Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Optimizing Cloud Data Costs: FinOps and Usage-Based Workload Segmentation Strategies

Venkateswarlu Boggavarapu

Visvesvaraya Technological University (VTU), India.

ARTICLE INFO

Received: 05 Nov 2025

Revised: 25 Dec 2025

Accepted: 03 Jan 2026

ABSTRACT

Financial institutions face escalating infrastructure expenses driven by consumption-based cloud pricing models and insufficient cost governance frameworks. Traditional capacity planning methodologies often fail to address the dynamic resource requirements of multi-cloud architectures that host transactional systems, regulatory compliance platforms, and analytical workloads. The article presents integrated strategies that combine financial operations principles with usage-based workload segmentation and platform-specific optimization techniques. Cloud storage costs accumulate across multiple service tiers, each exhibiting distinct pricing characteristics for data ingress, egress, persistence, and API requests. Financial Operations frameworks create broad accountability across financial, technology and business teams by following a phased approach targeting cost, efficiency, and quality. AI technologies assist in cost management through enhanced predictive modeling and anomaly detection. Multi-cloud resource allocation algorithms simultaneously consider multiple criteria and constraints related to infrastructure costs, performance service levels, and security compliance. Workload classification taxonomies enable targeted optimization strategies appropriate for production-critical systems, development environments, batch processing operations, and analytical queries. Serverless architectures eliminate idle resource costs through event-driven execution models, charging exclusively for actual consumption periods. Column-oriented database systems integrate compression directly into query execution reducing storage footprints while maintaining analytical performance. Comprehensive chargeback models establish financial accountability by allocating actual cloud expenses to consuming business units through granular cost attribution mechanisms.

Keywords: Cloud Cost Optimization, Financial Operations Framework, Workload Segmentation, Serverless Computing, Data Warehouse Optimization, Chargeback Implementation

Introduction

Cloud computing has transformed financial services infrastructure by enabling scalable data processing and analytics capabilities. Financial institutions run transaction processing systems, risk modeling platforms and regulatory compliance tools in the cloud. The transition from capital-intensive on-premises infrastructure to consumption-based cloud services fundamentally changed the way organizations pay for technology. This evolution brought several challenges for the companies about cost planning and control.

Financial institutions process transactional data across distributed cloud environments while generating continuous regulatory reports for multiple jurisdictions. Analytical workloads span various cloud regions and services simultaneously. The consumption-based pricing models create challenges

in tracking actual resource utilization patterns. Organizations grapple with budget overruns when the provisioned capacity surpasses what is necessary for regular operations. Additionally, the inability to trace cost allocations hinders the identification of which business activities or departments generate these expenses.

Cloud data warehouses and distributed analytics platforms automatically scale compute resources based on query complexity and the volume of processed data. Resource consumption fluctuates hourly according to trading activity during market operations. Month-end batch processing schedules create demand surges that strain capacity planning approaches designed for static infrastructure. Business analysts submit ad-hoc queries that generate unpredictable computational loads. Traditional capacity planning methodologies often fail to accommodate these dynamic consumption patterns effectively.

The absence of standardized cost allocation mechanisms hinders the implementation of accurate chargebacks across business units. Departments consuming substantial cloud resources often face no direct financial consequences for their usage patterns. This misalignment of incentives perpetuates resource overconsumption behaviors throughout organizational hierarchies. Cloud storage represents a particularly complex cost component requiring careful management across multiple service tiers and access patterns. Research examining cloud storage economics identifies distinct cost categories, including data ingress charges, egress fees, storage persistence costs, and API request pricing structures [1]. Each category exhibits unique pricing characteristics that demand specialized optimization strategies. Storage costs accumulate through retention of historical data, replication across geographic regions for disaster recovery, and maintenance of multiple environment copies for development and testing purposes.

Financial Operations frameworks address these challenges through systematic approaches to cloud cost governance. The discipline establishes collaborative practices involving finance teams, technology groups, and business stakeholders. Organizations implementing comprehensive FinOps methodologies achieve substantial cost optimization improvements through enhanced visibility mechanisms and automated resource management [2]. The framework operates through iterative phases focusing on cost transparency, optimization opportunities, and operational excellence. Visibility tools enable detailed tracking of resource consumption patterns across organizational boundaries. Automated rightsizing adjusts provisioned capacity to match actual workload requirements. Business unit accountability measures create financial incentives for efficient resource utilization and allocation.

Cloud cost optimization requires integration of organizational practices with technical architectural decisions. Workload segmentation approaches categorize resources based on usage characteristics and performance requirements. Platform-specific optimization techniques address unique cost structures inherent to different cloud services. This article presents integrated strategies that combine financial operations disciplines with intelligent workload placement and cloud data warehouse optimization methods tailored for financial services environments.

FinOps Framework and Cost Allocation Mechanisms

Financial Operations establishes organizational practices for cloud costs management and accountability. The framework emerged as cloud adoption matured, moving beyond experimental workloads to production-critical systems. Traditional IT financial management approaches have proven insufficient for dynamic cloud environments, where resources are provisioned and deprovisioned within minutes. Organizations require new methodologies that bridge financial planning disciplines with cloud infrastructure management practices.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

The framework operates on three foundational phases: inform, optimize, and manage. The inform phase establishes cost visibility through comprehensive data collection and reporting mechanisms. Organizations deploy tagging taxonomies that capture business unit identifiers, application classifications, environment types, and cost center allocations. These metadata structures enable multidimensional cost reporting capabilities. Expenses are attributed to specific projects, departments, or revenue-generating activities through hierarchical classification systems. Cost visibility requires integration of billing data from multiple cloud providers into centralized analytics platforms. Financial institutions operating multi-cloud architectures face particular complexity in normalizing cost data across different provider billing formats.

Artificial intelligence technologies enhance FinOps capabilities through predictive cost modeling and automated optimization recommendations. Machine learning algorithms analyze historical consumption patterns to forecast future spending trajectories across different cloud service categories. AI-powered analytics can gather data from other departments to figure out the areas where a company could be spending too much money. Predictive models identify discrepancies in spending that will occur before costs are increased. Pattern recognition algorithms detect inefficient resource configurations that human analysts might overlook during manual reviews. Multi-cloud environments particularly benefit from AI-enhanced approaches that optimize workload placement across different providers based on performance requirements and cost objectives [3].

It is possible to lower expenses and improve efficiency by making adjustments to architectural design and implementation. Analysis of how resources are utilized reveals several issues, including overprovisioned instances, idle resources, and inefficient architectural setups. Organizations implement automated rightsizing recommendations, which adjust compute capacity to align with workload requirements. These strategies automatically transfer infrequently accessed data to lower-cost storage locations. Commitment-based discount programs reduce the cost per unit by enabling customers to purchase capacity in advance for predictable workloads. The optimization phase operates continuously as new services deploy and consumption patterns evolve.

The operate phase embeds cost optimization practices into standard operational procedures. Engineering teams incorporate cost considerations into architectural design reviews and deployment approval workflows. Automated policies that enforce limits on project or departmental budgets. Anomaly detection systems send notifications when usage patterns significantly deviate from historical baselines. In the operational phase, cost management is an ongoing practice.

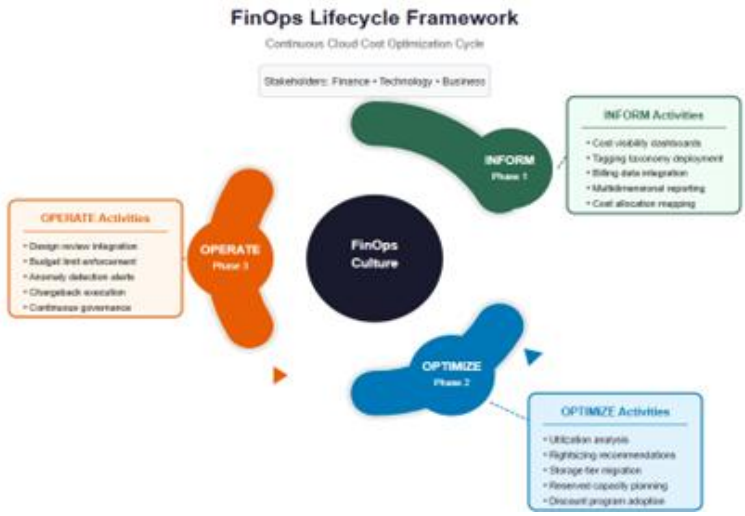

Fig 1. FinOps Lifecycle Framework illustrating iterative phases for cloud cost optimization [2].

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Hierarchical Cost Attribution Models

Implementing effective cost allocation requires hierarchical tagging strategies that capture organizational structure while maintaining flexibility for matrix reporting. Resource tags must include mandatory fields for department identifiers, application names, environment classifications, and data sensitivity levels. Financial institutions typically operate complex organizational hierarchies where applications serve multiple business units simultaneously. Shared services present particular challenges requiring cost distribution algorithms that allocate infrastructure expenses based on proportional usage metrics.

Resource allocation techniques directly impact the accuracy of cost attribution across cloud environments—dynamic allocation mechanisms that automatically adjust resources based on real-time demand. Load balancing algorithms primarily focus on distributing the workload across available infrastructure resources to maximize their utilization efficiency. Auto-scaling policies allocate additional capacity in times of heightened demand and release resources when activity declines. Resource allocation decisions affect both performance characteristics and cost outcomes. Effective allocation strategies balance multiple objectives, including application availability requirements, response time targets, and budgetary constraints [4].

Cost allocation accuracy depends on comprehensive tagging coverage across all cloud resources. Organizations struggle to maintain consistent tagging practices as engineering teams deploy new services. Automated validation systems prevent the deployment of untagged resources by utilizing policy engines and infrastructure provisioning workflows. These guardrails ensure new resources include required metadata at creation time rather than requiring retroactive tagging efforts.

API call volumes, storage consumption, and compute time consumed serve as standard basis metrics for proportional cost allocation. Shared database services distribute costs based on the query execution time attributed to different applications. The networking infrastructure allocates expenses based on the data transfer volumes originating from specific services. Organizations develop custom allocation algorithms that reflect their particular operational models and business structures. Financial institutions allocate trading platform infrastructure costs proportionally to transaction volumes processed for different product lines.

Chargeback models implement financial accountability by transferring actual infrastructure costs to consuming business units through internal billing mechanisms. Showback approaches provide cost transparency without formal budget transfers. Organizations typically begin with showback implementations to establish visibility before transitioning to full chargeback models. Chargeback creates direct financial incentives for efficient resource utilization by linking consumption decisions to departmental budgets, thereby promoting effective resource allocation.

Accountability Through Unit Economics

Unit economics frameworks translate infrastructure costs into business-relevant metrics that resonate with stakeholders outside technology organizations. For transaction processing systems, costs are expressed as per-transaction amounts or per-account maintenance fees. Analytics platforms measure expenses per query executed or per dataset analyzed. These unit cost metrics facilitate meaningful conversations between technology teams and business leaders.

Unit cost calculations require an accurate mapping between infrastructure resources and business activities. The transaction processing platform's instrument code is used to capture resource consumption attributable to individual transaction types. Storage costs are allocated across customer accounts based on data volumes maintained for each relationship. Compute expenses for analytical

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

workloads that track to specific report types or dashboard applications. Granular cost attribution enables the identification of high-cost business processes that warrant optimization attention.

Financial institutions calculate unit economics across diverse operational domains. Payment processing systems measure cost per authorization request, per settlement transaction, and per fraud detection screening. Lending platforms track expenses per application processed and per loan origination workflow. Customer relationship management systems express costs per user account and per service interaction. Unit economics provide business stakeholders with actionable cost intelligence that informs pricing strategies and technology investment decisions.

Table 1. FinOps Framework Phases and Cost Allocation Components Implementation Characteristics Across Organizational Practices [3, 4].

Framework Phase	Core Activities	Cost Allocation Method	Key Benefits
Inform	Cost visibility, tagging deployment, billing integration, multidimensional reporting	Business unit tags, application IDs, department codes, and cost center tracking	Real-time dashboards, spending trends, granular metrics, and consumption tracking
Optimize	Utilization analysis, rightsizing, storage migration, discount programs	API volume tracking, storage measurement, compute time allocation, usage-based distribution	Automated validation, policy enforcement, tag compliance, resource efficiency
Operate	Design reviews, spending limits, anomaly detection, operational embedding	Unit economics, per-transaction costs, per-query expenses, activity-based allocation	Cost baselines, automated alerts, continuous monitoring, and spending controls

FinOps Maturity and Value-Driven Optimization

Cloud cost management strategies have evolved beyond simple expense reduction toward value-driven optimization aligned with business outcomes. Early FinOps implementations concentrated primarily on identifying waste and cutting unnecessary spending. Mature organizations now prioritize business agility, time-to-market acceleration, and innovation enablement alongside cost efficiency. Financial operations frameworks create broad accountability across financial, technology, and business teams through phased approaches targeting cost visibility, operational efficiency, and quality improvement [2]. The strategic focus has shifted from minimizing cloud bills to maximizing return on cloud investments. Cost optimization decisions now incorporate business value assessments that weigh infrastructure expenses against competitive advantages gained through faster deployment cycles and enhanced scalability.

Automation with Human Guardrails

Automation represents a leading priority for organizations advancing FinOps maturity. Automated scaling policies adjust resource capacity based on real-time demand signals. Rightsizing recommendations execute automatically when utilization patterns fall below defined thresholds. AI-driven anomaly detection identifies spending deviations within hours of occurrence [3]. However, full

autonomous optimization remains a long-term aspiration rather than current practice. Most organizations maintain human oversight for final decision-making on significant resource changes. Guardrails establish boundaries within which automated systems operate independently. Threshold-based approvals require human confirmation before executing changes exceeding predefined cost or capacity limits.

Resource allocation techniques balance automation benefits against risk management requirements. Dynamic allocation mechanisms automatically adjust resources based on real-time demand while load balancing algorithms distribute workloads to maximize utilization efficiency [4]. Auto-scaling policies provision additional capacity during demand surges and release resources when activity declines. CPU optimization strategies include rightsizing instance types to match workload requirements and configuring scaling policies that respond to utilization metrics [6]. Multi-cloud environments benefit from AI-enhanced approaches that optimize workload placement across providers based on performance requirements and cost objectives simultaneously [5]. Machine learning models analyze historical performance data to predict optimal resource configurations. Pattern recognition algorithms detect inefficient configurations that manual reviews might overlook [3]. The progression toward autonomous optimization proceeds incrementally as organizations build confidence in automated decision-making accuracy.

Governance Frameworks and Cost Standardization

Robust cloud governance frameworks address operational gaps created during rapid cloud adoption phases. Many organizations migrated workloads hastily without establishing consistent tagging taxonomies or cost attribution mechanisms. Governance initiatives remediate these gaps through standardized metadata requirements and policy enforcement. Proper tagging enables financial accountability by attributing actual cloud expenses to consuming business units through granular cost allocation [2]. Automated validation systems prevent deployment of untagged resources by integrating policy engines with infrastructure provisioning workflows.

The FinOps Open Cost and Usage Specification represents a significant industry effort toward billing data standardization. Cloud providers historically delivered billing information in proprietary formats with inconsistent terminology and structure. Organizations operating multi-cloud architectures faced substantial complexity normalizing cost data across different provider billing systems [2]. FOCUS establishes common schemas and definitions enabling consistent cost analysis regardless of cloud provider. Standardized billing data simplifies multi-cloud cost aggregation and benchmarking. Financial institutions benefit from reduced integration overhead and improved accuracy in cross-provider cost comparisons.

Advanced Cost Forecasting and Predictive Analytics

Accurate cloud cost forecasting remains a persistent challenge for financial institutions operating complex multi-cloud environments. Traditional budgeting approaches rely on historical averages and linear projections. Such methods fail to capture dynamic consumption patterns inherent to cloud infrastructure. Workload variability, seasonal demand fluctuations, and unpredictable analytical queries introduce forecasting errors. Budget overruns occur when actual consumption exceeds projections based on static assumptions. FinOps teams increasingly adopt predictive analytics to transition from reactive cost management toward proactive financial planning.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

AI-Driven Forecasting Models

Machine learning algorithms analyze historical consumption patterns to forecast future spending trajectories. AI-enhanced FinOps platforms apply predictive cost optimization techniques across multiple cloud providers. Pattern recognition algorithms identify consumption trends that inform accurate budget projections. Predictive models detect spending anomalies before costs escalate significantly [3]. Financial institutions benefit from early warning systems that flag potential overruns during billing cycles rather than after month-end reconciliation. The proactive approach enables corrective actions before budget thresholds breach.

Advanced forecasting methodologies leverage multiple algorithmic approaches for improved accuracy. Machine learning models establish baseline relationships between workload characteristics and resource consumption. Deep learning architectures capture complex nonlinear dependencies in spending data. Regression-based techniques provide interpretable forecasts suitable for financial planning discussions. Hybrid models combine multiple algorithmic approaches to balance accuracy with explainability [12]. Financial institutions select forecasting techniques based on data availability, accuracy requirements, and organizational comfort with algorithmic complexity.

Pattern Recognition and Trend Analysis

Forecasting accuracy depends on robust pattern recognition capabilities. Cloud consumption exhibits multiple cyclical patterns at daily, weekly, and monthly intervals. Trading platforms generate predictable demand spikes during market hours. Month-end processing creates recurring capacity surges. Seasonal business cycles influence analytical workload volumes. Supervised learning methods train on labeled historical data to recognize these patterns. Unsupervised clustering techniques identify consumption segments without predefined categories. Semi-supervised hybrid approaches combine labeled examples with unlabeled data for improved generalization [8]. Pattern recognition algorithms decompose consumption time series into trend, seasonal, and residual components.

Anomaly-adjusted forecasting separates normal consumption growth from exceptional events. One-time migration projects or regulatory initiatives create temporary spending spikes. Forecasting models must distinguish between permanent consumption increases and transient anomalies. Historical anomalies receive appropriate weighting to avoid distorting future projections. Rolling forecast windows continuously update predictions as new consumption data arrives. Adaptive models adjust parameters in response to changing workload characteristics.

Proactive Financial Planning

Predictive analytics transforms FinOps teams from cost reporters into strategic advisors. Accurate forecasts enable informed capacity commitment decisions. Reserved instance purchases require confidence in future consumption levels. Undercommitment sacrifices available discounts. Overcommitment creates stranded capacity costs. Forecasting models quantify commitment risks under different consumption scenarios. Financial institutions optimize commitment portfolios balancing discount capture against flexibility preservation.

Budget allocation processes benefit from consumption forecasts at business unit levels. Department leaders receive projected costs enabling informed resource planning. Forecast variance analysis identifies areas requiring optimization attention. Continuous forecast refinement improves accuracy over successive planning cycles. FinOps maturity advances as organizations embed predictive capabilities into standard financial processes.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Usage-Based Workload Segmentation Strategies

Workload segmentation classifies cloud assets primarily based on usage styles, performance necessities, and criticality to enterprise operations. This classification enables centered optimization strategies appropriate to every workload category. Financial institutions handle different types of workloads ranging from real-time transaction systems to scheduled batch operations. These categories differ in terms of their sensitivity to performance, availability requirements, and cost tolerance levels. Effective segmentation requires a comprehensive analysis of application behavior patterns over extended observation periods.

Financial institutions typically segment workloads into four categories: production-critical, development and testing, batch processing, and analytical queries. Production systems support customer-facing applications and core banking functions that require continuous availability. Development environments support software engineering activities and quality assurance testing. Batch processing handles scheduled operations, including end-of-day settlement, regulatory reporting generation, and data warehouse updates. Analytical workloads execute queries against historical datasets for business intelligence and risk modeling purposes. Different categories reflect different resource allocation strategies, which demonstrate varying priorities for performance, availability, and cost optimization.

Multi-cloud environments introduce an additional layer of complexity to workload segmentation and resource allocation decisions. Companies spread their workloads across several cloud providers to avoid vendor lock-in and take advantage of specialized services. AI algorithms determine where to place a workload across different cloud platforms by considering the cost structures, performance capabilities, and security requirements simultaneously. Machine learning models analyze historical performance data to predict optimal resource configurations for different workload types. The AI-driven approach balances competing objectives, including minimizing infrastructure costs, meeting performance service level agreements, and maintaining security compliance standards [5]. Multi-cloud resource allocation particularly benefits financial institutions operating under strict regulatory requirements that mandate data residency controls and disaster recovery capabilities across geographic regions.

Workload Classification Taxonomy

Production-critical workloads demand high availability guarantees and consistent performance characteristics. Transaction processing systems cannot tolerate service interruptions during business hours without impacting customer experience and revenue generation. Trading platforms require low-latency response times for order execution functions. Payment authorization systems must maintain strict availability targets to prevent transaction declines and ensure seamless processing. Production workloads justify reserved capacity commitments that reduce per-unit costs through long-term usage commitments spanning extended contract periods.

Reserved capacity models exchange upfront financial commitments or long-term usage obligations for reduced hourly rates compared to on-demand pricing. Organizations analyze historical usage patterns to identify a steady-state baseline capacity suitable for reservation purchases. Variable demand above baseline levels is provided through on-demand resources charged at standard rates. Hybrid approaches combining reserved baseline capacity with on-demand burst capacity optimize cost efficiency while accommodating workload variability.

Development environments tolerate interruptions that would prove unacceptable for production systems. Software developers can restart interrupted processes without a significant business impact. Testing activities accommodate occasional resource unavailability through automated retry

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

mechanisms. Development workloads are well-suited to spot instances, which purchase unused capacity at substantial discounts compared to on-demand pricing. Spot instances face potential interruption when cloud providers require capacity for higher-priority workloads.

Batch processing workloads exhibit time-flexible characteristics enabling execution during off-peak periods. End-of-day settlement processes are complete overnight when interactive workloads decrease substantially. Regulatory report generation schedules are established during weekend periods to avoid competition with business-hour operations. Data warehouse extract-transform-load jobs execute during low-activity windows.

Analytical queries vary dramatically in resource requirements based on data volumes scanned and computational complexity. Simple aggregation queries execute quickly, consuming minimal resources. Complex statistical models process large datasets, requiring substantial computing capacity for extended durations. Query workload unpredictability necessitates dynamic scaling capabilities that provision resources only during periods of query execution.

Temporal Resource Optimization

Usage-based segmentation extends beyond workload types to incorporate temporal dimensions reflecting time-varying demand patterns. Financial institutions experience predictable cyclical patterns in computational demand aligned with business operational rhythms. Trading platforms exhibit pronounced activity spikes during market opening hours with substantially reduced loads after market close. Customer service applications show higher utilization during business hours compared to evening and overnight periods. Month-end close processes create a surge in demand for batch processing capacity during the final business days of each month.

CPU resource optimization represents a critical component of temporal workload management strategies. Processor utilization has a direct impact on both application performance and infrastructure costs. Overprovisioned CPU capacity wastes financial resources on idle processing power. While configurations are underprovisioned, it results in overall performance bottlenecks that degrade the person's experience. Strategies for CPU optimization include right-sizing example types to match workload requirements, setting up vehicle-scaling guidelines that respond to utilization metrics, and consolidating workloads onto fewer instances during periods of low demand. Various optimization methods address CPU resource management challenges, including static provisioning approaches, dynamic allocation algorithms, and predictive scaling based on historical patterns [6].

Development environments automatically shut down outside standard business hours, thereby eliminating costs associated with idle resources. Engineering teams access development systems during working hours, utilizing automated startup procedures to provision the required capacity at the beginning of each shift. Weekend shutdowns further reduce development infrastructure costs during periods when software engineering activities cease. Batch processing jobs shift execution to overnight windows when interactive workloads decrease and compute resources cost less under time-of-day pricing models.

Analytics platforms implement query queueing mechanisms that defer non-urgent analyses to periods of low demand. Business intelligence dashboards display cached results for frequently accessed reports, avoiding redundant query execution. Ad-hoc analytical requests enter priority queues, with urgent queries receiving immediate execution, while exploratory analyses are deferred to off-peak execution windows. Query cost awareness becomes transparent to end users through estimated execution costs displayed before the query is submitted.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Table 2. Workload Classification and Temporal Optimization Strategies, Resource Allocation Approaches Across Financial Operations [5, 6].

Workload Type	Key Characteristics	Resource Strategy	Temporal Approach	Cost Technique
Production-Critical	High availability, consistent performance, real-time processing	Reserved capacity, long-term commitments, premium tiers	Market hours alignment, continuous operation	Reduced hourly rates, guaranteed performance
Development and Testing	Interruption tolerance, restart capability	Spot instances, unused capacity purchases	Business hours only, weekend shutdowns	Substantial discounts, idle elimination
Batch Processing	Time-flexible, scheduled execution	Off-peak execution, job scheduling	Overnight windows, month-end processing	Low-cost periods, workload consolidation
Analytical Queries	Variable demands, complex models	Dynamic scaling, auto-scaling policies	Query queueing, off-peak deferral	Priority execution, baseline maximization

Serverless Computing and AI-Driven Cost Optimization

Serverless architectures fundamentally alter cloud economics by eliminating the costs of idle resources. Traditional server-based deployments provision compute capacity continuously regardless of actual utilization patterns. Organizations pay for reserved capacity during periods of zero activity. Serverless models charge exclusively for actual execution time and resource consumption. Functions are invoked in response to specific triggering events and terminate immediately after completing processing tasks. This consumption-based billing eliminates costs associated with idle infrastructure waiting for incoming requests.

Function-as-a-Service platforms are the central implementation model for serverless computing in enterprise environments. Leading cloud providers deliver FaaS (Function as a Service) features that enable enterprises to set up event-driven code execution without needing to manage the underlying infrastructure. The platforms handle server provisioning, scaling, and maintenance operations, which are abstracted from application developers. Enterprise adoption of serverless computing faces several critical considerations, including vendor lock-in risks, debugging complexity, and performance monitoring challenges. FaaS platforms differ significantly across providers in terms of execution environment specifications, programming language support, and integration capabilities with other cloud services [7]. Financial institutions evaluate multiple dimensions when selecting serverless platforms, including cold start latency characteristics, maximum execution duration limits, and available memory configurations for function instances.

Event-driven functions respond to specific triggers such as data ingestion events, API requests, or scheduled activities. Functions consume resources only during active execution periods measured in milliseconds or seconds. The serverless model proves particularly effective for intermittent workloads, which are typical in financial services operations. Fraud detection algorithms execute when transaction authorization requests arrive at payment processing systems. Regulatory report generation functions trigger on scheduled intervals aligned with compliance filing deadlines.

Research Article

Customer notification systems activate when account status changes occur, requiring immediate communication.

Financial institutions are adopting serverless architectures for workloads that exhibit sporadic execution patterns with variable processing times. Account opening workflows invoke functions to validate customer information and perform credit checks. Risk assessment systems execute serverless functions, analyzing loan applications against underwriting criteria. Investment advisory platforms trigger portfolio rebalancing calculations when market conditions meet predefined thresholds. Each use case benefits from serverless economics, which aligns infrastructure costs directly with business activity volumes.

Intelligent Cost Anomaly Detection

Artificial intelligence enhances cost management by leveraging automated pattern recognition and anomaly detection capabilities. Machine learning models analyze historical cost data to identify standard spending patterns for various services, regions, and time periods. Organizations accumulate billing data, providing training datasets for anomaly detection algorithms. Models learn standard cost patterns accounting for daily usage cycles, weekly business rhythms, and seasonal demand variations. The systems generate alerts when spending deviates significantly from expected patterns based on learned baselines.

Cloud network anomaly detection has advanced substantially through the application of machine learning and deep learning techniques. Various algorithmic approaches address anomaly identification, including supervised learning methods, unsupervised clustering techniques, and semi-supervised hybrid models. Deep learning architectures demonstrate particular effectiveness for detecting complex patterns in high-dimensional cloud operational data. Convolutional neural networks extract spatial features from network traffic patterns. Recurrent neural networks, including Long Short-Term Memory models, capture temporal dependencies in sequential cost data. Autoencoder architectures learn compressed representations of normal behavior, enabling reconstruction-based anomaly detection [8]. Financial institutions benefit from these advanced techniques by detecting subtle cost anomalies that traditional rule-based systems would miss.

Cost anomalies indicate potential issues requiring immediate investigation and remediation. Resource misconfiguration errors result in unexpected expenses when provisioning parameters exceed the intended specifications. Unexpected traffic spikes generate surge costs if auto-scaling responds to attack traffic rather than legitimate user demand. Inefficient code deployments consume excessive resources, performing poorly optimized operations. Artificial intelligence systems identify anomalous spending patterns within hours of occurrence, enabling rapid investigation before costs accumulate significantly.

Natural language processing capabilities interpret unstructured cost data by extracting insights from service descriptions and resource tags. NLP algorithms identify services that consume disproportionate resources relative to the business value they deliver. Text analysis correlates cost anomalies with deployment logs and configuration changes. Organizations gain contextual understanding of cost patterns beyond numerical spending analysis.

Automated Optimization Recommendations

AI-powered platforms continually assess useful resource utilization metrics to pinpoint optimization opportunities. The platforms collect performance telemetry data, including CPU utilization, memory consumption, and network throughput, from various sources. The analysis engines associate the usage

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

metrics with the provisioned capacity specifications, thus they can identify discrepancies between the actual requirements and the resources that have been configured. Organizations receive automated recommendations that specify concrete actions to improve cost efficiency without compromising application performance. Rightsizing actions adjust compute instance specifications to match observed workload requirements. Analysis reveals that overprovisioned instances consistently operate at low utilization levels. Recommendations suggest using smaller instance types, which provide adequate capacity at reduced hourly rates. Conversely, systems detect undersized resources experiencing performance constraints and recommend larger instances. Rightsizing operates continuously as workload characteristics evolve over application lifecycles.

Alternative service configuration recommendations identify opportunities to substitute cost-effective services for expensive implementations. Analysis compares current service selections with alternative offerings that provide similar capabilities at different price points. Database workloads may benefit from migration to managed services, eliminating operational overhead. Storage systems storing infrequently accessed data can transition to archival tiers, reducing per-gigabyte costs.

Idle resource identification identifies provisioned infrastructure that consumes costs without delivering business value. Development instances running continuously outside working hours generate unnecessary expenses. Orphaned storage volumes persist after application decommissioning, accumulating charges indefinitely. Automated systems flag idle resources for review and potential termination based on activity monitoring over extended observation windows. Advanced implementations incorporate reinforcement learning to dynamically optimize resource allocation decisions. Reinforcement learning agents learn optimal policies through interaction with cloud environments. Agents receive rewards for cost reductions achieved while maintaining performance service level agreements. Dynamic optimization continuously adapts resource allocation, responding to changing workload patterns without manual intervention.

Table 3. Serverless Computing and AI-Driven Optimization Capabilities Cost Management Through Intelligent Automation [7, 8].

Technology	Key Features	Cost Benefits	Detection Capability
Function-as-a-Service	Event-driven, millisecond billing, automatic management	Idle cost elimination, consumption-based charging	Cold start analysis, environment provisioning
Machine Learning Detection	Pattern learning, baseline establishment, forecasting	Cycle recognition, deviation alerts	Configuration errors, traffic spikes, inefficiencies
Deep Learning Models	Neural networks, LSTM models, and autoencoders	Feature extraction, dependency capture	Pattern recognition, subtle anomaly identification
Natural Language Processing	Unstructured analysis, metadata extraction	Resource consumption tracking, value assessment	Cost correlation, configuration tracking

AI/ML Cost Governance for Financial Institutions

Artificial intelligence and machine learning workloads represent the fastest-growing category of cloud expenditure within financial services. Generative AI initiatives and large language model deployments consume computational resources at unprecedented scales. Financial institutions investing in AI-

Research Article

driven fraud detection and algorithmic trading systems face rapidly escalating infrastructure expenses. Graphics processing units and specialized accelerator hardware constitute the primary cost drivers. Training large-scale models requires sustained access to high-performance GPU clusters that command premium pricing. Inference workloads serving real-time predictions generate continuous demand for accelerator resources throughout operational hours.

Dedicated AI Cost Playbooks

Financial institutions must establish specialized FinOps playbooks addressing distinct cost characteristics of AI/ML workloads. Traditional cost allocation mechanisms designed for transactional systems prove inadequate for machine learning pipelines. AI cost playbooks define granular tracking requirements spanning model training experiments, hyperparameter optimization iterations, and inference serving infrastructure. Machine learning deployment presents unique challenges across the entire workflow lifecycle. Data management complexities, model training inefficiencies, and infrastructure monitoring gaps create hidden cost accumulation points. Organizations frequently underestimate expenses associated with feature engineering, model validation, and continuous retraining cycles [11]. Experiment tracking platforms must integrate with cost monitoring systems to provide unified visibility across model performance metrics and associated infrastructure expenditures.

Granular Cost Visibility for AI Workloads

Achieving meaningful cost visibility requires instrumentation that captures resource consumption at multiple granularity levels. Coarse-grained reporting aggregated at project levels proves insufficient for identifying optimization opportunities. Fine-grained tracking mechanisms attribute costs to individual training jobs, inference requests, and pipeline stages. GPU utilization monitoring provides critical insights for optimization initiatives. Accelerator resources frequently exhibit suboptimal utilization when workloads fail to fully leverage available computational capacity. Memory bandwidth constraints and inefficient batch sizing reduce effective GPU utilization below provisioned capacity levels. Advanced cost estimation methodologies leverage machine learning, deep learning, and hybrid models to predict infrastructure expenditure patterns. Regression-based approaches establish baseline cost relationships while neural network architectures capture complex nonlinear dependencies in resource consumption data [12]. Financial institutions benefit from predictive cost models that anticipate expenditure trajectories before budget overruns materialize.

Rightsizing and Scheduling for GPU Resources

Rightsizing strategies address the selection of appropriate accelerator configurations, balancing performance requirements against cost efficiency. Cloud providers offer diverse GPU instance types spanning entry-level accelerators through high-end configurations for large-scale training. Training workload scheduling optimizes resource utilization by coordinating experiment execution across available infrastructure capacity. Priority queuing systems ensure production model retraining receives guaranteed resource access. Preemptible GPU instances provide cost-effective capacity for fault-tolerant training jobs. Inference workload optimization addresses the continuous operational costs of serving deployed models. Auto-scaling configurations adjust endpoint capacity based on prediction request volumes. Model optimization techniques, including quantization and pruning, reduce computational requirements. Multi-tenancy strategies consolidate inference workloads across shared GPU infrastructure. Container orchestration platforms schedule inference containers across

Research Article

GPU clusters while maintaining workload isolation. Chargeback mechanisms allocate shared infrastructure costs proportionally based on inference request volumes and computational complexity metrics.

Sustainability and GreenOps Integration

Cloud cost optimization increasingly intersects with environmental sustainability objectives. GreenOps represents an emerging discipline integrating carbon footprint considerations into cloud financial operations. Financial institutions face mounting pressure from regulators and investors to disclose technology-related carbon emissions. Data centers consume substantial electricity for computing operations and cooling infrastructure. Cloud resource optimization decisions carry dual implications for cost efficiency and environmental impact. The convergence of FinOps and sustainability creates opportunities for aligned strategies reducing both expenses and carbon footprints.

Carbon-Aware Cloud Spending Strategies

Carbon-aware computing adjusts workload placement based on electricity grid carbon intensity. Different geographic regions exhibit varying emission profiles depending on local energy sources. Regions powered by renewable energy produce lower emissions per compute hour than fossil fuel regions. Designing carbon-aware datacenters requires holistic frameworks considering both embodied carbon from hardware manufacturing and operational carbon from electricity consumption. Carbon intensity varies significantly across time and location based on grid energy mix composition. Workload scheduling and geographic placement decisions directly influence total carbon footprint outcomes [13]. Financial institutions must balance carbon reduction objectives against performance requirements and data residency regulations.

Temporal carbon optimization schedules flexible workloads during lower grid intensity periods. Renewable energy availability fluctuates based on weather and time of day. Solar generation peaks during daylight hours. Wind generation varies with atmospheric conditions. Batch processing operations tolerate scheduling flexibility that interactive workloads cannot accommodate. Carbon-aware schedulers defer non-urgent computations to periods of optimal renewable availability.

Integrated Carbon and Cost Reporting

Mature GreenOps implementations incorporate carbon emissions alongside cost metrics in dashboards. Cloud providers now offer carbon footprint reporting tools estimating consumption-related emissions. Carbon data integration enables unified visibility across financial and environmental dimensions. Business units receive carbon attribution reports paralleling cost chargeback statements. Department leaders gain awareness of environmental impacts alongside infrastructure expenses.

Modern FinOps services address cloud cost optimization through automated analysis and recommendation engines. Cost optimization platforms ingest billing data from multiple cloud providers and apply analytical techniques to identify savings opportunities. Automated services detect underutilized resources, recommend rightsizing actions, and identify scheduling optimizations [14]. Integrating carbon metrics into such platforms extends optimization scope beyond financial considerations. Unified dashboards displaying cost and carbon data enable coordinated decision-making across both dimensions.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Carbon unit economics extend traditional cost metrics to environmental measurements. Organizations calculate emissions per transaction processed and per query executed. Carbon intensity metrics enable comparison across applications. Rightsizing recommendations incorporate carbon reduction benefits alongside cost savings. Instance consolidation reduces both expenses and emissions through improved utilization.

European financial institutions demonstrate particular advancement in GreenOps adoption. Regulatory frameworks and sustainability reporting requirements drive accelerated implementation. The European Union Corporate Sustainability Reporting Directive mandates detailed environmental disclosures. EMEA-based organizations integrate carbon tracking into governance frameworks for compliance. Regional initiatives accelerate collaboration between FinOps practitioners and sustainability teams. Sustainability considerations become standard components of cloud financial operations.

Cloud Data Warehouse Optimization and Chargeback Implementation

Cloud data warehouses introduce specialized cost considerations related to storage persistence, compute separation, and query optimization. These platforms charge separately for data storage and query processing capabilities. The architectural separation enables independent scaling of storage and compute resources. Organizations scale storage capacity without provisioning additional processing power. Conversely, query processing capacity increases during peak analytical periods without expanding storage footprints. This flexibility requires careful management to prevent cost inefficiencies.

Financial institutions store massive historical datasets supporting regulatory compliance, risk analytics, and customer behavior analysis. Data volumes continue to grow as transaction systems generate new records daily. Analytical requirements demand the retention of multi-year historical records, enabling trend evaluation and comparative reporting. The mixture of increasing information volumes and complex analytical workloads creates significant fee control challenges. Organizations require systematic approaches to optimize storage expenses while maintaining query performance characteristics.

Storage Tiering and Lifecycle Management

Implementing storage tiering strategies reduces costs by migrating data to progressively cheaper storage classes as access frequency decreases. Recently accessed datasets reside in premium storage optimized for query performance characteristics. Hot storage tiers provide low-latency access supporting interactive analytical workloads. Historical data moves to archival tiers with lower costs but higher access latency. Organizations design tiering policies based on data access patterns observed over extended monitoring periods.

Column-oriented database architectures remodel how data warehouses store and process analytical records. Traditional row-oriented databases save whole data sets contiguously on disk. Column-oriented systems store values from single columns together, enabling more effective compression. Values within columns exhibit higher similarity than values across complete records. This storage organization allows compression algorithms to achieve superior compression ratios. Column-oriented architectures integrate compression directly into query execution engines. The systems operate on compressed data without complete decompression steps. Query operators process compressed column values directly, reducing memory bandwidth requirements and accelerating execution [9]. Financial

institutions benefit from reduced storage costs through higher compression ratios while maintaining or improving query performance characteristics.

Data lifecycle management encompasses multiple operational dimensions, including enforcement of retention policies, compliance archival, and deletion of obsolete data. Regulatory requirements mandate retention of financial transaction records for specified periods varying by jurisdiction. Automated lifecycle systems archive data approaching end-of-retention periods to lower-cost storage tiers. The systems delete obsolete datasets after retention periods expire. Compliance schedules dictate archival timing, ensuring that regulatory data remains accessible during the required retention windows.

Storage optimization extends beyond tiering and compression to encompass data modeling decisions affecting storage efficiency. Normalization reduces redundant data storage across related tables. Denormalization accepts storage redundancy, thereby improving query performance by eliminating the need for join operations. Organizations balance the benefits of normalization against query performance requirements. Partitioning strategies divide large tables into smaller segments based on date ranges or categorical dimensions.

Query Optimization and Cost Attribution

Query performance directly impacts warehouse costs, as inefficient queries consume excessive compute resources and process unnecessary data volumes. Poorly constructed queries scan entire tables when selective predicates could limit the scope of processing. Missing indexes force full table scans for point lookups that should execute through index seeks. Query optimization represents a critical cost control mechanism for organizations operating cloud data warehouses.

Modern cloud data warehouses require comprehensive optimization strategies addressing multiple performance dimensions. Query optimization techniques include predicate pushdown, join reordering, and aggregation optimization. Predicate pushdown evaluates filter conditions early in the execution plan, reducing the data volumes processed by downstream operators. Join reordering sequences join operations to minimize intermediate result sizes. Aggregation optimization precomputes summary statistics, avoiding repeated calculations. Indexing strategies accelerate data retrieval for selective queries. Materialized views cache frequently accessed aggregations, eliminating redundant computations. Partition elimination leverages table partitioning schemes to exclude irrelevant data segments from query scans [10].

Query result caching eliminates redundant computations for frequently accessed analyses, thereby improving performance. Dashboards displaying standard reports execute identical queries repeatedly throughout business days. Caching systems store query results in memory, serving subsequent identical queries without recomputation. Cache hit rates depend on query repetition patterns and the allocation of cache memory. Organizations configure cache expiration policies, balancing data freshness requirements against computational savings.

Query monitoring systems identify expensive operations enabling targeted optimization efforts. The systems capture query execution statistics including elapsed time, rows processed, and compute resources consumed. Cost attribution associates specific monetary amounts with individual queries based on resource consumption metrics. Organizations establish query cost budgets for different user groups or business units. Query optimization focuses on the highest-cost operations, providing the maximum return on optimization investment.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Chargeback Model Implementation

Chargeback models establish financial accountability by allocating actual cloud costs to the business units that consume them. Implementation requires robust cost collection mechanisms that capture resource usage at query, user, and project levels of granularity. Data warehouses log all query executions, along with associated metadata that identifies the submitting users, originating applications, and business context tags. Cost calculation engines process execution logs, computing monetary costs for individual queries based on resource consumption metrics.

Automated reporting distributes itemized cost statements to department leaders, creating visibility into technology expenses. Monthly reports itemize charges by user, application, and cost category. Trend analysis highlights consumption patterns and identifies cost anomalies warranting investigation. The transparency incentivizes efficient resource consumption as business unit leaders observe direct financial impacts of analytical activities.

Showback approaches provide cost transparency without formal billing mechanisms. Organizations display cost information to business units without transferring budget responsibility. Showback suits organizations beginning cloud cost governance journeys. Complete chargeback transfers budget responsibility to business units, aligning technology spending decisions with cost accountability. Business units receive budget allocations that cover anticipated cloud consumption, with actual usage charges against allocated budgets creating financial incentives for optimization.

Table 4. Cloud Data Warehouse Optimization and Chargeback Mechanisms, Storage Management and Query Performance Strategies [9, 10].

Optimization Area	Implementation	Performance Impact	Cost Benefit
Storage Tiering	Hot and archival tiers, lifecycle policies	Low-latency access, compliance adherence	Lower archival costs, obsolete deletion
Column-Oriented Storage	Dictionary encoding, columnar organization	Execution acceleration, bandwidth reduction	Superior compression, footprint reduction
Query Optimization	Predicate pushdown, materialized views, caching	Early filtering, precomputation, result reuse	Redundant elimination, cache hit improvement
Chargeback Models	Query-level tracking, usage capture, cost logging	Execution analysis, consumption computation	Budget responsibility, financial accountability

Conclusion

Effective cloud cost optimization in financial institutions requires comprehensive strategies that integrate organizational governance, architectural decisions, and intelligent automation capabilities. Financial Operations frameworks establish foundational cost visibility mechanisms enabling accurate expense attribution across complex organizational hierarchies. Hierarchical tagging taxonomies capture business unit identifiers and application classifications supporting multidimensional cost reporting. Unit economics frameworks translate infrastructure expenses into business-relevant metrics, facilitating meaningful conversations between technology teams and department leaders. Usage-based workload segmentation enables targeted optimization approaches reflecting distinct performance requirements and criticality levels across application portfolios. Multi-cloud resource

allocation algorithms optimize workload placement considering cost structures, performance capabilities, and regulatory compliance constraints simultaneously. Temporal optimization techniques leverage predictable demand patterns, aligning computational resources with value-powerful capacity availability. Serverless architectures transform cloud economics by eliminating idle resource expenses through event-driven execution models.—feature-as-a-provider structures abstract infrastructure control duties, allowing builders to focus on enterprise exemplary judgment implementation. Artificial intelligence enhances cost management by automating pattern recognition, which enables the identification of spending anomalies within hours of their occurrence. Machine learning models establish baseline spending patterns across services, regions, and time periods, generating alerts when consumption deviates significantly from expected norms. Deep learning architectures detect complex patterns in high-dimensional operational data, employing techniques such as convolutional neural networks and recurrent neural networks. Cloud data warehouse optimization addresses specialized cost considerations related to storage persistence and query processing separation. Column-oriented database architectures achieve superior compression ratios by storing values from single columns together. Query optimization techniques, including predicate pushdown, materialized views, and partition elimination, reduce computational expenses while maintaining analytical performance characteristics. Comprehensive chargeback models establish financial accountability, allocating actual infrastructure costs to consuming business units through granular resource usage tracking. Automated reporting distributes itemized cost statements, creating visibility into technology expenses and incentivizing efficient resource consumption behaviors. Financial institutions that adopt integrated optimization strategies gain competitive advantages through reduced infrastructure expenses, improved resource utilization efficiency, and an enhanced ability to allocate technology investments toward revenue-generating activities.

References

[1] Akif Quddus Khan et al., "Cloud storage cost: a taxonomy and survey," Springer, 2024. [Online]. Available: https://link.springer.com/content/pdf/10.1007/s11280-024-01273-4.pdf

[2] Parag Bhardwaj, "The Role of FinOps in Large-Scale Cloud Cost Optimization," International Journal of Scientific Research in Engineering and Management, 2024. [Online]. Available: https://www.researchgate.net/profile/Parag-Bhardwaj-3/publication/387983179_The_Role_of_FinOps_in_Large-Scale_Cloud_Cost_Optimization/links/6792bb4d4c479b26c9b08d45/The-Role-of-FinOps-in-Large-Scale-Cloud-Cost-Optimization.pdf

[3] Adedamola Abiodun Solanke, "AI-Enhanced FinOps: Predictive Cost Optimization Across AWS, Azure, and GCP," International Journal of Current Science, 2025. [Online]. Available: https://www.researchgate.net/profile/Adedamola-Solanke/publication/390427521_AI-Enhanced_FinOps_Predictive_Cost_Optimization_Across_AWS_Azure_and_GCP/links/67ed5d9795231d5ba5acfdb3/AI-Enhanced-FinOps-Predictive-Cost-Optimization-Across-AWS-Azure-and-GCP.pdf

[4] Vignesh Kuppa Amarnath, "Engaging Techniques for Effective Resource Allocation in Cloud Computing: Improving Performance, Availability, and Cost Management," International Journal on Science and Technology, 2025. [Online]. Available: https://www.ijsat.org/papers/2025/1/2975.pdf

[5] Deepak Kaul, "Optimizing Resource Allocation in Multi-Cloud Environments with Artificial Intelligence: Balancing Cost, Performance, and Security," JICET, 2019. [Online]. Available: https://www.researchgate.net/profile/Deepak-Kaul-

3/publication/394015247_Optimizing_Resource_Allocation_in_Multi-Cloud_Environments_with_Artificial_Optimizing_Resource_Allocation_in_Multi-Cloud_Environments_with_Artificial_Intelligence_Balancing_Cost_Performance_and_Securi/links/6884214496f3c0122ef40f36/Optimizing-Resource-Allocation-in-Multi-Cloud-Environments-with-Artificial-Optimizing-Resource-Allocation-in-Multi-Cloud-Environments-with-Artificial-Intelligence-Balancing-Cost-Performance-and-Secu.pdf

[6] Vivek Saxena et al., "A Review of Cloud Computing CPU Resource Optimization: Methods, Difficulties, and Prospects," ACM, 2024. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3745812.3745839

[7] Theo Lynn et al., "A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms," IEEE 9th International Conference on Cloud Computing Technology and Science, 2017. [Online]. Available: https://www.researchgate.net/profile/Pierangelo-Rosati/publication/321753133_A_Preliminary_Review_of_Enterprise_Serverless_Cloud_Computing_Function-as-a-Service_Platforms/links/5a2feb7d458515a13d851ec0/A-Preliminary-Review-of-Enterprise-Serverless-Cloud-Computing-Function-as-a-Service-Platforms.pdf

[8] AMIRA MAHAMAT ABDALLAH et al., "Cloud Network Anomaly Detection Using Machine and Deep Learning Techniques— Recent Research Advancements," IEEE Access, 2024. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10504797

[9] Daniel J. Abadi et al., "Integrating Compression and Execution in Column-Oriented Database Systems," ACM, 2006. [Online]. Available: https://www.cs.umd.edu/~abadi/papers/abadisigmod06.pdf

[10] PUNIT GOEL and ER. OM GOEL, "Optimizing Modern Cloud Data Warehousing Solutions: Techniques and Strategies," IJNRD, 2023. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/118855269/IJNRD2303501-libre.pdf?1728806241=&response-content-disposition=inline%3B+filename%3DOptimizing_Modern_Cloud_Data_Warehousing.pdf&Expires=1763444270&Signature=FbP2T9dY0xZ0oOdR3I0SL-Uju~Z~HEmlHPF1wnX2qpST2kmks5gmZ~6Fw0jCSvZvMCUt064a52I8Rw9OTtnxgtL6cVj5xV4iWvBXXkh-ub5J53m5cdJDPBoyzY6hxH9m2TgA2yh-XsnYEuEf73D6voLXjejutu3NzoC9sO-4lLD5voZ~MXZzUY0y0MzD9B5cVf6ev5wixyKGdg8Ew8w963VXxQ3Dk9KKbc2EfYcvKsCx404wTPr~Wfk1XUMftbbBq0FnyUMR~JiQv7Gr-zKivEhkuI7M6~KbQ7-rKHAmogmcjl5~RrNISNorbgEhv-x1rxzu3QjmRhbj5cMXi51moA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[11] ANDREI PALEYES et al., "Challenges in Deploying Machine Learning: A Survey of Case Studies," ACM Computing Surveys, 2022. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3533378

[12] Md. Mahfuzul Islam Shamim et al., "Advancement of Artificial Intelligence in Cost Estimation for Project Management Success: A Systematic Review of Machine Learning, Deep Learning, Regression, and Hybrid Models," MDPI, 2025. [Online]. Available: https://www.mdpi.com/2673-3951/6/2/35

[13] Bilge Acun et al., "Carbon Explorer: A Holistic Framework for Designing Carbon Aware Datacenters," arXiv, 2023. [Online]. Available: https://arxiv.org/pdf/2201.10036

[14] Saurabh Deochake, "ABACUS: A FinOps Service for Cloud Cost Optimization," arXiv, 2024. [Online]. Available: https://arxiv.org/pdf/2501.14753?