SIMULASI PERFORMANSI WEB CLUSTER

Reni Listiana
Jurusan Teknik Elektro Politeknik TEDC Bandung
E-mail: renilistiana@lycos.com

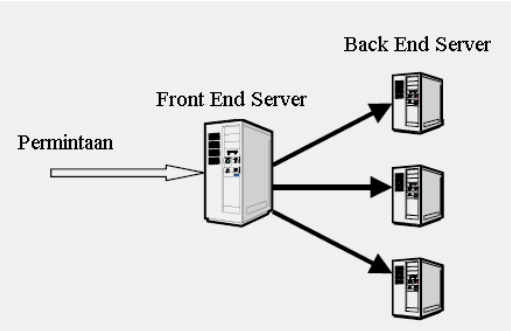
Abstrak
Pada masa akhir ini semakin mudah orang mencari informasi melalui internet dan semakin banyak dibangun web server.Beberapa solusi untuk peningkatan performansi telah muncul misalnya memperbesar lebar pita di sisi penerima, penggunaan caching (akses memory) baik di sisi client maupun di sisi server, optimasi topologi jaringan yaitu penggunaan cluster pada sisi server. Dengan sistem cluster ini maka penyedia layanan web site bisa menjadikan sistemnya skalabel di mana jika peminat terhadap web site meningkat maka penyedia layanan tersebut dapat menambahkan server secara terorganisir demikian pula jika peminatnya terus menurun padahal kemampuan server-servernya jauh lebih tinggi respon pelayanan dibandingkan dengan permintaan yang dating maka penyedia situs tersebut dapat mengurangi jumlah servernya. Disajikan model web cluster yang menggunakan admission control dengan buffer dan pendistribusian beban fewest server. Dari simulasi tersebut diperoleh performansi sistem web cluster yang menggunakan admission control dengan buffer maka dengan permintaan yang terus berdatangan akan diantrikan ke buffer dan apabila tidak dapat diproses maka akan dicek untuk dibuang pada satuan waktu berikutnya. Maka bisa diperkirakan seberapa jauh server dapat melayani permintaan dengan server-server yang tersedia dan bisa diperkirakan seberapa banyak server yang dibutuhkan dengan kualitas pelayanan yang dimiliki oleh server tersebut.

Katakunci : WEB Cluster, simulasi, performansi, server, database,

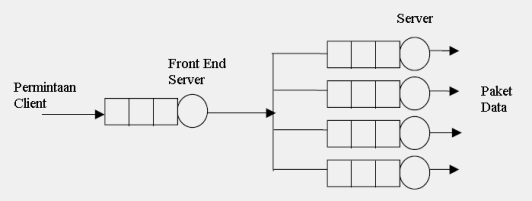
Pendahuluan

Dengan kemajuan teknologi akhir-akhir ini banyak perusahaan yang menggunakan internet untuk menyediakan data maupun aplikasi untuk dapat diakses oleh client dari berbagai tempat, dalam hal ini jumlah client bisa dibatasi dari batasan tertentu sampai dengan jumlah yang sangat besar bahkan idealnya dalam jumlah seolah-olah tak terbatas.

Pelayanan web site menjadi hal yang penting di dalam internet.Pada situs-situs yang terkenal memungkinkan terjadinya trafik yang melonjak dibandingkan dengan jumlah trafik sebelumnya, sehingga situs tersebut bisa menjadi overload. Oleh karena itu penting untuk mempunyai mekanisme control terhadap load atau beban. Tujuan dari control tersebut adalah untuk mengurangi beban pada server, di sini menggunakan metoda fewest server yaitu sebuah metoda yang berusaha memanfaatkan suatu server dalam hal ini kalau satu server bisa memegang semua permintaan yang datang, maka permintaan tersebut akan dikendalikan oleh server tersebut.



Gambar 1a



Gambar 1b.

Gambar 1a dan 1b, Arsitektur Dasar Metoda Fewest Server

Selain mekanis mekontrol yang telah disebutkan di atas, maka penggunaan sistem clustering tidak kalah penting dalam upaya untuk meningkatkan performansi dan skalabilitas di mana jika suatu server mengalami peningkatan pengunjung maka bisa diatur dengan cara menambah server pada cluster.

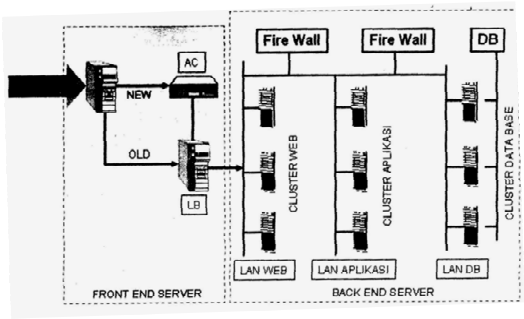
Batasan Masalah

Batasan masalah secara umum adalah sebagai berikut:

- Client yang dimaksud adalah user atau pelanggan atau konsumen yang mengakses server
- Server yang dimaksud adalah penyedia data atau aplikasi maupun layanan lain
- Distribusi kedatangan sebagai contoh adalah distribusi random
- Menggunakan admission control dengan buffer
- Algoritma pada load balancer adalah fewest server

Sistem Web Cluster

Sistem web cluster adalah suatu sistem di mana server-server yang tersedia menduduki cluster-secara teratur. Hal ini menjadikan cluster tersebut scalable, maksudnya adalah penyedia web site tersebut bisa menambah atau mengurangi server secara terorganisir.



Gambar 2. Arsitektur sistem web cluster

Sistem dari server yang dikaji di sini meliputi admission control (AC), load balancer (LB), dan beberapa cluster yaitu :

- cluster web yang mencakup beberapa server web
- cluster aplikasi yang mencakup beberapa server aplikasi, dan
- cluster database yang mencakup beberapa server database

Untuk permintaan baru, secara otomatis melalui admission control, jika permintaan total masih dalam jumlah yang terpenuhi dibandingkan dengan kapasitas maksimum yang mampu diterima oleh server maka permintaan tersebut diterima dan diteruskan ke load balancer untuk didistribusikan ke web server melalui LAN Web. UUntuk pengunjung lama, maka secara otomatis langsung dilewatkan ke load balancer yang selanjutnya juga diteruskan ke web server melalui LAN Web.

Server-server pada LAN yang satu menjalankan tugas yang berbeda dengan server-server pada cluster yang lain. Sebuah server web menjalankan software daemon HTTP. Software tersebut membaca permintaan HTTP dari Client, menyediakan isi yang diminta oleh Client, dan mengirimkan kembali isi dari yang diminta oleh Client. Ketika server web menyediakan isi yang diminta, server web tersebut mungkin memerlukan program aplikasi yang ada di server aplikasi. Sebuah server aplikasi menjalankan software yang menangani semua operasi aplikasi antara browser konsumen dan data base dari penyedia layanan site tersebut.

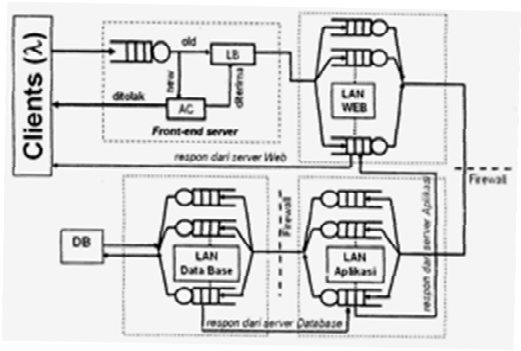
Server database mengatur database, Server tersebut menyediakan fungsi-fungsi manipulasi data kepada server aplikasi. Server aplikasi memanggil server databaseuntuk mengakses dan mengatur database. Setiap server yang menyediakan layanan yang sama dihubungkan dengan sebuah LAN, di sini dinotasikan sebagai WebLAN, ApplicationLAN, dan DatabaseLAN.

Sebuah mesin Firewall menghubungkan WebLAN dan ApplicationLAN. Demikian juga ApplicationLAN dan DatabaseLAN dihubungkan dengan Firewall.

Pada Firewall yang menghubungkan WebLAN dan ApplicationLAN maka hanya trafik antara server web dan server aplikasi sajalah yang dapat melaluinya. Demikian juga Firewall yang menghubungkan ApplicationLAN dan DatabaseLan hanya memperbolehkan trafik antara server aplikasi dan server database. Kedua Firewall ((bridge) tersebut meningkatkan keamanan bagi site e-commerce. Server database hanya dapat dihubungi oleh server aplikasi, sedangkan server aplikasi hanya dapat dihubungi oleh server web. Server web tidak diijinkan mengakses database secara langsung.

Model Web Cluster

Untuk menganalisa model web cluster, dimodelkan web cluster dengan teorema jaringan Jackson atau dikenal dengan jaringan terbuka sebagai berikut :



Gambar 3. Model Antrian dari Web Cluster

Untuk menganalisa model web ini dibagi menjadi 3 node yaitu server web, server aplikasi, dan server database yang merupakan tiga titik node yaitu server web, server aplikasi, dan server database yang merupakan tiga titik yang ditinjau performansinya. Untuk menganalisa digunakan teori antrian M/M/c series, kedatangan client dalam pola distribusi Poisson, memberikan permintaan ke web server yang dalam kerjanya memerlukan aplikasi yang diambil dari server aplikasi, dan server aplikasi akan mengambil data ke penyimpan data atau server database.

Laju waktu antar kedatangan adalah $1/\lambda$ dan waktu pelayanan $1/\mu$ terdistribusi secara eksponensial dan c adalah jumlah server. Pelangganan dilayani dalam orde kedatangan dan kemampuan server melayani dinyatakan dalam rumus :

$$\rho = \frac{\lambda}{c\mu} < 1$$

Source code sebagai berikut :

```
....
RVector(i) = lambda(k) / (MuVector(i)* NumSrvVector(i));
....
```

Dari aliran masuk dan keluar dengan {0, 1,…., n-1} aliran dari dua titik yang berdekatan pada n-1 dan n memberikan :

$$\lambda p_{n-1} = n\,\mu\,p_n, \qquad n \le c$$

$$\lambda p_{n-1} = m\,\mu\,p_n, \qquad n > c$$

Iterasi memberikan:

$$p_n = \frac{(c\rho)^n}{n!} p_0, \qquad n \le c$$

$$p_n = \frac{c^c \rho^n}{c!} p_0, \qquad n > c$$

Probabilitas pomenghasilkan :

$$p_0 = \left(\sum_{n=0}^{c-1} \frac{(c\rho)^n}{n!} p_0 + \frac{(c\rho)}{c!} \cdot \frac{1}{1-\rho} \right)^{-1}$$

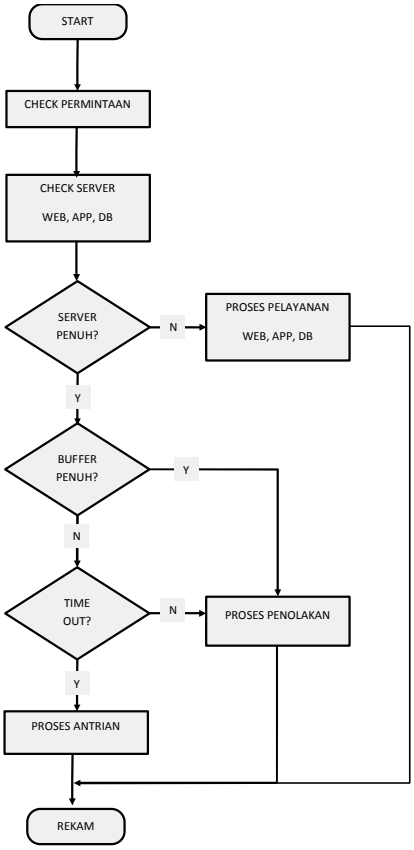
Programnya sebagai berikut :

```
....
for n = 1:(NumSrvVector(i) - 1
    dblWork2(i) = dblWork2(i) * RVector(i) / n;
    dblWork1(i) = dblWork1(i) + dblWork2(i);
end
dblWork2(i) = dblWork2(i) * RVector(i) / NumSrvVector(i);
p0Vector(i)=
1/(dblWork1(i)+dblWork2(i)/(1(RVector(i)/NumSrvVector(i))));
...
```

Simulasi

Program utama dimulai dengan membaca data masukan untuk simulasi, kemudian mengeset data-data tersebut. Proses dijalankan secara berturutan yaitu: proses kedatangan permintaan, dilanjut dengan proses di AC (admission control), load balancing yang mendistribusikan ke server-server di web, load balancing ke server aplikasi (apabila membutuhkan layanan aplikasi), load ke server database (apabila membutuhkan pelayanan database), kemudian mengeset tampilan pada file output. Apabila server-server yang diperlukan masih mampu menerima permintaan tersebut maka permintaan disalurkan ke load balancing untuk didistribusikan ke server sesuai dengan metoda yang berlaku pada load balancing.

Diagaram alir program dengan admission control memakai buffer sebagai berikut:

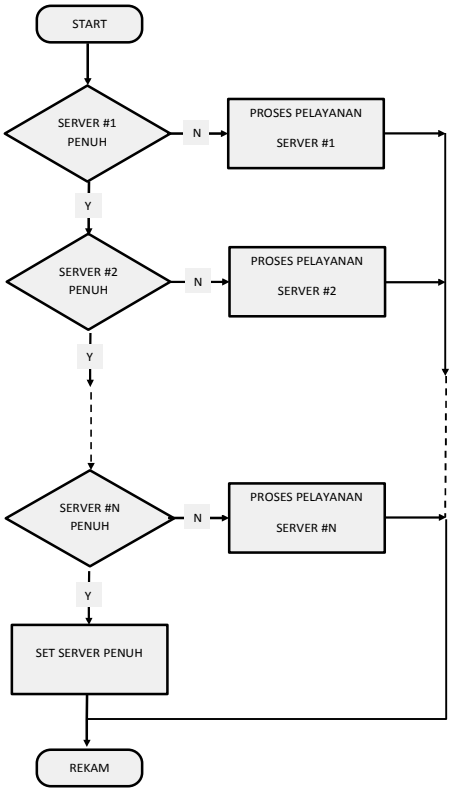


Gambar 4. Diagram alir program dengan admission control memakai buffer

Proses admission control dengan buffer sebagai berikut :

- Mengecek jenis permintaan dan mengecek apakah server-server yang dibutuhkan sibuk. Apabila server-server yang dibutuhkan masih mampu menampung permintaan tersebut untuk diproses maka permintaan diterima dan disalurkan ke load balancing untuk didistribusikan sesuai dengan metoda yang berlaku pada load balancing.
- Apabila server-server yang diperlukan ada yang sibuk (untuk satu satuan waktu tertentu) maka dicek apakah buffer penuh.
- Apabila buffer penuh maka permintaan tersebut ditolak; apabila buffer masih bisa menampung permintaan, maka permintaan tersebut ditampung untuk diproses pada kurun waktu tertentu (maksimal satu satuan waktu yang telah ditentukan). Permintaan yang ditampung diproses dengan aturan First In First Out (FIFO).
- Untuk satu satuan waktu yang telah ditentukan maka sisa permintaan yang ada di buffer ditolak (agar client tidak menunggu terlalu lama respon dari permintaan)

Diagram alir untuk load balancing sebagai berikut:



Gambar 5. Diagram alir load balancing dengan metode fewest servers

Pendistribusian beban dengan metoda fewest servers menggunakan aliran proses sebagai berikut :

- Permintaan yang datang diprioritaskan ke server pertama (prioritas yang tinggi), apabila server prioritas pertama sudah penuh maka permintaan baru didistribusikan ke server prioritas kedua, demikian seterusnya apabila server prioritas yang lebih tinggi ada yang kosong (mampu menampung permintaan baru dalam satu satuan waktu tertentu) maka permintaan didistribusikan ke server prioritas lebih tinggi tersebut.
- Demikian seterusnya sehingga bisa diketahui sejumlah minimal server yang dibutuhkan untuk melayani sejumlah rate kedatangan permintaan

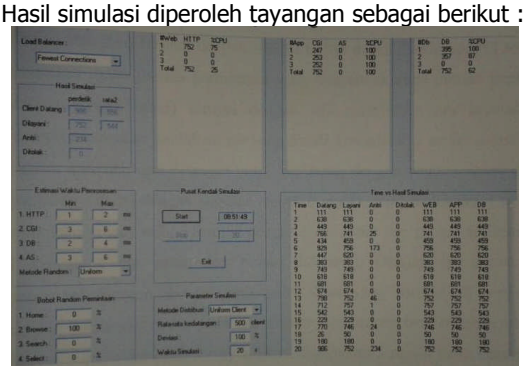
Load balancer bisa berupa software atau bisa juga berupa hardware.

Untuk load balancer yang berupa software mempunyai kelebihan yaitu harga bisa lebih murah dan memiliki banyak pilihan konfigurasi yang dapat disesuaikan dengan kebutuhan kita. Namun mempunyai kelemahan yaitu sebagian besar aplikasi tidak dapat menangani situs besar atau jaringan kompleks. Contoh load balancer berupa software yaitu: Linux Virtual Server, Ultra Monkey, Network Load Balancing.

Kelebihan load balancer berupa hardware adalah proses lalu-lintas pada tingkat jaringan lebih efisien dan biasanya bisa bekerja pada bermacam-macam OS atau platform, namun punya kelemahan yaitu biasanya biaya lebih mahal. Contoh load balancer berupa hardware : Cisco *System Catalyst*, Coyote Point, F5 Network BIG-IP, *Baraccuda Load Balancer*.

Hasil Simulasi

Hasil simulasi diperoleh tayangan sebagai berikut :



Gambar 6 Hasil Simulasi

Sedangkan tabel yang dihasilkan sebagai berikut :

Time	Datang	Layani	Antri	Ditolak	WEB	APP	DB
1	111	111	0	0	111	111	111
2	638	638	0	0	638	638	638
3	449	449	0	0	449	449	449
4	766	741	25	0	741	741	741
5	434	459	0	0	459	459	459
6	929	756	173	0	756	756	756
7	447	620	0	0	620	620	620
8	383	383	0	0	383	383	383
9	749	749	0	0	749	749	749
10	618	618	0	0	618	618	618
11	681	681	0	0	681	681	681
12	674	674	0	0	674	674	674
13	798	752	46	0	752	752	752
14	712	757	1	0	757	757	757
15	542	543	9	0	543	543	543
16	229	229	9	0	229	229	229
17	770	746	24	0	746	746	746
18	26	50	0	0	50	50	50
19	180	180	0	0	180	180	180
20	986	752	0	0	752	752	752

Gambar 7. Tabel Hasil Simulasi

Analisa

Pada percobaan ini digunakan jumlah server yang digunakan adalah 3 server untuk web, 3 server untuk aplikasi, dan 3 server untuk database; dengan waktu proses masing-masing 1 ampai 2 ms untuk tiap permintaan proses web, 3 sampai 6 ms untuk tiap proses aplikasi, dan 2 sampai 4 ms untuk tiap proses database. Dengan spesifikasi waktu pemrosesan tersebut maka sisi cluster web cukup bisa dilayani oleh 1 mesin server, sedangkan pada mesin aplikasi digunakan 3 mesin yang bekerja secara penuh (sibuk), dan server database melibatkan 2 mesin untuk melayani permintaan-permintaan tersebut.

Kesimpulan

Pada tulisan ini dipresentasikan sistem *web cluster* yang dimodelkan sebagai jaringan terbuka yang menerima permintaan dari *client* mana saja. Dengan model tersebut bisa disimulasikan performansi dari *server*.

Jika permintaan dari client, dalam hal ini *network arrival rate* (λ) terus meningkat dalam jumlah yang sangat tinggi maka akan menyebabkan konsumen mendapatkan waktu respon yang cukup lama bahkan bisa melebihi batas kesabaran konsumen untuk bersedia menunggu respon tersebut, untuk itu perlu diantisipasi dengan memberikan mekanisme tertentu dalam hal ini mekanisme kontrol menggunakan *admission control* untuk membatasi atau mengatur jumlah permintaan yang dating, serta menggunakan *load balancer* untuk mendistribusikan permintaan-permintaan ke server-servernya. Dengan sistem cluster maka penyedia layanan internet menjadikan sistem yang skalabel di mana penambahan maupun pengurangan server bisa diatur.

Daftar Pustaka

[1] Daniel A. Menasce, Virgilio A. 2000.*Scaling for E-Business (Technolology, Models, Performance, and Capacity Planning)*.Prentice Hall.

[2] Hendrawan Ph.D. 2004. Peningkatan Kinerja Load Balancing Dinamis Web Server Cluster. *Journal The 1st Converence on Telematics System, Services, and application.*

[3] Louis P. Slothouber Ph.D. 1997. *A Model of Wleb Server Performance*. Jurnal di internet

[4] Danang Taufik Karunia. 2002. Kajian Bisnis dan Implementasi *Web-Caching*. Bandung: Tesis Magister Program Pasca Sarjana ITB

[5] Yanggao Yang. 2000. *Load Balancing Technologies for E-Commerce Sites. Term paper for INST818*

[6] Maria Kihl, Niklasidel, *and* Christian Nyberg. *Performance Modeling and Control of s. E-Commerce Site*. Jurnal di internet www.telecom.lth.se