International Journal of Research in Computer Science & Information Technology (IJRCSIT)
ISSN: XXXX-XXXX/ Volume 1 Issue 2 November – December 2026
Paper Id: IJRCSIT-V1I2P106/ Page No: 90-104

International Journal of Research in Computer Science & Information Technology

Original Article

Reducing Unnecessary Cloud Resource Allocation through Workload-Aware Scaling Decisions

Kavin[1], Banu Priya[2], Prabha[3]

[1,2,3] De Britto's University of Technology, India.

Received Date: 28 November 2026 Revised Date: 02 December 2026 Accepted Date: 06 December 2026

Abstract

Today, Cloud Computing has become the foundation of today's digital world. It allows companies to deploy applications, store huge amounts of data and offer services based on a flexible, on-demand computing model. Unfortunately, today there is still a problem in effectively managing the allocation of Cloud Computing resources. Most companies use traditional ways of scaling their Cloud Computing resources. Some examples of these traditional ways are setting maximum limits for the use of Cloud Computing resources, reviewing historical use of Cloud Computing resources and adjusting to changes in the amount of work being done by the company. There are several problems that could arise from using traditional ways of managing Cloud Computing resources. For instance, you could over-allocate Cloud Computing resources. Also, you could spend too much money on Cloud Computing equipment. Moreover, you could unnecessarily use too much energy. Lastly, you could under-allocate Cloud Computing resources. This paper presents a workload-aware scaling model for managing Cloud Computing resources so that you do not waste money on Cloud Computing resources. The model helps determine whether to allocate, keep, scale up/down, or de-allocate Cloud Computing resources based on how much work is being done, how the use of Cloud Computing resources changes over time, what performance levels the application needs to achieve and how often people want to access the application. It is important to remember that while we want to save money on Cloud Computing resources, we also need to meet service-level objectives, availability and responsiveness. We also propose two ways to maximize the use of Cloud Computing resources and avoid bad scaling decisions. One way is predictive analysis of workload and the other way is adaptive decision making. We create a conceptual evaluation method using four performance metrics. The metrics include; resource allocation efficiency, infrastructure costs, response times, and frequency of scaling events. Instead of saying that we have proven empirically that our model performs better than existing models, we present a research-based model that can be evaluated using either cloud simulation or a controlled deployment experiment. We believe that our results will help develop more cost-effective, responsive, and environmentally-responsible ways of managing Cloud Computing resources.

Keywords

Cloud Computing, Workload-Aware Scaling, Resource Allocation, Auto-Scaling, Cloud Optimization, Predictive Analytics, Resource Utilization, Elastic Infrastructure, Cost Efficiency, Performance Management

INTRODUCTION

The way organizations develop, deploy, and maintain digital services has changed dramatically with the introduction of cloud computing. Organizations no longer need to buy and maintain large-scale physical infrastructure. Instead, they can lease the amount of computing power, storage, networking, and application platforms they need for their operations. Organizations can scale up or down as needed based on demand. This provides the flexibility to support many types of applications including online shopping, financial transactions, healthcare platforms, education systems, enterprise software and data-heavy services. The economic and operational benefits provided by cloud computing will be determined primarily by how efficiently resources are allocated and managed. Many organizations will configure their applications to include extra computing capacity in order to handle potential peaks in demand, avoid performance issues and ensure high levels of service availability. Although having some excess capacity may be required, over-allocating resources can lead to virtual machines, containers, memory, processing units, etc., sitting idle

This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/2.0/)

for long periods of time. Most traditional methods used to manage resources based on fixed utilization thresholds or pre-defined scaling rules often fail to take into account the actual usage patterns of the application. An example of this is when an application experiences a temporary spike in CPU utilization causing the system to automatically deploy additional instances, despite the fact that the workload is likely to return to normal shortly thereafter. Another example is when resources allocated at the height of a previous demand period continue to run after traffic levels decrease. Both examples create avoidable expenditures on infrastructure, waste resources and require greater amounts of energy. Research on cloud autoscaling has identified resource costs, service quality, scaling responsiveness and workload uncertainty as major factors to consider when developing effective scaling strategies.

Consequently, in order to minimize the number of unused cloud resources allocated, an organization should have a management system in place that looks beyond just the amount of resources being used at any given time, but rather at the workload type, how long the workload persists, what types of applications run on the workload, and what will probably happen to the resource usage in the future. A workload-aware scaling mechanism could potentially serve as a way to address these issues. This is because a workload-aware scaling mechanism uses the knowledge of the workload's behavior to make decisions regarding whether or not to add resources (provision), keep existing resources (retain), change the size of resources (resize), or remove resources (remove). Unlike a purely reactive mechanism that only reacts to the utilization of resources after a certain threshold is exceeded, a workload-aware scaling mechanism evaluates several different variables. These include: incoming request rate, CPU/Memory consumption, request processing time, queue length, workload periodicity, and past demand patterns. By evaluating each variable, a workload-aware scaling mechanism can determine whether the increase in demand is a sustained increase or if it is merely a short-term fluctuation that does not warrant additional infrastructure. For example, if there is a sudden increase in requests that threatens the application response time, then additional infrastructure should probably be added. On the other hand, if there is a sudden increase in utilization of a non-critical background process, then no additional infrastructure needs to be added. Predictive workload analysis can also help organizations make better decisions by predicting future demand and determining when current resources will likely be sufficient. Statistical forecasting, Machine Learning, and Adaptive Control Mechanisms can all aid in making predictions, however, the effectiveness of these tools depends upon how predictable the workload is, how accurate the data is that was collected for analysis, how large the forecasting error is, and what type of operational environment exists.

In spite of this, however, predictive scaling by itself will not ensure that there is less resource consumption. If the forecast was wrong then the system could be over provisioned too early; it could take longer than expected to scale in; or there might not be enough capacity when an unexpected increase in demand occurs. Therefore, workload-aware scaling should include demand estimation as well as resource utilization constraints, application performance targets, minimum capacity requirements and well-defined scale-in and scale-out criteria. Additionally, workload-aware scaling should consider instance startup delays, cooldown periods, scaling costs and the possibility of having multiple scaling events due to quickly changing workloads. By considering all of these aspects, a cloud resource management system can determine more accurately when more capacity is needed and when current resources are sufficient to meet application demands without additional allocations.

The primary purpose of this research is to create a workload aware scaling framework that minimizes the amount of wasted cloud resources allocated to applications while still meeting acceptable levels of application performance, availability and responsiveness. The proposed architecture includes several phases including workload monitoring, characterization of demand, short-term forecasting, evaluation of scaling decisions, adjustment of resources, and on-going performance feedback. In order to begin the process, operational measurements will be collected so that the current status of applications and the underlying infrastructure can be established. After establishing the current status, workload characteristics will be assessed in order to determine if demand variations are short-lived, repetitive, or long-lasting. Once the assessment has been made, the scaling decision making process will evaluate the estimated resource needs for applications against the existing capacity and the pre-defined service-level objective (SLO) criteria. Based upon the results of this evaluation, the scaling decision making process will recommend provisioning, retention, resizing or release of resources. Additionally, the framework incorporates safeguards to prevent pre-mature scaling-in of resources and over-scaling out of resources to ensure that reduction of resources does not impact service continuity in times of unanticipated increases in workload.

The performance of the proposed strategy could be evaluated using controlled simulation experiments or via cloud deployment experiments using real-world workload data. Performance evaluation criteria include total number of resource hours utilized, average resource utilization per hour, total cost of operating the infrastructure, response time, service level agreement (SLA) violations, frequency of scaling operations, and the amount of unutilized allocated

capacity. Such metrics will allow for direct comparison between the traditional threshold-based autoscaling approach and the proposed workload-aware approach for the same workloads. This paper's primary contribution is to integrate the various aspects of workload characteristics, predictive knowledge, and performance constraints into a single decision-making process instead of assuming that a higher utilization rate equals a superior resource allocation strategy. The objective of the framework is to facilitate cost-effective cloud operation, increase efficiency of cloud infrastructure, and minimize wastefully consumed computing resources while maintaining reliability. Limitations of the proposed framework, such as forecasting uncertainty, diversity of application requirements, monitoring overhead, and unexpected spikes in workload must also be evaluated. In general, the goal of this paper is to provide a solid basis for developing more dynamic cloud resource management approaches in which scaling decisions are based upon current and expected workload requirements as opposed to unnecessarily provisioning or rigid allocation policies

WORKLOAD-AWARE CLOUD RESOURCE MANAGEMENT FRAMEWORK

A. Workload Characterization and Monitoring

The way cloud applications utilize computing resources (i.e., CPU, memory, etc.) can vary depending on a number of factors; for example, some applications will have a constant level of demand, others will have periodic fluctuations in demand, and still others will have rapid increases in demand (spikes), and/or unpredictable changes in demand due to user behaviors and/or business activities. Therefore, continuously collecting relevant operational data, such as CPU utilization, memory consumption, number of incoming requests, network throughput, queue lengths and application response times, is essential to provide insight into current resource utilization levels and determine if the allocated capacity is adequate to meet current processing demands. In addition, collecting historical monitoring data can assist in identifying patterns of recurring demand, peak periods of demand and extended periods of under-utilization. Combining the real-time observation of application behavior with historical workload data allows the proposed framework to differentiate between sustained increases in demand and temporary fluctuations in demand. This differentiation is critical since making decisions regarding resource allocation based on each temporary spike in resource utilization can result in excessive expenditures on infrastructure. Ultimately, effective workload monitoring provides the basis for making rational provisioning decisions, improved capacity planning and more efficient utilization of cloud resources.

B. Predictive Analysis for Resource Demand Estimation

Predictive analysis can be used to determine future resource needs based upon the historical use of resources, rate of incoming requests, time required for each request, and patterns in how the demand for the resource changes over time. Statistical forecasting methods and/or machine learning can be applied to these variables to make predictions about future resource demands. Predictive analysis can help identify if future demands will either grow, shrink or stay about the same. As an example, a business application that experiences increased demand at specific times of day would benefit from being able to prepare the necessary resources ahead of time. Conversely, a service that consistently experiences low levels of traffic would likely not need to add more instances. The proposed framework uses demand estimates as a form of support to decision making; they are not considered a certainty. Demand uncertainty should be taken into account when making the final decision on whether to scale up or down. Uncertainty can lead to more conservative decisions about how much to adjust resources and what minimum amount of capacity to maintain in order to ensure application availability. Therefore, predictive analysis can minimize over-provisioning of resources by recognizing times when the current resources are likely to be sufficient. To evaluate the effectiveness of predictive analysis, it should be compared against baseline reactive scaling strategies using similar workloads and metrics to measure performance consistently.

C. Adaptive Scaling Decision and Resource Optimization

Adaptive scaling decision-making is about whether to provide, keep, change the size of, or release cloud resources based upon what is happening with the current workload and what we think will happen in the future. Rather than making all decisions based on a single CPU utilization metric, the adaptive scaling framework that I propose considers many metrics including processing demands, memory pressure, response time targets, workload persistence, and available capacity. If there is going to be high demand for an extended amount of time and this could hurt performance, or if we expect that we are going to run out of capacity shortly, then a scale out decision would probably be correct. On the other hand, if we have been using very little resource for some amount of time and we still have enough capacity left over to take care of what we expect to happen, then a scale in decision might be appropriate. There are things called cool down periods, minimum number of instances, and hysteresis levels which help avoid continually scaling up/down because of small changes in the workload. Before implementing any changes, the adaptive scaling framework should

consider other dependencies between resources, how long it takes for new instances to start up, and any other limitations that exist within the application. When all of these factors are taken into consideration, adaptive scaling tries to limit the amount of unused capacity we have while maintaining the reliability of our system. All adaptive scaling decisions should be monitored continually against how well they worked out for our workload so that we can modify our scaling policies as needed as our resource needs or application behavior changes.

D. Performance Evaluation and Cost-Efficiency Assessment

Performance assessment will help evaluate if the proposed workload-aware scaling approach will decrease the need for over-provisioning resources, while maintaining acceptable levels of delay and reliability. A comparative study can be conducted through either controlled laboratory simulation on cloud platforms, or via field deployments of the proposed workload-aware scaling approach against a conventional threshold-based auto-scaling approach with the same workload. The measurement of resource consumption can include the total number of hours each virtual machine was provisioned; the amount of container usage; the level of utilization of CPU and Memory; and the amount of time that resources were unutilized. An assessment of financial efficiency can be based on the cost of operating the infrastructure during a fixed timeframe. Additionally, an assessment of application performance can be based on average response times; tail latencies; throughputs; and service level objective violations. In addition, an assessment of operational stability can be based on how frequently scaling occurred; and how many unnecessary provisioning/termination events occurred. Data should be collected under conditions where workloads are stable; periodically vary; have sudden increases in traffic; and have random requests. Multiple iterations of testing; and clearly stated methods for collecting data can increase the reliability of the comparisons being made. Since the proposed workload-aware scaling approach has not yet been experimentally verified, any claims regarding reduced costs/resource consumption must be considered hypotheses until they are confirmed through measurements.

IMPLEMENTATION STRATEGIES FOR WORKLOAD-AWARE SCALING DECISIONS

A. Integration of Workload Prediction and Intelligent Scaling Policies

A workload-aware scaling solution needs to be built around an integrated framework which brings together workload monitoring, demand prediction, resource allocation, and application performance management. The proposed solution starts by gathering operational metrics from cloud based applications, VMs, containers, and orchestration platforms. Operational metrics collected include CPU usage, memory consumption, request arrival rate, queue length, response time, and historical scaling actions. Once operational metrics have been collected, they are used to recognize workload patterns and determine if resource consumption is growing, shrinking, or staying the same. Then, either statistical forecasting methods or ML models are utilized to forecast short term demand and support scaling decisions. If historical data indicates a predictable increase in application requests, the system can add capacity prior to the expected workload peak. Conversely, if there is a short lived increase in CPU usage without an accompanying increase in requests or response times, the system may delay adding capacity until there is additional evidence supporting a long term need for additional resources. By combining both assessments, reliance upon single utilization thresholds is decreased and decision making is encouraged based on how the entire application behaves. Additionally, the solution must take into account prediction confidence and the computational overhead of continually monitoring and forecasting.

The scaling policy translates the observations made above into operational steps that meet the performance and availability specifications. A scale-out event can be triggered if there is anticipated demand for more capacity than is currently available, or if the performance of the application begins to fall below acceptable standards. On the other hand, a scale-in event can only be performed once there has been sufficient time for the identification of excess capacity, taking into account any minimum instance requirements, workload variability, and the amount of time required to bring back additional capacity if demand were to rise unexpectedly. To help avoid frequent changes in response to small fluctuations in utilization near a decision point, cooldown periods and hysteresis mechanisms can be used. Application-specific policies need to differentiate between latency sensitive services and background processes due to the differences in how each type of service tolerates shortages in resources. A feedback loop can be established to compare the predicted demand against what actually occurs; if there are consistently large discrepancies, then the decision-making threshold or forecasting parameter can be adjusted accordingly. In essence, workload prediction and intelligent scaling policies are two separate but interrelated components of a larger system. Together, they create a logical framework for managing the provisioning of resources, preventing premature resource removal, and providing consistent cloud infrastructure management across various types of workloads.

B. Operational Challenges, Reliability, and Resource Efficiency

The implementation of workload-aware scaling as a systematic approach to enhancing resource utilization creates a number of technical and operational issues which must be managed appropriately. A major issue is workload variability, especially when an application experiences unexpected spikes in traffic as a result of unplanned events, promotions, or changes in user behaviors. Past data may not reflect such circumstances, therefore leading to incorrect predictions of required resources and inadequate provision of resources. Quality of monitoring will also impact the quality of decisions made based upon metrics that lag behind actual performance, missing observations, and variable intervals of measurement that distort measurements of actual demand. There are many different types of applications running in the cloud, each having unique demands for resources. Thus, a scaling policy that works well for a CPU-intensive application may not work as effectively for a memory-intensive or network-intensive service. Therefore, the framework used for scaling should take into account all relevant resource dimensions rather than focusing on only one measure of utilization. Other factors that need to be considered are the time needed to bring online new instances; the overhead associated with initializing applications; the cost of transferring data; and the interdependencies among interconnected services. If these factors are not incorporated into the decision making process, then the benefit of scaling decisions will be limited. Successful implementation of workload-aware scaling requires robust monitoring systems; specific policies tailored to individual workloads; and safeguards that ensure continuity of service when uncertainty exists.

In addition to reliability, operational costs, and application responsiveness, resource efficiency must be assessed in conjunction with resource reductions to evaluate the value of the reduction. A system that has minimized allocated capacity but consistently exceeds the response time objectives is not successful, regardless of the decrease in infrastructure costs. Therefore, evaluation should include comparison of workload-aware scaling versus a conventional baseline with identical workload conditions, deployment configuration, and observation period. Metrics that may be used to assess the results of the evaluation include resource-hours utilized; average utilization; infrastructure costs; response-time percentiles; throughput; frequency of scaling events; and number of service level agreement (SLA) violations. Evaluation must include predictable workloads, extended periods of low demand, irregular traffic patterns and rapid increases in request volume. Sensitivity analysis can provide additional insights into how changes in prediction windows; scaling thresholds; cooldown intervals; and minimum capacity settings affect performance. In addition, the implementation must consider the computational overhead associated with monitoring and predicting so that the computational overhead does not exceed the resource savings generated by the intelligent decision-making process. From an operations standpoint, phased deployment; validation of policies; and automatic fallback to established scaling rules will help mitigate risk during implementation. Using a systematic approach to evaluating the effectiveness of workload-aware scaling, organizations can determine if there are quantifiable improvements in both resource utilization and cost-efficiency while maintaining acceptable levels of performance and availability for their cloud-based applications.

PROPOSED WORKLOAD-AWARE SCALING ARCHITECTURE AND OPTIMIZATION MECHANISM

This paper presents a new workload-aware scaling architecture that is designed to reduce the amount of extra cloud resources allocated as a result of automatic scaling based on thresholds. This architecture integrates several different components of the scaling process, including continuous monitoring, workload analysis, estimating resource requirements, and making adaptive scaling decisions, into one centralized management system. At its core, the architecture uses a monitoring component to collect real-time data regarding CPU usage, RAM consumption, incoming request rates, network activity levels, queue lengths, and application response times. This collection of data provides an overall picture of how well a cloud-hosted application is currently performing. Once the data has been collected, it undergoes processing to eliminate inconsistent data points, determine when there have been significant changes in workload, and create a solid basis for future analyses. Additionally, historical data is retained to assist in identifying recurring usage patterns, times when there is little to no demand for resources, and when peak workloads can be predicted. In contrast to most existing scaling strategies, which tend to react to a single threshold violation, the proposed architecture takes multiple variables into consideration prior to determining if additional resources are actually needed. As such, the architecture is particularly suitable for applications experiencing non-uniform traffic patterns or experiencing short-term spikes in utilization. Through the integration of monitoring information with characteristics of the workload, the architecture attempts to differentiate between sustained resource pressure and brief periods of high utilization; thus, minimizing the probability of providing unnecessary instances of virtual machines and using resources inefficiently.

The next part of the architecture will deal with predicting workload demands and estimating how much resource each application needs. We will look at past performance data, as well as what the application is doing right now to determine how much computer power the application will require for the next observation time frame. There are many ways to make these estimates depending on how hard the application is to predict, what kind of data we have access to, and how much computing power we have. If we know that the application behaves predictably (i.e., its behavior hasn't changed over time), we can probably just use the past trend as an estimate. However, if the application is more complex, we can use statistical forecasting methods, regression analysis, or machine learning algorithms to find the nonlinear relationships between the different parts of the application and the amount of resources they consume. After making the estimate, we will compare it with the amount of resources we currently have available, the performance standards required by the application, and any pre-defined limits on the amount of resources the application can consume. We will also include an assessment of how certain we are in our prediction.

This is important because if our predictions are off, we could either over-provision our applications or run out of resources completely. If we are not very sure about our predictions, we can be more cautious in our scaling decisions and leave ourselves some extra buffer space for the most critical applications. Our goal isn't necessarily to eliminate every last bit of unused resources from our systems, as there will always be a need for some level of redundancy for reliability purposes. However, we want to avoid allocating additional resources unnecessarily when doing so won't result in any noticeable improvement in application performance or availability.In addition to implementing an adaptive decision engine that determines when to add, retain, resize, or remove resources, the third component also incorporates a decision engine that supports both scale-out and scale-in actions. An action of scale-out will occur if existing capacity does not support projected demand or if performance of applications approaches pre-defined service level thresholds. On the other hand, an action of scale-in will occur if the utilization of resources is consistently below predetermined thresholds and there is sufficient available capacity to meet future expectations of workload.

To prevent instability in the scaling of resources, the decision engine may incorporate various parameters such as; minimum and maximum numbers of instances, cooldown periods, hysteresis thresholds, and application-specific constraints. For example, if there was a short-term decrease in request volume, this decrease would not necessarily lead to the termination of resources if historical data indicates that demand is expected to return soon. Similarly, if there was an unexpected surge in traffic, appropriate protective measures would be taken to protect application response times. Furthermore, the framework allows for prioritization of critical workloads. Therefore, critical services will have enough capacity to handle the workload, whereas less demanding workloads can be scheduled more efficiently. After every scaling action, the system will assess actual resource consumption and application performance to evaluate whether the scaling action resulted in the desired outcome. This feedback loop enables refinement of prediction models, adjustments to decision thresholds, and improvement of consistency in scaling as workload characteristics change.The last section will assess the effectiveness of the proposed architecture by performing systematic performance and cost assessments. The architecture can be tested in either a simulated cloud environment or a real world controlled deployment, with applications that represent actual workloads, on comparable hardware platforms. Threshold-based autoscaling, which is commonly used today, can provide a basis for comparison with decision-making based on workload. Performance measures should include: the total number of resource hours utilized; the average amount of CPU and memory used; the total cost of the infrastructure; application response times; application throughputs; how often scaling occurs; and any violations of the service level agreements.

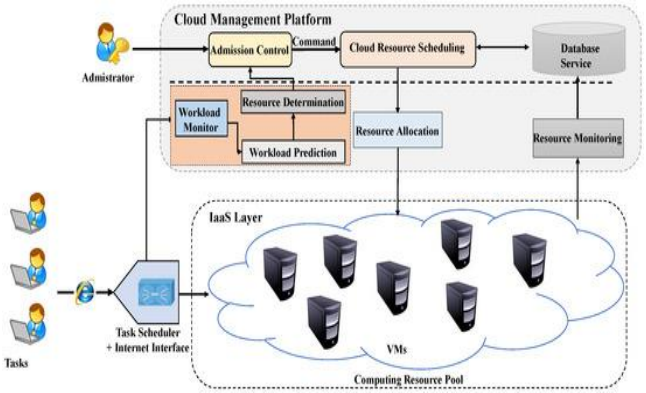


Figure 1: Proposed Workload-Aware Auto-Scaling Architecture in a Cloud Management Platform

COMPARATIVE ANALYSIS OF WORKLOAD-AWARE SCALING STRATEGIES

Cloud resource optimization can be achieved through a systematic evaluation of various scaling mechanisms (e.g., traditional threshold-based autoscaling versus workload-aware scaling) and how they apply to specific types of workloads. For example, traditional threshold-based autoscaling provides an easy way to scale computing resources based on user defined thresholds for resource usage. However, because traditional threshold-based autoscaling relies solely upon current resource usage to determine when to scale up or down, it can make false assumptions about what constitutes a short term spike in demand vs. a longer term trend. Therefore, workload-aware scaling is used as an alternative approach that evaluates several other factors such as historical trends in demand, estimates for future demands, performance requirements for the application, etc., in addition to evaluating current resource usage prior to determining whether to add/remove resources from the system. Table 1 provides a conceptual comparison of the primary scaling methods available today and their potential for use in optimizing cloud resource utilization. This comparison was created to help define subsequent experimental evaluations rather than reporting on actual experimental performance data.

Table 1: Comparison of Cloud Scaling Strategies

Scaling Strategy	Decision Criteria	Resource Allocation Approach	Main Advantage	Main Limitation
Static Allocation	Predetermined capacity	Maintains fixed resources	Simple management	High risk of overprovisioning
Threshold-Based Scaling	CPU or memory thresholds	Adds or removes resources when thresholds are crossed	Easy implementation	Sensitive to temporary fluctuations
Reactive Scaling	Current workload measurements	Responds after demand changes occur	Adapts to immediate conditions	May react too late to sudden demand
Predictive Scaling	Historical patterns and forecasts	Allocates resources before anticipated demand	Supports proactive provisioning	Forecasting errors may waste resources
Workload-Aware Scaling	Workload patterns and resource metrics	Adjusts resources according to workload characteristics	Improves decision relevance	Requires comprehensive monitoring
Hybrid Scaling	Current utilization and predicted demand	Combines reactive and predictive decisions	Balances responsiveness and preparation	Increased implementation complexity
Policy-Constrained Scaling	Performance objectives and capacity limits	Enforces minimum capacity and scaling safeguards	Supports service reliability	Strict constraints may retain idle capacity
Adaptive Workload-Aware Scaling	Workload behavior, forecasts, feedback, and performance targets	Continuously refines provisioning and release decisions	Supports ongoing resource optimization	Requires careful validation and tuning

This comparison shows that there is no universal scaling strategy for all cloud applications. For applications that require a fixed amount of capacity and have highly predictable usage, then static allocation will likely continue to be an effective strategy. On the other hand, if you want your system to react quickly to changes in demand and be simple to implement, then threshold-based and reactive approaches are best. Predictive scaling can help prepare in advance for regular peaks in demand; however, its success depends upon how accurately you can predict the future peak demand and how stable the patterns of your application's workload are. Workload-aware and hybrid strategies allow you to look at the current state of your system as well as what you anticipate will happen next, which gives you a wider range of options when making decisions about how to scale. The adaptive workload-aware approach we propose expands

upon these two types of strategies by taking into account performance feedback, scaling protections (e.g., preventing too much or too little scaling), and other factors related to the specific application in order to make decisions regarding managing resources. We believe that we need to conduct controlled experiments to determine if the increased complexity of the adaptive workload-aware approach provides any significant advantages over less complex approaches in terms of resource utilization, cost of maintaining infrastructure, speed of responding to users, number of times that resources are scaled up/down, and number of times that service-level objectives are violated. Therefore, we believe that the table presented in this paper provides a conceptual basis for determining which scaling policy is most suitable for each application and creating an experimental framework for evaluating unnecessary use of cloud resources.

ADVANCED OPTIMIZATION TECHNIQUES FOR WORKLOAD-AWARE CLOUD SCALING

A. Multi-Metric Resource Allocation and Decision-Making

The evaluation of multiple performance metrics, rather than relying solely on either CPU utilization or memory consumption, is critical to achieving effective cloud resource optimization. Cloud applications will have various types of resource pressure, and consequently no single metric will accurately reflect the true computing needs of that application. An example would be an application that is utilizing its CPU at a reasonable level, but is experiencing increasing request queue sizes, memory pressure, or network congestion. Therefore, the workload-aware framework presented here incorporates CPU utilization, memory consumption, request arrival rate, response time, queue length, and available computing capacity into a single decision making process. Each metric identifies whether additional resources are needed to meet performance requirements for the application or if the existing infrastructure has sufficient capacity to handle the current workload. Historical measurements offer additional insight by providing evidence of recurring demand patterns versus transient fluctuations. Also incorporated into the decision making process is the critical nature of the application and pre-determined Service Level Objectives (SLOs) to determine the resource needs. Using all of these metrics together, cloud operators can make more educated scaling decisions and avoid over-provisioning resources while maintaining the reliability of operations.

The assessment of the current state of a resource based on observed conditions, expected demands and established acceptable levels of performance are what drives the assessment of the need to take action. If several factors indicate that there will continue to be pressure on resources then the framework can either suggest increasing the capacity of the resource or moving some of the workload to another resource. Conversely, if there has been a pattern of low usage of resources with plenty of available capacity then the framework can find ways to decrease the amount of resources that are assigned to the resource. A system of weighting for each indicator may be employed to determine which indicators are more critical than others based upon the needs of the application. For example, response time may have a higher weight for latency-sensitive applications whereas processing throughput may have a higher weight for batch workloads. It is important to configuration and evaluation of these weights to ensure that the appropriate scaling decisions are made. Additionally, minimum capacity limits and cooldown periods can prevent excessive resource deletion during periods where there is a short-term reduction in demand. This multi-metric approach allows for flexibility in scaling based upon workload characteristics across multiple cloud platforms. To evaluate how effectively this strategy works, one can measure infrastructure consumption, application response time, frequency of scaling events and adherence to service level objectives (SLOs) under typical workload scenarios.

B. Resource Allocation Under Dynamic and Unpredictable Workloads

Cloud resource management presents serious problems when there are dynamic workloads; since demand is capable of changing dramatically and unpredictably. For example, some examples of applications that support on-line transactions, digital communications, data processing, and/or interactive services are subject to quick fluctuations in the amount of requests received. Traditionally, threshold-based scaling methods do poorly during periods of rapid fluctuation in request volumes, as such methods react only after a resource utilization limit has been exceeded. The delay in reacting to resource limits can result in a period of reduced performance. On the other hand, if a scaling mechanism reacts too quickly to fluctuations in request volumes then unnecessary provisioning will occur along with frequent scaling operations. The proposed workload-aware framework attempts to address the issues associated with traditional threshold-based scaling mechanisms by utilizing real-time monitoring of resource usage, historical analysis of workloads, and short-term demand estimations. The framework monitors changes in request arrival rates, resource consumption, and application performance to evaluate whether the changes being monitored indicate a long-term change in workload, or if the changes represent a short-term fluctuation. If the framework determines that the workload is predictable, the framework will begin to provision resources prior to the predicted increase in workload. Conversely, if the framework determines that the workload is unpredictable, the framework will maintain adequate

reserve capacity and will refrain from aggressively reducing available resources. Overall, the objective of the framework is to optimize the balance between achieving maximum resource utilization efficiency and providing adequate operational resiliency for cloud-hosted applications.

In addition to managing the uncertainty of workload variations, the architecture must provide protection mechanisms for applications during times when actual demand significantly exceeds the forecasted demand. For example, the architecture can incorporate a margin of error (conservative) in terms of capacity, require a minimum number of instances to run, establish rapid scaling out parameters, and define controlled scaling in procedures. These protections will ensure that the system does not under-provision in the event of a sudden surge in traffic. Additionally, the accuracy of forecasting models should be continually assessed through comparisons of forecasted demand to observed demand, as well as identification of common patterns of error within the forecasts. If there is a high degree of uncertainty regarding forecasting, then the scaling system may rely more heavily upon real-time metrics of performance rather than potentially unreliable forecasts. Due to varying tolerances for delays and/or resource shortages among various types of workloads (e.g., interactive applications, batch-processing systems, and background services), each workload type should be evaluated independently. Evaluations of the proposed architecture should include experiments involving sudden spikes in traffic, intermittent workloads, periodic peak demands, and extended periods of low demand. Measurements obtained from these experiments will demonstrate if the proposed architecture is capable of reducing idle capacity while still providing acceptable levels of response time and service availability. This method of evaluation will allow for greater flexibility in making scaling decisions, without requiring the assumption that every change in workload can be accurately predicted.

C. Continuous Feedback, Policy Adaptation, and Operational Sustainability

Continuous feedback is an important component of adaptive cloud resource management as the demands of the applications and workloads vary throughout the life of the systems. Scaling policies that were configured during one operating period may no longer be efficient based upon changes in user activity, changes in the application architecture, or variations in the services requested. Therefore, the proposed framework will compare the scaling decisions made with the results obtained in terms of how well the decision to adjust resources has improved utilization and maintained the desired level of performance. Observations that are considered relevant include changes in CPU and memory usage; changes in response-time distribution; changes in request throughput; changes in the number of allocated resource hours; and changes in the number of times a scaling action was taken. If continuous measurement indicates that the scaling policy has either provided too much resource too quickly or removed too much resource too quickly, then the threshold values used for making the scaling decision and/or the parameters used for forecasting future requirements should be reevaluated. Any changes to the scaling policy should occur using formal procedures in order to avoid generating erratic system behavior or unintentionally degrading the performance of the system. In addition, historical records related to previous scaling decisions may provide support for auditing purposes; may help explain why specific resources were allocated; and may assist in identifying ongoing opportunities for optimization. By utilizing operational feedback in conjunction with a systematic approach to evaluating scaling policies, cloud administrators can enhance their ability to manage resources effectively throughout multiple cycles of workload.

Operational sustainability provides another motivation for organizations to eliminate unused or underutilized cloud resources. Using cloud resources (i.e. computing instances) that are idle uses valuable data center space and could lead to unnecessary energy consumption, air conditioning needs, and other negative environmental consequences. While there is no guarantee as to whether there will be a positive environmental impact from eliminating idle cloud resources due to several variables including the type of hardware used by the organization, the cloud provider selected, the organization's current energy source, and the specific workload being run, optimizing resource utilization can help make cloud operations more efficient. This proposed framework includes considerations for resource efficiency in conjunction with service reliability and infrastructure costs; it does not treat the reduction of cloud resources as its primary goal.



Figure 2: Workload-Aware Cloud Scaling Framework

PERFORMANCE ASSESSMENT AND PRACTICAL DEPLOYMENT OF WORKLOAD-AWARE SCALING

A. Experimental Design and Performance Evaluation Metrics

The effectiveness of a workload-aware scaling approach will need to be evaluated experimentally; the experimental design must include comparison between the proposed approach and traditional approaches to managing cloud resources. The evaluation could take place in a cloud simulation environment, in a container orchestration platform, or in a controlled cloud environment. To establish a baseline for comparison, we can use a traditional threshold based auto-scaling mechanism. We will also run our proposed workload aware framework on the same types of workloads and on the same type of infrastructure as the traditional threshold-based auto-scaling mechanisms. Our experimental workloads must represent a variety of different types of application behavior (e.g., stable demand, predictable period-based variations, rapid increases in traffic, and extended periods of lower utilization). For each experiment we will collect data from our monitoring systems regarding CPU consumption, memory usage, request arrival rate, response time, throughput, and allocated resource hours. We will calculate the cost of our infrastructure based upon consistent pricing assumptions. We will also measure resource utilization to determine times during which allocated capacity greatly exceeded required demand. We can conduct multiple experiments to differentiate consistent performance patterns from random variation. Documentation of forecasting techniques, scaling thresholds, cool-down intervals, and minimum resource limitations will provide support for reproducibility. This methodology will provide a formal structure for evaluating if workload aware decisions provide additional value compared to conventional scaling techniques.

Resource allocation efficiency can be measured through the ratio of allocated computing capacity that is being used, while infrastructure cost will measure the monetary implications of resource provisioning decisions. Performance evaluations should include more than one performance indicator since improving one indicator could lead to a trade-off in another. Both average response times and high percentile latencies can help determine if reducing resources affects application performance adversely. Service level objectives violations and throughput are other indicators that show how well the system maintains acceptable levels of service quality. Scaling frequency will indicate whether the suggested approach reduces unnecessary provisioning and terminating of resources, or creates too many changes. Researchers may find the percentage of reduction in resource utilization by comparing the baseline with the workload-aware approaches over the same amount of time. However, any reported reductions in resource usage should come from measured data, not predicted benefits. In addition, researchers should take into account the forecasting overhead, resource initialization delays, and the amount of capacity available for unexpected demand. A successful framework should have a good balance between cost efficiency, utilization, responsiveness and reliability.

B. Deployment Architecture, Integration, and Operational Considerations

To apply workload-aware scaling to real-world applications, it will need to integrate with your existing cloud monitoring, orchestration, and resource management solutions. Your monitoring solution will collect application and infrastructure data via its telemetry services, container metrics, application performance monitoring (APM) tools, or cloud provider APIs. An application/infrastructure analysis solution will process this data and combine it with previous data to determine how you are using resources. An application/infrastructure prediction solution can use statistical forecasting methods, machine learning models, etc., to predict what your future resource needs may be. A decision engine will evaluate those predictions against your current available capacity, performance metrics, service-level agreements (SLAs), and resource limits that you have set up. Based upon those evaluations, the system can either make

recommendations for scaling out, scaling in, resizing, or retaining resources. Integration with orchestration platforms enables the decision engine to convert those recommendations into operational changes (e.g., changing the number of container replicas, changing the size of a compute instance). Prior to enabling automated actions, the policies you propose should be tested with representative workloads. Additionally, you will need to establish monitoring coverage, access permissions, decision logging, and recovery procedures. This layer-based approach allows you to gradually implement workload-aware scaling while minimizing the amount of disruption to your current cloud management infrastructure.

The primary focus during the deployment process should be the continued operational reliability of the system as incorrect scaling decisions could lead to decreased availability of applications and user experiences. There are several ways to help prevent the removal of resources which may be required in the future including; minimum capacity limits, controlled scale-in conditions, cool-down periods, and rapid response rules. Additionally, the deployment process should have protections built-in against monitoring failures, stale workload measurements, and unreliable predictions. If the forecasting component is unable to function correctly or produces highly unreliable estimates, then the system should utilize a previously tested reactive scaling policy. Finally, the security and governance requirements of the system should be met through restricted access to resource modification interfaces and auditable records of all scaling actions taken. Cloud administrators should continually monitor resource utilization trends and operational cost to determine if there are any issues with configurations or excessive overhead. A phased rollout allows for recommendations to be made by an administrator prior to being implemented automatically.

COST OPTIMIZATION AND RESOURCE UTILIZATION ANALYSIS

In order to minimize unneeded cloud resources, there must be a balance among the following: cost of infrastructure, amount of compute power available, variability in workloads, and performance of applications. There are many different types of workloads in a cloud environment; each one has its own specific needs regarding how much compute power it will require. Therefore, determining what is an optimal allocation of resources for all workloads is extremely difficult. If you over provision your cloud environment, you will have higher infrastructure costs. On the other hand, if you under provision your cloud environment, you may experience processing delays, decreased throughput, and violations of your service level agreements (SLAs).

Table 2: Comparative Evaluation of Workload-Aware Scaling Approaches

Evaluation Dimension	Measurement Indicator	Conventional Approach	Proposed Approach	Expected Benefit
Resource Utilization	CPU and memory utilization	Threshold-based monitoring	Multi-metric workload analysis	Better utilization of allocated capacity
Infrastructure Cost	Resource-hours and service expenditure	Reactive provisioning	Demand-aware allocation	Potential reduction in unnecessary expenditure
Application Performance	Response time and throughput	Scaling after threshold violations	Performance-constrained scaling	More consistent service quality
Scaling Stability	Number of scaling operations	Fixed thresholds and cooldowns	Adaptive thresholds and workload persistence	Fewer unnecessary adjustments
Capacity Planning	Forecast error and available capacity	Historical averages or static estimates	Workload prediction and feedback	Improved preparation for changing demand

This framework was developed based upon the evaluations of each of the five dimensions described above. Resource usage (utilization) metrics provide insight into areas where resources are consistently underutilized. Cost indicators assist with determining if resource allocation decisions result in financial penalties. Performance metrics for applications ensure that any reduction in resources will not negatively affect the ability of an application to respond quickly enough, or maintain reliability. Stability indicators for scaling demonstrate if a decision-making system reacts too quickly to short-term changes in demand. Capacity planning metrics help determine if a system has the ability to

forecast future demands. Since the metrics for each of the five dimensions are interrelated, they should be evaluated collectively. Optimizing for one metric alone could lead to negative consequences in other metrics. For example, if we were to optimize for utilization and not retain enough extra capacity, we would then run the risk of having poor performance when there is a sudden increase in demand. Conversely, if we constantly forecast and scale, we may create excessive computational overhead. Comparing our proposed approach to traditional auto-scaling would allow us to assess the advantages and disadvantages of each method.

BENEFITS AND PRACTICAL RECOMMENDATIONS FOR WORKLOAD-AWARE CLOUD SCALING

An organized method of enhancing the ability of an organization to manage its cloud resources through systematic approaches to matching the amount of computational power with both present application needs and future application needs based on expected workload changes. Typically, traditional approaches to resource allocation can result in organizations maintaining extra hardware at times of low demand for applications; conversely, traditional approaches to resource allocation can result in organizations responding too slowly to sudden increases in application traffic. The proposed approach will address both issues through real-time monitoring of application behavior, analysis of workload patterns, projection of future demand, and dynamic adjustment of resource allocations. The main goal of this approach is to eliminate unneeded resource usage while maintaining high levels of application responsiveness, service availability, and operational stability. Organizations must take into account workload attributes, hardware constraints, accuracy of predictions of future demand, and service level agreements (SLAs) when creating scaling rules. Below are some recommendations that provide a summary of the key methods that can facilitate successful implementation.

- **Constant Resource Tracking:** Monitor CPU usage, RAM consumption, number of requests per second, time to complete each request (response), and the size of the wait list (queue) to accurately determine what resources are required by an application, and to find out how often applications use fewer resources than expected.
- **Workload Pattern Analysis:** Evaluate current and previous workload data to differentiate between predictable fluctuations in demand, long-term trends of increased resource usage, and short-term peaks in usage which do not warrant the acquisition of more hardware.
- **Anticipate Computing Needs:** Use appropriate statistical forecasting or machine learning methods to predict future computing needs and ensure adequate resource availability prior to the occurrence of increased workloads that could degrade application performance.
- **Adaptive Scaling Policies:** Prior to either adding or removing resources from a system, consider how well it is performing based on a variety of different metrics. Consider such things as predicted workloads, existing resource use, the importance of the application(s) running on the system, the amount of additional capacity available within the system, and any pre-established service level agreements (SLAs) regarding the performance of the system.
- **Controlled Resource Reduction:** Use of cool-down periods, low resource levels as a threshold for automatic scaling-in and/or hysteresis controls to help avoid making early scale-in decisions, multiple scalings in a short time frame and negative impacts on performance during short-lived spikes in demand.
- **Evaluate Performance and Cost:** Analyze how the new architecture compares to standard autoscaling based upon cost, total hours of resources consumed, how efficiently resources are utilized, response times, throughput, and how often scaling events occur. Do not assume improvements; conduct controlled experiments to prove your claims.
- **Continual Improvement and Reliability:** Review the actual results of scaling; track the errors of predictions; adjust the parameters used to make decisions; and provide a back-up plan in case there are unexpected workloads or failures of monitoring.

FUTURE DIRECTIONS IN WORKLOAD-AWARE CLOUD RESOURCE OPTIMIZATION

As the complexity of cloud-based applications increases, so too does the need for intelligent predictions, automated decision making, and the ability to integrate various aspects of infrastructure management. For example, traditional static-threshold-based scaling approaches to managing cloud resources are likely to fall short in providing adequate responsiveness to meet rapidly changing workload characteristics and resource demands. Machine learning algorithms that analyze past usage patterns, request volume, application dependencies, and repeating operational cycles will allow for improved estimates of future resource requirements. Similarly, reinforcement learning techniques can help develop adaptive scaling strategies based upon what is learned through the effects of making different resource allocations across various types of workloads. Nevertheless, there are significant potential downsides to using such techniques. Specifically, if the predictions generated by such techniques are inaccurate; if they require too much computing power; or if they are trained on inappropriate learning goals, then they will ultimately fail to produce any real benefit.

Consequently, there is a need for additional research to identify simple forecasting models that provide accurate predictions while minimizing the computational costs associated with them. Additionally, incorporating uncertainty analysis into the process of determining how to scale resources can enhance reliability by enabling the system to maintain a safety margin of available resources during times when workload patterns cannot be predicted.Resource allocation is another significant research area. Resource allocation is concerned with coordinating resource allocation across containers, virtual machines, serverless platforms, and geographically distributed clouds.

Many applications today use several types of infrastructure to run their services. When each type of service makes independent scaling decisions, it can lead to inefficient scaling if the services depend upon each other. An integrated workload-aware framework could take into account how well the services communicate with each other, what resources are available, the amount of latency in the network, and the cost of the regional infrastructure to help make decisions about how much capacity needs to be adjusted. There are unique challenges associated with hybrid cloud and edge computing environments. For example, some computing resources may have better performance characteristics than others. They may also have different availability constraints and/or different price structures. Future implementations may examine ways to optimize both workload placement and scaling decisions at the same time as providing applications with responsiveness and adhering to data governance policies. Finally, energy-aware scaling is another promising research direction, especially if reliable measurements of the energy consumed by the computing resources and the energy efficiency of the infrastructure exist. These approaches should provide a balance between environmental considerations and performance requirements rather than assume that maximizing the use of resources will always reduce energy consumption.

Future validation must use reproducible experiments with representative workloads, well-defined baseline strategies, and clear evaluation metrics. Researchers will be able to evaluate the effectiveness of workload-aware scaling compared to reactive, predictive, and hybrid auto-scaling approaches in the same infrastructure environment. Researchers may also measure the effects of workload-aware scaling on resources (resource-hours), cost (infrastructure expenditure), accuracy of predictions (prediction error), performance (response time percentiles, throughput), rate of scaling (scaling frequency), and compliance with service level objectives (service level objective violations). Researchers should also assess the ability of the system to handle unexpected increases in traffic, failure of monitoring tools, and errors in forecasting. Researchers may use public workload traces and documented simulation configurations to make experimental results more reproducible and allow researchers to compare experimental results from different studies. In addition, researchers may examine issues related to the deployment of workload-aware scaling systems such as security, transparency in operations, policy management and fallback mechanisms that allow the system to operate safely in the event of failure. These areas of research will help researchers understand when workload-aware scaling provides measurable benefits, what its limitations are, and how researchers can implement reliable methods for minimizing unnecessary allocations of cloud resources while providing effective, responsive, and sustainable cloud services.

CONCLUSION

Reducing excess cloud resources used by application-driven scaling decisions will lead to better cost-effectiveness, operational reliability and long-term sustainability of cloud-based systems. With the growing reliance on cloud-based applications, proper management of resources is becoming increasingly critical in terms of managing infrastructure expenses while providing the required level of service to users. Many of the existing scaling methods (such as those that use set utilization thresholds) may provide extra resources in response to short-term increases in workload or continue to use resources beyond what is needed once the demand on the system declines. Both of these approaches can lead to inefficient use of resources, higher operational expenses, and avoidable energy consumption. This study addressed the limitations of current allocation strategies and proposed a conceptual workload-driven scaling model that incorporates real-time monitoring of workloads, historical analysis of workloads, prediction of future workloads and dynamic decision making. The goal of this research was to make sure that the amount of computing capacity allocated to a system matches both current and future workload needs, while also ensuring that there are sufficient safeguards in place to guarantee availability and responsiveness of the application. The proposed method demonstrates that optimizing cloud systems involves more than just decreasing the total number of running instances; it involves identifying the minimum amount of capacity needed to support dynamic application requirements in a timely and cost-effective manner.

The proposed framework includes several related elements that will allow us to characterize workloads; predict resource needs; establish adaptive scaling rules; and provide ongoing performance feedback. Workload monitoring provides data regarding CPU usage, memory consumption, request arrival rate, queue length, and response time.

Predictive analytics utilizes both past and present data to estimate the amount of resources that may be required in the future. The decision engine compares the estimated required resources with the current available capacity and with the application performance goals. After evaluating the comparison, a determination is made as to whether or not a scale-out or scale-in action should occur; whether or not to retain some of the existing resources; and whether or not to change the size of the existing resources based upon workload conditions and the defined operational limits. There are other mechanisms that could help to limit unnecessary scaling actions and protect applications from unanticipated changes in demand such as cooldown periods, hysteresis thresholds, minimum capacity levels, and fallback strategies. Additionally, the research has addressed the need for application specific criteria as different types of applications have differing performance characteristics and resource usage patterns. Together, these components create a structured foundation for improving our ability to make better-informed allocation decisions than those made using only isolated utilization thresholds.

The first major contribution of this study was the development of quantifiable metrics to assess the benefits of workload-aware scaling compared to traditional autoscale approaches. There are many ways to evaluate the performance of a system including resource-hours, infrastructure expenses, average utilization, response time percentiles, throughput, frequency of scaling events and number of violations of service level objectives. All these metrics must be considered together as reducing resource use alone does not ensure that an application is operating reliably. At present, the proposed model remains a concept and must be empirically validated via controlled simulation studies or cloud-based deployments utilizing representative workloads. Future experimental evaluations will include stable demands; periodic peaks; sudden spikes in traffic; and irregular workloads with consistent infrastructure settings. Additionally, future experiments will evaluate factors such as error in forecasting; overhead associated with monitoring; delay in starting instances; and costs associated with maintaining spare capacity. Therefore, it is impossible to state what specific amount (or percentage) of reduction in either resource usage or infrastructure expense can be achieved without validating the model experimentally.

In summary, a workload-aware scaling mechanism provides a solid base for minimizing excessive cloud resource allocation in an environment where cost efficiency, application performance, and operational stability are balanced. The development of future research may help strengthen the framework through the use of lightweight predictive models, reinforcement learning, coordinated scaling across multiple distributed services, and energy-aware resource management. Additionally, consideration should be given to handling uncertainties related to forecasting, security issues, scalability, and how to safely integrate automated decisions into current cloud architectures. Organizations implementing such techniques should start by conducting controlled evaluations, set measurable performance goals, and gradually automate the process once they have verified that it functions properly. Continued monitoring and feedback loops may enableorganizations to refine their scaling policies based on evolving workloads and application requirements. Ultimately, the success of workload-aware cloud resource management will depend upon whether organizations are able to make accurate, timely, and context-sensitive decisions while maintaining adequate capacity for unexpected demands. Through systematic implementation and empirical evaluation, this strategy can provide organizations with more efficient, dependable, and cost-effective cloud computing environments.

REFERENCE

[1] Armbrust, M. et al. (2010) 'A view of cloud computing', Communications of the ACM, 53(4), pp. 50–58.

[2] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I. (2009) 'Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility', Future Generation Computer Systems, 25(6), pp. 599–616.

[3] Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F. and Buyya, R. (2011) 'CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms', Software: Practice and Experience, 41(1), pp. 23–50.

[4] Chen, T., Bahsoon, R. and Yao, X. (2018) 'A survey and taxonomy of self-aware and self-adaptive cloud autoscaling systems', ACM Computing Surveys, 51(3), Article 61.

[5] Chiang, R.C. and Huang, H.H. (2011) 'TRACON: Interactively optimizing virtual machine placement in clouds', in Proceedings of the 2011 IEEE International Conference on Cloud Computing. Washington, DC: IEEE, pp. 81–88.

[6] Dean, J. and Barroso, L.A. (2013) 'The tail at scale', Communications of the ACM, 56(2), pp. 74–80.

[7] De Assunção, M.D., da Silva Veith, A. and Buyya, R. (2018) 'Distributed data processing and caching infrastructure for cloud-based big data analytics', Future Generation Computer Systems, 86, pp. 1023–1034.

[8] Fan, J., Wang, X., Wu, L. and Liu, J. (2015) 'An adaptive resource allocation approach for cloud computing based on workload prediction', International Journal of Distributed Sensor Networks, 11(9).

[9] Galante, G. and de Bona, L.C.E. (2012) 'A survey on cloud computing elasticity', in Proceedings of the 2012 IEEE Fifth International Conference on Utility and Cloud Computing. Washington, DC: IEEE, pp. 263–270.

[10] García-Valls, M., Cucinotta, T. and Lu, C. (2014) 'Challenges in real-time virtualization and predictable cloud computing', Journal of Systems Architecture, 60(9), pp. 726–740.
[11] Gmach, D., Rolia, J., Cherkasova, L. and Kemper, A. (2007) 'Workload analysis and demand prediction of enterprise data center applications', in Proceedings of the 2007 IEEE International Symposium on Workload Characterization. Washington, DC: IEEE, pp. 171–180.
[12] Guitart, J., Carrera, D., Beltran, V., Torres, J. and Ayguadé, E. (2007) 'A dynamic CPU provisioning model for virtualized service providers', in Proceedings of the 2007 International Conference on Parallel Processing. Washington, DC: IEEE, pp. 26–33.
[13] Herbst, N.R., Kounev, S. and Reussner, R. (2013) 'Elasticity in cloud computing: What it is, and what it is not', in Proceedings of the 10th International Conference on Autonomic Computing. San Jose, CA: USENIX Association, pp. 23–27.
[14] Islam, S., Keung, J., Lee, K. and Liu, A. (2012) 'Empirical prediction models for adaptive resource provisioning in the cloud', Future Generation Computer Systems, 28(1), pp. 155–162.
[15] Jiang, Y. and Perng, C.S. (2010) 'Cloud resource allocation and management for performance optimization', Proceedings of the IEEE International Conference on Cloud Computing, pp. 1–8.
[16] Lorido-Botran, T., Miguel-Alonso, J. and Lozano, J.A. (2014) 'A review of auto-scaling techniques for elastic applications in cloud environments', Journal of Grid Computing, 12, pp. 559–592.
[17] Mao, M. and Humphrey, M. (2012) 'A performance study on the VM startup time in the cloud', in Proceedings of the 2012 IEEE International Conference on Cloud Computing. Washington, DC: IEEE, pp. 423–430.
[18] Masdari, M. and Khoshnevis, A. (2020) 'A survey and classification of the workload forecasting methods in cloud computing', Cluster Computing, 23, pp. 1507–1526.
[19] Mauch, V., Kunze, M. and Hillenbrand, M. (2013) 'High performance cloud computing', Future Generation Computer Systems, 29(6), pp. 1408–1416.
[20] Qu, C., Calheiros, R.N. and Buyya, R. (2018) 'Auto-scaling web applications in clouds: A taxonomy and survey', ACM Computing Surveys, 51(4), Article 73.
[21] Ranjan, R. (2015) 'Streaming big data processing in datacenter clouds', IEEE Cloud Computing, 1(1), pp. 78–83.
[22] Rao, J., Bu, X., Xu, C.Z., Wang, L. and Yin, G. (2009) 'VCONF: A reinforcement learning approach to virtual machines auto-configuration', in Proceedings of the 6th International Conference on Autonomic Computing and Communications. New York: ACM, pp. 137–146.
[23] Reiss, C., Tumanov, A., Ganger, G.R., Katz, R.H. and Kozuch, M.A. (2012) 'Heterogeneity and dynamicity of clouds at scale: Google trace analysis', in Proceedings of the Third ACM Symposium on Cloud Computing. New York: ACM, Article 7, pp. 1–13.
[24] Singh, S. and Chana, I. (2016) 'A survey on resource scheduling in cloud computing: Issues and challenges', Journal of Grid Computing, 14, pp. 217–264.
[25] Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M. and Tantawi, A. (2008) 'Analytic modeling of multitier Internet applications', ACM Transactions on the Web, 2(1), Article 1.
[26] Verma, A., Pedrosa, L., Korupolu, M., Oppenheimer, D., Tune, E. and Wilkes, J. (2015) 'Large-scale cluster management at Google with Borg', in Proceedings of the 10th European Conference on Computer Systems. New York: ACM, Article 18, pp. 1–17.
[27] Wang, W., Niu, D., Li, B. and Liang, B. (2012) 'Dynamic cloud resource allocation: Models and algorithms', in Proceedings of the 2012 IEEE International Conference on Cloud Computing. Washington, DC: IEEE, pp. 1–8.
[28] Xu, J. and Fortes, J.A.B. (2011) 'Multi-objective virtual machine placement in virtualized data center environments', in Proceedings of the 2011 IEEE/ACM International Conference on Green Computing and Communications. Washington, DC: IEEE, pp. 179–188.
[29] Zhu, X., Young, D., Watson, B.J., Wang, Z., Rolia, J., Singhal, S., McKee, B., Hyser, C., Gmach, D., Gardner, R., Christian, T. and Cherkasova, L. (2008) '1000 islands: An integrated approach to resource management for virtualized data centers', Cluster Computing, 11, pp. 273–286.
[30] Jeong, B. and Jeong, Y.S. (2025) 'Autoscaling techniques in cloud-native computing: A comprehensive survey', Computer Science Review, 58, Article 100791.