Available online ***www.jsaer.com***

***Journal of Scientific and Engineering Research*, 2022, 9(3):306-314**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

AWS Global Accelerator vs Akamai: Choosing the Right Global Traffic Solution

Mohit Thodupunuri

MS in Computer Science,
Sr Software Developer - Charter Communications Inc.
Email id: Mohit.thodupunuri@gmail.com

Abstract: Delivering fast, reliable, and secure experiences to global users remains a critical challenge for modern applications. As businesses expand their digital footprints, managing traffic across geographically dispersed regions demands robust solutions to mitigate latency, ensure uptime, and defend against threats. AWS Global Accelerator and Akamai are leading platforms addressing these needs, yet their architectures and operational frameworks differ significantly. AWS Global Accelerator, tightly integrated with Amazon Web Services (AWS), uses anycast routing and AWS's global backbone to optimize traffic for cloud-native applications. Akamai, a pioneer in content delivery networks (CDNs), combines edge server distribution with advanced security features to accelerate content and APIs across multi-cloud environments. This article dissects both solutions, evaluating their technical capabilities, limitations, and suitability for specific use cases. By analyzing their histories, operational models, and real-world implementations, we provide actionable insights for organizations navigating cloud performance optimization.

Keywords: Global traffic management, AWS Global Accelerator, Akamai, cloud performance, multi-cloud, latency optimization, content delivery networks.

1. Introduction

The exponential growth of cloud computing and globalized user bases has reshaped how enterprises architect their digital infrastructure. Applications must now serve users across continents with minimal latency while maintaining security and resilience against outages or attacks. Traditional approaches, such as single-region hosting or basic load balancing, struggle to meet these demands, prompting the rise of advanced traffic management solutions.

AWS Global Accelerator, launched in 2018, is a network-layer service designed to improve availability and performance for applications running on AWS. Using static anycast IP addresses and AWS's global network infrastructure, it routes user traffic to the optimal AWS endpoint—such as Application Load Balancers or EC2 instances—based on real-time health checks and proximity. This AWS-native approach simplifies routing logic but is inherently tied to the AWS ecosystem. [1]

In contrast, Akamai traces its origins to 1998 as one of the first CDN providers, initially focusing on caching static content at edge locations. Over decades, Akamai has evolved into a multi-faceted platform offering dynamic content acceleration, API security, and DDoS mitigation through its Intelligent Edge network. Unlike AWS Global Accelerator, Akamai operates independently of cloud providers, making it a flexible choice for hybrid or multi-cloud environments. [2]

Both solutions address core challenges: reducing latency through intelligent routing, scaling during traffic surges, and integrating security without compromising performance. However, their divergent architectures—AWS's cloud-centric model versus Akamai's edge-first strategy—create distinct trade-offs in cost, complexity, and vendor lock-in. Understanding these differences is critical for architects and engineers tasked with merging technical capabilities to organizational needs.

2. Problem Statement

Selecting an optimal global traffic management solution requires balancing technical, operational, and financial constraints. Below, we look at four critical challenges influencing this decision.

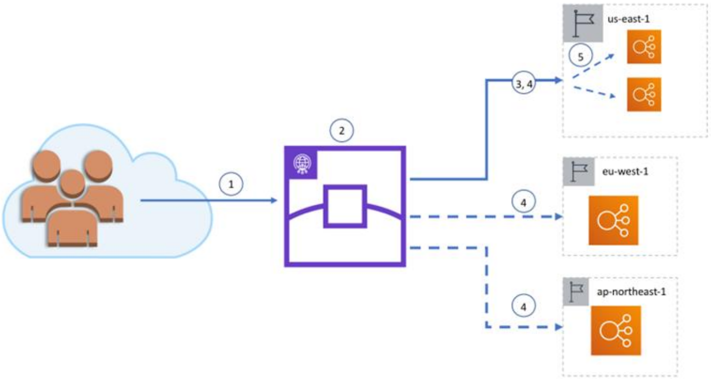

Figure 1: Typical Flow of AWS Global Accelerator. Source: AWS, 2018.

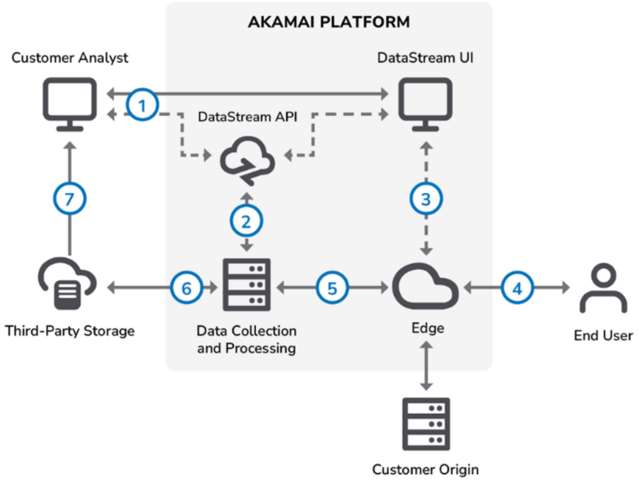

Figure 2: Typical Flow of Akamai transfer. Source: Akamai Tech Docs [3]

Selecting an optimal global traffic management solution requires addressing complex technical challenges that impact performance, scalability, security, and operational agility.

Latency and Performance Variability Across Geographically Dispersed Users

Latency arises from the physical distance between users and application servers, compounded by network congestion, suboptimal routing paths, and protocol inefficiencies. For example, a user in Tokyo accessing a London-based server may traverse multiple autonomous systems (ASes), each introducing potential bottlenecks.

Transmission Control Protocol (TCP) handshake delays and packet loss further degrade performance, particularly for real-time applications like video conferencing or online gaming. Traditional Domain Name System (DNS)-based routing often fails to account for real-time network conditions, directing users to the nearest server based on static mappings rather than live metrics like latency or jitter. [4]

This problem intensifies when applications rely on monolithic architectures hosted in a single region. Even with content delivery networks (CDNs) caching static assets, dynamic content—such as user-specific data or API responses—still routes to centralized origins, creating a "backhaul" effect. [4]

Additionally, Border Gateway Protocol (BGP) routing decisions, which prioritize path length over latency, can inadvertently route traffic through congested nodes.

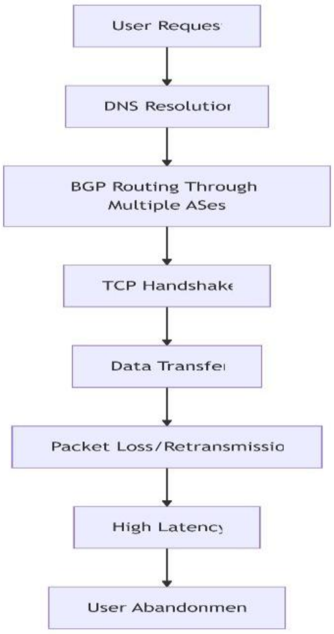

Figure 3: Latency Flowchart

Inefficient Scalability During Unpredictable Traffic Surges

Applications face sudden demand spikes during events like product launches, marketing campaigns, or breaking news. Vertical scaling (adding resources to existing servers) introduces downtime during provisioning, while horizontal scaling (adding servers) requires precise load balancing to avoid overloading new instances.

Legacy systems using static thresholds for auto-scaling often react too slowly, causing cascading failures. For instance, an e-commerce platform during Black Friday may experience request rates exceeding its scaling policies' upper limits, overwhelming databases and triggering throttling.

Stateful applications, such as multiplayer gaming or financial trading platforms, face additional challenges. Replicating session data across regions introduces synchronization overhead, while sticky sessions—binding users to specific servers—can create imbalanced loads. Serverless architectures mitigate some issues but struggle with cold-start delays during abrupt traffic increases. [5]

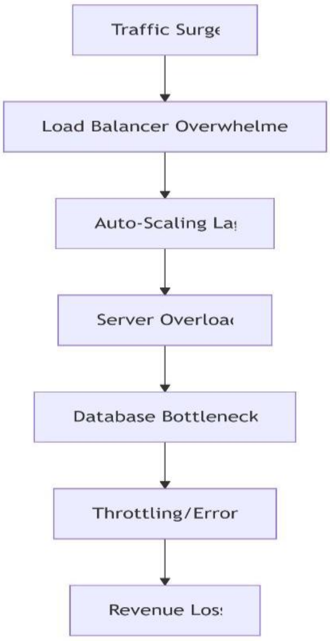

Figure 4: Scalability Flowchart

Security Vulnerabilities in Distributed Traffic Routing

Distributed denial-of-service (DDoS) attacks exploit the openness of global traffic systems. Volumetric attacks flood networks with junk traffic, while application-layer attacks target APIs or web servers with malicious requests. Solutions lacking integrated security force teams to deploy separate DDoS mitigation tools, which may conflict with traffic management policies or add latency. For example, a third-party web application firewall (WAF) inspecting every request can introduce milliseconds of delay per transaction, eroding performance gains from accelerated routing. [7]

Encryption overhead exacerbates these issues. Transport Layer Security (TLS) termination at edge nodes reduces origin server load but requires careful certificate management. Meanwhile, sophisticated attackers use DNS spoofing or BGP hijacking to reroute traffic through malicious nodes, compromising data integrity.

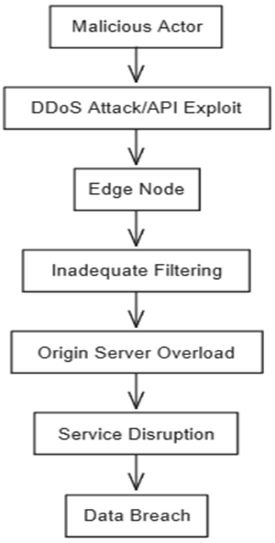

Figure 5: Security Flowchart

Fragmented Traffic Management in Multi-Cloud and Hybrid Environments

Enterprises using multiple cloud providers (e.g., AWS, Azure, Google Cloud) or hybrid infrastructures face disjointed traffic control. Each cloud's native tools—such as AWS Global Accelerator or Azure Front Door—operate within their ecosystems, complicating cross-platform policy enforcement. [7]

For instance, routing rules defined in AWS may not apply to Azure-hosted microservices, leading to inconsistent latency or security policies.

Data transfer costs add another layer of complexity. Inter-cloud traffic (e.g., AWS to Azure) incurs higher fees than intra-cloud traffic, discouraging optimal routing. Additionally, monitoring tools often lack cross-platform visibility, forcing teams to juggle multiple dashboards to diagnose issues like packet loss or misconfigured routes.[8]

Legacy systems using IPv4 face address exhaustion, necessitating complex network address translation (NAT) layers that introduce latency. Meanwhile, IPv6 adoption remains inconsistent across regions, forcing dual-stack configurations that complicate routing logic.

3. Aws Global Accelerator: Addressing Global Traffic Challenges Through Aws-Native Optimization

AWS Global Accelerator (AGA) is a networking service designed to improve the availability and performance of applications with the help of AWS's global infrastructure. Routing traffic through Amazon's redundant network backbone and intelligently directing users to optimal endpoints, AGA mitigates many challenges outlined in the problem statement, including latency, scalability limitations, and security gaps. [1]

History

AWS Global Accelerator launched in November 2018 as part of Amazon's broader strategy to enhance its cloud ecosystem's performance and reliability. The service came alongside AWS's rapid expansion of global regions and edge locations, which grew from 16 regions in 2016 to 31 by 2022. Initially, AGA focused on providing static anycast IP addresses and traffic management for applications hosted on AWS. Over time, AWS integrated it with core services like Elastic Load Balancing (ELB), Amazon EC2, and Elastic IPs, while adding features such as continuous health checks, traffic dials, and client affinity.

The development of AGA reflects AWS's recognition of enterprises' need for predictable performance in hybrid and multi-region architectures. Unlike traditional CDNs, which primarily cache static content, AGA operates at the network layer (Layer 3/4), making it agnostic to application protocols and suitable for dynamic workloads.

Usage and Operational Framework

AWS Global Accelerator employs a two-step routing mechanism to optimize traffic flow:[1][9]

1. Anycast IP Addresses: AGA assigns two static anycast IP addresses to each accelerator. These IPs are advertised from all AWS edge locations, ensuring that user requests automatically route to the nearest edge node. For example, a user in Sydney connects to the edge location in Australia, while a user in Frankfurt connects to the EU edge. This minimizes the distance data travels over the public internet, reducing latency by up to 60% compared to direct internet routing.

2. Endpoint Health-Based Routing: Once traffic enters AWS's network, AGA evaluates the health of registered endpoints (e.g., ALBs, EC2 instances, or Network Load Balancers) across regions. Using real-time health checks, it directs traffic only to healthy endpoints. If an endpoint fails, AGA reroutes traffic within seconds to the next-best option, ensuring high availability.

Operationally, AGA integrates with AWS services through three core components:

- **Listeners:** Define ports and protocols (TCP/UDP) for incoming traffic.
- **Endpoint Groups:** Group endpoints by region (e.g., US-East, EU-West) and configure traffic weights.
- **Traffic Dials:** Adjust the percentage of traffic sent to specific regions during maintenance or outages.

For security, AGA works with AWS Shield Advanced to mitigate DDoS attacks at the edge. It also supports AWS Web Application Firewall (WAF) to filter malicious HTTP/S requests before they reach origin servers.

Limitations

Despite its strengths, AWS Global Accelerator has notable constraints [1]:

- **AWS Ecosystem Dependency:** AGA only routes traffic to AWS resources. Applications relying on non-AWS infrastructure (e.g., on-premises servers or Azure VMs) cannot use AGA without complex workarounds like VPNs or Direct Connect.
- **Limited Protocol Support:** While AGA supports TCP and UDP, it lacks native optimization for newer protocols like HTTP/3 or QUIC, which are critical for modern real-time applications.
- **Cost Implications:** Data transfer through AGA incurs additional fees beyond standard AWS data transfer costs. For example, routing 100 TB of data through AGA can cost ~$3,000/month, excluding regional data fees.
- **Static IP Limitations:** While static IPs simplify DNS management, they can complicate migrations if applications later move outside AWS.

Implementation Example

Consider a global fintech platform hosting transactional APIs on AWS in three regions:

1. US-East (Virginia),
2. EU-West (Ireland), and
3. AP-Southeast (Singapore).

The platform struggles with latency for Asian users and reliability during regional outages.

Step 1: Accelerator Configuration

The team creates an AGA accelerator with two anycast IPs and configures a TCP listener on port 443.

Step 2: Endpoint Group Setup

Three endpoint groups are defined, each pointing to a Network Load Balancer (NLB) in their respective regions. Traffic dials allocate 40% to US-East, 30% to EU-West, and 30% to AP-Southeast.

Step 3: Health Checks and Security

AGA monitors NLBs every 30 seconds. If AP-Southeast fails due to an outage, traffic automatically shifts to US-East and EU-West. AWS Shield Advanced and WAF rules block SQL injection attempts and volumetric DDoS attacks.

Results

- Latency for APAC users drops from 220 ms to 95 ms due to optimized routing through Singapore's edge.
- During a simulated US-East outage, 100% of traffic reroutes within 45 seconds, maintaining 99.99% uptime.
- API response times stabilize at <200 ms globally, meeting service-level agreements (SLAs).

4. Akamai: Mitigating Global Traffic Challenges Via Edge-Native Intelligence and Multi-Cloud Flexibility

Akamai's Intelligent Edge Platform addresses global traffic management challenges by combining one of the world's largest distributed edge networks with advanced security and protocol optimizations. Unlike AWS Global Accelerator, Akamai operates independently of cloud providers, making it ideal for hybrid or multi-cloud architectures.

History

Akamai Technologies, founded in 1998 by MIT researchers Daniel Lewin and Tom Leighton, originated from academic work on algorithms to optimize internet traffic. The company pioneered the content delivery network (CDN) industry by solving the "flash crowd" problem—server overloads caused by sudden traffic spikes—using distributed edge caching. Akamai's early focus on static content delivery for media and e-commerce clients, such as Apple and Yahoo!, established its reputation.

Over two decades, Akamai expanded beyond CDN services. In 2012, it acquired Prolexic, integrating DDoS mitigation into its platform. The 2015 launch of Cloud Security Solutions marked its shift toward holistic traffic management, combining performance and security. Today, Akamai's edge network spans over 350,000 servers in 135 countries, handling 30% of global web traffic. Its evolution reflects a transition from caching static files to accelerating dynamic content, APIs, and real-time applications. [2]

Usage and Operational Framework

Akamai's Intelligent Edge Platform operates at both the network (Layer 3/4) and application (Layer 7) layers, using a three-tiered approach [2] [10]:

1. Edge Server Distribution: Akamai's edge servers cache static content (e.g., images, CSS) and dynamically optimize requests for APIs or databases. For example, a user in São Paulo accessing a New York-hosted application retrieves static assets from Akamai's São Paulo edge node, reducing round-trip time. Dynamic content routes through Akamai's backbone via proprietary protocols like Fast DNS and Secure CDN.

2. Adaptive Traffic Routing: Akamai's Global Traffic Management (GTM) uses real-time data from its Edge Network Map—a constantly updated database of network conditions—to route users to the optimal origin or edge server. Unlike AWS Global Accelerator's anycast IPs, Akamai employs DNS-based and anycast routing. For instance, GTM might direct a Paris user to a Frankfurt origin during peak EU congestion but reroute to Milan if Frankfurt experiences packet loss.

3. Integrated Security Layers: All traffic passes through Akamai's security stack, including:

a. Kona Site Defender: Blocks application-layer DDoS attacks and SQL injections.

b. Prolexic: Mitigates network-layer DDoS attacks up to 20 Tbps.

c. API Security: Validates API requests using JSON Schema and rate limiting.

Operational workflows integrate with DevOps pipelines via Akamai's Control Center and CLI tools. For example, teams deploy configurations using Terraform to enforce consistent routing rules across edge nodes.

Limitations

Akamai's strengths come with trade-offs [2][7]:

- **Cost Complexity:** Pricing models vary by service (e.g., data transfer, security requests), making cost forecasting challenging. Accelerating dynamic APIs costs ~$0.01–0.03 per request, which scales expensively for high-throughput applications.
- **Configuration Overhead:** Fine-tuning edge logic (e.g., caching rules, API validation) requires expertise in Akamai's Property Manager and EdgeWorkers scripting. Misconfigurations can inadvertently block legitimate traffic.
- **DNS Dependency:** DNS-based routing introduces latency during TTL (time-to-live) expiration. For example, shifting traffic from a failed origin may take minutes if clients cache outdated DNS records.

- **Limited Cloud-Native Integrations:** While Akamai supports AWS and Azure, its APIs lack deep hooks into cloud-specific services like AWS Lambda@Edge.

Implementation Example

A global media streaming platform uses Akamai to deliver live sports events to 10 million concurrent viewers. The platform previously faced buffering during peak traffic and suffered a 3-hour outage from a DDoS attack.

Step 1: Edge Caching and Dynamic Routing

Akamai's Ion solution caches video segments (e.g., HLS/DASH) at edge nodes in 50+ regions. For dynamic content like user authentication or live chat, Akamai's API Accelerator compresses JSON payloads and reuses TLS sessions, reducing handshake overhead.

Step 2: Security Integration

Kona Site Defender enforces rate limits of 1,000 requests/second per IP, blocking credential-stuffing attacks. Prolexic scrubs volumetric DDoS traffic at the edge, filtering 500 Gbps of junk data during a live event.

Step 3: Traffic Monitoring

Akamai's mPulse Real User Monitoring (RUM) tracks latency, buffering rates, and errors. Alerts trigger auto-scaling for origin servers via webhooks to Kubernetes clusters.

Results

- Latency for APAC users drops from 4.2 seconds to 800 ms, achieving sub-second video start times.
- During a Champions League final, the platform scales to handle 15 million viewers without throttling.
- A 1.2 Tbps DDoS attack during a live stream is mitigated within 90 seconds, with zero downtime.

5. Direct Comparison

Table 1: Comparing AWS Global Accelerator with Akamai

Criteria	AWS Global Accelerator	Akamai
Architecture	Relies on AWS's global backbone and anycast IPs; operates at Layer 3/4.	Uses a distributed edge network (350k+ servers) with Layer 3–7 optimizations.
Network Reach	31 AWS regions and 400+ Points of Presence (PoPs).	135 countries, 4,100+ PoPs, including non-AWS regions.
Protocol Support	TCP/UDP only; no native HTTP/3 or QUIC.	Full HTTP/3, QUIC, WebSocket, and legacy protocol support.
Security Integration	AWS Shield Advanced (DDoS) and WAF; requires separate configuration.	Built-in Kona (WAF), Prolexic (DDoS), and API Security; no third-party tools needed.
Multi-Cloud Support	Limited to AWS resources; requires workarounds for non-AWS infrastructure.	Native support for AWS, Azure, Google Cloud, and on-premises via edge scripting.
Traffic Routing	Anycast IPs with real-time health checks; sub-100ms rerouting during outages.	Hybrid DNS + anycast; rerouting depends on DNS TTL (1–5 minutes).
Cost Model	Data transfer fees ($0.01–0.03/GB) + hourly accelerator costs.	Tiered pricing based on traffic volume, security features, and edge compute usage.
Ideal Use Cases	AWS-native apps requiring low-latency failover (e.g., gaming, financial trading).	Multi-cloud apps with dynamic/static content (e.g., media streaming, global e-commerce).
Performance Optimization	Reduces latency by 30–60% for AWS workloads.	Cuts latency by 50–70% for global users via edge caching and protocol optimizations.
Ease of Use	Simplified setup via AWS Console; integrates natively with ELB, EC2.	Steeper learning curve; requires expertise in Akamai's Property Manager and EdgeWorkers.

6. Analysis/Recommendations

For AWS-Native Organizations

AWS Global Accelerator is the logical choice for enterprises fully invested in AWS. Its seamless integration with services like EC2, Lambda, and Elastic Load Balancing simplifies deployment, particularly for stateful applications requiring rapid failover.

For example, a stock trading platform using AWS Global Accelerator can maintain sub-200ms latency globally while rerouting traffic during regional outages in seconds. However, avoid AGA if your roadmap includes multi-cloud adoption, as its dependency on AWS creates vendor lock-in.

For Multi-Cloud or Hybrid Environments

Akamai excels in multi-cloud architectures due to its cloud-agnostic edge network. A retailer using Azure for inventory APIs and AWS for customer analytics can use Akamai's GTM to unify traffic policies and reduce cross-cloud latency. Its integrated security stack also eliminates the need for third-party DDoS tools, which is critical for industries like healthcare or finance. However, prepare for higher administrative overhead, as Akamai's advanced features require specialized expertise.

Latency-Sensitive Applications

Both solutions reduce latency, but their approaches differ. AWS Global Accelerator's anycast IPs minimize "time to first byte" (TTFB) for AWS-hosted apps, making it ideal for real-time gaming or VoIP. Akamai's edge caching and HTTP/3 support better serve media streaming or e-commerce platforms with global audiences. For instance, a video streaming service using Akamai reported a 60% reduction in buffering during peak traffic.

Security-Critical Workloads

Akamai's embedded security features provide broader protection out-of-the-box. Prolexic mitigates network-layer DDoS attacks up to 20 Tbps, while Kona blocks OWASP Top 10 threats. AWS Global Accelerator relies on AWS Shield Advanced, which lacks Akamai's granular API security controls. For industries under regulatory scrutiny (e.g., banking), Akamai's compliance certifications (SOC2, PCI-DSS) offer added assurance.

Cost Considerations

AWS Global Accelerator costs scale predictably with data transfer volume, but inter-region fees can add up. Akamai's pricing is opaquer, with charges for data, security requests, and edge computing. For example, a SaaS company spending $10,000 per month on AWS data transfer might pay $12,000 to $15,000 with Akamai for equivalent traffic but gain security savings. Conduct a TCO analysis weighing performance gains against budget constraints.

In essence;

- **Choose AWS Global Accelerator if:** You prioritize AWS integration, need sub-minute failover, or operate cost-sensitive, cloud-native apps.
- **Choose Akamai if:** You require multi-cloud flexibility, advanced security, or serve dynamic content to global users.
- **Hybrid Approach:** Use AGA for AWS workloads and Akamai for edge caching/security in hybrid setups.

7. Conclusion

AWS Global Accelerator and Akamai offer distinct solutions for global traffic management, each excelling in specific contexts. AWS Global Accelerator uses Amazon's backbone for low-latency routing and rapid failover, making it ideal for organizations deeply embedded in the AWS ecosystem. However, its lack of multi-cloud support and protocol limitations hinder flexibility. Akamai counters with a vast edge network, security, and multi-cloud agility, though its complexity and costs demand technical maturity.

Enterprises must work in line with their choice with infrastructure strategy, performance requirements, and security needs. For AWS-centric teams, AGA delivers simplicity and speed. For those prioritizing global reach and threat mitigation, Akamai remains unmatched. As cloud environments evolve, hybrid architectures combining both solutions may be a balanced approach to traffic optimization.

References

[1]. V. K. Manik, "AWS Global Accelerator - Varun Kumar Manik - Medium," Medium, Dec. 14, 2021. [Online]. Available: https://varunmanik1.medium.com/aws-global-accelerator-404c3964650f/

[2]. E. Nygren, R. K. Sitaraman, and J. Sun, "The Akamai network," ACM SIGOPS Operating Systems Review, vol. 44, no. 3, pp. 2–19, Aug. 2010, doi: 10.1145/1842733.1842736.

[3]. D. Fraser, "An introduction to the Akamai Content Delivery Network," Medium, Dec. 07, 2021. [Online]. Available: https://medium.com/free-code-camp/an-introduction-to-the-akamai-content-delivery-network-806aa16d8781

[4]. "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," IEEE Journals & Magazine | IEEE Xplore, Jun. 01, 1997. https://ieeexplore.ieee.org/abstract/document/611099

[5]. "MultiScaler: A Multi-Loop Auto-Scaling Approach for Cloud-Based Applications." https://ieeexplore.ieee.org/abstract/document/9226496/

[6]. Tague, P., Slater, D., Rogers, J., & Poovendran, R. (2008). Evaluating the vulnerability of network traffic using joint security and routing analysis. IEEE transactions on dependable and secure computing, 6(2), 111-123.

[7]. Raj, P., Raman, A., Raj, P., & Raman, A. (2018). Multi-cloud management: Technologies, tools, and techniques. Software-defined cloud centers: Operational and management technologies and tools, 219-240.

[8]. Guo, T., Sharma, U., Shenoy, P., Wood, T., & Sahu, S. (2014). Cost-aware cloud bursting for enterprise applications. ACM Transactions on Internet Technology (TOIT), 13(3), 1-24.

[9]. "Introducing AWS Global Accelerator custom routing accelerators | Amazon Web Services," Amazon Web Services, Jan. 13, 2021. https://aws.amazon.com/blogs/networking-and-content-delivery/introducing-aws-global-accelerator-custom-routing-accelerators/

[10]. Imran, H. A., Latif, U., Ikram, A. A., Ehsan, M., Ikram, A. J., Khan, W. A., & Wazir, S. (2020, November). Multi-cloud: a comprehensive review. In 2020 ieee 23rd international multitopic conference (inmic) (pp. 1-5). IEEE.