LOAD BALANCING PADA CLOUD COMPUTING MENGGUNAKAN METODE LEAST CONNECTION

LOAD BALANCING ON CLOUD COMPUTING USING LEAST CONNECTION METHOD

Ari Wibowo[1], Agus Virgono, MT[2], Roswan Latuconstina, MT[3]
[1,3]Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom
[1]ariwibowoitt@student.telkomuniversity.ac.id, [2]agv@telkomuniveristy.co.id, [3]rlc@telkomuniversity.ac.id

Abstrak

Load balancing adalah suatu teknik didalam lingkungan cloud computing yang memiliki tujuan utama untuk menyeimbangkan load server dalam menangani request. Load balancing juga merupakan salah satu teknik untuk meningkatkan performasi server secara keseluruhan dengan menyeimbangkan request yang diproses oleh server sehingga server mampu menyelesaikan request secara efektif dan efisien. Terdapat banyak library maupun tools yang dapat digunakan untuk menganalisis sistem cloud salah satunya adalah cloudsim. Pada tugas akhir disimulasikan dan dianalisis mengenai metode Least Connection dalam menangani load balancing pada sistem cloud yang disimulasikan pada cloudsim. Hasil simulasi menghasilkan hasil yang cukup baik yakni pemerataan CPU load sekitar 28% dari seluruh VM yang digunakan, kemudian tingkat pemerataan beban jaringan juga yang cukup baik dimana kecepatan rata rata proses yang mampu diselesaikan oleh masing masing server mampu diterima client dalam waktu yang relatif singkat (sekitar 40ms). Kecepatan proses akhir secara keseluruhan terlihat bahwa server mampu memberi waktu penyelesaian yang hampir sama bagi proses yang memiliki bobot yang sama (hanya terdapat selisih maksimal 40 mili second). Dengan demikian dapat disimpulkan bahwa algoritma Least Connection adalah algoritma yang layak digunakan oleh cloud server yang memiliki skalabilitas dan workload yang tinggi

Kata kunci: ***Least Connection, Cloud, Cloudsim, VM, server, client***

Abstract

Load Balancing is one of technique on the cloud computing methods that balancing server when handling requests. Load balancing is also one of the server performance improvement by balancing requests processed by the server. By using this technique, the server can process requests effectively and efficiently. There are so many tools and libraries that used to analyze cloud system. One of the cloud analyzing tools is called cloudsim. In this final project, cloudsim will be used to simulate and analyze Least Connection Algorithm to balancing server loads. The simulation results give a relatively good result by balancing request so that the CPU load is balanced about 28% across all Vm's. from the network aspect, network balancing is also acquired by least connection algorithm. It can be seen from the average processing time that almost all requests were received by the client on about 40ms. The final performance of the server is also good while the difference of received results is only about 40 milliseconds. Finally, this research earns to conclude that the Least Connection algorithm is feasible to implemented on high scalability cloud server with high workloads.

Keywords: Least Connection, Cloud, Cloudsim, VM, server, client

1. Pendahuluan

Meningkatnya penggunaan interner saat ini, mendorong akan meningkatnya *traffic* internet. *Cloud programing* memiliki peran penting daam penyediaan semua jenis sumber daya dibidang teknologi dan layanan internet ke semua penggunaanya. *Cloud computing* adalah cara baru dalam memanfaatkan teknologi informasi dan komunikasi, oleh karena itu *cloud computing* mengubah cara padangan dalam bidang insfrastruktur komutasi. Melalui *cloud computing* pengguna dapat menyimpan data mereka tanpa harus memiliki kapasitas *memory* yang sangat besar dan bisa mengakses data pribadi dimanapun dan kapanpun dengan hanya terhubung dengan jaringan internet.

Berdasarkan tabel dari penelitian sebelumnya *least connection* ini mendapatkan hasil yang sangat baik dalam beberapa parameter yaitu pengukuran *throughput, end to end delay* dan *jitter* [1]

Hasil Simulasi Perbandingan Rata-Rata Keseluruhan Jaringan *Load Balancing*

Faktor Pengukuran	Average Data 1 Mbps		
	Round Robin	Least Connection	Ratio
Throughput (Kbps)	897	**960**	948
End-to-End Delay (ms) – VoIP	738	336	**329**
Jitter (ms) – VoIP	0.041	0.031	**0.020**
Faktor Pengukuran	**Average Data 5 Mbps**		
	Round Robin	**Least Connection**	**Ratio**
Throughput (Kbps)	3660	**3751**	3703
End-to-End Delay (ms) – VoIP	**371**	406	426
Jitter (ms) – VoIP	0.006	0.047	**0.002**

Tabel 1.1 - Perbandingan Round Robin, Least Connection

Mengingat luasnya ruang lingkup rumusan masalah tugas akhir ini dan juga keterbataasan waktu tenaga dan pikiran, akan baiknya perlu adanya batasan masalah dengan mempertimbangkan aspek jaringan internet dengan menggunakan *load balancing* adapun batasan masalah tersebut adalah

Tugas akhir ini membahas bagaimana memberikan distribusi CPU dalam membangun server.

- Mengoptimalkan layanan server berbasis simulasi *cloudsim.*
- Server hanya berbentuk *virtual*
- Penelitian ini tidak membahas sistem keamanan pada *cloud computing*
- Pengujian ini hanya bersifat simulasi dengan menggunakan *toolkis cloudsim*.

2. Landasan Teori

2.1 Penelitian Terkait

Penelitian mengadopsi model adopsi *Roadmap for Cloud Computing Adoption (ROCCA)* menggunakan untuk implementasi *cloud computing* di Universitas Semarang (USM). Hasil penelitian ini berupa modifikasi model adopsi ROCCA, yang terdiri dari 5 tahap, yaitu tahap analisi, perancangan, adopsi, migrasi dan pengelolahan [2]. Penelitian yang berjudul "*Overhead Analysis as One Factor Scability of Private Cloud Computing for IAAS Servic*e" bertujuan untuk mengetahui *overload* pada lingkup virtualisasi. Implementasi *private cloud* menggunakan *OpenStack* dengan konfigurasi *multiple interface* dan *multiple server* [3]. Penelitian yang berjudul **Load Balancing pada Cloud Computing Dengan Sumber Daya Terbatas Menggunakan Penggabungan Algoritma ESCE dan Throedhold"** melakukan percobaan sebanyak tiga kali, ketiga percobaan tersebut menghasilkan waktu respon yang lebih cepat daripada menggunakan algoritma *throttled*, dengan waktu respon sebesar 10381,7ms [4], paper yang berjudul "***Dynamic Load Balancing in Cloud Computing using Cloudsim"*** membahas tentang *cloud computing* bersama dengan analisis dalam *load balancer*. Paper ini juga memfokuskan pada keuntungan dan kerugian dari *cloud computing*[5].

Berdasarkan penelitian yang berjudul "***Load Balancing Algorithms Round Robin, Least Connection and Least Loaded Efficiency"*** kesimpulannya adalah algoritma server load yang memiliki lebih banyak utilitasi CPU yang dibandingkan dengan algoritma *round robin* dan algoritma *Number of Connection* dan ketika algortima *server* load telah digunakan, akan ada perbedaan besar dalam beban antara 8 HTTP[7].**Pengujian Distribusi Beban Web dengan Algoritma *Least Connection* dan *Weighted Least Connection*"** merupakan pengujian distribusi beban kerja HTTP secara statis (dengan pesat HTTP/*second* yang tetap) dan secara dinamis (dengan pesat permintaan HTTP/*second* yang berubah atau naik secara teratur) dari *client* ke *pool* sistem *server*, kemudian penelitian ini dilanjutkan dengan analisa lalu lintas paket data [8].

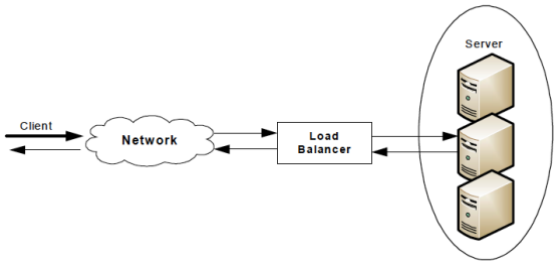
2.2 Cloud Computing

Dalam beberapa tahun terakhir *cloud computing* tidak hanya jadi topik di dunia statis. Keuntungan utama dari pemanfaatan *cloud computing* adalah sisi ekonomis didapatkan oleh pihak pelanggan layana yang hanya perlu menyewa insfrastruktur, perangkat lunak maupun layanan jangka waktu tertentu sesuai dengan kenutuhan[9]. *Cloud computing* adalah sebuah model *client-server* dipandang sebagai layanan yang dapat diakses pengguna secara mudah dan setiap saat. *Cloud computing* merupakan hasil evolusi dari *grid computing* dapat dipandang sebagai penggabunagn komputer [10]. Saat ini *cloud computing* sudah berkembang dengan sangat cepat, sehingga sehingga meningkatkan jumlah pengguna yang mengakses *cloud server*. Salah satu isu yang muncul akibat semakin meningkatnya penggunaan ini adalah masalah *load balancing* [12].

2.3 Load Balancing

Load balancing adalah suatu teknik dalam pendistribusian beban *traffic* dalam dua jalur atau lebih sehingga menghasilkan *traffic* yang seimbang dan dapat mengoptimalkan *throughput* data. Pada dasarnya teknologi *load balancing* ini untuk memanfaatkan server secara efektif, sehingga jaringan komputer dapat bekerja dengan baik tanpa adanya suatu kendala. 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 antara server yang dikenal dengan load sharing. Secara konsep, *load balancer* merupakan jembatan antaran *server* dan *network* seperti gambar dibawah ini



Gambar 1. *Server cluster* dengan *load balancer*

Dengan *load balancing* mempunyai keuntungan antara lain menurunkan jumlah pemprosesan yang harus dilakukan oleh *server* penerima utama sehingga memungkinkan untuk *server* penerima utama dan lebih banyak perangkat yang memproses beban [13].

2.4 Sistem berbasis cluster

Ada dua upaya yang bisa dilakukan, yaitu upaya scale-up (platform server tunggal) dan upaya scale-out (platform server jamak). Upaya pertama sudah cukup baik, akan tetapi mempunyai beberapa kelemahan. Pertama, membutuhkan biaya yang lebih besar agar dapat selalu mengikuti perkembangan teknologi yang mutakhir. Kedua, tidak dapat menghilangkan fakta bahwa titik tunggal kegagalan (single point of failure) justru ada pada server itu sendiri.

2.5 Algoritma Least Connection

Least Connection merupakan salah satu bentuk algoritma penjadwalan yang dinamis, karena memerlukan perhitungan koneksi aktif untuk masing-masing real server secara dinamis. Untuk server virtual yang mengelolah kumpulan server dengan kinerja serupa, penjadwalan least connection sangat baik untuk memperlancar distribusi saat banyak permintaan yang berbeda [7].

2.6 Penentuan Bobot

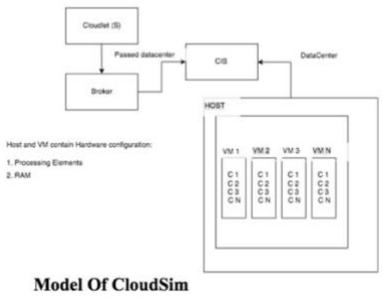
Penentuan bobot dipengaruhi oleh jenis isi web (web-content) yang disediakan oleh server web. Apabila isi web bersifat statis (static web-content) maka bobot yang hanya akan dipengaruhi oleh faktor kecepatan maka media penyimpanan, Pm. Apabila isi web besifat dinamis (dynamic web-content) maka bobot yang hanya akan dipengaruhi oleh factor kecepatan prosessor, Pp. Jika isi web merupakan gabungan statis dan dinamis, maka rumusnya akan menjadi, dimana, a adalah rasio yang menentukan besaran kontribui Pm dan Pp terhadap bobot w.

$$\alpha = \frac{N_d}{(N_d + N_s)}$$

dengan nilai Nd dan Ns adalah jumlah statistic akses isi web [8].

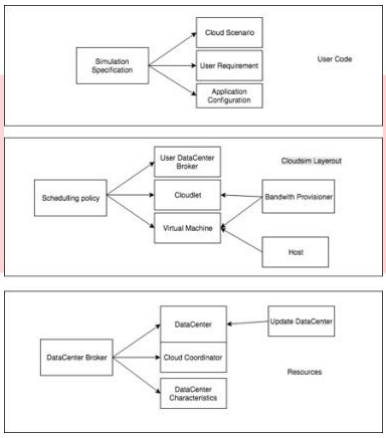
2.7 Cloudsim

Cloudsim adalah *toolkit* untuk simulasi *cloud computing* yang ditulis dalam bahasa java. *Cloudsim* dibangun di laboratorium CLOUDS (*Cloud computing and distributed System*) oleh *University of Melbourne Australia* [6]. Toolkit cloudsim mendukung pemodelan sistem dan perilaku komponen sistem *cloud* seperti *data center*, *virtual machine* (VM), *processing elements* (PE) dan kebijaksanaan penyedia sumber daya [16].



Gambar 2. Pemodelan Cloudsim.

Lapisan ini juga menangani masalah mendasar seperti penyediaan host ke VM, mengelola eksekusi aplikasi dan monitoring dinamic sistem state [20]. Ditumpukan ini, lapisan paling atas dari cloudsim adalah user code yang digunakan untuk mengekspos entitas dasar seperti (jumlah mesin dan spesifikasinya dll) untuk host, (jumlah tugas dan kebutuhannya) untuk aplikasi, penjadwalan penggunaan tipe aplikasinya.



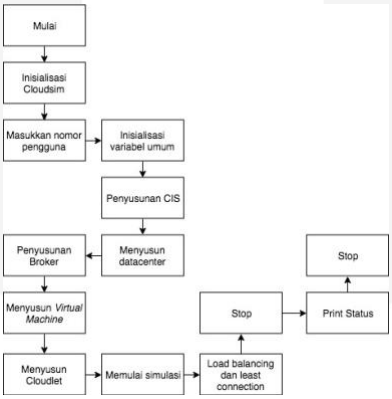
Gambar 3. Arsitektur Cloudsim

Artektur *cloudsim* adalah desain yang berlapis-lapis, awalnya lapisan ini terhubung dengan sim java sebagai mesin simulasi diskrit. *Sim java* mendukung fungsi inti yang berbeda seperti pembuatan entitas cloud sistem seperti *host, datacenter, broker VM,* dan lainnya, pemprosesan *event queue* dan komunikasi antara komponen dan manajemen simulasi waktu [17].

3. Perancangan Sistem

3.1. Cara Pengajuan dan Pengutipan

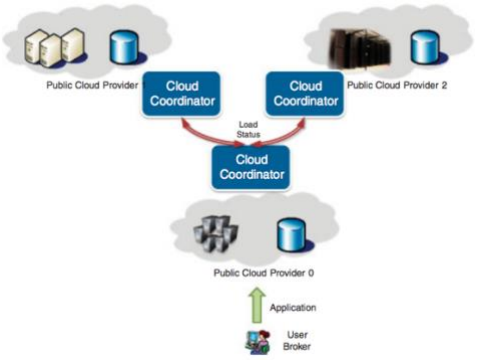
Simulator cloudsim terdiri dari berbagai jenis modul, kemudian modul tersebut dipanggil untuk mengeksekusi setiap fungsi tertentu yang ada pada program dan dapat dilihat dari diagram diatas, alur cloudsim mulai dari proses inisialisasi hingga proses eksekusi program.



Gambar 4. Gambaran Umum Sistem

3.2. Perancangan Topologi

Pada penelitian ini, akan digunakan topologi cloudsim yang umum digunakan pada penelitian penelitian sejenis seperti pada [19] [17]. Pada gambar 5 digambarkan tentang perancangan topologi pada *cloudsim* untuk sebuah arsitektur *cloud* yaitu terdiri atas beberapa komponen utama seperti *user, datacenter*, administrator dan juga *provider* ataupun pengembang program.



Gambar 5. Topologi umum network data center.

Pengguna (*User*) adalah entitas pertama karena *user* memiliki akses untuk melalukan perintah *request* ke pengembang sesuai dengan kebutuhan yang dibutuhkan oleh *user* tersebut.

3.3. Perancangan Metode Least Connection

Algoritma *load balancing* akan digunakan untuk mengatur dan menyeimbangkan *traffic* akses ke *server* sedemikian sehingga beban *server* akan lebih merata dengan beban jaringan yang juga dibuat merata.

3.4. Perancangan Algoritma Least Connection dalam mengatasi load balancing

Struktur dibawah ini adalah sebuah proses pencarian struktur data pada pengujian dengan

menggunakan metode least connection.

```
for (m = 0; m < n; m++) {
    if (W(Sm) > 0) {
        for (i = m+1; i < n; i++) {
            if (C(Sm)*W(Si) > C(Si)*W(Sm))
                m = i;
        }
        return Sm;
    }
}
return NULL;
```

Gambar 7. Pseudo code Least Connection

Implementasi penggunaan metode *load balancing* digunakan untuk optimasi penggunaan *server* dengan menggunakan aturan pembagian secara *bandwith* seimbang dan merata, pengambilan data uji coba di kolektif pada data server yang terkoneksi oleh *device* yang lain dalam hal ini perangkat *mobile*, PC dan *smartphone*.

4. Implementasi dan Pengujian Sistem.

4.1. Implementasi Sistem

Pada simulasi menggunakan cloudsim diimplementasikan server – server virtual yang dapat dispesifikasikan secara manual mulai dari processor, RAM, kecepatan jaringan dan lain sebagainya. mun demikian berikut adalah spesifikasi sistem baik sistem *real* yang digunakan untuk menjalankan simulasi maupun spesifikasi komputer *virtual* didalam cloudsim yang digunakan untuk penelitian pada tugas akhir ini:

4.1.1 Spesifikasi komputer fisik

- Processor Inel i5
- Memory DDR 3 8GB 1867 MHz
- Harddisk SSD 500GB -6GBPS
- Intel Iris Graphics 6100 1536MB

4.1.2 Spesifikasi komputer virtual

- Processor Intel Xeon
- RAM 1GB

4.1.3 Spesifikasi perangkat lunak

- MacOs High Sierra
- NetBeans 8.1
- Cloudsim 4.2
- Sourcetree

4.2. Implementasi Topologi

Membuat kelas yang digunakan untuk layanan aplikasi cloud sebagai merepresentasikan komplesitas aplikasi dalam bentuk persyaratan komputasi.

```
for(int i=0;i<cloudlets;i++){
    //cloudlet parameters
    long length = random.nextInt(60000) + 30000; //40000;
    long fileSize = random.nextInt(500) + 200;//300;
    long outputSize = random.nextInt(600) + 200; //300;
    int pesNumber = 1;
    UtilizationModel utilizationModel = new UtilizationModelFull();
```

Gambar 8. Pseudo code Cloudlet

Selanjutnya membangun *datacenter* untuk melakukan perhitungan terhadap jumlah host seperti yang dijelaskan di bab sebelumnya.

```
public Datacenter(
        String name,
        DatacenterCharacteristics characteristics,
        VmAllocationPolicy vmAllocationPolicy,
        List<Storage> storageList,
        double schedulingInterval) throws Exception {
    super(name);
```

Gambar 9. Pseudo code Datacenter.

Proses dalam membangun sebuah cloudsim juga memerlukan host yang berfungsi untuk mengalokasikan kebijakan untuk menyediakan memori dan *bandwith* ke *virtual machine*.

4.3. Pengujian tingkat pemerataan beban server

Pengujian pertama akan menguji seberapa Algoritma *Least Connection* mampu menyeragamkan beban pada server.

4.3.1 Tujuan Pengujian

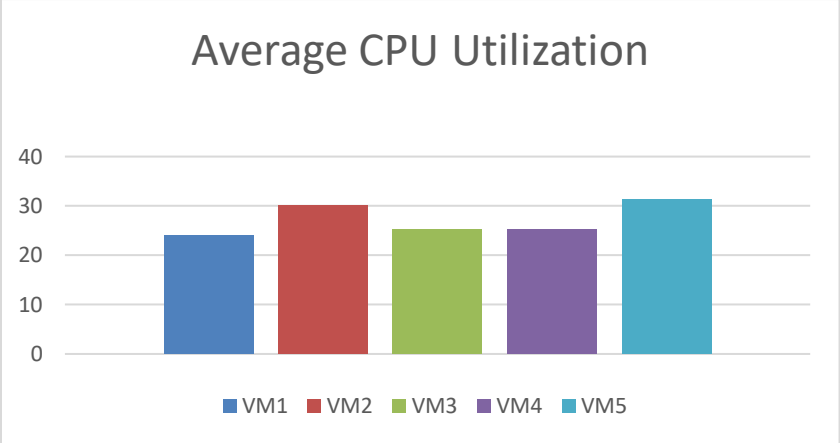
Tujuan dari pengujian tingkat **pemerataan beban server yakni untuk memastikan bahwa tidak ada server yang mengalami *overload.***

4.3.2 Cara Pengujian

Setelah *server* siap, dimasukkan sejumlah *request* secara periodik untuk menguji seberapa *balance server* yang dibangun dengan Algoritma *Least Connection*.

4.3.3 Hasil Pengujian dan Analisa

Berikut adalah grafik pengujian simulasi dengan 5 (lima) buah instance Virtual Machine (VM).



Gambar 10. Persentase CPU Load per VM

Pengujian dilakukan selama dua jam dengan pencatatan rata rata CPU load setiap menitnya. Berdasarkan pengujian didapat hasil yang merata baik ketika *request* baru masuk, *peak loa*d, hingga *ending*. Dengan demikian dapat dikatakan bahwa algoritma *Least Connection* mampu membagi load secara merata ke seluruh server yang terhubung.

4.4 Pengujian response time setiap cloudlets

Pengujian response time ini dilakukan untuk mendapatkan pelaporan dan gambaran mengenai kualitas response time dari server terhadap beban kerja yang diberikan (dalam hal ini berupa cloudlets).

4.4.1 Tujuan Pengujian

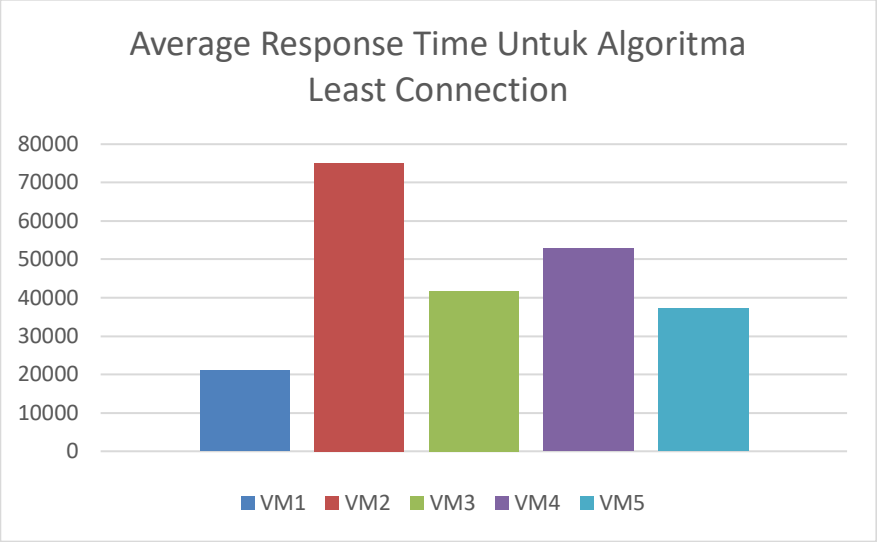
Tujuan dari Pengujian response time ini adalah **untuk memastikan bahwa dengan algoritma *Least Connection* response time terhadap cloudlets secara umum tidak mengalami degradasi.**

4.4.2 Cara Pengujian

Cara pengujian untuk mendapatkan analisis cloudlets yakni dengan mengimplementasikan server pada Cloudsim sesuai spesifikasi yang telah dijelaskan pada bab sebelumnya. Setelah server siap, diajukan sejumlah task berupa cloudlets kepada *datacenter* untuk menguji seberapa cepat pemrosesan seluruh request tersebut dengan Algoritma Least Connection sebagai *balancer*.

4.4.3 Hasil Pengujian dan Analis

Untuk menguji performa algoritma *Least Connection* dari segi *response time* terhadap cloudlets, akan dianalisis resonse time dari semua Virtual Machine (VM) yang digunakan saat simulasi. Grafik berikut merepresentasikan hasil pengujian algoritma Least Connection dalam menyelesaikan sejumlah request (600 cloudlets) pada lima buah Virtual Machine.



Berdasarkan Gambar 4.3, dapat dilihat bahwa untuk seluruh task (cloudlets), algoritma *Least connection* mampu membagi rata beban ke semua Virtual Machine. Hal ini terlihat dari kurva *response time* dari algoritma *least square* yang cukup merata dengan maksimal processing time 75 second dan minimal processing time sebesar 22 second. Dengan demikian dapat dilihat bahwa algoritma *Least Connection* menghasilkan *response time* yang cukup baik untuk task yang diberikan.

5. Kesimpulan dan saran.

Pada bagian ini akan dijelaskan tentang kesimpulan yang diperoleh dari hasil pengujian sistem dan pengambilan data dari hasil simulasi server menggunakan Cloudsim serta observasi selama penggerjaan Tugas Akhir berlangsung.

5.1 Kesimpulan

Pada Tugas Akhir ini, telah dirancang, diimplementasikan, dan dianalisis mengenai load balancing server menggunakan algoritma Least Connection. Administrator memiliki tugas dalam pengoprasian datacenter dan perangkatnya. Berdasarkan analisis yang dilakukan dapat ditarik beberapa kesimpulan, yakni:

1. Algoritma Least Connection menjamin tingkat pemerataan beban server yang baik dan merata. Hal tersebut dapat dilihat dari pemerataan CPU load sekitar 28% dari seluruh VM yang digunakan.
2. Algoritma Least Connection memberikan tingkat pemerataan beban jaringan yang cukup baik dilihat dari kecepatan rata rata proses yang mampu diselesaikan oleh masing masing server mampu diterima client dalam waktu yang relatif singkat (sekitar 40ms).

5.2 Saran

Penulis mengharapkan pengembangan selanjutnya agar riset ini dapat maju dan berkesinambungan. Saran yang dapat dikontribusikan oleh penelitian ini bagi penelitian selanjutnya yakni:

1. Penelitian ini terbatas pada teknologi Cloudsim yang ada saat ini yakni seluruh konfigurasi di set manual, sehingga akan sangat lama apabila harus mengkonfigurasikan ratusan server secara manual, dengan demikian akan sangat baik apabila pada penelitian selanjutnya dapat dilakukan setup dengan teknik pemrograman yang mampu mengubah sistem awal Cloudsim sehingga server farm yang dibangun dapat berjumlah ratusan bahkan ribuan dan dianalisis dalam waktu singkat.
2. Penelitian selanjutnya diharapkan dapat mengkustomisasi Cloudsim sehingga dapat memodelkan multi *setup* untuk satu *datacenter*. Dengan demikian akan dapat diketahui performansi algoritma yang dibangun pada dunia nyata. Karena sistem Cloudsim mengasumsikan semuanya ada dalam kondisi ideal sehingga tidak memodelkan kondisi dunia nyata yang memiliki banyak batasan dan tidak ideal.

Daftar Pustaka:

[1] H. N. d. T. Witono, "Analisis Algoritma Round Robin, Least Connection dan Ratio pada Load Balancing Menggunakan Opnet Modeller," vol. 12, 2016.

[2] M. S. Prayogi, "Implementasi Cloud Computing Menggunakan Model Adopsi Roadmap for Cloud Computing Adoption (ROCCA) Pada Institusi Pendidikan," no. Tesis, 2014.

[3] T. Adji, Nggilu.F.S dan Sumaryono.S, "Overhead Analysis As One Factor Scalibility Of Private Cloud Computing For IAAS Service," vol. 4, no. 5, pp. 288-295, May 2013.

[4] Endang wahyu pamungkan dan Divi Galih Prasetyo putri, "Load Balancing pada Cloud Computing dengan Sumber Daya Terbatas Menggunakan Penggabungan Algoritma ESCE dan Throttles," vol. VII, no. 1, Juni 2015.

[5] Jayprakash MaltRE, Balwant Prajapat, "Dynamic Load Balacing in Cloud Computing using CloudSim," International Journal of Computing Application, vol. 148, no. 5, August 2016.

[6] Dr Umang Singh, Ms. Ayushi sharma, "CloudSim Simulator Used for Load Balancing in Cloud Computinh," International Journal of Emerging Technology and Advanced Engineering, vol. 6, no. 4, April 2016.

[7] Dr Mustafa El Gili Mustafa, "Load Balancing Algorithms Round Robin, Least Connection and Least Loaded Efficiency," vol. 1, p. 51, 2017.

[8] Nongki Angsar, "Pengujian Distribusi Beban Web dengan Algoritm Least Connection dan Weighted Least Connection," Jurnal Ilmiah Semesta Terbuka, vol. 3, no. 1, 2014.

[9] Marios D. Dikaiakos, Dimitrios Katsaros and Pankaj Mehra, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research," vol. 13, no. 5, september 2009.

[10] Sofana, Teori dan Praktik Cloud Computing (Opennebila, VMware dan Amazon Aws) Informatika, Yogyakarta: Graha Ilmu, 2012.

[11] Domanal and Shridal G, "Load Balancing in Cloud Computing using Modified Throttled Algorithm," 2013.

[12] Zenon Chaczko and Venkatesh Mahadevan, "Availability and Load balancing in Cloud Computing," November 2015.

[13] Kurniawan, Y., Sabriansyah dan Sakti, E.,, "Analisis Kinerja Algoritma Loat Balancer dan Implementasi pada Layanan Web. Unversitas Brawijaya," vol. 3, 2013.

[14] Ruixia Tong and Xiongfeng Zhu, "A Load Balancing Strategy Based on the Combination of Static and Dynamic".

[15] Valeria Cardellini, Emiliano Casalicchio, Michele Colajanni, Modena, Philip S. Yu, "The state of the art in locally distributed Web-server systems," vol. 34, no. 2, pp. 263-311, june 2002.

[16] Yongsheng Hao, Guanfeng Liu and Junwen Lu, "Three Level Load Balancing on Cloudism," 2015.

[17] odrigo N. Calheiros, Rajiv Ranjan, César A. F. De Rose, "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services," p. 9, march 2009.

[18] Jun-Kwon Jung, Sung-Min Jung, Tae-Kyung Kim, and Tai-Myoung Chung , "A Study on the Cloud Simulation with a Network Topology Generator," vol. 6, 2012.

[19] Tarun Goyal,Ajit Singh, Akansha Agra, "Clousim: Simulator for cloud computing infrastructure and modeling," *International Conference on Modeling Optimization adn Computing (ICMOC)*, vol. 38, 2012.

[20] Baptiste Louis, Karan Mitra and Saguna Saguna, "CloudSimDisk: Energy-Aware Storage Simulation in CloudSim," March 2016.

[21] N. R. d. Z. Munawar, "Pengambilan Keputusan dengan Teknik Soft Computing," *Jurnal Ilmiah Teknologi Informasi Terapan Volume III, No 3, 15 Agustus 2016*, 2016.

[22] M. Falahuddin, Lebih Jauh Mengenal Koputasi Awan., Dipetik dari Detiknet: Http://Inet.Detik.Com/Read/2010/02/24/084138/1305595/328/6/Lebih- Jauh- Mengenal-Komputasi-Awan. penyunt., Bandung: Informatika, 2010.

[23] Bhaskar Prasad Rimal, Eunmi Choi and Ian Lumb, "A Taxonomy and Survey of Cloud Computing Systems," November 2009.