УДК 004 DOI: 10.26102/2310-6018/2021.34.3.016

Development of an algorithm for resource allocation in distributed systems based on two-criteria process assessment

Y.V. Bondarenko[1,2], A.E. Azeez[1]
[1]*Voronezh State University, Voronezh, Russia*
[2]*Voronezh State Technical University, Voronezh, Russia*

Abstract: One of the most important factors that are affecting of quality of processes service in a distributed system is the way of managing shared resources. At the same time, it is necessary to take into account that each process (task) has a set of characteristics, the complex accounting of which makes it possible to increase the efficiency of executing processes. Among the most important characteristics are the execution time of the process and its significance for solving system-wide tasks. This article is devoted to the development of a resource allocation algorithm based on a two-criteria process assessment. The priority and order of tasks execution is determined based on the importance weights formed by the PROMETHEE II multicriteria decision-making method. The paper describes the features of the application of this method to solve the problem and design an algorithm for the resource's allocation based on a two-criteria assessment of processes. The algorithm provides for the possibility of interrupting the service of processes and forming a queue based on the importance weights. To automate the resource planning process, a software product has been developed that implements the stages of the algorithm. The calculations have shown that the proposed algorithm improves the quality of management of distributed systems, making the resource planning process more flexible and efficient. The approach described in the work is universal and can be extended for the case of an arbitrary number of criteria for evaluating processes.

Keywords: distributed systems, resource allocation, Multi-criteria decision-making (MCDM), PROMETHEE II, virtual machine

For citation: Bondarenko Y.V., Azeez A.E. Development of an algorithm for resource allocation in distributed systems based on two-criterial process assessment. *Modeling, Optimization and Information Technology*. 2021;9(3). Available from: https://moitvivt.ru/ru/journal/pdf?id=1025 DOI: 10.26102/2310-6018/2021.34.3.016

Разработка алгоритма распределения ресурсов в распределенных системах на основе двухкритериальной оценки процессов

Ю.В. Бондаренко[1,2], А.И. Азиз[1]
[1]*Воронежский государственный университет,*
Воронеж, Российская Федерация
[2]*Воронежский государственный технический университет,*
Воронеж, Российская Федерация

Резюме: Одним из важнейших факторов, влияющих на качество обслуживания процессов в распределенной системе, является механизм управления общими ресурсами. При этом необходимо учесть, что каждый процесс (задача) обладает набором характеристик, комплексный учет которых позволяет повысить эффективность обработки процессов. Среди наиболее важных характеристик можно отметить время выполнения процесса и его значимость для решения общесистемных задач. Настоящая статья посвящена разработке алгоритма распределения ресурсов на основе двухкритериальной оценки процессов. Приоритет и порядок выполнения задач определяется на основе весов важности, сформированных методом принятия многокритериальных решений PROMETHEE II. В работе описаны особенности применения данного метода для решения поставленной задачи и сформирован алгоритм распределения ресурсов на основе двухкритериальной оценки процессов. В алгоритме предусмотрена возможность прерывания обслуживания процессов и формирования очереди на основе весов важности. Для автоматизации процесса планирования ресурсов разработан программный продукт, реализующий этапы работы алгоритма. Проведенные расчеты показали, что предложенный алгоритм позволяет повысить качество управления распределенными системами, делая процесс планирования ресурсов более гибким и эффективным. Описанный в работе подход является универсальным и может быть расширен для случая произвольного числа критериев оценки процессов.

Ключевые слова: распределенные системы, распределение ресурсов, многокритериальное принятие решений (MCDM), PROMETHEE II, виртуальная машина

Для цитирования: Бондаренко Ю.В., Азиз А.И. Разработка алгоритма распределения ресурсов в распределенных системах на основе двухкритериальной оценки процессов. *Моделирование, оптимизация и информационные технологии.* 2021;9(3). Доступно по: https://moitvivt.ru/ru/journal/pdf?id=1025 DOI: 10.26102/2310-6018/2021.34.3.016 (на англ.)

Introduction

Recently, the use of high performant computing significantly increased. While the cost of providing such kinds of systems is relatively high, therefore the only solution is to share these resources and use them by multiple processes in parallel, which provides high-performance processing with low cost and is particularly more efficient [1]. Modern distributed systems environment is providing the resource sharing, furthermore according to the architecture of these systems contains multi VMs or nodes each of them has its local CPUs and local memory, which improve the capacity of the whole system.

The most important benefits of using distributed systems are giving the ability for the tasks to access the systems resources, reliability, fault tolerance, scalability, and high-speed performance [2] and this will lead to reducing the cost. In the distributed environment can execute multiple processes by using different available VMs in the system. The clearest examples are Grids and cloud computing, which can provide these services: in infrastructure (IaaS) that represent Infrastructure as a service (PaaS), that refer to the platform as service and software as a service (SaaS). The principle of virtualization clearly used in (IaaS) when the system is sharing the VMs. So, the resource management is controlling the dataflow of jobs and allocate the required resource for each task [3].

The resource allocation (RA) is massively dependent on the strategy that will control system use, especially when a massive number of tasks wait in the Queue to be processed [4]. Therefore, the problem of developing an effective strategy for resource allocation is an urgent task.

A fairly large number of works [4-5] were studied planning of resource allocation. However, the authors of the works propose to use only one criterion as an optimality criterion.

Such a criterion can be the cost of resources or maximization of the quality of service (QoS), the issues of developing effective algorithms for scheduling resources, taking into account the possibility of taking into account several criteria.

In this paper, the proposed algorithm will focus on the optimization of tasks scheduling by using the attribute of these tasks as criteria. The algorithm tries to achieve the effective prediction and the best method in this situation is to make the system automatically decide the task scheduling according to two criteria for choosing the order of service by using the multi-criteria decision-making (MCDM) [6]. In this case, many processes represent the alternatives that the system must choose which of them will serve first and rearrange in the queue to be served by VMs which represent the resources. According to the perspective of (MCDM), this method will give a lot of flexibility to the system.

Materials and Methods

Resource allocation problem (RAP) is the main problem that needs an efficient strategy to be solved in all shared systems. The main goal of scheduling problems is to determine the resources for each process to be executed. Therefore, the efficiency of the scheduling strategy is about minimizing the tasks execution time and achieve the best resource allocation that combines with the QoS. In the distributed system, the resources are shared between tasks (CPU, memory capacity, software, etc.), as showed in Figure 1.

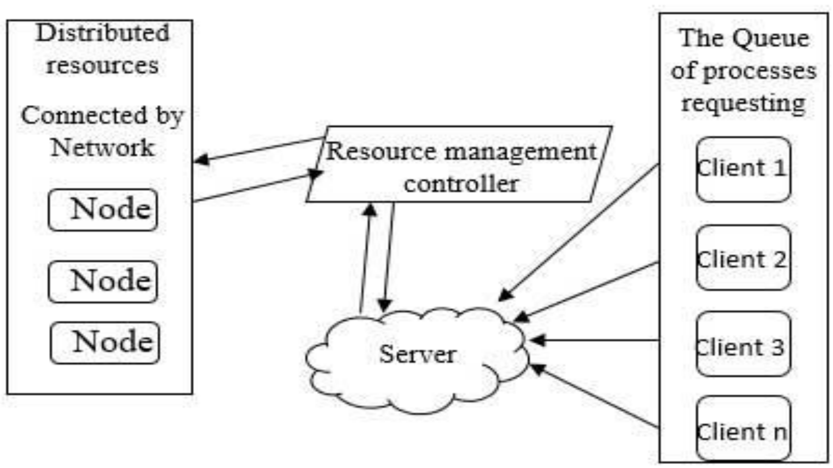

Figure 1 – The main structure for resource allocation in a distributed system
Рисунок 1 – Основная структура распределения ресурсов в распределенной системе

The scheduling problem can simply have described by *n+1* processes or tasks request resources (*m* – number of VMs) from the system, as a result of a limitation of resources and there is no shared clock [2], the tasks scheduling method has to minimize the gap between the request time and the submission time, fault avoidance for example deadlock and considering the task importunacy.

According to the attribute of the distributed system, the proposed algorithm will depend on the MCDM method to allocate resources, the most efficient and flexible ranking method is the Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE II) [7]. This method can solve the decision-making (DM) problem for a finite set of alternatives, by using a comparison pairwise to give the ranking depending on the values of each criterion. Therefore, in the proposed algorithm we used this method to solve the (DM) problem for a massive finite number of tasks that may request the resources of the distributed system.

In general, modelling of resource allocation problem in distributed systems must be providing an environment that has the ability of resource sharing. This ability allows the requesting process to use the hardware and software capacity of the whole system while executing at a high-performance rate [7]. Furthermore, according to the concepts of dataflow in distributed systems, the management of resources will increase the efficiency of the runtime, as well as the task-scheduling mechanism is closed to the optimal resource allocation. Our proposed algorithm will use mainly Multi-criteria decision-making (MCDM) to reach the optimal matrix of alternatives. And this will lead to determining the priorities of each process execution. To model this problem, we assume that a collection of the process (denoted by P) that required resources and waiting in the Queue.

Assuming that we have this set of $n+1$ ($n>0$) number of the process, which is asking for a resource to use in the distributed system that consists of several (VMs). Let P refer to the set of processes:

$$P=\{p_0, p_1, ..., p_n\}.$$

Every VM in the system has its own capacity. On the other hand, the processors have their own attributes. The main idea is to allocate and assign the right resource that satisfied all the processes in the system.

To achieve this goal, the proposed algorithm taking consideration and analyzed all attributes of the process itself.

In this model taking the two of most important characters of the process that affect the system one is the execution time which is denoted by t_i ($i=0,...,n$) and the second will be the priority of the process. Which is considered here as arrival time to the waiting queue or the start requesting time for the resource or even the assignment of importance of processes that could be assigned by user or system itself, in this way give the priority to the job that requests early and this, in turn, leads to decrease the waiting time for all process which improves the level of system performance, will denote as ar_i ($i=0,...,n$).

We assume that we have a distributed system consist of many VMs that contain resources, and an array of the processes that request to be served by the available resources, as well as these resources are limited, must give a priority for one process before another. By using the MCDM, the technique will allocate the resources for the process.

First, when the process p_i is just requesting the resource j in the amount of P_{ij}. In this case we will have a matrix of the process which is representing the alternatives:

$$\tilde{P} = \begin{pmatrix} P_{01} & P_{02} & \dots & P_{0m} \\ P_{11} & P_{12} & \dots & P_{1m} \\ \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & \dots & P_{nm} \end{pmatrix}. \quad (1)$$

In our approach, depending on the criteria of processes itself rather than the VMs capacity and this is while costarring the main condition of resource allocation is achieved:

$$P_{ij} \geq A_{ij}, \; A_j = \sum_{i=0}^{n} A_{ij} = T_j - L_j,$$

Where A_{ij} is the amount of available resource *j* in VMs pool for p_i; T_j is the total of capacity of resource *j* in VMs; L_j is the amount of resource j that is already located and not free in the time of requesting.

The proposed algorithm will apply the MCDM by using PROMETHEE II. and it will mainly in this example use only two criteria for decision-making is the execution time of the process (t_i) and the order of requesting order or time of requesting (ar_i) which we suppose to give these attributes equal priority, so this method is very effective by providing the optimal allocation in a short time. The second stage of the new approach is to check if the system in the save status from deadlock if the exaction of the job will cause a deadlock, then this process will wait in Queue. Then when the system is getting free resources will serve processes that waiting in the Queue.

In the algorithm, we applied the steps of the PROMETHEE II method to the recourse allocation in distributed systems to get a new efficient algorithm. We assumed that there is a request from many processes for VMs as the following (Table 1).

Table 1 – The processes that request resources
Таблица 1 – Процессы, запрашивающие ресурсы

Process request	Execution time (ms.)	Request priority
p_0	t_0	ar_0
p_1	t_1	ar_1
p_2	t_2	ar_2
…	…	…
p_n	t_n	ar_n

First, we will use these two steps of the PROMETHEE II method.

Step1: make all alternatives as a normalized matrix according to direct (Execution time) and indirect criteria (Request priority). The execution time is indirect, normally the system wants to serve the shortest job first, and the process that requests first will be served to improve the QoS of the whole system.

Now we apply the normalization of criteria [8]:

- for direct criteria normalization as:

$$t_i^{norm} = \frac{\max\limits_{0 \le s \le n}(t_s) - t_i}{\max\limits_{0 \le s \le n}(t_s) - \min\limits_{0 \le s \le n}(t_s)};$$

- for the indirect criteria normalization as:

$$ar_i^{norm} = \frac{ar_i - \min\limits_{0 \le s \le n}(ar_s)}{\max\limits_{0 \le s \le n}(ar_s) - \min\limits_{0 \le s \le n}(ar_s)}.$$

Step 2: find the evolution by calculating the difference with respect to all alternatives, then finding the preference function for the process [8].

In the flowing step for each pair (p_i, p_j) finding the d_1 and d_2:

- $d_1(p_i, p_j) = t_i^{norm} - t_j^{norm};$
- $d_2(p_i, p_j) = ar_i^{norm} - ar_j^{norm}, \ \forall i, j = 0,1,...,n.$

For all criteria (direct and indirect) get preference function:

- $Q_1(p_i, p_j) = f_1(d_1(p_i, p_j));$
- $Q_2(p_i, p_j) = f_2(d_2(p_i, p_j)), \forall i, j = 0,1,...,n,$

where $Q_1(p_i, p_j), Q_2(p_i, p_j) \in [0,1]$.

The algorithm applying identification six types of particular preference functions [9]. Average of them:

1) $$f(d) = \begin{cases} 0, if \ d \le 0, \\ 1, if \ d > 0, \end{cases}$$

2) $$f(d) = \begin{cases} 0, if \ d \le b, \\ 1, if \ d > b, \end{cases}$$

3) $$f(d) = \begin{cases} 0, if \ d \le 0, \\ d/b, if \ 0 < d \le b, \\ 1, if \ d > b. \end{cases}$$

The final step is to get the order of process that will be served from the VMs by calculating the outranking of the positive and negative values:

$$\varphi^+(p_i) = \frac{1}{n}\sum_{j=0}^{n} \pi(p_i, p_j),$$

$$\varphi^-(p_i) = \frac{1}{n}\sum_{j=0}^{n} \pi(p_j, p_i),$$

where $\pi\left(p_i, p_j\right) = \alpha \cdot Q_1\left(p_i, p_j\right) + \left(1-\alpha\right) \cdot Q_2\left(p_i, p_j\right)$, α – the coefficient of the importance of direct criteria. The coefficient α $(0 \le \alpha \le 1)$ get by an expert.

Then find the final ranking of processes as:

$$\phi_i = \varphi^+\left(p_i\right) - \varphi^-\left(p_i\right),\ i = 0,1,...,n.$$

According to this decision-making method that depending on values of weight (φ_i) the resource will be allocated and determined which process is suitable for serving in that time to get a higher rate of QoS in distributed systems. To perform this solving, we proposed the following steps of the algorithm of resource allocation based on two-criterial process assessment.

Algorithm of resource allocation based on two-criterial process assessment

Step1:

Set n – a number of tasks;

m – a number of resources;

$\tilde{P}$ – a matrix of requesting resources (1);

$T = \left(T_1,...,T_m\right)$ – a vector of maximum available resources;

$L = \left(L_1,...,L_m\right)$ – a vector of reserved resources (in the initial time of requesting);

$A = \left(A_1,...,A_m\right)$ – a vector of available resources.

Step 2:

Let q_i – an index of process p_i in waiting Queue; $q = 0$ – a length of waiting Queue;

Begin

Step 3:

Apply the MCMD method PROMETHEE II. Get the weight vector for alternatives $\Phi = \left(\varphi_0,...,\varphi_n\right)$.

Step 4:

Rearrange the process in non-increasing ranking that we get from method PROMETHEE II:

$$p_{k_0}, p_{k_1},..., p_{k_n},\ \text{where: } \varphi_{k_0} \ge \varphi_{k_1} \ge ... \ge \varphi_{k_n}.$$

Step5: For $h = 0$ to n

Begin

check p_{k_h} for deadlock free by using the Wait for Graph method or any detection method [10];

if not deadlock then allocate resources for p_{k_h}

else assign p_{k_h} to the waiting Queue, $q := q+1;$, $q_{k_h} := q;$

End For

Step 6: until no new requesting in the system

than End.

Results and discussion

We consider for each process there are two criteria the time of execution and the order of requesting in the queuing system (requesting time). The proposed algorithm will base on these two criteria to find the ranking for each process from the comparing pairwise matrix, which results from the values of these two criteria by using the MCDM technique. So, by using a special program (visual PROMETHEE) to simulate the allocation problem in a numerical example (Figure 2 and Figure 3).

PROMETHEE Flow Table

Rank	action	Phi	Phi+	Phi-
1	P0	1,0000	1,0000	0,0000
2	P2	0,2500	0,6250	0,3750
3	P3	0,0000	0,5000	0,5000
4	P1	-0,2500	0,3750	0,6250
5	P4	-1,0000	0,0000	1,0000

Figure 2 – The decision of ordering the tasks to execute after applying the method PROMETHEE II
Рисунок 2 – Решение о порядке выполнения задач после применения метода PROMETHEE II

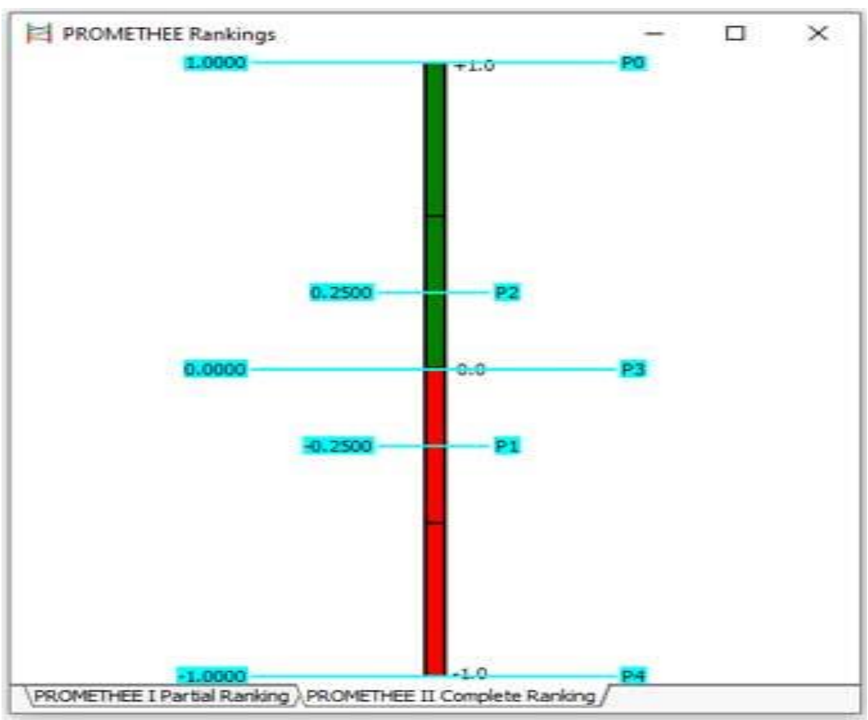

Figure 3 – The general look of the processes ranking after normalized the alternatives matrix and get the final order with weights
Рисунок 3 – Общий вид ранжирования процессов после нормализации матрицы альтернатив и получения окончательного порядка с весами

By applying the numerical example to the program which calculates the processes weight W according to the PROMITHEEII equations we get the weight table (Table 2).

Table 2 – The weight of each process that will determine which processes will be allocated first
Таблица 2 – Вес каждого процесса, определяющий порядок очередности

Processes	Execution time(ms)
p_0	1.00
p_1	-0.25
p_2	0.25
p_3	0.00
p_4	-1.00

According to this weight, the order for processes will be:

$$p_0, p_2, p_3, p_1, p_4.$$

Now by comparing the results of the proposed algorithm that we get from the simulation program with the traditional method like First Come First Serve (FCFS) as in the following example (Table 3).

Table 3 – Processes that request for resources, there are two criteria in consideration for each process
Таблица 3 – Значения характеристик процессов

Processes request	Execution time(ms)	Request priority
p_0	160	500
p_1	300	200
p_2	250	300
p_3	400	400
p_4	700	100

Firstly, we solved the allocation problem by using the (FCFS) method. This method is allocating the resources for the processes depending on the priority of requesting. In this example, the order of processes according to this method will be as the following: p_0, p_1, p_2, p_3, p_4.

The values of the two criteria for this order of processes are shown in Figure 4.

Then the same processes were allocated by using the proposed algorithm that used PROMETHEE II, we note that as a result of using (MCDM) which depends on all criteria in the system the order of processes is different from the first method.

The values of all criteria obtained as a result of applying the proposed algorithm of resource allocation are shown in Figure 5.

As well, for each process, there are two criteria therefore to get the optimal allocation must take into consideration two criteria, so the process algorithm finds the best balance between them, in additional the proposed algorithm reduces the waiting time of each process by trying to execute the short time process first from the queue with consideration of other criteria.

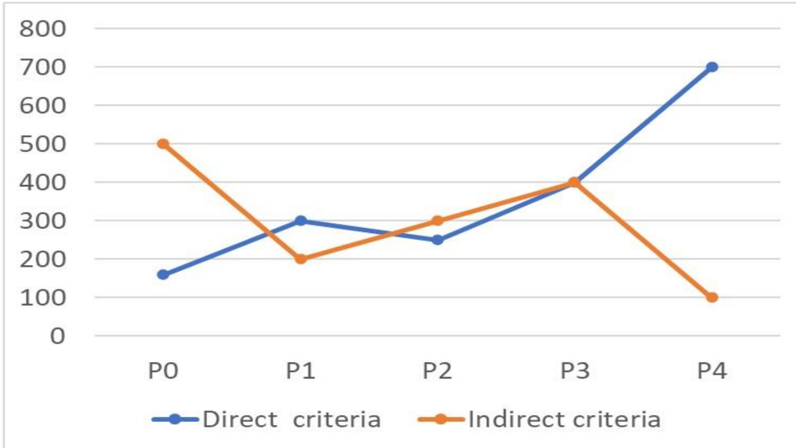

Figure 4 – The result of the FCFS method that is showing the unbalancing between the two criteria in allocation decisions for resources

Рисунок 4 – Результат метода FCFS, демонстрирующий несбалансированность между двумя критериями при принятии решений о выделении ресурсов

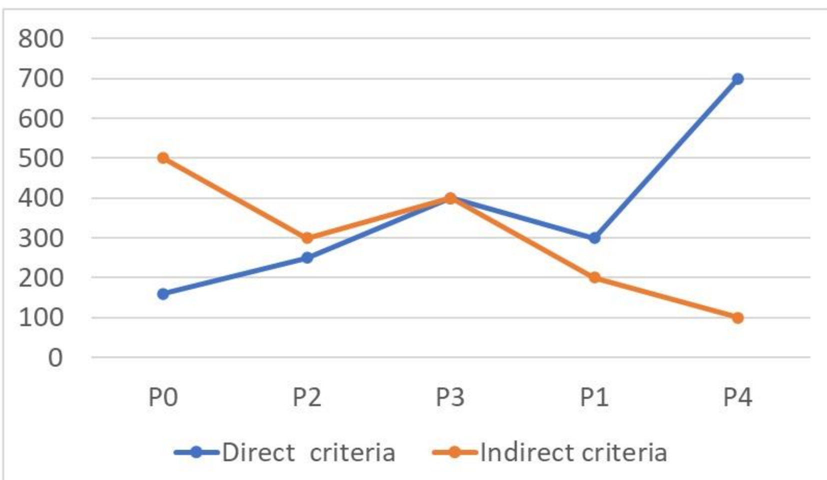

Figure 5 – The result of the proposed algorithm that is showing the balancing between the two criteria in allocation decisions for resources

Рисунок 5 – Результат предложенного алгоритма, демонстрирующий баланс между двумя критериями при принятии решений о выделении ресурсов

So, the result of these tasks is to allocate resources comparing this method and any other method for example (FCFS) will see this method is more efficient topically depend on more than one criterion. Furthermore, finding balancing between importunacy of criteria. And the algorithm is more effective in the case of many processes.

Conclusion

The proposed algorithm finds the system balancing, while the resources are already provided in a distributed system by solving the (RAP). Simultaneously keep the system in higher-performance and (QoS) by using a new strategy, which corresponding to the criteria of tasks and performed the MCDM technique, so the result was efferent resource allocation between tasks and the algorithm give the system flexibility when using PROMETHEE II to change the scenarios of allocation by giving some tasks priority more than other by entering effective criteria that determined manually which is the future work we recommend and extend the system to take criteria of resources and tasks and combined it to achieve most efficient resource allocation.

REFERENCES

1. Thamsen L., Verbitskiy I., Beilharz J., Renner T., Polze A., Kao O. Dynamically Scaling Distributed Dataflows to Meet Runtime Targets. *Proc. Int/ Conf. Cloud Comput Technol Sci CloudCom*. 2017; 37:146-153.
2. Silberschatz A., Galvin P.B., Gagne G. *Operating Systems Concepts*. New York: John_Wiley_&_Sons. 2008.
3. Thamsen L., Renner T., Kao O. Continuously Improving the Resource Utilization of Iterative Parallel Dataflows. *Proceedings of the 6th International Workshop on Big Data and Cloud Performance, ser. DCPerf 2016. IEEE*. 2016;1(4):1-6.
4. Castillo G., Rouskas N., Harfoush K. Efficient QoS resource management for heterogeneous Grids. *22nd. IEEE International Parallel and Distributed Processing Symposium (IPDPS'08), Miami, Florida, US.* 2008:1-15.
5. Jiang H., Ni T. PB-FCFS-a task scheduling algorithm based on FCFS and backfilling strategy for grid computing. *Proceedings of Joint Conferences on Pervasive Computing (JCPC)*. 2009:507- 510.
6. Triantaphyllou E. Multi-Criteria Decision-Making Methods in Multi-Criteria Decision-Making Methods, a Comparative Study. *Applied Optimization*. 2000; 44(1):5-21.
7. Chen L., Xu Z., Wang H., Liu S. An ordered clustering algorithm based on K-means and the PROMETHEE method. *International Journal of Machine Learning and Cybernetics*. 2018; 9(6):917-926.
8. Chakraborty S., Yeh C. H. A Simulation Based Comparative Study of Normalization Procedures in Maldistributed Decision Making. *Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*.2007:102-109.
9. Mareschal B., Brans J.-P. PROMETHEE Methods. *International Series in Operations Research and Management Science*.2014;78(2):163-195.
10. Yu L., Chen L., Cai Z., Shen H., Liang Y., Pan Y. Stochastic Load Balancing for Virtual Resource Management in Datacenters. *IEEE Transactions on Cloud Computing*. 2018; 8(2):459-472.

ЛИТЕРАТУРА

1. Thamsen L., Verbitskiy I., Beilharz J., Renner T., Polze A., Kao O. Dynamically Scaling Distributed Dataflows to Meet Runtime Targets. *Proc. Int/ Conf. Cloud Comput Technol Sci CloudCom*. 2017;37:146-153.
2. Silberschatz A., Galvin P.B., Gagne G. *Operating Systems Concepts*. New York: John_Wiley_&_Sons. 2008.

3. Thamsen L., Renner T., Kao O. Continuously Improving the Resource Utilization of Iterative Parallel Dataflows. *Proceedings of the 6th International Workshop on Big Data and Cloud Performance, ser. DCPerf 2016. IEEE.* 2016;1(4):1-6.
4. Castillo G., Rouskas N., Harfoush K. Efficient QoS resource management for heterogeneous Grids. *22nd. IEEE International Parallel and Distributed Processing Symposium (IPDPS'08), Miami, Florida, US.* 2008:1-15.
5. Jiang H., Ni T. PB-FCFS-a task scheduling algorithm based on FCFS and backfilling strategy for grid computing. *Proceedings of Joint Conferences on Pervasive Computing (JCPC).* 2009:507- 510.
6. Triantaphyllou E. Multi-Criteria Decision-Making Methods in Multi-Criteria Decision-Making Methods, a Comparative Study. *Applied Optimization*. 2000; 44(1):5-21.
7. Chen L., Xu Z., Wang H., Liu S. An ordered clustering algorithm based on K-means and the PROMETHEE method. *International Journal of Machine Learning and Cybernetics.* 2018; 9(6):917-926.
8. Chakraborty S., Yeh C. H. A Simulation Based Comparative Study of Normalization Procedures in Maldistributed Decision Making. *Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*.2007:102-109.
9. Mareschal B., Brans J.-P. PROMETHEE Methods. *International Series in Operations Research and Management Science*.2014;78(2):163-195.
10. Yu L., Chen L., Cai Z., Shen H., Liang Y., Pan Y. Stochastic Load Balancing for Virtual Resource Management in Datacenters. *IEEE Transactions on Cloud Computing*. 2018; 8(2):459-472.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Бондаренко Юлия Валентиновна, доктор технических наук, доцент; профессор Воронежского государственного университета, Воронеж, Российская Федерация; профессор Воронежского государственного технического университета, Воронеж, Российская Федерация.
e-mail: bond.julia@mail.ru
ORCID: 0000-0002-5442-9123

Bondarenko Yulia Valentinovna, Doctor of Technical Sciences, assistant professor; Professor of Voronezh State University, Voronezh, Russian Federation; Professor of Voronezh State Technical University, Voronezh, Russian Federation.

Азиз Аммар Имад, аспирант Воронежского государственного университета, Воронеж, Российская Федерация.
e-mail: ammar.azeez548@gmail.com
ORCID: 0000-0001-7461-8207

Azeez Ammar Emad, postgraduate student of Voronezh State University, Voronezh, Russian Federation.

Статья поступила в редакцию 26.07.2021; одобрена после рецензирования 20.09.2021; принята к публикации 23.09.2021.

The article was submitted 26.07.2021; approved after reviewing 20.09.2021; accepted for publication 23.09.2021.