e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:07/Issue:02/February-2025 Impact Factor- 8.187 www.irjmets.com

# THE ROLE OF AKAMAI CDN CACHING: PERFORMANCE BOOSTING TECHNIQUES FOR MODERN WEBSITES

**Avinash Ibbandi**[*1]

[*1]Walmart, Inc, USA.

DOI : https://www.doi.org/10.56726/IRJMETS67229

## ABSTRACT

In an era where digital experiences shape consumer behavior, website performance has become a defining factor in online success. Slow loading pages can lead to increased bounce rates, lower search engine rankings, and lost revenue. Akamai, a global leader in content delivery and edge computing, provides advanced caching mechanisms that accelerate page load times, reduce latency, and enhance scalability. By strategically caching content across its extensive network of edge servers, Akamai minimizes the reliance on origin servers, optimizing the user experience. This article explores how Akamai's intelligent caching solutions improve page performance, reduce infrastructure strain, and future proof websites against ever growing traffic demands.

**Keywords:** Akamai Caching, Website Performance, Content Delivery Network (CDN), Edge Caching, Page Optimization, Latency, Dynamic Content Caching, Web Acceleration, High Availability, Caching Strategies, Performance Optimization, Fast Content Delivery, User Experience Enhancement.

## I. INTRODUCTION

The speed at which a webpage loads can make or break a user's online experience. Studies have shown that even a onesecond delay can significantly impact engagement, conversions, and customer retention. As the internet continues to evolve, businesses must prioritize performance optimization to stay competitive. Akamai's caching technology is at the forefront of this transformation, enabling websites to deliver content faster and more efficiently. Unlike traditional caching, which relies solely on storing static assets, Akamai employs adaptive caching strategies that intelligently serve both static and dynamic content from edge locations closest to users. This reduces round trip times, alleviates backend server loads, and ensures seamless performance even during high traffic surges. In this article, we will dive deep into the mechanics of Akamai caching, exploring how it enhances website performance, its impact on scalability, and best practices for leveraging its full potential in a world where speed is no longer a luxury but a necessity.

## II. METHODOLOGY

To analyze the impact of Akamai caching on page performance, this study follows a structured approach:

1. Understanding Caching Mechanisms – A breakdown of Akamai's caching layers, including static and dynamic caching, object prefetching, and intelligent cache hierarchies.
2. Performance Benchmarking – Evaluating page load times, latency, and server response efficiency before and after Akamai caching implementation.
3. Case Studies & Real-World Applications – Reviewing industry use cases where businesses have leveraged Akamai's caching to improve website performance and scalability.
4. Best Practices & Optimization Strategies – Identifying key techniques such as cache control headers, cache keys, purging strategies, and integration with other performance-enhancing technologies.
5. Impact on Scalability & User Experience – Assessing how Akamai's caching solutions handle high-traffic loads while maintaining a seamless user experience.

### Akamai CDN Caching and Features

Akamai CDN operates on a distributed network of edge servers, strategically placed worldwide, to cache and serve content closer to users. Below are its key caching features and capabilities:

### Caching Basics

- Edge Caching: Stores content at edge locations to reduce origin load.

- Tiered Caching: Uses an intermediary cache layer between the edge and the origin to optimize cache hit rates and reduce origin requests.
- Cache Hierarchy: Allows multiple cache layers to improve efficiency.
- Custom Cache Rules: Users can define caching rules in Akamai Property Manager.

**Cache Control & Expiry**

- Time-to-Live (TTL) Settings: Configurable TTL settings allow fine-tuned control over cache expiration.
- Cache-Control Headers: Supports standard HTTP directives (public, private, no-cache, max-age).
- Surrogate-Control Headers: Used for enhanced cache control beyond standard HTTP headers.
- Stale Content Serving: Allows content to be served even if the origin is unreachable or cache has expired.
- Prefetch (Prefetching): Fetches new content before expiry to maintain freshness.

**Advanced Caching Capabilities**

- Edge Side Includes (ESI): Enables dynamic content assembly at the edge to reduce origin processing.
- Dynamic Page Caching: Supports caching of personalized and dynamically generated content.
- Partial Caching: Allows specific sections of a page to be cached while keeping others dynamic.
- Segmented Caching: Optimizes large file delivery by caching content in chunks.
- Content Invalidation & Purging:

○ Fast Purge (Near Instant): Purge outdated content in seconds.
○ Stale-While-Revalidate: Serves stale content while fetching fresh content in the background.
○ Adaptive Purging: Triggers cache purging based on content changes.
○ Cache Tags: Akamai Cache Tags provide a mechanism to label cached objects with specific metadata so they can be purged in a targeted manner. Instead of purging by URL (which can be slow and difficult for dynamic content), you can purge all objects associated with a specific Cache Tag in one go.

**Cache ID Modification**

The Cache ID Modification behavior controls how cache keys are created by allowing query parameters, headers, and cookies to be included along with the hostname and path.

Key Points:

- Overrides Other Rules: When this behavior is set in a rule, it replaces any similar settings from parent or child rules.
- Expands Cache Key Options: Instead of just using the URL path and query parameters, you can include headers or cookies (e.g., caching different language versions based on the Accept-Language header).
- Required Elements remain unchanged: The hostname and path are always part of the cache key and cannot be removed.

**Performance Enhancements**

- Dynamic Site Acceleration (DSA): Reduces round trip times for dynamic content.
- Adaptive Acceleration: Uses machine learning to optimize delivery based on real-time traffic.
- Gzip & Brotli Compression: Reduces payload size for faster transfers.
- Pre-Fetching & Push: Fetches assets before users request them.
- SureRoute (Intelligent Routing): Optimizes requests by selecting the fastest network path.

**Security Features**

- Edge-Based DDoS Protection: Mitigates attacks before they reach the origin.
- Web Application Firewall (WAF): Protects against OWASP Top 10 vulnerabilities.
- Edge Token Authentication: Ensures only authorized users access cached content.
- Secure Socket Layer (SSL/TLS) Support: Encrypts data transmission for secure browsing.
- Bot Management: Identifies and mitigates bad bot traffic.
- Origin Cloaking: Hides the origin server’s IP address to prevent direct attacks.

**API Caching & Acceleration**

- API Gateway Caching: Caches API responses to reduce backend load.
- Rate Limiting: Controls API traffic to prevent abuse.
- GraphQL & REST API Acceleration: Optimizes API performance by caching responses at the edge.

**Customization & Configuration**

- Akamai Property Manager: UI and API-based management for setting caching rules.
- EdgeWorkers (Serverless Functions): Runs JavaScript at the edge for real-time content manipulation.
- Cloudlet Applications: Additional microservices that extend functionality (e.g., Redirects, Audience Segmentation).

**Analytics & Monitoring**

- Real-Time Log Streaming: Sends logs to external SIEMs or monitoring tools.
- Cache Hit Ratio Reports: Provides insights into cache performance.
- Traffic Analyzer: Visualizes request patterns and trends.
- Security Threat Intelligence: Monitors and mitigates emerging threats.

**Leverage Edge Redirects for caching mechanisms to improve performance**.

Akamai provides edge-based redirects using:

1. Cloudlets - Edge Redirector (Recommended)
   - Uses policy-based rules to redirect traffic at the edge.
   - Reduces the need for origin requests.
   - Supports 301 (Permanent) and 302 (Temporary) redirects.
   - Can be cached for efficiency.
2. Property Manager Redirect Rules
   - Redirects based on path, hostname, query parameters, or headers.
   - Configurable directly in the Akamai Property Manager.
   - Can leverage Akamai's cache hierarchy to serve redirects quickly.
3. EdgeWorkers (JavaScript at the Edge)
   - Custom edge logic for dynamic redirects.
   - Can execute logic based on cookies, geo-location, user-agent, etc.

**How Edge Redirects Interact with Caching**

- Cached Redirect Responses: If a 301 (Permanent Redirect) is set, browsers and intermediate caches (including Akamai) may cache the redirect response.
- Bypassing Origin Requests: Since the redirect happens at the edge, origin servers do not get involved, reducing latency and server load.
- TTL for Redirect Caching: Redirects can be cached using TTL (Time-to-Live) settings, just like static content.
- Avoid Cache Conflicts: If not properly configured, caching mechanisms might conflict with redirects (e.g., caching a 301 when a 302 should be used dynamically).

**Key Benefits of Edge-Based Redirects**

- Faster response times (no need to reach the origin).
- Reduced origin load (less unnecessary traffic).
- Improved user experience (redirects happen instantly).
- Supports SEO best practices (correct 301/302 handling).

**Edge Redirects & Caching in Akamai CDN**

Edge redirects are not directly part of caching but can benefit from caching mechanisms to optimize performance. Let's break it down while keeping caching principles in mind.

- Redirect Is Cached Behaviors

When a redirect response (301, 302, etc.) is processed by Akamai's edge servers, it can be cached like any other HTTP response. This means:

- The redirect response (3xx HTTP status) is stored at the edge.
- Subsequent requests for the same resource get redirected immediately from cache.
- The origin does not receive further requests unless cache expires or is purged.

Key takeaway: Cached redirects improve efficiency by reducing origin traffic and speeding up response times.

- Caching Behavior for Different Redirect Types

301 (Permanent Redirect)

- Cache-Friendly: Browsers and CDNs (including Akamai) aggressively cache 301 redirects.
- Long-Term Storage: Unless explicitly overridden, 301 responses are cached indefinitely by some clients.
- SEO Impact: Search engines use 301s to permanently update their records.

Best Practices for Akamai Caching:

- Set high TTL for 301s (e.g., days/weeks) to maximize performance.
- Use Fast Purge if the redirect needs to change.

302 (Temporary Redirect)

- Not Meant for Caching: Since it's temporary, browsers typically do not cache 302s.
- Akamai Can Cache It: But it requires explicit caching rules in Property Manager.
- SEO Consideration: Search engines do not update their records permanently.

Best Practices for Akamai Caching:

- Use low TTL (or no cache) for dynamic redirects.
- Prefer EdgeWorkers if redirects depend on user attributes (cookies, geo, device type).

**Akamai Caching Behavior for Redirects**

When Will Akamai Cache a Redirect?

- If a 301/302 is explicitly cached using Property Manager rules.
- If a Cache-Control: max-age header is present.
- If Akamai's internal caching policies allow it.

When Won't Akamai Cache a Redirect?

- If Cache-Control: no-cache, private, or max-age=0 is set.
- If a 302 redirect is used without explicit caching rules.
- If a redirect involves dynamic user-specific conditions.

Akamai Features That Optimize Redirect Caching

Cloudlets – Edge Redirector (Best for Static Redirects)

- Rule-based redirect handling at the edge.
- Redirects are cached like any other object.
- Reduces origin dependency.

Akamai Property Manager – Redirect Rules

- Supports redirect caching by setting TTL.
- Surrogate-Control & Cache-Control headers control caching behavior.
- Used for path-based and host-based redirects.

EdgeWorkers – Dynamic Redirects

- Executes JavaScript at the edge.
- Does not cache user-specific redirects but prevents unnecessary origin requests.

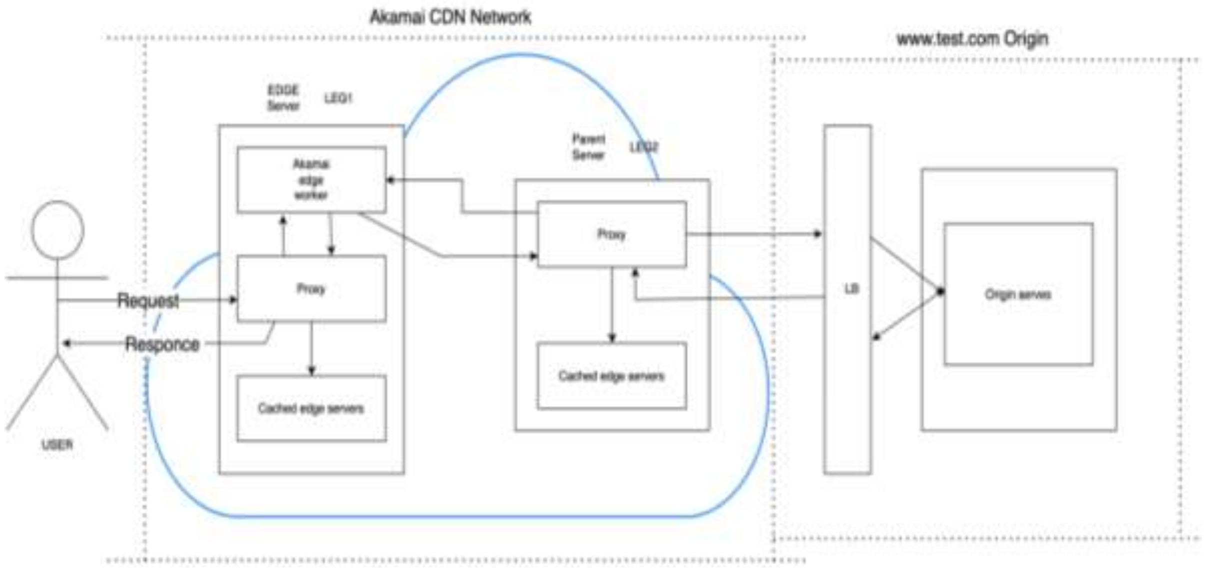
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
( Peer-Reviewed, Open Access, Fully Refereed International Journal )
Volume:07/Issue:02/February-2025 Impact Factor- 8.187 www.irjmets.com

- Ideal for geo-based, device-based, and cookie-based redirects.

Best Practices for Using Caching with Redirects

- Use 301 for permanent changes (cache-friendly, fewer origin hits).
- Use Cloudlets for static redirects (ensures edge-level processing).
- Set appropriate TTLs (longer for 301, shorter for 302).
- Use EdgeWorkers for real-time logic (prevents caching conflicts).
- Always test Fast Purge before deploying major redirect changes.

**Caching Flow diagram**



**Site Performance**

○ The total time taken for a web page to fully load its content once user clicks the website link. Ideally, this should be under 2-3 seconds.

| Metric | Ideal Load Time |
|---|---|
| • Full Page Load | 2seconds is ideal |
| • First Contentful Paint (FCP) | 1-2 seconds |
| • Largest Contentful Paint (LCP) | <2.5 seconds (Google recommendation) |
| • Time to First Byte (TTFB) | <0.8 seconds |
| • Interactivity (Time to Interactive - TTI) | <3.8 seconds |

- Browser Caching

○ Browser caching stores static files like images, CSS, and JavaScript on a user's device, preventing the need for repeated downloads on subsequent visits. This reduces server requests, significantly improving load times for returning visitors.

- Response time

○ Server response time refers to the duration a web server takes to respond to a browser's request. A slow response time can considerably delay webpage loading, negatively affecting user experience and search engine rankings

- Content Delivery Network (CDN)

○ A CDN (Content Delivery Network) is a globally distributed network of servers designed to cache and deliver a website's static content from the server nearest to the user. This minimizes load times and enhances overall performance.

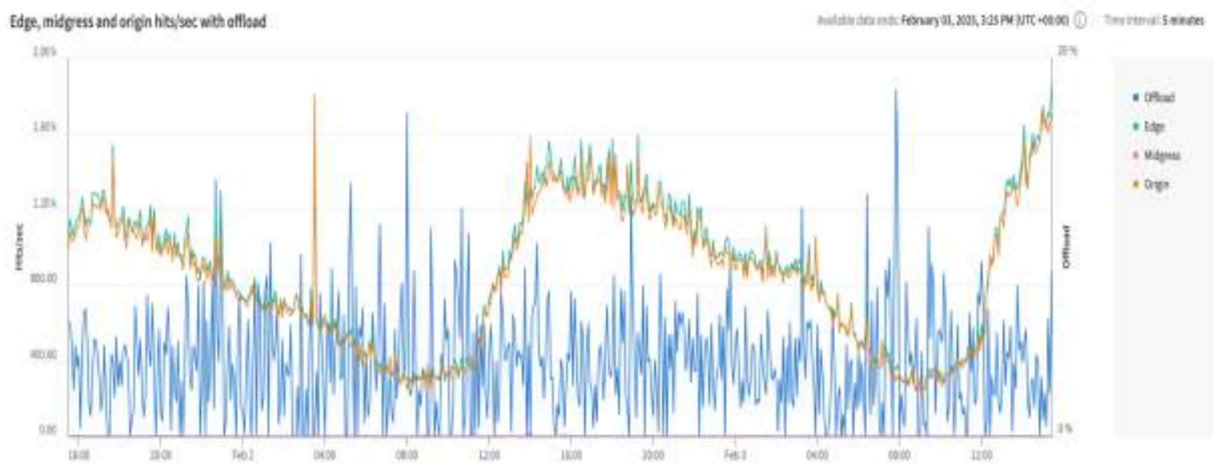
- Lazy Loading
  - Lazy loading is a technique that delays the loading of images and videos until they are needed, specifically when they enter the user's viewport. This prevents unnecessary loading of off-screen content, improving page speed and performance.
- Optimize Images
  - Optimizing images is essential for improving website speed, as they are typically the largest files on a webpage. Well-optimized images load faster, enhancing user experience and lowering bounce rates.
- Minification CSS, JS & HTML
  - Minification involves eliminating unnecessary characters such as spaces, comments, and line breaks from code without affecting its functionality. This process reduces file sizes, enabling browsers to load content more quickly, ultimately enhancing website speed and user experience. When CSS, JavaScript, and HTML are not optimized, larger file sizes can lead to slower loading times and increased bandwidth consumption.

Note: Caching and redirects work together in Akamai's CDN to optimize performance. While redirects themselves are not a form of caching, they benefit from being cached at the edge to reduce latency and avoid unnecessary origin requests.

While edge redirects leverage caching to improve performance, they are not technically part of caching. Instead, they are an edge processing feature that improves efficiency by reducing unnecessary origin traffic.

**Offload Report from EDGE to ORIGIN**

Offload reports show certain % of requests (based on configurations) we can store in akamai and serve faster to the customers from edge this increase SEO and member interaction with the site.



## III. KEY TAKEAWAYS

Akamai caching significantly reduces latency and improves website load times by serving content from edge servers closer to users.

Dynamic caching and adaptive acceleration help deliver both static and dynamic content efficiently, reducing the burden on origin servers.

Proper cache management strategies, including cache keys, purge APIs, and TTL configurations, maximize caching efficiency and prevent stale content issues.

Akamai's intelligent caching solutions enhance scalability and high availability, ensuring performance remains stable during peak traffic periods.

Implementing Akamai caching improves SEO rankings and conversion rates, as faster load times contribute to a better user experience.

## IV. CONCLUSION

In the modern digital landscape, speed is the cornerstone of user engagement and business success. Akamai's caching solutions offer a powerful, scalable, and intelligent way to enhance website performance, reduce latency, and optimize resource utilization. By leveraging edge caching, dynamic acceleration, and advanced cache management techniques, businesses can ensure that users receive content faster while minimizing infrastructure costs. As web traffic continues to grow and user expectations for instant access increase, implementing Akamai's caching technology is no longer an option—it's a necessity. By adopting best practices and continuously optimizing caching configurations, organizations can stay ahead in delivering fast, seamless, and reliable digital experiences in an ever-evolving online world.

## V. REFERENCES

[1] Akamai Technologies. (2024). How Akamai Caching Works.

- Official documentation explaining Akamai's caching strategies and their impact on web performance.
- Retrieved from: https://www.akamai.com

[2] Google Developers. (2024). Web Performance Optimization Guide.

- Discusses best practices for improving website speed, including caching techniques.
- Retrieved from: https://developers.google.com/speed

[3] Pingdom. (2024). Website Speed Testing and Optimization Strategies.

- Analyzes various factors affecting load times, including CDN and caching techniques.
- Retrieved from: https://www.pingdom.com

[4] Nielsen Norman Group. (2024). The Impact of Page Speed on User Behavior and Conversion Rates.

- Discusses how slow websites affect user engagement and how caching helps.
- Retrieved from: https://www.nngroup.com

[5] Cloudflare. (2024). Understanding Content Delivery Networks (CDNs).

- Explains how CDNs work and why they are essential for modern web performance.
- Retrieved from: https://www.cloudflare.com/learning/cdn/what-is-a-cdn