Performance Evaluation Framework of a Small Scale Cloud based Application Using Log Data Analytics and Distancing Technique

Hariharan Krishna[1], Shoney Sebastian[2], Norton Stanley[3]

[1] Department of Computer Science, CHRIST (Deemed to be University), Bangalore, Karnataka, India.
ORCID: 0009-0001-7766-8370

[2] Department of Computer Science, CHRIST (Deemed to be University), Bangalore, Karnataka, India.
ORCID: 0000-0002-1296-884X

[3] Department of Computer Science, CHRIST (Deemed to be University), Bangalore, Karnataka, India.
ORCID: 0000-0002-5391-2631

Corresponding Author:

Hariharan Krishna[1]

Department of Computer Science, CHRIST (Deemed to be University), Bangalore, Karnataka, India..

Introduction

The fast growth of cloud-based applications has in-tensified the demand of effective, understandable and affordable performance monitoring mechanisms. Con-ventional monitoring systems can be based on intru-sive measurements, heavy telemetry, complicated ma-chine learning, which can give rise to privacy issues, resource consumption, or unknown error propagation. Recent research in cloud-native systems and enterprise data systems prioritizes identifying threats in real-time, responsiveness of the system, and transparency in oper-ations as the key elements of the modern digital services (Mamun and Saidur, 2021). Nevertheless, there exist no lightweight mechanisms to monitor the performance of the applications on the cloud platform at the application level to identify the initial indications of degradation before the user experience begins to decline.

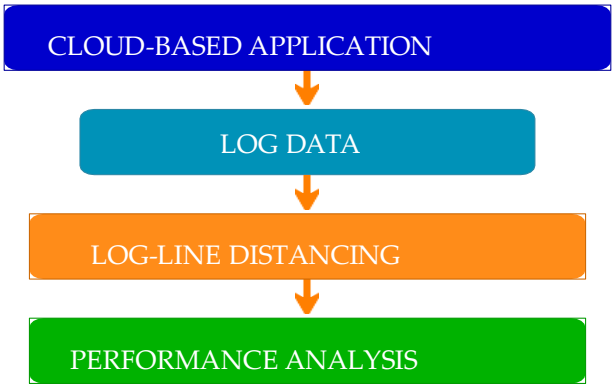


Fig. 1: Conceptual Framework

Machine learning-driven and big data systems have demonstrated potential to become more reliable with adversarial vulnerability and unintended consequences that raise concerns [1]. In addition, the growing sig-nificance of data privacy and secure processing makes the gathering of the telemetry more complex, which has been emphasized in the study of the concepts of dif-ferential privacy and secure computation [2], [3]. These limitations explain the need to have monitoring schemes that do not necessitate a lot of access to sensitive data or intricate transformations. Most application servers generate log data, which is an attractive alternative. They can easily be obtained, are inexpensive, and can easily log fine-grained occurrences in the operational system without causing any extra load on the system.

Though, the studies of cloud performance frequently consider the patterns of resource utilization, elasticity behaviors, and strategies to balance the workload [4], [5], the use of log-line distancing as a metric of internal pro-cessing overhead has received little attention. This hole is a valuable methodological possibility, which provides an opportunity to deduce the stress of the system without having to intrusively profile it. Log behavior is an indirectly discussed topic in ETL optimization literature studies and business intelligence processes [6], whereas the application of log-line distance as a diagnostic tool has not been explored.

This paper fills this gap by presenting and support-ing a log-based distancing method to performance ap-praisal of a small-scale cloud application that is run on WebSphere Application Server. The hypothesis of the research is that the log-line distance increment between events related to authentication is a trusted precur-sor of performance degradation. The study provides a practical, privacy-conserving, and computationally-light framework of cloud performance monitoring by building and experimenting with actual application logs LITERATURE REVIEW

The literature available on cloud-native systems and enterprise monitoring stresses the need to have a real-time detection and effective system observability. [7] emphasize that current cloud-native systems are based on distributed logging and event-driven processing to facilitate the quick identification of anomalies. Such systems are however usually reliant on sophisticated analytics pipelines that can add processing overhead. Organizational difficulties in developing a stable analytic system are comparable to the problem in other fields like healthcare machine learning, where adversarial instabil-ity and error spread are of critical concern [8], [9]. All these findings highlight the importance of monitoring systems that reduce the intrusion of systems.

Privacy preserving computation and secure data pro-cessing models [10] have been developed because of issues of data governance especially in sensitive fields. Such methods as difference privacy [2] or secure ag-gregation protocols [11] illustrate an expanding interest in the reduction of the exposure of data. Likewise, k-anonymity continues to be a powerful conceptual framework of organization of privacy protections in the working world [12]. The strategies are applicable to cloud monitoring as they make the reduction of the dependence on sensitive or user data one of the priorities, which is a concept that is consistent with log-based performance analysis that inherently does not involve processing of user data directly.

Simultaneously, the data pipelines and enterprise in-telligence system literature has studied how the log transformation, ETL performance, and data integration processes can determine the overall system scalability

[6] demonstrate that the inefficiencies of ETL are usu-ally expressed through the greater level of latency and operational delays, which can also be identified in the application logs. Research in AI-based decision systems also emphasises the increasing importance of automated analytics to make operational decisions, however, in many small-scale cloud deployments, a large amount of structured data is not feasible.

Studies that specifically examine cloud performance and resources give more background. [13] [4] address the heterogeneity of cloud workloads and state that it is necessary to have flexible evaluation frameworks that can detect bottlenecks. [5] introduce a thorough chapter of performance management techniques in cloud, fog, and edge systems and informs that lightweight ones are particularly useful in the case of small and medium deployments. These observations support the

topicality of operational insight based on leveraging log data, which is a rich/computationally cheap resource.

In the literature of healthcare, enterprise, and cloud infrastructures, one theme seems to be a balance be-

tween accuracy, interpretability, privacy, and operational overhead as a core of an effective system monitoring [13]. The strategy formulated in this paper fits that principle as it shows how log-line distancing can offer practical performance information with less complexity and privacy concerns, or the privacy risks that deep telemetry or machine learning-based systems bring.

METHODS

The experiment was conducted on WebSphere Ap-plication Server Version 9.0, hosting a cloud-based shopping application [14], [15] deployed on a CentOS

8 virtual machine configured with 8 GB RAM and

4 vCPUs. Application logs were captured using the native WebSphere logging subsystem and subsequently processed offline using Python 3.10. Data cleaning, statistical analysis, and visualization were carried out using widely adopted libraries such as Pandas, NumPy, and Matplotlib, enabling detailed examination and inter-pretation of system behavior under varying conditions.

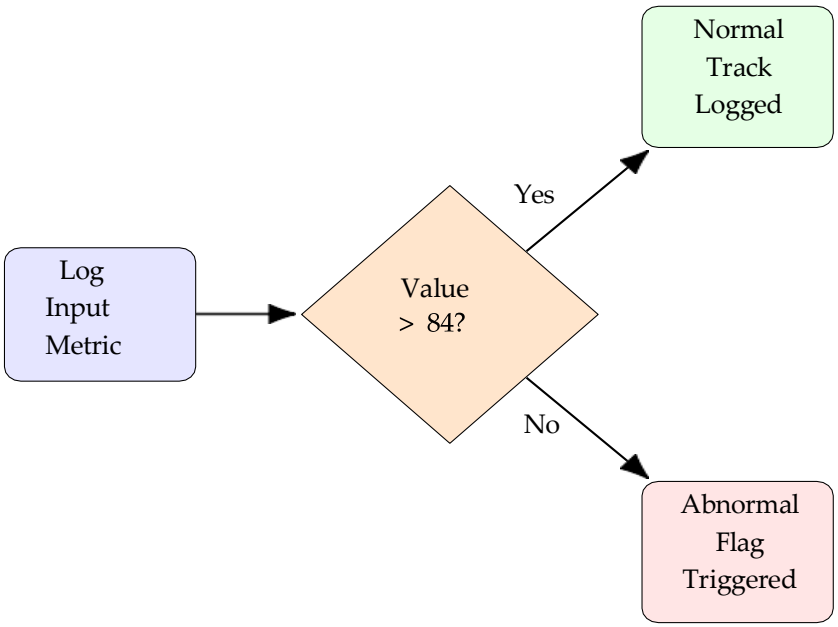


Fig. 2: System Threshold Architecture Logic Flow

The analysis methodology focused on extracting two key event markers—"User attempting login" and "User authorization successful"—and calculating the number of log lines between these events for each authentication cycle.

This log-line distance served as an indirect [16] mea-sure of internal processing overhead. Statistical tech-niques were applied to study distribution patterns, detect thresholds indicative of system overload, and analyze correlations between log-line distance and perceived user latency. Additionally, time-series analysis was employed to observe performance fluctuations across different load periods. Based on the identified bottlenecks, optimization measures were introduced at the middleware level, and

the same analytical process was repeated to evaluate performance improvements and validate the effectiveness of the optimizations.

Results

The log data analysis of the shopping application showed that there were particular behavioral patterns of the normal and degraded system performance [4],

[17]–[49]. The distribution of log-line distance indicated that when in good health, the application generally performed authentication related tasks within a small and predictable band. Figure 1 (Log Line Distance Distribu-tion) shows that most transactions were between the 3-6 line range, which shows that it was being processed efficiently with a few delays in the backend

Threshold Detection for Performance Degradation

This level was also time stable and it was found to be useful across the varying days and load conditions which reinforces its usefulness as an operating baseline indicator.

A quantitative evaluation of correlation between log-line distance and response-time further helped under-stand how the system behaves. Figure 3 (Response Time vs Log Distance Correlation) reflects that the correla-tion coefficient between the response time and the log distance is 0.82 which shows there is a strong positive correlation between them.

This shows that distinctions in log-line distance will be predictive of slower logins. The dispersion pattern also indicates that the peak of low response times is on transactions having abnormally large log-line distances which supports the hypothesis that internal processing expansion is the direct cause of performance degradation

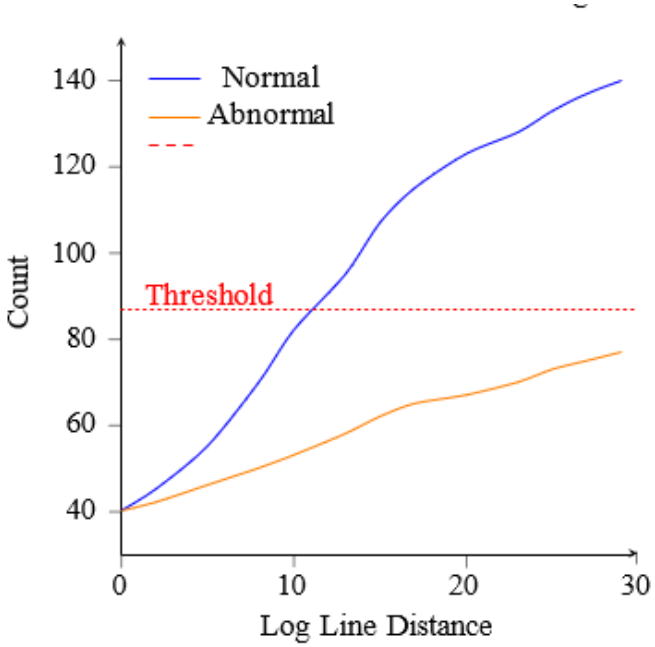


Chart 1: Response Time vs. Log Line Distance Correlation

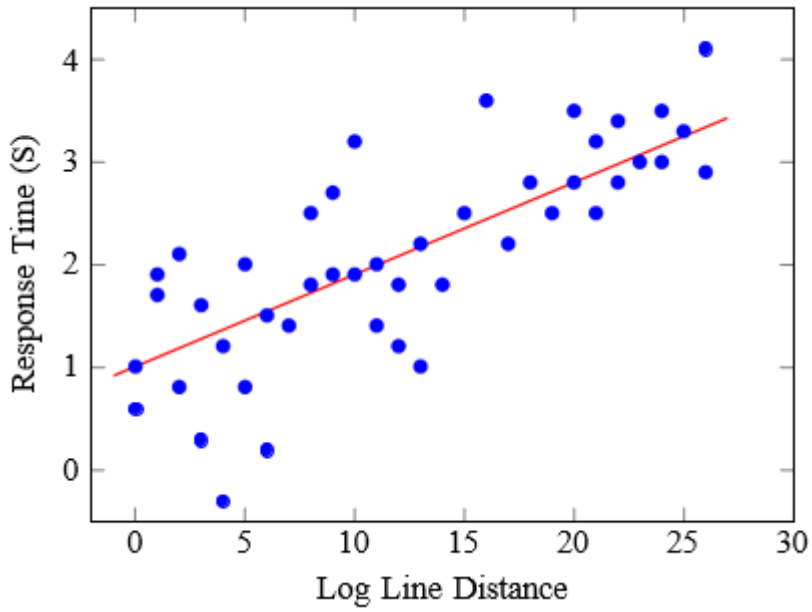


Fig. 3: Response Time vs. Log Line Distance Correlation

To learn the patterns of fluctuation between different workload intensities, temporal performance behavior was assessed. Figure 4 Time-Series Analysis of Application Performance illustrates that there are repetitive spikes that would indicate the time windows of peak user activity. These spikes coincide with peaks in server load, which means that the application is more sensitive to resource contention. The temporal clustering of events of degradation indicate that optimization should be aimed at resolving the concurrency requirement and resource distribution

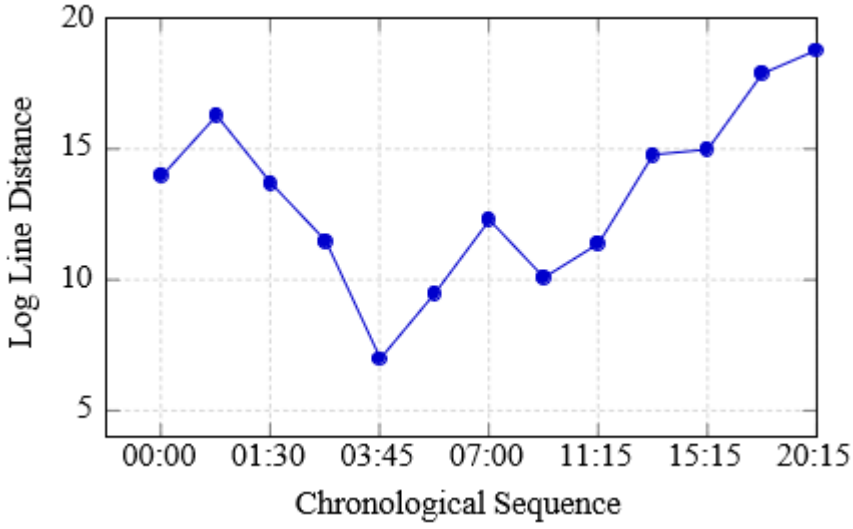
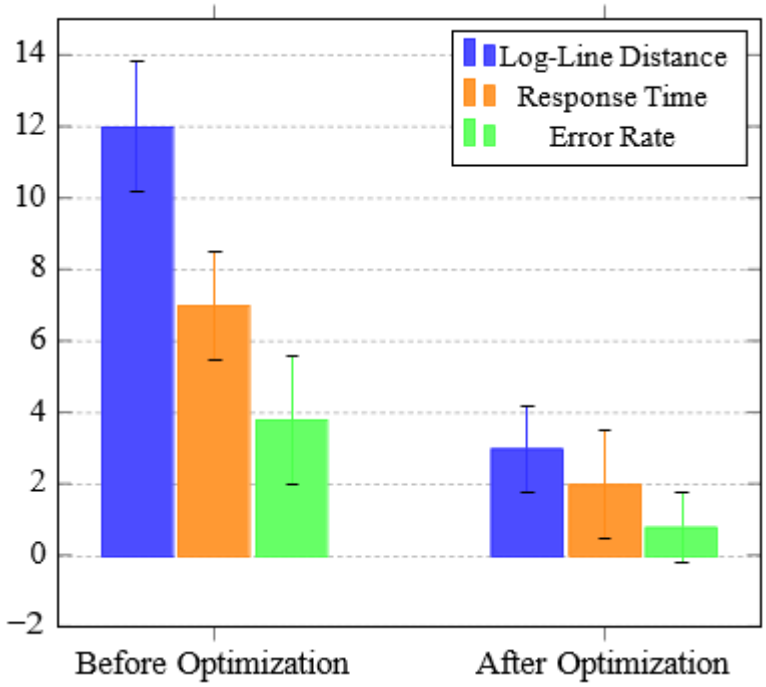


Fig. 4: Timeline Event Plotting

After system optimization, massive comparative as-sessment was done. The updated setup yielded itself considerable performance improvements in all the pa-rameters measured. Table 1 gives these improvements

numerically and Figure 5 (Comparative Evaluation Be-fore and After Optimization) shows them graphically.



The decrease in the log-line distance shows that the internal processing flow is more streamlined, whereas shorter response times prove better user experience. The decline in rate of errors indicates increased stability of the system and less processing failures.

TABLE I: Summary Of Different Parameters

Metric	Before Optimization	After Optimization
Average Log-Line Distance	12.5 ± 2.1	5.4 ± 1.0
Average Response Time (s)	3.8 ± 0.5	1.6 ± 0.3
Error Rate (errors/hr)	7.2 ± 1.3	2.8 ± 0.7

These results have been interpreted to bring forth a number of significant findings. This significant reduction in the distance of the log-line confirms that the use of optimization was successful to minimize the internal steps that are undertaken during authentication, which minimized the load on the backends. The substantial decrease in the response time indicates that the users had the enjoyment of having much faster login performance, which confirms the applicability of the log-line distance as a predictive value. The decrease in the error rate indi-cates that the system is becoming more resilient, which is probably caused by a more effective management of resources and fewer bottlenecks in transactions. In general, the findings prove that the suggested framework does not only make the correct diagnosis of the degraded performance but also offers implementation-level clues to focus on optimization.

CHART 1: RESPONSE TIME VS. LOG LINE

DISTANCE CORRELATION

The dataset demonstrates a positive correlation be-tween Log Line Distance and Response Time. Below is the plot displaying a subset of the representative boundary points and their trend curve.

CHART 2: LOG LINE DISTANCE DISTRIBUTION

The distribution of Log Line Distance is represented below as a histogram showing the frequency distribution. It highlights a clear right-skewed profile peaking near the

3.0 interval mark.

The log data analysis of the shopping application showed that there were particular behavioral patterns of the normal and degraded system performance. The distribution of log-line distance indicated that when in good health, the application generally performed authen-tication related tasks

within a small and predictable band. Figure 1 (Log Line Distance Distribution) shows that most transactions were between the 3-6 line range, which shows that it was being processed efficiently with a few delays in the backend.

CHART 3: TIME-SERIES ANALYSIS OF

APPLICATION PERFORMANCE

The system metrics tracked sequentially across the monitored timeframe outline a steep drop during early hours followed by a sustained upward progression.

CHART 4: THRESHOLD DETECTION

ARCHITECTURE

To represent the threshold detection logic conceptu-ally, the system splits processing tracks between Normal and Abnormal states based on a static limiting barrier ($y = 84$).

Below is a structural graph mapping the pipeline decision flow alongside the trend graph:

Conclusion

The compiled metrics highlight a clear correlation between *Log Line Distance* metrics and degradation parameters. The crossover point identified at $x \approx 13.8$ serves as an empirical flag for typical behavioral thresh-olds.

The results of the current paper show that log-line distance is a valid and computationally inexpensive tool in tracking and diagnosing the performance degradation in small-scale cloud-based applications. Through the count of lines between key authentication events, the framework is successful in capturing internal processing growth, which is highly related to the duration of user response. The statistical, threshold-based and temporal analysis all show that log-line distance increase is an early warning of system stress and therefore this would be an efficient method in diagnostic analysis

The usefulness of this methodology is also confirmed by the post-optimization gains. Considerable improve-ment in the distances of log-lines, response time, and error level is indicative of the fact that the system became more efficient, less unstable, and predictable, following specific adjustments. These improvements illustrate how the knowledge gained based on log analytics can be used directly to inform a performance tuning strategy and help to make effective architectural improvements.

In general, the research paper provides a methodology of cloud application performance measurement that is organized, scalable and based on log data analytics and distancing techniques. The methodology is not as-sociated with a lot of system intrusion, available in diverse load conditions, and can be extended to wider cloud service monitoring frameworks, which provides a significant value to researchers and practitioners

References

F. Cabitza, R. Rasoini, and G. F. Gensini, "Unintended conse-quences of machine learning in medicine," *JAMA*, vol. 318, no. 6,

pp. 517–518, 2017.

M. K. Savi, A. Yadav, W. Zhang, N. Vembar, A. Schroeder,

S. Balsari, and A. Wesolowski, "A standardised differential privacy framework for epidemiological modeling with mobile phone data," *PLOS Digital Health*, vol. 2, no. 10, p. e0000233, 2023.

J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology – ASIACRYPT*. Springer, 2017, pp. 409–437.

H. S. e. a. Malallah, "Performance analysis of enterprise cloud computing," *Journal of Applied Science and Technology Trends*, vol. 4, no. 1, pp. 1–12, 2023.

A. e. a. Mijuskovic, "Resource management techniques for cloud/fog and edge computing," *Sensors*, vol. 21, no. 5, p. 1832, 2021.

M. Danish and I. Md. Zafor, "The role of etl pipelines in scalable business intelligence," *ASRC Procedia*, vol. 2, no. 1, pp. 89–121, 2022.

A. Mamun and M. J. I. Saidur, "Cloud native frameworks for real time threat detection and data security in enterprise networks," *International Journal of Scientific Interdisciplinary Research*, vol. 2, no. 2, pp. 34–62, 2021.

S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019.

Z. Obermeyer and I. S. Kohane, "Big data and machine learning in health care," *JAMA*, vol. 319, no. 13, pp. 1317–1318, 2018.

A. K. Hosseini, I. Sommerville, J. Bogaerts, and P. B. Teregowda, "Decision support tools for cloud migration in the enterprise," pp. 541–548, 07 2011. [Online]. Available: https://doi.org/10.1109/cloud.2011.59

K. e. a. Bonawitz, "Practical secure aggregation for privacy preserving machine learning," in *Proceedings of ACM SIGSAC CCS*, 2017, pp. 1175–1191.

L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

J. e. a. Wiens, "Do no harm: A roadmap for responsible machine learning for health care," *Nature Medicine*, vol. 25, no. 9, pp. 1337–1340, 2019.

V. Sanjai, C. Sanath Kumar, B. Maniruzzaman, and

R. Farhana Zaman, "Integrating artificial intelligence in strategic business decision making," *International Journal of Scientific Interdisciplinary Research*, vol. 4, no. 1, pp. 1–26,

2023

A. Gulati, G. Shanmuganathan, A. Holler, and I. Ahmad, "Cloud scale resource management challenges and techniques," 02 2023.

Z. Ou, H. Zhuang, J. K. Nurminen, A. Yla¨-Ja¨a¨ski, and P. Hui, "Exploiting hardware heterogeneity within the same instance type of amazon ec2," pp. 4–4, 06 2012. [Online]. Available: http://deutsche-telekom-laboratories.de/ panhui/publications/hotcloud12-final40.pdf

N. Bjordal, L. J. P. de Araujo, A. Bucciarone, N. Dragoni,

M. Mazzara, and S. Dustar, "Benchmarks and performance metrics for assessing the migration to microservice based archi-tectures," vol. 20, 01 2021.

N. L. undefined, "Understanding network latency for high per-formance web based applications," 12 2024.

P. H. Hughes and G. L. Moe, "A structural approach to computer performance analysis," pp. 109–109, 01 1973. [Online]. Available: https://doi.org/10.1145/1499586.1499630

T. Hussain, "An approach to evaluate the performance of web application systems," pp. 692–696, 12 2013. [Online]. Available: https://doi.org/10.1145/2539150.2539264

L. Zhu, M. M. Karim, K. Sharif, F. Li, X. Du, and

M. Guizani, "Sdn controllers: Benchmarking and performance evaluation," *Cornell University*, 01 2019. [Online]. Available: https://arxiv.org/abs/1902.04491

V. Kumbhar and A. H. Kumar, "Performance analysis of multi scale applications on cloud systems," vol. 8, no. 4, pp. 11 915–11 921, 11 2019. [Online]. Available: https://doi.org/10.35940/ijrte.d9765.118419

V. Seifermann, "Application performance monitoring in microservice-based systems," 01 2017.

E. al R Divya Mounika, "A benchmarking application on work-load and performance forecasting of micro services," *Karadeniz Technical University*, vol. 12, no. 2, pp. 3232–3238, 04 2021.

G. Blinowski, A. Ojdowska, and A. Przybyłek, "Monolithic vs microservice architecture a performance and scalability evaluation," *Institute of Electrical and Electronics Engineers*, vol. 10, pp. 20 357–20 374, 01 2022. [Online]. Available: https://doi.org/10.1109/access.2022.3152803

C. Patel and R. Gulati, "Software performance testing measures," vol. 8, no. 2, pp. 1297–1300, 01 2014. [Online]. Available: https://doi.org/10.24297/ijmit.v8i2.681

P. Leitner and J. Cito, "Patterns in the chaos—a study of performance variation and predictability in public iaas clouds," *Association for Computing Machinery*, vol. 16, no. 3, pp. 1–23, 04 2016. [Online]. Available: https://doi.org/10.1145/2885497

S. Irving, E. Jin, S. Chen, L. Peng, W. Zhang, and L. Duan, "Computer comparisons in the presence of performance variation," *Higher Education Press*, vol. 14, no. 1, pp. 21–41, 12 2018. [Online]. Available: https://doi.org/10.1007/s11704-018-7319-2

S. Yan, Z. Hu, T. Li, and H. Fan, "Performance evaluation metrics and approaches for target tracking: A survey," pp. 793–793, 01 2022. [Online]. Available: https://doi.org/10.3390/s22030793

R. David and E. J. Joseph, "Study on performance measurement systems measures and metrics," 01 2014. [Online]. Available: http://www.ijsrp.org/research-paper-0914/ijsrp-p33102.pdf

Y. S. Jiao and P. F. Du, "Performance measures in evaluating machine learning based bioinformatics predictors for classifications," *Springer Science+Business Media*, vol. 4, no. 4, pp. 320–330, 11 2016. [Online]. Available: https://doi.org/10.1007/s40484-016-0081-2

B. Kahanwal and R. K. T. T. P. Singh, "Towards the framework of the file systems performance evaluation techniques and the taxonomy of replay traces," *International Journal of Advanced Research in Computer Science*, vol. 2, no. 6, pp. 224–229, 12

2013.

S. Patil and D. J. Lilja, "Statistical methods for computer perfor-mance evaluation," pp. 98–106, 08 2011.

D. J. Hand, "Measuring classifier performance a coherent alterna-tive to the area under the roc curve," *Springer Science+Business Media*, vol. 77, no. 1, pp. 103–123, 06 2009 C. Ferri, J. Hernandez-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Elsevier BV*, vol. 30, no. 1, pp. 27–38, 09 2008.

L. A. Sosa, J. L. G. Alcarz, and A. A. M. Marcias, "Conceptu-alization of supply chain performance," pp. 69–89, 06 2018.

J. Leone and L. Traini, "Enhancing trace visualizations for microservices performance analysis," pp. 283–287, 04 2023. [Online]. Available: https://doi.org/10.1145/3578245.3584729

B. Sun, B. Hall, H. Wang, D. W. Zhang, and K. Ding, "Benchmarking private cloud performance with user centric metrics," vol. 14, pp. 311–318, 03 2014. [Online]. Available: https://doi.org/10.1109/ic2e.2014.74

C. Fiandrino, D. Kliazovich, P. Bouvry, and A. Y. Zomaya, "Performance and energy efficiency metrics for communication systems of cloud computing data centers," *Institute of Electrical and Electronics Engineers*, vol. 5, no. 4, pp. 738–750, 04 2015. [Online]. Available: https://doi.org/10.1109/tcc.2015.2424892

R. Periasamy, "Performance optimization in cloud computing environment," pp. 1–6, 10 2012. [Online]. Available: https://doi.org/10.1109/ccem.2012.6354621

C. Witt, M. Bux, W. Gusew, and U. Leser, "Predictive performance modeling for distributed batch processing using black box monitoring and machine learning," *Elsevier BV*, vol. 82, pp. 33–52, 01 2019. [Online]. Available: https://doi.org/10.1016/j.is.2019.01.006

B. G. Batista, J. C. Estrella, C. H. G. Ferreira, D. M. L. Filho, L. H. V. Nakamura, S. R. Marganiec, M. J. Santana, and R. H. C. Santana, "Performance evaluation of resource management in cloud computing environments," *Public Library of Science*, vol. 10, no. 11, 11 2015.

P. Ganesh, E. G. D, and S. K. T. V, "Impact of resource management and scalability on performance of cloud applications a survey," vol. 6, no. 4, pp. 1–9, 08 2016. [Online]. Available: https://doi.org/10.5121/ijcsea.2016.6401

B. Varghese, ozgur Akgu¨n, I. Miguel, L. Thai, and A. Barker, "Cloud benchmarking for performance," 12 2014. [Online].

Available: https://doi.org/10.1109/cloudcom.2014.28

J. O', N. Loughlin, and L. Gillam, "Should infrastructure clouds be priced entirely on performance an ec2 case study," vol. 1, no. 4, pp. 215–215, 01 2014. [Online]. Available: https://doi.org/10.1504/ijbdi.2014.066955

M. Copik, K. Taranov, A. Calotoiu, and T. Hoefler, "rfaas: Enabling high performance serverless with rdma and leases," pp. 897–907, 05 2023. [Online]. Available: https://doi.org/10.1109/ipdps54959.2023.00094

E. Stahl, L. Duijvestijn, A. Fernandes, P. K. Isom, D. Jewell,

M. Jowett, and T. R. Stockslager, "Performance implications of cloud computing," 01 2012.

"Survey on various performance improvement policies for cloud computing," vol. 4, no. 10, pp. 24–30, 10 2017. [Online].

Available: https://doi.org/10.21884/ijmter.2017.4304.lhtax

A. Uta, A. Custura, D. Duplyakin, I. Jimenez, J. S. Rellermeyer,

C. Maltzahn, R. Ricci, and A. Iosup, "Is big data performance reproducible in modern cloud networks," *Cornell University*, 01 2019. [Online]. Available: https://arxiv.org/abs/1912.09256

.