Management of Optimal Resource Allocation in the Cloud

Manoj Kumar
Department of Computer Science
Excel College of Engineering, Namakkal, India

ABSTRACT
The management of resource allocation in the cloud is a critical issue that has received significant attention in recent years due to the increasing demand for cloud-based services. The efficient allocation of resources is crucial to meet the requirements of different applications and to optimize the utilization of available resources. This research paper explores the concept of optimal management of resource allocation in the cloud. The paper analyzes different approaches to resource allocation and discusses the advantages and limitations of each approach. The research also examines various factors that affect resource allocation in the cloud, including workload, resource availability, and resource utilization. The paper proposes a novel approach to resource allocation that is based on machine learning algorithms. The approach uses historical data to predict resource utilization and allocate resources accordingly. The research also investigates the impact of different factors on the performance of the proposed approach and compares it with other existing approaches. The findings of this research paper provide insights into the optimal management of resource allocation in the cloud. The proposed approach is shown to be effective in improving resource utilization and meeting the requirements of different applications. The research also highlights the importance of considering different factors that affect resource allocation in the cloud to achieve optimal performance.

Keywords
Resource allocation, Cloud, software, SRGM

1. INTRODUCTION
Resource allocation in the cloud refers to the process of assigning computing resources to different workloads in a cloud computing environment. These resources can include virtual machines, storage, networking, and other infrastructure components. Effective resource allocation is critical to ensuring optimal performance, scalability, and cost-efficiency in cloud computing. By allocating resources appropriately, you can ensure that workloads have the necessary resources to operate efficiently without incurring unnecessary costs. In cloud computing, resource allocation is typically managed through a combination of automation, policies, and monitoring. Automation tools can dynamically allocate resources based on workload demands, while policies can define rules for allocating resources based on factors such as workload type, priority, and business needs. [5] Monitoring tools can provide insights into resource utilization, enabling administrators to adjust resource allocation as needed to optimize performance and cost efficiency. Optimizing resource allocation in the cloud requires careful planning, including forecasting demand, understanding workload requirements, and selecting appropriate cloud infrastructure components. Effective resource allocation also requires ongoing monitoring and analysis to ensure that resources are being used efficiently and effectively.

[32,33,34] One of the advantages of Cloud computing lies in the possibility offered to users to carry out parallel calculations. These consist of dividing an application into elementary tasks distributed over several resources that can operate simultaneously. The objective of this distribution is to improve the performance of applications as well as executed compared to sequential execution. [6] Furthermore, the tasks constituting the distributed applications can be linked by constraints such as time and/or data constraints. As a result, the optimal management of available resources and the scheduling of tasks are fundamental aspects in the parallelization of applications. Resource allocation and task scheduling of an application consist in determining the resources to be assign to each of these tasks and the order in which they should be performed.

2. REVIEW OF LITERATURE
Resource allocation in software refers to the process of assigning and managing resources such as memory, processing power, storage, and network bandwidth to software applications. [35,36,37] Effective resource allocation is critical for software performance, scalability, and reliability. Here are some key aspects of resource allocation in software:

Memory allocation: Managing memory allocation is critical for software performance and stability. Inefficient use of memory can lead to crashes, hangs, and poor performance. Effective memory allocation involves allocating memory dynamically, deallocating memory when it is no longer needed, and managing memory fragmentation.

CPU allocation: Assigning processing power to software applications is important for achieving optimal performance. This involves balancing CPU resources among different applications and processes running on a system, as well as managing CPU affinity and scheduling[38,39,40].

Storage allocation: Efficient storage allocation is important for managing data access and storage in software applications. This includes allocating disk space dynamically, managing file systems and directories, and optimizing data access patterns.

Network bandwidth allocation: Allocating network bandwidth is critical for ensuring that software applications can communicate effectively over a network. This involves managing network traffic, optimizing network protocols, and allocating network resources among different applications.

Resource monitoring and management: Effective resource allocation requires monitoring resource usage and performance metrics, and making adjustments to resource allocation as needed. This involves implementing tools and techniques for monitoring resource usage and performance, and developing algorithms and policies for managing resource allocation dynamically.

[1,2,3] conducted a systematic review of resource allocation for cloud-based software systems. The paper provides a comprehensive survey of resource allocation techniques and frameworks for cloud-based software systems. [4]surveyed

resource allocation techniques for real-time software systems. The paper reviews different resource allocation techniques and their applicability to real-time software systems. [7] surveyed resource allocation for software-defined networks (SDNs). The paper provides an overview of resource allocation techniques and frameworks for SDNs. [8] surveyed resource allocation techniques for multimedia software systems. The paper reviews different resource allocation techniques and their applicability to multimedia software systems. [9] surveyed resource allocation for mobile software systems. The paper provides an overview of resource allocation techniques and frameworks for mobile software systems. [10,11,12] conducted a survey on resource allocation in virtualized environments. The paper provides an overview of virtualized environments and reviews different resource allocation techniques and their applicability in such environments. [13-14] surveyed resource allocation techniques in cloud computing. The paper reviews different resource allocation techniques and frameworks in cloud computing and identifies their limitations and challenges. [15-17] surveyed resource allocation for big data processing in cloud computing. The paper reviews different resource allocation techniques and frameworks for big data processing in cloud computing. [18-22] surveyed resource allocation techniques for software-defined cloud computing environments. The paper provides an overview of software-defined cloud computing environments and reviews different resource allocation techniques and frameworks. [23] surveyed resource allocation for container-based virtualization. The paper reviews different resource allocation techniques and frameworks for container-based virtualization and identifies their limitations and challenges.

3. PROPOSED APPROACHES TO ALLOCATE THE RESOURCES

The main idea of the approaches described below consists first of all in release the initial problem and then recalculate the two objective functions taken into account, using in particular the heuristic described above. [41]More precisely, the approaches consist in a first phase in finding an assignment tasks without considering all precedence constraints using three complementary strategies to traverse the task graph in question. [24,25] The solutions thus obtained are lower bounds of the exact values (overall time and overall cost execution). The subsequent calculation of these objective functions taking into account the precedence constraints allow to obtain upper bounds. Proposed approaches for resource allocation and scheduling tasks of an instance of a business process are based on the following key observation.

Given a task graph and a set of heterogeneous resources. Use a strategy of traversing from top to bottom of the graph of tasks to be executed (which is often used in task graph scheduling algorithms) [26] often does not yield better results compared to the strategy which consists, for example, of traversing the graph in question from bottom to top.

The global execution cost function denoted cost can be defined as being the sum of these last two quantities considering all the tasks of the instance at execute. Formally, it is given by the following equation:

$$cost = \sum_{j=1}^{n} \left\{ EC(r(t_j)) + \sum_{p \in pred(t_j)} TC(r(t_j), r(t_p)) \right\}$$

this approach is driven by the objective function of the overall execution time. Since human resources may be required to perform tasks not belonging to the instance of the process model that interests us, the call to the heuristic allowing to predict their availability is carried out before any assignment [272,28,29]. As for the previous approach for each of the solutions obtained the values of the two objective functions taken into account are calculated. The three approaches of the resource allocation phase of this approach are described below. The three traversal strategies of the task graph in question follow the same pattern as the previous approach. [30,310 Algorithm gives an overview of the function-directed approach overall execution time.

Algorithm Time-based approach

1: read the DAG, the RG and associated attributes values ;

2: sort tasks at each level by traversing the DAG in a top-down fashion ;

3: k ← 1 ;

4: while (k ≤ L) do

5: for all tasks ti ∈ lk, compute r(ti)

// assign task tk to the virtual machine r(ti)

//that minimizes the execution time

mintime[k, ti

] ← r(ti); // mincost is a L × m matrix

6: h ← k + 1 ;

7: while (h ≤ L) do

8: for all tasks ti ∈ lh, compute r(ti) using equation

mintime[k, ti] ← r(ti);

9: h ← h + 1

10: end while

11: h ← k − 1 ; // compute r(ti) for all tasks that belong // to levels h < k

12: while (h ≥ 1) do

13: for all tasks ti ∈ lh, compute r(ti) using equation

mintime[k, ti

] ← r(ti);

14: h ← h − 1

15: end while

16: k ← k + 1

17: endwhile

18: for each assignment, compute cost using equation;

19: select the Pareto solutions among L solutions ;

In order to assess the quality of the solutions obtained by the approaches proposed in this chapter, we performed a series of simulations. The simulation scheme used consists of randomly generating cycle-free graphs of tasks representing the instance to be executed. For this, we consider five families of process models determined by the number of tasks that compose them, namely n ∈ {50, 100, 300, 600, 1000}. For each instance (the number of tasks is fixed), we define three families of process models called Small, Medium and Large which are denoted respectively by S, M and L. The latter are defined by associating a probability for the existence a precedence constraint between each pair of tasks ti and tj . These probabilities are ps = 0.2, pm = 0.4 and pl = 0.6 for families S, M and L respectively. The number of virtual machines is set to m = n/2. The other parameters of the model that we propose for resource allocation and task scheduling of a process model are generated randomly.

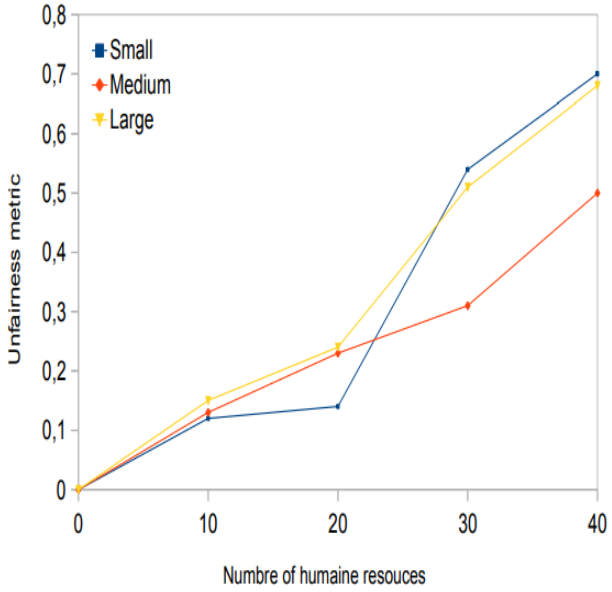

Figure 1– The unfairness criterion for each family of instances with n = 50 gives better results than the other two. Therefore, we recommend users the use of the three strategies simultaneously and to retain those that provide the best results.

4. CONCLUSIONS

In this paper, we are interested in the problem of resource allocation and task scheduling of a business process. we have distinguished in these types of resources, namely virtual machines and human resources. This is justified by the fact that it is difficult, if not impossible, to automate all the tasks of a process given job. The fact that the number of human resources is limited requires management optimal of their queues. In addition, these resources may be required to perform other tasks (which do not necessarily belong to the process that interests us). In order to take into account these last two points, we respectively proposed an objective function of the overall execution time of an instance of a process (taking into account the fact that the number of human resources is limited) and the use of forecasting models to estimate the availability of the resources used. Additionally, a heuristic for estimating the availability of used resources is proposed. So, we have shown the importance of using a heuristic to predict the availability of especially human resources.

5. REFERENCES

[1] S. Shirazipourazad and S. Mashayekhi, "Resource allocation for cloud-based software systems: A systematic review," Journal of Systems and Software, vol. 162, pp. 79-104, 2020. DOI: 10.1016/j.jss.2019.110498

[2] Md. Abu Kausar, Md. Nasar, and Sanjeev Kumar Singh. (2013). "A Detailed Study on Information Retrieval using Genetic Algorithm," Journal of Industrial and Intelligent Information, Vol.1, No.3, pp. 122-127, doi: 10.12720/jiii.1.3.122-127.

[3] K. Saleh and M. H. Alhazmi, "Resource allocation techniques for real-time software systems: A survey," ACM Computing Surveys, vol. 53, no. 5, 2020. DOI: 10.1145/3418318

[4] J. Gong, J. Peng, and Y. Wang, "A survey on resource allocation for software-defined networks," Computer Networks, vol. 165, 2019. DOI: 10.1016/j.comnet.2019.106950

[5] Md. Nasar, Prashant Johri and Udayan Chanda, "A Differential Evolution Approach for Software Testing Effort Allocation," Journal of Industrial and Intelligent Information, Vol. 1, No. 2, pp. 111-115, June 2013. doi: 10.12720/jiii.1.2.111-115.

[6] M. Nasar and P. Johri. (2016). ―Testing resource allocation for fault detection process‖. In Smart Trends in Information Technology and Computer Communications. A. Unal et al. (Eds.). 683--690. DOI:10.1007/978-981-10-3433-6_82.

[7] Johri, P., Nasar, M., Chanda, U. (2013). ―A genetic algorithm approach for optimal allocation of software testing effort‖. International Journal of Computer Applications. 68, 21–25.

[8] M. H. Alhazmi and K. Saleh, "Resource allocation techniques for multimedia software systems: A survey," Journal of Multimedia Tools and Applications, vol. 77, no. 1, pp. 47-80, 2018. DOI: 10.1007/s11042-016-4114-8

[9] P. Johri, M. Nasar, S. Das, and M. Kumar. (2016). ―Open source software reliability growth models for distributed environment based on component-specific testing-effort. In Proceedings of the 2nd International Conference on Information and Communication Technology for Competitive Strategies. 75. DOI:10.1145/2905055.2905283

[10] G. Gao and J. Shao, "A survey on resource allocation for mobile software systems," Journal of Network and Computer Applications, vol. 75, pp. 189-202, 2016. DOI: 10.1016/j.jnca.2016.09.012

[11] Nasar, M., Johri, P. (2014). ―Testing and Debugging Resource Allocation for Fault Detection and Removal Process‖. International Journal of New Computer Architectures and their Applications, no. 4, pp. 193—200.

[12] F. Chen, X. Liu, and Y. Liu, "Survey on resource allocation in virtualized environments," Journal of Network and Computer Applications, vol. 94, pp. 72-84, 2017. DOI: 10.1016/j.jnca.2017.07.020

[13] S. S. Islam, M. N. B. Chowdhury, and M. A. Alam, "A survey on resource allocation techniques in cloud computing," Journal of Cloud Computing, vol. 8, no. 1, 2019. DOI: 10.1186/s13677-019-0143-3

[14] M. M. S. Rana, M. H. Alhazmi, and K. Saleh, "Resource allocation for big data processing in cloud computing: A survey," Journal of Big Data, vol. 6, no. 1, 2019. DOI: 10.1186/s40537-019-0209-9

[15] Md. Nasar, Prashant Johri, Udayan Chanda,"Dynamic Effort Allocation Problem Using Genetic Algorithm Approach", IJMECS, vol.6, no.6, pp.46-52, 2014.DOI: 10.5815/ijmecs.2014.06.06

[16] R. V. Bonam, K. Raju, and M. V. R. K. Murthy, "A survey on resource allocation techniques for software-defined cloud computing environments," Journal of Network and Computer Applications, vol. 109, pp. 98-119, 2018. DOI: 10.1016/j.jnca.2018.03.012

[17] Kausar, M. A., Fageeri, S. O., & Soosaimanickam, A. (2023). Sentiment Classification based on Machine Learning Approaches in Amazon Product Reviews. Engineering, Technology & Applied Science Research, 13(3), 10849-10855.

[18] Saleem Basha and Mohamed Nasar. Resource Allocation in Cloud: History Kerberos based Approach. International Journal of Computer Applications 184(12):36-43, May 2022

[19] Y. Zheng, X. Zhang, and D. Yuan, "A survey on resource allocation for container-based virtualization," IEEE Access, vol. 8, pp. 121039-121054, 2020. DOI: 10.1109/ACCESS.2020.3002513

[20] Nasar, M., & Johri, P. (2015). Testing Resource Allocation for Modular Software using Genetic Algorithm. IJNCAA, Vol. 5, No. 1, pp. 29-38.

[21] Oprescu, T. Kielmann, (2010). "Bag-of-Tasks Scheduling under Budget Constraints", IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), pages 351-359.

[22] Md. Nasar, Prashant Johri, Udayan Chanda,"Resource Allocation Policies for Fault Detection and Removal Process", IJMECS, vol.6, no.11, pp.52-57, 2014.DOI: 10.5815/ijmecs.2014.11.07

[23] F. Zhang, J. Cao, K. Hwang, and C. Wu. (2011). "Ordinal Optimized Scheduling of Scientific Workflows in Elastic Compute Clouds", In Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science.

[24] Md. Nasar, Prashant Johri, Udayan Chanda,"Software Testing Resource Allocation and Release Time Problem: A Review", IJMECS, vol.6, no.2, pp.48-55, 2014.DOI: 10.5815/ijmecs.2014.02.07

[25] Ejarque J. (2010). "A Multi-agent Approach for Semantic Resource Allocation". 2010 IEEE Second International Conference on Cloud Computing Technology and Science, pp. 335- 342. Mohammad Nasar. Web 3.0: A Review and its Future. International Journal of Computer Applications 185(10):41-46, May 2023.

[26] Nasar, M.; Kausar, M.A. Suitability of Influxdb Database for Iot Applications. Int. J. Innov. Technol. Explor. Eng. 2019, 8, 1850–1857.

[27] M. A. Kausar, A. Soosaimanickam, and M. Nasar, "Public Sentiment Analysis on Twitter Data during COVID-19 Outbreak," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 2, pp. 415–422, 2021

[28] Saidi, K. S. S. A., Kausar, M. A., & Elshaiekh, N. E. M. (2021). The Impact of COVID-19 on Economic of Oman and Omani Customer's Behaviour. International Journal of Scientific Research and Management (IJSRM), 9(07), 2266-2279.

[29] M. A. Kausar, M. Nasar and A. Moyaid, "SQL Injection Detection and Prevention Techniques in ASP .NET Web Application," International Journal of Recent Technology and Engineering (IJRTE, pp. 7759-7766, September 2019

[30] R. Van Bossche, K. Vanmechelen, and J. Broeckhove. (2011). Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds, IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pages 320-327.

[31] Kausar MA, Nasar M (2021) SQL versus NoSQL databases to assess their appropriateness for big data application. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 14(4), pp.1098–1108.

[32] Kausar MA, Nasar M, Singh SK. Maintaining the repository of search engine freshness using mobile crawler. In: *2013 Annu. International Conference of the Emerg. Res. Areas 2013 Int. Conf. Microelectron. Commun. Renew. Energy*, IEEE, 2013, 1–6.

[33] Abu Kausar M, Nasar M, Soosaimanickam A (2022) A Study of Performance and Comparison of NoSQL Databases: MongoDB, Cassandra, and Redis Using YCSB. Indian Journal of Science and Technology 15(31): 1532-1540

[34] Kausar MA, Nasar M. An effective technique for detection and prevention of SQLIA by utilizing CHECKSUM based string matching. International Journal of Scientific & Engineering Research. 2018;9(1):1177–1182

[35] Kausar, M. A., Dhaka, V., and Singh, S. K.. 2013. Web crawler: a review. International Journal of Computer Applications 63:31–36

[36] D. Niyato, A.V. Vasilakos, and K. Zhu. (2011). Resource and Revenue Sharing with Coalition Formation of Cloud Providers: Game Theoretic Approach, 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pages 215-224

[37] Md. A. Kausar, V. S. Dhaka, and S. K. Singh, "An Effective Parallel Web Crawler based on Mobile Agent and Incremental Crawling," Journal of Industrial and Intelligent Information, vol. 1, no. 2, pp. 86–90, Jun. 2013.

[38] Md. A. Kausar, V. S. Dhaka, and S. K. Singh, "Web Crawler Based on Mobile Agent and Java Aglets," International Journal of Information Technology and Computer Science, vol. 5, no. 10, pp. 85–91, Sep. 2013

[39] Khan, M.S.; Kausar, M.A.; Nawaz, S.S. Big Data Analytics Techniques to Obtain Valuable Knowledge. *Indian J. Sci. Technol.* **2018**, *11*, 14

[40] Md. Abu Kausar, Md. Nasar & Sanjeev Kumar Singh, "Information Retrieval using Soft Computing: An Overview", IJSER, Vol. 4, Issue. 4, April 2013

[41] Kausar, M.A., Dhaka, V.S., Singh, S.K.: Implementation of parallel web crawler through .NET technology. Int. J. Mod. Educ. Comput. Sci. (IJMECS) **6**(8), 59–65 (2014)

[42] Kausar A, Dhaka VS, Singh SK. Design of Web Crawler for the Client – Server Technology. Indian Journal of Science and Technology. 2015 Dec; 8(36):1–7.

[43] Abu Kausar M, Dhaka VS, Singh SK. A novel web page change detection approach using Sql Server. International Journal of Modern Education and Computer Science (IJMECS), Hong Kong. 2015; 7(9):36–43.