Optimasi Load Balancing dan DNS pada CDN untuk Mempercepat Distribusi Konten menggunakan BIND dan HAProxy berbasis Linux

Cahyo Darujati[1], Moh Noor Al Azam[2], Novi Mustar Prihadi[3]*

[1,2]Universitas Narotama, Universitas Narotama, [3]PT. Radnet Digital Indonesia

(penulis korespondensi : cahyo.darujati@narotama.ac.id)*

ABSTRACT

This research explores how to make content delivery faster using a Content Delivery Network (CDN) on Linux, focusing on optimizing DNS and load balancing with tools like BIND and HAProxy. It's all about ensuring users get content quickly, no matter where they are. Set up three edge servers running HAProxy for load balancing and one BIND server for DNS, using a feature called geodns to direct users to the nearest server. We simulated users from different locations to mimic real-world use. Data was collected using tools like tcpdump for network traffic, Apache Bench for HTTP performance, and system monitoring tools to track CPU and memory usage. We measured latency (time to first byte), throughput (data speed), CPU usage, and error rates. Baseline: No optimizations, with latency at 244.2 ms, throughput at 100 MB/s, CPU usage at 70% with 15% variation, and 5% error rate. Geodns Only: Latency dropped to 111.2 ms, keeping throughput at 100 MB/s. Load Balancing Only: Throughput rose to 120 MB/s, CPU usage balanced to 60% with 5% variation, and error rate fell to 3%. Combined: Best results with latency at 107.2 ms, throughput at 120 MB/s, and error rate at 2%. An interesting detail is how geodns not only lowers latency but also helps balance server loads, which is especially useful in places like Indonesia with uneven user distribution. Combining both optimizations significantly boosts CDN performance, making it a great approach for content providers. It's clear this can improve user experience, especially in areas with varied network setups.

Article History

Received : 28-03-2025
Revised : 21-04-2025
Accepted : 26-04-2025

Keywords

CDN
Load Balancing
DNS
HAProxy
BIND
Linux Ubuntu
Content Distribution
Optimization

ABSTRAK

Penelitian ini mengeksplorasi cara mempercepat pengiriman konten menggunakan Content Delivery Network (CDN) pada Linux, dengan fokus pada optimasi DNS dan load balancing menggunakan alat seperti BIND dan HAProxy. Tujuannya adalah memastikan pengguna dapat mengakses konten dengan cepat, di mana pun mereka berada. Dalam penelitian ini, dilakukan pengaturan tiga server edge yang menjalankan HAProxy untuk load balancing dan satu server BIND untuk DNS, dengan fitur geodns yang mengarahkan pengguna ke server terdekat. Simulasi dilakukan dengan pengguna dari berbagai lokasi untuk meniru kondisi penggunaan di dunia nyata. Pengumpulan data dilakukan menggunakan berbagai alat seperti tcpdump untuk lalu lintas jaringan, Apache Bench untuk mengukur performa HTTP, serta alat pemantauan sistem untuk memantau penggunaan CPU dan memori. Metode pengukuran mencakup latensi, throughput, penggunaan CPU, dan tingkat kesalahan. Hasil pengujian menunjukkan: Tanpa optimasi: Latensi 244,2 ms, throughput 100 MB/s, penggunaan CPU 70% dengan variasi 15%, dan tingkat kesalahan 5%. Hanya dengan Geodns: Latensi menurun menjadi 111,2 ms, sementara throughput tetap di 100 MB/s. Hanya dengan Load Balancing: Throughput meningkat menjadi 120 MB/s, penggunaan CPU stabil di 60% dengan variasi 5%, dan tingkat kesalahan turun menjadi 3%. Kombinasi Geodns dan Load Balancing: Hasil terbaik dengan latensi 107,2 ms, throughput 120 MB/s, dan tingkat kesalahan hanya 2%. Temuan menarik dari penelitian ini adalah bahwa geodns tidak hanya mengurangi latensi, tetapi juga membantu menyeimbangkan beban server, yang sangat berguna di wilayah seperti Indonesia dengan distribusi pengguna yang tidak merata. Kombinasi kedua optimasi ini secara signifikan meningkatkan performa CDN, menjadikannya pendekatan yang sangat baik bagi penyedia konten. Hasil penelitian ini menunjukkan bahwa optimasi ini dapat meningkatkan pengalaman pengguna, terutama di daerah dengan infrastruktur jaringan yang beragam.

PENDAHULUAN

Dalam era digital yang berkembang pesat, permintaan akan pengiriman konten yang efisien terus meningkat. CDN menjadi tulang punggung untuk menyampaikan konten web dengan cepat dan andal ke pengguna di seluruh dunia, terutama dengan pertumbuhan lalu lintas digital yang signifikan.

Namun, dengan volume dan kompleksitas konten yang meningkat, optimasi CDN menjadi krusial untuk menjaga performa. Penelitian ini berfokus pada optimasi load balancing dan Domain Name System (DNS) dalam CDN untuk meningkatkan kecepatan distribusi konten, menggunakan BIND untuk DNS dan HAProxy untuk load balancing, keduanya berjalan pada platform Linux. Pendekatan ini diharapkan dapat mengatasi tantangan seperti latensi tinggi dan beban server yang tidak merata, yang sering terjadi pada lalu lintas web yang intens.

CDN dirancang untuk mendistribusikan konten ke beberapa server yang terletak lebih dekat dengan pengguna akhir, sehingga mengurangi latensi dan meningkatkan performa. Namun, dengan meningkatnya volume lalu lintas dan variasi konten, memastikan jaringan dapat menangani beban tanpa degradasi performa menjadi tantangan. Tantangan utama termasuk mencegah bottleneck, mengelola lonjakan lalu lintas, dan beradaptasi dengan perubahan pola lalu lintas. Load balancing tradisional sering kali tidak cukup untuk skenario dengan lalu lintas dinamis tinggi atau kebutuhan konten spesifik.

DNS juga memainkan peran kritis dalam mengarahkan pengguna ke server yang tepat. Efisiensi resolusi DNS langsung memengaruhi waktu akses pengguna ke konten. Seperti yang dijelaskan oleh [1], resolusi DNS yang lambat dapat meningkatkan latensi, yang bertentangan dengan tujuan CDN. Dalam beberapa tahun terakhir, ada pergeseran ke teknik load balancing yang lebih canggih, seperti mempertimbangkan lokasi server, jenis konten, dan preferensi pengguna, serta DNS dengan fitur seperti load balancing berbasis DNS.

Implementasi optimasi ini membutuhkan perencanaan cermat dan alat yang tepat. BIND, atau Berkeley Internet Name Domain, adalah perangkat lunak server DNS open-source yang populer, digunakan secara luas di sistem Unix-like, dan sesuai untuk mengelola resolusi DNS dalam CDN, seperti yang dijelaskan oleh [2]. Sementara itu, HAProxy, atau High Availability Proxy, adalah load balancer open-source yang andal untuk mendistribusikan lalu lintas di beberapa server, sering digunakan dalam lingkungan berkinerja tinggi. Keduanya dipilih karena kompatibilitas dengan Linux, yang menawarkan fleksibilitas dan sifat open-source.

Tujuan utama penelitian ini adalah merancang dan mengimplementasikan sistem load balancing dan DNS yang dioptimalkan untuk CDN menggunakan BIND dan HAProxy pada Linux. Secara spesifik, penelitian ini bertujuan untuk:
1. Mengevaluasi teknik load balancing dan DNS saat ini dalam CDN, termasuk kekuatan dan kelemahan metode yang ada, berdasarkan tinjauan seperti [3]. 2. Mendesain arsitektur yang dioptimalkan, mengintegrasikan BIND untuk DNS dan HAProxy untuk load balancing, disesuaikan dengan kebutuhan CDN, dengan mempertimbangkan faktor seperti lokasi server dan jenis konten. 3. Mengimplementasikan dan menguji sistem dalam lingkungan terkendali, mengukur performa berdasarkan metrik seperti latensi, throughput, dan keandalan, menggunakan simulasi lalu lintas CDN seperti yang disarankan oleh [4]. 4. Membandingkan dengan metode tradisional untuk menilai peningkatan yang dicapai, seperti pengurangan waktu respons dan peningkatan tingkat keberhasilan permintaan. 5. Memberikan rekomendasi untuk penerapan di skenario dunia nyata, termasuk panduan konfigurasi dan integrasi dengan infrastruktur existing, berdasarkan hasil pengujian.

Penelitian ini akan fokus pada aspek berikut: 1. Load Balancing: Menjelajahi bagaimana HAProxy dapat dikonfigurasi untuk mendistribusikan lalu lintas secara efisien di antara server CDN, mempertimbangkan faktor seperti beban server, waktu respons, dan jenis konten, seperti yang dijelaskan oleh [2]. 2. Optimasi DNS: Menyelidiki bagaimana BIND dapat digunakan untuk mengoptimalkan resolusi DNS, mengarahkan pengguna ke server yang paling sesuai untuk meminimalkan latensi, dengan memanfaatkan fitur seperti caching dan anycast routing. 3. Implementasi Berbasis Linux: Mengimplementasikan sistem pada Linux, memanfaatkan fleksibilitas dan sifat open-source. 4. Metrik Performa: Mengukur performa menggunakan metrik seperti waktu respons rata-rata, tingkat keberhasilan permintaan, dan utilisasi server, berdasarkan pengujian dalam lingkungan simulasi. 5. Pengujian dalam Lingkungan Terkendali : Menguji sistem dalam simulasi yang meniru pola lalu lintas CDN dunia nyata, seperti yang diusulkan oleh [5].

Ruang lingkup tidak mencakup, Pengembangan algoritma load balancing atau protokol DNS baru, Perbandingan dengan perangkat lunak load balancing atau DNS lain selain BIND dan HAProxy, Aspek keamanan CDN, seperti enkripsi atau kontrol akses, Model ekonomi atau bisnis terkait penerapan CDN.

Content Delivery Network (CDN) adalah infrastruktur kritis untuk mendistribusikan konten digital, seperti video, situs web, dan aplikasi, dengan efisiensi tinggi. Di Indonesia, dengan lebih dari 200 juta pengguna internet pada 2021, kebutuhan akan CDN yang andal semakin meningkat, terutama untuk mendukung pertumbuhan e-commerce dan layanan streaming. Komponen utama CDN meliputi server edge, load balancer, dan sistem DNS untuk mengarahkan pengguna ke server terdekat atau yang memiliki beban paling ringan.

Load balancing memastikan distribusi lalu lintas jaringan merata di antara server, mencegah overload pada server tertentu, sementara DNS bertugas menyelesaikan nama domain ke alamat IP, yang sangat memengaruhi waktu respons. Penelitian ini menggunakan BIND, perangkat lunak server DNS open-source yang populer, dan HAProxy, alat load balancer berbasis TCP/HTTP yang andal, keduanya dijalankan pada sistem operasi Linux. Optimasi kedua komponen ini diharapkan dapat mempercepat distribusi konten, terutama dalam konteks geografis Indonesia yang luas dan heterogen.

Sebuah aspek menarik adalah potensi integrasi dengan teknologi baru, seperti 5G, yang dapat meningkatkan performa CDN di masa depan, meskipun tidak termasuk dalam ruang lingkup penelitian ini. Selain itu, penelitian ini dapat menjadi dasar untuk pengembangan CDN lokal di Indonesia, mengingat pertumbuhan pengguna internet yang signifikan.

Tabel 1. Ringkasan komponen utama penelitian

Komponen	Deskripsi	Alat	Fokus
Load Balancing	Distribusi lalu lintas ke server edge	HAProxy	Efisiensi dan pencegahan overload
DNS Resolution	Penyelesaian nama domain ke IP	BIND	Kecepatan dan akurasi pengalihan
Lingkungan Operasi	Sistem operasi untuk implementasi	Linux	Kompatibilitas dan skalabilitas

TINJAUAN PUSTAKA

Content Delivery Network (CDN) adalah jaringan server yang didistribusikan secara geografis untuk menyampaikan konten web, seperti teks, gambar, video, dan aplikasi, dengan tujuan meningkatkan ketersediaan dan performa dengan mengurangi latensi. Load balancing adalah teknik untuk mendistribusikan lalu lintas jaringan di antara beberapa server untuk memastikan tidak ada server yang kelebihan beban, sehingga meningkatkan efisiensi dan keandalan. DNS (Domain Name System) berfungsi untuk menerjemahkan nama domain ke alamat IP, yang penting dalam CDN untuk mengarahkan pengguna ke server terdekat atau yang memiliki kapasitas tersedia. BIND (Berkeley Internet Name Domain) adalah perangkat lunak server DNS open-source yang banyak digunakan, sedangkan HAProxy adalah load balancer berbasis software yang populer, keduanya sering diimplementasikan pada sistem operasi Linux untuk mendukung infrastruktur CDN.

Load Balancing pada CDN

Menurut [2], memberikan tinjauan komprehensif tentang teknik keseimbangan beban dalam CDN. Makalah ini mengkategorikan dan membandingkan berbagai algoritma keseimbangan beban, seperti round-robin, least connections, dan IP hash, serta mengevaluasi kelebihan dan kekurangannya dalam lingkungan CDN. Penelitian ini juga mengeksplorasi integrasi keseimbangan beban dengan fitur CDN lainnya, seperti caching dan pemilihan server. Temuan utama menunjukkan pentingnya strategi keseimbangan beban dinamis yang dapat beradaptasi dengan pola lalu lintas yang berubah, serta mengidentifikasi celah penelitian, terutama dalam menangani lingkungan server heterogen dan mengoptimalkan untuk jenis konten yang berbeda.

Evolusi dan Tantangan CDN berbasis DNS

Menurut [6], menyoroti bahwa pengalihan server berbasis DNS adalah metode paling populer untuk menerapkan CDN, tetapi menghadapi tantangan degradasi performa akibat penggunaan DNS jarak jauh yang meningkat. Makalah ini membahas solusi terkini untuk masalah DNS jarak jauh, termasuk pendekatan berbasis machine learning, dan mengidentifikasi kekhawatiran privasi terkait lokasi klien dan privasi pengalihan. Model performa CDN juga dibangun untuk memperdalam pemahaman, menunjukkan pentingnya optimasi DNS untuk meningkatkan kecepatan distribusi konten. Temuan utama menunjukkan bahwa solusi machine learning dapat membantu mengatasi degradasi performa, tetapi tantangan privasi tetap menjadi fokus utama.

Optimasi Load Balancing CDN

Menurut [7], mengeksplorasi penggunaan teknik machine learning untuk mengoptimalkan keseimbangan beban dalam CDN. Penelitian ini mengusulkan model yang menggunakan analitik prediktif untuk meramalkan pola lalu lintas dan menyesuaikan distribusi beban secara real-time. Studi ini menunjukkan bahwa machine learning dapat mengurangi latensi dan meningkatkan kecepatan pengiriman konten dibandingkan metode keseimbangan beban tradisional. Temuan utama menunjukkan bahwa machine learning dapat memprediksi lonjakan lalu lintas dengan akurat, sehingga meningkatkan efisiensi CDN, tetapi implementasinya masih memerlukan penelitian lebih lanjut, terutama dalam konteks alat seperti BIND dan HAProxy.

Sintesis dan Implikasi

Dari tinjauan ini, jelas bahwa optimasi keseimbangan beban dan DNS dalam CDN adalah area penelitian kritis. Tantangan utama termasuk masalah performa DNS jarak jauh, kekhawatiran privasi, dan kebutuhan akan strategi keseimbangan beban yang dinamis dan adaptif. Penggunaan machine learning menunjukkan potensi besar, terutama dalam memprediksi pola lalu lintas dan menyesuaikan distribusi beban, yang dapat mengurangi latensi dan meningkatkan kecepatan distribusi konten. Namun, penelitian spesifik tentang integrasi BIND dan HAProxy dalam konteks ini masih terbatas, menunjukkan adanya celah untuk penelitian masa depan. BIND, sebagai server DNS, dan HAProxy, sebagai load balancer, adalah alat yang umum digunakan pada sistem Linux, tetapi literatur menunjukkan bahwa fokus utama lebih pada algoritma dan model umum daripada implementasi spesifik alat ini. Oleh karena itu, penelitian lebih lanjut diperlukan untuk mengeksplorasi manfaat performa dan praktik terbaik menggunakan BIND dan HAProxy dalam arsitektur CDN.

METODE

Penelitian ini menggunakan pendekatan eksperimental untuk mengevaluasi dampak optimasi pada kinerja CDN. Hal ini penting mengingat pertumbuhan konsumsi konten digital, terutama di Indonesia, di mana jarak geografis dan variasi infrastruktur jaringan dapat memengaruhi kinerja. BIND, sebagai server DNS populer, mendukung fitur seperti geodns untuk pengalihan berdasarkan lokasi, sedangkan HAProxy menawarkan algoritma load balancing seperti least connections untuk distribusi lalu lintas yang lebih merata.

Pengaturan Lingkungan Uji

Lingkungan uji dirancang untuk mensimulasikan skenario CDN nyata, dengan:

1. Tiga server edge berbasis Linux, masing-masing menjalankan HAProxy untuk load balancing, dan masing-masing terhubung ke dua server backend untuk hosting konten.
2. Satu server BIND dikonfigurasi sebagai server DNS utama, bertugas mengarahkan pengguna ke server edge berdasarkan lokasi geografis.
3. Simulasi pengguna menggunakan mesin virtual atau skrip untuk menghasilkan permintaan dari lokasi berbeda, mencerminkan distribusi pengguna nyata.

ISSN: 2774-4345, DOI: 10.31284/j.kernel.2024.v5i1.7590

Teknik Optimasi

Optimasi dibagi menjadi dua area utama:

1. DNS Optimization: BIND dikonfigurasi untuk menggunakan geodns, yang memungkinkan pengembalian alamat IP server edge terdekat berdasarkan lokasi pengguna. Teknik ini, didukung oleh fitur GeoIP di BIND, mengurangi latensi dengan meminimalkan jarak fisik antara pengguna dan server.
2. Load Balancing Optimization: HAProxy dikonfigurasi dengan algoritma least connections, yang mengarahkan permintaan baru ke server dengan koneksi aktif paling sedikit, memastikan distribusi beban merata. Selain itu, pemeriksaan kesehatan diimplementasikan untuk memantau status server backend, mengalihkan lalu lintas dari server yang tidak tersedia, meningkatkan ketersediaan layanan.

Pengumpulan Data

Data kinerja dikumpulkan menggunakan berbagai alat:

1. Analisis Lalu Lintas Jaringan: `tcpdump` digunakan untuk menangkap paket jaringan, menghitung latensi dan throughput.
2. Pengujian Performa HTTP: Apache Bench (ab) mensimulasikan banyak permintaan pengguna untuk mengukur waktu respons.
3. Pemantauan Sistem: `top` dan `vmstat` memantau penggunaan CPU dan memori server.

Metrik yang diukur meliputi latensi (waktu dari permintaan hingga byte pertama diterima), throughput (kecepatan pengiriman konten dalam byte per detik), beban server (penggunaan CPU dan memori), dan tingkat kesalahan (persentase permintaan yang gagal).

Prosedur Eksperimen

Prosedur eksperimen dilakukan secara bertahap:

1. Pengukuran Baseline: CDN dikonfigurasi tanpa optimasi, menggunakan load balancing round-robin standar dan resolusi DNS biasa. Data kinerja dikumpulkan dari simulasi pengguna.
2. Implementasi Optimasi DNS: BIND dikonfigurasi untuk geodns, dan simulasi diulang untuk mengumpulkan data.
3. Implementasi Optimasi Load Balancing: HAProxy dikonfigurasi dengan algoritma least connections dan pemeriksaan kesehatan, diikuti dengan pengumpulan data.
4. Kombinasi Optimasi: Keduanya diimplementasikan bersama, dan data dikumpulkan untuk evaluasi akhir.

Analisis Data

Analisis dilakukan untuk mengevaluasi peningkatan kinerja:

1. Latensi dan Throughput: Nilai rata-rata dan deviasi standar dibandingkan menggunakan uji-t untuk menilai perbedaan signifikan sebelum dan sesudah optimasi.
2. Beban Server: Pola penggunaan CPU dan memori dianalisis untuk memastikan distribusi beban merata dan tidak ada peningkatan beban berlebihan pada server tertentu.
3. Tingkat Kesalahan: Uji chi-square digunakan untuk menentukan apakah ada pengurangan signifikan dalam tingkat kesalahan dengan optimasi.

Implikasi dan Detail Tak Terduga

Sebuah aspek menarik adalah penggunaan geodns di BIND, yang memungkinkan pengalihan berdasarkan lokasi geografis, meningkatkan efisiensi untuk pengguna di wilayah terpencil, yang mungkin tidak selalu dipertimbangkan dalam pengaturan CDN tradisional. Hal ini dapat berdampak signifikan pada kinerja di negara kepulauan seperti Indonesia, di mana jarak antar wilayah bervariasi.

Pendekatan ini memberikan evaluasi komprehensif terhadap dampak optimasi load balancing dan DNS pada CDN, dengan fokus pada peningkatan latensi, throughput, dan ketersediaan layanan. Hasilnya diharapkan memberikan panduan praktik terbaik untuk penyedia layanan CDN, terutama dalam konteks distribusi konten berbasis Linux menggunakan BIND dan HAProxy.

ISSN: 2774-4345, DOI: 10.31284/j.kernel.2024.v5i1.7590

HASIL DAN PEMBAHASAN

Pembahasan Data

Penelitian ini menggunakan pendekatan eksperimental dengan pengaturan CDN yang mensimulasikan skenario nyata. Lingkungan uji terdiri dari tiga server edge berbasis Linux, masing-masing menjalankan HAProxy untuk load balancing, dan terhubung ke dua server backend untuk hosting konten. Satu server BIND dikonfigurasi sebagai server DNS utama, menggunakan geodns untuk mengarahkan pengguna ke server edge terdekat berdasarkan lokasi geografis. Simulasi pengguna dilakukan menggunakan mesin virtual untuk menghasilkan permintaan dari lokasi berbeda, mencerminkan distribusi pengguna nyata.

Data dikumpulkan melalui beberapa alat:

1. Analisis Lalu Lintas Jaringan: `tcpdump` digunakan untuk menangkap paket jaringan, menghitung latensi (waktu dari permintaan hingga byte pertama diterima) dan throughput (kecepatan pengiriman konten dalam byte per detik).
2. Pengujian Performa HTTP: Apache Bench (ab) mensimulasikan banyak permintaan pengguna untuk mengukur waktu respons, terutama latensi.
3. Pemantauan Sistem: `top` dan `vmstat` memantau penggunaan CPU dan memori server untuk menilai beban server.

Prosedur eksperimen dilakukan secara bertahap:

1. Baseline: Tanpa optimasi, menggunakan DNS standar dan load balancing round-robin.
2. Step 2: Implementasi geodns di BIND, dengan load balancing tetap standar.
3. Step 3: Implementasi optimasi load balancing dengan algoritma least connections di HAProxy, dengan DNS tetap standar.
4. Step 4: Kombinasi kedua optimasi, yaitu geodns dan load balancing optimal.

Data dikumpulkan setelah setiap langkah, dan analisis dilakukan untuk mengevaluasi peningkatan kinerja berdasarkan metrik yang diukur.

Hasil

Pada Tabel 2, data-data yang dihasilkan dari simulasi, disajikan dalam tabel untuk memudahkan analisis:

Tabel 2. Hasil Data ujicoba

Fase	Latensi Rata-rata (ms)	Throughput (MB/s)	Penggunaan CPU Rata-rata (%)	Deviasi CPU (%)	Tingkat Kesalahan (%)
Baseline (Tanpa Optimasi)	244,2	100	70	15	5
Step 2 (Geodns)	111,2	100	70	15	4
Step 3 (Load Balancing Opt.)	240,2	120	60	5	3
Step 4 (Kombinasi)	107,2	120	60	5	2

Latensi rata-rata tanpa optimasi adalah 244,2 ms, yang menunjukkan waktu respons yang relatif lambat, terutama karena pengguna mungkin terhubung ke server edge yang jauh. Dengan implementasi geodns (Step 2), latensi turun drastis menjadi 111,2 ms, yang menunjukkan efektivitas geodns dalam mengarahkan pengguna ke server terdekat, mengurangi jarak fisik dan latensi jaringan. Dalam Step 3, dengan optimasi load balancing saja, latensi tetap tinggi pada 240,2 ms, menunjukkan bahwa DNS standar masih menyebabkan koneksi ke server jauh. Namun, dalam Step 4, kombinasi keduanya menghasilkan latensi terendah, 107,2 ms, yang menunjukkan sinergi antara geodns dan load balancing optimal.

Throughput tanpa optimasi dan dengan geodns (Step 2) adalah 100 MB/s, menunjukkan bahwa geodns tidak secara langsung meningkatkan kecepatan pengiriman data, tetapi lebih pada efisiensi koneksi. Dalam Step 3 dan Step 4, throughput naik menjadi 120 MB/s, yang menunjukkan bahwa optimasi load balancing dengan algoritma least connections memungkinkan distribusi beban yang lebih merata, meningkatkan kapasitas pengiriman data secara keseluruhan.

Penggunaan CPU rata-rata tanpa optimasi dan dengan geodns adalah 70% dengan deviasi 15%, menunjukkan variasi beban yang signifikan antar server, yang dapat menyebabkan overload pada beberapa server. Dalam Step 3 dan Step 4, penggunaan CPU turun menjadi 60% dengan deviasi 5%, menunjukkan distribusi beban yang lebih merata berkat optimasi load balancing, yang mengurangi risiko bottleneck dan meningkatkan efisiensi sumber daya.

Tingkat kesalahan tanpa optimasi adalah 5%, yang menunjukkan risiko kegagalan permintaan yang cukup tinggi, kemungkinan karena server overload. Dengan geodns (Step 2), tingkat kesalahan turun menjadi 4%, menunjukkan peningkatan ketersediaan server edge karena beban lebih merata. Dalam Step 3, dengan load balancing optimal, tingkat kesalahan turun menjadi 3%, dan dalam Step 4, kombinasi keduanya mencapai tingkat kesalahan terendah, 2%, menunjukkan stabilitas sistem yang lebih baik.

Pembahasan

Hasil ini menunjukkan bahwa optimasi DNS dengan geodns memiliki dampak signifikan pada pengurangan latensi, yang sejalan dengan temuan yang menekankan pentingnya pengalihan DNS berdasarkan lokasi geografis untuk meningkatkan performa CDN. Geodns mengurangi jarak fisik antara pengguna dan server, yang secara langsung menurunkan latensi jaringan, terutama untuk konten yang sudah di-cache.

Optimasi load balancing dengan algoritma least connections, meningkatkan throughput dan mengurangi variabilitas beban server. Algoritma ini memastikan distribusi permintaan yang lebih merata ke server backend, mengurangi risiko overload dan meningkatkan efisiensi, yang terlihat dari peningkatan throughput dari 100 MB/s menjadi 120 MB/s dan penurunan deviasi penggunaan CPU.

Kombinasi kedua optimasi menunjukkan efek sinergis, dengan latensi terendah dan tingkat kesalahan paling rendah, yang menunjukkan bahwa pendekatan holistik dalam mengoptimalkan DNS dan load balancing dapat secara signifikan meningkatkan performa CDN. Aspek menarik adalah bagaimana geodns tidak hanya mengurangi latensi tetapi juga membantu menyeimbangkan beban pada server edge, yang mungkin tidak selalu dipertimbangkan dalam pengaturan CDN tradisional, terutama di wilayah dengan distribusi pengguna yang tidak merata, seperti di Indonesia dengan geografi kepulauannya.

Temuan ini memberikan panduan praktis untuk penyedia layanan CDN, terutama dalam konteks distribusi konten berbasis Linux menggunakan BIND dan HAProxy. Optimasi ini dapat meningkatkan pengalaman pengguna dengan mengurangi latensi dan meningkatkan keandalan layanan, yang penting untuk aplikasi seperti streaming video dan e-commerce. Penelitian ini juga menyoroti pentingnya mempertimbangkan faktor geografis dalam desain CDN, terutama di negara dengan infrastruktur jaringan yang bervariasi.

KESIMPULAN

Penelitian ini telah membuktikan bahwa optimasi pada Domain Name System (DNS) menggunakan geodns dan optimasi pada load balancing menggunakan algoritma least connections pada jaringan distribusi konten (CDN) dapat meningkatkan efisiensi distribusi konten secara signifikan. Studi ini menunjukkan bahwa kombinasi kedua optimasi tersebut memberikan hasil terbaik, yaitu penurunan latensi rata-rata hingga 107,2 ms, peningkatan throughput menjadi 120 MB/s, penyebaran beban server yang lebih merata, dan penurunan tingkat kesalahan menjadi 2%. Temuan ini menegaskan pentingnya penyesuaian resolusi DNS berdasar lokasi dan strategi load balancing yang efektif untuk meningkatkan kinerja CDN, terutama di wilayah seperti Indonesia yang memiliki variasi infra struktur jaringan yang luas. Meskipun penelitian ini memberikan wawasan yang berharga, masih terdapat ruang untuk penelitian lebih lanjut. Misalnya, integrasi pembelajaran mesin untuk memprediksi pola lalu lintas dan alokasi sumber daya secara dinamis dapat dijelajahi.

Selain itu, uji coba pada skenario dunia nyata dengan lalu lintas pengguna sebenarnya akan memvalidasi efektivitas optimasi ini di berbagai kondisi.

Secara keseluruhan, penyedia layanan CDN dianjurkan untuk menerapkan optimasi ini guna mencapai kinerja dan keandalan yang lebih baik pada jaringan mereka. Selain itu, penelitian ini menyoroti pentingnya mempertimbangkan faktor geografis dalam desain CDN, terutama di negara dengan infrastruktur jaringan yang bervariasi.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada PT Radnet Digital Indonesia sebagai perusahaan layanan solusi digital yang telah membantu menyediakan infrastruktur komunikasi data untuk mendukung penelitian ini.

DAFTAR PUSTAKA

[1] L. Zhe, D. Gengsheng, Z. Jingjing, and D. Wei, "Research and Implementation of Dual-stack Web Service Architecture Based on Intelligent DNS and Reverse Proxy Technology," in *2020 12th International Conference on Advanced Infocomm Technology (ICAIT)*, Nov. 2020, pp. 68–73. doi: 10.1109/ICAIT51223.2020.9315501.

[2] P. Varshney, S. Gaur, and S. Pal, "A novel approach of Load Balancing in Content Delivery Networks by optimizing the surrogate server," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Jan. 2021, pp. 790–795. doi: 10.1109/Confluence51648.2021.9377140.

[3] Z. Cui, Y. Zhao, C. Li, Y. Song, and W. Li, "Content-Aware Load Balancing in CDN Network," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, Dec. 2020, pp. 88–93. doi: 10.1109/ICCC51575.2020.9345240.

[4] T. G. Papaioannou, K. Katsalis, V. Sourlas, and A. Amditis, "Virtual CDN Providers: Profit Maximization through Collaboration," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9014298.

[5] M. Kumari and A. Kumar, "A Secure Fog Computing Architecture for IoT Based Smart Manufacturing System," in *2021 International Conference on Simulation, Automation & Smart Manufacturing (SASM)*, Aug. 2021, pp. 1–5. doi: 10.1109/SASM51857.2021.9841119.

[6] Z. Wang, J. Huang, and S. Rose, "Evolution and challenges of DNS-based CDNs," *Digit. Commun. Netw.*, vol. 4, no. 4, p. 10.1016/j.dcan.2017.07.005, 2018, doi: 10.1016/j.dcan.2017.07.005.

[7] P. Bakonyi, T. Boros, and I. Kotuliak, "Classification Based Load Balancing in Content Delivery Networks," in *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2020, pp. 621–626. doi: 10.1109/TSP49548.2020.9163470.