SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

SIRS (Student Involvement Report System) Case Study – Studies on QOS (Quality of Service) Perimeters, Daily Error Types and Activities Within the Server

Mohd Ikhsan Md Raus[1,*], Juhaida Ismail[1], Atilla Elci[2], Mohd Eizan Shafiq Abd Aziz[1], Mohd Khairul Ikhwan Zolkefley[1], Ahmad Kamal Ramli[1], Aisyah Mat Jasin[1]

[1] Department of Computing, College of Computing, Informatics and Mathematics, Universiti Teknologi MARA (UiTM), Bandar Jengka, 26400 Bandar Tun Razak, Pahang, Malaysia
[2] Department of Software Engineering, Hassan Kalyoncu University, Yeşilkent Mahallesi Havaalanı Yolu Üzeri, 8.km, Şahinbey, 27410, Gaziantep, Türkiye

ARTICLE INFO

Article history:
Received 11 June 2024
Received in revised form 4 October 2024
Accepted 7 October 2024
Available online 18 November 2024

Keywords:
SLA; QoS; MAPE-K; Autonomic computing; RDBMS

ABSTRACT

Relational Database Management System (RDBMS) is commonly applied in Small, Medium, and Enterprise businesses. In this research, we investigated Quality of Service (QoS) perimeters over day-to-day operations in RDBMS and how the running process able to be expedited with the adaptation and combination of autonomic computing approach using MAPE-K (Monitor, Analyse, Plan, Execute, and Knowledge Base) and Fuzzy rules. The core function of the system is to monitor student attendance, assessment, and lastly escalation of identified student issues to the department of academic. In a nutshell, it is called the Student Involvement Report (Sir). Two segments have been analysed and focused on errors in request time out, bandwidth, hits, and user activities. The outcomes of this are significant on activities that related to the peak of academic activity, due to the submission of student reports and escalation among departments. The MAPE-K and Fuzzy rules can capture the scenarios and identify rules that are significantly important to reduce and enhance the Request Time Out and Hits activities.

1. Introduction

Relational Database Management System (RDBMS) is an application that caters to database application servers, such as transaction, backup, storage, and other related services. Some of the examples are MySQL, MSSQL, PostgreSQL, Oracle, and IBM DB2. All mentioned applications are

* *Corresponding author.*
E-mail address: mohdikhsan@uitm.edu.my

https://doi.org/10.37934/araset.531.209218

highlighted from the student environment to the corporate working environment. In a nutshell, the users are either motivated by the application from their studies or because of the features developed continuously inside that software.

In our early development, due to the remote access from academic users, the transactions were fractured with lots of uncertainty. On the early findings, we discovered cache, session, maximum file upload, maximum file size, and single sign-on are among the vital issues that contributed to our concerns. To confirm our findings, this paper will have Service Level Agreement (SLA) and Quality of Service (QoS) combinations to justify the discovered elements.

A typical SLA sets out:

i. A description of all the service levels that the supplier commits to meet or exceed.
ii. How those service levels will be measured and reported on
iii. Corresponding remedies if one or more service levels are unmet.
iv. A process for changing or removing service levels.

Quality of Service (QoS) is a key factor for a viable Service Level Agreement (SLA) which ensures the formation of a reliable provider-consumer relationship in cloud computing. SLA ties up a service provider with a service consumer on agreed upon terms and conditions so that the provider's promised services and the consumer's requirements are fulfilled, which results in a trustful relationship between them. In this paper, an analysis of the existing literature is made to identify the gaps that need to be filled to establish a trustful relationship. The literature is organized as the QoS potential attributes. Such as trust, resource management, risk management, security, task scheduling, and performance. The overall contributes to maintaining a feasible SLA.

2. Literature Reviews

In this section, there are a few related issues highlighted by researchers. On the fault tolerance and high availability, it applies in geographic zones on every partition in the database and nodes. It is to ensure that high availability is always present through automatic recovery mechanisms [1]. Cheetah and NetAccel studied by [2], are both deployed in between the work and the master server, whereas Jumpgate stands between the storage engines. It then resulted in getting higher compute nodes. It is just merely on filtering and partial aggregation and cannot cope with packet loss. Cheetah is a query processing system that partially offloads queries to serve managed switches.

The policy is very crucial to ensure QoS are placed and working as expected in specific network environments. The policies and regulations need to be enforced to ensure the conditions accordingly and not just access to the services. It specifies how the data can flow and is applied by different users, applications, segments, and other dependents. Some of that might require declassification, revealing some information that is connected to the running services. Proposed services such as contextual data flow policies, and granular policies for complete mediation and enable a clean separation between the policy specification, the application, and its underlying services [3]. It is also connected to admission control [8] to manage the fluid limit.

In [4], discussed the importance of Segment Routing (SR), it is an architecture based on the source routing paradigm that seeks the correct coordination between intelligence and centralized programmability by steering packets through and instructions. It is an evolution of MPLS, an open technology for label switching technologies. Data is known now in research as new 'oil' [4] and it is very related to configuration and processing. With great combinations, it will lead to tons of advancement in information related and the continuous uses of advanced technologies as well, such as CPU [6] and databases [5].

A rich and dynamic SLA formalism is the key to handling the cloud service context sensed and enforcing the right SLA management and modifications [9]. It is an adaptable model with new and enhanced. Multi-objective optimization (MOD) methods can achieve a good balance among multiple targets which satisfies the offered services [10].

Quality QoS is very important to understand the outlier issues [11] and perform better predictions on the performance. Such as in medical health [12], green computing [13], logic bugs in databases [14], and lastly validation on SLA breaches [16]. The establishment of clear SLA characteristics leads to a better understanding of the Service Level Objective (SLO) on the specific domain, and this guarantees the implementation of QoS in desired targets [16].

In a nutshell, the combination of previously implemented research work focused on database performance, node, query processing time, and QoS. This information is very helpful to progress in this work to measure the importance of QoS parameters which are then related to Relational Database Management Software (RDBMS). Tons of queries and data are stored within SIRS execution, and it is very important to undergo thorough assessments.

3. Methodology

The research proposal is divided into 5 phases to achieve the highlighted objectives.

3.1 Phase 1: Running Logs Acquisition

Identify and acquire running logs generated by the network operating system. This implementation is referred to as Microsoft Server 2020. It is the composition of server performances, which are downtime, uptime, peak transactions, off-peak transactions, and response. The acquired data will be subjected to processing to verify the quality of the materials and later proceed for assessment and evaluation.

3.2 Phase 2: Modality Exploration and Extraction

At this phase, the sentiment of the extracted modality is assessed and able to feed the state status as the base result for an adaptive framework. The state is a result generated by self-attention. Based on that result, it will be known as a State in the MAPE-K framework. The state itself will go through four elements in MAPE-K, which are Monitor, Analyse, Plan, Execute, and lastly Knowledge Base. Figure 1 illustrates the generic MAPE-K framework and Q-Learning is explained in Figure 2. The Self-Attention is represented by Figure 3. Every movement of the state in the framework will be processed by each element to justify the state condition.

3.3 Phase 3: Adaptive Framework

The quality of results from Phase 2 was assessed and evaluated as the state and matched with the set of predefined rule bases. This rule base is the pre-set of quality for mixed messages condition. Something like if the result from self-attention is good, then mixed messages are good, or if the result from self-attention is uncertain, then mixed messages will be neutral. The conditions are not limited to highlighted results.

3.4 Phase 4: Experiment Evaluation

The evaluation is based on the result from Phase 3, which are the components of MAPE-K adaptive framework supported with rule base elements. The quality of output enhanced the decision, and it adaptively interacts with previous results for reference and enhancement. That is stored in the knowledge base as part of the MAPE-K element.

3.5 Phase 5: Result Analysis and Updating of Proposed Research

Based on the results from Phase 4, the outcome of proposed research will be iteratively fine-tuned for quality optimization.

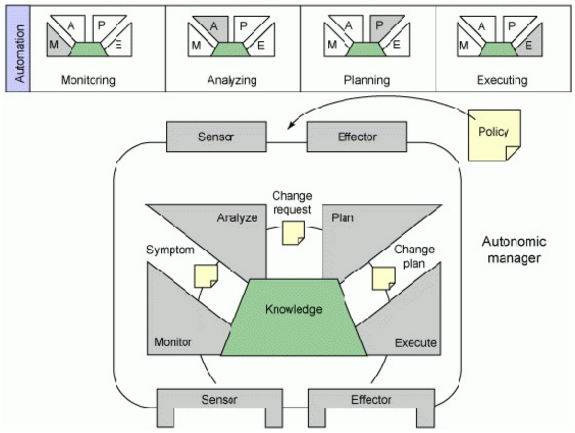

Fig. 1. MAPE-K Framework

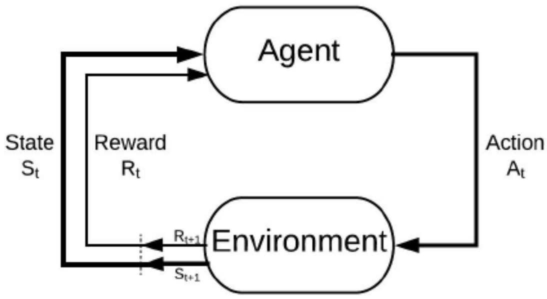

Fig. 2. Q-Learning Model

The overall methodology is the adaptation framework developed by IBM and known as MAPE-K. Q-learning is the on-policy approach that is available under Reinforcement Learning and to make it integrated with the predefined policy, fuzzy is introduced. The combination of both is significantly great in the iteration process under the development of continuous knowledge management rules enhancements.

4. Result and Analysis

In Q-Learning, it is derived from Markov Decision Process (MDP) and consists of three inputs. State, action, and reward. Every state will have a different score depending on the steps taken toward the final policy and continuing with the learning update.

The algorithm will have an iteration process. This is known as an episode to ensure that it will optimize with the alpha and gamma values:

i. γ = *Gamma* value in the range of 0 and 1. The lowest value will instruct Q-Learning to find the instance reward and ignore the total score of the accumulated reward.
ii. α = Alpha value identical to gamma, which is a range between 0 and 1. In normal circumstances, the value is set low to optimize the algorithm, such as 0.2.

Definition 1: Q-Learning Update

$$Q(s_t, a_t) \longleftarrow Q(s_t, a_t) + a[r_{t+1} + \gamma_a^{max} Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (1)$$

Q-Learning Algorithm

Initialize Q_0 (s, a) to random values
Choose a starting point s_0
While the policy is not good enough
Choose at according to values $Qt\ (st,.)$
$at = f(Qt(st,.))$
Obtain in return: $st + 1\ (s')$ and rt
Update using Definition 13
End While

In this assessment, two types of segments are reviewed and analysed. The duration of the analysis ranges from November 2020 till January 2023. The focused elements are:

i. Daily Error Types and Error Types (focused on the Request time out)
ii. Activities within server (Hits, Bandwidth and Visitors)

As for the error types, the categories as below.

i. 400 Bad Request
ii. 403 Forbidden
iii. 404 Not Found
iv. 405 Method Not Allowed
v. 408 Request Timeout
vi. 500 Internal Server Error

The interesting information is the 408 Request Timeout. That is the relation to Quality of Service (QoS) and elements in the Service Level Agreement (SLA). Request time outpointed on months July for the year 2020, April 2020, and November 2022. It is illustrated in Figure 3 and Figure 4. Populated patterns due to the nature of the academic calendar, whereby at that duration, lecturers uploaded the information about student absence and possible suggestions to be implemented. The amount of time to access the system, files such as PDF upload, and functions within that system is very consuming.

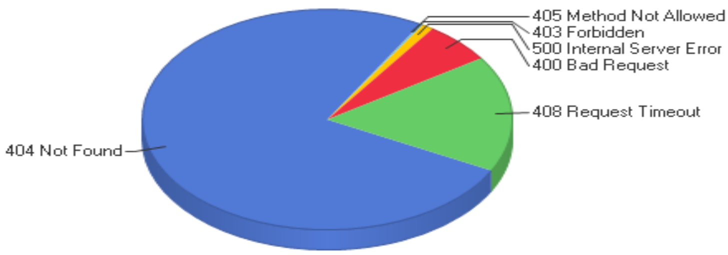

Fig. 3. Error Types November 2020 till Mei 2022 (Pie Chart)

It can either be on the bandwidth of the user or the server exhausted as well to handle multiple requests from academic staff. The reports on the escalation based, whereby it will go directly to the Head of Programme and later forwarded to the academic section for further actions and investigation on populated report.

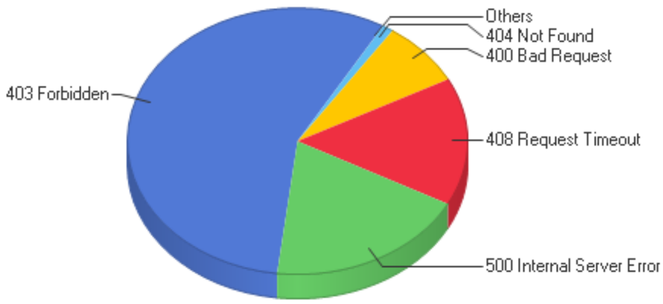

Fig. 4. Error Types Mei 2022 till January 2023 (Pie Chart)

Information in Figure 5 to Figure 14 is very much on the activities that consumed tasks outputted by server resources. Resources such as processor, memory, network connection, bus, admission control, etc. This is very much related to Figure 3 and Figure 4. Bandwidth, Hits, and Visitors produced by the above activities. The correlation is significant in the months July 2021, April 2022, July 2022, and January 2023. The increased-on Hits activities and it also implies Bandwidth and Visitors.

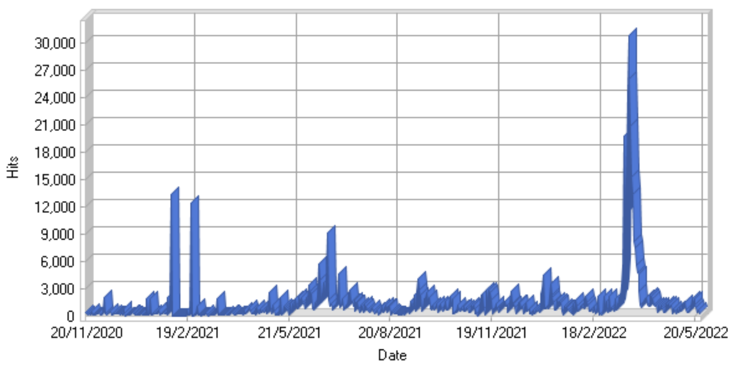

Fig. 5. Hits Activity November 2020 till May 2022

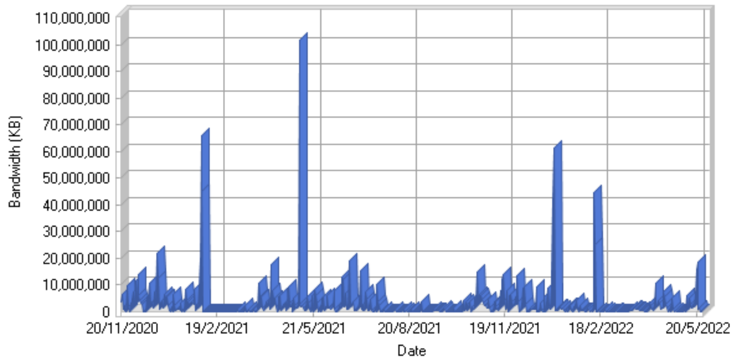

Fig. 6. Bandwidth Activity November 2020 till May 2022

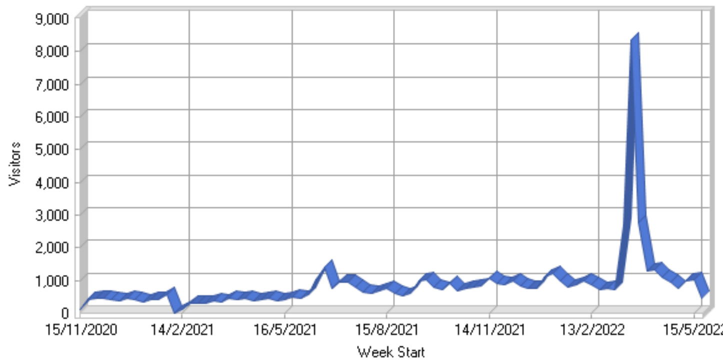

Fig. 7. Visitors Activity – Week - November 2020 till May 2022

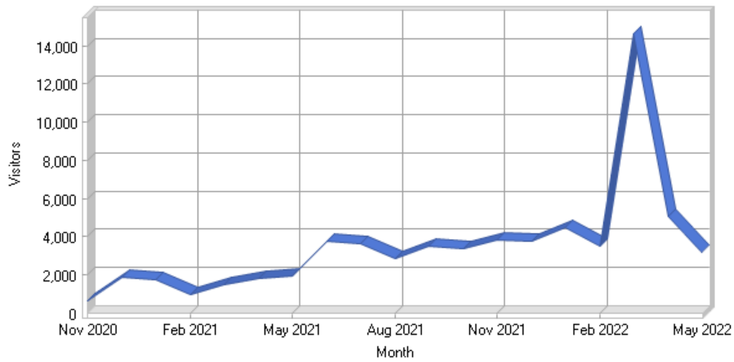

Fig. 8. Visitors Activity – Month - November 2020 till May 2022

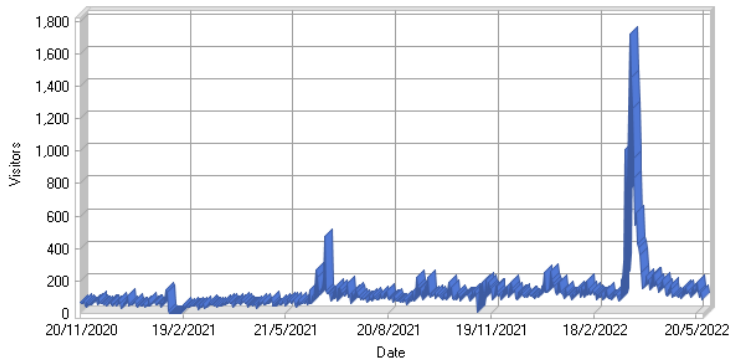

Fig. 9. Visitors Activity November 2020 till May 2022

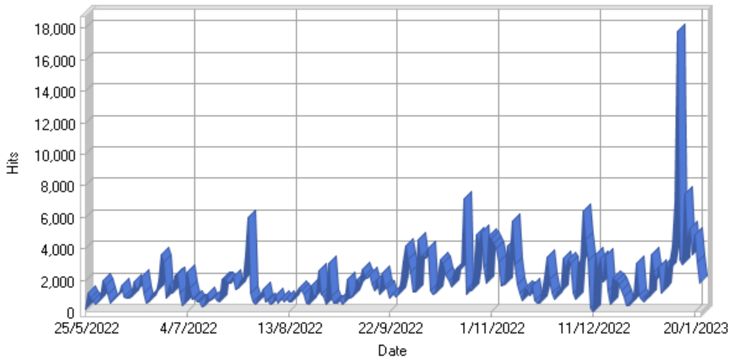

Fig. 10. Hits Activity May 2022 till January 2023

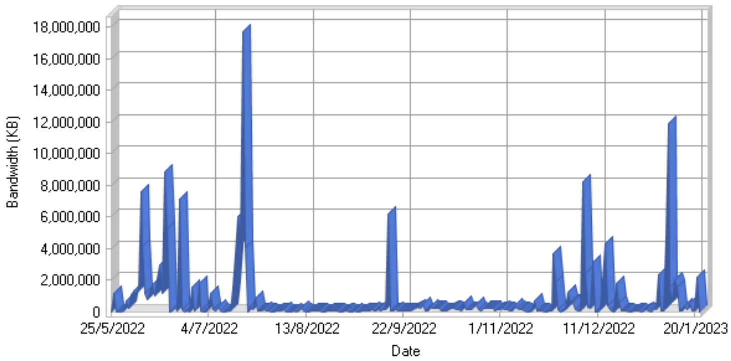

Fig. 11. Bandwidth Activity May 2022 till January 2023

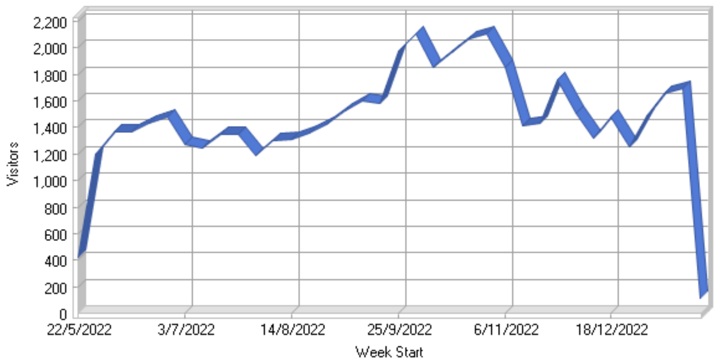

Fig. 12. Visitors Activity -Week - May 2022 till January 2023

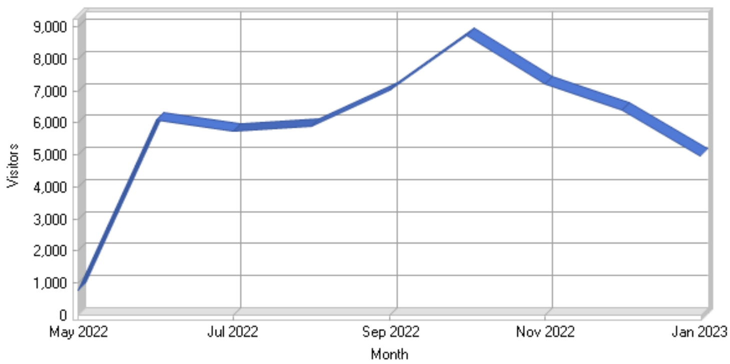

Fig. 13. Visitors Activity -Month - May 2022 till January 2023

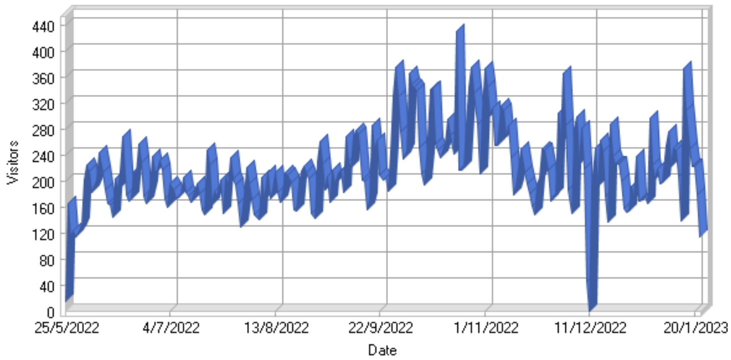

Fig. 14. Visitors Activity May 2022 till January 2023

From the findings, useful information demonstrated the connection of users, activities, and errors recorded on time out. The usage of MAPE-K in another research exhibited in previous works [17-20] demonstrated that few learning parameters can be configured and it can ensure the smooth process transacted by the approval from admission control.

5. Conclusion

This research demonstrated that the QoS logs recorded from SIRs accessed in a few categories and with targeted SLA perimeters. The result is according to the number of users accessing the system during peak and off-peak periods. With the combination of the adaptive approach and MAPE-K, the SLAs have been segmented in stages to understand the impact of gathered information. It is evidenced that from the gathered output, this result can suggest how to ensure QoS are stable, and users are able to cope with the intermittent performance of the SIRs server.

6. Future Works

Suggestions for the extension of these works can be made in two ways. The first one is to ensure there is a larger set of information and it is coordinated with another system in UiTM to ensure the impacted systems can be in good shape during demanding periods. The second suggestion is to apply the autonomic computing approach, whereby each function can negotiate and understand its loads and continue with auto-negotiation. It is very similar to admission control approaches. The registration of each function, understanding their limits, and being able to perform their current task correctly with the available resources before approving another incoming task.

Acknowledgement
This research was funded by a grant from UiTM Pahang (Internal).

References
[1] Taft, Rebecca, Irfan Sharif, Andrei Matei, Nathan VanBenschoten, Jordan Lewis, Tobias Grieger, Kai Niemi *et al.,* "Cockroachdb: The resilient geo-distributed sql database." In *Proceedings of the 2020 ACM SIGMOD international conference on management of data*, pp. 1493-1509. 2020. https://doi.org/10.1145/3318464.3386134
[2] Tirmazi, Muhammad, Ran Ben Basat, Jiaqi Gao, and Minlan Yu. "Cheetah: Accelerating database queries with switch pruning." In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 2407-2422. 2020. https://doi.org/10.1145/3318464.3389698
[3] Bichhawat, Abhishek, Matt Fredrikson, Jean Yang, and Akash Trehan. "Contextual and granular policy enforcement in database-backed applications." In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pp. 432-444. 2020. https://doi.org/10.1145/3320269.3384759

[4] Mebarkia, Khalil, and Zoltán Zsóka. "Analysis of Network's QoS in Service Chains." In *2020 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pp. 1-9. IEEE, 2020.

[5] Pei, Ouya, Zhanhuai Li, Wenjie Liu, and Hongtao Du. "A Novel Scalable Distributed Database System." In *Proceedings of the 2020 International Conference on Computers, Information Processing and Advanced Education*, pp. 31-36. 2020. https://doi.org/10.1145/3419635.3419649

[6] Böttcher, Jan, Viktor Leis, Jana Giceva, Thomas Neumann, and Alfons Kemper. "Scalable and robust latches for database systems." In *Proceedings of the 16th International Workshop on Data Management on New Hardware*, pp. 1-8. 2020. https://doi.org/10.1145/3399666.3399908

[7] Abadi, Daniel, Anastasia Ailamaki, David Andersen, Peter Bailis, Magdalena Balazinska, Philip Bernstein, Peter Boncz *et al.,* "The Seattle report on database research." *ACM Sigmod Record* 48, no. 4 (2020): 44-53. https://doi.org/10.1145/3385658.3385668

[8] Rattaro, Claudina, Laura Aspirot, Ernesto Mordecki, and Pablo Belzarena. "QoS Provision in a Dynamic Channel Allocation Based<? brk?> on Admission Control Decisions." *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)* 5, no. 1 (2020): 1-29. https://doi.org/10.1145/3372786

[9] Uriarte, Rafael Brundo, Rocco De Nicola, and Kyriakos Kritikos. "Towards distributed sla management with smart contracts and blockchain." In *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 266-271. IEEE, 2018. https://doi.org/10.1109/CloudCom2018.2018.00059

[10] Jia, Yichen, and Feng Chen. "Kill two birds with one stone: Auto-tuning rocksdb for high bandwidth and low latency." In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 652-664. IEEE, 2020. https://doi.org/10.1109/ICDCS47774.2020.00113

[11] Ye, Fanghua, Zhiwei Lin, Chuan Chen, Zibin Zheng, and Hong Huang. "Outlier-resilient web service qos prediction." In *Proceedings of the Web Conference 2021*, pp. 3099-3110. 2021. https://doi.org/10.1145/3442381.3449938

[12] Flores, Nikka, Maryl Enteria, Marco Pefianco, and Michael Nayat Young. "Electronic medical record database with accessible online summary and statistics." In *Proceedings of the 6th International Conference on Industrial and Business Engineering*, pp. 86-90. 2020. https://doi.org/10.1145/3429551.3429592

[13] Feng, Mingbin, and Jeremy Staum. "Green simulation with database Monte Carlo." *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 31, no. 1 (2021): 1-26. https://doi.org/10.1145/3429336

[14] Rigger, Manuel, and Zhendong Su. "Finding bugs in database systems via query partitioning." *Proceedings of the ACM on Programming Languages* 4, no. OOPSLA (2020): 1-30. https://doi.org/10.1145/3428279

[15] Nurmela, Tuomas. "Evaluation framework for service level management in federated service management context." In *Preproceedings of ISTSPQ 2007 The 2nd international workshop on Interoperability solutions to Trust, Security, Policies and QoS for Enhanced Enterprise Systems*, p. 27. 2007.

[16] Akbari-Moghanjoughi, Ayyoub, José Roberto De Almeida Amazonas, Germán Santos-Boada, and Josep Solé-Pareta. "Service level agreements for communication networks: A survey." *arXiv preprint arXiv:2309.07272* (2023).

[17] Bin Ramli, Ahmad Kamal. "Service Level Agreement-based adaptation management for Internet Service Provider (ISP) using Fuzzy Q-learning." PhD diss., University of Leeds, 2018.

[18] Qureshi, Haneya Naeem, Marvin Manalastas, Syed Muhammad Asad Zaidi, Ali Imran, and Mohamad Omar Al Kalaa. "Service level agreements for 5G and beyond: overview, challenges and enablers of 5G-healthcare systems." *IEEE Access* 9 (2020): 1044-1061. https://doi.org/10.1109/ACCESS.2020.3046927

[19] Tan, Wenan, Hai Zhu, Jinjing Tan, Yao Zhao, Li Da Xu, and Kai Guo. "A novel service level agreement model using blockchain and smart contract for cloud manufacturing in industry 4.0." *Enterprise Information Systems* 16, no. 12 (2022): 1939426. https://doi.org/10.1080/17517575.2021.1939426

[20] Uriarte, Rafael Brundo, Huan Zhou, Kyriakos Kritikos, Zeshun Shi, Zhiming Zhao, and Rocco De Nicola. "Distributed service-level agreement management with smart contracts and blockchain." *Concurrency and Computation: Practice and Experience* 33, no. 14 (2021): e5800. https://doi.org/10.1002/cpe.5800