IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 May 2022, accepted 15 June 2022, date of publication 24 June 2022, date of current version 1 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185987

Scalable Management of Heterogeneous Cloud Resources Based on Evolution Strategies Algorithm

PETRA LONCAR[1], (Graduate Student Member, IEEE), AND PAULA LONCAR[2]
[1]Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, 21000 Split, Croatia
[2]Faculty of Science, University of Split, 21000 Split, Croatia

Corresponding author: Petra Loncar (ploncar@fesb.hr)

ABSTRACT The emergence and rapid development of Cloud computing have intensively changed the Information Technology paradigm. Cloud computing found its wide application in science and engineering. Cloud-based systems provide new solutions for scientific experiments and access to data distributed across data centers. Optimization of Cloud resource management is becoming a significant area of research. This research focuses on scheduling tasks and processing scientific data in a heterogeneous Cloud environment where most jobs require a large number of resources and computing power. Our approach proposes a strategy for optimizing task scheduling across different virtual machines and data centers based on the metaheuristic Evolution Strategies algorithm. The Evolution Strategies algorithm was tested, which has not been used in this domain. As an essential property of the system, scalability has played an important role in selecting the algorithm. We created a model and added a Longest Job First broker policy. Compared to the standard Genetic Algorithm, our approach has shown improvements in measured metrics. After testing under different loads, the proposed strategy gave promising results and achieved a better makespan, larger average resource utilization, better throughput, less average execution time, a smaller degree of imbalance, and scalability.

INDEX TERMS Cloud computing, distributed management, evolution strategies, heterogeneous cloud computing, resource management, scalability, task scheduling.

I. INTRODUCTION

Data with their sources, management, and quality are crucial in the digital environment. Data science and data analytics are becoming more sophisticated. Data are the fuel for algorithms and many digital processes. Algorithms enable a more optimal flow of processing large amounts of data. Big Data technologies involve using highly scalable distributed infrastructure for parallel processing, storage, transmission, and access to remote resources for the joint work of a large number of geographically remote researchers. The Cloud computing market has been growing in all world regions, as estimated by the International Data Corporation (IDC) [1]. Cloud is the general standard for running all Information Technology (IT) workloads. In addition to the leading global providers of Cloud computing services and infrastructure, Amazon, Microsoft, and Google, there are national and regional Cloud service providers. Data storage on dislocated servers leads to the need to establish a national e-infrastructure in the Cloud [2], [3] to achieve economic competitiveness, effective data control, and accelerate innovation in science and technology. The goal is to move the workload to the Cloud as much as possible to ensure reliability, performance, and efficiency requirements. That leads to many unknowns and future challenges in the Cloud and IT [4], as the ecosystem and complexity of services and data grow exponentially.

The concept of heterogeneity in the highly distributed data centers has been developing in parallel with the development of Cloud computing. The data center heterogeneity is manifested in the diversity of resource capacities and capabilities to deliver adequate Quality of Service (QoS). Due to the growing interest, the necessary High Performance Computing (HPC) infrastructure is provided by Cloud computing through data centers. The applications and benefits of this interdisciplinary data science technology are diverse. There are several issues to consider when adopting an approach to data analytics, like data reliability, collection, and

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Tsun Cheng.

storage. Example use cases are genomics, particle physics, and weather prediction.

The fundamental property that defines the quality of a Cloud computing system is scalability. The constant challenge is to achieve higher performance and greater scalability in managing data and Cloud infrastructure with optimized workload distribution. The real-time adaptive management of resources is mandatory in large-scale distributed architectures. The problem is that some systems do not scale to the volume of data requested by the data-intensive application. Another problem is defining the associated metrics when evaluating scalability and Cloud infrastructures. The scalability has to be understood in the context of exact optimization methods. It is possible to connect a large number of heterogeneous resources to a system that grows almost linearly in performance and meets multiple optimization goals applied to QoS Cloud metrics using appropriate algorithms. The emerging use of Artificial Intelligence (AI) techniques and a software-defined approach to resource management creates space for prompt and agile customization and management of the dynamic needs of workflows. These technologies address the optimal use of resources for dynamically requested services with an intelligent and centrally controlled approach.

The main research question of this paper is how to optimize centered resource management considering scalability in heterogeneous Cloud environments using the task scheduling approach. This paper focuses on performance evaluation and system scalability analysis of heterogeneous Cloud infrastructure executing resource-intensive tasks.

This work extends the author's preliminary work on modeling and simulating heterogeneous resources in the Cloud published in [5] in several directions. In this paper, we consider the Cloud for scientific research. The motivation for this research stems from the analysis of computing needs in High Energy Physics (HEP), focusing on a Large Hadron Collider's (LHC) [6] A Large Ion Collider Experiment (ALICE) experiment [7] workloads. The author is an ALICE Collaboration member. The LHC detectors operate at the European Laboratory for Particle Physics (CERN) [8], the world's largest particle laboratory. Executing the different HEP workloads on powerful resources can reduce the time needed to find a new scientific breakthrough. For this purpose, we analyze the ALICE simulation jobs and study how to improve performance and ensure scalability when processing them on heterogeneous Cloud data centers. To properly distribute the workload and ensure scalability in this environment, we analyze the resource usage of scientific tasks.

One of the objectives of this research is to optimize the task allocation in the modeled complex system of five heterogeneous data centers and to use the processing data from one production of the ALICE experiment. In this paper, the task scheduling problem is considered from various aspects.

The Evolution Strategies algorithm is chosen because of its property to adapt to the requirements of modern systems and different loads and the potential for intelligent resource management. Simulation is the chosen approach used to model a large-scale environment and test our solution. It offers an environment that provides good flexibility to anticipate the system behavior and test algorithms in a reproducible way.

We propose a novel task allocation approach using the nature-inspired Evolution Strategies algorithm, which has not been used so far in the field of Cloud resource management and optimization. Evolution Strategies algorithm is inspired by the theory of evolution and natural selection that uses random mutation, recombination, and selection based on the fitness function value and applies these steps to a population of individuals containing candidate solutions to evolve better solutions during the iterated procedure. We wanted to test and apply the properties of the Evolution Strategies algorithm in the allocation of tasks to virtual machines (VMs).

Another objective of the proposed model is to optimize task scheduling via the central broker. Therefore, the Longest Job First policy is used additionally to optimize resource allocation. This approach is chosen due to the complexity of task properties in the created workload. The broker assures the longest job priority in executing the resource-demanding tasks.

An extensive evaluation of the algorithm performance and comprehensive comparison with the state-of-the-art solution using a Cloud simulator is provided. In this work, we have contributed a task scheduling approach and explored the use of Evolution Strategies algorithm as an alternative to the popular Genetic Algorithm, the most used strategy from the same group of Evolutionary Algorithms. The results of the Evolution Strategies algorithm and Evolution Strategies with Longest Job First policy are compared with the results of the Genetic Algorithm based on the performance evaluation metrics which are makespan, average resource utilization, throughput, average execution time, degree of imbalance, scalability analysis, load distribution analysis, presented in Section IV. Evolution Strategies-based approaches have shown that they scale well with the number of available VMs and exhibit better results than the Genetic Algorithm.

The main contributions of this paper can be summarized as follows:

- Model built based on the actual observational data and the real Cloud system for scalable resource usage
- Novel adaptive metaheuristic based on the Evolution Strategies algorithm for the task scheduling problem in heterogeneous Cloud data centers
- Simulation-based evaluation and performance and scalability analysis of the proposed algorithm.

The rest of the paper is organized as follows. Section II describes the existing knowledge helping to understand the current state and existing problems. It also underlies the proposed approach. Section III provides the literature review on scheduling strategies in the heterogeneous Cloud. Section IV presents the experimental setup, performance metrics, and employed workloads. The proposed approach and heuristic algorithm are explained in Section V. The experimental results of the algorithm and overall performance are discussed

in Section VI. Finally, concluding remarks and future directions are given in Section VII.

II. BACKGROUND

In this section, we describe the background knowledge related to our work. This research is based on a software approach for optimizing performance and efficiency in a heterogeneous Cloud.

A. CERN AND WLCG

At the CERN, there are four large-scale LHC experiments, ATLAS, CMS, LHCb, and ALICE. Each experiment is named after a unique particle detector and tries to answer and explain specific primordial and particle physics phenomena. Furthermore, CERN is developing and improving many significant technologies applicable in various fields. Daily, LHC experiments collect and generate large amounts of data using highly sophisticated sensors and instruments. They generate more than 1 PB of data per second during particle collisions at high energy. Physicists and engineers analyze these data to explore the particles that compose the Universe. Significant storage and processing infrastructure capacities are needed to support their research. The CERN's data center has about 11000 servers, 470000 processor cores, 110000 disks, and 30000 tapes for long-term data storage [9]. Data from the LHC experiments are also distributed around the distributed computing and data storage infrastructure of more than 170 data centers called the Worldwide LHC Computing Grid (WLCG) [10]. The WLCG system is organized into Tiers providing special services. The Collaborations member institutes provide a computing environment for researchers. The centers are connected with a dedicated high-bandwidth network. The CERN Data Centre makes Tier 0 and is responsible for the safe-keeping and reconstruction of the raw data and distributing them to Tier 1s for further reprocessing and storage. The experiments use Tier 2s centers mainly for simulation. More than 300000 jobs run concurrently on the WLCG Grid.

We studied the resources used and pledged for the ALICE experiment. ALICE collaboration stores, processes, and analyzes data from several workflows, including real-data processing, simulation, reconstruction, and analysis. After the experiment upgrade, a significant increase in the number of collisions and the amount of generated data is expected. Therefore, it is necessary to provide resources for their future needs and configure them optimal to the workflow requirements. The Republic of Croatia is an associate member of CERN. Establishing a national scientific and educational e-infrastructure for HPC and Cloud computing named HR-ZOO (Croatian Scientific and Educational Cloud) could contribute to the computing and storage needs of LHC experiments at Tier 2. Typically, batch processing takes place within the WLCG Tier 2. This paper considers the ALICE simulation jobs for processing on the HR-ZOO infrastructure.

B. HR-ZOO

In [5], the HR-ZOO Cloud environment was modeled and simulated using the widely used Discrete-Event Simulator CloudSim simulator [11]. The system consists of five data centers distributed in four cities providing High Performance Computing (HPC), High Throughput Computing (HTC), High Scalability Computing (HSC), and data storage resources in the Cloud. The data centers are networked with high bandwidth links. In addition to the broadband backbone of the scientific and academic Cloud, the HR-ZOO national Cloud infrastructure will be connected with e-infrastructures in Europe and the world. Building this kind of infrastructure poses new challenges on many different levels. Different computing paradigms have different network requirements. The challenges in this heterogeneous Cloud will be related to resource management and ensuring balanced and optimal processing, networking, and storage.

C. SCALABILITY

Scalability is a core concern and the primary design challenge of every system. All system components are closely connected and share hardware resources, which with a large flow of data and a large number of users complicates scaling and creates performance problems. Future data centers will require adequate processing in the Cloud to effectively manage new challenges around scalability to effectively and efficiently make sense of the enormous amounts of data. Scalability is a frequently used term in the literature, but there is no uniform definition for scalability [12], [13]. It should be considered a multi-criteria optimization problem [14]. The steps for the analysis in the context of software development that also can be applied to system scalability analysis were proposed. These steps include identifying critical use cases, selecting representative scalability scenarios, determining scalability requirements, planning measurement studies, performing measurements, evaluating data, and presenting results. Scalability can be affected by a variety of properties and factors. Therefore, it is important to examine how it is impacted and at what point with a particular factor in a given environment. Horizontal scaling and vertical scaling are two ways to perform scalability and achieve proportional capacity growth in line with the increase in system load. Horizontal scaling (or scaling-out) is achieved through system upgrades by adding more server instances to the existing environment. In vertical scaling (or scaling-up), more computing power or storage is added to existing nodes, or it involves switching to a server with higher capacity. Scaling requires solutions to achieve a uniform load. If the system load is reduced, capacity should be reduced (scaled down) to adjust operating costs. Metrics for measuring the scalability of heterogeneous and homogeneous computing differ. Efforts for defining system scalability were based on metrics for system loads, system performance, resource utilization [15], and efficiency and speedup [16]. However, other attributes also need to be considered when assessing the potential for

system scalability. That includes performance, usability, cost, operability, reliability, security, availability, extensibility, and functionality.

Scalability is a vital Cloud feature. It should be considered and addressed from the resource demands and management perspectives. Scalability is a critical non-functional Cloud requirement that should be distinguished from Cloud elasticity. System scalability is the ability of a system to support an increase in workload by using additional resources over a while. At the same time, elasticity is the system's ability to adapt to changes in workload autonomously at any point in time [17]. Cloud autoscaling feature provides required QoS [18], [19]. With the introduction of emerging technologies in crucial parts of data center infrastructure, there is a need to analyze scalability in even more complex environments. Achieving scalability between several networked data center locations in the Cloud needs to be addressed. Traditional techniques used to efficiently manage Cloud systems assume different resource management and scheduling approaches and heuristics. Resource management in Cloud environments has to provide mechanisms for global scheduling, local scheduling, demand profiling, utilization estimation, pricing, application scaling, workload management, Cloud management, and measurement studies [20]. Scalability is affected by the choice of algorithm for resource management to make more effective use of all sites' capabilities and efficiently place tasks. Resource management involves the dynamic allocation of computing, networking, and storage resources needed for accomplishing applications. Resource provisioning and resource monitoring are just part of the resource management problems. Monitoring data from geographically distributed resources can serve for real-time reactions and resource allocation where and when resources are needed. Provisioning should analyze aspects of heterogeneity, speed, and volume of the generated data over time. Gathering information on computer facilities, network traffic, and the status and progress of many concurrently running tasks can help assess system scalability. The Cloud computing paradigm is based on virtualization applied at different levels. The management of resources is based on different scheduling algorithms and metaheuristics for controlling the use of shared resources.

D. MANAGEMENT AND SCHEDULING ALGORITHMS

Resource management is an essential function required for any system, especially for large-scale infrastructures such as Cloud systems. It is a process of planning, allocating, scheduling, and organizing efficient sharing of Cloud resources among multiple users with different requirements. The resource management of this infrastructure requires innovative, efficient, reliable, and adaptable solutions, with the minimum need for human supervision. An ideally managed system would be the one that would achieve maximum resource utilization, minimum energy consumption, reduced costs, and proper scaling. Resource management affects scalability, both horizontal scalability and vertical scalability. Several studies have been conducted to improve some Cloud properties (cost, power, latency) by applying scheduling algorithms [21], [22]. However, the authors have not found an appropriate systematic classification or survey of algorithms in this area. The choice of resource management algorithm affects performance, functionality, and cost. The objective of these environments, designed as geo-distributed infrastructures, is to optimize the management of primary resources and perform suitable computations. Admission control, capacity allocation, load balancing, energy optimization, and QoS guarantees are Cloud resource management policies that can be implemented through control theory, machine learning, and utility-based or market-oriented mechanisms [23]. Resource management activities, workflow scheduling, and task scheduling are NP-hard problems approached by different techniques [24]. Scheduling strategies are designed to support scaling and resource sharing. Often considered working resources are CPU cores and VMs or containers. Since VMs are the most crucial resource in the Cloud data center, VM consolidation is a relevant optimization problem. For overcoming this problem, it is necessary to consider various factors, from heterogeneity, task/workflow needs, infrastructure features, and QoS defined through service-level agreements.

Heuristic and metaheuristic are two main approaches on which algorithms are based. Scheduling algorithms are categorized as static and dynamic algorithms [25]. Resource management algorithms are used to solve optimization in real-time and dynamically adapt decisions to the actual behavior and the current state of the resources. The resources are granted to tasks through the task scheduling process. The scheduling algorithm determines the choice between available VMs. This process includes the selection of the site or Cloud for placing and running tasks. Tasks submitted to the Cloud are mapped to matching resources. Tasks will be executed on selected VMs based on task properties. Workflow and task scheduling approaches are classified using different criteria. Tasks scheduling algorithms can be classified according to the Cloud environment, VM type, data set and application type, execution time, cost, and power [26]. An efficient task scheduling algorithm impacts the system performance and has to satisfy some imposed objective functions. Assigning tasks to selected machines aims to minimize the makespan and total task completion time that affects running price and cost related to the task processing on different Cloud sites while achieving load balancing at each site. Different resource management policies for computing and communications in a Cloud have been evaluated. In this research area, algorithms are based on preemptive or non-preemptive scheduling, depending on the type of application. Vast amounts of data are crucial to AI. AI methods are being introduced in Cloud task scheduling to improve efficiency and control scalability. The strategies inspired by natural, evolutional, biological, physical, or social systems may be applied to discrete or continuous domain problems. The most common implementations are: Round-Robin (RR),

priority-based, First Come First Served (FCFS), Shortest Job First (SJF), Max-Min, Min-Min, Minimum Execution Time (MET), Heterogeneous Earliest Finish Time (HEFT), Earliest Deadline First (EDF), Tabu Search (TS), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). These algorithms serve as reference scheduling algorithms with which different algorithms are compared. However, there is still a significant scope for further improving resource management components. It is necessary to improve the limitations of existing solutions in terms of performance and scalability in changing conditions, taking into account the location of resources and network conditions.

III. RELATED WORK

This section analyzes relevant studies in heterogeneous Cloud computing and related resource management and scheduling strategies. The system performances of the models proposed in the surveyed papers were predominantly evaluated using simulation.

Several review surveys have been done on Cloud resource management, including resource provisioning and task scheduling. The related definitions and classification taxonomy were presented in a study on resource management and its components [27]. The survey focused on addressing dynamic heterogeneous environments with data-intensive workflows. Data-intensive loads, hybrid and multi-Cloud scenarios, rescheduling and performance fluctuations, and reliability must be addressed to enable task execution while optimizing infrastructural efficiency.

Numerous strategies aimed to improve task scheduling and consequently affect load balancing improvement in different Cloud environments. Approaches to solving task scheduling problems implement various heuristic, metaheuristic, and hybrid algorithms [28], among which metaheuristics [29] algorithms predominate. Dynamic strategies based on historical data scheduling requirements [30] or VM status at runtime [31] have improved performance management, resource use, and task response time. For example, the authors in [32] proposed a load balancing algorithm to optimize the QoS task parameters addressing load migration among VMs. The task scheduling process and load balancing approach considered deadline and completion time, VM priority, and resource allocation. The simulation results showed that the proposed algorithm reduces makespan, improves resource utilization, and achieves less execution time.

Task scheduling algorithms are widely based on stochastic optimization algorithms. The most common task scheduling algorithms belong to the fields of Swarm Intelligence and Evolutionary Algorithms. Particle Swarm Optimisation, Ant Colony, Tabu Search, and Harmony Search from the Swarm Intelligence field are adaptive strategies applied to task scheduling optimization. The group of Evolutionary Algorithms includes the Genetic Algorithm and Evolution Strategies, among other well-known algorithms from the same field. Swarm-based optimization algorithms mimic the social behavior of biological systems. The current position of each individual in the randomly initialized swarm is updated based on the social interactions in the swarm. Individuals adapt to the environment by constantly looking for better positions over time. Evolutionary Algorithms are based on the principles of natural evolution. They use evolutionary operators such as selection, crossover, and mutation to find an optimal solution through generations of a randomly initialized population. Approaches for solving optimization scheduling issues based on an adapted PSO algorithm were used [33]–[35] to better balance local and global search. Enhanced Ant Colony optimization algorithms were used to ensure the VM load balance [36] and enhance the performance of the task scheduler [37]. The application of the stochastic Tabu Search algorithm was examined in [38] as a job scheduling policy to solve the resource allocation and task scheduling problem in Grid/Cloud networks. A hybrid job scheduling algorithm merged Tabu and Harmony Search algorithms [39] and achieved satisfactory results in terms of makespan and cost.

The Genetic Algorithm is the most common Evolutionary Algorithm in the domain of Cloud scheduling. Duan *et al.* [40] modeled the task scheduling as an objective optimization problem and proposed an Adaptive Incremental Genetic Algorithm (AIGA). The main contribution of the algorithm that used the different mutation and crossover rates to achieve minimum makespan is progress in some measured parameters compared to other relevant algorithms. Multi-objective Balancer Genetic Algorithm (BGA) [41] optimization attempted to improve makespan and load balancing for tasks arriving in batches. The Modified Genetic Algorithm combined with Greedy Strategy (MGGS) [42], a hybrid Genetic Algorithm, was proposed to improve makespan and load balancing. It considered the expected total time for the VM to execute all assigned tasks as the fitness criteria.

We also briefly overviewed the research of software-defined approaches in the Cloud based on heterogeneous resources and highlighted several main ideas and directions of research. Intelligent adaptive algorithms have an essential role in Cloud environments that apply the dynamic and configurable software-defined approach for the software-based central control, managing, and optimizing usage of remote hardware resources (network, storage, CPU) for different system architectures. Concerning network resources, Software-Defined Networking (SDN) is a widely adopted agile approach for dynamic environments that require constant adaptation to achieve greater efficiency. The SDN is used to automatically provision and manage network traffic and balance load, which is especially important in Cloud data center environments. An essential feature of SDN is the separation of control and forwarding functions from the infrastructure level. Control-level software services enable automated, programmable, centralized, and remote management of network infrastructure that facilitates the application of new technologies, new features, and network elements. In a study on utilizing a software-defined approach for Cloud computing and related taxonomy [43], energy efficiency,

security, virtualization, and performance (network throughput, latency, availability, and QoS) were highlighted as common challenges for large-scale Cloud data centers. According to the given taxonomy, the objective of our research can be defined as improving performance in heterogeneously configured inter-Cloud data center architecture that will focus on joint optimization of resources for batch processing. Further, an adaptive Genetic Algorithm for resource allocation and VM placement in SDN-based Cloud data centers was proposed in [44]. The proposed model considered energy, VMs, and intra-data center communication costs. Results showed that the proposed model optimized application deployment cost, intra-data center communication cost, and power consumption. Regarding HPC Cloud, the research [45] highlighted challenges in HPC Cloud viability, performance optimization, and usability. An appropriate environment for ensuring the expected QoS cost-effective model and balancing resource requirements can be improved by using resource virtualization technologies and proper resource allocating algorithms, focusing on the hybrid Cloud.

Based on the in-depth literature review, we concluded that a Genetic Algorithm, a metaheuristic algorithm, is a dominant algorithm for optimization purposes. However, when considering existing and proposed solutions for managing the scalability of such systems, further improvements could still be made to increase scalability. The survey shows that the literature has a significant number of works with a different focus on cloud resource management. The number of works introducing centralized approaches for task scheduling among geographically-distributed heterogeneous data centers is quite limited in cloud resource management. Load balancing is often not considered but is significant for VM-task allocation when dealing with the heterogeneity of tasks and resources. We have found a limited amount of work considering using both performance metrics and the scalability property. The reviewed related literature has used a smaller number of metrics in individual studies.

We consider that algorithms that mimic natural processes will have a future significance as intelligent optimization and resource management methods, especially in AI and reinforcement learning. Furthermore, a review of the works has shown that the Evolution Strategies algorithm has not been applied so far in resource and task scheduling, and it certainly has the potential to optimize the use and load of resources. To better assess the effectiveness of the algorithm, we have expanded the number of metrics. We have modeled and simulated a larger number of resources and used a workload with tens of thousands of tasks, which is a significantly larger number of tasks than is the case in most existing papers dealing with resource allocation approaches. Those properties affect the scalability of the system.

IV. SYSTEM MODEL

This section describes the concept of our system model for simulation that includes the setup and realization of a system model, performance metrics determination, and workload creation.

TABLE 1. Data center and virtual machine specifications.

VM INSTANCE TYPES	1	2	3
CPU CORES	2	4	8
CPU [MIPS]	1000-1500	1000-1500	2000-2350
RAM [GB]	8	8	35
STORAGE [GB]	10	10	10
BANDWIDTH [Gbps]	1	1	1
DATA CENTER TYPE	1, 2, 3	1, 2	2
DATA CENTER INSTANCES	**1**	**2**	**3**
PARADIGMS	HSC, Storage	HPC, HSC, HTC	HSC, HTC, Storage
NUMBER OF HOSTS	15	40	5
NUMBER OF CPU CORES	64	52047	1532
CPU [MIPS]	2900	2200/2450/2900	2200/2900
TOTAL RAM [GB]	10000	171389	14000
TOTAL STORAGE [GB]	600000	7357165	303800

A. EXPERIMENTAL SETUP

The proposed algorithm is extensively evaluated based on a simulation environment with the created workload. The experiments were performed by CloudSim version 4.0 integrated with the Eclipse environment and executed in a 64-bit Windows 10 Operating System running on an Intel Core i5-1035G1 at 1.00 GHz with 4 cores and 8 GB of RAM. CloudSim supports model and resource management development in the Cloud environment using available classes and enables the implementation of models with different features. The Cloud infrastructure entities are modeled using CloudSim entities for data centers, hosts, VMs, tasks (cloudlets), inter-host agreements, and VM allocation policies. The proposed solution integrates the approximation Evolution Strategies algorithm to assign tasks to resources. The model represents key characteristics of the selected heterogeneous system. We set up a system configuration close to a realistic geo-distributed and heterogeneous Cloud computing environment. Many geographically distributed resources with heterogeneous type and capacity characteristics of resources are simulated. The simulated system consists of 5 connected data centers composed of physical machines in the same geographical region and time zone. Experiments were implemented with a total of 2100 VMs. Different amounts of cloudlets were processed on the simulation platform varying from 1000 to 20000 from created workload, detailed in D. The characteristics of VMs and data centers used in experiments are summarized in Table 1, respectively. The characteristics reflect the hardware capacity of instance types. Different sized VM instance types comprised of varied CPU, memory, storage, and network capacities are used with space shared policy. With their availability, scaling may provide significant improvements in resource utilization. A function for loading tasks with the appropriate values from the workload in CSV format has been created. General-purpose computing performance is measured by Millions of Instructions Per Second (MIPS). HPC nodes are constructed with 64 cores and a base clock of 2.45 GHz modeled on the

features of the world's highest-performing server CPU for general-purpose computing and solving advanced scientific problems and compute-intensive models [46]. The number of total CPU cores per HPC host is adjusted according to the performance of the HPC server processor.

B. PERFORMANCE METRICS

Among several indicators to assess the heuristic's performance and scalability, the following metrics are considered to measure the performance of our proposed model.

- Makespan (MS) is the often-used metric [29] for scheduling efficiency on parallel machines concerning the time required to complete submitted tasks. Minimizing makespan is related to assigning tasks to virtual machines (VM). It is defined as the maximum execution time (ET) required to get tasks j finished on a single assigned virtual machine i:

$$ET_{VM_i} = \sum_{j=1}^{m} ET_i(task_j), \text{ where i} \in \text{VMs} \quad (1)$$

$$MS = Max(ET_{VM_1}, ET_{VM_2}, ET_{VM_3}, \ldots, ET_{VM_n}) \quad (2)$$

- Average Resource Utilization (RU_{avg}) is an essential indicator of the efficiency of utilizing resources in the Cloud system. Optimizing average resource utilization means achieving equal balance in the Cloud data centers. This quantitative metric is calculated using the equation below [29]. MS denotes total Makespan, ET denotes total execution time of VM resources that process tasks, and n denotes the number of VM resources.

$$RU_{avg} = \frac{\sum_{i=1}^{n} ET_{VM_i}}{MS \times n} \quad (3)$$

- Throughput (T) is defined as the number of tasks N processed per unit time (second) [31]. The aim is to maximize throughput, which characterizes CPU efficiency.

$$T = \frac{N_{tasks}}{MS} \quad (4)$$

- Average Execution Time (ET_{avg}) is an indicator of the efficiency of utilizing resources in the Cloud system. It is a time spent by the task actively using the VM as a Cloud resource. This quantitative metric is calculated using the following equation [32]. ET denotes the execution time of task j, and m denotes the number of tasks.

$$ET_{avg} = \frac{\sum_{j=1}^{m} ET(task_j)}{m} \quad (5)$$

- Degree of Imbalance (DI), employed in [47] according to the formula below, is the parameter that affects the load-balancing performance between virtual resources. The formula is based on execution times on all VMs and includes maximum execution time (T_{max}), minimum execution time (T_{min}), and average execution time (T_{avg}).

$$DI = \frac{T_{max} - T_{min}}{T_{avg}} \quad (6)$$

- Number of VM instances over time (scale-out and scale-in)

C. EMPLOYED WORKLOADS

Our approach focuses on the Cloud environment for scientific research. For this purpose, we analyzed the resource requirements of the CERN ALICE experiment. The significant change in the experiment settings was an incentive to create scientific workloads with current data. The test workload used in the experiment consists of computationally intensive tasks related to the ALICE simulation production of Monte Carlo events. The data of one proton-lead (p-Pb) Monte Carlo production master job from the current data-taking period (February 2022) have been extracted from the ALICE grid monitoring system. We created a synthetic workload by utilizing Monte Carlo logs data with the 49026 production jobs (tasks) distributed and processed in more than 60 locations. We extracted and adapted the raw workload data to fit the corresponding properties of the Standard Workload Format [48] and, consequently, the used simulation tool. A created workload is in CSV format, which is the most used today in data analysis. The created file contains 18 columns characterizing tasks' properties accordant to the standard workload format. The job characteristics are: task ID, submit/wait/run/requested time, number of CPU cores (allocated and requested), average CPU time used, used/requested memory, status, user/group ID, executable number, queue number, partition number, preceding job number, and think time from the preceding job. The jobs are submitted dynamically. Submit time is approximated to real-time job submission. All ALICE jobs are submitted to the central task queues. In two previous periods of data taking, jobs coming to WLCG sites were assigned to a VM configured in advance for processing (with one CPU core and 2 GB of RAM per core). To incorporate HPC and Cloud resources, multicore VMs are used for testing. Jobs in the log are modeled to require either 1 or 8 cores. ALICE jobs are represented as tasks (cloudlets) in simulation based on the CloudSim framework. The execution length of the cloudlet varies from 2340 to 60780 (in Million Instructions). VM placement strategy is needed to optimize the workload distribution in a heterogeneous environment. The LCG workload log [49] was used in previous research. It contains jobs of 11 days of activity in 2005 from multiple has 188041 jobs, while our from only one master job has about 50000 jobs. By comparing the LCG workload log and the workload log we created from the current Monte Carlo production master job, it can be concluded that many more physics events are being generated over time, resulting in more data and more running jobs. Compared to previous

data-taking periods, a significant increase in the complexity and amount of data was observed.

V. PROPOSED METAHEURISTIC ALGORITHM

Metaheuristic algorithms are problem independent and give good performance for high-level optimization problems from different domains, both in scientific research and practical implementation. Often used, hybrid metaheuristics combine metaheuristic algorithms with the ideas and components of other algorithms. In this section, the heterogeneity-aware metaheuristic algorithm based on principles of the Evolution Strategies algorithm is proposed for solving the task scheduling problem in Cloud computing.

A. EVOLUTION STRATEGIES ALGORITHM

The Evolution Strategies algorithm aims to improve efficiency by selecting suitable candidates with the objective function from the domain. That is, in this domain, the selection of appropriate resources to accept the processing of tasks of specific characteristics. The optimization goals are to minimize the makespan, maximize resource use, and execute all tasks assigned to heterogeneous Cloud resources through the iterative Evolution Strategies process. Evolution Strategies is a population-based metaheuristic algorithm introduced by Rechenberg and Schwefel [50]. The Evolution Strategies technique uses the ideas of the biological evolution process - natural selection and reproduction. Evolution is the result of natural selection where only those individuals who are the best in the struggle for life resources survive. This global optimization technique belongs to the Evolutionary Computation field. The Evolutionary Computation field is a branch of AI that uses a group of Evolutionary Algorithms to which the Evolution Strategies algorithm and Genetic Algorithm belong. Both methods most likely find a solution close to optimal and have applications in many different areas. The main difference between the Genetic Algorithm and Evolution Strategies algorithm is in using selection mechanisms. Evolution Strategies algorithms use selection and mutation operator to get the overall fittest solution, while Genetic Algorithms use selection, crossover, and mutation to search for optimization space.

The population in multimembered (μ, λ)-Evolution Strategies algorithm consists of μ parents, which after the process of mutation, give λ offspring, with the condition of $\lambda > \mu$. Each of the λ offspring has its factor of goodness. Using mutation operators, created λ children form the next generation's population. The mutation operator modifies the selected solution to create offspring and thus maintains the diversity of the population. Offspring individuals are created by a randomly generated change in the parent. For the transition to the new generation, λ offspring are observed. From them, μ of the individuals are passed on to the next generation. Each iteration represents a generational step. During the evaluation process, the individuals for the next iteration of the algorithm are evaluated by applying the fitness function that determines how optimal/suitable a solution is. The numerical

Algorithm Pseudocode of Evolution Strategies algorithm

Input: $(\mu, \lambda, \text{GenerationSize}, \text{CloudletList}, \text{VirtualMachineList})$
Output: S_{best} (Mapping of cloudlets to appropriate VMs)

```
begin
Population = InitializePopulation(Random, pairs < Cloudlet, VM >)
EvaluatePopulation(Population)
S_best = FindBest(Fitness)
add S_best to BestSolution
while ¬StopCondition(GenerationSize) do
    for i = 0 to λ do
        Parent = SelectParent(Population, μ)
        Child_i = Mutate
        add Child_i to Children
    end
    EvaluatePopulation(Children)
    S_best = FindBest(Fitness)
    add S_best to BestSolution
    Population = Children
end
return best S_best from BestSolution
```

FIGURE 1. Pseudocode of the applied (μ, λ)-Evolution Strategies algorithm.

values (fitness value) of each individual that describe the suitability of individuals are compared. Therefore, in (μ, λ)-Evolution Strategies, individuals survive only one generation. This procedure proceeds until fulfilling the stop condition – a predefined number of generations.

In an effort to find and preserve the optimal solution from each generation, we have added the principle of elitism to the existing basic steps of the Evolution Strategies algorithm.

The implementation of the (μ, λ)-Evolution Strategies approach to task allocation contains the following steps:

1) Initialization of the population
   The method used for initializing a population was random initialization, generating random solutions that strive for population optimality. The population is defined as pairs of cloudlets (read from workload file) and VMs. Every population element is a vector that contains n parameters $X = (x_1, x_2, x_3, \ldots, x_n)$, where $x_i = (cloudlet, VM), x_i \in X$.
2) Evaluation of population
   Each individual of the population is evaluated by the fitness function (FF) that considers the capacity of VMs and length of cloudlets defined as:

$$FF = \frac{CloudletLength}{VMProcessingSpeed} \tag{7}$$

3) Selection
   μ parents are selected from the population for the further process of creating offspring for each selected parent.
4) Mutation
   Cloudlets are unique, and VM may vary. For creating λ offspring, we use a random resetting mutation operator where a random VM from the set of permissible values is assigned to a randomly selected cloudlet. In this research, mutations occur in 10% of the individual population unit. While selecting

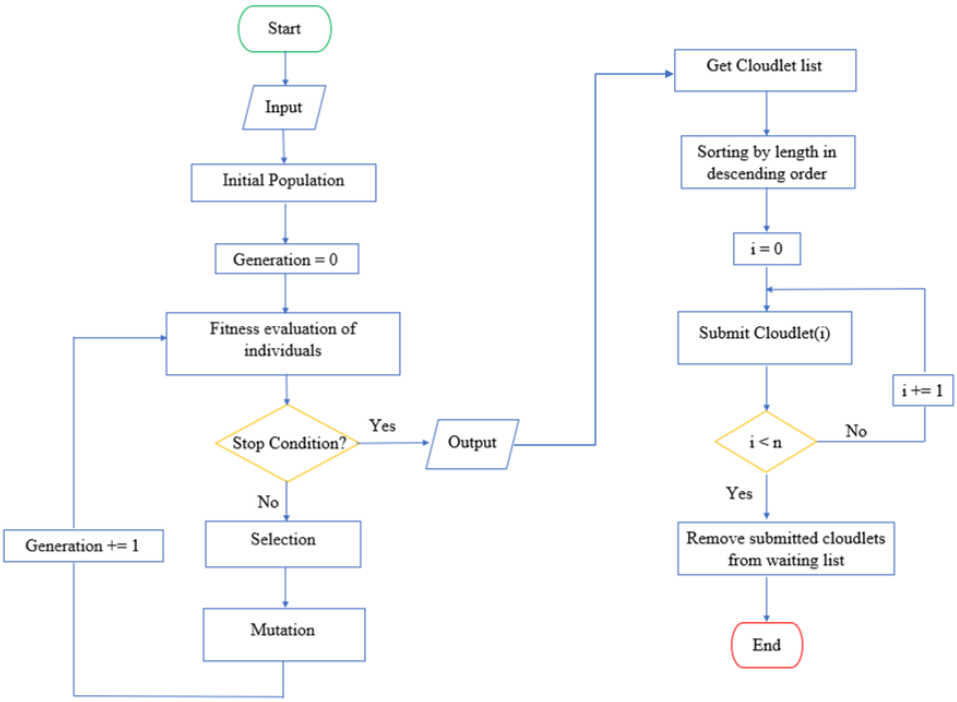

FIGURE 2. Flowchart of the Evolution Strategies-based task scheduling and Longest Job First broker policy.

parameters for the Evolution Strategies algorithm, we conducted a preliminary study and tested different combinations of parameters. We implemented the $\frac{1}{7}$-rule to the parent-offspring ratio because it gave optimal results compared to the other tested combinations of parent-offspring pair parameters. According to the previous studies [51], [52], this ratio is the optimal and recommended ratio for maintaining high selective pressure.

5) Elitism
The elitism principle is applied to preserve the most optimal candidate solution in the population for as long as possible. The elitism method enables saving and passing the fittest solutions from each population. At the end of the algorithm, the overall best solution is the fittest solution from every generation, and the cloudlet is assigned to the chosen VM.

Fig. 1 presents the pseudocode of the proposed Evolution Strategies algorithm.

B. LONGEST JOB FIRST DATA CENTER BROKER POLICY

The cloudlet list is submitted to the central broker to distribute them to data centers and bind cloudlets to assigned VMs. The Longest Job First scheduling broker policy was applied to optimize the load balancing and manage resource utilization for processing compute-intensive tasks. This non-preemptive load balancing policy was adapted to prioritize the processes having greater cloudlet length. Cloudlets are sorted in descending order of their pre-assigned instruction length and submitted to the VMs specified by the Evolution Strategies procedure. This centrally defined Longest Job First load balancing policy maximizes system utilization for high system loads. Fig. 2 shows the flowchart of the Evolution Strategies task scheduling metaheuristic algorithm with the Longest Job First data center broker policy. The validation of the Evolution Strategies-based approaches will be explained in the following section.

VI. EXPERIMENTAL RESULTS AND PERFORMANCE AND SCALABILITY ANALYSIS

This section shows the empirical results of our proposed Evolution Strategies algorithm when tackling the task scheduling problem in a heterogeneous Cloud computing environment. Our approach consists of two following implementations based on the principles of the (μ, λ)-Evolution Strategies algorithm:

- Evolution Strategies task scheduling
- Evolution Strategies task scheduling with Longest Job First load balancing.

In order to measure the performance of our proposed approach more accurately, we employed simulation with the data set described in Section IV. Furthermore, the performance of the proposed algorithms was compared with a Genetic Algorithm, a more commonly used algorithm from the same group of algorithms. The main objectives of the proposed algorithm for task scheduling and load balancing in the context of the considered metrics are to improve the use and allocation of Cloud resources, reduce makespan, latency, and

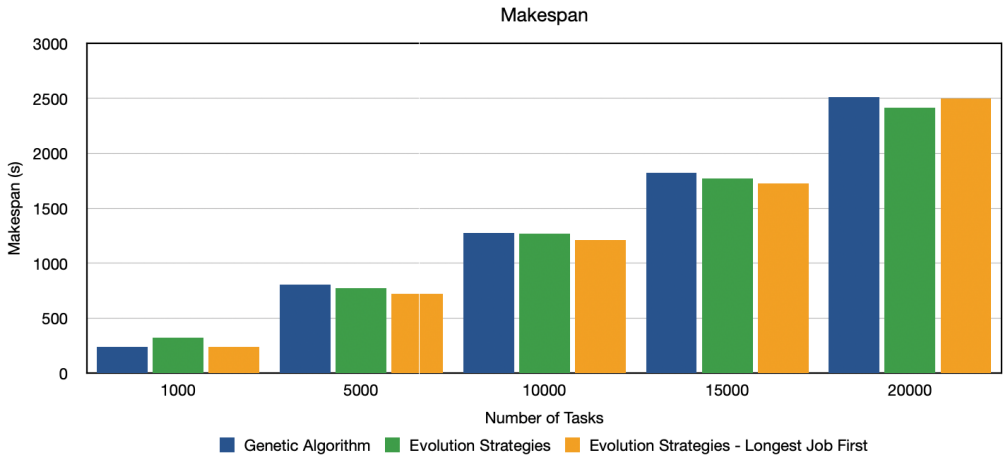

FIGURE 3. Makespan.

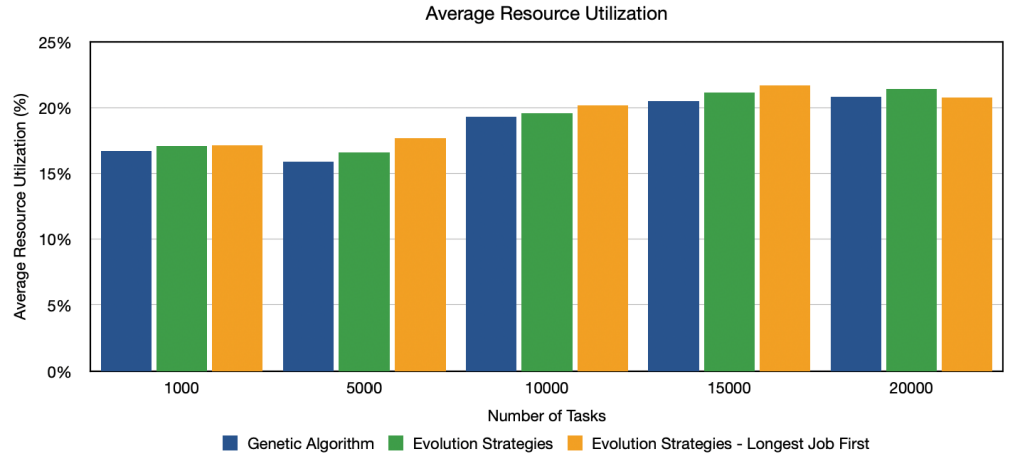

FIGURE 4. Average Resource Utilization.

the mismatch between VM resource loads. We executed tests with different numbers of tasks (1000, 5000, 10000, 15000, and 20000 tasks) to evaluate the performance and scalability of the proposed metaheuristic algorithm. Each policy was evaluated 10 times, and it gave average results. The results of three algorithms were compared based on the following performance evaluation metrics presented in Section IV:

- Makespan
- Average Resource Utilization
- Throughput
- Average Execution Time
- Degree of Imbalance
- Scalability analysis
- Load distribution analysis.

Our algorithm considers the task characteristic and selects a host with the most appropriate VM instance. The following figures visualize performance indicators for each policy.

Makespan represents the time difference between a task list's start and finish times. The reduced makespan indicates better performance. The graph in Fig. 3 indicates the makespan values of the algorithms. The figure shows that the makespan of the Evolution Strategies-based algorithms outperforms the Genetic Algorithm. Compared to the two policies, the Evolution Strategies with Longest Job First approach shows better results overall with the lowest makespan value. The values show an expected increase of overall makespan due to workload increase.

A comparative analysis of resource usage between Evolution Strategies and the state-of-the-art algorithm is shown in Fig. 4. Resource utilization shows the availability of resources, and it is a good indicator of overall resource efficiency. Evolution Strategies approaches outperform the Genetic Algorithm, with Evolution Strategies with Longest Job First having a higher resource utilization value than the other approaches. The observed data center system in the Cloud has a large capacity of resources and is intended for use in various scientific research. Such percentages correspond to the purpose of data centers, where the used tasks would be

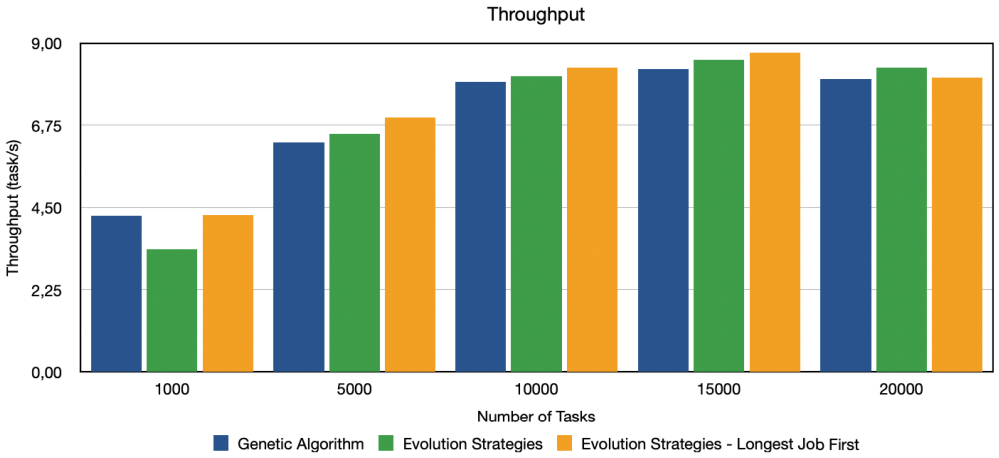

FIGURE 5. Throughput.

FIGURE 6. Average Execution Time.

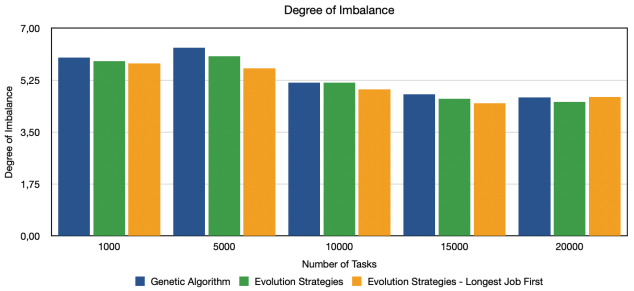

FIGURE 7. Degree of Imbalance.

just a part of the workload. It can be noticed that the resource utilization value increases with more tasks.

Throughput is a crucial metric to evaluate and compare the performance of our algorithm. It is an important attribute to consider when testing and analyzing scalability at different workloads. Fig. 5 shows the achieved throughput for all compared policies. The results display that the Evolution Strategies approaches, particularly Evolution Strategies with Longest Job First approach, have improved the throughput for every test.

Time-related parameters have a significant impact on delivering high performance to Cloud users. For example, Fig. 6 shows no significant deviations in the average execution time of the cloudlets measured in seconds between the Genetic Algorithm and Evolution Strategies approaches.

Dynamic allocation of tasks reduces the values of the degree of imbalance. Fig. 7 shows a desired lower degree of imbalance of Evolution Strategies approaches, especially Evolution Strategies with Longest Job First.

The system successfully processes the increased number of tasks in terms of scalability. Fig. 8 and Fig. 9 show that VM instances in the Cloud system are rescaled periodically. According to task execution, resource capacity is provisioned dynamically (added and removed). Scalability is achieved across all 5 data centers (horizontally) and by the number of VMs on the hosts of all data centers (vertically). The Evolution Strategies and the Evolution Strategies with Longest Job First approaches exhibit similar scalability performance. These approaches have a common task allocation principle that follows the algorithm of Evolution Strategies. Such scalability behavior is expected as Evolution Strategies-based algorithms assign tasks to VMs and broker distributes tasks over assigned VMs. Evolution Strategies-based algorithms affect changes in the number of VMs and type of VMs and thus affect scalability. Longest Job First principle affects load balancing and prioritizes the longest tasks, but it does not affect the number and choice of VMs to be used. High-capacity data centers receive proportionally more tasks. Therefore, load in data centers is balanced across 5 data centers (DC) in all test scenarios, as seen in Fig. 10 and Fig. 11. That contributes to system scalability. Analyzing evaluated cases shown in Fig. 8 and Fig. 9, and Fig. 10 and Fig. 11, it can be concluded that VM instances are scaled during measured intervals over hosts and data centers depending on the number of currently active tasks.

Due to the large capacity of the data centers and the resource settings set in the simulator, optimized metric values of the measured metrics were achieved. In almost all test scenarios, Evolution Strategies-based approaches performed better.

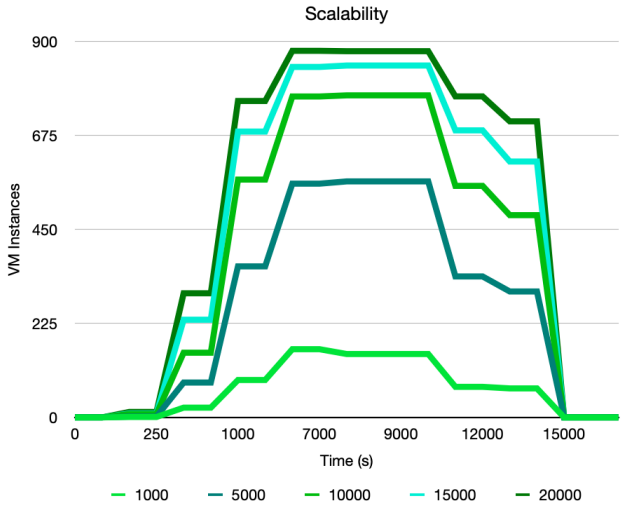

FIGURE 8. Scalability analysis of Evolution Strategies algorithm.

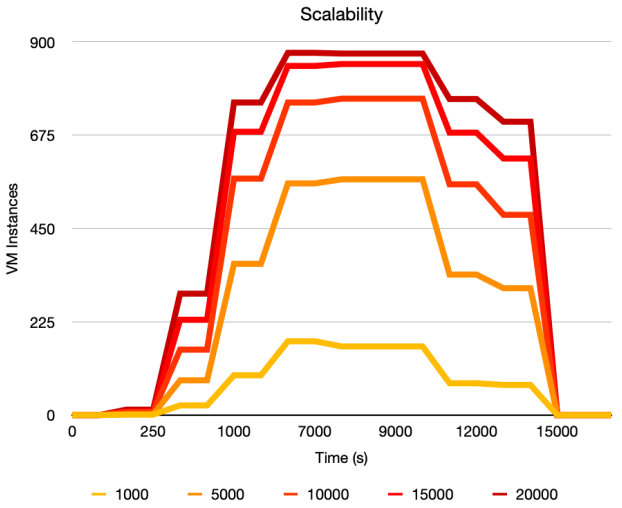

FIGURE 9. Scalability analysis of Evolution Strategies algorithm with Longest Job First data center broker policy.

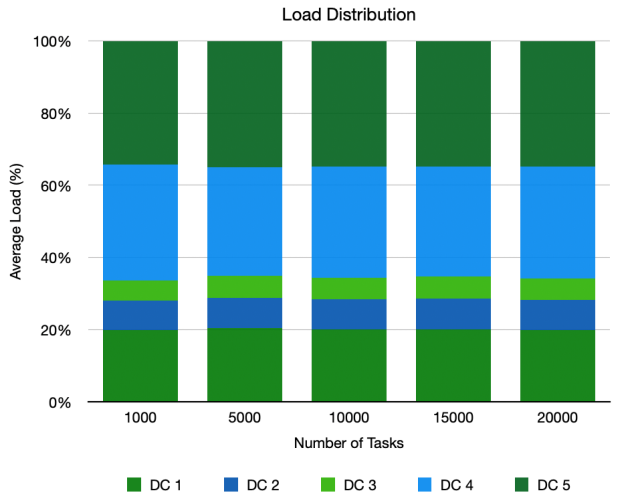

FIGURE 10. Data center load distribution for Evolution Strategies algorithm.

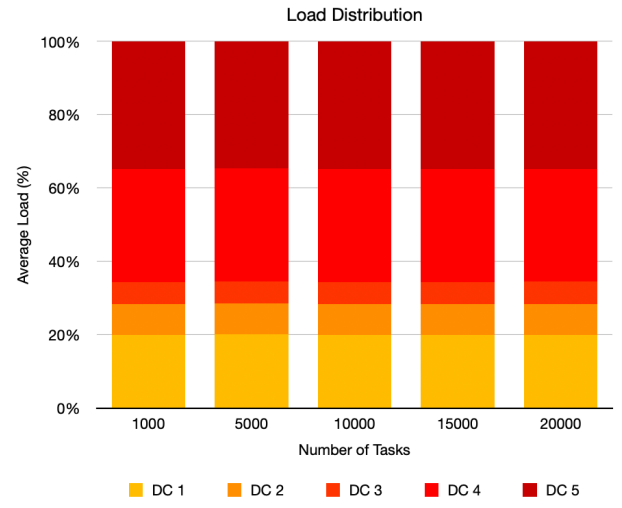

FIGURE 11. Data center load distribution for Evolution Strategies algorithm with Longest Job First broker policy.

VII. CONCLUSION

This paper proposes an implementation of the (μ, λ)-Evolution Strategies metaheuristic algorithm adapted for tackling the task scheduling challenge in heterogeneous Cloud computing environments. The presented research has conducted task scheduling simulations in the CloudSim framework. The simulation workload was created based on ALICE jobs from one production. Different simulation tests have been performed to establish and validate the performance of the proposed Evolution Strategies approach. Experiments show that the Evolution Strategies task scheduling with implemented Largest Job First broker policy model is reliable. It achieves substantially better performance than Genetic Algorithm and Evolution Strategies metaheuristics. The proposed task scheduling algorithm can optimally schedule the tasks to the VMs. The Evolution Strategies-based approach minimizes makespan, reduces average execution time and imbalance, increases resource utilization and throughput, and achieves scalability. Dynamic task allocation and managing heterogeneous high-capacity data center resources improve the system performance in all scenarios. We conclude that the proposed solution achieves scalability in using Cloud resources in different data centers. Evolution Strategies algorithm takes an important place in AI and has the potential to answer the performance challenges of standard reinforcement learning techniques.

In future research work, it is intended to optimize the algorithm's fitness operator by integrating more workload characteristics and resource characteristics for better solutions. The aspect of designing a proactive Evolution Strategies-based task scheduling algorithm that balances the task assignment time, task completion time, cost, and load balancing is considered for future work. The algorithm will be extensively compared to the related state-of-the-art algorithms.

REFERENCES

[1] *IDC Forecasts Worldwide 'Whole Cloud' Spending to Reach $1.3 Trillion by 2025*. Accessed: Mar. 2022. [Online], Available: https://www.idc.com/getdoc.jsp?containerId=prUS48208321

[2] *Croatian Scientific and Educational Cloud (HR-ZOO)*. Accessed: Mar. 2022. [Online]. Available: https://www.srce.unizg.hr/hr-zoo/en

[3] *Gaia-X*. Accessed: Mar. 2022. [Online]. Available: https://www.gaia-x.eu/

[4] R. Buyya, "A manifesto for future generation cloud computing: Research directions for the next decade," *ACM Comput. Surv.*, vol. 51, no. 5, Sep. 2019, Art. no. 105.

[5] P. Loncar, "Modeling and simulation of heterogeneous resources in the cloud: (Work in progress)," in *Proc. IEEE 20th Int. Symp. Netw. Comput. Appl. (NCA)*, Nov. 2021, pp. 1–3.

[6] *Large Hadron Collider*. Accessed: Mar. 2022. [Online]. Available: https://home.cern/science/accelerators/large-hadron-collider

[7] *A Large Ion Collider Experiment*. Accessed: Mar. 2022. [Online]. Available: https://alice.cern/

[8] *European Organization for Nuclear Research (CERN)*. Accessed: Mar. 2022. [Online]. Available: https://home.cern/

[9] *CERN Data Center*. Accessed: Mar. 2022. [Online]. Available: https://information-technology.web.cern.ch/about/data-centre

[10] *Worldwide LHC Computing Grid*. Accessed: Mar. 2022. [Online]. Available: https://wlcg.web.cern.ch/

[11] R. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[12] M. D. Hill, "What is scalability?" *ACM SIGARCH Comput. Archit. News*, vol. 18, no. 4, pp. 18–21, Dec. 1990.

[13] C. Weinstock and J. Goodenough, "On system scalability," Softw. Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Note CMU/SEI-2006-TN-012, 2006. [Online]. Available: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=7887

[14] L. Duboc, D. Rosenblum, and T. Wicks, "A framework for characterization and analysis of software system scalability," in *Proc. 6th Joint Meeting Eur. Softw. Eng. Conf. ACM SIGSOFT Symp. Found. Softw. Eng. (ESEC-FSE)*, Sep. 2007, pp. 375–384.

[15] J. Gao, P. Pattabhiraman, X. Bai, and W. T. Tsai, "SaaS performance and scalability evaluation in clouds," in *Proc. IEEE 6th Int. Symp. Service Oriented Syst. (SOSE)*, Dec. 2011, pp. 61–71.

[16] X.-H. Sun, Y. Chen, and M. Wu, "Scalability of heterogeneous computing," in *Proc. Int. Conf. Parallel Process. (ICPP)*, Jun. 2005, pp. 557–564.

[17] N. Herbst, S. Kounev, and R. Reussner, "Elasticity in cloud computing: What it is, and what it is not," in *Proc. Int. Conf. Autonomic Comput.*, Jun. 2017, pp. 23–27.

[18] C. Qu, R. N. Calheiros, and R. Buyya, "Auto-scaling web applications in clouds: A taxonomy and survey," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–33, Jul. 2019.

[19] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A review of auto-scaling techniques for elastic applications in cloud environments," *J. Grid Comput.*, vol. 12, no. 4, pp. 559–592, Dec. 2014.

[20] B. Jennings and R. Stadler, "Resource management in clouds: Survey and research challenges," *J. Netw. Syst. Manage.*, vol. 23, no. 3, pp. 567–619, Jul. 2015.

[21] M. A. Rodriguez and R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 222–235, Apr.-Jun. 2014.

[22] W. Ding, F. Luo, C. Gu, H. Lu, and Q. Zhou, "Performance-to-power ratio aware resource consolidation framework based on reinforcement learning in cloud data centers," *IEEE Access*, vol. 8, pp. 15472–15483, Jan. 2020.

[23] D. C. Marinescu, "Cloud resource management and scheduling," in *Cloud Computing: Theory Practice*, 2nd ed. Cambridge, MA, USA: Morgan Kaufmann, 2018, Ch. 9.

[24] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP–Completeness* (Series of Books in the Mathematical Sciences). New York, NY, USA: Freeman Company, 1979, p. 339.

[25] M. Kumar, S. Sharma, A. Goel, and S. Singh, "A comprehensive survey for scheduling techniques in cloud computing," *J. Netw. Comput. Appl.*, vol. 143, pp. 1–33, Oct. 2019.

[26] M. Masdari and M. Zangakani, "Efficient task and workflow scheduling in inter-cloud environments: Challenges and opportunities," *J. Supercomput.*, vol. 76, no. 1, pp. 499–535, Jan. 2020.

[27] N. M. Gonzalez, T. C. M. D. B. Carvalho, and C. C. Miers, "Cloud resource management: Towards efficient execution of large-scale scientific applications and workflows on complex infrastructures," *J. Cloud Comput.*, vol. 6, no. 1, p. 20, Jun. 2017.

[28] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," *J. King Saud Univ. Comput. Inf. Sci.*, p. 24, Mar. 2021, doi: 10.1016/j.jksuci.2021.02.007.

[29] M. Kalra and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," *Egyptian Informat. J.*, vol. 16, no. 3, pp. 275–295, Nov. 2015.

[30] P. Zhang and M. Zhou, "Dynamic cloud task scheduling based on a two-stage strategy," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 772–783, Apr. 2018.

[31] S. Nabi, M. Ibrahim, and J. M. Jimenez, "DRALBA: Dynamic and resource aware load balanced scheduling approach for cloud computing," *IEEE Access*, vol. 9, pp. 61283–61297, Apr. 2021.

[32] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, "A load balancing algorithm for the data centres to optimize cloud computing applications," *IEEE Access*, vol. 9, pp. 41731–41744, Mar. 2021.

[33] Z. Zhou, J. Chang, Z. Hu, J. Yu, and F. Li, "A modified PSO algorithm for task scheduling optimization in cloud computing," *Concurrency Comput. Pract. Exper.*, vol. 30, p. 11, Sep. 2018, Art. no. e4970.

[34] S. Nabi, M. Ahmad, M. Ibrahim, and H. Hamam, "AdPSO: Adaptive PSO-based task scheduling approach for cloud computing," *Sensors*, vol. 22, no. 3, Jan. 2022, Art. no. 920.

[35] N. Mansouri, B. M. H. Zade, and M. M. Javidi, "Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory," *Comput. Ind. Eng.*, vol. 130, pp. 597–633, Apr. 2019.

[36] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," *J. Ambient Intell. Humanized Comput.*, p. 12, Oct. 2020, doi: 10.1007/s12652-020-02614-7.

[37] Y. Moon, H. Yu, J.-M. Gil, and J. Lim, "A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments," *Hum. Centric Comput. Inf. Sci.*, vol. 7, no. 1, p. 10, Oct. 2017.

[38] P. Yi, H. Ding, and B. Ramamurthy, "A Tabu search based heuristic for optimized joint resource allocation and task scheduling in grid/clouds," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2013, pp. 1–3.

[39] H. Alazzam, E. Alhenawi, and R. Al-Sayyed, "A hybrid job scheduling algorithm based on Tabu and harmony search algorithms," *J. Supercomput.*, vol. 75, no. 12, pp. 7994–8011, Jun. 2019.

[40] K. Duan, S. Fong, S. Siu, W. Song, and S. Guan, "Adaptive incremental genetic algorithm for task scheduling in cloud environments," *Symmetry*, vol. 10, no. 5, May 2018, Art. no. 168.

[41] R. Gulbaz, A. B. Siddiqui, N. Anjum, A. A. Alotaibi, T. Althobaiti, and N. Ramzan, "Balancer genetic algorithm—A novel task scheduling optimization approach in cloud computing," *Appl. Sci.*, vol. 11, no. 14, Jul. 2021, Art. no. 6244.

[42] Z. Zhou, F. Li, H. Zhu, H. Xie, J. H. Abawajy, and M. U. Chowdhury, "An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments," *Neural Comput. Appl.*, vol. 32, no. 6, pp. 1531–1541, Mar. 2020.

[43] J. Son and R. Buyya, "A taxonomy of software-defined networking (SDN)-enabled cloud computing," *ACM Comput. Surv.*, vol. 51, no. 3, May 2019, Art. no. 59.

[44] S. Rawas, "Energy, network, and application-aware virtual machine placement model in SDN-enabled large scale cloud data centers," *Multimedia Tools Appl.*, vol. 80, pp. 15541–15562, Apr. 2021.

[45] M. A. S. Netto, R. N. Calheiros, E. R. Rodrigues, R. L. F. Cunha, and R. Buyya, "HPC cloud for scientific and business applications: Taxonomy, vision, and research challenges," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 1–29, Jan. 2019.

[46] *3rd Generation AMD EPYC Processors*. Accessed: Mar. 2022. [Online]. Available: https://www.amd.com/en/processors/epyc-7003-series

[47] M. A. Elaziz, S. Xiong, K. P. N. Jayasena, and L. Li, "Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution," *Knowl.-Based Syst.*, vol. 169, pp. 39–52, Apr. 2019.

[48] *The Standard Workload Format*. Accessed: Mar. 2022. [Online]. Available: https://www.cs.huji.ac.il/labs/parallel/workload/swf.html

[49] *The LCG Grid LOG*. Accessed: Mar. 2022. [Online]. Available: https://www.cs.huji.ac.il/labs/parallel/workload/l_lcg/index.html

[50] H. G. Beyer and H. P. Schwefel, "Evolution strategies—A comprehensive introduction," *Natural Comput.*, vol. 1, pp. 3–52, Mar. 2002.
[51] A. E. Eiben and J. E. Smith, "Popular evolutionary algorithm variants," in *Introduction to Evolutionary Computing* (Natural Computing Series), 2nd ed. Berlin, Germany: Springer, 2015, pp. 99–116.
[52] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms," in *Encyclopedia of Information Systems*, vol. 2. Amsterdam, The Netherlands: Elsevier, 2003, pp. 259–267.

PETRA LONCAR (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering and information technology and the M.S. degree in electronics and computer engineering from the University of Split, Croatia, in 2012 and 2014, respectively. She is currently pursuing the Ph.D. degree with the University of Split. She resided at Blaise Pascal University, Clermont-Ferrand, France, during her graduate studies. From 2014 to 2015, she was employed by Ericsson Nikola Tesla, Zagreb, Croatia. She worked on several projects abroad in the regions of Western and Central Europe. Since 2015, she has been a Research and Teaching Assistant with the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture (FESB), University of Split. Her research interests include big data, data science, cloud computing, and scalable resource management. She is a member of the ALICE Collaboration at the CERN, the European Organization for Nuclear Research. She has authored or coauthored several conference papers and many publications as a member of the ALICE Collaboration. She received the Dean's Commendation for Academic Excellence and the Rector's Award for Excellence.

PAULA LONCAR received the B.S. and M.S. degrees in computer science from the University of Split, Croatia. She has been a Teaching Assistant with the Faculty of Science (PMF), University of Split. Her research interests include data science, artificial intelligence, interaction design, and human–computer interaction. She has received the Dean's Award for Excellence.