

OPTIMASI SERVER UNTUK VIRTUAL HOSTING DENGAN LOAD BALANCING DI PT. KINEMA SYSTRANS MULTIMEDIA

Wilhelmus Ervando Walangare¹, Andik Yulianto^{2*}

Universitas Internasional Batam

email: 2232049.wilhelmus@uib.edu¹, andik@uib.ac.id^{2*}

Abstrak

Perkembangan pesat layanan digital menuntut ketersediaan dan performa tinggi dari infrastruktur server, khususnya pada lingkungan virtual hosting. Permasalahan utama yang dihadapi adalah penurunan kinerja dan potensi kegagalan layanan akibat lonjakan lalu lintas yang tidak terdistribusi dengan baik. Kegiatan pengabdian kepada masyarakat (PkM) ini bertujuan untuk mengimplementasikan solusi optimisasi server melalui penerapan load balancing menggunakan Nginx. Metode yang digunakan adalah simulasi dan substitusi Ipteks pada lingkungan virtual yang mereplikasi kondisi nyata, di mana server tunggal yang rentan digantikan oleh sebuah klaster yang terdistribusi. Proses implementasi mencakup penyiapan server backend pada sistem operasi Red Hat Enterprise Linux (RHEL), konfigurasi keamanan, dan penerapan Nginx sebagai load balancer dengan algoritma round-robin. Hasil pengujian fungsional menunjukkan bahwa sistem berhasil mendistribusikan permintaan secara merata ke seluruh server backend, memvalidasi efektivitas solusi dalam mitigasi single point of failure dan peningkatan skalabilitas. Rekomendasi untuk kegiatan selanjutnya adalah eksplorasi algoritma dinamis, implementasi keamanan SSL/TLS, dan integrasi dengan sistem monitoring real-time.

Kata Kunci: *Load Balancing, Nginx, Optimisasi Server, Virtual Hosting, Ketersediaan Tinggi*

Abstract

The rapid development of digital services demands high availability and performance from server infrastructure, especially in virtual hosting environments. The main problem faced is performance degradation and potential service failure due to surges in und-distributed traffic. This community service activity aims to implement a server optimization solution through the application of load balancing using Nginx. The method used is an IPTEKS simulation and substitution in a virtual environment that replicates real-world conditions, where a vulnerable single server is replaced by a distributed cluster. The implementation process includes setting up backend servers on the Red Hat Enterprise Linux (RHEL) operating system, security configurations, and deploying Nginx as a load balancer with the round-robin algorithm. Functional testing results show that the system successfully distributes requests evenly across all backend servers, validating the solution's effectiveness in mitigating single points of failure and enhancing scalability. Recommendations for future activities include exploring dynamic algorithms, implementing SSL/TLS security, and integrating with real-time monitoring systems.

Keywords: *Load Balancing, Nginx, Server Optimization, Virtual Hosting, High Availability*

Pendahuluan

PT. Kinema Systrans Multimedia (Infinite Learning) adalah sebuah pusat pelatihan teknologi yang berlokasi di Nongsa Digital Park, Batam. Aktivitas operasional mitra berfokus pada penyelenggaraan program pelatihan intensif (bootcamp) dan studi independen di berbagai bidang teknologi seperti Web Development, Game Development, dan Data Science.

Transformasi digital telah mendorong ketergantungan yang masif terhadap layanan berbasis web dan cloud. Hal ini menyebabkan penyedia layanan virtual hosting menghadapi tantangan besar dalam menjaga Quality of Service (QoS), terutama dalam hal latensi dan ketersediaan. Arsitektur server tunggal, meskipun mudah dikelola, memiliki kelemahan fundamental: ia menjadi single point of failure (SPOF) yang dapat melumpuhkan seluruh sistem jika terjadi kegagalan. Ketika server tersebut mengalami kelebihan beban atau kegagalan, seluruh layanan yang bergantung padanya akan terhenti, yang mana ini menjadi isu krusial di lingkungan mitra, PT. Kinema Systrans Multimedia (Infinite Learning), di mana portal pembelajaran harus dapat diakses secara simultan oleh banyak peserta.

Upaya untuk mengatasi masalah ini telah banyak dilakukan, salah satunya melalui implementasi kluster server yang didistribusikan menggunakan load balancer. Berbagai penelitian telah membandingkan kinerja algoritma load balancing serta mengevaluasi efisiensi berbagai perangkat lunak load balancer. Kegiatan PkM ini terinspirasi dari kebutuhan nyata di lapangan dan didukung oleh kematangan teknologi load balancing. Tujuan kegiatan ini adalah untuk merancang dan mengimplementasikan sebuah prototipe sistem load balancing menggunakan Nginx pada lingkungan virtual hosting untuk meningkatkan

stabilitas, skalabilitas, dan keandalan server.

Masalah

Persoalan utama yang ditemukan pada infrastruktur mitra adalah potensi terjadinya penurunan performa saat lalu lintas pengguna meningkat tajam. Arsitektur yang ada, yang bergantung pada akses ke node server tunggal, sangat rentan terhadap tiga masalah utama:

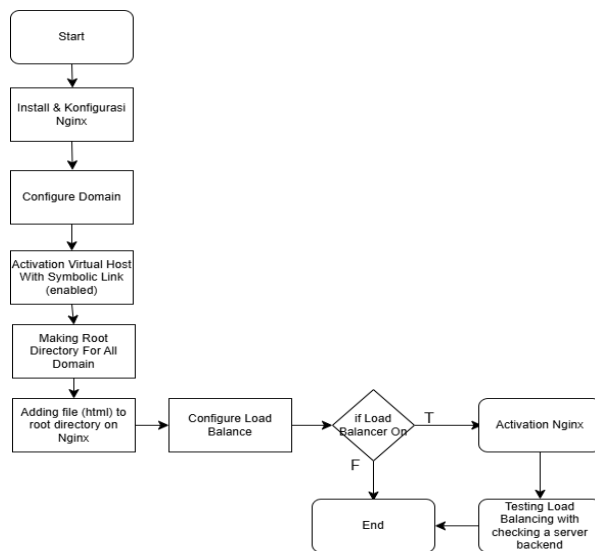
1. **Kelebihan Beban (Overload):** Server tunggal dapat mencapai batas kapasitas CPU, memori, atau koneksi jaringan, yang mengakibatkan waktu respons melambat secara drastis (peningkatan latensi) bagi semua pengguna.
2. **Single Point of Failure (SPOF):** Kegagalan pada perangkat keras atau perangkat lunak server tunggal akan menyebabkan seluruh layanan tidak dapat diakses, menciptakan gangguan signifikan terhadap kegiatan operasional.
3. **Skalabilitas Terbatas:** Menambah kapasitas server tunggal (vertical scaling) sering kali lebih mahal dan memiliki batasan fisik, berbeda dengan penambahan server baru ke dalam sebuah kluster (horizontal scaling) yang lebih fleksibel dan hemat biaya.

Masalah-masalah ini secara langsung berkaitan dengan target kegiatan, yaitu menciptakan sebuah arsitektur server yang lebih tangguh dan dapat diandalkan melalui distribusi beban kerja.

Metode

Metode yang digunakan untuk menyelesaikan persoalan ini adalah kombinasi antara Simulasi Ipteks dan Substitusi Ipteks. Kegiatan ini menyimulasikan arsitektur server kluster di lingkungan virtual dan mengusulkan substitusi dari arsitektur server tunggal ke arsitektur terdistribusi yang lebih modern dan efisien menggunakan Nginx.

Tahapan implementasi teknis dari metode tersebut digambarkan dalam flowchart berikut:



Gambar 1. Flowchart Proses Implementasi

Perangkat lunak utama yang digunakan adalah Nginx dan Red Hat Enterprise Linux (RHEL). Nginx dipilih karena kinerjanya yang unggul untuk lalu lintas HTTP, sedangkan RHEL dipilih karena fitur keamanan bawaan seperti SELinux serta stabilitasnya untuk lingkungan produksi.

Teknik pengumpulan data dilakukan melalui studi literatur terhadap dokumentasi teknis, serta observasi kebutuhan pada mitra. Teknik analisis data menggunakan metode pengujian fungsional, di mana fungsionalitas sistem dievaluasi secara kualitatif berdasarkan hasil keluaran dari setiap perintah dan pengujian akses.

Kegiatan dilaksanakan di lingkungan laboratorium virtual Universitas Internasional Batam dari 02 September 2024 hingga 13 Desember 2024, yang mereplikasi infrastruktur server yang ada di PT. Kinema Systrans Multimedia (Infinite Learning).

Pembahasan

Implementasi solusi menghasilkan sebuah prototipe sistem load balancing yang fungsional. Fokus utama dari luaran ini adalah konfigurasi Nginx yang mampu mendistribusikan lalu lintas HTTP secara merata.

1. Model Solusi yang Diterapkan

Model yang diterapkan adalah arsitektur reverse proxy dengan Nginx sebagai penyeimbang beban. Nginx menerima semua permintaan masuk pada port 80 dan meneruskannya ke salah satu dari tiga server backend. Metode distribusi yang digunakan adalah Round Robin, yang merupakan metode default Nginx. Algoritma ini sederhana namun efektif untuk aplikasi di mana setiap permintaan memiliki beban komputasi yang seragam.

2. Dokumentasi Implementasi

Proses implementasi didokumentasikan dalam bentuk skrip konfigurasi. Konfigurasi ini dari load balancer melibatkan dua blok utama pada Nginx. Pertama, blok upstream digunakan untuk mendefinisikan grup server backend yang akan menerima lalu lintas, dengan mendaftarkan alamat IP dan port dari setiap server. Kedua, blok server dikonfigurasi untuk mendengarkan permintaan masuk pada port 80 dan menggunakan direktif proxy_pass untuk meneruskan lalu lintas tersebut ke grup upstream yang telah dibuat.

```

upstream backend_servers {
    #hash $request_uri;
    #ip_hash;
    #least_conn;
    server node1:81;
    server node1:82;
    server node1:83;
}

server {
    listen 80;
    access_log /var/log/nginx/demo-access.log;
    error_log /var/log/nginx/demo-error.log;

    location / {
        proxy_pass http://backend_servers;
    }
}
  
```

Gambar 2. Konfigurasi Load Balancer

Hasil pengujian fungsionalitas menggunakan curl menunjukkan keberhasilan implementasi, di mana setiap permintaan secara berurutan diterima oleh server yang berbeda.

```
[student@workstation ~]$ curl node1
Web Application 1
Load Balancer 1
[student@workstation ~]$ curl node1
Web Application 1
Load Balancer 2
[student@workstation ~]$ curl node1
Web Application 1
Load Balancer 3
[student@workstation ~]$ curl node1
Web Application 1
Load Balancer 1
```

Gambar 3. Pengujian Curl pada Load Balancer

3. Keunggulan dan Kelemahan

Keunggulan utama dari solusi ini adalah kesederhanaan implementasi dan peningkatan drastis dalam hal ketersediaan dan skalabilitas. Dengan arsitektur ini, kegagalan pada satu server backend tidak akan menghentikan layanan. Penambahan kapasitas dapat dilakukan dengan mudah dengan menambahkan server baru. Kelemahannya adalah metode Round Robin tidak mempertimbangkan beban aktual atau kapasitas masing-masing server. Algoritma dinamis seperti Weighted Least Connections bisa lebih unggul dalam lingkungan server yang heterogen.

4. Tingkat Kesulitan dan Peluang

Tantangan utama dalam implementasi adalah konfigurasi keamanan, khususnya SELinux. Memastikan konteks keamanan yang benar untuk port dan koneksi jaringan Nginx memerlukan pemahaman yang mendalam. Peluang pengembangan sangat besar, termasuk adopsi platform orkestrasi kontainer seperti Kubernetes untuk skalabilitas yang lebih dinamis dan penerapan sistem

pemantauan real-time menggunakan Grafana dan Prometheus untuk pengambilan keputusan load balancing yang lebih cerdas.

Simpulan

Kegiatan pengabdian ini berhasil mencapai target utamanya, yaitu membangun sistem load balancing yang fungsional menggunakan Nginx. Penerapan metode Simulasi dan Substitusi Ipteks terbukti sangat tepat, karena memungkinkan perancangan dan pengujian solusi teknis dalam lingkungan yang terkontrol sebelum implementasi skala penuh. Dampak dan manfaat utama dari kegiatan ini adalah tersedianya sebuah prototipe arsitektur server yang andal, skalabel, dan memiliki ketersediaan tinggi, yang secara efektif menjawab masalah single point of failure pada infrastruktur tunggal. Untuk pengembangan selanjutnya, direkomendasikan agar dilakukan analisis kinerja kuantitatif terhadap berbagai algoritma load balancing dan mengintegrasikan sistem pemantauan real-time untuk visualisasi beban server yang lebih baik.

Daftar Pustaka

Gea, C., Lase, K. J. D., & Syamsudin, M. (2023). Implementasi virtual private server untuk mini hosting. *Indonesian Journal on Information and Computer Science*, 7(1), 5–9.

Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2022). Load balancing techniques in cloud computing environment: A review. *Journal of King Saud University - Computer and Information Sciences*, 34(7), 3910–3933. <https://doi.org/10.1016/j.jksuci.2021.02.007>

Hamdan, M., Hassan, E., Abdelaziz, A., Elhigazi, A., Mohammed, B., Khan, S., Vasilakos, A. V., & Marsono, M. N. (2021). A comprehensive survey of load balancing techniques in software-defined

network. *Journal of Network and Computer Applications*, 174, Article 102856.

<https://doi.org/10.1016/j.jnca.2020.102856>

Shafiq, D. A., Jhanjhi, N. Z., Abdullah, A., & Alzain, M. A. (2021). A load balancing algorithm for the data centres to optimize cloud computing applications. *IEEE Access*, 9, 41731–41744.
<https://doi.org/10.1109/ACCESS.2021.3065308>

Khunger, A., & Anand, K. (2023, June 25). *Fault-tolerant load balancing in cloud-based financial analytics: A reinforcement learning approach*. SSRN.
<https://doi.org/10.2139/ssrn.5146521>

P., A., Sharma, A., V., D. G., Sharma, A., S., K., & Arun, M. R. (2022). *Priority queueing model-based IoT middleware for load balancing* [Paper presentation]. 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India.
<https://doi.org/10.1109/ICICCS53718.2022.9788218>

Omer, Y. A. H., Mohammedel-Amin, M. A., & Mustafa, A. B. A. (2021). *Load balance in cloud computing using software defined networking* [Paper presentation]. 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, Sudan.
<https://doi.org/10.1109/ICCCEEE49695.2021.9429607>

Pradhan, A., & Bisoy, S. K. (2022). A novel load balancing technique for cloud computing platform based on PSO. *Journal of King Saud University - Computer and Information Sciences*, 34(7), 3988–3995.
<https://doi.org/10.1016/j.jksuci.2020.10.016>

Hung, L.-H., Wu, C.-H., Tsai, C.-H., & Huang, H.-C. (2021). Migration-based load balance of virtual machine servers in cloud computing by load prediction using genetic-based methods. *IEEE Access*, 9, 49760–49773.

<https://doi.org/10.1109/ACCESS.2021.3065170>

Lee, J.-B., Yoo, T.-H., Lee, E.-H., Hwang, B.-H., Ahn, S.-W., & Cho, C.-H. (2021). High-performance software load balancer for cloud-native architecture. *IEEE Access*, 9, 123704–123716.
<https://doi.org/10.1109/ACCESS.2021.3108801>

Patra, M. K., Misra, S., Sahoo, B., & Turuk, A. K. (2022). GWO-Based Simulated Annealing Approach for Load Balancing in Cloud for Hosting Container as a Service. *Applied Sciences*, 12(21), 11115.
<https://doi.org/10.3390/app122111115>

Ibrahim, I. M., Ameen, S. Y., Yasin, H. M., Omar, N., Kak, S. F., Rashid, Z. N., Salih, A. A., Salim, N. O. M., & Ahmed, D. M. (2021). Web server performance improvement using dynamic load balancing techniques: A review. *Asian Journal of Research in Computer Science*, 10(1), 47–62.
<https://doi.org/10.9734/AJRCOS/2021/v10i130234>

Assudani, P. J., & Balakrishnan, P. (2023). An efficient approach for load balancing of VMs in cloud environment. *Applied Nanoscience*, 13, 1313–1326.
<https://doi.org/10.1007/s13204-021-02014-z>

Moparthy, N. R., Balakrishna, G., Chithaluru, P., Kolla, M., & Kumar, M. (2023). An improved energy-efficient cloud-optimized load-balancing for IoT frameworks. *Heliyon*, 9(11).

Tawfeeg, T. M., Yousif, A., Hassan, A., Alqhtani, S. M., Hamza, R., Bashir, M. B., & Ali, A. (2022). Cloud dynamic load balancing and reactive fault tolerance techniques: a systematic literature review (SLR). *IEEE Access*, *10*, 71853-71873.