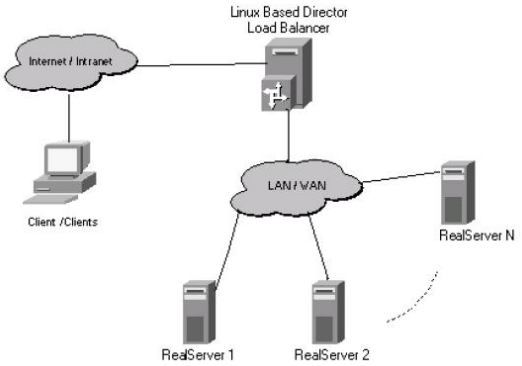
2.3 Linux Virtual Server (LVS)

2.3.1 Pengertian Linux Virtual Server (LVS)

Linux virtual server (LVS) adalah alternatif utama *open-source* yang menyediakan solusi untuk menciptakan sistem *load balancing*. Seperti yang ditulis di home site *linux virtual server project*, "*Linux virtual server is a highly scalable and highly available server built on a cluster of real servers, with the load balancer running on the linux operating system*".

Maksud dari *real server* adalah mesin komputer yang benar-benar melayani *request* yang ada. *Real server* dapat dihubungkan menggunakan LAN maupun WAN yang berkecepatan tinggi. Dan mesin yang menjadi ujung tombak *cluster server* adalah *director*, yaitu berupa *single server* yang menghubungkan *real server* dengan *client* dari luar sistem cluster.

2.3.2 Cara Kerja Linux Virtual Server (LVS)



Gambar 2.2 Skema sistem *linux virtual server*

Saat *request* A datang dari *client* ke *director*, *request* tersebut akan di teruskan ke salah satu *real-server* tertentu. Dan semua paket yang ada hubungan dengan *request* A akan diproses oleh *virtual server* sampai sinkronisasi *FIN* dari koneksi *TCP* atau *connection timeout*, tergantung pada *idle time* maksimum untuk sebuah koneksi. Pemilihan *real server* untuk melayani *request* baru dari *client* ditentukan oleh aturan yang diterapkan dan algoritma penjadwalan yang ada.

Perangkat lunak LVS diimplementasikan di *kernel linux* yang sudah ter-*patch* pada mesin yang akan dijadikan *director* untuk memperoleh performa yang optimal. Walaupun *patch* dilakukan, LVS akan berfungsi jika dikompilasi ulang dalam kernel maupun dikompilasi sebagai modul *kernel*.

Director yang digunakan umumnya adalah sebuah *router* dengan tabel *routing* yang diatur khusus untuk kegunaan LVS. Tabel *routing* tersebut mengijinkan *request* dari banyak *client* untuk mengakses layanan yang disediakan oleh LVS dan meneruskan *request* tersebut ke *real server*. Perlu diketahui bahwa *director* tidak memiliki *socket listener* untuk *service port* yang berasosiasi dengan layanan yang diteruskan kepada *real server*.

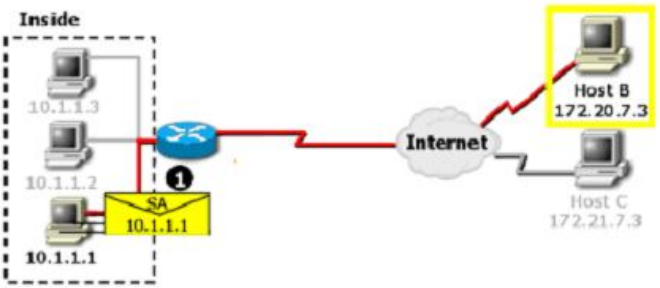
Tahapan aliran *request* yang dikirimkan oleh *client* kepada sistem dapat diketahui melalui ilustrasi sebagai berikut :

1. *Client* membangun koneksi *http* dengan mengetikkan alamat penyedia layanan pada *browser*. Kemudian koneksi akan diterima oleh *director*.
2. *Director* menerima *request* kemudian menulis kembali alamat tujuan yang baru yaitu alamat IP *real server* dengan tidak mengubah alamat pengirim yaitu alamat IP *client*.
3. *Real server* (*web server*) memberikan respon terhadap *request* yang diterima kemudian dikirimkan kembali kepada alamat IP milik *client*. Paket tersebut dilewatkan melalui director sebagai satu-satunya *gateway* yang menghubungkan *real server* dengan jaringan luar sistem.
4. *Director* mengubah alamat IP asal menjadi alamat IP *director* kemudian paket tersebut diteruskan kepada *client*.

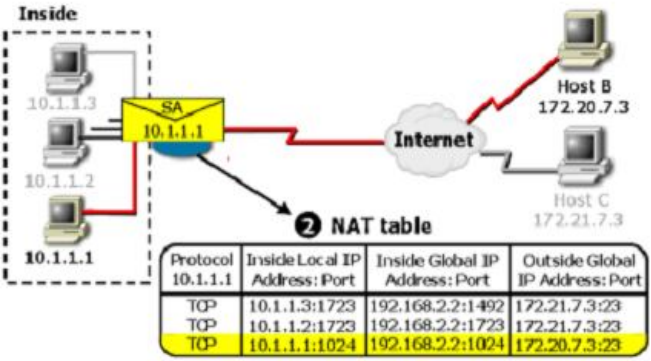
2.4 Network Address Translation (NAT)

NAT merupakan suatu mekanisme translasi yang memetakan satu alamat IP publik menjadi beberapa alamat IP lokal, sehingga meskipun terdapat banyak perangkat dalam jaringan local, akan tampak sebagai satu alamat IP publik. Untuk itu perlu suatu *gateway* yang akan menghubungkan jaringan lokal dengan jaringan internet.

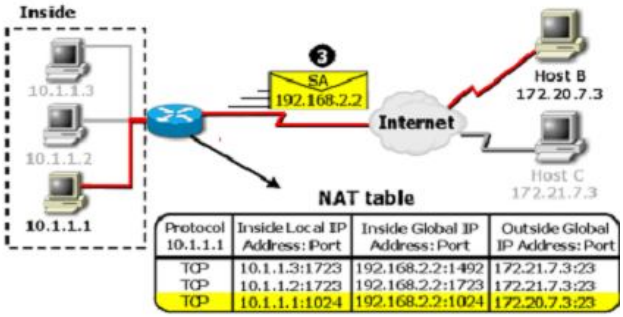
Proses pertukaran data antara jaringan lokal dan jaringan internet pada metode NAT dapat dilihat pada ilustrasi berikut :



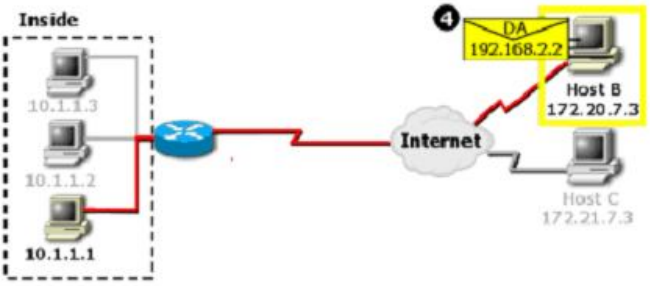
Client dalam jaringan lokal membuat koneksi dengan *host* B yang berada di jaringan internet.



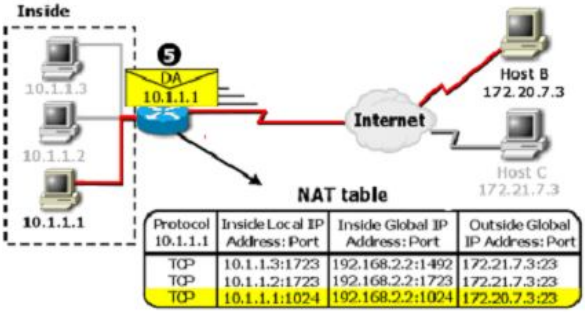
Gateway akan menerima data dari *client* kemudian mengecek translasi alamat publik/global pada tabel NAT. Jika terdapat beberapa *client* yang datang bersamaan maka masing-masing akan ditranslasikan pada alamat global yang sama tetapi berbeda alamat *port*.



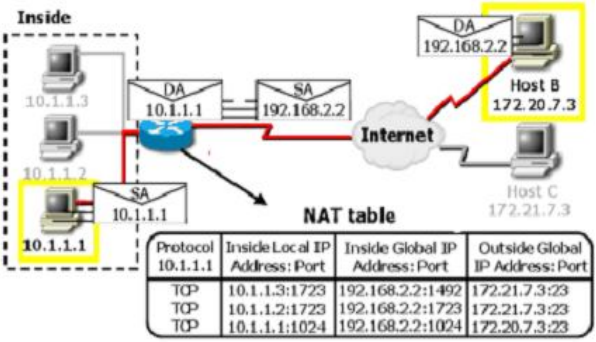
Gateway akan mengganti alamat IP *client* lokal, misal 10.1.1.1 menjadi alamat IP global 192.168.2.2 dan meneruskan paket kepada tujuan, yakni *Host* B.



Host B pada jaringan internet menerima paket dan merespon *client* dengan alamat yang diketahui yaitu 192.168.2.2



Ketika paket balasan kembali diterima oleh *gateway*, lalu akan dicari pada tabel NAT alamat IP global dan alamat *port* yang sesuai dengan catatan saat pengiriman paket.



Kemudian *gateway* akan mentranslasikan alamat IP global tersebut menjadi alamat IP lokal, misal 10.1.1.1, sesuai dengan catatan pada tabel NAT, dan meneruskan paket kepada *client*. *Client* akan menerima paket dan memprosesnya.

Hal-hal sebagai pertimbangan dalam implementasi NAT yaitu keuntungan dan kerugian metode NAT yang dapat dilihat pada **Tabel 2.2** berikut.

Tabel 2.1 Keuntungan dan Kerugian NAT

Keuntungan	Kerugian
Menghemat alamat IP global yang legal	Translasi memerlukan delay switching
Meminimalisir terjadinya bentrok alamat IP	Aplikasi tertentu tidak dapat berjalan
Meningkatkan fleksibilitas	Tidak dapat dilacak end-to-end IP

2.5 Algoritma Penjadwalan

Beberapa jenis algoritma penjadwalan yang dapat diterapkan pada sistem *linux virtual server* pada proses distribusi *request* kepada *real server*, antara lain yaitu :

1. *Round Robin* (rr), yaitu algoritma penjadwalan yang memperlakukan semua *real server* sama menurut jumlah koneksi atau waktu respon.

2. *Weighted Round Robin* (wrr), penjadwalan ini memperlakukan *real server* dengan kapasitas proses yang berbeda. Masing-masing *real server* dapat diberi bobot bilangan *integer* yang menunjukkan kapasitas proses, dimana bobot awal adalah 1.

3. *Least Connection* (lc), merupakan algoritma penjadwalan yang mengarahkan koneksi jaringan pada *server* aktif dengan jumlah koneksi yang paling sedikit. Penjadwalan ini termasuk salah satu algoritma penjadwalan dinamik, karena memerlukan perhitungan koneksi aktif untuk masing-masing *real server*

secara dinamik. Metode penjadwalan ini baik digunakan untuk melancarkan pendistribusian ketika *request* yang datang banyak.

4. *Weighted Least Connection* (wlc), merupakan sekumpulan penjadwalan *least connection* dimana dapat ditentukan bobot kinerja pada masing-masing *real server*. *Server* dengan nilai bobot yang lebih tinggi akan menerima persentase yang lebih besar dari koneksi-koneksi aktif pada satu waktu. Bobot pada masing-masing *real server* dapat ditentukan dan koneksi jaringan dijadwalkan pada masing-masing *real server* dengan persentase jumlah koneksi aktif untuk masing-masing *server* sesuai dengan perbandingan bobotnya (bobot awal adalah 1).

5. *Locality Based Least Connection* (lblc), metode penjadwalan yang akan mendistribusikan lebih banyak *request* kepada *real server* yang memiliki koneksi kurang aktif. Algoritma ini akan meneruskan semua *request* kepada real server yang memiliki koneksi kurang aktif tersebut sampai kapasitasnya terpenuhi.

6. *Destination Hashing* (dh), merupakan algortima penjadwalan statik yang dapat meneruskan *request* dari client kepada satu *real server* tertentu sesuai dengan layanan yang diminta. Terdapat suatu tabel *hash* berisi alamat tujuan dari masing-masing *real server* beserta layanan yang tersedia pada setiap *real server*.

7. *Source Hashing* (sh), hampir sama dengan metode *destination hashing* tetapi pada metode ini tabel berisi mengenai informasi alamat asal paket yang dikirimkan oleh *client*.

8. *Shortest Expected Delay Scheduling* (sed).

9. *Never Queue Scheduling* (nq).

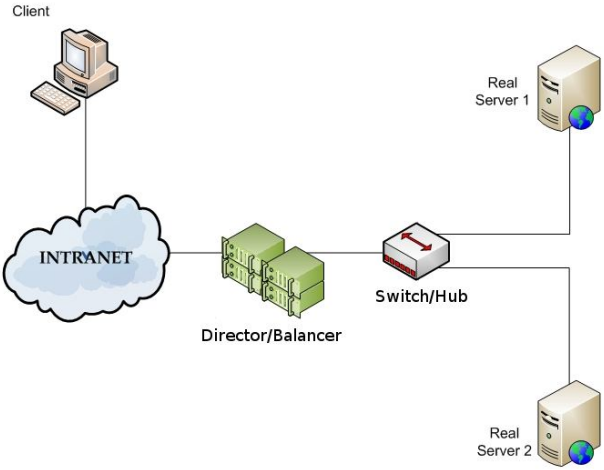
10. *Locality-Based Least-Connection with Replication* (lblcr).

BAB III
PERANCANGAN DAN IMPLEMENTASI

3.1 Perancangan Konfigurasi Sistem

Untuk menganalisis algoritma *scheduling* di *linux virtual server* (LVS), pada tugas akhir ini dilakukan perancangan, implementasi dan pengukuran beberapa parameter meliputi *throughput, request loss, waktu respon,* dan *cpu utilization.*

Model sistem linux virtual server yang dirancang secara lebih jelas digambarkan pada **Gambar 3.1**.



Gambar 3.1 Rancangan model sistem LVS

Pada *linux virtual server* terdapat dua komponen penyusun utama yaitu:

a. *Director*, sebagai sebuah server yang bertugas untuk meneruskan paket dari *client* kepada *real server*. *Client* dari jaringan luar akan menganggap director seolah sebagai *server* tunggal.

b. *Real Server*, sebagai *server* yang melayani permintaan *client*, dalam hal ini sebagai web server yang memberikan layanan *http*.

Sistem *virtual cluster* yang dirancang menggunakan metode NAT dengan beban pada *real server* diseimbangkan oleh *load balancer*/*director*. Terdapat sebuah *director* aktif. *Director* dan *real server* dihubungkan oleh switch dan kedua *real server* memberikan layanan sama yaitu *http*.

3.2 Komponen Sistem

3.2.1 Komponen Perangkat Keras

Sistem *linux virtual cluster* diuji dengan cara membangkitkan *request* secara simultan dalam jumlah yang banyak. Untuk melakukan hal tersebut tentunya akan memerlukan jumlah *client* yang banyak pula. Maka digunakan *request generator* agar cukup dengan satu buah komputer seolah dapat menghasilkan *request* yang dihasilkan oleh banyak client secara simultan.

Spesifikasi komputer client:
a. Processor : Intel Pentium IV CPU 2,4 GHz
b. RAM : 1 GB

Spesifikasi komputer *director*:
1. Processor : : Intel Pentium IV CPU 2,4 GHz
2. RAM : 1 MB
3. Ethernet - 10/100 Mbps

Minimal diperlukan dua *real server* untuk menciptakan sistem *clustering*, dimana aplikasi yang digunakan yaitu aplikasi *http*. Spesifikasi komputer yang digunakan yaitu :
a. Real Server I
Processor : Intel Pentium IV CPU 2,4 GHz, RAM 1 GB
b. Real Server II
Processor : Intel Pentium IV CPU 2,4 GHz RAM 1 MB

3.2.2 Komponen Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut :

a. Sistem Operasi

Komputer sebagai *director* yang diberi label *ld* menggunakan sistem operasi *Ubuntu Server 10.4*. Sedangkan komputer *real server* keduanya menggunakan sistem operasi *Debian Lenny*. Dan untuk komputer *client* (pembangkit *request*) menggunakan sistem operasi *Ubuntu Desktop 10.4*.

b. Kernel

Kernel merupakan inti dari suatu sistem operasi yang berhubungan langsung dengan perangkat keras. Pada bagian kernel terdapat berbagai pilihan untuk memilih dan mengaktifkan *driver* perangkat keras dan algoritma penjadwalan yang akan digunakan. *Kernel* yang digunakan saat penelitian tugas akhir ini adalah *kernel versi 2.6.26* untuk komputer *director* yang diberi label *ld*. Sedangkan untuk komputer sebagai *real server* dan *client* menggunakan *kernel* versi *2.6.32*.

c. Web Server

Perangkat lunak untuk *web server* yaitu *apache*.

d. Ipvsadm

IP Virtual Server Administration (Ipvsadm) digunakan untuk mengatur kerja *director* sehingga dapat menambahkan layanan-layanan apa saja yang dapat diberikan, memilih algoritma penjadwalan yang digunakan, dan meneruskan *request* kepada *real server* yang sedang aktif. Pada *director* dengan label *ld* dan LVS keduanya menggunakan *ipvsadm* versi 1.24.

e. Httperf

Perangkat lunak yang digunakan untuk membangkitkan *request* agar dapat menghasilkan banyak *request* dalam satu waktu yaitu *httperf*. Dengan menggunakan *httperf* maka tidak perlu ada banyak komputer sebagai *client*, cukup satu komputer sebagai *client* tetapi mampu menghasilkan banyak *request* secara simultan. Pada penelitian tugas akhir ini digunakan *httperf* versi 0.8.

3.3 Pembuatan Sistem

3.3.1 Kernel Sistem (Linux Director)

Sistem operasi linux yang menggunakan kernel versi dibawah 2.4.28 belum mendukung linux virtual server, sehingga diperlukan patching dengan melakukan kompilasi kernel, namun pada tugas akhir ini penulis menggunakan kernel yang telah mendukung linux virtual server, namun perlu mengaktifkan modul-modul virtual server yang ada pada kernel menggunakan *modprobe.*

```
echo ip_vs_dh >> /etc/modules
echo ip_vs_ftp >> /etc/modules
echo ip_vs >> /etc/modules
echo ip_vs_lblc >> /etc/modules
echo ip_vs_lblcr >> /etc/modules
echo ip_vs_lc >> /etc/modules
echo ip_vs_nq >> /etc/modules
echo ip_vs_rr >> /etc/modules
echo ip_vs_sed >> /etc/modules
echo ip_vs_sh >> /etc/modules
echo ip_vs_wlc >> /etc/modules
echo ip_vs_wrr >> /etc/modules

modprobe ip_vs_dh
modprobe ip_vs_ftp
modprobe ip_vs
modprobe ip_vs_lblc
modprobe ip_vs_lblcr
modprobe ip_vs_lc
modprobe ip_vs_nq
modprobe ip_vs_rr
modprobe ip_vs_sed
modprobe ip_vs_sh
modprobe ip_vs_wlc
modprobe ip_vs_wrr
```

Sedangkan untuk kernel yang belum mendukung linux virtual server harus dilakukan kompilasi yaitu dengan mengambil source kernel dari situs kernel.org. Atau dapat menggunakan tool *Synaptic Package Manager*.

Setelah semua persiapan selesai, selanjutnya melakukan kompilasi ulang kernel linux. Untuk melakukan kompilasi ulang kernel dibutuhkan paket-paket yang digunakan untuk kompilasi, yaitu :

- *kernel-package*
- *libnucrses5-dev*

 # apt-get install kernel-package libncurses5-dev

Setelah paket-paket di atas terinstall, kompilasi kernel sudah siap dilakukan. Langkah-langkah selanjutnya untuk kompilasi kernel adalah :

a. Masuk ke directori */usr/src* dan ekstrak source kernel-2.6.x yang ada.

```
# tar -jxvf linux-2.6.x.tar.bz2
```

b. Masukkan patch *ipvs* pada directori */usr/src* dan ekstrak patch tersebut.

```
# tar -zxvf ipvs-1.1.7.tar.gz
```

c. Masuk ke directori linux-2.6.x hasil dari peng-ekstrakan yang telah dilakukan, serta lakukan penge-patch-an kernel.

```
# cd /linux-2.6.x

# patch -pq < ../ipvs-1.1.7/linuxkernel_ksyms_c.diff
# patch -pq < ../ipvs-1.1.7/linuxnet_netsyms_c.diff
```

Jika service pem-forward-an dari real-server yang diinginkan bisa untuk direct routing, maka ditambahkan penge-patch-an kernel :

```
# patch -pq < ../ipvs-1.1.7/contrib/patches/hidden-
2.6.xpre10-1.diff
```

d. Lakukan konfigurasi kernel.

```
# make menuconfig atau make xconfig
```

e. Masuk ke menu awal konfigurasi kernel.

f. Masuk ke menu networking options.

j. Aktifkan fitur-fitur yang mendukung LVS, yaitu :

```
# cd /linux-2.6.x
# patch -pq < ../ipvs-1.1.7/linuxkernel_ksyms_c.diff
# patch -pq < ../ipvs-1.1.7/linuxnet_netsyms_c.diff
#  patch  -pq  <  ../ipvs-1.1.7/contrib/patches/hidden-
2.6.xpre10-1.diff
# make menuconfig atau make xconfig
Networking options --->
Network packet filtering (replaces ipchains)
<m> IP: tunnelling
IP: Netfilter Configuration --->
<m> Connection tracking (required for masq/NAT)
<m> FTP protocol support
<m>     IP     tables     support     (required     for
filtering/masq/NAT)
<m> Packet filtering
<m> REJECT target support
```

h. Build dan Install Kernel.

```
# make dep
```

Kemudian membuat modules dari kernel.

```
# make bzImage modules
```

Setelah itu install seluruh modules yang telah dibuat pada directori */lib/modules/2.6.x*.

```
# make install modules_install
```

Yang terakhir adalah mengintall kernel linux pada directori */boot/vmlinuz-2.6.x*.

```
# apt-get install kernel-image-2.6.x
```

i. Update boot-loader.
Bila *grub* digunakan sebagai bootloader, maka tambahkan konfigurasi pada */etc/grub.conf*.

```
title 2.6.x LVS
root (hd0,0)
kernel /vmlinuz-2.6.x ro root=/dev/hda3
```

Namun bila bootloader-nya adalah *lilo*, maka tambahkan konfigurasi pada */etc/lilo.conf*.

```
image=/boot/vmlinuz-2.6.x
label=2.6.x-lvs
read-only
root=/dev/hda2
```

j. Reboot dan Booting dengan Kernel Linux yang baru.

k. Build dan Install LVS.
Perintah yang dilakukan pada tahap ini harus berada pada directori */ipvs-1.1.7/ipvs*.

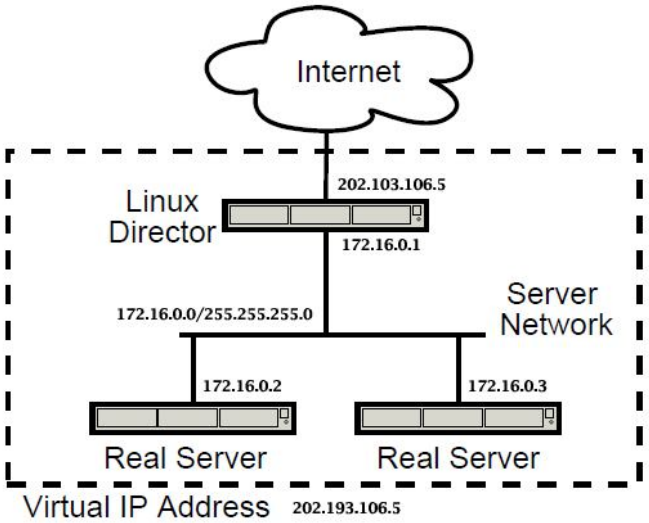
```
# make KERNELSOURCE=/kernel/source/linux-2.6.x all
# make KERNELSOURCE=/kernel/source/linux-2.6.x
modules_install
```

3.3.2 Konfigurasi LVS (Linux Director)

Setelah proses kompilasi, maka sudah terinstall pula paket penting untuk administratif Load Balancing, yaitu *ipvsadm*. Namun untuk kernel yang telah mendukung load balancer seperti yang penulis gunakan, ipvsadm tidak terinstall secara defaul, untuk itu kita perlu menginstall paket tersebut:

```
apt-get install ipvsadm
```

Dan berikut ini contoh konfigurasi linux director *(ld)* untuk memforward data serta membagi bebannya ke real server.



Gambar 3.2 Topologi sistem jaringan LVS

Untuk membuat load balancer dapat mem-forward-kan paket secara masquerade (penyamaran) :

```
# echo “1” > /proc/sys/net/ipv4/ip_forward
# iptables -A FORWARD -s 172.16.0.0/24 -d 0.0.0.0/0 -p
80
-j ACCEPT
# iptables -t nat -P POSTROUTING DROP
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Selanjutnya, menambahkan virtual service http (80) dan algoritma penjadwalannya, misal *wlc*.

```
# ipvsadm -A -t 202.103.106.5:80 -s wlc
```

Kemudian akan ditambahkan kedua real-server dengan weight yang sama yaitu 2:

```
# ipvsadm -a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
# ipvsadm -a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
```

3.3.3 Httperf (Client)

Httperf merupakan workload generator untuk membangkitkan request secara simultan ke sistem load balancing yang kita bangun. Aplikasi ini diinstall pada komputer client.

```
apt-get install httperf
```

Contoh penggunaan httperf untuk membagkitkan request secara simultan:

```
httperf --hog --server 202.103.106.5 --num-conn 20000
--ra 2000 --timeout 5
```

Perintah diatas untuk membentuk 20000 koneksi dengan 2000 koneksi per detik

Dan contoh hasil eksekusinya sebagai berikut:

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000 --
send-buffer=4096 --recv-buffer=16384 --num-conns=20000
--num-calls=1
Maximum connect burst length: 316

Total: connections 5151 requests 5122 replies 4390
test-duration 15.468 s

Connection rate: 333.0 conn/s (3.0 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 19.1 avg 1637.0 max 7369.7
median 1365.5 stddev 943.4
Connection time [ms]: connect 603.0
Connection length [replies/conn]: 1.000
```

```
Request rate: 331.1 req/s (3.0 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 133.0 avg 292.5 max 451.4
stddev 159.2 (3 samples)
Reply time [ms]: response 1038.7 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer 0.0
(total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4390 4xx=0 5xx=0

CPU time [s]: user 0.33 system 14.08 (user 2.1% system
91.0% total 93.2%)
Net I/O: 203.2 KB/s (1.7*10^6 bps)

Errors: total 15610 client-timo 761 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 14849 addrunavail 0 ftab-full 0
other 0
```

3.3.4 Apache Web Server (Real Server)

Pada kedua Real Server diinstall Apache Web Server yang bertugas untuk menerima dan membalas request yang datang dari komputer klien.

Installasi apache dapat dilakukan dengan apt-get:

```
apt-get install apache2
```

Selain itu pada Real Server sebaiknya dilakukan tuning agar dapat menerima beban dengan baik seperti; menambah nilai MaxSpareServer, MaxClients dan MaxRequestPerChild di konfigurasi file apache.

BAB IV
UJI COBA DAN ANALISA

4.1 Pendahuluan

Prinsip kerja dari LVS via NAT pada proyek tugas akhir ini adalah bagaimana membagi beban kerja yang telah dihasilkan oleh perangkat lunak pembangkit *workload generator httperf* kepada Real-Server, dengan proses peng-clusteran yang mempunyai IP Private yang terletak dalam satu jaringan dan Director yang mempunyai IP Public. Dalam bab ini penulis melakukan pengujian dan analisa terhadap beberapa algoritma LVS yang ada untuk mengetahui performa masing-masing algoritma tersebut, dengan skenario dan parameter yang telah ditentukan.

Algoritma penjadwalan pada LVS NAT yang diuji:

Round Robin (rr)
Weighted Round Robin (wrr)
Least Connection (lc)
Weighted Least Connection (wlc)
Locality Based Least Connection (lblc)
Locality-Based Least-Connection with Replication (lblcr)
Destination Hashing (dh)
Source Hashing (sh)
Shortest Expected Delay Scheduling (sed)
Never Queue Scheduling (nq)

4.2 Pengujian

4.2.1 Sistem

Sistem yang dibangun di atas virtulization software VirtualBox, terdiri dari:

- Director (1 computer)
- Real Server (3 computer)
- Clients (1 computer)

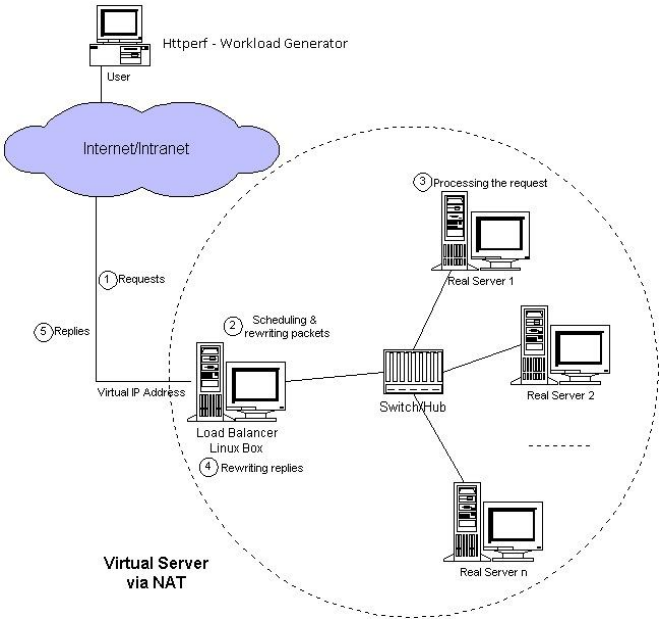
Pada client diinstall *httperf* yang berfungsi sebagai generator workload dan alat untuk mengambil data.

4.2.2 Parameter Pengujian

Parameter uji pada percobaan ini, yaitu:

Time Response
Throughput
Request Lost
CPU utilization

4.2.3 Proses Pengujian



Gambar 4.1 Detail proses yang terjadi di Sistem LVS NAT

Berikut akan dijelaskan beberapa proses yang terjadi di LVS-NAT dan bagaimana mendapatkan data hasil uji untuk membandingkan antar Algoritma penjadwalan di LVS-NAT

1. Pada proses 1 di gambar 4.1, client melakukan request, dimana pada client terdapat httperf yang berfungsi sebagai workload generator dan alat penghasil data uji.
   Contoh proses workload request pada httperf:

   ```
   httperf --hog --server 202.103.106.5 --num-conn 20000 --ra 2000 --timeout 5
   ```

   Perintah diatas akan menghasilakan 20000 dengan 2000 koneksi per detik.

   Contoh hasil workload dari perintah diatas sebagai berikut:

```
httperf --hog --timeout=5 --client=0/1 --server=202.103.106.5 --port=80 --uri=/ --rate=2000 --send-buffer=4096 --recv-buffer=16384 --num-conns=20000 --num-calls=1
Maximum connect burst length: 316

Total: connections 5151 requests 5122 replies 4390 test-duration 15.468 s

Connection rate: 333.0 conn/s (3.0 ms/conn, <=1022 concurrent connections)
Connection time [ms]: min 19.1 avg 1637.0 max 7369.7 median 1365.5 stddev 943.4
Connection time [ms]: connect 603.0
Connection length [replies/conn]: 1.000

Request rate: 331.1 req/s (3.0 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 133.0 avg 292.5 max 451.4 stddev 159.2 (3 samples)
Reply time [ms]: response 1038.7 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer 0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4390 4xx=0 5xx=0

CPU time [s]: user 0.33 system 14.08 (user 2.1% system 91.0% total 93.2%)
Net I/O: 203.2 KB/s (1.7*10^6 bps)

Errors: total 15610 client-timo 761 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 14849 addrunavail 0 ftab-full 0 other 0
```

2. Pada proses 2 digambar 4.1, load balancer merupakan inti dari sistem, dimana kita akan mengganti antara satu algoritma penjadwalan yang satu dengan yang lainya.

   Contoh penggunaan salah satu algoritma penjadwalan rr:

```
ipvsadm -A -t 202.103.106.5:80 -s rr
ipvsadm -a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
ipvsadm -a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
```

Bila ingin mencoba algoritma lain misalnya wlc, cukup dengan mengubah perintah diatas dari rr dengan wlc, dengan sebelumnya menghapus IPVS table rr, yaitu

```
# menghapus counter data packet LVS
ipvsadm –Z
# menghapus IPVS table
ipvsadm –C
# Menggunaka algoritma baru wlc
ipvsadm -A -t 202.103.106.5:80 -s rr
ipvsadm -a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
ipvsadm -a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
```

4.2.4 Mendapatkan Data Hasil Uji

Selain sebagai workload generator httperf juga digunakan sebagai alat uji algoritma. Httperf dapat menghasilkan keempat parameter uji yang dibutuhkan, yaitu time response, throughput, cpu utilization dan request lost.

Berikut hasil dari workload httperf:

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 316

Total: connections 5151 requests 5122 replies 4390
test-duration 15.468 s

Connection rate: 333.0 conn/s (3.0 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 19.1 avg 1637.0 max
7369.7 median 1365.5 stddev 943.4
Connection time [ms]: connect 603.0
Connection length [replies/conn]: 1.000

Request rate: 331.1 req/s (3.0 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 133.0 avg 292.5 max
451.4 stddev 159.2 (3 samples)
Reply time [ms]: response 1038.7
transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4390 4xx=0 5xx=0

CPU time [s]: user 0.33 system 14.08 (user 2.1%
system 91.0% total 93.2%)
Net I/O: 203.2 KB/s (1.7*10^6 bps)

Errors: total 15610 client-timo 761 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 14849 addrunavail 0 ftab-full 0
other 0
```

Berdasarkan manual dari httperf, hasil dari workload httperf terdapat 6 bagian penting yaitu:

Total section
Connection section
Request section
Reply section
Miscellaneous section
Error Section

Dari beberapa section tersebut dapat ditemukan parameter-parameter yang dibutuhkan untuk pengujian.

Time Response
Time response didapatkan dari *Reply Section*

```
Reply time [ms]: response 1038.7 transfer 0.0
```

Throughput
Throughput berdasarkan manual httperf dapat diambil dari *Miscellaneous Section,* yaitu Net I/O yang merupakan rata-rata throghput jaringan yang mepunyai satuan kilobytes per detik dan megabits per detik

```
Net I/O: 203.2 KB/s (1.7*10^6 bps)
```

Request Lost
Request Lost didapatkan dari *Error Section* pada *connrefused* dan *connreset*

```
Errors: total 15610 client-timo 761 socket-timo 0
connrefused 0 connreset 0
```

4.2.5 Data Hasil Uji
Detail hasil uji tiap-tiap algoritma dilampirkan.

4.3 Analisa

Berikut table hasil pengujian LVS-NAT:

Table 4.1 Throughput

Connection /s	Throughput (KB/s)									
	rr	wrr	lc	wlc	lblc	lblcr	dh	sh	sed	nq
2000	203.2	217.5	222.2	184.1	156.8	183.7	152.4	247.5	155.7	219.4
8000	176.3	149.1	140.7	136.1	102.9	121.5	138.6	110.3	143.7	128.2
14000	102.1	104.6	130	107.8	72.9	196.9	220.7	110.8	114.6	110.1

Table 4.2 Time Response

Connection /s	Time Response (ms)									
	rr	wrr	lc	wlc	lblc	lblcr	dh	sh	sed	nq
2000	1038.7	830.5	1028.8	826.7	611.4	1525.7	888.8	853.1	763.6	1042.6
8000	793.1	812.2	651.8	698.8	577.3	708.9	872.7	225.8	639.6	670.2
14000	983.4	889.6	700.9	1025.3	401.4	1028.7	483.5	1032.8	717.6	689.2

Setelah mengamati dan mempelajari hasil data-data yang didapatkan dari httperf, dapat diambil kesimpulan hanya 2 parameter uji saja yang bisa dijadikan indikator pemilihan algoritma LVS yang baik yaitu: Throughput dan Time Response.

Untuk CPU utilization dan Error tidak dapat dijadikan parameter penentuan algoritma terbaik. Hasil CPU Utilization yang didapat dari httperf merupakan CPU utilization dari client machine terhadap kinerjanya saat membangkitkan reqeuest secara simultan, jadi tidak berkaitan dengan kinerja dari suatu algoritma. Sedangkan untuk Error yang dihasilkan dari httperf pada percobaan yang dilakukan penulis merupakan error pada client, error berupa refused dari server tidak ditemukan dan selalu bernilai 0 (lihat hasil uji detail pada lampiran), untuk itu parameter Error juga tidak dapat diambil sebagai parameter penentu algoritma terbaik.

Dari hasil uji berdasarkan parameter Throughput dan Time Response, algoritma *wrr* meperlihatkan performa yang lebih baik dibandingkan algoritma lainnya. Algoritma *wrr* memiliki time response yang cukup stabil pada berbagai percobaan dengan jumlah koneksi yang berbeda serta menunjukkan throughput yang semakin baik walaupun jumlah koneksi semakin banyak.

Halaman ini sengaja dikosongkan

BAB V
PENUTUP

5.1 Kesimpulan

Dari hasil analisa yang telah dilakukan pada bab sebelumnya, dapat diambil kesimpulan, yaitu :

- Pada percobaan tugas akhir ini penulis menyimpulkan algoritma *wrr* sebagai algoritma terbaik dibandingkan algoritma lainnya.
- Algoritma *wrr* dipilih sebagai algoritma terbaik karena menunjukkan hasil uji *througput* yang lebih baik dibandingkan algoritma lainya, serta meliliki hasil uji *time respon* yang lebih stabil walaupun mengalami perubahan jumlah koneksi per detik yang berbeda-beda dari rendah, menengah maupun tinggi.

5.2 Saran

Untuk hasil komparasi yang lebih baik, perlu dipertimbangkan hal-hal berikut ini:

- Penggunaan real hardware yang lebih baik, tidak hanya perangkat lunak virtualisasi seperti VirtualBox ataupun VMware.
- Jumlah cluster dan client yang lebih banyak tentunya akan memberikan hasil uji yang lebih baik
- Pemilihan penggunaan software penguji agar menghasilkan hasil yang lebih detail dan parameter yang lebih banyak.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

[1] Zhang, W. *Linux Virtual Server for Scalable Network Services.* National Laboratory for Parallel & Distributed Processing. China.

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

[4] Mack, J. 2001-2006. *LVS-mini-HOWTO*. jmack (at) wm7d (dot) net.

[5] O'Rourke Patrick dan Mike Keefe. 2001. *Performance Evaluation of Linux Virtual Server*. Mission Critical Linux, Inc.

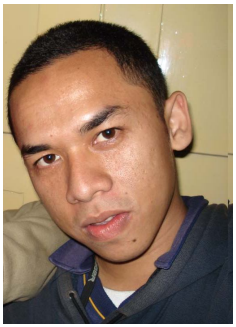
[6] Mosberger David dan Tai Jin. Httperf—A Tool for Measuring Web Server Performance. HP Reasearch Labs.

[7] Kusuma, P.H., Rumani R, Mulyana, A. *Implementasi Penyedia Layanan Terdistribusi Berbasis Linux Dengan Konsep Penyeimbang Beban Jaringan.* Institut Teknologi Telkom.

[8] Alhaadii, Y, J. 2006. *Implementasi Load Balancing Multi Serve Menggunakan LVS (Linux Virtual Server) VIA NAT (Network Address Translation).* Politeknik Elektronika Negeri Surabaya.

Halaman ini sengaja dikosongkan

TENTANG PENULIS



Nama	:	Abdul Haris Nasution
NRP	:	7406.040.036
Tempat, Tanggal Lahir	:	Duri, 05 Juni 1985
Alamat Rumah	:	Jl. Sudirman No. 278 Duri - Riau
Telepon/HP	:	0857-031-8000-1
Website	:	http://hariscorner.com
Email	:	*me@hariscorner.com*

Riwayat Pendidikan :

- SDN 006 Duri - Riau 1993-1997
- SLTPN 2 Duri - Riau 1997-2000
- SMUN 5 Medan 2000-2003
- Industrial Engineering – UGM 2004-2006
- D4 Teknik Informatika – EEPIS-ITS 2006-2011

Halaman ini sengaja dikosongkan

LAMPIRAN

Dalam buku ini dilampirkan:

1. Konfigurasi Detail Jaringan LVS NAT
2. Bash Script Konfigurasi Sistem
3. Hasil Uji Httperf

1. Konfigurasi Detail Jaringan LVS NAT

Clients I

IP	**: 202.103.106.2**
Netmask	**: 255.255.255.0**

Clients II

IP	**: 202.103.106.3**
Netmask	**: 255.255.255.0**

Linux Director

IP 1 eth0	**: 202.103.106.5**
Netmask	**: 255.255.255.0**
IP 2 eth0:1	**: 172.16.0.1**
Netmask	**: 255.255.255.0**

Real Server 1

IP	**: 172.16.0.2**
Netmask	**: 255.255.255.0**
Gateway	**: 172.16.0.1**

Real Server 1

IP	**: 172.16.0.3**
Netmask	**: 255.255.255.0**
Gateway	**: 172.16.0.1**

Real Server 1

IP	**: 172.16.0.4**
Netmask	**: 255.255.255.0**
Gateway	**: 172.16.0.1**

2. Bash Script Konfigurasi Sistem

step1-aktif-forwarder.sh (Linux Director)

```
#!/bin/bash
# aktifkan forwarder
echo "1" > /proc/sys/net/ipv4/ip_forward
```

step2-iptables.sh (Linux Director)

```
#!/bin/bash
# set forwarder
iptables -A FORWARD -s 172.16.0.0/24 -d 0.0.0.0/0 -p
80 -j ACCEPT
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Algoritma setup (Linux Director)

wrr.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 -s wrr
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

wlc.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 -s wlc
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

sh.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 -s sh
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

sed.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 -s sed
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

rr.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 -s rr
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

nq.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 -s nq
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

lc.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 -s lc
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

lblc.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 –s lblc
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

lblcr.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 -s lblcr
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

dh.sh

```
#!/bin/bash
echo "
-A -t 202.103.106.5:80 -s dh
-a -t 202.103.106.5:80 -r 172.16.0.2:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.3:80 -m
-a -t 202.103.106.5:80 -r 172.16.0.4:80 -m
" | ipvsadm -R
```

Httperf Workload

Workload-2000.sh

```
#!/bin/bash
# This is script to generate http workload
# fix 20000 connection, and 2000 connection per second
httperf --hog --server 202.103.106.5 --num-conn 20000
--ra 2000 --timeout 5
```

Workload-8000.sh

```
#!/bin/bash
# This is script to generate http workload
# fix 20000 connection, and 8000 connection per second
httperf --hog --server 202.103.106.5 --num-conn 20000
--ra 8000 --timeout 5
```

Workload-14000.sh

```
#!/bin/bash
# This is script to generate http workload
# 20000 connection, and 14000 connection per second
httperf --hog --server 202.103.106.5 --num-conn 20000
--ra 14000 --timeout 5
```

3. Hasil Uji Httperf

WRR-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 665

Total: connections 5393 requests 5298 replies 4729
test-duration 15.497 s

Connection rate: 348.0 conn/s (2.9 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 10.2 avg 1609.9 max
8497.6 median 1168.5 stddev 1159.0
Connection time [ms]: connect 797.2
Connection length [replies/conn]: 1.000

Request rate: 341.9 req/s (2.9 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 128.6 avg 315.2 max
418.0 stddev 161.8 (3 samples)
Reply time [ms]: response 830.5 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4729 4xx=0 5xx=0

CPU time [s]: user 0.42 system 13.37 (user 2.7%
system 86.3% total 89.0%)
Net I/O: 217.5 KB/s (1.8*10^6 bps)

Errors: total 15271 client-timo 664 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 14607 addrunavail 0 ftab-full 0
other 0
```

WRR-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 2302

Total: connections 2016 requests 2016 replies 1571
test-duration 7.620 s

Connection rate: 264.6 conn/s (3.8 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 293.1 avg 1079.9 max
3623.8 median 1035.5 stddev 427.5
Connection time [ms]: connect 257.5
Connection length [replies/conn]: 1.000

Request rate: 264.6 req/s (3.8 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 310.5 avg 310.5 max
310.5 stddev 0.0 (1 samples)
Reply time [ms]: response 812.2 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1571 4xx=0 5xx=0

CPU time [s]: user 0.15 system 6.93 (user 1.9%
system 90.9% total 92.9%)
Net I/O: 149.1 KB/s (1.2*10^6 bps)

Errors: total 18429 client-timo 445 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 17984 addrunavail 0 ftab-full 0
other 0
```

WRR-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 4084

Total: connections 1335 requests 1335 replies 912
test-duration 6.409 s

Connection rate: 208.3 conn/s (4.8 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 658.7 avg 1400.4 max
3202.7 median 1242.5 stddev 606.3
Connection time [ms]: connect 484.7
Connection length [replies/conn]: 1.000

Request rate: 208.3 req/s (4.8 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 182.4 avg 182.4 max
182.4 stddev 0.0 (1 samples)
Reply time [ms]: response 889.6 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=912 4xx=0 5xx=0

CPU time [s]: user 0.12 system 5.87 (user 1.8%
system 91.6% total 93.4%)
Net I/O: 104.6 KB/s (0.9*10^6 bps)

Errors: total 19088 client-timo 423 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18665 addrunavail 0 ftab-full 0
other 0
```

WLC-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 298

Total: connections 5058 requests 5056 replies 3903
test-duration 15.351 s

Connection rate: 329.5 conn/s (3.0 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 43.2 avg 1301.0 max
5355.0 median 992.5 stddev 1056.9
Connection time [ms]: connect 454.3
Connection length [replies/conn]: 1.000

Request rate: 329.4 req/s (3.0 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 98.1 avg 259.9 max
382.7 stddev 146.2 (3 samples)
Reply time [ms]: response 826.7 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=3903 4xx=0 5xx=0

CPU time [s]: user 0.32 system 13.74 (user 2.1%
system 89.5% total 91.6%)
Net I/O: 184.1 KB/s (1.5*10^6 bps)

Errors: total 16097 client-timo 1155 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 14942 addrunavail 0 ftab-full 0
other 0
```

WLC-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 1773

Total: connections 2009 requests 2009 replies 1381
test-duration 7.452 s

Connection rate: 269.6 conn/s (3.7 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 375.4 avg 1047.2 max
3300.3 median 828.5 stddev 659.3
Connection time [ms]: connect 302.5
Connection length [replies/conn]: 1.000

Request rate: 269.6 req/s (3.7 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 275.8 avg 275.8 max
275.8 stddev 0.0 (1 samples)
Reply time [ms]: response 698.8 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1381 4xx=0 5xx=0

CPU time [s]: user 0.11 system 7.03 (user 1.5%
system 94.4% total 95.9%)
Net I/O: 136.1 KB/s (1.1*10^6 bps)

Errors: total 18619 client-timo 628 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 17991 addrunavail 0 ftab-full 0
other 0
```

WLC-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 3531

Total: connections 1136 requests 1136 replies 899
test-duration 6.023 s

Connection rate: 188.6 conn/s (5.3 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 1020.3 avg 1792.0 max
3427.4 median 1794.5 stddev 565.0
Connection time [ms]: connect 724.3
Connection length [replies/conn]: 1.000

Request rate: 188.6 req/s (5.3 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 179.8 avg 179.8 max
179.8 stddev 0.0 (1 samples)
Reply time [ms]: response 1025.3 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=899 4xx=0 5xx=0

CPU time [s]: user 0.12 system 5.18 (user 1.9%
system 86.1% total 88.0%)
Net I/O: 107.8 KB/s (0.9*10^6 bps)

Errors: total 19101 client-timo 237 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18864 addrunavail 0 ftab-full 0
other 0
```

SH-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 182

Total: connections 5184 requests 4801 replies 4542
test-duration 13.004 s

Connection rate: 398.6 conn/s (2.5 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 8.4 avg 1645.4 max 7546.1
median 761.5 stddev 1585.1
Connection time [ms]: connect 805.0
Connection length [replies/conn]: 1.000

Request rate: 369.2 req/s (2.7 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 337.8 avg 366.0 max
394.1 stddev 39.8 (2 samples)
Reply time [ms]: response 853.1 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4542 4xx=0 5xx=0

CPU time [s]: user 0.21 system 12.04 (user 1.6%
system 92.6% total 94.2%)
Net I/O: 247.5 KB/s (2.0*10^6 bps)

Errors: total 15458 client-timo 642 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 14816 addrunavail 0 ftab-full 0
other 0
```

SH-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 1198

Total: connections 2260 requests 2163 replies 1535
test-duration 10.180 s

Connection rate: 222.0 conn/s (4.5 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 137.2 avg 584.1 max
3606.9 median 222.5 stddev 948.8
Connection time [ms]: connect 378.2
Connection length [replies/conn]: 1.000

Request rate: 212.5 req/s (4.7 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 0.0 avg 153.5 max 307.0
stddev 217.1 (2 samples)
Reply time [ms]: response 225.8 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1535 4xx=0 5xx=0

CPU time [s]: user 0.16 system 9.88 (user 1.5%
system 97.1% total 98.6%)
Net I/O: 110.3 KB/s (0.9*10^6 bps)

Errors: total 18465 client-timo 725 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 17740 addrunavail 0 ftab-full 0
other 0
```

SH-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 3334

Total: connections 1453 requests 1452 replies 1443
test-duration 9.184 s

Connection rate: 158.2 conn/s (6.3 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 197.2 avg 2053.5 max
5763.8 median 1680.5 stddev 1464.2
Connection time [ms]: connect 1023.5
Connection length [replies/conn]: 1.000

Request rate: 158.1 req/s (6.3 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 279.8 avg 279.8 max
279.8 stddev 0.0 (1 samples)
Reply time [ms]: response 1032.8 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1443 4xx=0 5xx=0

CPU time [s]: user 0.24 system 8.69 (user 2.6%
system 94.6% total 97.2%)
Net I/O: 110.8 KB/s (0.9*10^6 bps)

Errors: total 18557 client-timo 10 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18547 addrunavail 0 ftab-full 0
other 0
```

SED-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 444

Total: connections 4860 requests 4806 replies 3707
test-duration 17.240 s

Connection rate: 281.9 conn/s (3.5 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 10.5 avg 1404.1 max
8512.1 median 755.5 stddev 1538.6
Connection time [ms]: connect 721.9
Connection length [replies/conn]: 1.000

Request rate: 278.8 req/s (3.6 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 110.3 avg 247.0 max
363.9 stddev 127.9 (3 samples)
Reply time [ms]: response 763.8 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=3707 4xx=0 5xx=0

CPU time [s]: user 0.27 system 15.54 (user 1.6%
system 90.2% total 91.7%)
Net I/O: 155.7 KB/s (1.3*10^6 bps)

Errors: total 16293 client-timo 1153 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 15140 addrunavail 0 ftab-full 0
other 0
```

SED-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 2120

Total: connections 1937 requests 1937 replies 1263
test-duration 6.501 s

Connection rate: 297.9 conn/s (3.4 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 416.0 avg 855.6 max
3091.6 median 777.5 stddev 454.5
Connection time [ms]: connect 209.5
Connection length [replies/conn]: 1.000

Request rate: 297.9 req/s (3.4 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 252.6 avg 252.6 max
252.6 stddev 0.0 (1 samples)
Reply time [ms]: response 639.6 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1263 4xx=0 5xx=0

CPU time [s]: user 0.08 system 6.13 (user 1.2%
system 94.3% total 95.6%)
Net I/O: 143.7 KB/s (1.2*10^6 bps)

Errors: total 18737 client-timo 674 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18063 addrunavail 0 ftab-full 0
other 0
```

SED-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 3243

Total: connections 1561 requests 1561 replies 1030
test-duration 6.635 s

Connection rate: 235.3 conn/s (4.3 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 364.0 avg 929.5 max
3110.1 median 872.5 stddev 271.9
Connection time [ms]: connect 244.8
Connection length [replies/conn]: 1.000

Request rate: 235.3 req/s (4.3 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 206.0 avg 206.0 max
206.0 stddev 0.0 (1 samples)
Reply time [ms]: response 717.6 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1030 4xx=0 5xx=0

CPU time [s]: user 0.11 system 6.23 (user 1.6%
system 93.9% total 95.6%)
Net I/O: 114.6 KB/s (0.9*10^6 bps)

Errors: total 18970 client-timo 531 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18439 addrunavail 0 ftab-full 0
other 0
```

RR-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 316

Total: connections 5151 requests 5122 replies 4390
test-duration 15.468 s

Connection rate: 333.0 conn/s (3.0 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 19.1 avg 1637.0 max
7369.7 median 1365.5 stddev 943.4
Connection time [ms]: connect 603.0
Connection length [replies/conn]: 1.000

Request rate: 331.1 req/s (3.0 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 133.0 avg 292.5 max
451.4 stddev 159.2 (3 samples)
Reply time [ms]: response 1038.7 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4390 4xx=0 5xx=0

CPU time [s]: user 0.33 system 14.08 (user 2.1%
system 91.0% total 93.2%)
Net I/O: 203.2 KB/s (1.7*10^6 bps)

Errors: total 15610 client-timo 761 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 14849 addrunavail 0 ftab-full 0
other 0
```

RR-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 2140

Total: connections 2101 requests 2101 replies 1665
test-duration 6.819 s

Connection rate: 308.1 conn/s (3.2 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 291.4 avg 1021.0 max
3112.3 median 939.5 stddev 329.5
Connection time [ms]: connect 230.1
Connection length [replies/conn]: 1.000

Request rate: 308.1 req/s (3.2 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 332.9 avg 332.9 max
332.9 stddev 0.0 (1 samples)
Reply time [ms]: response 793.1 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1665 4xx=0 5xx=0

CPU time [s]: user 0.14 system 6.36 (user 2.1%
system 93.2% total 95.3%)
Net I/O: 176.3 KB/s (1.4*10^6 bps)

Errors: total 18335 client-timo 436 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 17899 addrunavail 0 ftab-full 0
other 0
```

RR-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 6337

Total: connections 1187 requests 1187 replies 831
test-duration 5.964 s

Connection rate: 199.0 conn/s (5.0 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 845.9 avg 1563.8 max
3487.6 median 1531.5 stddev 498.4
Connection time [ms]: connect 587.6
Connection length [replies/conn]: 1.000

Request rate: 199.0 req/s (5.0 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 166.2 avg 166.2 max
166.2 stddev 0.0 (1 samples)
Reply time [ms]: response 983.4 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=831 4xx=0 5xx=0

CPU time [s]: user 0.10 system 5.30 (user 1.6%
system 88.8% total 90.4%)
Net I/O: 102.1 KB/s (0.8*10^6 bps)

Errors: total 19169 client-timo 356 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18813 addrunavail 0 ftab-full 0
other 0
```

NQ-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 1029

Total: connections 5365 requests 5217 replies 4620
test-duration 15.020 s

Connection rate: 357.2 conn/s (2.8 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 14.1 avg 1586.8 max
8262.9 median 1121.5 stddev 1404.0
Connection time [ms]: connect 499.9
Connection length [replies/conn]: 1.000

Request rate: 347.3 req/s (2.9 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 160.2 avg 307.8 max
410.0 stddev 131.0 (3 samples)
Reply time [ms]: response 1042.6 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4620 4xx=0 5xx=0

CPU time [s]: user 0.29 system 13.45 (user 1.9%
system 89.5% total 91.5%)
Net I/O: 219.4 KB/s (1.8*10^6 bps)

Errors: total 15380 client-timo 745 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 14635 addrunavail 0 ftab-full 0
other 0
```

NQ-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 1637

Total: connections 1963 requests 1963 replies 1315
test-duration 7.556 s

Connection rate: 259.8 conn/s (3.8 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 143.5 avg 1295.7 max
3975.0 median 955.5 stddev 968.8
Connection time [ms]: connect 507.7
Connection length [replies/conn]: 1.000

Request rate: 259.8 req/s (3.8 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 257.6 avg 257.6 max
257.6 stddev 0.0 (1 samples)
Reply time [ms]: response 670.2 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1315 4xx=0 5xx=0

CPU time [s]: user 0.09 system 6.95 (user 1.2%
system 92.0% total 93.1%)
Net I/O: 128.2 KB/s (1.1*10^6 bps)

Errors: total 18685 client-timo 648 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18037 addrunavail 0 ftab-full 0
other 0
```

NQ-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 3538

Total: connections 1527 requests 1527 replies 960
test-duration 6.480 s

Connection rate: 235.7 conn/s (4.2 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 594.4 avg 1003.3 max
3079.8 median 939.5 stddev 335.3
Connection time [ms]: connect 312.0
Connection length [replies/conn]: 1.000

Request rate: 235.7 req/s (4.2 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 192.0 avg 192.0 max
192.0 stddev 0.0 (1 samples)
Reply time [ms]: response 689.2 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=960 4xx=0 5xx=0

CPU time [s]: user 0.11 system 5.97 (user 1.7%
system 92.2% total 93.8%)
Net I/O: 110.1 KB/s (0.9*10^6 bps)

Errors: total 19040 client-timo 567 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18473 addrunavail 0 ftab-full 0
other 0
```

LC-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 435

Total: connections 5411 requests 5389 replies 4800
test-duration 15.403 s

Connection rate: 351.3 conn/s (2.8 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 16.4 avg 1721.1 max
8527.7 median 1324.5 stddev 1138.5
Connection time [ms]: connect 694.3
Connection length [replies/conn]: 1.000

Request rate: 349.9 req/s (2.9 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 124.6 avg 319.8 max
432.2 stddev 169.7 (3 samples)
Reply time [ms]: response 1028.8 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4800 4xx=0 5xx=0

CPU time [s]: user 0.27 system 13.70 (user 1.7%
system 88.9% total 90.7%)
Net I/O: 222.2 KB/s (1.8*10^6 bps)

Errors: total 15200 client-timo 611 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 14589 addrunavail 0 ftab-full 0
other 0
```

LC-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 1253

Total: connections 1816 requests 1816 replies 1238
test-duration 6.470 s

Connection rate: 280.7 conn/s (3.6 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 66.9 avg 1261.4 max
3977.4 median 913.5 stddev 1019.2
Connection time [ms]: connect 500.6
Connection length [replies/conn]: 1.000

Request rate: 280.7 req/s (3.6 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 247.5 avg 247.5 max
247.5 stddev 0.0 (1 samples)
Reply time [ms]: response 651.8 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1238 4xx=0 5xx=0

CPU time [s]: user 0.08 system 5.92 (user 1.3%
system 91.4% total 92.7%)
Net I/O: 140.7 KB/s (1.2*10^6 bps)

Errors: total 18762 client-timo 578 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18184 addrunavail 0 ftab-full 0
other 0
```

LC-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 2877

Total: connections 1464 requests 1464 replies 1164
test-duration 6.462 s

Connection rate: 226.6 conn/s (4.4 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 710.2 avg 1420.8 max
3962.0 median 1119.5 stddev 887.1
Connection time [ms]: connect 655.1
Connection length [replies/conn]: 1.000

Request rate: 226.6 req/s (4.4 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 232.8 avg 232.8 max
232.8 stddev 0.0 (1 samples)
Reply time [ms]: response 700.9 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1164 4xx=0 5xx=0

CPU time [s]: user 0.18 system 5.65 (user 2.7%
system 87.4% total 90.1%)
Net I/O: 130.0 KB/s (1.1*10^6 bps)

Errors: total 18836 client-timo 300 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18536 addrunavail 0 ftab-full 0
other 0
```

LBLC-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 1276

Total: connections 4781 requests 4719 replies 3183
test-duration 14.948 s

Connection rate: 319.8 conn/s (3.1 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 76.0 avg 989.8 max 7725.2
median 614.5 stddev 1024.9
Connection time [ms]: connect 414.8
Connection length [replies/conn]: 1.000

Request rate: 315.7 req/s (3.2 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 216.7 avg 294.0 max
371.3 stddev 109.3 (2 samples)
Reply time [ms]: response 611.4 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=3183 4xx=0 5xx=0

CPU time [s]: user 0.29 system 13.00 (user 1.9%
system 87.0% total 88.9%)
Net I/O: 156.8 KB/s (1.3*10^6 bps)

Errors: total 16817 client-timo 1598 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 15219 addrunavail 0 ftab-full 0
other 0
```

LBLC-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 1569

Total: connections 1730 requests 1730 replies 1029
test-duration 7.492 s

Connection rate: 230.9 conn/s (4.3 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 356.5 avg 1206.7 max
4805.2 median 832.5 stddev 954.6
Connection time [ms]: connect 455.3
Connection length [replies/conn]: 1.000

Request rate: 230.9 req/s (4.3 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 205.0 avg 205.0 max
205.0 stddev 0.0 (1 samples)
Reply time [ms]: response 577.3 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1029 4xx=0 5xx=0

CPU time [s]: user 0.11 system 7.11 (user 1.5%
system 94.9% total 96.4%)
Net I/O: 102.9 KB/s (0.8*10^6 bps)

Errors: total 18971 client-timo 701 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18270 addrunavail 0 ftab-full 0
other 0
```

LBLC-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 3501

Total: connections 1506 requests 1495 replies 798
test-duration 8.339 s

Connection rate: 180.6 conn/s (5.5 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 66.5 avg 927.6 max 3488.8
median 550.5 stddev 985.6
Connection time [ms]: connect 573.6
Connection length [replies/conn]: 1.000

Request rate: 179.3 req/s (5.6 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 159.6 avg 159.6 max
159.6 stddev 0.0 (1 samples)
Reply time [ms]: response 401.4 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=798 4xx=0 5xx=0

CPU time [s]: user 0.08 system 7.66 (user 1.0%
system 91.9% total 92.8%)
Net I/O: 72.9 KB/s (0.6*10^6 bps)

Errors: total 19202 client-timo 708 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18494 addrunavail 0 ftab-full 0
other 0
```

LBLCR-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 1624

Total: connections 4258 requests 4230 replies 4032
test-duration 15.544 s

Connection rate: 273.9 conn/s (3.7 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 36.9 avg 2402.3 max
7710.7 median 2174.5 stddev 1285.4
Connection time [ms]: connect 897.4
Connection length [replies/conn]: 1.000

Request rate: 272.1 req/s (3.7 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 173.8 avg 268.5 max
331.8 stddev 83.5 (3 samples)
Reply time [ms]: response 1525.7 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4032 4xx=0 5xx=0

CPU time [s]: user 0.35 system 12.38 (user 2.2%
system 79.7% total 81.9%)
Net I/O: 183.7 KB/s (1.5*10^6 bps)

Errors: total 15968 client-timo 226 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 15742 addrunavail 0 ftab-full 0
other 0
```

LBLCR-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 3129

Total: connections 1843 requests 1843 replies 1234
test-duration 7.482 s

Connection rate: 246.3 conn/s (4.1 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 403.5 avg 1124.9 max
3956.7 median 1030.5 stddev 560.5
Connection time [ms]: connect 314.5
Connection length [replies/conn]: 1.000

Request rate: 246.3 req/s (4.1 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 246.8 avg 246.8 max
246.8 stddev 0.0 (1 samples)
Reply time [ms]: response 780.9 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1234 4xx=0 5xx=0

CPU time [s]: user 0.10 system 6.58 (user 1.3%
system 88.0% total 89.3%)
Net I/O: 121.5 KB/s (1.0*10^6 bps)

Errors: total 18766 client-timo 609 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18157 addrunavail 0 ftab-full 0
other 0
```

LBLCR-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 2595

Total: connections 1472 requests 1472 replies 1466
test-duration 5.251 s

Connection rate: 280.3 conn/s (3.6 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 508.9 avg 1558.3 max
3276.6 median 1319.5 stddev 808.7
Connection time [ms]: connect 528.2
Connection length [replies/conn]: 1.000

Request rate: 280.3 req/s (3.6 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 293.2 avg 293.2 max
293.2 stddev 0.0 (1 samples)
Reply time [ms]: response 1028.7 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1466 4xx=0 5xx=0

CPU time [s]: user 0.12 system 4.62 (user 2.3%
system 87.9% total 90.2%)
Net I/O: 196.9 KB/s (1.6*10^6 bps)

Errors: total 18534 client-timo 6 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18528 addrunavail 0 ftab-full 0
other 0
```

DH-20000 Connection–2000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=2000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 273

Total: connections 4826 requests 4553 replies 3677
test-duration 17.380 s

Connection rate: 277.7 conn/s (3.6 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 49.7 avg 1458.4 max
8016.7 median 829.5 stddev 1414.5
Connection time [ms]: connect 661.5
Connection length [replies/conn]: 1.000

Request rate: 262.0 req/s (3.8 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 63.8 avg 244.7 max
341.8 stddev 156.8 (3 samples)
Reply time [ms]: response 888.8 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=3677 4xx=0 5xx=0

CPU time [s]: user 0.34 system 16.39 (user 2.0%
system 94.3% total 96.3%)
Net I/O: 152.4 KB/s (1.2*10^6 bps)

Errors: total 16323 client-timo 1149 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 15174 addrunavail 0 ftab-full 0
other 0
```

DH-20000 Connection–8000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=8000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 1985

Total: connections 2251 requests 2226 replies 1589
test-duration 8.380 s

Connection rate: 268.6 conn/s (3.7 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 147.6 avg 968.2 max
8170.3 median 232.5 stddev 1950.1
Connection time [ms]: connect 409.0
Connection length [replies/conn]: 1.000

Request rate: 265.6 req/s (3.8 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 299.0 avg 299.0 max
299.0 stddev 0.0 (1 samples)
Reply time [ms]: response 483.5 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=1589 4xx=0 5xx=0

CPU time [s]: user 0.11 system 8.00 (user 1.3%
system 95.5% total 96.8%)
Net I/O: 138.6 KB/s (1.1*10^6 bps)

Errors: total 18411 client-timo 662 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 17749 addrunavail 0 ftab-full 0
other 0
```

DH-20000 Connection–14000 Connection/s

```
httperf --hog --timeout=5 --client=0/1 --
server=202.103.106.5 --port=80 --uri=/ --rate=14000
--send-buffer=4096 --recv-buffer=16384 --num-
conns=20000 --num-calls=1
Maximum connect burst length: 2164

Total: connections 1330 requests 1290 replies 709
test-duration 8.255 s

Connection rate: 161.1 conn/s (6.2 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 274.8 avg 1566.6 max
5929.1 median 630.5 stddev 1378.0
Connection time [ms]: connect 796.2
Connection length [replies/conn]: 1.000

Request rate: 156.3 req/s (6.4 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 141.2 avg 141.2 max
141.2 stddev 0.0 (1 samples)
Reply time [ms]: response 374.2 transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=709 4xx=0 5xx=0

CPU time [s]: user 0.13 system 8.06 (user 1.6%
system 97.6% total 99.2%)
Net I/O: 65.1 KB/s (0.5*10^6 bps)

Errors: total 19291 client-timo 621 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 18670 addrunavail 0 ftab-full 0
other 0
```