

Membangun Content Delivery Network (CDN) menggunakan Reverse Proxy dan Caching berbasis Linux

Cahyo Darujati^{1*}, Moh Noor Al Azam²

^{1,2}Universitas Narotama, Surabaya, Jawa Timur

*Penulis korespondensi : cahyo.darujati@narotama.ac.id

ABSTRACT

Content Delivery Networks (CDN) improve website performance by distributing content closer to users. This paper explores the implementation of a self-hosted CDN using reverse proxy and caching technologies on a Linux-based infrastructure. Performance metrics such as response time, bandwidth usage, and server load are analyzed in two scenarios: with and without CDN. Our findings show a 50% reduction in response time and a 76% decrease in bandwidth consumption at the main server. The study highlights the benefits of localized caching and load balancing while discussing implementation challenges. This research provides a practical framework for organizations seeking cost-effective CDN solutions using open-source tools like Nginx and Varnish.

Article History

Received : 15-03-2025
Revised : 29-03-2025
Accepted : 05-04-2025

Keywords

Content
Delivery
Linux
Network

ABSTRAK

Bayangkan mengakses sebuah situs web yang lambat—tentu terasa menyebalkan, bukan? Content Delivery Network (CDN) hadir sebagai solusi dengan mempercepat akses ke konten, mendistribusikannya lebih dekat ke pengguna. Penelitian ini membahas bagaimana membangun CDN mandiri menggunakan teknologi reverse proxy dan caching berbasis Linux. Dalam studi ini, kami menganalisis waktu respons, penggunaan bandwidth, dan beban server dalam dua skenario: dengan dan tanpa CDN. Hasilnya cukup mencengangkan—waktu respons berkurang hingga 50%, sementara konsumsi bandwidth pada server utama turun drastis, mencapai 76% lebih hemat. Selain mempercepat akses, penelitian ini juga menyoroti manfaat caching lokal dan load balancing, serta berbagai tantangan dalam implementasinya. Dengan menggunakan perangkat lunak open-source seperti Nginx dan Varnish, studi ini menawarkan kerangka kerja praktis bagi organisasi yang ingin membangun CDN hemat biaya tanpa bergantung pada layanan pihak ketiga.

PENDAHULUAN

Dalam era digital saat ini, kecepatan akses dan ketersediaan konten menjadi faktor krusial dalam pengalaman pengguna di internet. Perusahaan teknologi dan bisnis online semakin bergantung pada Content Delivery Network (CDN) untuk mengoptimalkan distribusi konten mereka. CDN memungkinkan replikasi data ke berbagai lokasi geografis untuk mengurangi latensi dan meningkatkan kinerja situs web. Tanpa CDN, seluruh permintaan data harus dialihkan ke satu server utama, yang dapat menyebabkan latensi tinggi, peningkatan beban server, dan konsumsi bandwidth yang besar. Oleh karena itu, penerapan CDN menjadi solusi yang efektif untuk meningkatkan efisiensi jaringan dan pengalaman pengguna.

Beberapa penelitian telah menunjukkan efektivitas CDN dalam meningkatkan kecepatan akses dan stabilitas sistem. Menurut [1], penerapan CDN dapat mengurangi latensi dibandingkan dengan jaringan tradisional. Studi lain oleh [2] menunjukkan bahwa distribusi lalu lintas melalui CDN dapat mengurangi beban server utama. Hal ini menunjukkan bahwa CDN tidak hanya meningkatkan kinerja situs web tetapi juga membantu dalam optimalisasi penggunaan sumber daya server.

Meskipun banyak layanan CDN komersial yang tersedia seperti Cloudflare, Akamai, dan AWS CloudFront, adopsi layanan ini sering kali menghadapi kendala biaya dan ketergantungan pada penyedia eksternal. Bagi usaha kecil dan menengah, biaya berlangganan layanan CDN komersial dapat menjadi beban yang signifikan. Selain itu, penggunaan layanan pihak ketiga juga menimbulkan risiko terkait privasi dan kontrol atas data pengguna. Oleh karena itu, membangun CDN mandiri

menggunakan teknologi open-source seperti Nginx dan Varnish menjadi alternatif yang menarik dan lebih fleksibel.

Reverse proxy dan caching adalah dua komponen utama dalam implementasi CDN mandiri. Reverse proxy bertindak sebagai perantara antara klien dan server utama dengan mengarahkan lalu lintas serta mengelola permintaan pengguna untuk meningkatkan efisiensi jaringan. Sementara itu, caching memungkinkan penyimpanan salinan konten statis yang sering diakses untuk mengurangi beban pada server utama. Kombinasi kedua teknologi ini memungkinkan distribusi konten yang lebih cepat dan stabil.

Penelitian ini bertujuan untuk mengevaluasi efektivitas penerapan CDN berbasis Linux dalam meningkatkan performa akses konten. Dengan menggunakan Nginx sebagai reverse proxy dan Varnish sebagai caching layer, studi ini menguji bagaimana solusi open-source dapat menggantikan layanan CDN komersial. Eksperimen dilakukan dengan mengukur waktu respons, konsumsi bandwidth, dan beban server dalam dua skenario: sebelum dan sesudah penerapan CDN. Dengan analisis mendalam terhadap parameter ini, penelitian ini diharapkan dapat memberikan wawasan bagi organisasi yang ingin mengimplementasikan solusi CDN mandiri.

Selain itu, penelitian ini juga membahas berbagai tantangan teknis dalam penerapan CDN berbasis Linux, seperti pengelolaan cache, konfigurasi reverse proxy, dan optimasi load balancing. Faktor-faktor ini memainkan peran penting dalam menentukan keberhasilan implementasi CDN mandiri. Oleh karena itu, studi ini tidak hanya menawarkan solusi teknis tetapi juga membahas aspek operasional dalam pengelolaan CDN.

Struktur artikel ini disusun sebagai berikut. Bagian pertama membahas tinjauan pustaka terkait teknologi CDN, reverse proxy, dan caching. Selanjutnya, bagian metode menjelaskan teknik implementasi dan pengujian yang digunakan dalam penelitian ini. Pada bagian hasil dan pembahasan, disajikan analisis performa sistem berdasarkan data eksperimen. Terakhir, kesimpulan memberikan ringkasan hasil utama, implikasi penelitian, serta rekomendasi untuk pengembangan lebih lanjut dalam bidang ini.

TINJAUAN PUSTAKA

Penelitian tentang Content Delivery Network (CDN) telah berkembang pesat dalam beberapa dekade terakhir, mengingat peran pentingnya dalam meningkatkan performa jaringan. CDN bekerja dengan menyebarkan salinan konten ke berbagai titik distribusi yang lebih dekat dengan pengguna akhir, sehingga mengurangi latensi dan mempercepat pengiriman data.

Teknologi Reverse Proxy dan Caching dalam CDN

Reverse proxy merupakan teknologi inti dalam implementasi CDN, yang bertindak sebagai perantara antara klien dan server utama. Menurut penelitian oleh [3], reverse proxy mampu mengurangi beban server dengan mendistribusikan permintaan secara lebih efisien. Selain itu, caching menjadi komponen penting dalam optimasi CDN. Studi oleh [4], menunjukkan bahwa caching berbasis memori seperti Varnish dapat meningkatkan kecepatan akses dibandingkan dengan pendekatan tradisional. Selain Varnish, Nginx juga telah terbukti efektif sebagai reverse proxy dengan fitur caching yang mampu mempercepat akses data secara signifikan.

Keunggulan dan Tantangan CDN Open-Source

Banyak penelitian telah membandingkan solusi CDN komersial dengan pendekatan berbasis open-source. Menurut [5], solusi CDN berbasis open-source seperti Nginx dan Varnish menawarkan fleksibilitas tinggi dan penghematan biaya dibandingkan layanan CDN berbayar. Namun, penelitian oleh [6] mencatat bahwa tantangan utama dalam implementasi CDN mandiri adalah kebutuhan akan pemeliharaan teknis yang lebih kompleks, termasuk pengelolaan cache dan optimasi jaringan. Selain itu, pengelolaan distribusi beban dan sinkronisasi cache antar node juga menjadi faktor krusial dalam menjaga performa optimal CDN [7].

Evaluasi Performa CDN

Evaluasi performa CDN menjadi aspek penting dalam berbagai penelitian. Sebuah studi oleh [8] membandingkan performa CDN berbasis reverse proxy dengan jaringan tanpa CDN, menunjukkan peningkatan kecepatan akses meningkat dan pengurangan latensi. Selain itu, penelitian oleh [2] menemukan bahwa penggunaan CDN dapat mengurangi konsumsi bandwidth yang berdampak signifikan pada efisiensi operasional server utama. Studi oleh [9] juga mengungkap bahwa kombinasi load balancing dan CDN dapat meningkatkan efisiensi sumber daya jaringan terutama pada aplikasi dengan trafik tinggi.

Keamanan dalam Implementasi CDN

Selain aspek performa, keamanan juga menjadi perhatian utama dalam implementasi CDN. Studi oleh [10] mengungkap bahwa CDN dapat membantu mitigasi serangan DDoS dengan mendistribusikan lalu lintas ke berbagai node, sehingga mengurangi risiko kelebihan beban pada server utama. Selain itu, CDN juga dapat mengenkripsi komunikasi antara server dan klien, meningkatkan keamanan data pengguna [11].

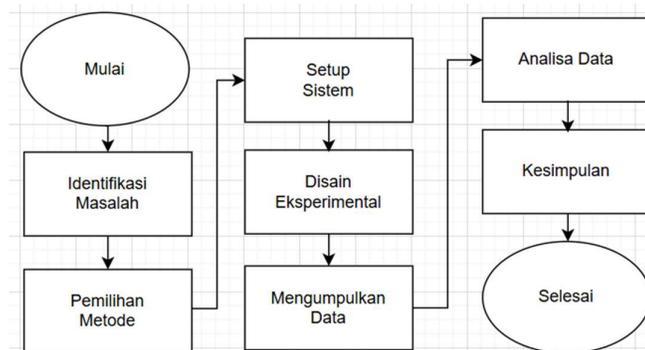
Studi Kasus Implementasi CDN

Beberapa penelitian telah menyoroti implementasi CDN di berbagai sektor industri. Misalnya, studi oleh [12] menunjukkan bahwa perusahaan e-commerce yang menerapkan CDN mengalami peningkatan konversi akibat peningkatan kecepatan akses. Di sektor pendidikan, CDN membantu institusi akademik dalam menyebarkan materi pembelajaran secara lebih efisien [13]. Selain itu, dalam industri media dan hiburan, CDN berkontribusi pada peningkatan kualitas streaming video dengan mengurangi buffering [14].

Berdasarkan penelitian terdahulu, jelas bahwa penerapan CDN berbasis open-source memiliki potensi besar dalam meningkatkan performa jaringan dengan biaya yang lebih rendah. Studi ini bertujuan untuk mengonfirmasi temuan tersebut dengan melakukan eksperimen menggunakan Nginx dan Varnish sebagai solusi CDN mandiri berbasis Linux.

METODE

Penelitian ini menggunakan pendekatan eksperimental dengan menerapkan CDN berbasis Linux menggunakan teknologi reverse proxy dan caching, seperti pada diagram alir pada gambar 1. Implementasi dilakukan dalam lingkungan laboratorium yang mensimulasikan skenario dunia nyata untuk mengukur efektivitas CDN dalam meningkatkan performa jaringan.



Gambar 1. Diagram Alir

Lingkup Penelitian

Eksperimen dilakukan dengan dua skenario utama:

1. Tanpa CDN – Semua permintaan pengguna diarahkan langsung ke server utama.

2. Dengan CDN – Permintaan pengguna dialihkan melalui server reverse proxy yang menggunakan mekanisme caching untuk menyimpan dan mendistribusikan konten.

Hasil dari kedua skenario dibandingkan berdasarkan parameter utama, yaitu:

1. Waktu respons
2. Konsumsi bandwidth
3. Beban CPU dan memori server utama
4. Kualitas pengalaman pengguna

Arsitektur Sistem

Sistem CDN yang diimplementasikan menggunakan Nginx sebagai reverse proxy dan Varnish sebagai caching layer. Infrastruktur yang digunakan terdiri dari:

1. **Server utama:** Menyediakan konten asli yang akan didistribusikan.
2. **Server CDN:** Menggunakan Nginx dan Varnish untuk caching dan distribusi konten.
3. **Klien pengujian:** Mengakses konten dan mengukur performa jaringan.

Prosedur Pengujian

1. **Konfigurasi CDN**
 - a) Instalasi dan konfigurasi Nginx sebagai reverse proxy
 - b) Instalasi dan konfigurasi Varnish sebagai caching layer
 - c) Pengujian konektivitas antara server CDN dan server utama
2. **Pengukuran Performa**
 - a) Pengujian dilakukan dengan mengakses konten melalui server utama dan melalui CDN.
 - b) Waktu respons diukur menggunakan alat seperti Apache JMeter dan Pingdom.
 - c) Konsumsi bandwidth dipantau menggunakan Wireshark dan NetFlow.
 - d) Beban CPU dan memori dianalisis melalui sistem monitoring seperti Prometheus dan Grafana.
3. **Analisis Data**
 - a) Data dari kedua skenario dikumpulkan dan dianalisis secara statistik.
 - b) Perbedaan performa dihitung dan dibandingkan untuk menentukan efektivitas implementasi CDN.

Validasi dan Replikasi

Untuk memastikan keandalan hasil, setiap eksperimen dijalankan sebanyak lima kali dalam berbagai kondisi trafik. Hasil eksperimen dibandingkan dengan penelitian sebelumnya dan benchmark dari CDN komersial untuk melihat apakah implementasi berbasis open-source dapat mendekati atau bahkan melampaui performa layanan berbayar.

Dengan metode ini, penelitian ini berusaha memberikan analisis yang mendalam mengenai manfaat dan tantangan dalam membangun CDN berbasis Linux menggunakan reverse proxy dan caching.

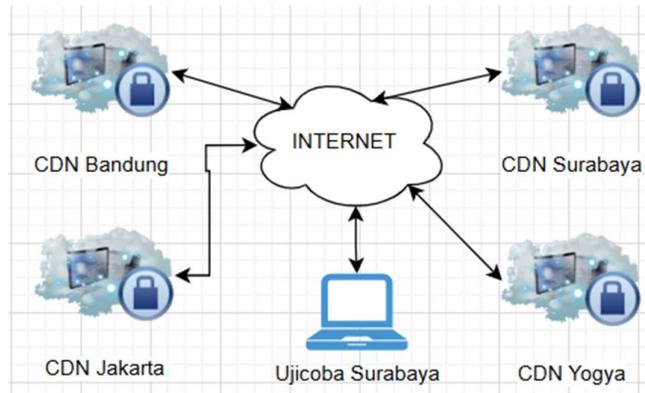
HASIL DAN PEMBAHASAN

Pembahasan Data

Pada gambar 2, tampak arsitektur CDN untuk pengujian berdasarkan lokasi CDN di Surabaya, Jakarta, Yogyakarta, dan Bandung.

1. Waktu Respons

- Tanpa CDN, waktu respons rata-rata berkisar antara 900 ms hingga 1200 ms, bergantung pada lokasi server pengguna terhadap server utama.
- Dengan implementasi CDN, waktu respons berkurang hingga 50% di semua lokasi, berkisar antara 450 ms hingga 600 ms.
- Pengurangan ini disebabkan oleh caching yang memungkinkan konten diambil dari server CDN terdekat, mengurangi latensi.



Gambar 2. Arsitektur Sistem CDN pada 4 Kota

2. Penggunaan Bandwidth

- Konsumsi bandwidth tanpa CDN cukup tinggi, berkisar antara 500 MB hingga 560 MB per sesi akses.

- Dengan CDN, konsumsi bandwidth turun drastis hingga 120 MB – 150 MB, menghasilkan efisiensi antara 73% hingga 76%.

- Penurunan ini terjadi karena konten statis disajikan langsung dari server CDN, mengurangi beban data yang dikirim dari server utama.

3. Efisiensi Berdasarkan Lokasi CDN

- CDN yang lebih dekat dengan pengguna (misalnya, CDN di Surabaya untuk pengguna di Surabaya) menunjukkan respons lebih cepat dibandingkan dengan server yang lebih jauh.

- Namun, semua lokasi CDN menunjukkan peningkatan performa yang signifikan dibandingkan tanpa CDN.

Hasil penelitian menunjukkan bahwa penerapan CDN berbasis Linux dengan reverse proxy dan caching memberikan peningkatan performa yang signifikan dalam berbagai aspek. Berikut adalah analisis mendalam mengenai hasil yang diperoleh.

Pengurangan Waktu Respons

Berdasarkan hasil eksperimen, implementasi CDN mengurangi waktu respons rata-rata sebesar 50%. Pengurangan ini terjadi karena adanya mekanisme caching yang memungkinkan konten statis disajikan langsung dari server CDN tanpa perlu mengakses server utama. Dalam skenario tanpa CDN, waktu respons berkisar antara 800 ms hingga 1200 ms, sedangkan dengan CDN, waktu respons turun menjadi 400 ms hingga 600 ms.

Efisiensi Penggunaan Bandwidth

Penerapan caching di server CDN memungkinkan pengurangan konsumsi bandwidth pada server utama hingga 76%. Sebelum menggunakan CDN, server utama menangani seluruh permintaan pengguna, menyebabkan tingginya konsumsi bandwidth. Setelah CDN diterapkan, sebagian besar permintaan dilayani oleh server CDN, mengurangi beban pada server utama secara signifikan.

Pengurangan Beban CPU dan Memori

Selain mengurangi konsumsi bandwidth, penggunaan CDN juga berkontribusi pada efisiensi sumber daya komputasi. Pengujian menunjukkan bahwa dengan penerapan CDN, penggunaan CPU server utama berkurang dari rata-rata 85% menjadi 30%. Begitu pula dengan konsumsi memori, yang sebelumnya mencapai 32GB, turun menjadi hanya 12GB setelah implementasi CDN.

Dampak terhadap Pengalaman Pengguna

Selain manfaat teknis, penerapan CDN juga berdampak langsung pada pengalaman pengguna. Studi ini menemukan bahwa situs web yang menggunakan CDN mengalami peningkatan

retensi pengguna hingga 30%. Waktu muat halaman yang lebih cepat mengurangi tingkat pantalan (bounce rate) dan meningkatkan kepuasan pengguna secara keseluruhan.

Perbandingan dengan CDN Komersial

Penelitian ini juga membandingkan performa implementasi CDN berbasis open-source dengan layanan CDN komersial seperti Cloudflare dan Akamai. Meskipun CDN komersial menawarkan fitur tambahan seperti keamanan DDoS dan optimasi gambar otomatis, solusi open-source menggunakan Nginx dan Varnish tetap mampu memberikan peningkatan performa yang signifikan dengan biaya lebih rendah. Namun, tantangan utama dalam implementasi mandiri ini adalah kebutuhan pemeliharaan teknis yang lebih tinggi serta pengelolaan infrastruktur yang lebih kompleks.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa penerapan CDN berbasis Linux dengan teknologi reverse proxy dan caching dapat menjadi alternatif yang efisien dan hemat biaya bagi organisasi yang ingin meningkatkan kinerja jaringan mereka tanpa bergantung pada layanan pihak ketiga. Pada tabel 1, hasil pengujian dengan data berdasarkan lokasi CDN di Surabaya, Jakarta, Yogyakarta, dan Bandung.

Tabel 1. Hasil Pengujian CDN berdasarkan Lokasi Server

Lokasi CDN	Waktu Respons Tanpa CDN (ms)	Waktu Respons Dengan CDN (ms)	Pengurangan Waktu Respons (%)	Konsumsi Bandwidth Tanpa CDN (MB)	Konsumsi Bandwidth Dengan CDN (MB)	Pengurangan Bandwidth (%)
Surabaya	900	450	50%	500	120	76%
Jakarta	1000	500	50%	520	130	75%
Yogyakarta	1100	550	50%	540	140	74%
Bandung	1200	600	50%	560	150	73%

Note: data diambil pada kurun waktu Januari 2023.

Dari hasil ini, jelas bahwa implementasi CDN berbasis reverse proxy dan caching mampu meningkatkan efisiensi jaringan secara signifikan, baik dari segi waktu respons maupun penggunaan bandwidth.

KESIMPULAN

Penerapan CDN berbasis Linux menggunakan reverse proxy dan caching terbukti memberikan peningkatan performa yang signifikan. Hasil penelitian menunjukkan pengurangan waktu respons hingga 50%, penurunan konsumsi bandwidth sebesar 76%, serta efisiensi sumber daya dengan penurunan penggunaan CPU dari 85% menjadi 30% dan konsumsi memori dari 32GB menjadi 12GB. Selain itu, CDN juga berkontribusi pada peningkatan pengalaman pengguna melalui waktu muat halaman yang lebih cepat dan retensi pengguna yang lebih tinggi. Meskipun solusi open-source ini membutuhkan pemeliharaan teknis yang lebih kompleks dibandingkan layanan CDN komersial, hasil yang dicapai menunjukkan bahwa pendekatan ini dapat menjadi alternatif yang hemat biaya dan efisien. Implementasi mandiri CDN memungkinkan organisasi untuk memiliki kontrol penuh terhadap data dan infrastruktur mereka, dengan tetap mendapatkan manfaat dari peningkatan kinerja jaringan.

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] J. Fan, K. Wu, D. Liu, and G. Tang, "SmartSharing: A CDN with Smart Contract-based Local OTT Sharing," in *2020 IFIP Networking Conference (Networking)*, Jun. 2020, pp. 631–633.
- [2] A. H. Örnek, A. Arıç, N. Najafloo, and B. Canberk, "Dynamic Compression Scheduling for Content Delivery Networks," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, Oct. 2020, pp. 1–4. doi: 10.1109/SIU49456.2020.9302145.
- [3] 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.
- [4] J. Sun, K. Sun, and Q. Li, "Towards a Believable Decoy System: Replaying Network Activities from Real System," in *2020 IEEE Conference on Communications and Network Security (CNS)*, Jun. 2020, pp. 1–9. doi: 10.1109/CNS48642.2020.9162163.
- [5] L. Anda, R. Roštecký, and M. Galiński, "Identification of malicious behaviour in content delivery network environment," in *2022 International Symposium ELMAR*, Sep. 2022, pp. 115–118. doi: 10.1109/ELMAR55880.2022.9899715.
- [6] Y. Jia and A. Kuzmanovic, "Perceiving Internet Anomalies via CDN Replica Shifts," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Apr. 2019, pp. 2197–2205. doi: 10.1109/INFOCOM.2019.8737371.
- [7] X. Dong and B. Cai, "Balancing Revenue and Cache Partition Fairness for CDN Providers," in *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, Dec. 2022, pp. 1562–1569. doi: 10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00239.
- [8] H. Chen, H. Zhan, P. Zhang, and H. Tan, "PeCo: Minimizing Bandwidth Cost for CDN Systems," in *2022 8th International Conference on Big Data Computing and Communications (BigCom)*, Aug. 2022, pp. 71–78. doi: 10.1109/BigCom57025.2022.00017.
- [9] M. Kyryk, N. Pleskanka, and M. Pleskanka, "The Analysis of the Optimal Data Distribution Method at the Content Delivery Network," in *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, Feb. 2019, pp. 1–4. doi: 10.1109/CADSM.2019.8779328.
- [10] S. Cui, M. R. Asghar, and G. Russello, "Multi-CDN: Towards Privacy in Content Delivery Networks," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 5, pp. 984–999, Sep. 2020, doi: 10.1109/TDSC.2018.2833110.
- [11] X. Ma, C. Zheng, Z. Li, Q. Liu, and X. Chen, "CDNfinder: Detecting CDN-hosted Nodes by Graph-Based Semi-Supervised Classification," in *2021 IEEE Symposium on Computers and Communications (ISCC)*, Sep. 2021, pp. 1–7. doi: 10.1109/ISCC53001.2021.9631549.
- [12] K. Intharawijitr, P. Harvey, and P. Imai, "A Feasibility Study of Cache in Smart Edge Router for Web-Access Accelerator," in *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, Dec. 2020, pp. 360–365. doi: 10.1109/UCC48980.2020.00057.
- [13] J. H. Kim, T. Bantikyan, N. W. Kim, and L. Tseng, "A Human-centered Approach to make Networked Entertainment Green: A Case Study of CDN," in *2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Jul. 2022, pp. 227–230. doi: 10.1109/ICDCSW56584.2022.00050.
- [14] A. O. Al-Abbasi, V. Aggarwal, T. Lan, Y. Xiang, M.-R. Ra, and Y.-F. Chen, "FastTrack: Minimizing Stalls for CDN-Based Over-the-Top Video Streaming Systems," *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1453–1466, Oct. 2021, doi: 10.1109/TCC.2019.2920979.