The Journal of Supercomputing (2020) 76:6113–6139
https://doi.org/10.1007/s11227-020-03305-y

A cost saving and load balancing task scheduling model for computational biology in heterogeneous cloud datacenters

Wenwei Cai[1] · Jiaxian Zhu[1] · Weihua Bai[1] · Weiwei Lin[2] · Naqin Zhou[3] · Keqin Li[4]

Published online: 26 May 2020

Abstract
Cloud-based scientific workflow systems can play an important role in the development of cost-effective bioinformatics analysis applications. There are differences in the cost control and performance of many kinds of servers in heterogeneous cloud data centers for bioinformatics workflows running, which can lead to imbalance between operational/maintenance management costs and quality of service of server clusters. A task scheduling model that responds to the peaks and valleys of task sequencing—the number of tasks that arrive in a given unit of time—is related to indicators such as cost saving, load balancing and system performance (average task wait time, average response time and throughput). This study proposes a large-scale cost-saving and load-balancing scheduling model, called HDCBS, for the optimization of system throughput. First, queuing theory is used to model each computing node as an independent queuing system and to obtain the average system wait time and average task response time. Then, using convex optimization theory, a task assignment solution is proposed with a load-balancing mechanism. The validity of the task scheduling model is verified by simulation experiments, and the model performance is further validated through a comparison with other frequently used scheduling methods. The simulation results show that the credibility of HDCBS is greater than 95% in task scheduling.

Keywords Bioinformatics · Cost saving · Large-scale task scheduling · Load balancing · Queuing theory

✉ Weihua Bai
bandwerbai@gmalil.com

✉ Weiwei Lin
linww@scut.edu.cn

Extended author information available on the last page of the article

1 Introduction

At present, computer science solutions for molecular biology problems are often presented in the form of workflows. Cloud-based scientific workflow systems can play an important role in the development of cost-effective bioinformatics analysis applications. Existing workflow application processes lack effective computing resource management capabilities, such as the provided cloud computing environment. Insufficient computational resources destroy the execution of workflow applications, wasting time and money. A server cluster in a cloud data center can be composed of servers with differing batches, configurations, performance and energy consumption. In a heterogeneous cloud, the cost control and performance of these different servers can vary, which is related to the trade-off between operational/maintenance cost and quality of service (QoS). The balancing of these two metrics needs to be optimized in heterogeneous cloud data centers. A task scheduling model that responds to the peaks and valleys of task sequencing [1–3] is related to performance indicators such as cost savings, load balancing and system performance (average task wait time, average response time and throughput). Traditional task scheduling methods [4–6] simply attempt to optimize system throughput or system response time or make load balancing the overarching objective while ignoring other factors, such as the server cluster's recognition of task intensity, operation/maintenance management cost control and the dynamic adjustment of large-scale cluster computer points in a static scheduling strategy.

In this paper, a large-scale cost saving and load balancing task scheduling model (HDCBS) is proposed in order to optimize system throughput, and the task scheduling process in a cloud data center is modeled mathematically using queuing theory. Each computing node was modeled as an independent M/G/1/∞ (M/M/1/∞) queuing system for analysis, and the process of the main scheduling server assigning tasks to computing nodes in a cluster was analyzed. Next factors of cost control, average system wait time and average task response time were modeled. A task assignment solution with load balancing mechanism was then proposed using convex optimization theory, and the feasibility and validity of the large-scale task scheduling model were verified by simulation experiments on MultiRECloudSim [7], framework for modeling and simulation of cloud computing infrastructures and services. With the actual task sequence as the input, the performance of the model was validated by comparing it with frequently-used scheduling methods. Our contributions in this paper can be summarized as follows:

- We propose a task scheduling model to deal with the tradeoff between the operation costs and system performance in heterogeneous cloud data centers for bioinformatics workflows running.
- We propose an efficient and novel cost savings and load distribution scheduling model based on treating every execution node as an M/G/1/∞(M/M/1/∞) queuing system for dynamic assignment with variable task arrival rate.

- We use queuing theory to describe and formulate the target function and solve the optimization problem. We also conduct extensive simulation experiment to evaluate and validate our method in the end.

2 Related work

Cloud computing technology can hide technical details and make it easier for users to build such a responsive environment. Hondo et al. conducted a study on bioinformatics workflows running in an IaaS cloud computing environment, used different types of NoSQL database systems to persist provenance data based on the PROV-DM model [8]. Liu et al. proposed a cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analysis, capable of reliable, highly scalable execution and fully automated manner of sequencing analyses workflows [9]. Abouelhoda et al. [10] proposed the new progress in designing scalable and cost-effective cloud workflows based on Tavaxy workflow system and emphasized the application of genome analysis.

For improving effective computing resource management capabilities in the workflow application, Emeakaroha et al. [11] proposed a workflow optimization management method of bioinformatics based on cloud computing. Xie et al. [12] concerned the balance between the cost of storing intermediate data and the computing costs incurred in regenerating this data when large bioinformatics or other workflows are implemented using cloud resources.

Current studies on task scheduling, load balancing of computational nodes and system performance optimization in heterogeneous cloud data centers have attracted the attention of cloud service providers. Many of these studies have achieved productive results. Bai et al. proposed a performance evaluation model for the heterogeneous cloud data center. The model consisted of an M/M/1/∞ queuing system of a main scheduling server and an M/M/C/∞ queuing system of computing nodes. Key technical indicators of the queuing system will be provided later in this study. The effectiveness of the performance evaluation model was verified by experiments [13]. Jin et al. defined a scheduling model based on task sequence perception to minimize the overall delays of data transmission and task execution in geographically distributed data centers. These researchers also proposed an online heuristic algorithm to achieve load balancing and optimize the overall response time of data centers [14]. Chen et al. [15] proposed a cloud load balancing (CLB) mechanism, which uses an additional dynamic balancing method to optimize balanced loading performance when users log in at the same time. Tripathi et al. established a non-cooperative game model between front-end and terminal proxy servers to represent the load balancing problems in distributed data centers. In addition, a Nash balancing algorithm based on distributed load balancing was proposed to minimize the operational costs of data centers [16]. In order to solve the problem of the capacity constraints faced by a single cloud data center during peak periods, Panda et al. proposed task scheduling and four related algorithms (CZSN, CDSN, CDN and CNRSN) for cross-cloud joint load balancing. This proposed solution optimizes the system's task completion time and improves cloud resource utilization [17].

In addition to the performance analysis of servers in cloud data centers, heterogeneous cloud data centers were modeled using the queuing theory to study the factors related to task assignment, energy consumption, load balancing and system throughput of computing nodes, thereby optimizing the relevant performance indicators [18–20]. For the scenario of the irregular arrival of tasks in a service cloud that changes with time, Yuan analyzed the mathematical relationship between the task rejection rate and the service rate provided by service cloud and proposed a multi-queue scheduling (MQS) method based on profit maximization. The method combined simulated annealing, particle swarm optimization and the new meta-heuristic optimization method of a genetic algorithm, in order to optimize profit and throughput by meeting the task response time limit [21].

There are three general disadvantages to the methods proposed by the above studies:

- The proposed task scheduling algorithms did not take into account the impact of timeliness of task sequence intensity on load balancing. The static state of a computing node cluster was taken as the main constraint, which restricted the dynamic adjustment based on service intensity and computing node capacity.
- The objectives were uniform. In the task scheduling process, load balancing was the focus over optimization of operation/maintenance cost control, which resulted in overlooking the effect of system performance.
- Most of the task scheduling models were aimed at the optimization of isomorphic computing nodes, which failed to meet the needs of heterogeneous cloud data centers and could not be widely applied.

Such task scheduling models could not perceive or adapt to the impact of a complicated, dynamically changing task arrival rate on the task assignment rate of computing nodes. Additionally, they could not meet the optimization requirements for operation cost, energy consumption and system performance.

3 A tasks scheduling framework for bioinformatics workflow running

In the process of bioinformatics research, there are a lot of computing and services requirements for storage, genomic and metagenomics sequencing, data statistics, experimental and simulation datasets, and high throughput proteomics. In order to accelerate biological research, it is an effective solution by using the technologies of virtualization and computing service encapsulation to construct a cloud-based computing platform for bioinformatics workflows running.

The cloud computing platform can meet the needs of computational biology, reflected mainly in:

- Storage capacity: The big storage capacity will meet the needs of bioinformatics for large databases, such as genome sequence database, transcriptomics

databases, and heterosis-related gene database, and the size of these databases is more than 100TB or even 10PB.
- Flexible computing models: In computation-as-a-service, it will adjust computational resources, such as computing nodes, storage capacity, network bandwidth, and so on, according to users' requirements.
- Parallel computing and large-scale data analyses: It can schedule large-scale cluster to run heavy duty calculations for high-speed analyses.
- Supporting tools and algorithms: The running environment can support many different tools, programming software, and algorithms for different data analyses and users requirements.

We present a cloud-based tasks scheduling service platform for metabolomics data analyses, traditional Chinese medicine (TCM) candidates research, gene and protein sequences analyses. Some tools, applications and software for computational biology and bioinformatics which can run in the platform are listed in Table 1.

The tasks scheduling framework of the platform is shown as Fig. 1.

As shown in Fig. 1, in the application container layer, all the users' requirements (such as RNA-Seq services, PCA, FCS and K-means, and so on) will be encapsulated and transformed into fine-grained applications. These task sets consist of a large number of applications, services, and computing units which is a large-scale bioinformatics workflow. The scheduling engine with the core controller-HDCBS will assign all the arrival tasks to run in the cloud computing infrastructures for balancing operational/maintenance management cost control and QoS in heterogeneous cloud data centers.

4 Construction of HDCBS

Bioinformatics workflows can be mapped into task sequences for running in a cloud computing environment/ cloud data center. In the cloud datacenter, there is a server to serve as a task filter and adapter. It analyses the user requests (bioinformatics workflows running applications packaged in containers) and transforms them (jobs) into arrival tasks with resources require or abandon it. By the task filter and adapter, the core node server will get the arrival tasks whose arrival time can be considered as a Poisson distribution and then assign them to the computing nods. Each job contains several subtasks and the service time for each subtask [2, 25, 26] will be considered as following an exponential distribution. In this way, the following parameters in the model of HDCBS are made.

- Tasks arrive at the core node server according to a Poisson distribution with parameter λ. That means the tasks arrival rate is λ.
- The service times of computing nodes for tasks is considered as an exponential distribution with parameter μ_i. The service rate of a computing node is μ_i.

Table 1 Some tools, applications and software for computational biology

Service name	Research domain	Function	Task pattern
SBMLDock [22]	Systems biology	Model analyses (Model comparison, Model check)	Docker(SBMLCompare, SBMLChecker)
NanoOK [23]	Genetic engineering	Metagenomics analysis, Multiplexed samples analysis	Docker(LAST, marginAlign, BLASR)
Mash [24]	Systems biology	Biological species identification, species comparison	Docker/application (C++)
RNA-Seq	Genetic engineering	High-throughput sequencing, gene screening	Docker/application (Java)
Random Forest	TCM Formula	TCM filter and activity prediction	Docker/application (Java, Python)
SVM	TCM formula	TCM filter and activity prediction	Docker/application (Java, Python)
MLR	TCM formula	TCM filter and activity prediction	Docker/Application (Java, Python)
PCA	Metabolomics data analysis	Unsupervised learning methods (Data testing, Quality analysis)	Docker/application (R, Python)
K-means Cluster	Metabolomics data analysis	Cluster analysis	Docker/application (Java, Python)
Hierarchical clustering	Metabolomics data analysis	Cluster analysis	Docker/application (Java, Python)
PLS-DA	Metabolomics data analysis	Variation analysis	Docker/application (Java, Python)
Functional class scoring	Metabolomics data analysis	Gene set enrichment analysis	Docker/application (Java, Python)

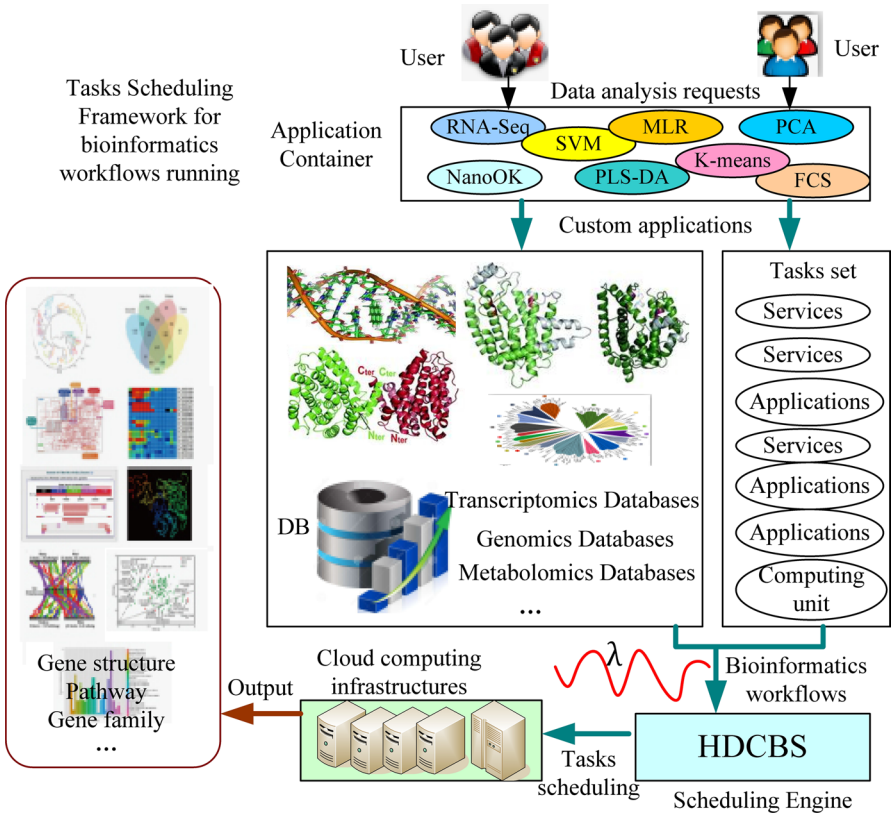

Fig. 1 A tasks scheduling framework for bioinformatics workflows running

Following the different arrival rate of the task sequence, HDCBS will control the core node server to assign tasks to computing nodes with different strategy which will contribute for reconfiguring the datacenter by a cycle window time for saving costs and resources.

4.1 Logical structure modeling

The task scheduling model for heterogeneous cloud data centers consisted of a core node server and m computing nodes to execute delegated tasks. First, the core node server undertook the intensity analysis of current task sequence to obtain the arrival rate λ of system service requests, and on this basis, completed the task assignment/task distribution for all m computing nodes in a balanced manner, that is, delegating all tasks to each computing node for execution following $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$. Secondly, according to the process of task execution and the characteristics of the computing nodes, each computing node was modeled as an independent M/G/1/∞(M/M/1/∞) queuing system using queuing theory, and its service rate was defined as $\mu_i (i = 1, 2, \ldots, m)$ according to CPU computing

capability. This established the logical structure of the task scheduling model for heterogeneous cloud data center, as shown in Fig. 2. The main objective of the task scheduling model was to acquire the optimal average task response time (or average task waiting time) and improve system throughput, while limiting the operational costs of servers in heterogeneous cloud data centers and satisfying the load balancing requirements of the computing nodes.

4.2 Formal description and theoretical model

Cloud data centers provide service and application operation platforms through virtualization technology, the main forms of which are virtual machines (VM) and containers, which appear in the resource pool as independent computing nodes. In addition, tasks [2, 25, 26] in the task sequence were taken as a scheduled small-grained application service, and its granularity was evaluated using the number I of instructions contained in the application set (unit: G). Based on this application scenario, the service rate $\mu_i(i = 1, 2, \ldots, m)$ of each computing node took the main frequency of its CPU as the computing power, i.e., f_i, to calculate the ability of the CPU of computing node N_i to execute instructions (unit: GIPS, or billion instructions per second), which was taken as the reference point.

When studying the basic characteristics of task sequence [1–3, 25, 26], the time of service requests (task sequence) arriving at the cloud data center was assumed to be $1/\lambda$ of independent random exponential distribution, that is, the arrival rate of tasks was Poisson distribution with λ as its parameter. The parameters of the theoretical task scheduling model that integrated cost saving and load balancing are shown in Table 2.

According to queuing theory [27], the time of service requests (tasks) arriving in a cloud data center are independently and identically distributed, and any computing node N_i is an M/G/1/∞(M/M/1/∞) queuing system. Assuming that the service intensity of computing node is $\rho_i = \lambda_i/\mu_i$, $\lambda_i < \mu_i$, and that the service time is general distribution G, the average response time W_{s_i} and the average waiting time W_{q_i} of each computing node are:

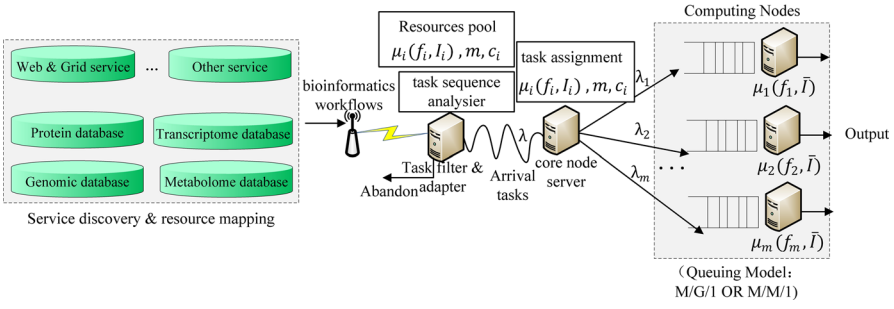

Fig. 2 Logical structure of task scheduling model for heterogeneous cloud data center

Table 2 Meanings of parametric symbols in cost savings and load balancing scheduling model

Symbol	Description and meaning
m	The number of computing nodes in data center, i.e., the number of servers in data center
λ	The arrival rate of service requests, i.e., the number of tasks entering the system within a unit time
$\bar{I}$	The average number of instructions in the instruction set of service application, representing the granularity of tasks
N_i	The mark number of the computing node in the server cluster, representing the ith computing node, where $i \in \{1, 2, \ldots, m\}$
f_i	The number of instructions executed by the N_i CPU core of computing node per second (unit: GIPS)
c_i	The operation cost coefficient of computing node N_i, obtained by normalizing each type of computing node with the computing power level, energy consumption, network, storage and memory of the computing node that has the highest cost in the server cluster as the reference
μ_i	The number of tasks executed by computing node N_i in unit time, i.e., the service rate of computing node N_i, where $\mu_i = 1/(\bar{I}/f_i) = f_i/\bar{I}$
λ_i	Arrival rate of tasks delegated to compute node N_i where $\lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_m$
ρ_i	The service intensity of computing node, $\rho_i = \lambda_i/\mu_i, \lambda_i < \mu_i$
p_i	Probability of core server assigning a task to computing node N_i where $\lambda_i = \lambda p_i$ and $\sum_i^m p_i = 1$
T_n	The time required for the nth task in the task sequence to leave the system and to execute the next task (i.e., the $n+1$th task) in the computing node

$$W_{s\_i} = \frac{1}{\mu_i} + \frac{\lambda_i E(T_n^2)}{2(1-\rho_i)} = \frac{\bar{I}}{f_i} + \frac{\lambda_i E(T_n^2)}{2\left(1 - \frac{\lambda_i \bar{I}}{f_i}\right)}, \tag{1}$$

$$W_{q\_i} = \frac{\lambda_i E(T_n^2)}{2(1-\rho_i)} = \frac{\lambda_i E(T_n^2)}{2\left(1 - \frac{\lambda_i \bar{I}}{f_i}\right)}, \tag{2}$$

According to the above average response time and average waiting time of computing nodes, for the M/G/1/∞ queuing system whose number of computing nodes is m, the service rate is $\mu_i\,(i = 1, 2, \ldots, m)$ and the probability of each task assigned to node N_i is ρ_i, the average response time W_s and the average waiting time W_q of the system are:

$$W_s = \sum_{i=1}^{m} p_i W_{s\_i} = \sum_{i=1}^{m} p_i \left(\frac{\bar{I}}{f_i} + \frac{\lambda_i E(T_n^2)}{2\left(1 - \frac{\lambda_i \bar{I}}{f_i}\right)} \right) = \sum_{i=1}^{m} \frac{\lambda_i}{\lambda} \left(\frac{\bar{I}}{f_i} + \frac{\lambda_i E(T_n^2)}{2\left(1 - \frac{\lambda_i \bar{I}}{f_i}\right)} \right), \tag{3}$$

$$W_q = \sum_{i=1}^{m} p_i W_{q\_i} = \sum_{i=1}^{m} p_i \left(\frac{\lambda_i E(T_n^2)}{2\left(1 - \frac{\lambda_i \bar{I}}{f_i}\right)} \right) = \sum_{i=1}^{m} \frac{\lambda_i}{\lambda} \left(\frac{\lambda_i E(T_n^2)}{2\left(1 - \frac{\lambda_i \bar{I}}{f_i}\right)} \right), \tag{4}$$

Because the operational/maintenance management costs of each type of computing node differ in terms of computing power, energy consumption, network and storage, the cost is a function of the above-mentioned factors. (the function can be obtained by statistical analysis in practical application, but no in-depth analysis is performed here because of space limitations.) In this study, computing power (i.e., service rate μ_i) is the sole influencing factor in cost control. When selecting or starting a computing node, the cost $Ct_i(\lambda_i)$ can be defined as:

$$Ct_i(\lambda_i) = p_i c_i \mu_i = \frac{\lambda_i}{\lambda} c_i \mu_i, \quad i \in \{1, 2, \ldots, m\}, \tag{5}$$

The cost function $Fc(\lambda)$ of the whole system can be defined as:

$$Fc(\lambda) = \sum_{i=1}^{m} Ct_i(\lambda_i) = \sum_{i=1}^{m} \frac{\lambda_i}{\lambda} c_i \mu_i = \sum_{i=1}^{m} \frac{\lambda_i c_i f_i}{\lambda \bar{I}}, \tag{6}$$

4.3 Task scheduling target model

Based on the above analysis and definitions, the task scheduling model for heterogeneous cloud data centers with computing node size m is a scheduling mechanism that can not only realize load rebalancing during the process of task assignment, but can also achieve cost savings and good system performance (average response time or average waiting time of tasks). Accordingly, the target model with parameters $\lambda, f_i, \bar{I}, c_i, m$ can be defined as:

Definition 1 The objective function $\min(Ob_s(\lambda))$ of average task response time with cost saving and system optimization is:

$$\min_{\lambda, f_i, \bar{I}, c_i, m} \left(Ob_s(\lambda) = W_s + Fc(\lambda) = \sum_{i=1}^{m} \frac{\lambda_i}{\lambda} \left(\frac{\bar{I}}{f_i} + \frac{\lambda_i E(T_n^2)}{2\left(1 - \frac{\lambda_i \bar{I}}{f_i}\right)} + \frac{c_i f_i}{\bar{I}} \right) \right), \tag{7}$$

$$\text{s.t.} \quad \sum_{i=0}^{m} \lambda_i = \lambda, \tag{7-1}$$

$$\lambda_i \geq 0, \quad \forall i \in \{1, 2, \ldots m\}, \tag{7-2}$$

$$f_i/\bar{I} - \lambda_i > 0, \quad \forall i \in \{1, 2, \ldots m\}, \tag{7-3}$$

Definition 2 The objective function $\min(Ob_q(\lambda))$ of average task waiting time with cost saving and system optimization is:

$$\min_{\lambda, f_i, \bar{I}, c_i, m} \left(Ob_q(\lambda) = W_q + Fc(\lambda) = \sum_{i=1}^{m} \frac{\lambda_i}{\lambda} \left(\frac{\lambda_i E(T_n^2)}{2\left(1 - \frac{\lambda_i \bar{I}}{f_i}\right)} + \frac{c_i f_i}{\bar{I}} \right) \right), \tag{8}$$

$$\text{s.t.} \quad \sum_{i=0}^{m} \lambda_i = \lambda, \tag{8-1}$$

$$\lambda_i \geq 0, \quad \forall i \in \{1, 2, \ldots m\}, \tag{8-2}$$

$$f_i/\bar{I} - \lambda_i > 0, \quad \forall i \in \{1, 2, \ldots m\}, \tag{8-3}$$

The above two objective functions were attained by solving $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ on the premise of known parameters $\lambda, f_i, \bar{I}, c_i, m$. Thus, the scheme λ_i of the core node server assigning tasks to each computing node was obtained, and a task scheduling method that takes into account cost savings and load balancing was realized.

5 Model solution and verification

5.1 Target model solution

Expanding Eq. (7) yields the sum of n terms of $Ob_s(\lambda)$. Each term corresponds to an objective function of a computing node with task intensity λ_i:

$$\sum_{i=1}^{m} Ob_{s\_i}(\lambda_i) = \sum_{i=1}^{m} (W_{s\_i} + Fc(\lambda_i)), \tag{9}$$

Lemma 1 *Identifying the optimal solution of Eq. (7) in Definition 1 is a convex optimization problem.*

Proof In Definition 1, $\lambda, f_i, \bar{I}, c_i, m$ are known parameters. Therefore, for each item $Ob_{s\_i}(\lambda_i)$ in Eq. (7), the first-order and the second-order derivatives of λ_i can be obtained:

$$Ob'_{s\_i}(\lambda_i) = \frac{d}{d\lambda_i}\frac{\lambda_i}{\lambda}\left(\frac{\bar{I}}{f_i} + \frac{\lambda_i E(T_n^2)}{2(1-\frac{\lambda_i \bar{I}}{f_i})} + \frac{c_i f_i}{\bar{I}}\right) = \frac{\bar{I}}{\lambda f_i} + \frac{\bar{I}E(T_n^2)\lambda_i(2f_i/\bar{I}-\lambda_i)}{2\lambda f_i(f_i/\bar{I}-\lambda_i)^2} + \frac{c_i f_i}{\lambda \bar{I}}, \tag{10}$$

$$Ob''_{s\_i}(\lambda_i) = \frac{d^2(Ob_{si}(\lambda_i))}{d\lambda_i} = \frac{E(T_n^2) f_i/\bar{I}}{\lambda(f_i/\bar{I}-\lambda_i)^3}, \tag{11}$$

From the constraint set of Eq. (7), $\lambda_i \geq 0$, $f_i/\bar{I} - \lambda_i > 0$ and $E(T_n^2) = c > 0$, it was obtained that $\bar{I}E(T_n^2)\lambda_i(2f_i/\bar{I} - \lambda_i) > 0$. Further analysis of Eqs. (9) and (10) showed that objective function $Ob'_{s\_i}(\lambda_i) > 0$ and $Ob''_{s\_i}(\lambda_i) > 0$, that is, the first-order derivative $Ob'_{s\_i}(\lambda_i)$ and the second-order derivative $Ob''_{s\_i}(\lambda_i)$ of $Ob_{s\_i}(\lambda_i)$ are always greater than 0 in the solution interval, which easily proves that $Ob_{s\_i}(\lambda_i)$ is a convex function in the solution interval.

Equation (7) in the objective model is the sum of m terms of $Ob_{s\_i}(\lambda_i)$, and each term of $Ob_{s\_i}(\lambda_i)$ is a convex function in the solution interval. As "the function composed of the sum of finite-term convex functions is still a convex function," [28] Eq. (7) and all its constraints are convex functions in the solution interval. According to convex optimization theory, the optimal solution of Eq. (7) is essentially a convex optimization problem with a real number of linear multiple equality constraints and inequality constraints, and the proof process is omitted. Similarly, the optimal solution of Eq. (8) is also a convex optimization problem, and the proof process is omitted here as well.

Lemma 1 is proved. □

Lemma 2 *Identifying the optimal solution of Eq.* (8) *in Definition* 2 *is a convex optimization problem.*

Proof Similar to the proof of Lemma 1, identifying the optimal solution of Eq. (8) is also a convex optimization problem. □

The task scheduling of a cloud data center is classifiable [2, 25, 26]. It can be classified into I/O, intensive computing, time-consuming and short tasks. Therefore, the program instruction sets contained in tasks can also be classified accordingly, so that the execution of tasks in computing nodes can satisfy the negative exponential distribution with the parameter μ_i. In this case, the computing nodes can be modeled as an M/M/1/∞ queuing system. Thus, in Eqs. (1) and (2), $E(T_n^2) = 2$, and the $\min(Ob_s(\lambda))$ of objective function in Definition 1 can be transformed as:

$$\min_{\lambda, f_i, \bar{I}, c_i, m}\left(Ob_s(\lambda) = W_s + Fc(\lambda) = \sum_{i=1}^{m}\frac{\lambda_i}{\lambda}\left(\frac{1}{f_i/\bar{I}-\lambda_i} + \frac{c_i f_i}{\bar{I}}\right)\right), \tag{12}$$

$$\text{s.t.} \quad \sum_{i=0}^{m} \lambda_i = \lambda, \tag{12-1}$$

$$\lambda_i \geq 0, \quad \forall i \in \{1, 2, \ldots m\}, \tag{12-2}$$

$$f_i/\bar{I} - \lambda_i > 0, \quad \forall i \in \{1, 2, \ldots m\}, \tag{12-3}$$

Due to space limitations, only the $\min(Ob_s(\lambda))$ of objective function in Eq. (11), i.e., the convex optimization problem with real number linear inequality constraints, was solved here. According to convex optimization theory, the problem satisfied Karush-Kuhn-Tucker (KKT) conditions and was solved using the Lagrange multiplier method to obtain the optimal solution.

Assuming the Lagrange multipliers are $\beta, \gamma_i, \delta_i$, the corresponding Lagrange function of Eq. (12) is:

$$\begin{aligned} L(\lambda_i, \beta, \gamma_i, \delta_i) = &\left(\sum_{i=1}^{m} Ob_{s\_i}(\lambda_i)\right) - \beta\left(\sum_{i=1}^{m} \lambda_i - \lambda\right) \\ &- \sum_{i=1}^{m} \gamma_i \lambda_i - \sum_{i=1}^{m} \delta_i \left(\frac{f_i}{\bar{I}} - \lambda_i\right), \end{aligned} \tag{13}$$

According to the KKT conditions and Eqs. (12) and (13), the following conditions can be obtained:

$$\begin{cases} \frac{\triangle}{\triangle \lambda_i} L(\lambda_i, \beta, \gamma_i, \delta_i) = 0 \\ \sum_{i=0}^{m} \lambda_i - \lambda = 0 \\ \lambda_i \geq 0, \quad \forall i \in \{1, 2, \ldots m\} \\ \frac{f_i}{\bar{I}} - \lambda_i > 0, \quad \forall i \in \{1, 2, \ldots m\} \\ \beta(\sum_{i=1}^{m} \lambda_i - \lambda) = 0 \\ \gamma_i \lambda_i = 0, \quad \forall i \in \{1, 2, \ldots m\} \\ \delta_i(\frac{f_i}{\bar{I}} - \lambda_i) = 0, \quad \forall i \in \{1, 2, \ldots m\} \\ \beta, \gamma_i, \delta_i \geq 0, \quad \forall i \in \{1, 2, \ldots m\} \end{cases}, \tag{14}$$

The $\frac{\triangle}{\triangle \lambda_i} L(\lambda_i, \beta, \gamma_i, \delta_i)$ in (14) is:

$$\begin{aligned} \frac{\triangle}{\triangle \lambda_i} L(\lambda_i, \beta, \gamma_i, \delta_i) = &\frac{d}{d\lambda_i} \frac{\lambda_i}{\lambda} \left(\frac{\bar{I}}{f_i} + \frac{\lambda_i E(T_n^2)}{2(1 - \frac{\lambda_i \bar{I}}{f_i})} + \frac{c_i f_i}{\bar{I}}\right) - \frac{d(\beta(\sum_{i=1}^{m} \lambda_i - \lambda))}{d(\lambda_i)} \\ &- \frac{d(\sum_{i=1}^{m} \gamma_i \lambda_i)}{d(\lambda_i)} - \frac{d\left(\sum_{i=1}^{m} \delta_i \left(\frac{f_i}{\bar{I}} - \lambda_i\right)\right)}{d(\lambda_i)}, \end{aligned} \tag{14-1}$$

The equations in (14) are solved using convex optimization theory. Because $\delta_i(\frac{f_i}{\bar{I}} - \lambda_i) = 0$ and $\frac{f_i}{\bar{I}} - \lambda_i > 0$, it can be obtained that $\delta_i = 0$. According to Eq. (13), the corresponding solution expression of $\lambda_i(\beta)$ is:

$$\lambda_i(\beta) = \begin{cases} \frac{f_i}{\bar{I}} - \sqrt{\frac{f_i/\bar{I}}{\beta\lambda - c_i f_i/\bar{I}}}, & \beta \geq \frac{1+c_i(f_i/\bar{I})^2}{\lambda f_i/\bar{I}} \\ 0, & \beta < \frac{1+c_i(f_i/\bar{I})^2}{\lambda f_i/\bar{I}} \end{cases}, \tag{15}$$

According to Eq. (15) and constraint $\sum_{i=0}^{m} \lambda_i - \lambda = 0$, the value of variable β can be obtained, and the task assignment (task arrival rate λ_i)$\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ of each computing node can be obtained. Thus, the optimal solution of Eq. (11) in objective function is obtained.

Next, the monotonicity of Eq. (14) was analyzed, because $\lambda_i'(\beta) > 0$ and $\lambda_i''(\beta) < 0$, $\lambda_i(\beta)$ were strictly monotone increasing functions. As variable $\beta \geq \frac{1+c_i(f_i/\bar{I})^2}{\lambda f_i/\bar{I}}$ and $\sum_{i=0}^{m} \lambda_i < \lambda$, the value of β in interval $[0, +\infty)$ under the condition of $|\sum_{i=0}^{m} \lambda_i - \lambda| < \varepsilon$ was obtained through an approach using binary search. The pseudocode of the corresponding algorithm is:

Algorithm 1 BSearch$(f_i[], c_i[], \bar{I}, \lambda, m)$

Input: $f_i[], c_i[], \bar{I}, \lambda, m$
Output: result $[\beta, \lambda_1, \lambda_2, ..., \lambda_n]$ //the optimal solution
1: $s^{(1)} \leftarrow 0$;
2: $s^{(u)} \leftarrow 1$;//$s^{(u)}$ is the upper bound of β, and $s^{(u)}$ is determined by fast linear interpolation
3: $\triangle \leftarrow 2$;//$\triangle$ Search for the set step size variable, and the initialization of step size is 2
4: **while** $\sum_{i=1}^{m} \lambda_i(s^{(u)}) < \lambda$ **do**
5: $s^{(1)} \leftarrow s^{(u)}$;
6: $s^{(u)} \leftarrow \triangle s^{(u)}$;
7: **end while**
8: **if** $|\sum_{i=1}^{m} \lambda_i(s^{(l)} - \lambda)| \leq \varepsilon$ **then**
9: **return** $result[s^{(l)}, \lambda_1, \lambda_2, ..., \lambda_n]$
10: **end if**
11: **if** $|\sum_{i=1}^{m} \lambda_i(s^{(u)} - \lambda)| \leq \varepsilon$ **then**
12: **return** $result[s^{(u)}, \lambda_1, \lambda_2, ..., \lambda_n]$
13: **end if**//If accuracy does not meet requirements, then $\beta \in (s^{(l)}, s^{(u)})$
14: Binary_Search $(s^{(l)}, s^{(u)})$ //Solve β using dichotomy in interval $[s^{(l)}, s^{(u)}]$
15: **return** $result[s^{(m)}, \lambda_1, \lambda_2, ..., \lambda_n]$;

5.2 Experimental verification

To verify the validity and feasibility of HDCBS for heterogeneous clouds, MatLab14.0 was used to generate a task sequence with arrival time satisfying independent Poisson distribution λ, which was then submitted to the CloudSim framework to process the service requests of the task sequence. The actual experimental data were verified and compared with the solution data of objective model. The experiment used two Dell PowerEdge T720s (dual Xeon 6-core E5-2630 2.3Gb CPUs, for a total of 12 CPU cores). Based on the prices of different servers in the

cloud market, we present a cost coefficient for each computing node according to its CPU, memory, storage and network. The settings in the experiments are shown in Table 3.

The two objectives of the experiment were: (1) to obtain task assignment schemes $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ for each computing node by solving the objective model; (2) to obtain the calculated average response time based on the objective model and the actual response time used to execute the task sequence using MultiRE-CloudSim [7], which were then compared to analyze the credibility of the performance indicator.

Experimental parameters and configuration instructions were as follows:

- Task sequence set: $\text{Task}_k(\ I_k, t_{k\_a}, t_{k\_o})$, $k \in \{1, 2, \ldots, 10^6\}$. Where I_k, t_{k_a}, t_{k_o} denotes the instruction set size of the kth task (the number of instructions, the average number of instructions in the instruction set is 1 Giga, i.e., $\overline{I} = 1$ Giga); t_{k_a} denotes the time when the instruction was assigned to computing node by the core server; t_{k_o} denotes the time when the task was completed and left the system. Accordingly, the system response time of task Task_k is t_{k_o} - t_{k_a}, and the task sequence set size is 10^6. The mean response time of the system $\overline{W_s}$ can be calculated as average$(t_{k_o} - t_{k_a})$
- Computing node cluster: In the experiment, two clusters of nodes (VMs or containers) were configured on the MultiRECloudSim [7] platform: SNT1 and SNT2. Size $m = 5$. Each computing node was equipped with a CPU of different computing power, numbered $N\#i(1 \le i \le 5)$, and its configuration and cost coefficient $c_i(1 \le i \le 5)$ are shown in Table 3.

Validation of the task scheduling model is conducted as follows.

According to experimental parameters and configuration instructions, in order to reduce the possibility of accidental error in the experiment, MultiRECloudSim [7] was used to independently simulate task intensity $\lambda = 5 + 0.5i, i \in \{0, 1, 2, \ldots, 20\}$ on cluster SNT1 and SNT2 for 30 times with the configuration listed in Table 3. The average task response time of each cluster under the task scheduling strategy based on HDCBS for heterogeneous cloud was recorded. A 95% confidence interval (CI) of the average response time was obtained using the sample data obtained in MultiRECloudSim [7]

Table 3 Configurations of VMs in SNT1 and SNT2

Node ID (VM)	Cluster SNT1		Cluster SNT2	
	c_i	f_i(GIPS)	c_i	f_i(GIPS)
N#1	0.1253	2.37	0.1732	3.25
N#2	0.1762	2.61	0.1454	3.42
N#3	0.2278	3.81	0.2102	4.26
N#4	0.2652	4.11	0.2423	4.46
N#5	0.3255	5.15	0.3302	5.84

Table 4 The simulations and analytical results of $\overline{W_s}$

λ	Cluster No.	$\overline{W_s}$ 95% confidence interval (95% C.I)			HDCBS value
		Mean	Lower	Upper	
5.5	SNT1	0.576635945	0.56432199	0.588949901	0.587620328
5.5	SNT2	0.450046853	0.435172625	0.46492108	0.452477279
10.5	SNT1	0.704930066	0.690911117	0.718949015	0.709056916
10.5	SNT2	0.57279	0.559335657	0.586244344	0.564034666
15	SNT1	1.911687857	1.900311326	1.923064388	1.915436249
15	SNT2	0.868892425	0.854508464	0.883276386	0.868347777

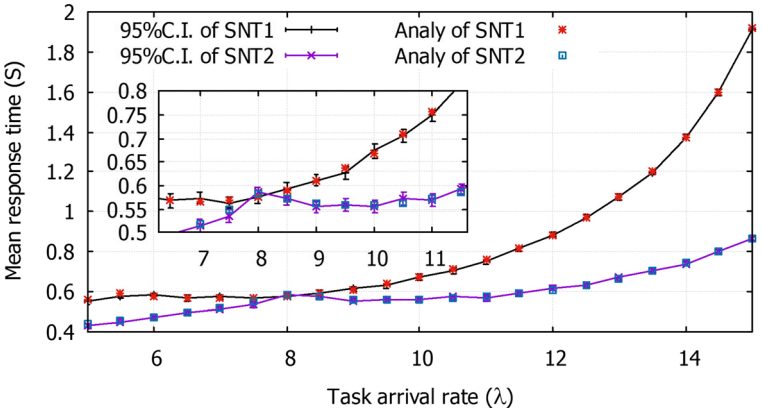

Fig. 3 Comparison of 95% confidence interval between the calculation results of HDCBS and that of mean response time

experiment, which was then compared with the results calculated using the objective model, i.e., the scheduling model to validate the model. The experimental results obtained using the simulation platform and the calculated results $\lambda : \{5.5, 10.5, 15\}$ of HDCBS are shown in Table 4.

Comparison results between the experimental data and HDCBS data at task intensity $\lambda = 5.5 + 0.5i, i \in \{0, 1, 2, \ldots, 20\}$ and 95% confidence interval (95% CI) are shown in Fig. 3. (The sub-graph in Fig. 3 is the magnified effect figure of the corresponding position.)

Table 4 and Fig. 3 show the average response time obtained from 30 independent experiments on MultiRECloudSim [7] with two cluster configurations in Table 4 and the 95% confidence interval of HDCBS calculation results. By analyzing the average task response time comparison table $\overline{W_s}$, i.e., Table 4, and the data comparison analysis chart, it was found that all results obtained using the proposed HDCBS for heterogeneous cloud fall within the 95% confidence interval of the experimental results, which proves that the credibility of the proposed HDCBS is greater than 95%.

6 Efficiency and performance comparison and experimental analysis

6.1 Impacts of c_i and f_i on $\overline{W_s}$

c_i and f_i represent the current state and the processing capacity of the data center server cluster, both of which are known parameters in a fixed computing node cluster. To study the influence of c_i and f_i on the average system response time $\overline{W_s}$, three other clusters (note: the total computing processing capacity is the same, that is, $\Sigma f_i = 18.05$) with different configurations were used in the experiment to respond to different task intensities. The configuration of the corresponding clusters is shown in Table 5.

The total computing processing capacity of all clusters is the same. f_i of cluster SNT3 is the same as that of SNT1, but c_i is different between the two. f_i and c_i of cluster SNT4 are different from those of SNT1. The configuration of all VMs in cluster SNT5 is the same. Using the configurations in Tables 3 and 5, the corresponding computing node clusters were constructed on MultiRECloudSim [7]. Task sequences with computing node response intensity of $0.5 \leq \lambda \leq 18$ were assigned using our proposed algorithm. The average task response time $\overline{W_s}$ of each cluster was determined, and the obtained experimental data were compared and analyzed.

(1) $c_i \neq 0$

The average system response time $\overline{W_s}$ of clusters SNT1 and SNT3-SNT5 when $c_i \neq 0$ and the task response intensity is $0.5 \leq \lambda \leq 18$ is shown in Fig. 4.

$\overline{W_s}$ was further compared and analyzed. With cluster SNT1 as the comparison benchmark, the percentage of $\overline{W_s}$ performance improvement (i.e., $\frac{\overline{W_{s\mathrm{SNT1}}} - \overline{W_{s\mathrm{SNT}i}}}{\overline{W_{s\mathrm{SNT1}}}} \times 100\%$) of SNT2-SNT5 in response to different task intensities is shown in Fig. 5.

According to Figs. 4 and 5, because both the time performance and the total cost factors were taken into account in the objective function, c_i and f_i affected the task allocation by the algorithm to each computing node, thus affecting the $\overline{W_s}$ of the cluster. SNT1 and SNT3 have the same f_i value but different c_i values; as shown in Figs. 4 and 5, when the task response intensity $\lambda \in (0, 6]$, $\overline{W_{s\mathrm{SNT3}}}$ is shorter than $\overline{W_{s\mathrm{SNT1}}}$, the percentage of improvement is between 1.5% and 11.4%, with the average percentage of improvement at 5.4%. When $\lambda = 1.5$, the percentage of improvement is 11.4%. When $\lambda \in (6, 18.05]$, the percentage of

Table 5 Configuration information for the VMs for three clusters

Node ID (VM)	Cluster SNT3		Cluster SNT4		Cluster SNT5	
	c_i	f_i(GIPS)	c_i	f_i(GIPS)	c_i	f_i(GIPS)
N#1	0.1852	2.37	0.136	2.5	0.224	3.61
N#2	0.1364	2.61	0.215	2.9	0.224	3.61
N#3	0.2012	3.81	0.221	3.25	0.224	3.61
N#4	0.2322	4.11	0.255	4.2	0.224	3.61
N#5	0.3125	5.15	0.3215	5.2	0.224	3.61

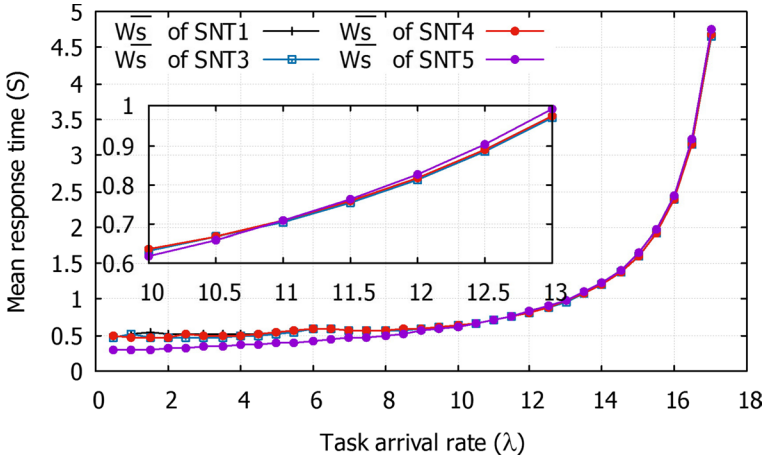

Fig. 4 $\overline{W_s}$ of each cluster when $c_i \neq 0$

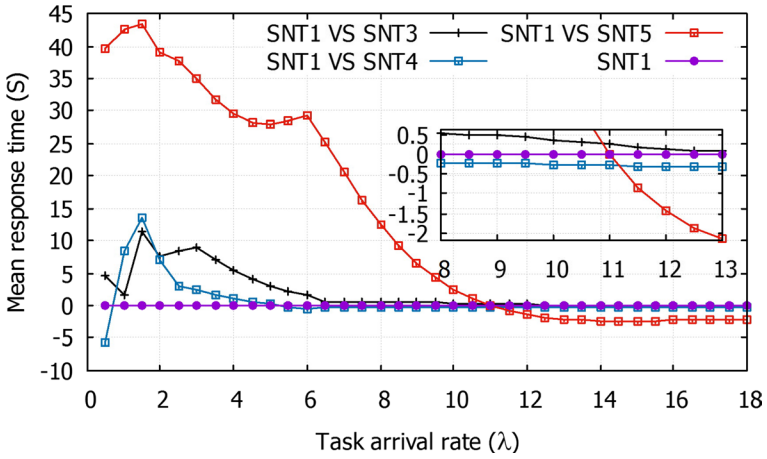

Fig. 5 Improvement percentage of $\overline{W_s}$ of clusters SNT3, SNT4 and SNT5 compared with cluster SNT1

improvement is decreased to less than 0.5%, and as the task intensity increases, the performance improvement becomes minimal. The total computing processing capacity of SNT1 is the same as that of SNT4, but the values of f_i and c_i are different. When $\lambda = 1.5$, $\overline{W}_{s\text{SNT4}}$ is improved by 13.5% compared with $\overline{W}_{s\text{SNT1}}$, but overall $\overline{W}_{s\text{SNT1}}$ is better than $\overline{W}_{s\text{SNT4}}$; especially when $\lambda > 5.5$, $\overline{W}_{s\text{SNT1}} < \overline{W}_{s\text{SNT4}}$. For SNT5 with isomorphic configuration, when $0 < \lambda < 11$, $\overline{W}_{s\text{SNT5}} < \overline{W}_{s\text{SNT1}}$, the average improvement is 24.3%. However, when $11 < \lambda < 18.05$, the processing capacity of SNT1 is greater than that of SNT5, that is $\overline{W}_{s\text{SNT1}} < \overline{W}_{s\text{SNT5}}$.

(2) $c_i = 0$

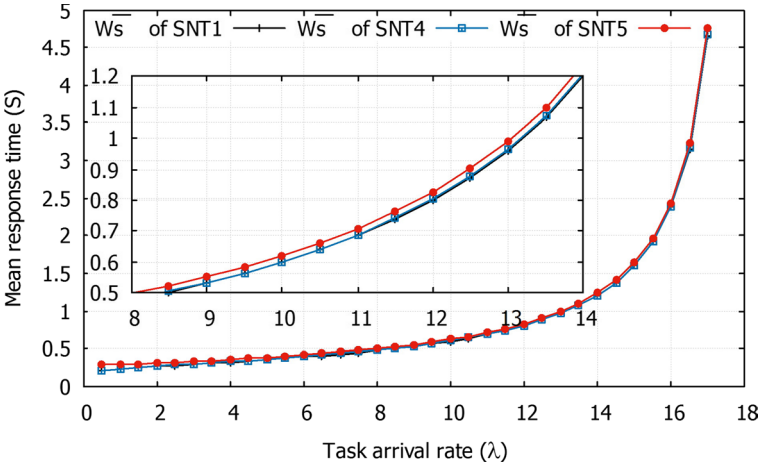

Fig. 6 $\overline{W_s}$ of SNT1, SNT4 and SNT5 when $c_i \neq 0$

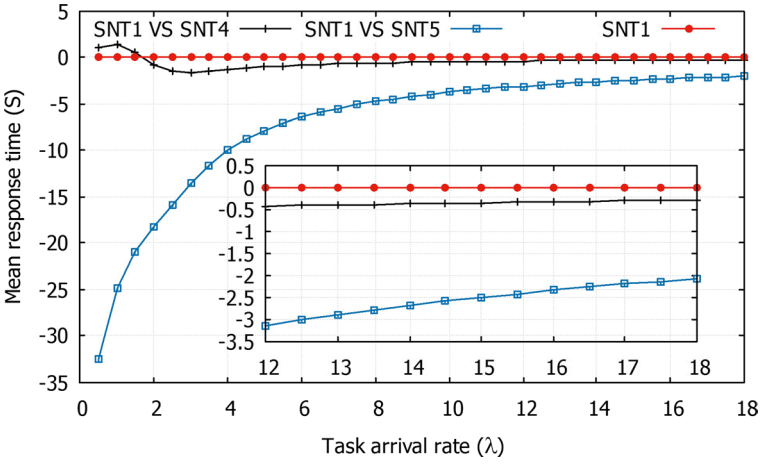

Fig. 7 Improvement percentage of $\overline{W_s}$ of clusters SNT4 and SNT5 compared with cluster SNT1 when $c_i \neq 0$

The performance of the HDCBS model in computing node clusters with different configurations was studied in terms of the performance of the system, i.e., when $c_i = 0$. The corresponding $\overline{W_s}$ curves and performance comparison curves of clusters SNT1, SNT4 and SNT5 obtained in task sequence experiments with intensity of $0.5 < \lambda < 18$ are shown in Figs. 6 and 7.

According to Figs. 6 and 7, when cost is not taken into consideration, for clusters with the same computing power but with different configurations of the nodes, the average system response time $\overline{W_s}$ obtained using the proposed algorithm is different in response to the task sequence with the same

intensity. SNT1 and SNT4 are heterogeneous computing node clusters, the $\overline{W_s}$ of which are similar overall, and the average percentage of improvement average$((\overline{W_{s\text{SNT1}}} - \overline{W_{s\text{SNT4}}})/\overline{W_{s\text{SNT1}}} \times 100\%) \approx -0.5\%$. The performance of isomorphic cluster SNT5 is the lowest among the three; $\overline{W_{s\text{SNT1}}} < \overline{W_{s\text{SNT5}}}$. Overall, $\overline{W_{s\text{SNT1}}}$ is improved by 7.19% in average compared with $\overline{W_{s\text{SNT5}}}$. Particularly in task sequences with low intensity $\lambda \in (0, 4]$, the average improvement percentage of $\overline{W_{s\text{SNT1}}}$ is higher than 10%, with a maximum of 32.4%.

Based on the above analysis of $c_i \neq 0$ and $c_i = 0$, it can be concluded that, based on the proposed HDCBS, the proper configuration should be selected for the computing node cluster to respond to task sequences with different characteristics in order to obtain faster time performance and to lower costs. With both cost and time performance taken into account, the cluster of isomorphic systems should be selected to respond to low-intensity task sequences and that the cluster of heterogeneous systems should be selected to respond to high-intensity task sequences. When only considering the time performance of the system, a reasonable heterogeneous system cluster should be configured to handle task sequences with different intensities in order to increase system performance.

6.2 Performance and efficiency comparison with commonly used algorithms

In order to analyze and evaluate the efficiency and performance of HDCBS in the heterogeneous cloud, HDCBS was compared with two widely used representative algorithms: SRPT [29] and proportional fair scheduling mechanism (PF) [30, 31]. The efficiency and performance of the task scheduling scheme based on objective model and the currently commonly used allocation schemes (allocation by average, allocation by computing power) were comparatively analyzed in terms of cost control.

Using the configuration of cluster SNT1 in Table 2, the results of task assignment under the HDCBS, SRPT and PF mechanisms with task intensity $\lambda \in [0.5, 18)$ were compared in terms of efficiency and performance, i.e., average task response time, cost of node in unit time and target cost that includes cost savings. The results are shown in Figs. 8, 9 and 10. (The sub-graphs are the magnified effect figures of the corresponding positions.)

In an SRPT scheduling strategy, because only the response time of each node was considered while the cost of each node was not taken into account and the shortest response time was prioritized, the tasks were assigned according to the response time of each node. In the PF mechanism, based on the computing power of nodes, tasks were assigned evenly according to the proportion of computing power in the cluster. In the proposed HDCBS, the number of tasks assigned to each computing node was controlled with the goal of comprehensive efficiency and performance, cost savings and rapid response time. By analyzing the corresponding efficiency and performance indicators in Figs. 8, 9 and 10, it was found that SRPT and PF are better than HDCBS in average task response time, but HDCBS is the best in cost of unit time and target cost that combines cost savings.

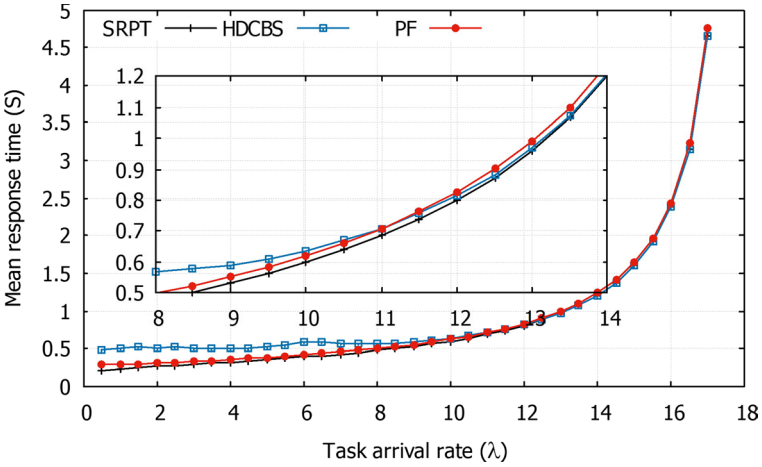

Fig. 8 Comparison of mean response times between the three strategies

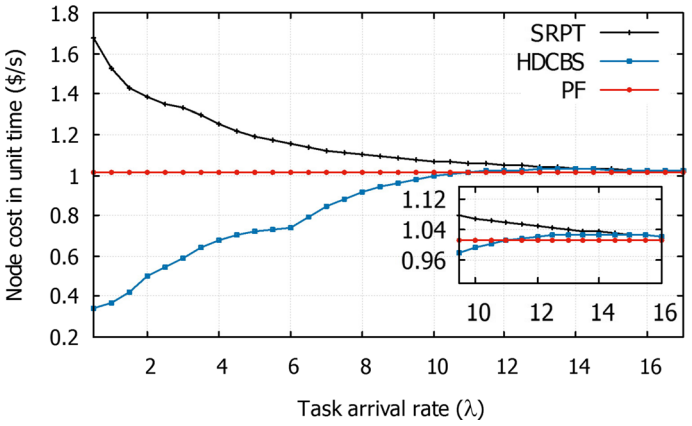

Fig. 9 Comparison of node cost in unit time ($/s) between the three strategies

In particular, the performance of HDCBS in node cost saving is highest when low- and medium-intensity $\lambda \in (0, 7]$, that is, when service intensity of computing node cluster $\rho \leq 40\%$.

Equation (16) is used to calculate the efficiency and performance improvement of the HDCBS than the SRPT and PF:

$$Iv = \frac{\text{Tvalue}_{\text{A}} - \text{Tvalue}_{\text{HDCBS}}}{\text{Tvalue}_{\text{HDCBS}}} * 100\%, \quad \text{A} = \{\text{SRP}, \text{PF}\}, \tag{16}$$

where $\text{Tvalue}_{\text{A}}(\text{A} = \{\text{SRP}, \text{PF}\})$ is the target value of Eq. (7). The target value of Eq. (7) and the improvement percentage values of HDCBS among of the three algorithms are shown in Table 6.

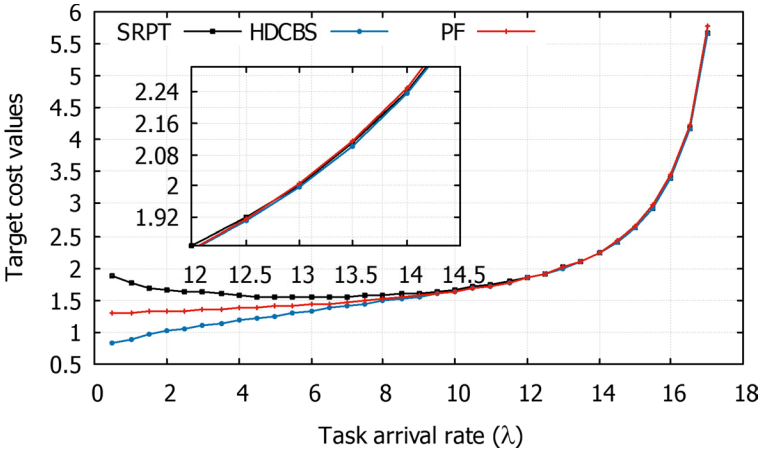

Fig. 10 Comparison of target cost that combining cost savings between the three strategies

Table 6 The target value and the improvement percentage of HDCBS among of the three algorithms

λ		HDCBS	SRPT	PT
0.5	Tvalue	0.814130469	1.891378742	1.300064597
0.5	Iv	0	132.32%	59.69%
3.5	Tvalue	1.14359085	1.60246021	1.358806924
3.5	Iv	0	40.13%	18.82%
6.5	Tvalue	1.371733023	1.544372357	1.448064745
6.5	Iv	0	12.59%	5.565%
14	Tvalue	2.233921289	2.237231259	2.249732214
14	Iv	0	0.1482%	0.7078%

Compared with the SRPT and PF algorithms, HDCBS improves efficiency and performance by more than 10% and 5%, respectively, in processing task sequences within this intensity range, which can be seen from the efficiency and performance improvement percentage comparison chart in Fig. 11.

From the values shown in Table 6 which is the smaller the best it is and the curve in Fig. 11, it can be observed that compared with the performance and efficiency improvement in SRPT and PF, the average savings is $\geq 50\%$ relative to target cost that takes into account cost savings in HDCBS when the cluster service intensity is at medium and low intensity, i.e., $\lambda \in (0, 7]$ and $\sum_{i=1}^{m} f_i = 18.05$. This feature can be used to guide the selection or dynamic assembly of server clusters with corresponding service capabilities $\sum_{i=1}^{m} f_i$ in heterogeneous cloud data centers to respond to the appropriate task intensity, so as to achieve a better performance to efficiency ratio.

The task assignment rate of HDCBS to each computing node in cluster SNT1 under differing task intensities is shown in Fig. 12. When the cluster responded to low task intensity, tasks were assigned to computing nodes according to cost coefficient. Nodes with high cost coefficients were dormant at low task intensity. With

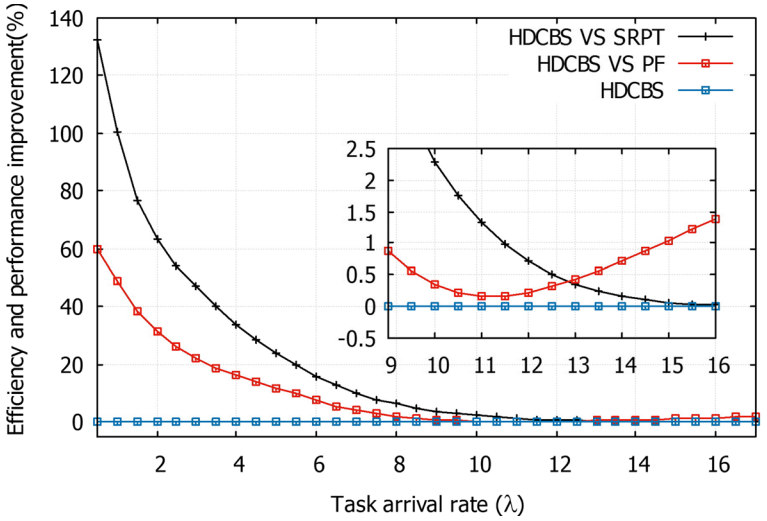

Fig. 11 Efficiency and performance improvement of HDCBS relative to SRPT and PF

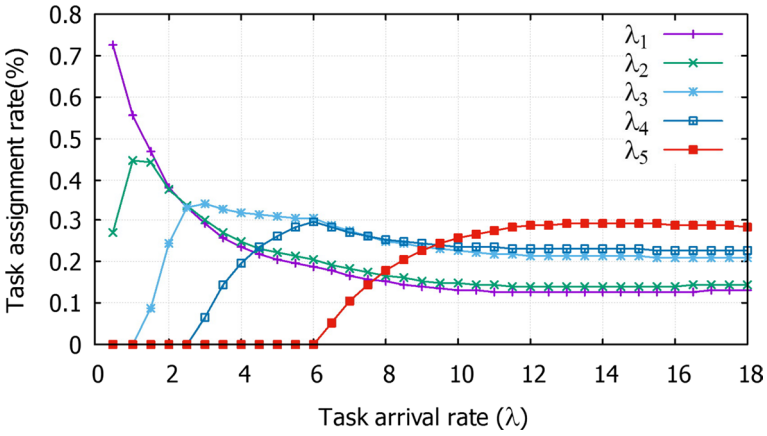

Fig. 12 Task assignment rate of each computing node in HDCBS

the increase in task intensity, the service intensity of nodes with strong service capability was increased (i.e., the amount of assigned tasks gradually increased) to ensure the optimal target cost that takes into account cost savings. As shown in Fig. 13, SRPT mainly assigns tasks to computing nodes with the strongest processing capacity with the shortest task as priority. At this point, computing nodes with low processing capacity are in a dormant state. The operation and maintenance costs of computing nodes with low processing power are usually small, which inevitably leads to SRPT being a higher cost approach than HDCBS.

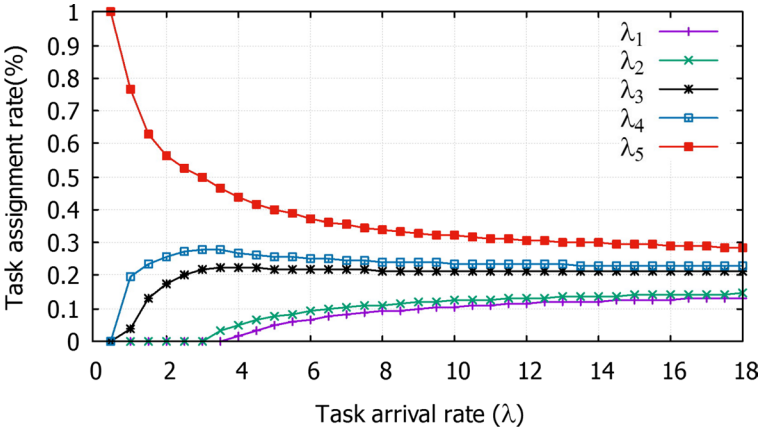

Fig. 13 Task assignment rate of each computing node in SRPT

When the service intensity reached the limit of the cluster's processing capacity (i.e., the task intensity was close to the service capacity of the cluster), the number of assigned tasks was close to the fair task scheduling mechanism (i.e., the task assignment was based on the proportion of the node's service capacity in the cluster).

7 Conclusions

In the development of cost-effective bioinformatics analysis applications, bioinformatics workflows are mapped into task sequences to run in the cloud data center. Task scheduling in large-scale server clusters in heterogeneous cloud data centers is a topic of much research. In this study, to balance operational/maintenance management cost control and QoS in heterogeneous cloud data centers, HDCBS was proposed. The goal was to find the most appropriate task assignment for each node in order to obtain the optimal target cost, which takes into account cost savings by controlling the task assignment of computing nodes, based on the synthetic consideration of the impacts of cluster service capability $\sum_{i=1}^{m} f_i$, cost of each node c_i, task granularity $\bar{I}$ and task sequence intensity λ on cost savings, average task response time and load balancing.

First, each computing node was modeled as an independent M/G/1/∞ (M/M/1/∞) queuing system. On this basis, the mathematical model of HDCBS was constructed and analyzed, and an algorithm for solving the objective model was proposed. Finally, through the simulation experiment of CloudSim, the confidence in the reliability and feasibility of HDCBS was proved to be more than 95%. In addition, a comparative analysis of HDCBS with SRPT and PF showed that its performance and efficiency are the highest among the three.

In this study, a scheduling model that takes into account cost savings and load balancing was proposed for large-scale server clusters in heterogeneous cloud data centers. In practice, our scheduling method can be used general for cloud

computing datacenter and also be applied to fine-granularity applications, such as traffic flow scheduling [32–35] and resource allocation service computing [36, 37]. Future work should focus on how to dynamically build the most suitable server cluster to obtain the optimal cost and highest QoS corresponding to the task sequence intensity. This task is both important and challenging.

Funding This work is supported by National Natural Science Foundation of China (Grant Nos. 61772205, 61872084), Guangdong Science and Technology Department (Grant No. 2017B010126002), Guangzhou Science and Technology Program key projects (Grant Nos. 201802010010, 201807010052, 201902010040 and 201907010001), Nansha Science and Technology Projects (Grant No. 2017GJ001), Guangzhou Development Zone Science and Technology (Grant No. 2018GH17) and the Fundamental Research Funds for the Central Universities, SCUT (Grant No. 2019ZD26).

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

References

1. Lu C, Ye K, Xu G, Xu C-Z, Bai T (2017) Imbalance in the cloud: an analysis on alibaba cluster trace. In: 2017 IEEE International Conference on Big Data (Big Data), IEEE, pp 2884–2892
2. Cheng Y, Chai Z, Anwar A (2018) Characterizing co-located datacenter workloads: an alibaba case study. In: Proceedings of the 9th Asia-Pacific Workshop on Systems, APSys 2018, Jeju Island, Republic of Korea, pp 12:1–12:3
3. Jiang Congfeng, Han Guangjie, Lin Jiangbin, Jia Gangyong, Shi Weisong, Wan Jian (2019) Characteristics of co-allocated online services and batch jobs in internet data centers: a case study from alibaba cloud. IEEE Access 7:22495–22508
4. Kameda H, Li J, Kim C, Zhang Y (2012) Optimal load balancing in distributed computer systems. Springer, New York
5. Domanal SG, Reddy GRM (2014) Optimal load balancing in cloud computing by efficient utilization of virtual machines. In: Sixth International Conference on Communication Systems and Networks, COMSNETS 2014, Bangalore, India, pp 1–4
6. Andrews Jeffrey G, Singh Sarabjot, Ye Qiaoyang, Lin Xingqin, Dhillon Harpreet S (2014) An overview of load balancing in hetnets: old myths and open problems. IEEE Wireless Commun 21(2):18–25
7. Lin Weiwei, Siyao Xu, He Ligang, Li Jin (2017) Multi-resource scheduling and power simulation for cloud computing. Inf Sci 397:168–186
8. Hondo F, Wercelens P, da Silva WMC, Castro K, Santana I, Walter MET, de Araújo APF, Holanda M, Lifschitz S (2017) Data provenance management for bioinformatics workflows using NOSQL database systems in a cloud computing environment. In: 2017 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2017, Kansas City, MO, USA, pp 1929–1934
9. Liu Bo, Madduri Ravi K, Sotomayor Borja, Chard Kyle, Lacinski Lukasz, Dave Utpal J, Li Jianqiang, Liu Chunchen, Foster Ian T (2014) Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses. J Biomed Inf 49:119–133
10. Abouelhoda Mohamed, Issa Shadi, Ghanem Moustafa (2013) Towards scalable and cost-aware bioinformatics workflow execution in the cloud—recent advances to the tavaxy workflow system. Fundam Inf 128(3):255–280
11. Emeakaroha Vincent C, Maurer Michael, Stern Patrick, Labaj Pawel P, Brandic Ivona, Kreil David P (2013) Managing and optimizing bioinformatics workflows for data analysis in clouds. J Grid Comput 11(3):407–428

12. Xie Z, Han L, Baldock RA (2013) Augmented petri net cost model for optimisation of large bioinformatics workflows using cloud. In: Seventh UKSim/AMSS European Modelling Symposium, EMS 2013, Manchester UK, pp 201–205
13. Bai W-H, Xi J-Q, Zhu J-X, Huang S-W (2015) Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model. In: Mathematical Problems in Engineering 2015
14. Jin Y, Gao Y, Qian Z, Zhai M, Peng H, Lu S (2016) Workload-aware scheduling across geo-distributed data centers. In: 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, pp 1455–1462
15. Chen Shang-Liang, Chen Yun-Yao, Kuo Suang-Hong (2017) CLB: a novel load balancing architecture and algorithm for cloud services. Comput Electr Eng 58:154–160
16. Tripathi R, Vignesh S, Tamarapalli V, Chronopoulos AT, Siar H (2017) Non-cooperative power and latency aware load balancing in distributed data centers. J Parallel Distrib Comput 107:76–86
17. Panda Sanjaya K, Jana Prasanta K (2018) Normalization-based task scheduling algorithms for heterogeneous multi-cloud environment. Inf Syst Front 20(2):373–399
18. Cao Junwei, Hwang Kai, Li Keqin, Zomaya Albert Y (2013) Optimal multiserver configuration for profit maximization in cloud computing. IEEE Trans Parallel Distrib Syst 24(6):1087–1096
19. Chiang Y-J, Ouyang Y-C (2014) Profit optimization in SLA-aware cloud services with a finite capacity queuing model. In: Mathematical Problems in Engineering 2014
20. Cao J, Li K, Stojmenovic I (2014) Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers. IEEE Trans Comput 63(1):45–58
21. Yuan H, Bi J, Zhou M (2019) Multi-queue scheduling of heterogeneous tasks with bounded response time in hybrid green IAAS clouds. IEEE Trans Ind Inf 15(10):5404–5412
22. Gnimpieba EZ, Thavappiragasam M, Chango A, Conn B, Lushbough CM (2015) Sbmldock: Docker driven systems biology tool development and usage. In: International Conference on Computational Methods in Systems Biology. Springer, New York, pp 282–285
23. Leggett RM, Heavens D, Caccamo M, Clark MD, Davey RP (2016) Nanook: multi-reference alignment analysis of nanopore sequencing data, quality and error profiles. Bioinformatics 32(1):142–144
24. Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, Phillippy AM (2016) Mash: fast genome and metagenome distance estimation using minhash. Genom Biol 17(1):132
25. Liu Q, Yu Z (2018) The elasticity and plasticity in semi-containerized co-locating cloud workload: a view from alibaba trace. In: Proceedings of the ACM Symposium on Cloud Computing, SoCC 2018, Carlsbad, CA, USA, pp 347–360
26. Alam M, Shakil KA, Sethi S (2016) Analysis and clustering of workload in Google cluster trace based on resource usage. In 2016 IEEE International Conference on Computational Science and Engineering, CSE 2016, and IEEE International Conference on Embedded and Ubiquitous Computing, EUC 2016, and 15th International Symposium on Distributed Computing and Applications for Business Engineering, DCABES 2016, Paris, France, pp 740–747
27. Shortle JF, Thompson JM, Gross D, Harris CM (2018) Fundamentals of queueing theory, vol 399. Wiley, Hoboken
28. Boyd Stephen, Vandenberghe Lieven (2004) Convex optimization. Cambridge University Press, Cambridge
29. Ren Xiaoqi, Ananthanarayanan Ganesh, Wierman Adam, Minlan Yu (2015) Hopper: decentralized speculation-aware cluster scheduling at scale. Comput Commun Rev 45(5):379–392
30. Margolies Robert, Sridharan Ashwin, Aggarwal Vaneet, Jana Rittwik, Shankaranarayanan N K, Vaishampayan Vinay A, Zussman Gil (2016) Exploiting mobility in proportional fair cellular scheduling: measurements and algorithms. IEEE/ACM Trans Netw 24(1):355–367
31. Singh Sarabjot, Geraseminko Mikhail, Yeh Shu-ping, Himayat Nageen, Talwar Shilpa (2016) Proportional fair traffic splitting and aggregation in heterogeneous wireless networks. IEEE Commun Lett 20(5):1010–1013
32. Cai Weihong, Yang Junjie, Yidan Yu, Song Youyi, Zhou Teng, Qin Jing (2020) Pso-elm: a hybrid learning model for short-term traffic flow forecasting. IEEE Access 8:6505–6514
33. Cai L, Yu Y, Zhang S, Song Y, Xiong Z, Zhou T (2020) A sample-rebalanced outlier-rejected k-nearest neighbour regression model for short-term traffic flow forecasting. IEEE Access 1–11
34. Cai Lingru, Lei Mingqin, Zhang Shuangyi, Yidan Yu, Zhou Teng, Qin Jing (2020) A noise-immune lstm network for short-term traffic flow forecasting. Chaos 30(3):1–10

35. Zhou Teng, Jiang Dazhi, Lin Zhizhe, Han Guoqiang, Xuemiao Xu, Qin Jing (2019) Hybrid dual kalman filtering model for short-term traffic flow forecasting. IET Intell Transp Syst 13(6):1023–1032
36. Bai Weihua, Zhu Jiaxian, Zhang Huibing, Lin Weiwei, Xi Jianqing (2019) A multi-dimensional resource scheduling strategy based on multilateral complementarity. IEEE Access 7:88481–88503
37. Lin Miao, Xi Jianqing, Bai Weihua, Jiayin Wu (2019) Ant colony algorithm for multi-objective optimization of container-based microservice scheduling in cloud. IEEE Access 7:83088–83100

Affiliations

Wenwei Cai[1] · Jiaxian Zhu[1] · Weihua Bai[1] · Weiwei Lin[2] · Naqin Zhou[3] · Keqin Li[4]

[1] School of Computer Science, Zhaoqing University, Zhaoqing 526061, China

[2] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

[3] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, Guangdong, China

[4] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA