

IMPLEMENTASI DAN ANALISA LOAD BALANCING WEB SERVER MENGUNAKAN METODE DISPATCHER RECEIVER PULL BASED

Dhuka Dwi Cahyanto¹, Tri Brotoharsono², Endro Ariyanto³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Meningkatnya penggunaan internet pada saat ini mengakibatkan stress pada web server sehingga performansi web server menjadi menurun. Di lain sisi, user mengharapkan low latency, high throughput dan high availability. Load balancing merupakan salah satu teknik untuk meningkatkan performansi distribusi web server. Load balancing dapat menyediakan high availability melalui redundant server dan mengurangi latency dengan membagi load diantara back end server.

Banyak algoritma dan teknik untuk mengimplementasikan load balancing. Salah satu teknik adalah penerapan dispatcher yang difungsikan sebagai controlling. Pada umumnya dispatcher ini menjadi penentu pemilihan back end server yang akan melayani request sesuai algoritma yang diterapkan. Dispatcher ini mendorong request ke back end server. Pada tugas akhir ini, bertujuan untuk mengimplementasikan load balancing dengan teknik dispatcher, namun dispatcher pada tugas akhir ini berfungsi sebagai central queue untuk menyimpan request dari client. Keputusan siapa yang akan melayani request ditentukan sendiri oleh back end server berdasarkan utilitas cpu yang dimiliki.

Sistem dispatcher yang dihasilkan kemudian di uji dengan parameter nilai throughput yang hasilnya dibandingkan dengan sistem dispatcher menggunakan algoritma round robin. Hasil pengujian menunjukkan bahwa, overhead yang dialami pada sistem receiver pull based menyebabkan response rate yang dihasilkan tidak maksimal sehingga nilai throughput yang didapatkan lebih kecil dibandingkan sistem dispatcher round robin

Kata Kunci : load balancing, dispatcher, central queue, pull based

Abstract

Nowadays, the use of internet has increased rapidly. This situation has caused stress to web server so that the performance of web server to decrease. On the other hand, the user wants low latency, high throughput and high availability. Load balancing is a technique for improving web server performance distribution. Load balancing can provide high availability through redundant server and reduce latency by dividing load between the back end server.

There are many algorithms and techniques for implementing load balancing. One technique is the application of which functioned as the controlling dispatcher. In most cases this dispatcher also determines the selection back end server that will serve the request based on the algorithm applied. These dispatchers push request to the back end server. The aims in this final task is to implement load balancing technique with dispatchers, but dispatchers in this final act as central queue to store requests from the client. The decision who will serve the request is determined solely by the back end server based on utilities of CPU that owned. The result of dispatchers system then is being tested with the parameter value of throughput. After that, the results compared with the system dispatchers using round robin algorithm. The results show that, overhead experienced in the pull-based receiver system causing the resulting response rate is not maximal so that the values obtained throughput is smaller than dispatchers' round robin system.

Keywords : load balancing, dispatcher, central queue, pull based

1 Pendahuluan

1.1 Latar belakang

Meningkatnya pengguna internet pada saat ini, mendorong akan meningkatnya trafik internet. Internet sudah menjadi sebuah kebutuhan yang harus dipenuhi. Aplikasi internet seperti *web sites* sudah menjadi bagian dari kehidupan dengan kelebihanannya yang dapat diakses dari manapun. Konten web site semakin interaktif dan semakin beragam, bahkan saat ini sudah digunakan pada teknologi transaksi online, hal ini tentunya berdampak pada *request rate* dari sebuah web sites terlebih web sites populer yang semakin hari semakin meningkat.

Dengan kondisi seperti ini akan mengakibatkan *stress* pada sebuah web server karena meningkatnya *cpu usage* pada saat melayani request konten dari user. Di lain sisi, user mengharapkan *low latency*, *high throughput* dan *high availability*. Distribusi web server menyediakan solusi efektif untuk meningkatkan kualitas dari web server. Kumpulan dari beberapa web server digunakan sebagai kesatuan resource untuk menyediakan service ke client. *Load balancing* merupakan salah satu teknik untuk meningkatkan performansi dari distribusi web server. *Load balancing* dapat menyediakan *high availability* melalui *redundant server* dan mengurangi *latency* dengan membagi load diantara server yang dikenal dengan *load sharing*. Banyak algoritma dan teknik untuk mengimplementasikan *load balancing*. Pada umumnya strategi *load balancing* yang ada, request dialihkan ke server, dimana server dipaksa untuk menerima request dalam kondisi apapun.

Pada tugas akhir ini, mencoba melakukan pendekatan dengan memanfaatkan kelebihan dari metode *dispatcher* sebagai *full control* yang akan digantikan dengan sebuah proxy untuk menyimpan antrian request dari client. Namun pemilihan server untuk memproses request dilakukan oleh server-server itu sendiri dengan melihat kondisi resource server. Server sendiri yang nantinya akan menentukan apakah dirinya akan memproses request yang ada di antrian atau tidak. Dengan metode seperti ini diharapkan mendapatkan performansi yang lebih baik, terlebih dalam lingkungan hardware yang tidak sama

Telkom
University

1.2 Perumusan masalah

Masalah yang akan diselesaikan dalam tugas akhir ini adalah

1. Bagaimana mengaplikasikan teknik yang berbeda dalam *load balancing* web server yaitu *dispatcher receiver pull based* dimana *dispatcher* akan digantikan dengan metode antrian
2. Bagaimana membangun sebuah *load balancing* yang *fair share* sesuai dengan *current load* diantara web server terlebih spesifikasi hardware diantara web server yang berbeda-beda sehingga mempengaruhi *response time* dan *request rate* server dalam memproses request dari client dan akhirnya meningkatkan performansi web server dalam hal *throughput*.

Ruang lingkup yang membatasi tugas akhir ini adalah

1. Metode antrian yang akan digunakan adalah *single central* antrian, yaitu hanya ada satu server antrian yang dijadikan sebagai *proxy* dari distribusi web server
2. Untuk menentukan apakah web server dapat melayani request dari client atau tidak akan digunakan sebuah parameter atau metrik yaitu nilai utilisasi cpu web server kurang dari batas maksimal *threshold* cpu utilisasi yang telah ditentukan sehingga dinyatakan dalam kondisi *health*. Jika utilisasi cpu web server melebihi batas maksimal *threshold* cpu utilisasi maka, web server tidak dapat melakukan pelayanan request dan harus menunggu sampai kondisinya dalam keadaan *health*.
3. Implementasi tugas akhir ini menggunakan sistem operasi linux

1.3 Tujuan

Tujuan dari penelitian Tugas Akhir ini adalah mengimplementasikan metode *dispatcher receiver pull based* dan membandingkan dengan metode *dispatcher load balancing* yang menggunakan algoritma *Round Robin* dan menganalisa performansi server dalam hal *throughput*.

1.4 Metodologi penyelesaian masalah

Dalam penelitian Tugas Akhir ini, metode yang digunakan untuk menyelesaikan Tugas Akhir sebagai berikut :

1. Studi literatur

Mengumpulkan bahan-bahan pustaka sebagai referensi tugas akhir dan mempelajari literatur-literatur yang relevan dengan permasalahan meliputi melakukan studi pustakan dan referensi mengenai konsep dari load balancing, algoritma load balancing dan penerapannya.

2. Pengembangan aplikasi yang meliputi :

1. Analisa dan perancangan

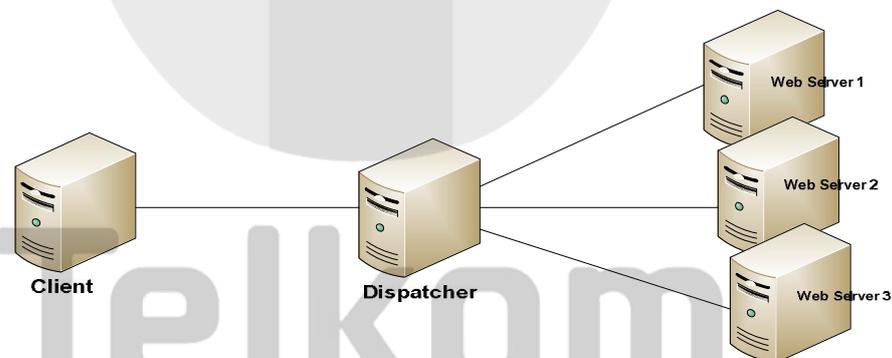
Memodelkan perancangan aplikasi ke dalam suatu diagram berdasarkan analisis dan desain

2. Pengkodean

Mengimplementasikan perancangan menjadi sebuah aplikasi.

3. Pengujian aplikasi

Melakukan pengujian dari sistem yang telah dibangun pada tahap implementasi kemudian menganalisa performansi dengan bantuan beberapa tool dan dibandingkan dengan sistem tradisional. Pengujian menggunakan sistem operasi linux yang terdiri dari beberapa web server dan satu sebagai dispatcher yang merupakan single central antrian. Topologi seperti gambar dibawah ini



4. Analisis hasil

Menganalisa hasil dari percobaan yang akan dibandingkan dengan sistem tradisional load balancing

5. Pembuatan laporan

Mendokumentasikan tahap-tahap yang dilakukan selama penelitian dan melaporkan hasil penelitian dalam tugas akhir ini

5 Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil pembangunan sistem ini serta hasil dari uji coba yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut :

1. Interval waktu pengukuran utilisasi cpu berpengaruh terhadap nilai response rate yang dihasilkan. Semakin besar file distribusi maka interval waktu untuk mendapatkan response rate yang maksimal semakin besar.
2. Hipotesa awal bahwa dispatcher receiver pull based akan mendapatkan performansi throughput yang lebih baik tidak didapatkan dari percobaan tugas akhir ini. Terjadinya overhead pada sistem receiver pull based menyebabkan response rate rata-rata yang dihasilkan tidak maksimal sehingga mengakibatkan nilai throughput lebih kecil dibandingkan dengan sistem dispatcher round robin. Lama waktu proses setiap requestnya pada sistem dispatcher receiver pull based masih terlalu tinggi yaitu sebesar 0.06 secs jika dibandingkan dengan lama waktu proses pada sistem dispatcher menggunakan algoritma round robin yang hanya 0,01 secs.

5.2 Saran

Untuk pembangan penelitian Tugas Akhir ini, saran yang dapat diberikan adalah sebagai berikut :

1. Untuk mengurangi overhead dan mengurangi delay pada saat pengambilan job, sebaiknya digunakan banyak worker pull job request pada saat bersamaan
2. Untuk mengurangi overhead dan mempercepat proses pada sistem dispatcher receiver pull based, sebaiknya dilakukan *interrupt* adanya job terhadap back end server sehingga delay pengambilan job dapat dikurangi
3. Untuk mengurangi overhead pada dispatcher, relay response dapat digunakan metode lain tanpa melewati relay melalui dispatcher.

Telkom
University

Daftar Pustaka

- [1] Berggrem Arild. Presenting a Prototype for Pull Based Load Balancing for WebServer
- [2] Bryhni Haakon Kure Ivind, Klovnin Espen. A comparison of load balancing techniques for scalable web server. IEEE Network, 200
- [3] Candara Kopparapu. Load balancing server, firewalls, dan caches. John Wiley and Sons.
- [4] J.A.Lacord,A.Pont,J.A gil, J.Sahuquillo., A Comprehensive Web Workload Characterization
- [5] Request for Comments(RFC) : 2616, Hypertext Transfer Protocol – HTTP/1.1
- [6] S. Yu, Philip., Colajanni, Michele.,Cardellini, Varleria., Dynamic Load Balancing on Web Server Systems
- [7] SIEGE-2.6.2009, <http://www.joedog.org/index/siege-home>
- [8] Teo Ment Yong Ayani Rassul. Comparison of load balancing strategies on cluster-based web server. Transaction of the Society for Modeling and Simulation, 2001