https://doi.org/10.33472/AFJBS.6.10.2024.903-908

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Cloud Cost Optimization Techniques of Software Development using CloudDevOps

[1] Boga. Mahi Kiran, [2] Kanneboyina. Rakesh Yadav, [3] BSSN. Kowsik, [4] Durgempudi. Srikar

[5] Bulla. Suneetha [6] Mothukuri. Radha

[1,2,3,4,5,6] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P., India-522502.

[1] 2100030970cseh@gmail.com,[2] 2100030969cseh@gmail.com,[3] 2100032254cseh@gmail.com, [4] 2100032499cse@gmail.com,[5] suneethabulla@kluniversity.in [6] radha@kluniversity.in

Volume 6, Issue 10, 2024

Received: 09 March 2024

Accepted: 10 April 2024

Published: 20 May 2024

doi:10.33472/AFJBS.6.10.2024.903-908

Abstract : Cloud computing and DevOps approaches have drastically changed how software applications are developed, deployed, and managed in businesses. The teams in development and operations collaborate more successfully and with quicker development cycles.As cloud applications get bigger, cost minimization becomes more important. In these streamlining operations to reduce the cost and increase the performance, based on our survey conducted, Kubernetes is an effective container orchestrating technology which provides a standardized and scalable architecture. Its capacity to automate the deployment, scaling, as well as management of containerized applications in accordance with the DevOps values, which enhances effectiveness and consistency.

Keywords--- Cloud Cost Optimization, Kubernates, DevOps, Software Development.

I. INTRODUCTION

The software system may be the important thing to providing software development, as it defines the manner of reading customer desires and turning them right into a laptop device that fulfills them SDLC (Software Development Life Cycle) is the system of growing a utility or product. a. Installation and managing centers are available. It is mainly used to construct softwareinside the IT industry. A design, layout, specification, or summary representation of a software program machine is called a software program machine version.**[14]**

Software development processes can be optimized and improved by the combined use of Kubernetes and CI/CD.

Just like CI/CD improves the code merging and release parts of the SDLC, Kubernetes simplifies and automates the management of containerized apps. In Kubernetes, containers are like adjusted pieces that package a program and its dependencies, ensuring stability across different settings. CI/CD fits easily into this by handling the testing, building, and deployment of these containers.

In the CI phase, developers develop the code changes that are automatically in a open repository.Kubernetes, in this case, acts as the orchestrator, which ensures that these modifications integratedinto containers smoothly and their configurations. This helps quick identification and settlement of integrating problems. Moving to the CD step, Kubernetes takes the lead in creatively applyingthe code changes by handling the launch of containers. It provides a

quick and stable release of apps. CI/CD techniques, when paired with Kubernetes, add faster development processes, better discussions between the teams working on development and operations, and the release of high- quality software.**[1]**

The combination of CI/CD with Kubernetes helps organizations to limit mistakes, reduce human involvement in distribution processes, create a continuous development culture, and eventually produce software products that match with changing user needs. The mix brings efficiency, stability, and speed to the full software creation and release process.**[15]** DevOps is like a teamwork approach that combines development and operations to make software quickly. Automation, which is using machines to do tasks automatically, is a big part of this process. Automated testing, where machines check if the software works correctly, is super important for making things fast. But there's also manual testing, where humans explore different situations, check how users experience the software, and find unexpected problems. Manual testingis good because it adds a human touch, but it can take a lot of time, humans might make mistakes,and it's not great for keeping up with frequent updates. to make sure the software is good quality and can be delivered quickly in the DevOps way.

Software development can be optimized through integrating automated checking out with cloud computing, which makes use of the internet to offer computer sources. Developers may now take a look at their product on line without having to fear about difficult setup techniques. Fast checking out, value financial savings by using best procuring what is wanted at some stage in testing, and advanced teamwork because of easy environment sharing are the various benefits. It increases the performance and affordability of the improvement manner with the aid of optimizing it.

II. LITERATURE SURVEY

Companies which they start new or existing generally they use the on-premises data center. It hassome advantages like security, data control and if the network problem exists, they can access thedata and services. But the capital costs like the power, transportation, maintainence andinstallation services etc. They must bear all these expenses. If the data center is under maintenance, it is not possible to run the application.

As there is a vast growth in Cloud Day by day, the need for computing, network, infrastructure and optimizing the cost is also increasing to improve speed, efficiency and faster deployment, automatic scaling. To meet these requirements Kubernetes comes into place, which is Container Orchestration system for Efficient Management of containerized Applications.**[1]** It involves providing various tools for monitoring and managing applications with some mechanisms like detection and recovery whenever a failure occurs.

[1] For cloud optimization, the authors have categorized several scheduling techniques using Kubernetes like multi-objective optimization-based scheduling and autoscaling enabled scheduling. Auto Enabled Scheduling is an important attribute in Kubernetes which will adjust theresources based on current demand. **[2]** For example, there is an adjustable auto scalar called Libra which will automatically find the optimal resource and promotes horizontal scaling, when it compared to default Kubernetes auto scaler, it results in reducing the utilizations from 48% to 38%.

[1] Multi-Objective Optimization-based scheduling helps in allocating the resources with the combination of multiple objectives like Performance, Scalability, Resource Utilization, Cost Efficiency to achieve balanced and optimized system. Usage of multi-objective optimization algorithms in Kubernetes like Ant Colony Optimization, and optimization of particle swarms improves the Kubernetes cluster's overall performance.

[3] The author proposed a new Cloud Cost Optimization strategy for cost efficient Kubernetes container orchestration using Heterogenous Task Allocation using Elastic Computing Resources contains three key characteristics, the first is integrating the containers with the resources that are already available using task packing. Second is using scaling algorithms, adjust the cluster size forvarying workloads and third one involves in establishing a rescheduling strategy to find and turn off unused virtual machine instances with the goal of cost savings, while seamlessly reallocating pertinent jobs without compromising task progress. When this strategy compared to default Kubernetes cost, it results in lowering costs overall by 23% to 32% on different workloads and patterns.

[4] For orchestration of job execution on virtual clusters by allocating a collection of virtual machine instances on public Infrastructure as a Service (IaaS) dynamically, the authors introduceda new cluster scheduler called Stratus. The main aim of stratus is to optimize the dollar costs and tries to allocate the resources in two methods: first is either fully allocated (Maximum Utilization)or empty to save money which is guided by job runtime estimates. These simulations result in reducing costs by 17% to 44% compared to virtual cluster scheduling given by Google and Two Sigma based on traces and cluster workloads.

[5] The author introduced a new feedback control mechanism which is specifically targeting Elastic Containerized Web Systems used for Kubernetes. This method integrates two models, linear model, and varying-processing-rate queuing model which reduces the output errors during provisioning containers. When we compare this mechanism with ultramodern algorithms and cutting-edge algorithms, it offends the least percentage of Service Level Agreements (SLA) and secures the second-best cost efficiency compared to other algorithms.

[6] The author says about the how the cloud computing use

in more simple and more efficient andmore specialized and concluded use more Web Based Applications instead of the Desktop based applications.

[7] The authors proposed a scheduler for perfecting the scheduling of the pods in the serverless framework on Kubernetes platform and they had argued the default Kubernetes scheduler run on pod-by-pod basis is not suited for the rapid deployment. They gave an algorithm that makes utilization of simultaneous scheduling of pods to enhance the scheduling and after the testing thatalgorithm reduced the delay in pod startup while supporting the balance of resources.

[9] The author discussed the concepts of container placement and migration on edge servers and focuses on a planning model specifically designed for this purpose. As the authors point out, mostof these planning models are primarily based on heuristic algorithms, including multi-aim optimization models and graph network models. A noteworthy aspect of this study is the existenceof a gap in research on container scheduling models that consider distributed edge computing activities. The authors predict that future research in this area will focus more on container planning for mobile edge nodes. This prediction highlights the evolving landscape of edge computing and the need for customized planning approaches to address the unique challenges of mobile edge environments.

[10] The authors present4 KaiS, an3 edge pall Kubernetes cataloging frame grounded on literacy.KaiS models system state data are using graph neural networks and coordinated multi-agent actor- critic system for decentralized request dispatch. Research in comparison to nascence, KaiS could increase average system out turn rate by 14.3% and to reduce scheduling cost by 34.7%.

[11] The author conducted different experiments to reduce the cost and CPU usage. He conductedresearch based on the effects of requests and limit configuration. If there is only one pod scheduledon a node, and if that pod has no limits set, then it should be allowed to consume all the node's resources. This experiment explores this notion using a Cassandra application. Finally, It shows the 95th percentile latencies of the re-quests, as reported by the experiment controller. At roughly 400requests per second, the SLO is breached. It indicates that at roughly the same number of requests per second, the node uses all its available CPU, since 2.0K milli cores = 2 CPUs. This validates that the bottleneck is truly the CPU.

[12] The author says Containers are quickly replacing Virtual Machines (VMs) as the compute instance of choice in cloud-based deployments. The substantially cheaper overhead of deploying containers (relative to VMs) has frequently been noted as one reason for this. We investigate performance of the Kubernetes system and build a Reference net-based model of resourcemanagement inside this system. Our model is characterised using actual data from a Kubernetes deployment and can be used as a foundation to create scalable apps that make use of Kubernetes.

A performance model for Kubernetes based deployment is given. With rising interest in applications (such as stream processing) which require to start and terminate instances on a per second basis, the overhead associated with VMs remains a restriction. We employ a benchmark- based approach to better describe behavior of a Kubernetes system (using Docker containers). Wepropose a Reference net-based approach for Pod & container lifecycle in Kubernetes

[13] The author discusses the importance of flexible and cost-effective network design for 5G technology, focusing on modular network functions for fast deployment and scalability. It uses virtualization tools like Kubernetes to manage containers, improve availability, and enhance resilience. The paper proposes a solution to avoid resource shortages in cluster nodes while protecting critical functions, using a statistical approach for problem modeling and resolution. He proposed resource scheduling in a cloud-native environment, focusing on Kubernetes technology. The proposed algorithm, KRS, optimizes memory and CPU provisioning to avoid resource shortages, particularly for critical Cloud Node Functions (CNFs). The algorithm considers consumption profiles and allocates resources with a trade-off between efficiency and risk. Comparing KRS with Kubernetes best effort mode shows significant gains. Optimizing network service resilience requires considering all stages of the service lifecycle.

III. COMPARATIVE STUDY & ANALYSIS:

TABLE I: COMPARATIVE ANALYSIS OF VARIOUS COST OPTIMIZATION TECHNIQUES.

Objectives	Algorithms	Experiments	Findings	Limitations

[1] Optimizethe cloud allocation of resources in Kubernetes. Balance, several goals, which involve performance, scalability, usage, and cost.	Ant Colony Optimization (ACO andParticle Swarm Optimization (PSO) are examples of multi-objective optimization. Autoscaling: Adjusts resourcesbased on demand.	Compared the speed, scalability, efficiency, and cost of different schedule methods. Evaluated theefficiency of Autoscaling in Resource Management.	Multi-objective optimization methods improve general efficiency and resource usage.	Multi-objective optimization can be computationally expensive. Autoscaling requires accurate demand estimationand efficient scaling methods.
[2] Reduce the consumption of resources example: 48% to 38%.	The auto-scaling feature is adjustable and focuses on horizontal scaling.	Compared to thedefault Kubernetes AutoScaler.	Reduced resource utilization while promoting horizontal expansion.	May need appropriate adjust-ments, depending on application demand.
[3] Reduce the cost of Kubernetes container orchestration on the cloud. Achieve high- cost efficiency through resource usage improvement.	HeterogeneousTask Allocation:Tightly packs containers into available resources. Rescheduling Mechanism: Deactivates idle virtual machinesand reallocates work.	Compared to thedefault Kubernetes cluster setup. Tested on varied workloads and patterns	Compared to default Kubernetes, it lowers total costs by 23% to 32%. Improves resource utilization and improves cost savings.	For efficient scalability, preciseworkload forecasting is necessary. Rescheduling may create costs and slow down work progress in variousinstances.
[4] Reduce the amount of money spent on virtual cluster job execution on open IaaS. Using dynamicvirtual machine provisioning.	Allocate virtual machines (VMs) just as required using on-demandprovisioning. Job runtime estimates determine whether to allocate resources or not.	Comparing the cost of the virtualcluster schedulingsolutions currently in use (Google, Two Sigma). tested withdifferent workloads.	Lower costs by 17% to 44% in compariso n to current methods, makes cost minimization a top priority while using resources effic iently.	For efficient allocation, it is dependent on precise work runtime estimations. It may not be appropriate for alljob kinds and workloads
[5] Reduce Kubernetes SLA breachesfor elastic containerizedweb systems. Cost- effectively optimize container provisioning.	A combination ofthe linear model and the variable-processing-rate queuing model (VQPM). Feedback Control modifies container provisioning in response to output faults.	Compared to the most recent container provisioning methods. analyzed the cost-efficiency and SLA breaches across arange of workloads.	Minimum proportion of SLA breaches in contrast to other methods. among the evaluated algorithms, the second-best cost efficiency.	For certain applications, implementing VQPM could be difficult. Training models and accurately predicting workloads may have an impact onperformance.
[7] Reduce pod starting delays in serverless apps on Kubernetes. Improve schedule effectivenesswhile maintaining resource balance.	Multiple pods are scheduled Simultaneously, rather than one at a time, using simultaneous podscheduling.	Compared the resource use andstartup latency with the Kubernetes scheduler by default.	Significantly cut pod starting delay compared to normal schedule and maintained resource balance within reasonable bounds.	Not all serverless applications and deployment patterns may benefit from this.

[9] Create a planning Model for moving and positioning containers on mobile edge servers.	Heuristic algorith ms in use today include graph netw ork models and multi-objective optimization models.	Examination of the literature on edge computing container scheduling	A Study has been done on scheduling models designed formobile edge nodes. Container planning for dynamic and resource constrained mobile edge settingswill be the focus of future studies.	For mobile edge sit uations, existing heuristic methods may not effective. New scheduling mo dels that take network dynamics, reso urce constraints, and mobility into acco unt are required.
[10] Increase thesystem throughput rate on average by 14.3%. Cut the cost ofscheduling by 34.7%.	Graph Neural networks (GNNs)represent relationships andstate informationabout a system.	KaiS was compared with baseline systems considering throughput andthe cost.	When compared to baselines, KaiS dramatically increases system throughput and lowers scheduling costs.	Drawbacks of the decentralized method, such as issues with scalability and coordination, are not addressed in the paper.
[12] Create aKubernetes performance model to comprehend resource management behavior.	Refer to the net- based method, which simulates the Kubernetes pod and container lifecycle. Analysis based on Docker containers to measure system performance.	Used actual data from a Kubernetes deployment toassess the model.	The model faithfully capturesthe behavior of Kubernetes resource management. may be used for application design, resource allocation, and capacity planning.	Needs further investigation to examine model characteristics and forecast possible behaviors. Pod and container lifecycles are the focus; Kubernetes performance may not be fully covered.

IV. DISCUSSIONS & CONCLUSIONS:

The research focuses on everything from on-premises, traditional data centers to cloud-based solutions emphasizes the importance of Kubernetes in the context of cost savings and improved efficiency in operations. In exploring DevOps approaches and the Software Development Life Cycle (SDLC), the literature analysis highlights the evolving nature of cloud technology and the growing significance of resource management. Moving on to the intersection of cloud and DevOps technologies, the study highlights Kubernetes as an essential catalyst for serverless frameworks.

To maximize efficiency, our analysis shows the significance of cost control within Kubernetes, highlighting important elements like auto-scaling, resource quotas, and ideal pod scheduling. In the future, this research will concentrate on how Kubernetes is changing and adopting new features and tools to provide increased control over resource allocation and optimize the cost on the cloud by deploying a real-time application.

V. REFERENCES:

[1] A survey of Kubernetes scheduling algorithms (2023) Khaldoun Senjab1, Sohail Abbas1*, Naveed Ahmed1 and Atta ur Rehman Khan2

[2] Balla D, Simon C, Maliosz M (2020) Adaptive scaling of Kubernetes pods. IEEE/IFIP Network Operations and Management Symposium 2020: Management in the Age of Softwarization and Artificial Intelligence, NOMS

[3] Zhong Z, Buyya R (2020) A Cost-Efficient Container Orchestration Strategy in Kubernetes- Based Cloud Computing Infrastructures with Heterogeneous Resources. ACM Trans Internet Techno 20(2):1–24

[4] Chung A, Park JW, Ganger GR (2018) Stratus: cost-aware container scheduling in the public cloud Andrew Chung Carnegie Mellon University Jun Woo Park Carnegie Mellon University Gregory R. Ganger Carnegie Mellon University

[5] Haja D, Szalay M, Sonkoly B, Pongracz G, Toka L (2019) Sharpening Kubernetes for the Edge. ACM SIGCOMM Conference Posters and Demos, Part of SIGCOMM. pp 136–137

[6] Research and Development on Cloud Computing March 2021, Aliasghar Azma, Nima Kianfar, Hossein Chitsazi

[7] Fan D, He D (2020) A Scheduler for Serverless Framework base on Kubernetes. ACM International Conference Proceeding Series. pp 229–232

[8]. Teaching and learning cloud computing, Mohd Zamri Murah* Procedia - Social and Behavioral Sciences 59 (2012) 157 – 163

[9] Oleghe O (2021) Container placement and migration in edge computing: concept and scheduling models. IEEE Access 9:68028–68043

[10] Han Y, Shen S, Wang X, Wang S, Leung VCM (2021) Tailored learningbased scheduling for kubernetes-oriented edge-cloud system. Proceedings - IEEE INFOCO

[11] Verreydt, Stef; Beni, Emad Heydari; Truyen, Eddy; Lagaisse, Bert; Joosen, Wouter (2019). *[ACM Press the 5th International Workshop - Davis, CA, USA (2019.12.09-2019.12.13)] Proceedings of the 5th International Workshop on Container Technologies and Container Clouds - WOC '19 - Leveraging Kubernetes for adaptive and cost-efficient resource management., (), 37– 42.* doi:10.1145/3366615.3368357

[12] Medel, Víctor; Rana, Omer; Bañares, José ángel; Arronategui, Unai (2016). *[ACM Press the 9th International Conference - Shanghai, China (2016.12.06-2016.12.09)] Proceedings of the 9th International Conference on Utility and Cloud Computing - UCC '16 - Modelling performance & resource management in kubernetes. (), 257–262.* doi:10.1145/2996890.3007869

[13] Mohamed Rahali, Cao-Thanh Phan, Gerardo Rubino. KRS: Kubernetes Resource Scheduler for resilient NFV networks. GLOBECOM 2021 - IEEE Global Communications Conference, Dec 2021, Madrid, Spain. pp.1-6. ffhal-03507257

[14] P. H. Feiler, W. S. Humphrey, "Software process development and enactment: Concepts anddefinitions" Proceedings of the Second International Conference on Software Process (February 1993) 28-40

[15] Cloud Cost Optimization: Model, Bounds, and Asymptotics46 Pages Posted: 25 Jul 2022 Lastrevised: 25 Sep 2023, Zihao Qu University of Texas at Dallas - Department of Information Systems & Operations Management, Milind Dawande University of Texas at Dallas - Department of Information Systems & Operations Management, Ganesh Janakiraman University of Texas at Dallas - Naveen Jindal School of Management