

Dual Load balancing menggunakan Algoritma Round-robin pada Cloud Computing

Awang Andhyka^{1*} dan Fawaidul Badri¹

¹Sistem Informasi, Fakultas Ilmu Komputer Universitas Nahdlatul Ulama Sidoarjo, Sidoarjo

*Corresponding Author

E-mail: awang85.si@unusida.ac.id

Abstrak

Downtime pada cloud computing seringkali memiliki dampak langsung penggunanya. Situs besar dapat kehilangan jutaan dana atau lebih dalam pendapatan setiap menit situs mereka tidak tersedia. Dengan menggunakan server load untuk balancing yang dimungkinkan memberikan keandalan yang lebih baik karena algoritma load pada balancing digunakan untuk meminimalkan kesalahan serta memungkinkan kemungkinan pada peningkatan dan pemeliharaan server tanpa mengganggu layanan yang ditawarkan. Jika server macet dan perlu dimatikan karena pemeliharaan, itu hanya akan menyebabkan penurunan kinerja, sementara layanan masih tersedia di server yang masih aktif. Penyedia layanan perlu menyesuaikan dengan perjanjian tingkat pada sebuah layanan yang menggambarkan batasan untuk layanan yang diberikan, seperti waktu respons, waktu aktif, bandwidth, dll. Karena padatnya komunikasi lintas jaringan dapat berubah dengan cepat sehingga sangat penting bagi penyedia layanan untuk dapat beradaptasi dengan perubahan yang mungkin terjadi karena untuk meningkatkan lalu lintas terhadap server. Load balancing menyediakan kemungkinan untuk menyesuaikan jaringan dengan permintaan yang banyak dan meningkat pada lalu lintas berdasarkan dinamika cloud computing.

Kata Kunci: *cloud computing, server, bandwidth, load balancer.*

Abstract

Downtime in cloud computing often has a direct impact on its users. Large sites can lose millions or more in revenue every minute their site is not available. By using server load balancing it is possible to provide better reliability because the load algorithm on balancing is used to minimize errors and allow the possibility of server upgrading and maintenance without disrupting the services offered. If the server is stuck and needs to be shut down due to maintenance, it will only cause a decrease in performance, while the service is still available on the server which is still active. Service providers need to adjust to the agreement level on a service that describes the restrictions for services provided, such as response time, active time, bandwidth, etc. Because the density of communication across the network can change quickly, it is very important for service providers to be able to adapt to changes that might occur due to increasing traffic to the server. Load balancing provides the possibility to adjust the network with a lot of demand and increase traffic based on the dynamics of cloud computing.

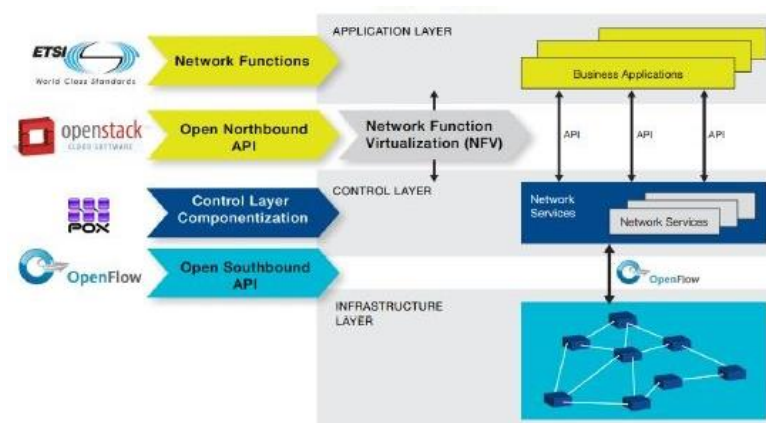
Keywords: *cloud computing, server, bandwidth, load balancer.*

1. PENDAHULUAN

Beban pada beban yang terdapat pada koneksi internet klien dikurangi menggunakan teknik prediksi dan dengan perawatan yang cermat dari wilayah yang diminati pemain. Teknik prediksi berkonsentrasi pada minimalisasi pembaruan status pemain. Pembaruan dikirim hanya ketika

tindakan klien belum diprediksi pada model perilaku yang terdiri dari tindakan klien sebelumnya (Alakeel, 2010). Contoh dasar dari perilaku tersebut adalah gerakan. Ketika seorang pemain melaporkan ke server bahwa ia bergerak pada lintasan tertentu, tidak ada pemberitahuan lebih lanjut diperlukan sampai lintasan itu berubah. Demikian pula, teknik wilayah kepentingan (RoI) di server mengirim informasi ke pemain hanya ketika informasi itu dapat bermanfaat dalam siklus pembaruan saat ini: pesan dari pemain belum mampu mempengaruhi pemain segera tidak disebarkan ke pemain itu. Dari perspektif server, dua masalah paling penting adalah bandwidth jaringan dan beban CPU. Karena total bandwidth jaringan server sebanding dengan penggunaan bandwidth rata-rata dari masing-masing pemain, yang disebutkan sebelumnya teknik prediktif dan RoI untuk setiap klien terdiri pada sebagian jaringan besar atau realokasi kota adalah kenyataan (B. De Vleeschauwer, 2005). Keputusan ini memerlukan beberapa perspektif, karena memengaruhi kepentingan pada pihak ketiga. Jenis masalah ini cenderung melibatkan beberapa organisasi, pemerintah, lingkungan dan perusahaan swasta untuk membuat keputusan akhir tentang lokasi. Keputusan ini dibuat supaya mencapai tujuan jangka menengah atau panjang dan perlu didasarkan pengurangan beban. Beban CPU di server harus seimbang dengan biaya RoI dan teknik prediksi, karena skema rumit mempengaruhi secara mudah menggeser hambatan dari koneksi jaringan ke biaya pemrosesan, tidak memberikan manfaat keseluruhan ke jumlah maksimum pemain (G. Boss, 2010).

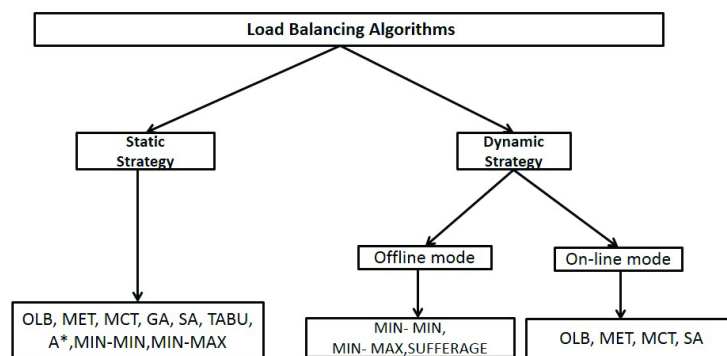
Load pada Balancing cloud server telah membawa paradigma paling baru dalam jaringan. Ini memisahkan bidang kontrol yang didistribusikan dari bidang data dan memindahkan bidang kontrol ke pengontrol terpusat. Dengan demikian, pengontrol memiliki pandangan lengkap tentang topologi jaringan ditambah kontrol penuh sumber daya jaringan. Bersama dengan kemampuan pemrograman pengontrol, SDN menawarkan cara yang efisien serta fleksibel untuk menghadirkan fungsi jaringan. Operator jaringan organisasi besar dan penyedia cloud menghadapi tantangan besar dalam membangun data center yang besar ke ukuran yang sebelumnya tidak terbayangkan yang mendukung ribuan sakelar dan server (Kazem, 2007). Karena pusat dari data dan aplikasinya terus tumbuh, memanfaatkan sumber pada daya di jaringan untuk meningkatkan kinerja pusat data menghadirkan tantangan tertentu. Beban kerja pusat data bervariasi dalam waktu dan ruang (R.Buyya, 2009). Akibatnya, alokasi sumber daya statis tidak efisien.



Gambar 1. Arsitektur cloud Network

Dalam datacenter baru-baru ini, desain bergantung pada multiplisitas jalur untuk mencapai penskalaan horizontal host. Untuk alasan ini, topologi pusat data sangat berbeda dari jaringan perusahaan pada umumnya. Dalam jaringan pusat data, sakelar akan berhubungan pada banyak

jalur untuk menyediakan redundansi. Untuk memanfaatkan bandwidth secara efisien, jaringan pusat data menggunakan TRILL, standar protokol penerusan data IETF, atau Fabric Path, protokol milik Cisco, untuk mengimplementasikan jalur Equal Cost Multiple Paths (ECMP). Dalam ECMP, (R.N.Calheiros, 2011) lalu lintas di-load secara seimbang beberapa jalur memiliki biaya sama dengan menetapkan aliran ke jalur yang berbeda. Pemilihan jalur biasanya berdasarkan hashing. Masalah dengan hashing atau variasinya, skema round robin, tidak menjamin keseimbangan muatan yang sebenarnya. Hal ini menghasilkan situasi yang tidak diinginkan bahwa beberapa tautan kelebihan permintaan sementara tautan lain kelebihan permintaan. Dalam tahun-tahun terakhir, SDN telah semakin diadopsi dalam jaringan pusat data. SDN memberikan solusi baru untuk masalah jaringan lama. Karena pandangan terpusat dari jaringan, SDN menyediakan platform yang lebih efisien untuk mengimplementasikan penyetelan beban ECMP (R.N.Calheiros, 2011). Dalam tesis ini, kami mengusulkan skema penyeimbangan beban dengan memanfaatkan karakteristik SDN. Implementasinya ringan tetapi memperkenalkan fitur yang lebih canggih daripada protokol yang ada. Di pusat pengiriman konten, ada banyak server replikasi yang dapat mengirimkan konten yang sama. Masalah penyeimbangan beban dalam kasus seperti itu tidak akan sepenuhnya diatasi tanpa mengatasi penyeimbangan beban server. Dalam jaringan non-SDN, penyeimbangan beban server biasanya menggunakan round robin atau seleksi acak. Sekali lagi, dengan mengambil dari keuntungan pada karakteristik SDN, kami mengusulkan skema penyeimbangan beban server (Stewart, 1998). Baik superkomputer dan kelompok workstation membutuhkan penyetelan yang tepat untuk mencapai tingkat kinerja yang diharapkan. Ketika perhitungan untuk aplikasi paralel dibagi di antara sejumlah proses tetap yang akan dieksekusi pada node paralel, setiap proses melakukan sejumlah pekerjaan. Namun, sangat mungkin pada beberapa prosesor akan menyelesaikan tugas lebih awal dari yang lain dan menjadi menganggur karena pekerjaan tidak dibagi secara merata, atau beberapa prosesor memiliki pengoperasian yang cepat dari lainnya, dan dengan demikian menyelesaikan lebih awal. Kedua situasi dapat terjadi. Pemanfaatan yang efektif dari arsitektur pada komputer yang paralel membutuhkan beban komputasi untuk didistribusikan secara merata di atas prosesor yang tersedia. Salah satu persyaratan terpenting dari komputasi paralel yang efisien adalah membuat semua node sibuk; (Stewart, 1998) idealnya, ini bertepatan dengan penyelesaian pekerjaan dalam waktu singkat yang memungkinkan.



Gambar 2. Algoritma *Balancing* Klasik

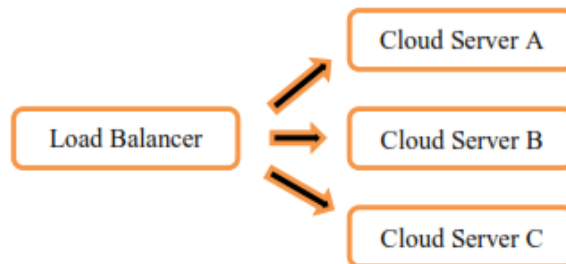
Beberapa studi telah dilakukan dengan menggunakan strategi penyeimbangan beban yang tersentralisasi. Solusi terpusat umumnya dianggap tidak menarik untuk sistem terdistribusi. Telah diterima secara luas bahwa algoritma load balancing terpusat cenderung menciptakan hambatan

kinerja dan titik kegagalan tunggal. Juga, kinerja algoritma ini menurun dengan peningkatan ukuran sistem terdistribusi. Namun, telah menunjukkan bahwa solusi terbaik adalah masalah lingkungan dan masalah. Studi ini menunjukkan bahwa pendekatan terpusat untuk distribusi informasi dan penempatan tugas mungkin sederhana dan efisien asalkan komunikasi antarprocessor relatif efisien dan skala sistem terbatas (hingga 50-100 host). Algoritma terdistribusi dianggap sebagai pilihan yang lebih cocok dan alami untuk load balancing dalam sistem komputer terdistribusi. Ini tidak menyebabkan kemacetan komunikasi seperti yang diperkenalkan oleh pengirim muatan pusat tunggal yang digunakan dalam algoritma keseimbangan beban terpusat. Ini juga meningkatkan keandalan sistem dan ketersediaan sumber daya sistem jika terjadi kegagalan. Jika prosesor gagal, proses penyeimbangan beban akan berlanjut di antara semua prosesor lain dalam sistem yang tidak terjadi dalam penjadwalan terpusat. Mayoritas penelitian saat ini difokuskan pada menemukan algoritma didistribusikan sub-optimal efisien untuk load balancing. Algoritma yang didistribusikan selanjutnya diklasifikasikan ke dalam koperasi dan non-koperasi. Dalam yang pertama, penjadwal lokal di setiap prosesor bekerja sama dan membuat keputusan yang didasarkan pada situasi di seluruh sistem. Dalam yang terakhir, penjadwal lokal membuat keputusan secara mandiri yang didasarkan pada informasi yang tersedia secara lokal. Mayoritas pekerjaan yang disajikan dalam tesis ini didasarkan pada algoritma load balancing yang didistribusikan secara kooperatif. Meminimalkan ketidakseimbangan beban adalah kegiatan utama dalam menghasilkan implementasi yang efisien aplikasi pada arsitektur paralel. Distribusi beban komputasi di seluruh sumber daya komputasi yang tersedia disebut dari masalah pada keseimbangan beban jaringan di literatur. Tujuan load balancing server adalah untuk meminimalkan waktu eksekusi untuk suatu program (T. R. Armstrong, 1998). Ini adalah dicapai dengan strategi yang menggeser tugas dari node yang banyak ke yang ringan. Load balancing dapat dilakukan secara statis sebelum eksekusi dari setiap proses dimulai, atau secara dinamis selama seluruh proses eksekusi. Metode statis memeriksa distribusi global beban komputasi dan menetapkan porsi beban pada kerja jaringan di sumber daya sebelum pemrosesan dimulai. Sebaliknya, metode dinamis memeriksa kemajuan perhitungan dan memanfaatkan pada sumber dalam daya yang diharapkan, dan menyesuaikan penugasan beban kerja sebagai perhitungan berlangsung. Perimbangan beban statis tergantung sumber pada daya yang secara lokal dan biasanya efisien pada titik awal eksekusi untuk beberapa aplikasi. Namun, karena penyeimbangan beban dilakukan sekali saja pada tahap inisialisasi perhitungan, penyeimbangan muatan statis tidak cocok untuk perhitungan yang memiliki beban pada kerja di jaringan yang berubah dinamis selama pelaksanaan. Dynamic load balancing mengacu pada redistribusi beban kerja di antara prosesor paralel selama proses perhitungan, sehingga setiap prosesor memiliki, lebih atau kurang, jumlah pekerjaan yang sama untuk dilakukan. Strategi penyeimbangan beban dinamis memiliki keunggulan dibandingkan strategi statis: Sistem tidak memerlukan penentuan apriori perilaku eksekusi run-time (Vouk, 2008). Dynamic load balancing telah menjadi fokus penelitian pada bidang komputasi paralel. Beberapa jadwal daftar yang optimal dan sub-optimal telah diusulkan dalam literatur. Daftar jadwal diklasifikasikan sebagai jadwal di mana tugas dipesan dalam daftar prioritas untuk diproses. Prioritas setiap tugas yang termasuk dalam daftar ditentukan oleh penjadwalan heuristik yang diterapkan. Daftar jadwal paling cocok untuk masalah penjadwalan di mana sistem tugas diwakili oleh grafik prioritas dan jumlah prosesor dalam jaringan diketahui. [Adam et. Al. 1974] memberikan perbandingan jadwal daftar untuk sistem pemrosesan paralel. Masalahnya terdiri dari penjadwalan satu set tugas yang dipesan sebagian pada dua atau lebih prosesor. Dalam studi

ini, komunikasi antar-tugas dan antarprocessor tidak dipertimbangkan. Beberapa algoritma sederhana berdasarkan prioritas level dan jadwal prioritas acak dibandingkan dengan jadwal optimal. Tingkat grafik prioritas didefinisikan dalam cara yang mirip dengan yang diterbitkan sebelumnya. Hasil yang diperoleh adalah sugestif bahwa algoritma yang didasarkan pada Level Tertinggi Pertama dengan Estimasi Waktu berkinerja lebih baik daripada heuristik berdasarkan Koefisien Level Tertinggi Pertama dengan estimasi dan tidak ada perkiraan waktu dan acak. Namun, tidak jelas mengapa heuristik ini lebih unggul dari yang lainnya. [Shirazi dan Wang 1988] menyajikan dua heuristik penjadwalan tugas statis dengan asumsi yang sama dengan yang dijelaskan dalam [Adam et. Al. 1974]. Fungsi heuristik pertama Heaviest Node First (HNF) memberikan prioritas tertinggi ke simpul terberat di setiap level. Algoritma ini menggunakan informasi level demi level.

2. METODOLOGI

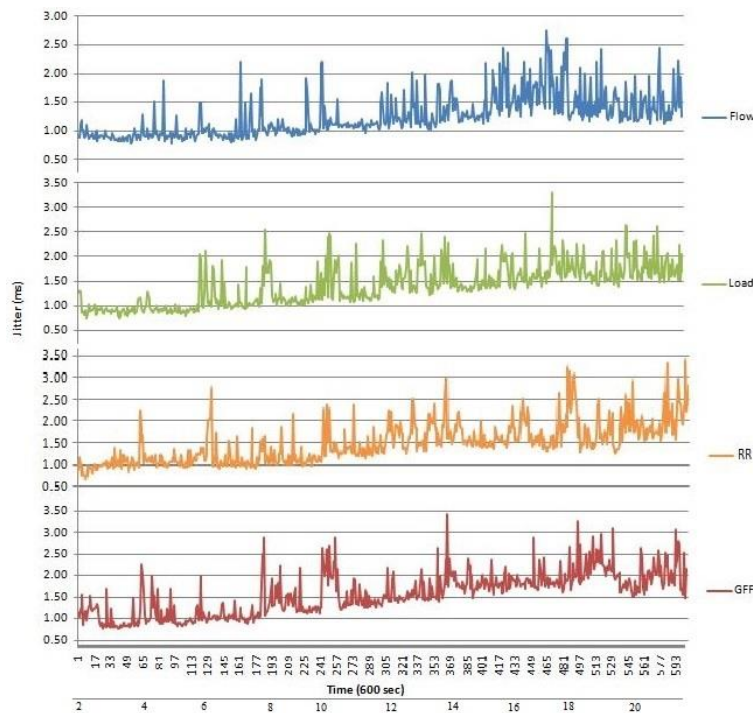
Untuk Tujuan pada penelitian ini adalah untuk merancang Server dan Network Load balancing skema (SNLB) berjalan pada Pengendali yang dapat secara merata mendistribusikan lalu lintas yang masuk melalui Server dan Jalur untuk memaksimalkan kinerja jaringan Datacenter pohon-lemak. SNLB mengadopsi teknik penjadwalan aliran dinamis di lingkungan SDN. Dalam bab ini, pertama-tama kita membahas tantangan arsitektur baru dan menyajikan solusi yang diusulkan untuk jaringan pusat data pohon-lemak. Kami kemudian menyelidiki peningkatan yang mungkin dalam hal fleksibilitas, kompleksitas serta kinerja. Dalam skema SNLB, kami telah mengategorikan aliran klien eksternal yang datang ke konten pengiriman jaringan pusat data ke aliran kecil dan besar. Aliran kecil adalah aliran yang membutuhkan kurang 10% dari bandwidth tautan dengan kapasitas paling sedikit. Ini adalah biasanya tautan antara sakelar akses dan sakelar agregasi. Aliran besar, terus sisi lain, adalah aliran yang membutuhkan 10% atau lebih dari bandwidth dengan tautan kapasitas paling sedikit.



Gambar 3. Metode Penelitian

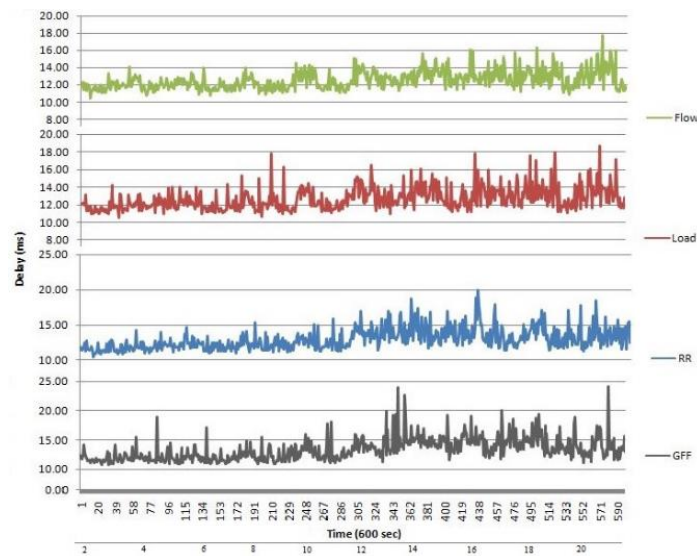
Dalam desain yang di usulkan, menerapkan fitur aktivasi server yang akan mengaktifkan server idle yang tersedia untuk menangani kondisi lalu lintas yang berlebihan. Ketika total beban (beban kecil ditambah aliran beban) dari semua server aktif mencapai nilai ambang batas kelebihan beban yang ditentukan, skema SNLB akan memicu sistem untuk mengaktifkan salah satu server siaga untuk menangani aliran baru. Kami meminta statistik port dari switch agregasi secara berkala (5 detik dalam skema SNLB). Statistik ini digunakan untuk melacak beban hulu saat ini di sakelar tepi. Catatan: untuk switch agregasi, lalu lintas yang diterima pada port yang terhubung ke tepi switch adalah lalu lintas hulu dari tepi switch.

Bagian ini membahas hasil pengujian yang diperoleh setelah menjalankan simulasi. Hasil ini disajikan sebagai bentuk grafik dan / atau tabular untuk membuatnya mudah amati hasilnya dan dibandingkan di antara algoritma yang berbeda. Skema SNLB memiliki jitter yang lebih kecil secara umum daripada yang lain skema. Semua skema memiliki nilai Jitter yang hampir sama di awal (4 menit pertama) karena lalu lintasnya ringan. Tetapi karena lalu lintas meningkatkan jitter dalam kasus GFF semakin cepat, pengalaman berdasarkan beban lebih sedikit jitter daripada Round robin tetapi SNLB memiliki kinerja terbaik.



Gambar 4. Latensi pada jaringan Jitter

Perbedaan jaringan disebabkan oleh perbedaan pemanfaatan tautan. GFF telah mengalokasikan lebih banyak aliran pada jalur tunggal, Round robin mendistribusikan aliran pada beberapa jalur dengan biaya yang sama tetapi tidak mempertimbangkan jenis aliran atau beban, berdasarkan beban dibagi berdasarkan pada sedikit beban tetapi mungkin telah mengalokasikan lebih banyak permintaan gajah pada setiap tautan tunggal yang mungkin menyebabkan kemacetan yang tiba-tiba. SNLB memiliki distribusi lalu lintas terbaik di antara server dan jalur dengan memanfaatkan semua tautan secara merata. Hal tersebut menyebabkan penundaan dan kinerja jitter lebih baik seperti pada Gambar 4. Kita dapat mengamati pada jaringan datacenter skema SNLB berbasis aliran memiliki latensi paling sedikit atas peningkatan lalu lintas dengan waktu. Karena skema ini bersama menyebarkan lalu lintas jaringan melalui tautan dan server sehingga menggunakan semua jalur secara merata, maka jalur tersebut mengalami paling sedikit beban pada jalur yang langsung mempengaruhi pada nilai latensi. Skema berbasis aliran SNLB hampir selalu berkinerja lebih daripada skema lain dalam hal latensi. Skema berbasis beban juga berkinerja lebih konstan daripada skema GFF dan Round skema robin. GFF berkinerja terburuk karena fakta bahwa dalam mengalokasikan seperti pada Gambar 5.



Gambar 5. Latensi pada jaringan SNLB

Gambar 5 menunjukkan latensi rata-rata dalam jaringan dalam 60 detik terakhir simulasi ketika beban lalu lintas adalah yang tertinggi. Kita dapat melihat bahwa SNLB berkinerja lebih konstan daripada semua skema penyeimbangan beban lainnya. Kita berpendapat pada penggunaan hasil untuk memprediksi bahwa menerapkan skema SNLB di pusat data jaringan bisa menjadi pilihan yang baik untuk meningkatkan kinerja jaringan untuk Skema SNLB lebih konstan dari skema lainnya. Dalam algoritma penyeimbangan beban dinamis, tugas ditransfer dari prosesor yang banyak dimuat ke prosesor yang ringan untuk mencapai peningkatan kinerja. Dua jenis mekanisme digunakan untuk mentransfer tugas; mekanisme penempatan tugas dan mekanisme migrasi tugas. Dalam sebelumnya, setelah tugas memulai eksekusi pada prosesor, tugas ini tidak dapat ditransfer dan akan tetap pada prosesor yang berasal untuk seumur hidup. Namun, dalam migrasi tugas, tugas yang sedang dijalankan dapat ditransfer ke prosesor jarak jauh untuk eksekusi lebih lanjut untuk meningkatkan keseimbangan beban. Dalam hal penempatan tugas, memuat informasi digunakan untuk membuat proses baru pada prosesor yang dimuat dengan ringan alih-alih mencoba untuk memindahkan proses yang sedang berjalan. Dalam migrasi tugas, kode, data, dan lingkungan proses yang berjalan, dipindahkan untuk melanjutkan eksekusi proses pada prosesor jarak jauh. Variabel-variabel dan struktur data yang terkait dengan proses ini tersebar di dalam sistem operasi dan melibatkan sejumlah besar overhead komunikasi yang membuatnya tidak layak tanpa menerapkan protokol khusus untuk tujuan ini dalam sistem. Sebenarnya proses migrasi yang sedang berjalan itu sepele dalam teori, tetapi hampir tidak mungkin dalam praktiknya. Algoritme penempatan tugas dikenakan biaya overhead rendah dan dapat dengan mudah diimplementasikan di sebagian besar sistem operasi, sedangkan hanya beberapa sistem operasi dalam praktik yang mendukung migrasi tugas. Algoritma berbasis beban juga berkinerja lebih konstan daripada algoritma berbasis GFF dan Round robin pada situasi lalu lintas yang padat di cloud computing.

4. KESIMPULAN

Load pada Balancing cloud server dipandang sebagai jaringan yang muncul dan penting teknologi. Ini telah diadopsi di banyak jaringan pusat data. SDN dapat memberikan fleksibilitas dalam mengendalikan komponen arsitektur, penggunaan sumber pada daya jaringan secara pintar

dan pengontrolan jaringan cloud yang cerdas (mis. Kualitas Layanan (QoS)), sambil mengurangi biaya modal pada pengeluaran dan Biaya Pengeluaran Operasional (OPEX). Load pada balancing server yang diusulkan meningkatkan kinerja di Jaringan Datacenter Pengiriman Konten. Skema ini secara cerdas memanfaatkan fitur yang muncul dari SDN. Dalam skema ini, statistik real time dari jaringan untuk menyebarkan data lalu lintas jaringan secara merata di antara server dan jalur ke server digunakan. Selain itu, situasi lalu lintas yang berlebihan dan toleransi kesalahan ditangani secara efisien menggunakan sumber daya jaringan yang tersedia. Hasil kinerja menunjukkan bahwa skema SNLB berkinerja lebih baik daripada GFF, dan Algoritma round robin.

5. UCAPAN TERIMA KASIH

Penelitian ini didanai hibah dari Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia (Ristekdikti) dengan skema Penelitian Dosen Pemula (PDP) pada tahun 2019.

6. DAFTAR PUSTAKA

- [1] A. M. Alakeel. A guide to dynamic load balancing in distributed computer systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(6), 153–160. 2010.
- [2] A. Vouk. Cloud computing-issues, research and implementations. *Information Technology Interfaces*, 31–40. 2008.
- [3] B.Wickremasinghe., R.N.Calheiros., R. Buyya. Cloudanalyst: A Cloudsim-Based Visual Modeller for Analysing Cloud Computing Environment and Applications. *24th IEEE International Conference on Advanced Information Networking and Applications, AINA*. 2010.
- [4] G. Boss, P. Malladi, D. Quan, L. Legreg, *Cloud computing, in: High Performance on Demand Solutions (HiPODS)*. IBM. 2007.
- [5] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6). pp. 599–616. 2009.
- [6] R. Buyya, R. Ranjan, Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. *ICA3PP 2010, Part I, LNCS 6081*. pp. 13–31. 2010.
- [7] R. Armstrong, D. Hensgen. The relative performance of various mapping algorithms is independent of sizable variances in run- time predictions. *7th IEEE Heterogeneous Computing Workshop (HCW '98)*. pp. 79–87. 1998
- [8] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. Rose, R. Buyya, Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience (SPE)*, Vol. 41, No.1, ISSN: 0038-0644, Wiley Press, New York, USA. pp. 23–50. 2011.