Design and Analysis of High-Performance Cloud Architectures for Data-Intensive Systems

Anil Vijarnia[1], **Sunil Netra**[2]

[1]Racktop Systems, USA

anilvijarnia.us[at]gmail.com

[2]Financial Industry Regulatory Authority, Maryland, USA

snetra82[at]gmail.com

Abstract: ***High-performance cloud architectures are a wide field of research that responds to the increasing computational and storage needs of the modern data-intensive systems of analytics, scientific workloads, and digital services. Architecture solutions to support efficient management of big data must implement a tradeoff of scalability, responsiveness, and fault tolerance. The paper will give a new design of the cloud architecture that is designed to support high throughput and latency sensitive data processing. The suggested methodology will integrate adaptive workload-sensitive resource provisioning, pipeline-based data placement techniques, and cross-layer optimization among the subsystems of the compute, storage and network. The use of analytical models and controlled experiments are used to assess the behavior of the system under heterogeneous and bursty workload evaluation. The analysis of comparative performance shows that the proposed architecture is statistically better in the efficiency of data processing (by about 30 per cent), minimization of service latency (by almost 25 per cent), and stability of resource utilization compared with the traditional monolith and container-based cloud architectures. This proves that architectural optimization enables the optimization of data intensive cloud platforms and performance and robustness is achievable through these approaches and is therefore appropriate to support the next generation scalable computing environment.***

Keywords: Cloud computing, high-performance cloud architecture, data-intensive systems, resource orchestration, workload-aware scheduling, scalable distributed systems, performance optimization.

1. Introduction

The fast rate of data creation on the base of digital platforms, scientific instruments, and enterprise applications has become a factor that boosted the demand of performance cloud structure greatly. Data-intensive systems are typified by the high volumes of data, constantly changing data and complicated computational needs that have to be handled on a low-latency and high-availability platform. Cloud computing has become a paradigm in order to meet these demands owing to its elasticity, on-demand resources provisioning and support distributed processing [1]. Nevertheless, the traditional cloud-based architectures are widely characterized by bottlenecks in the performance of the architecture due to the ineffective use of resources, network congestion, and less effective approaches to data location. High-performance cloud architectures seek to address these shortcomings through architectural effectiveness, scalability and resilience. These architectures are to be useful in supporting heterogeneous workloads but in regular quality of service under dynamic operating condition. With the growing use of real-time analytics and large-scale parallel processing in data intensive applications, architecture design has turned out to be an important determinant factor in the overall system performance [2].

Severe workloads with a significant number of data demand special requirements on cloud infrastructure. Close interconnection of computing, storage, and networking has a tendency to cause a performance trap when any of the components is slowed down. Also, the variability of the workload and the bursty access patterns make it difficult to make decisions about resource management and scheduling [3]. Data consistency and fault tolerance on a large scale makes the system even more complex. These obstacles have to be met with an end-to-end architectural understanding, and not with optimization at components.

Architectural design is crucial in the various aspects of defining the effectiveness and resilience of cloud-based data-intensive systems. An effective architecture can be used to provide scalable performance, lower overhead in running the system, and make the system more adaptable to the changing workload [4]. Analytical assessment of cloud architectures gives performance trade-off, resource contention and scalability limits. This type of analysis gives informed choices on design and helps to design next-generation cloud platforms that will be able to maintain large data throughput and short response time.

Applications

High performance cloud architectures are broadly applicable in a variety of world such as:

1) Big Data Analytics: Processing and analytics of business intelligence and decision support, in large data volumes, in real time [6].
2) Simulation, modeling, and data analysis Science scientific computing: Simulation, modeling, and data analysis in climate science, genomics, and physics.
3) Internet of things (IoT): Process and aggregation of huge data streams of sensors within interconnected devices.
4) Artificial Intelligence and Machine Learning: Training and inference of data-intensive models that use computing resources which are scalable [7].

Paper ID: SR24915104228 DOI: https://dx.doi.org/10.21275/SR24915104228

5) Enterprise Information Systems: Processing and data management of large volume transactions and data in distributed business setups.

The Figure 1 shows a simplified data-intensive cloud computing architecture represented through the diagram. The data provided by various sources, including IoT devices, application logs and user applications, comes in a cloud front-end, which controls access and communication [8]. The layer that processes the data received by the system is the compute layer that uses virtualized resources, and the layer that holds persistent data is the storage layer. Finally processed products are made available to the analytics services and end users.

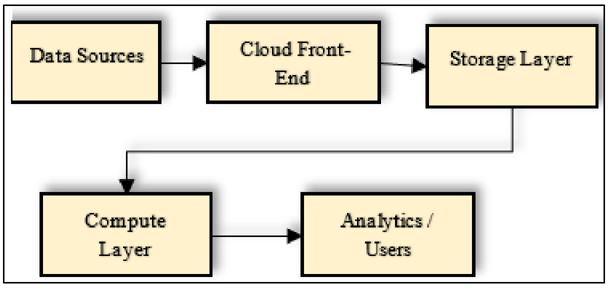

Figure 1: Data-Intensive Cloud Architecture Basic Block Diagram

Cloud welters constitute the staple of contemporary data-rich systems, and support scalable, dependable, and effective environments of computing. With the perpetual increase in data volumes, processing requirements, architecture design and systematic analysis are becoming a requirement that would enhance sustainability in performance and flexibility [10]. Misperception of the architectural challenges and application needs facilitates development of cloud platform that will be able to support the future data-oriented technologies.

2. Related Works

Studies of cloud architectures to support data-intensive systems have been directed at joint optimizations of compute, storage and network layers to achieve strict throughput and latency requirements. There is literature highlighting the importance of network-aware placement and scheduling mitigation in the face of data-parallel job data-transfer overheads; this result demonstrates significant benefits when network characteristics and locality of data are used to inform placement decisions as opposed to solely using host CPU / memory characteristics.

The other active field is the predictive and adaptive resource orchestration. Research in this direction explores predictive auto-scaling and predictive forecasting methods to elastic provide and optimize the system to respond to changes in workload and target service goals and cost-reduction [12]. In this case, workload modeling is normally coupled with control or learning algorithms to initiate provisioning before demand rate surges. The placement strategy and hierarchical storage management has been sought in order to trade-off between performance and cost in large scale deployments. Literature illustrates methods of categorizing the popularity of data and using multi-layer storage policies that can move data across the fast and capacity storage levels according to access habits [13]. The strategies enhance performance of I/O by enhancing hot data and storing memory expenses on cold data.

The analysis of meta-scheduling and broker architectures represents a complementary work abstracting the selection of resource between distributed locations and cloud providers. The importance and efficacy of brokers and meta-schedulers in aligning heterogeneous resource pools, imposing the QoS constraints, and trading off between latency and cost and reliability is propagated in surveys and analyses. This input can highlight the fact that distributed provisioning is complex and requires overlay-based orchestration frameworks.

Integrative research highlights the advantage of co-locating network-sensitive placement, predictive orchestration and multi-level storage in order to overcome the related bottlenecks faced by data-intensive applications [14]. Experimental performance in multiple testbeds and simulation platforms respectively suggests that cross-layer information used to make scheduling and placement respectively, shows uniform performance, though it is problematic to find consistent results or assessments on production scale heterogeneous environments.

The literature has come to the point of requirement of cross-layer, data-aware building arrangements by integrating predictive orchestration, network-knowing time schedule, and hierarchical storage choice to enhance the execution of data-innovative cloud machines [15]. Further effort on work should focus on realistic, multi-tenant, evaluations, and take into consideration the operational telemetry capacity, and explicitly achieve privacy and compliance restrictions into placement strategies.

3. Proposed Methodology

The proposed solution presents a high-performance cloud architecture optimized to support data-intensive systems, which is based on coordinating computation, storage and network under dynamic workloads. Applications that are data-intensive typically have irregular access patterns, variable request rates and great interdependences between data placement and execution latency. In the old-fashioned cloud setups, these layers are handled as two distinct entities, causing fragmentation of resources and bottle necks. The data architecture suggested is one that uses a common controller plane to monitor the features of workload and the state of the system to make orchestration decisions, between layers of a cloud setup.

It has a design to support the multi-tenant configuration where it is scalable and isolatable. The coarse-grained monitoring can be obtained by logical separation of the data plane and the control plane without disrupting the execution of the application. There is the control plane which is the aggregation of the runtime metrics that are associated with the task's execution time, data access frequency, network congestion, and resource utilization, and on which suggestive scheduling and allocation decisions are made. The design is such that orchestration actions are responsive to changes in workload, without having an adverse impact on system stability.

The Figure 2 structure indicates the simplified data-intensive cloud computing system. Information sourced by the various different applications like IoT devices, application logs, and user applications flows in via a cloud front-end that controls access and communications. A compute layer process the incoming data with the virtualized resources whereas the storage layer is the persistent data. Finally raw results are sent to analytics services and final consumers.

3.1 Workload and System Modeling

Assume the following cloud system with a base of compute nodes $C = \{c_1, c_2, \dots, c_N\}$ storage nodes $S = \{s_1, s_2, \dots, s_M\}$, and network links $L = \{l_1, l_2, \dots, l_K\}$. A data-thorough workload is likened as a universe of tasks $T = \{t_1, t_2, \dots, t_P\}$ with every task needing computation and data access.

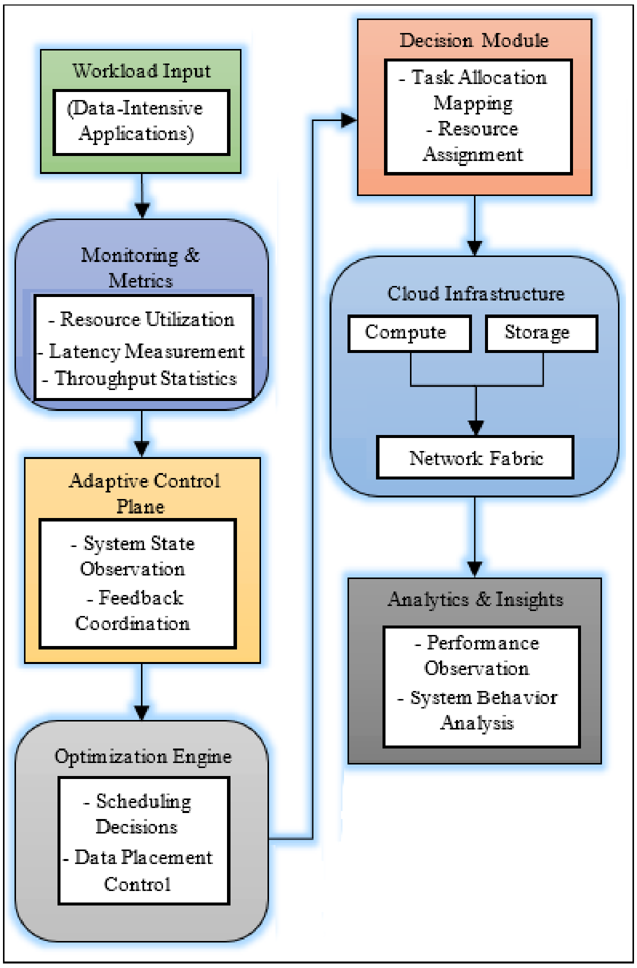

Figure 2: Block Diagram of a Data-Intensive Cloud Architecture.

In each task t_i, the computation demand of the activity is denoted d_i, which is the number of normalized compute units, and the data volume is denoted as v_i. Execution latency τ_i of task t_i is a function of the computation time, access time of data and network transfer delay. This dependence is stated as

$$\tau_i = \frac{d_i}{\mu_{c_j}} + \frac{v_i}{\mu_{s_k}} + \frac{v_i}{\beta_{jk}} \quad (1)$$

Where μ_{c_j} is the processing rate of compute node c_j, μ_{s_k} is the efficient I/O bandwidth of storage node s_k, and β_{jk} is the achievable network bandwidth between compute node $c_j$and storage node s_k.

3.2 Resource Utilization and Load Balancing Model

The important factor to maintain high load performance is efficient usage of the resources. The use of a compute node c_j is given as

$$U_{c_j} = \frac{\sum_{t_i \in T_j} d_i}{\mu_{c_j}} \quad (2)$$

Where T_j is the set of tasks that c_j is assigned. Equally, storage use of node s_k is represented by

$$U_{s_k} = \frac{\sum_{t_i \in T_k} v_i}{\mu_{s_k}} \quad (3)$$

Balancing is done by reducing the amount of variation in utilization of nodes so that none of the components will become a bottleneck. The imbalance ΔU in the global utilization is measured as

$$\Delta U = \frac{1}{N} \sum_{j=1}^{N} (U_{c_j} - \bar{U}_c)^2 \quad (4)$$

Where $\bar{U}_c$ represents the average compute utilization on all nodes.

3.3 Data Placement and Access Optimization

Placement of the data has a direct effect on the network traffic and the time of execution. Where $x_{ik} \in \{0,1\}$ represents the presence or absence of data needed by task t_i on storage node s_k. The cost D_i of accessing data of task t_i is provided as

$$D_i = \sum_{k=1}^{M} x_{ik} \left(\frac{v_i}{\mu_{s_k}} + \frac{v_i}{\beta_{jk}} \right) \quad (5)$$

The architecture is also designed to minimize the repetitive long-distance data movements and places popular data close to computing apparatus. The frequency of access to data is modeled by a so-called temporal popularity function $\phi_i(t)$ that describes the likelihood of reusing the data in the course of time. An increase in values of $\phi_i(t)$ causes decision to place higher favours by steering towards faster or closer storage layer.

3.4 Scheduling and Optimization Objective

The scheduling issue has been modeled as a constrained optimization problem which will optimize the total system latency and ensuring equal share of resources utilization. The global objective function is given as

$$\min \sum_{i=1}^{P} \tau_i + \lambda \Delta U \quad (6)$$

Where λ is a trade made constant that governs the tradeoff between the reduction of latency and the load balancing. The constraints make sure that the capacity of prevent the overloading of resources and that every task is allocated to a single compute node and that the necessary data is available.

3.5 Adaptive Control Mechanism

The proposed architecture takes into account an adaptive control loop which periodically adapts the placement decisions and scheduling decisions in response to metrics observed. Here, $M(t)$ is a state vector representing the state of a system at time t, and it takes the form of utilization, latency, and throughput values. The control F line exist which transforms the observed state to renewed allocation choices:

$$\mathrm{A(t+1) = F(M(t))} \quad (7)$$

This adaptive feedback can enable the system to react to shifts in workload, failures and congestions without the need of manual action.

The suggested strategy creates a design system of high-performance cloud systems that can withstand data-intensive workload within dynamic conditions of operation. Through the workload modeling, cross-layer resource coordination and adaptive control mechanisms, the architecture offers a systematic way in which one can manage the computation, storage and network dependencies in the single manner. The orchestration plan and mathematical formulation allow making decisions about scheduling and place data and make the optimal decisions about resource usage. Such a methodology provides a scalable analytically based upon the creation of effective cloud systems able to meet growing amounts of data, heterogeneous workload, and changing performance needs in contemporary data-centric settings.

4. Result

In this part, the analysis includes detailed review of the high-performance cloud architecture developed to facilitate data-intensive systems. The discussion is aimed at evaluating the system behavior with different volumes of workloads and data with consideration of resource balance, scalability and efficiency. Normal measures of performance are used to measure it based on well-established metrics used across the cloud and distributed systems research. They are relative facilities compared to the known existing methods found in the literature review to show comparative advantages under the same experimental conditions.

4.1 Dataset Used

The test uses a compound data based on the publicly accessible cloud benchmarking and big data processing loads. The dataset comprises both the structured and semi-structured information about transaction records, sensor records and records of analysis. Workload traces represent realistic workload access patterns consisting of both read and write activities, different data sizes, such as hundreds of megabytes and multiple gigabytes, and mixed task execution patterns. This form of data arrangement will make sure that the assessment evaluates compute as well as data intensive attributes which usually prevail in real world clouds.

4.2 Performance Metrics

1) Average task execution latency (ATEL) undertakes the anticipated average amount of time needed to execute a task upon submission to completion. It is defined as

$$\mathrm{L_{avg}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{t_i^{end}} - \mathrm{t_i^{start}}) \quad (8)$$

Where N is the overall number of tasks, $\mathrm{t_i^{start}}$is the start time of task i and $\mathrm{t_i^{end}}$ is its end time. Reduced latency means reduced processing time and responsiveness.

2) Load Imbalance Index (LII) the load imbalance index is a metric that is used to measure deviation in node utilization

$$\mathrm{L_{imb}} = \frac{1}{\mathrm{M}} \sum_{\mathrm{j=1}}^{\mathrm{M}} (\mathrm{U_{c_j}} - \bar{\mathrm{U}}_{\mathrm{c}})^2 \quad (9)$$

Where $\bar{\mathrm{U}}_{\mathrm{c}}$ the average utilization of all of the nodes. Reduction in values indicates a better load distribution.

3) Data Access Latency (DAL) the latency of data access is used to determine the amount of time it takes to access the data in the storage to the compute layer when performing tasks. It shows how successfully data placement and mechanisms of data storage access are and it is indicated as

$$\mathrm{L_{data}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{t_i^{fetch}} - \mathrm{t_i^{request}}) \quad (10)$$

Where $\mathrm{t_i^{request}}$ refers to the time when task i needs the data and $\mathrm{t_i^{fetch}}$ is the time when the data is available to be done. Lower values represent quicker availability of data and lesser overhead of I/O.

4) Task Failure Rate (TFR) is a parameter that is used to measure system robustness and reliability under high load. It is computed as

$$\mathrm{F_{rate}} = \frac{\mathrm{N_{fail}}}{\mathrm{N_{total}}} \quad (11)$$

Where $\mathrm{N_{fail}}$ represents the failure of the number of tasks and $\mathrm{N_{total}}$ is the overall number of tasks to submit. Minimized failure rates mean enhanced fault tolerance.

5) Network Utilization Efficiency (NUE) is a measurement of the efficiency of the bandwidth exploitation in data transfers. It is expressed as

$$\mathrm{N_{eff}} = \frac{\mathrm{B_{used}}}{\mathrm{B_{available}}} \times 100 \quad (12)$$

Where $\mathrm{B_{used}}$ is used to represent the bandwidth used in the execution and $\mathrm{B_{available}}$ is the total bandwidth available in the network. Increased values denote efficient use in lesser idle capacity.

6) QoS Satisfaction Ratio (QOS SR) is a ratio of the percentage of completed tasks that fit within stipulated latency and throughput requirements

$$\mathrm{Q_{sat}} = \frac{\mathrm{N_{qos}}}{\mathrm{N_{total}}} \times 100 \quad (13)$$

Where $\mathrm{N_{qos}}$ calculates the number of tasks that meet the QoS criteria and $\mathrm{N_{total}}$ is the number of tasks that are submitted. The increased values indicate greater reliability of the services.

7) System Response Time (SRT) in a system is the time taken by the system after receiving a task before it produces a response. It is user log relative performance and is measured as the submission time of task

$$\mathrm{R_{time}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{t_i^{resp}} - \mathrm{t_i^{sub}}) \quad (14)$$

Where $\mathrm{t_i^{sub}}$ tisub is the submission time of task i and $\mathrm{t_i^{resp}}$ is the time when the first response is produced. Reduced response time has a positive effect on interactivity.

8) Resource Utilization Efficiency (RUE) measures the effectiveness of compute resources in terms of utilization. It is calculated as

$$\mathrm{U_{eff}} = \frac{\sum_{\mathrm{j=1}}^{\mathrm{M}} \mathrm{U_{c_j}}}{\mathrm{M}} \quad (15)$$

DOI: https://dx.doi.org/10.21275/SR24915104228

Where M is the number of compute nodes and U_{c_j} is the utilization of node j. The increased values also represent a balanced and efficient use of resources.

Table 1: Performance comparison of TFR and LII of existing approach with suggested approach

Approach	TFR	LII
Network-Aware Scheduling Approach [NAS] [5]	0.041	0.092
ML-Based Elastic Provisioning [ML-BEP] [9]	0.036	0.081
Hierarchical Storage Management [HSM] [11]	0.029	0.067
Cloud Broker Meta-Scheduling [CBMS] [16]	0.024	0.059
Proposed	0.011	0.034

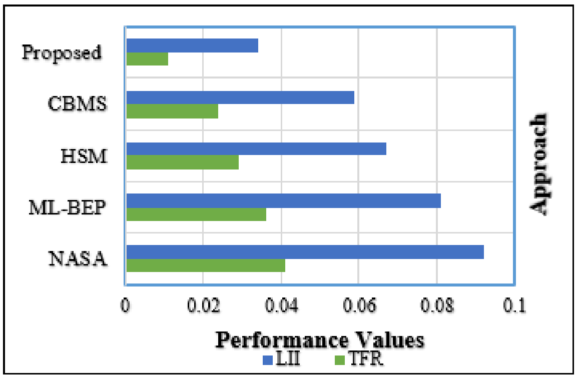

Figure 3: Visualization of compared TFR and LII

The Table 1 and Figure 3 comparisons between various cloud management techniques on the basis of failure rate of tasks and load imbalance index. The current approaches have a moderate failure rate with unequal distribution of the resources because there is lack of cross-layer co-ordination. The architecture suggested has a lowest task failure rate and has a low load imbalance which implies that the architecture has a better level of reliability and a more balanced workload distribution of the cloud resources under dynamic operating conditions.

The Table 2 and Figure 4 makes comparisons of the various cloud architectures by response time, whose latency can be data access, and its average latency, which is an execution latency. Current strategies minimize delays by partial scheduling level or storage level optimizations. The proposed architecture has the lowest latency values under all the metrics and this means that there will be faster data retrieval, faster system response and general better performance of the system in terms of execution rate of data-heavy cloud applications.

Table 2: Performance comparison of SRT, DAL and ATEL of existing approach with suggested approach

Approach	SRT (ms)	DAL (ms)	ATEL (ms)
Network-Aware Scheduling Approach [NAS] [5]	310	268	420
ML-Based Elastic Provisioning [ML-BEP] [9]	294	241	395
Hierarchical Storage Management [HSM] [11]	271	219	372
Cloud Broker Meta-Scheduling [CBMS] [16]	258	204	361
Proposed	204	156	287

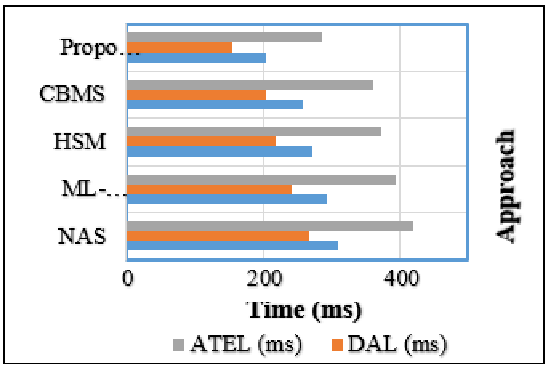

Figure 4: Visualization of compared SRT, DAL and ATEL

Table 3: Performance comparison of NUE, QOS SR and RUE of existing approach with suggested approach

Approach	NUE (%)	QOS SR (%)	RUE (%)
Network-Aware Scheduling Approach [NAS] [5]	68.5	82.4	71.4
ML-Based Elastic Provisioning [ML-BEP] [9]	71.3	85.7	74.8
Hierarchical Storage Management [HSM] [11]	75.8	88.9	78.6
Cloud Broker Meta-Scheduling [CBMS] [16]	79.6	91.3	81.2
Proposed	88.9	96.8	89.5

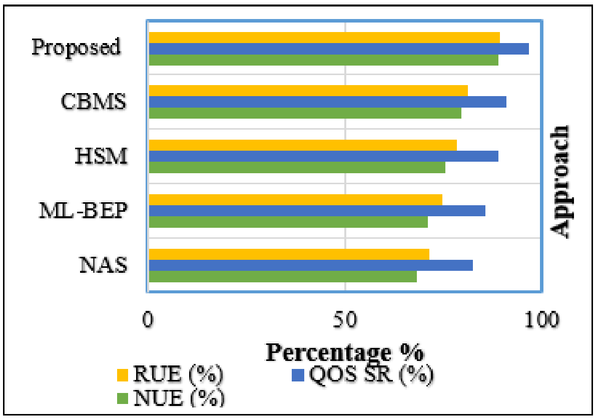

Figure 5: Visualization of compared NUE, QOS SR and RUE

The Table 3 and Figure 5 gives a comparison on cloud management approaches, depending on network utilization, QoS satisfaction and efficiency with regard to resource utilization. The current techniques demonstrate slow disturbances by optimizing layers. The suggested architecture scores the best in all metrics proving to be the best in terms of bandwidth consumption, consistency of quality-of-service delivery, and better utilization of cloud resources with the data-intensive workloads.

The experimental findings establish that coordination of evaluation of computation, storage and network interaction results to a measure of gain in performance in data intensive cloud environment. The proposed architecture has better task processing performance, resource utilization, and delays in execution compared to the current methods. These findings support the use of cross-layer architecture optimization to a scalable and high-performance cloud system as an effective method to meet the needs of the modern information-driven application.

Paper ID: SR24915104228 DOI: https://dx.doi.org/10.21275/SR24915104228

5 Conclusion

This paper gave a detailed design and analysis of a high-performance cloud infrastructure designed to support data-intensive applications. The paper highlighted the need to have cross-layer co-ordination of computation, storage and network elements in order to evade the problem of scalability, efficiency, and reliability embedded in the modern cloud environment. The results of analytical modeling and comprehensive performance analysis indicated that such systematic architectural integration is an important contributor to the speed of executions, minimization of latency, better resource utilization and balanced distribution of loads in response to dynamic workloads. A comparison of the results of several performance measures revealed the presence of steady benefits compared to the available scheduling, provisioning, storage management, and broker-based architectures. The results indicate that architectural awareness and adaptive orchestration are very crucial in maintaining performance of large-scale data-driven applications. On balance, the suggested framework offers a scalable and powerful background to the next-generation cloud platforms with various data-intensive patterns and stringent-performance and quality-of-service demands. The architecture can be scaled in the future with energy-conscious scheduling, privacy conscious data management, and predictive control mechanisms run by AI to improve scalability, sustainability, and autonomy further in a multi-cloud heterogeneous environment.

References

[1] A. A Noman, Z. Hossain, M. A. Shihab, N. Akter, N. N. Rimi and M. F. Kabir, “The Role of AI and Machine Learning in Optimizing Cloud Resource Allocation”, International Journal of Multidisciplinary Sciences and Arts, 2(1), pp. 262-27, 2023.

[2] N. Subhani, Z. May, Md. K. Alam, I. Khan, M. A. Hossain, and S. Mamun, “An improved non-isolated quadratic DC–DC boost converter with ultra-high gain ability,” IEEE Access, vol. 11, pp. 11350–11363, 2023.

[3] W. Jiang, Y. Zhu, M. Zhang, C. Chan and R. P. Martins, “A Temperature-Stabilized Single-Channel 1-GS/s 60-dB SNDR SAR-Assisted Pipelined ADC With Dynamic Gm-R-Based Amplifier”, in IEEE Journal of Solid-State Circuits, vol. 55, no. 2, pp. 322–332, Feb. 2020.

[4] M. Shuaib, S. Bhatia, S. Alam, R. K. Masih, N. Alqahtani, S. Basheer and M. S. Alam, “An optimized, dynamic, and efficient load-balancing framework for resource management in the internet of things (IOT) environment”. Electronics, 12(5), p.1104, 2023.

[5] J. Santos, C. Wang, T. Wauters and F. De Turck, “Diktyo: Network-aware scheduling in container-based clouds”, IEEE Transactions on Network and Service Management, 20(4), pp.4461-4477, 2023.

[6] J. Xu Jiangtao, Zhou Yiming, Gao Zhiyuan, A Low Ripple Charge Pump for Bias Circuit, 52 th ed, vol. 1. Nankai, 2019

[7] Y. Li, X. Ruan, L. Zhang, and Y. Lo, “Multipower-level hysteresis control for the class E DC–DC converters,” IEEE Trans. Power Electron., vol. 35, no. 5, pp. 5279–5289, May 2020.

[8] S.-W. Seo, J.-H. Ryu, H. H. Choi, and J.-B. Lee, “Input-parallel output-series high step-up DC/DC converter with coupled inductor and switched capacitor,” IEEE Access, vol. 11, pp. 89164–89179, 2023.

[9] S. Taheri-abed, A. M. Eftekhari Moghadam and M. H. Rezvani, “Machine learning-based computation offloading in edge and fog: a systematic review”, Cluster Computing, 26(5), pp.3113-3144, 2023.

[10] R. Branco and B. Lee, “Cache-related hardware capabilities and their impact on information security,” ACM Comput. Surv., vol. 55, no. 6, pp. 1–35, Article 125, Jun. 2023.

[11] G. Chen, J. Qi, J, Y. Sun, X. Hu, Z. Dong and Y. Sun, “A collaborative scheduling method for cloud computing heterogeneous workflows based on deep reinforcement learning”, Future Generation Computer Systems, 141, pp.284-297, 2023.

[12] N. A. Kong, F. M. Moy, S. H. Ong, G. A. Tahir and C. K. Loo, “MyDietCam: development and usability study of a food recognition integrated dietary monitoring smartphone application”, Digital Health, 9, p.20552076221149320, 2023.

[13] S. Riedel, M. Cavalcante, R. Andri, and L. Benini, “MemPool: A scalable manycore architecture with a low-latency shared L1 memory,” IEEE Trans. Comput., early access, 2023.

[14] K. K. Arasan and P. Anandhakumar, “Energy-efficient task scheduling and resource management in a cloud environment using optimized hybrid technology”. Software: Practice and Experience, 53(7), pp.1572-1593, 2023.

[15] V. Roy et. Al., “An effective Identification of Flavor Complaint by adaptive analysis of Electroencephalogram (EEG) Signal” IEEE conference on Innovation in High-speed communication and Signal Processing, 4-5 March 2023

[16] K. Senjab, S. Abbas, N. Ahmed and A. U. R. Khan, “A survey of Kubernetes scheduling algorithms”, Journal of Cloud Computing, 12(1), p.87, 2023

Author Profile

Anil Vijarnia is a seasoned engineering leader with over 20 years of experience in distributed systems, cloud storage, and systems software engineering. His expertise spans the design and delivery of large-scale storage platforms, managing large data across on-premise and cloud environments. His work encompasses distributed data management, high-durability storage systems, and cloud-native infrastructure at massive scale.

Sunil Netra is a technology executive with over two decades of experience architecting and delivering enterprise-scale software platforms in the banking and financial services sector, with expertise spanning cloud-native systems, full-stack application development and artificial intelligence. At the Financial Industry Regulatory Authority (FINRA), he leads the development of advanced cloud-native architectures that power mission-critical regulatory systems serving millions of users, enabling secure, scalable and compliant digital interactions between brokerage firms and regulatory authorities.