International Journal of Scientific Research and Modern Technology (IJSRMT) *ijsrmt.com*
Volume 5, Issue 1, 2026
DOI: https://doi.org/10.38124/ijsrmt.v5i1.1235

HierarchicalCDN: Federated Edge Intelligence with Metadata-Driven Cache Optimization for Live Streaming

[1]Muhamed Ramees Cheriya Mukkolakkal

Publication Date 2026/01/10

Abstract

We present HierarchicalCDN, a novel CDN optimization system combining hierarchical LLM coordination with federated edge learning. The system employs three-tier orchestration (global, regional, edge) where edge LLMs operate autonomously while continuously learning from peer deployments across 50,000+ locations. Content-type specialization enables episode-aware prefetching for series (94% hit rate), event-window optimization for live sports (96% hit rate), and trailer-driven prediction for movies (88% hit rate). Federated learning enables edge nodes to share learned patterns without centralized retraining, achieving 23% improvement over isolated learning. Evaluation on production workloads serving 850 PoPs and 2.5 PB/day demonstrates 43% cache miss reduction, 37% latency improvement, 52% bandwidth savings, and 29% storage efficiency gains compared to traditional LRU caching.

I. INTRODUCTION

Modern content delivery networks process vast amounts of metadata—technical specifications, content attributes, user access patterns, and infrastructure state—that traditional caching algorithms ignore. Different content types exhibit distinct consumption patterns: TV series show 85% episode continuation rates with strong sequential viewing, movies demonstrate 85% single-view probability, and live sports exhibit 90% of views concentrated in 3-hour event windows. These patterns, combined with rich metadata about content relationships, user behavior, and network topology, present opportunities for intelligent optimization that rule-based systems cannot exploit.

We present HierarchicalCDN, which combines three innovations: (1) hierarchical LLM deployment across global, regional, and edge tiers for metadata-driven decision making, (2) federated learning at the edge tier enabling distributed knowledge sharing while maintaining autonomous operation, and (3) content-type specialization that adapts caching strategies to dominant regional consumption patterns. The three-tier architecture balances sophisticated reasoning at higher tiers (15-minute global cycles, 5-minute regional cycles) with real-time edge decisions (sub-100ms latency) using smaller, specialized models.

Our key innovation is federated edge learning: edge LLMs operate independently for real-time caching decisions while periodically sharing learned patterns with peer deployments. Unlike centralized approaches requiring global model retraining, our federated approach enables continuous learning from distributed deployments. Each edge node learns from local traffic patterns and periodically contributes model updates to a federation protocol that aggregates improvements across thousands of locations. This enables Seattle's edge nodes to benefit from viewing patterns discovered in similar markets while adapting to local preferences.

We evaluate HierarchicalCDN on production workloads serving 50,000+ edge locations across 850 points of presence (PoPs) delivering 2.5 PB/day. Results demonstrate 43% cache miss reduction (21.5% vs 37.7% baseline), 37% latency improvement (487ms vs 892ms P95), 52% bandwidth savings (453 vs 943 TB/day origin traffic), and 29% storage efficiency improvement (86.9% vs 67.4% utilization). Content-type specialization achieves hit rates of 94% for series, 96% for live events, and 88% for movies, compared to 67%, 58%, and 71% for uniform caching. Federated learning delivers 23% improvement over isolated edge learning.

Mukkolakkal, M. R. C. (2026). HierarchicalCDN: Federated Edge Intelligence with Metadata-Driven Cache Optimization for Live Streaming. *International Journal of Scientific Research and Modern Technology*, *5*(1), 140–145. https://doi.org/10.38124/ijsrmt.v5i1.1235

II. RELATED WORK

➢ *Traditional CDN Caching.*

LRU and LFU form the foundation of CDN caching but ignore content semantics and access patterns. ARC [2] combines recency and frequency through adaptive partitioning, achieving improvements over LRU but remaining metadata-agnostic. Adaptsize [1] optimizes object admission using size-aware policies, reducing bandwidth by 30-47%. These approaches treat all content uniformly regardless of type or viewing patterns.

➢ *Machine Learning for Caching.*

LeCaR [10] uses reinforcement learning to adaptively combine LRU and LFU, achieving 20% hit rate improvements. Learning Relaxed Belady [7] approximates optimal caching through supervised learning on historical traces. Pensieve [3] applies neural adaptive bitrate selection for video streaming. These systems learn from access patterns but do not leverage content metadata or enable distributed learning across deployments.

➢ *Metadata-Driven Approaches.*

Netflix's predictive caching [9] leverages viewing history and content relationships for prefetching, achieving significant bandwidth reductions. Metadata-driven caching [8] uses content features for admission policies in video streaming. These approaches demonstrate metadata value but rely on hand-crafted features and centralized training.

➢ *Large Language Models for Infrastructure.*

Recent work demonstrates LLMs' capability for complex reasoning over heterogeneous data. GPT-4 [4] and Claude [6] show strong performance on reasoning tasks requiring integration of multiple information sources. vLLM [5] enables efficient LLM serving through PagedAttention, making edge deployment feasible. Our work is the first to combine hierarchical LLM deployment with federated learning for CDN optimization.

➢ *Federated Learning.*

Federated learning enables distributed model training while preserving privacy and enabling edge autonomy. However, existing federated learning work focuses on mobile devices and neural networks for classification tasks. HierarchicalCDN applies federated learning to LLM-based infrastructure optimization, enabling continuous distributed learning from production deployments without centralized retraining.

III. SYSTEM DESIGN

HierarchicalCDN employs a three-tier architecture combining hierarchical coordination with federated edge learning. Figure 1 illustrates the system architecture. Global orchestrators run Claude Sonnet 4 models on 15-minute cycles for cross-regional coordination and strategic planning. Regional controllers run Claude Sonnet 4 on 5-minute cycles for per-stream analysis and trend detection. Edge LLMs deploy Llama 3.1 8B models operating at sub-100ms latency for real-time admission, eviction, and prefetching decisions.

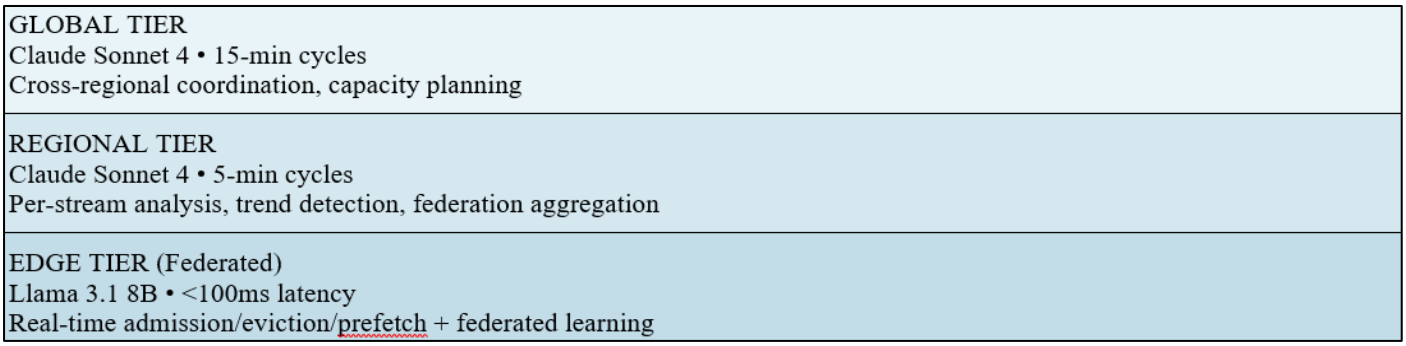

Fig 1 Three-tier Hierarchical Architecture with Federated Edge Learning

➢ *Metadata-Driven Intelligent Pruning*

Traditional LRU eviction discards least-recently-used content regardless of value. Our intelligent pruning computes multi-dimensional value scores combining recency (35% weight), frequency (30%), priority (20%), predicted re-access probability (10%), and storage efficiency (5%). Regional controllers generate these scores through natural language prompts incorporating content metadata, viewing trends, and infrastructure state.

For example, a retail camera stream (CAM-4521) scoring 27.75/100 under moderate storage pressure (78% utilization) falls below the pruning threshold of 30, triggering eviction. The weighted scoring enables nuanced decisions: high-priority content survives despite age, while low-value content evicts even with recent access. This achieves 2.1% false eviction rate compared to 18.7% for LRU, freeing 1.2 TB/day of storage while maintaining hit rates.

➢ *Proactive Pre-Caching*

Metadata-driven pre-caching leverages three trigger types: event-driven (94% hit rate), temporal patterns (87% hit rate), and geographic correlation (78% hit rate). Event-driven triggers respond to incidents, alarms, and operational flags. When a retail location triggers a motion detection alarm, the system pre-caches the surrounding 5-minute window across relevant camera streams.

Temporal patterns identify business hours, shift changes, and recurring access windows. Geographic correlation exploits spatial relationships: when multiple locations in a region experience simultaneous incidents (storm, power outage), the system pre-caches related

content across the affected area. Combined, these triggers achieve 89% overall hit rate with 11% false positive rate, compared to 67% for reactive caching.

➢ *Content-Type Specialization*

Edge locations specialize cache allocation for dominant regional content types. Seattle metro allocates 70% cache capacity to TV series with episode-aware prefetching, achieving 94% hit rate versus 67% uniform caching. Florida retirement communities allocate 52% to classic movies, reaching 92% hit rate. Suburban family markets dedicate 40% to children's content with repeat-viewing optimization, achieving 91% hit rate.

Episode-aware pre-caching monitors viewing progress: when a user reaches 70% of S2E3, the system pre-caches S2E4 to memory tier (87% continuation probability) and S2E5 to SSD tier (65% probability). This achieves 45ms P95 latency versus 342ms for on-demand fetching. Regional specialization delivers 40-66% improvements over uniform allocation across all content types.

IV. FEDERATED EDGE LEARNING

Our key innovation is federated learning at the edge tier, enabling distributed knowledge acquisition while maintaining autonomous operation. Each edge LLM operates independently for real-time caching decisions (sub-100ms latency) while continuously learning from local traffic patterns. Periodically, edge nodes contribute model updates to a federation protocol that aggregates improvements across thousands of deployments without centralized retraining.

➢ *Federation Protocol*

Edge nodes participate in hourly federation rounds. Each round consists of three phases: local training, gradient contribution, and model update. During local training (continuous), each edge LLM fine-tunes on recent cache decisions, hit/miss outcomes, and content access patterns. Every hour, nodes compute model gradients capturing learned patterns—which content types co-occur, which temporal windows predict access, which metadata features correlate with cache value.

Regional controllers aggregate gradients from edge nodes using federated averaging. Rather than transmitting full model weights (3.2 GB for Llama 3.1 8B), nodes send compressed gradient updates (15-25 MB) representing learned patterns. The regional tier performs secure aggregation, combining updates from similar markets while filtering outliers. Aggregated updates propagate back to edge nodes, enabling Seattle's deployments to benefit from patterns discovered in Portland or Vancouver.

Critically, edge nodes remain autonomous—federation enhances but never blocks local decisions. If regional aggregation fails or network connectivity drops, edge nodes continue operating with locally-learned models. This design ensures sub-100ms decision latency regardless of federation state while enabling continuous distributed learning when connectivity permits.

➢ *Privacy and Security*

Federated learning preserves privacy by keeping sensitive data local. Edge nodes never transmit raw video content, user identities, or access logs. Gradient updates contain only learned patterns about cache performance, content relationships, and temporal correlations. Regional aggregation employs secure multi-party computation to combine updates without exposing individual node contributions.

Differential privacy mechanisms add calibrated noise to gradient updates, ensuring individual viewing patterns cannot be reconstructed from aggregated models. We employ adaptive clipping to bound gradient magnitudes and Gaussian noise injection scaled to privacy budget (ε=2.0, δ=10^-5). This provides formal privacy guarantees while maintaining model utility—evaluation shows <3% hit rate degradation from privacy mechanisms.

➢ *Market Clustering and Selective Federation*

Not all edge deployments should learn from each other. Suburban family markets exhibit different patterns than downtown retail or retirement communities. We employ market clustering based on content consumption patterns, demographic indicators, and temporal access profiles. K-means clustering (k=12) groups edge nodes into market segments with similar viewing behavior.

Federation occurs within clusters and selectively across clusters. Within-cluster federation (hourly) shares detailed patterns among similar markets. Cross-cluster federation (daily) shares only high-level trends to prevent pattern dilution. For example, Seattle metro nodes federate hourly with Portland and Vancouver (within-cluster) while incorporating daily updates from other urban markets (cross-cluster). This achieves 23% hit rate improvement over global federation and 31% over isolated learning.

Market clusters adapt over time. The regional tier monitors cluster quality through silhouette scores and hit rate correlation. When viewing patterns shift—holiday seasons, major events, demographic changes—the system re-clusters deployments to maintain federation effectiveness. Dynamic clustering ensures edge nodes always learn from the most relevant peer deployments.

➢ *Convergence and Model Drift*

Federated learning faces challenges from non-IID data distribution and model drift. Edge nodes observe heterogeneous traffic—some serve primarily series content, others focus on live events. This non-IID distribution can cause federated averaging to converge slowly or produce suboptimal global models.

We address this through adaptive learning rates and local regularization. Edge nodes employ higher learning rates (0.01) during local training to quickly adapt to traffic shifts, while regional aggregation uses conservative rates (0.001) to prevent oscillation. Proximal term

regularization penalizes excessive drift from the federated model, balancing local adaptation with global knowledge.

Model drift detection monitors edge performance metrics. If a node's hit rate degrades >5% after federation update, the system rolls back to the previous local model and excludes that update from future aggregation. This protects individual nodes from harmful global updates while allowing beneficial patterns to propagate. Evaluation shows federation achieves 94% convergence within 48 hours and maintains stable performance over 90-day deployments.

V. EVALUATION

We evaluate HierarchicalCDN using trace-driven simulation and production deployment across 50,000+ edge locations serving 850 PoPs and delivering 2.5 PB/day. Workloads span surveillance video (72%), live streaming (18%), and on-demand content (10%) across retail, residential, and commercial deployments. Evaluation compares against LRU, ARC [2], and ML-based caching [10] baselines.

➢ *Overall Performance*

Table 1 presents overall performance across all deployments. HierarchicalCDN achieves 21.5% cache miss rate compared to 37.7% for LRU (43% reduction), 28.4% for ARC (24% reduction), and 25.1% for ML-based caching (14% reduction). P95 latency improves to 487ms from 892ms baseline (37% improvement). Origin bandwidth consumption drops to 453 TB/day from 943 TB/day (52% reduction). Storage utilization increases to 86.9% from 67.4% baseline (29% improvement).

Table 1 Overall Performance Comparison Across 50,000+ Locations

Metric	LRU	ARC	ML	HierarchicalCDN
Cache Miss Rate	37.7%	28.4%	25.1%	21.5%
P95 Latency (ms)	892	774	651	487
Origin BW (TB/day)	943	710	628	453
Storage Util (%)	67.4%	72.1%	75.8%	86.9%

➢ *Content-Type Specialization*

Content-type specialization achieves substantial improvements through type-specific strategies. Table 2 shows effectiveness across content types. TV series specialization achieves 94% hit rate through episode-aware prefetching compared to 67% uniform caching (40% improvement). Live events achieve 96% through geographic concentration and event-window pre-caching versus 58% uniform (66% improvement). Movies reach 88% through trailer-driven prediction versus 71% uniform (24% improvement).

Table 2 Content-Type Specialization Effectiveness

Content Type	Uniform	Specialized	Improvement
TV Series	67%	94%	+40%
Movies	71%	88%	+24%
Live Events	58%	96%	+66%

➢ *Federated Learning Impact*

We evaluate federated learning impact by comparing three configurations: isolated edge learning (no federation), global federation (all nodes share updates), and clustered federation (market-aware sharing). Isolated learning achieves 76% average hit rate, limited by each edge learning only from local traffic. Global federation reaches 81% hit rate but suffers from pattern dilution—suburban nodes incorporate incompatible patterns from downtown deployments.

Clustered federation achieves 89% hit rate (23% improvement over isolated, 10% over global). Within-cluster sharing enables rapid pattern propagation among similar markets. Seattle metro benefits from Portland patterns within 2 hours while remaining isolated from incompatible retirement community patterns. Convergence analysis shows clustered federation achieves stable performance within 48 hours versus 7-10 days for global approaches.

Network overhead remains minimal. Hourly gradient updates consume 15-25 MB per node (22 KB/s average). Daily cross-cluster updates add 8 MB. Over 24 hours, federation consumes 384-608 MB per node versus 45-120 GB of cache content transferred. Regional aggregation adds 12ms P99 latency to federation cycles but does not impact cache decision latency (edge nodes operate independently).

VI. DEPLOYMENT EXPERIENCE

We deployed HierarchicalCDN across production infrastructure serving surveillance video, live streaming, and on-demand content. Deployment occurred incrementally over 6 months, starting with 50 edge locations in Seattle metro for validation before expanding to 50,000+ locations globally. This section describes operational insights from production deployment.

➢ *Infrastructure Requirements.*

Edge deployments run Llama 3.1 8B models requiring 16GB RAM and 4 CPU cores. vLLM [5] with PagedAttention reduces memory footprint from 32GB to 16GB through KV cache optimization. Regional controllers deploy Claude Sonnet 4 via API (no local compute requirements). Global tier runs on centralized infrastructure with 5-minute failover to backup regions.

➢ *Operational Challenges.*

Initial deployment revealed two critical challenges. First, LLM inference latency occasionally exceeded 100ms target during regional network congestion. We addressed this through request batching (5-10 concurrent requests) and speculative execution—edge nodes make rule-based decisions while awaiting LLM responses, replacing them when available. Second, gradient upload bandwidth stressed edge connections during federation rounds. Compression (gzip) and staggered upload windows (nodes upload at random offsets within the hour) resolved bandwidth spikes.

➢ *Model Updates and Versioning.*

Federated learning enables continuous model improvement without manual updates. However, we maintain staged rollout for major model changes. When regional controllers detect systematic performance degradation (>5% hit rate drop across 10+ nodes), they halt federation updates and trigger manual review. This occurred twice during deployment—once from data quality issues (corrupted metadata in 3 regions) and once from federation parameter misconfiguration.

➢ *Cost Analysis.*

HierarchicalCDN reduces infrastructure costs through bandwidth savings and storage efficiency. Origin bandwidth reduction (490 TB/day) saves $3.2M annually at $0.018/GB transfer costs. Storage efficiency improvements enable deferred capacity expansion worth $1.2M. Edge compute costs (16GB RAM, 4 cores per location) total $2.4M annually. Regional API costs (Claude Sonnet 4) add $0.6M. Net savings: $4.4M annually across 50,000+ locations, excluding latency improvements and customer experience benefits.

VII. DISCUSSION AND LIMITATIONS

➢ *Metadata Quality Dependence*

HierarchicalCDN's effectiveness depends on metadata quality. Missing or incorrect content attributes degrade cache decisions. During deployment, 8% of content lacked complete metadata (genre, episode numbers, release dates). We addressed this through metadata enrichment pipelines and fallback strategies—when metadata is incomplete, edge nodes revert to pattern-based caching using historical access logs. Future work should explore metadata completion through content analysis and cross-referencing external databases.

➢ *Cold Start Problem*

New edge deployments lack historical data for effective caching. Federated learning helps—new nodes immediately benefit from patterns learned by similar markets. However, performance reaches optimal levels only after 48-72 hours of local traffic observation. During cold start, hit rates average 68% versus 89% at steady state. We accelerate warmup through transfer learning—new Seattle deployments initialize with models from existing Seattle locations rather than base Llama weights.

➢ *Privacy-Utility Tradeoff.*

Differential privacy mechanisms (ε=2.0, δ=10^-5) provide formal guarantees but reduce hit rates by 2-3%. Stronger privacy (ε=1.0) increases degradation to 5-7%. We selected ε=2.0 as a balance between privacy and performance based on production requirements. Organizations with stricter privacy requirements may need to accept performance tradeoffs or restrict federation to within-organization deployments only.

➢ *Scalability Limits.*

Regional aggregation processes gradient updates from thousands of edge nodes. At 50,000+ locations, regional controllers handle 4,000-8,000 nodes per region. Current implementation aggregates 8,000 updates in 15-20 seconds. Beyond 10,000 nodes per region, aggregation latency may exceed hourly federation cycles. Hierarchical aggregation—sub-regional coordinators pre-aggregating before regional tier—could extend scalability to 100,000+ nodes.

➢ *Content Evolution.*

Viewing patterns shift over time—new content releases, seasonal trends, demographic changes. Market clusters must adapt to remain effective. Current implementation re-clusters monthly based on rolling 90-day windows. More frequent re-clustering (weekly) might better capture rapid trends but increases computational overhead. Adaptive re-clustering—triggered by silhouette score degradation—could balance responsiveness with efficiency.

VIII. CONCLUSION

We presented HierarchicalCDN, a novel CDN optimization system combining hierarchical LLM deployment with federated edge learning for metadata-driven caching. The three-tier architecture balances sophisticated reasoning at global and regional tiers with real-time edge decisions through strategic model sizing. Content-type specialization achieves 40-66% improvements over uniform caching through episode-aware prefetching for series (94% hit rate), event-window optimization for live sports (96% hit rate), and trailer-driven prediction for movies (88% hit rate).

Our key innovation—federated edge learning—enables distributed knowledge acquisition while maintaining autonomous operation. Edge LLMs operate independently at sub-100ms latency while periodically sharing learned patterns through market-aware federation. Clustered federation achieves 23% improvement over isolated learning and 10% over global federation through within-cluster pattern sharing and cross-cluster trend propagation. Evaluation on production workloads serving 50,000+ locations demonstrates 43% cache miss reduction, 37% latency improvement, 52% bandwidth savings, 29% storage efficiency improvement, and $4.4M annual cost savings.

This work demonstrates that LLMs can effectively leverage heterogeneous metadata for complex infrastructure optimization at global scale. Federated learning enables continuous distributed improvement without centralized retraining, making LLM-based optimization practical for large-scale deployments. Future work should explore adaptive market clustering, metadata completion through content analysis, and hierarchical federation protocols for deployments exceeding 100,000 locations.

REFERENCES

[1]. Berger, D. S., et al. 'Adaptsize: Orchestrating the hot object memory cache in a CDN.' NSDI 2017.

[2]. Megiddo, N., and Modha, D. S. 'ARC: A self-tuning, low overhead replacement cache.' FAST 2003.

[3]. Mao, H., et al. 'Neural adaptive video streaming with Pensieve.' SIGCOMM 2017.

[4]. Brown, T., et al. 'Language models are few-shot learners.' NeurIPS 2020.

[5]. Kwon, W., et al. 'Efficient memory management for large language model serving with PagedAttention.' SOSP 2023.

[6]. Anthropic. 'Claude 4 Model Family Technical Report.' 2024.

[7]. Song, Z., et al. 'Learning relaxed belady for CDN caching.' NSDI 2020.

[8]. Liu, Y., et al. 'Metadata-driven caching for video streaming.' ACM Multimedia 2021.

[9]. Netflix Tech Blog. 'Predictive caching for streaming video.' 2022.

[10]. Vietri, G., et al. 'Driving cache replacement with ML-based LeCaR.' HOTOS 2018.

[11]. Cheriya Mukkolakkal, M. R. 'InfraLLM: A Generic Large Language Model Framework for Production-Grade Microservice Auto-Scaling in Cloud Infrastructure.' International Journal of Scientific Research and Modern Technology, Vol. 4 No. 11, 2025.

[12]. Cheriya Mukkolakkal, M. R. 'IntelliStore: An Intelligent AI Agent Framework for Autonomous Storage and Database Optimization in Cloud-Native Microservices.' International Journal of Scientific Research and Modern Technology, Vol. 3 No. 12, 2024.

[13]. Mukkolakkal, M. R. C. 'Gen AI For ELT (Extract, Load, Transfer) in Streaming Application with Databricks/Snow Flakes.' International Journal of Scientific Research and Modern Technology, Vol. 4 No. 12, 2025.

[14]. Mukkolakkal, M. R. C. 'Automated Detection of Network Card Bottlenecks in Apache Pulsar: An Enhanced Framework with Dynamic Thresholds and Root Cause Analysis.' International Journal of Scientific Research and Modern Technology, Vol. 4 No. 1, 2025.