2024, 21(6): 2156-2166
https://doi.org/10.21123/bsj.2023.8507
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Baghdad Science Journal

Enhancing Cloud Resource Management Based on Intelligent System

Manal Fadhil Younis

Department of Computer, College of Engineering, University of Baghdad, Baghdad, Iraq.

Received 02/02/2023, Revised 10/07/2023, Accepted 12/07/2023, Published Online First 20/11/2023, Published 1/6/2024

Abstract

Cloud computing is a significant technology model with the strongest potential to change the way IT activities are carried out. The cloud offers low-cost and scalable IT resources. Cloud providers strive to improve business outcomes so cloud systems become more complex. The intelligent cloud has a number of issues, including optimizing the cloud service and designing and distributing resources adaptively and economically. There is a rising trend toward adopting intelligent technologies to better cloud management in particular. This paper introduces an intelligent method to manage cloud resources without delay or trouble. Starting with designing a system based on a genetic algorithm (GA) as a computational tool to achieve the goal and plan a predictive request then depending on the round robin (RR) algorithm to allocate the requests for processing. To evaluate the performance of this method, the proposed algorithm was verified in 145 requests. Where the implementation system achieved a reasonable result.

Keywords: Cloud computing, Genetic algorithm, Resource management, Round-robin algorithm, Scheduling.

Introduction

Cloud computing has received great attention due to the rapid development of Internet technology[1]. To reduce the cost by allowing them to use less expensive, more powerful resources, or by enabling them to benefit from the scale economies provided by cloud computing platforms. In a cloud computing system, distributed computing technology and many open service interfaces are utilized to get benefits by selling redundant computing and storage capabilities to users[2], that is, the concept of cloud computing dependent on a pay-per-model that contains a general set of on-demand allocation of programmable computer resources, and users can easily access the network[3]. Service providers can make profits by providing essential services during a short period of time through the advantages of their hardware resources[1], while users and certain companies that want to reduce the cost by allowing to use less expensive, more powerful resources, or by enabling them to benefit from the scale economies provided by cloud computing platforms[4].

However, due to its low-cost and on-demand availability, it is common for cloud asset misuse to arise, minimizing resource utilization or even risking infrastructure. As a result, resource management is critical to ensuring continuous supply and better utilization, which is accomplished through scheduling[5]. The round-robin method is a time-sharing operating system's efficient CPU scheduling process. The system processes the strategies regarding the time slice; even so, the researchers have found calculating the time slice to be extremely difficult. CPU scheduling is a resource allocation technique that deals with assigning CPU time to a particular process. The way the CPU selects the process from the ready queue differs between this algorithm[6]. Several scheduling methods, including FCFS, SJF, priority, and Round-Robin (RR), is used in the time-sharing system. The traditional RR provides an equal amount of time for all processes, which is known as a time slice. Each process uses its time slice numerous times until its execution is complete. Furthermore, the following criteria must be considered when selecting a scheduling algorithm: CPU utilization, throughput, response time, turnaround time, and waiting time. Firstly, resource management must be understood and the numerous issues associated with various scheduling techniques to develop a good scheduling algorithm[6]. This paper offers a thorough resource analysis management in cloud computing using optimization factors appropriate for cloud computing environments. Then will assist researchers in determining the best approach for recommending an appropriate technique for scheduling user applications in a cloud environment.

Cloud computing works as a virtual resource provider for the multi-user with a random selection for whom they need service. Thus to improve stability must intelligent monitoring methods without delay to avoid instability. Then two algorithms GA and RR are combined to ensure that the processing of the tasks is not delayed and reaches a stable state.

The rest of this paper is organized as follows: Cloud resource management is presented, resource management challenges in the cloud are mentioned, a description of genetic and round-robin algorithms, and related works, the proposed cloud resource management algorithm is described, then the results and discussion are presented. Finally, in the last section, the conclusion and future work will be discussed.

Cloud Resource Management

When dealing with predictable or unpredictable requirements, cloud computing is a good choice to take into account[7]. The robust framework of cloud computing enables people as well as companies to acquire the necessary services in accordance with their requirements. The model provides a wide range of services, including storage, deployment platforms, easy access to web services, and other[8]. The management of resources is a major challenge in distributed computing, like cloud computing. Different cloud users require various services based on their changing requirements in cloud computing. As a result, the goal of cloud computing is to present all of the necessary services. Nevertheless, because of the limited resources available, cloud service providers struggle to provide all required services at the same time. Classifications of major types of resources based on their services, like as storage, network, computation, energy, and, security. In the following section, the different types of challenges for resource management will be discussed in the cloud computing situation[9].

Resource Management Challenges in Cloud

Resource mapping, resource provisioning, resource allocation, resource selection and discovery, resource brokering, resource adaptation, and resource scheduling are some of the major challenges based on resource cloud system management systems. The main concept of these challenges is presented[9]:

Resource mapping: The consistency of available resources with a service provider and resources required by users is referred to as resource mapping.

Resource provisioning: this is the method for assigning a service provider's resources to cloud users while maintaining the quality of service guarantee specified in the service level agreement (SLA).

Resource allocation: is the cost-effective assigning of cloud resources among various applications via the internet.

Resource selection and discovery: this is the method of finding all resources offered up in the system, gathering current resource state information, and deciding which target resource to select based on the data collected from the discovery.

Resource brokering: is the method of negotiation for the resources needed with an agent to ensure that they are available in time to accomplish the objectives.

Resource adaptation: this system's ability to dynamically regulate resources to satisfy user needs.

2024, 21(6): 2156-2166
https://doi.org/10.21123/bsj.2023.8507
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Resource scheduling: as a schedule of events and resources that registers when an action should begin or end based on its: dedicated resources, duration, preceding activities, and preceding relationships.

Genetic and Round-Robin Algorithms

Genetic algorithm: GA is a heuristic technique; it solves a difficult optimization problem based on natural evolution, high efficiency, and stability. GA is motivated by the improved fitness of biological systems as a result of evolution. In this algorithm, the solution to a given problem is represented as a chromosome in the form of a string, and a set of elements is represented as genes, which have the values of optimized variables. GA works with a random population of chromosomes. On each chromosome, an objective function is used to assess its fitness. To simulate the natural survival of the fittest process, chromosomes are produced through crossover or mutation[10]. Children have a better casual of surviving if their parents are more physically fit. This cycle repeats until the generation of fittest is identified[11].

Round-robin scheduling: this is the most basic scheduling algorithm and is used in a wide range of applications. When the RR algorithm is implemented for an extended period of time, all devices are scheduled equally[9]. In this situation, each task takes its role all tasks have been assigned time slots[11]. Because round-robin performs on a static time quantum, for execution, each process gets a time slice. When a process completes its time slice, it is moved to the end of the ready queue for later implementation[5]. As a result, it is a time-slicing technique in which every time slot offers the same period of time, and tasks are given one by one[9]. However, many researchers were interested in the constant time quantum problem to minimize the waiting time and turnaround time; while others reorganized the ready queue in increasing or decreasing order, to get better performance. Mainly, an algorithm composite of multiple steps, a genetic algorithm is one of the problem-solving methods that use genetics in its work[10].

Related Works

Many studies have been conducted in order to increase CPU utilization and then improve the performance of cloud management. The following are some related work.

Fan et al.[12] proposed A multi-objective Genetic Algorithm is used for dynamic resource usage and power consumption forecasting. To raise average CPU and memory usage, the prediction VM assignment technique is used, but Genetic Algorithm alone increases execution and waiting time, so a static algorithm can be used such as a round-robin to decrease them.

Kousalya et al.[13] presented a new cloud-based job scheduling algorithm that depended on basic GA merged with PSO for task scheduling. In this method, bandwidth is utilized as a fitness circumstance to validate the scheduling results. Most clouds are taken into account in this strategy to enhance resource utilization. To improve the allocation process, dedicated virtual machines are divided based on their CPU priority. The hybrid algorithm uses cloud measurements such as execution time, bandwidth, node requirement, and deadline to generate an optimized scheduling scheme. The experimental outcomes illustrate that the proposed method maximizes resource utilization. But, this paper combined GA and PSO for task scheduling, which are heuristic methods that take a long time.

Henrique et al.[14] proposed the Genetic Algorithm (GA) and Multi-Population Genetic Algorithm (MPGA), along with the suggested operators, could be a viable different to the earlier PSO approach. According to the findings, GA and MPGA were consistently better than PSO. For the three workflow scheduling algorithms tested, evolutionary algorithms returned improved scheduling schemes that reduced costs in an acceptable amount of time. But there was a delay in the distribution of tasks and it can be solved using a static scheduling algorithm such as round-robin, in order to fulfill the stability

Mahendra et al.[15] proposed a heuristic approach for the allocation of resources and job scheduling be used. Modules such as task scheduling and resource allocation using the divide and conquer method, a modified analytic hierarchy process, longest expected processing time preemption, and bandwidth-aware splitting scheduling are used in this approach. Tasks are processed prior to assignment. He could have used a method, for example, round-robin, to distribute tasks to the VMs and reduce waiting time.

2024, 21(6): 2156-2166
https://doi.org/10.21123/bsj.2023.8507
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Fang[2] determines the resource scheduling execution amount by optimizing the genetic algorithm's execution process, planning the completion time of job scheduling and load balancing, and establishing a set of cloud resource scheduling search schemes based on the genetic algorithm. Experiment results show that the improved genetic algorithm described in this paper has a greater effect on cloud resource scheduling, and achieves more acceptable task scheduling. Firstly, it did not convert the requests to binary code that come from users to the queue which takes up a large storage space and is also difficult to deal with. Secondly, it did not achieve stability in the system, which is important in distributing cloud resources.

Hatem[16] proposed a cloud hybrid adaptive strategy for proper scientific scheduling problems (WS) that are deadline and budget constrained. This problem is solved by combining the heterogeneous earliest finish time HEFT solution into the initial population used by our approach to achieving the best execution time and least execution cost, but the waiting time was not considered into account.
Praveenchandar[17] a Dynamic Resource Allocation method with enhanced job scheduling and improved power management modules is proposed in this paper. This is a new method for energy-efficient scheduling tasks in cloud environments with dynamic resource allocation. The job size and inter-arrival time were considered in the proposed job scheduling method. However, it is possible that several of the tasks are similar in size and arrive at the same time. The limitation of this method is that the delay did not take into account how long the task was waiting in the queue.

Singh et al.[18] suggested the merged the non-dominated sorting and flower pollination using a genetic algorithm (NSGA). The Google cluster dataset is used to evaluate the algorithm. For both static and dynamic situations metrics for performance like utilization of resources, energy use, and emissions of carbon values are computed. Unfortunately, stability was not achieved.

In the following section, a tabular comparison of the literature is presented as illustrated in Table 1. That compares the performance metrics such as delay and stability which are applied to assess the effectiveness of the suggested algorithms.

Table 1. Comparison between the literature and the proposed paper

Ref.	Year	Method	Stability	Delay	Limitation and Solution
12	2017	GA	No	Yes	Increases execution and waiting time, so static algorithms can be used such as round-robin to decrease them.
13	2017	GA merged with PSO	No	Yes	GA with PSO algorithms are used for task scheduling, which are heuristic methods that take a long time, it can merge them with static algorithms.
14	2018	GA	No	Yes	There is a delay in the distribution of tasks and it can be solved using a static scheduling algorithm such as round-robin, in order to fulfill the stability
15	2018	divide and conquer	No	Yes	The waiting time is long, so could use a round-robin, to distribute tasks to the VMs and reduce waiting time.
2	2019	Adaptive genetic algorithm (AGA)	No	Yes	It did not achieve stability in the system, which is important in distributing cloud resources, so can solve this by merging with a static algorithm such as round-robin.
16	2020	Hybrid GA- with HEFT	No	Yes	Best execution time and least execution cost, but the waiting time was not consideration into account.
17	2021	Dynamic Resource Allocation	No	Yes	The delay did not take into account how long the task was waiting in the queue. It can solve by using a static algorithm with it.
18	2023	NSGA	No	Yes	The stability is not achieved so can modify it.
Proposed paper	2023	GA+RR	Yes	NO	This paper solved the delay and stability problems.

2024, 21(6): 2156-2166
https://doi.org/10.21123/bsj.2023.8507
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

The Proposed Cloud Resource Management Algorithm

In this section, a cloud resource management scheme is suggested, which is intended to be an extension of existing cloud frameworks. The process of allocating the various resources depends on intelligent algorithms, which are the hybrid Genetic Algorithm (GA) and Round-Robin (RR) as dedicated in Fig. 1. Using GA would help in estimating the process length to minimize average waiting time and then apply the RR algorithm in order to not enter in a deadlock case.

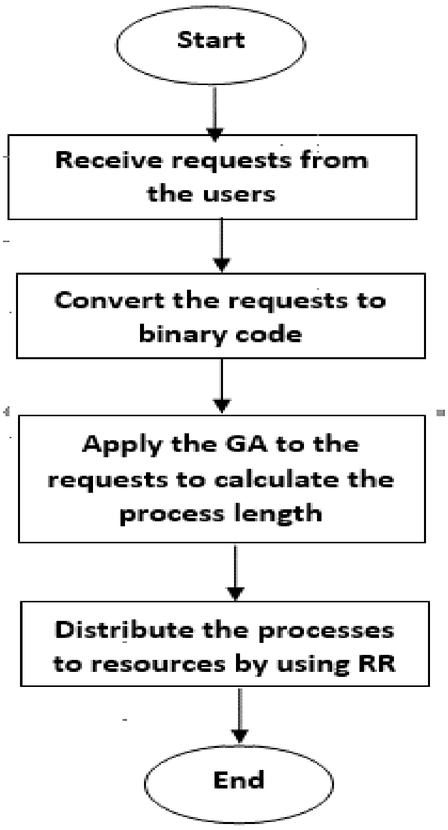

Figure 1. Flowchart of proposed resource management

The relationship between the main algorithm and the two procedures is:

It starts receiving the requests from users by calling the fetch procedure and converting them into binary code to make it easier to deal with them, then calls the genetic procedure, here it starts collecting requests and roughly calculating the length of tasks. After that, go back to the main algorithm, and distribute tasks to resources through the Round-Robin implementation.

The proposed algorithm's steps are as follows:

Main Algorithm
1- Start
2- Call fetch requests from IOT (users).
3- Convert the requests to binary code.
4- Call genetic procedure.
5- Process the requests using the Round-Robin algorithm.
6- Read the current status and save the user request to the info predictive table.
7- End

Algorithm Fetch Request
1- Start
2- While online received requests from users (IoT)
compare with the predictive table and update
save data
3- End

Algorithm Genetic Procedure
1- Strat
2- Start with the initial population for user requests
3- loop- until finish population or fitness
select operation, cross-over, and mutation
4- End

GA is performed by implementing the development process in order to improve toward the best solution. A genetic algorithm has the ability to determine the best job sequence to assign to the processor. The following steps are used by GA to perform its general operations:

- Choose the population set's fixed-size chromosomes that are represented by binary codes.
- Perform any encoding procedure on the chromosome sets'.
- Using their fitness value which is maxing job time=10, choose the best two chromosomes from the set.
- Cross two chromosomes and you will get two different offspring.
- On those offspring, simply swap the bit positions to perform the mutation operation.
- Continue until the population's optimal solution is obtained.
- Finally performing the elitism operation of the chromosomes means storing the best chromosomes in the system.

2024, 21(6): 2156-2166
https://doi.org/10.21123/bsj.2023.8507
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

As shown in Table 1 pre_value represents the process length for each job. After the GA is applied for the IP addresses then the action is checked it is either equal to "P" which is processed or is equal to "N" which is mean normal. Then check if the process length is greater than 10 ms; which must be less than 10, a RR algorithm is applied in order to avoid the process entering a deadlock case that is due to suspend the system.

Results and Discussion

This section describes the experimental findings of the suggested hybrid GA-RR algorithm, beginning with a depiction of the experimental environment. The users send their requests in the form of IoT data. Each data is converted to binary numbers which have the properties of file size. Firstly, the user requests which are fetched as IP addresses convert to 32-bit binary digits in order to represent the chromosomes that the genetic algorithm depended on it. As shown in Table 2.

Table 2. Predictive and real values differences

No.	IP address	Convert IP address to binary	Pre_value	Real_value	Action
1	10:42:231:192	00001010001010101110011111000000	30	26	P
2	10:151:47:92	00001010100101110010111101011100	28	24	P
3	0:133:118:227	00001010100001010111011011100011	33	23	P
4	10:164:140:212	00001010101001001000110011010100	29	19	P
5	10:57:223:218	00001010001110011101111111011010	12	6	N
6	10:42:136:225	00001010001010101000100011100001	2	1	N
7	10:97:167:185	00001010011000011010011110111001	16	7	N
8	10:59:108:151	00001010001110110110110010010111	32	28	P
9	10:80:171:54	00001010010100001010101100110110	18	16	P
10	10:51:182:222	00001010001100111011011011011110	15	11	P
11	10:179:174:152	00001010101100111010111010011000	24	21	P
12	10:112:139:205	00001010011100001000101111001101	18	12	P
13	10:151:89:153	00001010100101110101100110011001	28	25	P
14	10:147:198:82	00001010100100111100011001010010	30	28	P
15	10:49:97:183	00001010001100010110000110110111	1	1	N
16	10:102:233:124	00001010011001101110100101111100	8	6	N
17	10:119:218:168	00001010011101111101101010101000	20	19	P
18	10:141:223:205	00001010100011011101111111001101	15	9	N
19	10:115:230:130	00001010011100111110011010000010	30	22	P
20	10:99:190:52	00001010011000111011111000110100	22	19	P
21	10:94:209:194	00001010010111101101000111000010	2	10	N
22	10:32:112:138	00001010001000000111000010001010	24	21	P
23	10:40:80:87	00001010001010000101000001010111	14	6	N
24	10:144:210:43	00001010100100001101001000101011	5	4	N
25	10:157:64:83	00001010100111010100000001010011	22	21	P
26	10:120:211:188	00001010011110001101001110111100	15	7	N
27	10:51:140:55	00001010001100111000110000110111	22	21	P
28	10:86:41:98	00001010010101100010100101100010	26	16	P
29	10:157:35:88	00001010100111010010001101011000	26	21	P
30	10:146:80:154	00001010100100100101000010011010	25	16	P
31	10:150:169:45	00001010100101101010100100101101	6	1	N
32	10:156:41:69	00001010100111000010100101000101	31	22	P
33	10:125:216:51	00001010011111011101100000110011	4	2	N
34	10:140:159:77	00001010100011001001111101001101	9	6	N
35	10:177:102:139	00001010101100010110011010001011	21	11	P
36	10:153:199:197	00001010100110011100011111000101	10	1	N
37	10:141:208:76	00001010100011011101000001001100	21	17	P
38	10:181:194:210	00001010101101011100001011010010	18	17	P

2024, 21(6): 2156-2166
https://doi.org/10.21123/bsj.2023.8507
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

39	10:141:99:96	00001010100011010110001101100000	4	6	N
40	10:48:219:189	00001010001100001101101110111101	27	22	P
41	10:164:209:165	00001010101001001101000110100101	6	4	N
42	10:79:147:60	00001010010011111001001100111100	10	5	N
43	10:113:212:63	00001010011100011101010000111111	12	9	N
44	10:51:131:125	00001010001100111000001101111101	33	26	P
45	10:145:209:199	00001010100100011101000111000111	15	11	P
46	10:86:190:201	00001010010101101011111011001001	10	7	N
47	10:43:163:229	00001010001010111010001111100101	23	19	P
48	10:41:78:43	00001010001010010100111000101011	23	20	P
49	10:113:137:161	00001010011100011000100110100001	28	27	P
50	10:80:216:124	00001010010100001101100001111100	30	23	P
51	10:110:150:123	00001010011011101001011001111011	3	5	N
52	10:38:182:171	00001010001001101011011010101011	2	2	N
53	10:128:128:187	00001010100000001000000010111011	10	7	N
54	10:121:105:57	00001010011110010110100100111001	4	1	N
55	10:74:146:80	00001010010010101001001001010000	21	13	P
56	10:83:203:162	00001010010100111100101110100010	20	10	N
57	10:81:89:182	00001010010100010101100110110110	24	20	P
58	10:139:163:213	00001010100010111010001111010101	24	22	P
59	10:125:165:207	00001010011111011010010111001111	2	5	N
60	10:105:193:70	00001010011010011100000101000110	21	18	P
61	10:152:154:81	00001010100110001001101001010001	33	28	P
62	10:85:125:73	00001010010101010111110101001001	5	5	N
63	10:145:35:237	00001010100100010010001111101101	14	9	N
64	10:95:87:188	00001010010111110101011110111100	10	10	N
65	10:120:82:96	00001010011110000101001001100000	26	22	P
66	10:79:151:196	00001010010011111001011111000100	24	20	P
67	10:61:222:131	00001010001111011101111010000011	29	19	P
68	10:92:44:116	00001010010111000010110001110100	25	15	P
69	10:46:155:203	00001010001011101001101111001011	27	21	P
70	10:181:121:210	00001010101101010111100111010010	18	17	P
71	10:47:99:170	00001010001011110110001110101010	32	23	P
72	10:48:93:189	00001010001100000101110110111101	27	21	P
73	10:69:210:92	00001010010001011101001001011100	13	10	N
74	10:80:220:60	00001010010100001101110000111100	9	3	N
75	10:58:159:190	00001010001110101001111110111110	19	14	P
76	10:141:177:44	00001010100011011011000100101100	13	6	N
77	10:179:128:191	00001010101100111000000010111111	23	19	P
78	10:176:36:120	00001010101100000010010001111000	25	21	P
79	10:77:82:221	00001010010011010101001011011101	31	27	P
80	10:60:177:215	00001010001111001011000111010111	17	22	P
81	10:37:162:60	00001010001001011010001000111100	29	24	P
82	10:99:168:149	00001010011000111010100010010101	3	2	N
83	10:34:175:221	00001010001000101010111111011101	4	3	N
84	10:57:134:196	00001010001110011000011011000100	11	3	N
85	10:53:154:139	00001010001101011001101010001011	25	17	P
86	10:140:204:83	00001010100011001100110001010011	12	4	N
87	10:148:172:179	00001010100101001010110010110011	23	22	P
88	10:102:155:188	00001010011001101001101110111100	28	24	P
89	10:115:208:56	00001010011100111101000000111000	16	12	P
90	10:100:37:160	00001010011001000010010110100000	9	8	N
91	10:181:113:81	00001010101101010111000101010001	29	23	P
92	10:106:67:71	00001010011010100100001101000111	20	16	P
93	10:97:155:208	00001010011000011001101111010000	5	9	N
94	10:86:199:73	00001010010101101100011101001001	4	3	N
95	10:50:36:111	00001010001100100010010001101111	6	1	N
96	10:65:120:148	00001010010000010111100010010100	18	13	P

97	10:34:42:130	00001010000100010001010101010000010	26	17	P
98	10:159:131:50	00001010100111111000001100110010	3	3	N
99	10:46:128:165	00001010001011101000000010100101	29	24	P
100	10:172:224:223	00001010101011001110000011011111	18	14	P
101	10:120:180:102	00001010011110001011010001100110	29	20	P
102	10:111:155:170	00001010011011111001101110101010	25	18	P
103	10:111:59:203	00001010011011110011101111001011	32	28	P
104	10:168:179:149	00001010101010001011001110010101	34	32	P
105	10:134:170:125	00001010100001101010101001111101	13	5	N
106	10:160:54:114	00001010101000000011011001110010	23	15	P
107	10:43:66:171	00001010001010110100001010101011	5	3	N
108	10:71:157:151	00001010010001111001110110010111	6	5	N
109	0:164:234:225	00001010101001001110101011100001	23	14	P
110	10:75:175:186	00001010010010111010111110111010	26	23	P
111	10:32:148:214	00001010001000001001010011010110	3	9	N
112	10:180:210:175	00001010101101001101001010101111	25	23	P
113	10:102:122:146	00001010011001100111101010010010	9	8	N
114	10:179:148:203	00001010101100111001010011001011	17	14	P
115	10:59:82:55	00001010001110110101001000110111	4	2	N
116	10:137:115:50	00001010100010010111001100110010	24	14	P
117	10:38:60:119	00001010001001100011110001110111	11	9	N
118	10:172:173:125	00001010101011001010110101111101	3	7	N
119	10:85:214:148	00001010010101011101011010010100	21	13	P
120	10:134:71:177	00001010100001100100011110110001	5	4	N
121	10:163:198:91	00001010101000111100011001011011	15	5	N
122	10:181:219:69	00001010101101011101101101000101	1	5	N
123	10:57:221:62	00001010001110011101110100111110	14	4	N
124	10:37:49:179	00001010001001010011000110110011	12	4	N
125	10:53:222:149	00001010001101011101111010010101	33	27	P
126	10:137:194:88	00001010100010011100001001011000	25	16	P
127	10:46:104:106	00001010001011100110100001101010	13	7	N
128	10:146:155:56	00001010100100101001101100111000	10	0	N
129	10:131:97:112	00001010100000110110000101110000	11	8	N
130	10:90:166:158	00001010010110101010011010011110	30	29	P
131	10:33:37:146	00001010001000010010010110010010	14	5	N
132	10:55:141:66	00001010001101111000110101000010	9	8	N
133	10:97:223:72	00001010011000011101111101001000	27	21	P
134	10:42:137:79	00001010001010101000100101001111	17	15	P
135	10:98:168:238	00001010011000101010100011101110	31	21	P
136	10:60:108:204	00001010001111000110110011001100	12	7	N
137	10:80:225:54	00001010010100001110000100110110	19	9	N
138	10:91:36:222	00001010010110110010010011011110	15	12	P
139	10:149:134:165	00001010100101011000011010100101	12	3	N
140	10:121:151:218	00001010011110011001011111011010	6	5	N
141	10:120:49:166	00001010011110000011000110100110	15	7	N
142	10:117:104:188	00001010011101010110100010111100	24	20	P
143	10:129:204:143	00001010100000011100110010001111	9	3	N
144	10:71:192:230	00001010010001111110000000111001 10	2	3	N
145	10:49:177:75	00001010001100011011000101001011	14	11	P

The graphic shown in Fig. 2 represents the values that were predicted using the genetic algorithm by implementing the following steps:

- Generation of population steps: The genetic algorithm operates on a fixed bit string representation of each individual solution. As a result, all of the solutions in the solution space are encoded as binary strings. This step is done in 145 chromosomes (IP addresses).
- Crossover: A crossover is a simple mechanism that exchanges one part of a chromosome for another. The two-point crossover method is

2024, 21(6): 2156-2166
https://doi.org/10.21123/bsj.2023.8507
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

used in this paper to transfer parental genetic material to children. The goal of this step is to select the best-fitting pair of individuals for crossover most of the time. The fitness value of each chromosome is 10 process length. This pool of chromosomes goes through a random single-point crossover, where the portion on one side of the crossover site is exchanged with the portion on the other side, depending on the crossover point. As a result, it creates a new pair of people.

- Mutation: The mutation operation broadens the search space by modifying a single chromosome. The quality of the created chromosomes is not especially good at the start of the genetic algorithm. As time passes on, the quality of the chromosome improves. In this paper, a very low value (0.05) is chosen as the mutation probability. This startup operation assists the GA scheduling purpose in avoiding local optimums and discovering various search locations.

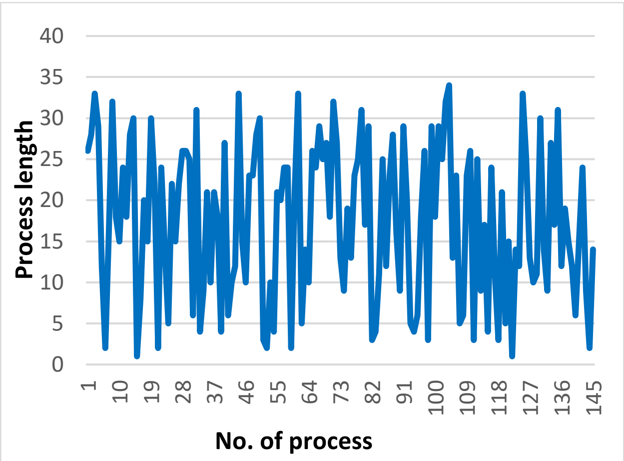

Figure 2. Process predictive values

As illustrated in Fig. 2 The process lengths for the 145 processes range between 1 and 34 process lengths, representing predictive values as a result of GA.

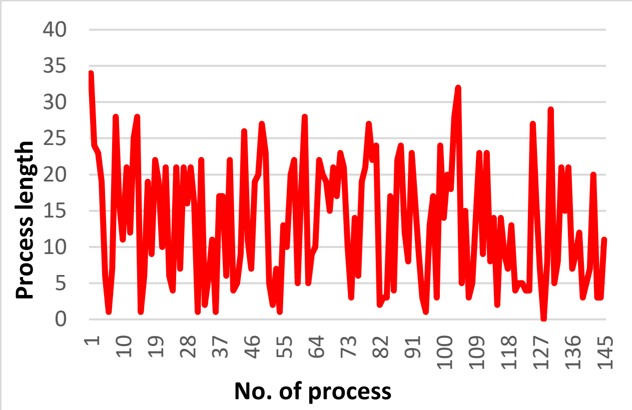

Figure 3. Process real values

As depicted in Fig. 3 The real values range from 1 to 32 for 145 processes. The final result is concluded that there is little difference between predictive and real values. This proves the importance of using the genetic algorithm to find approximate values for user requests. Where almost every time a user request is this long. This is shown in Fig. 4, it is observed from this experiment that the differences between the process lengths of predictive and real values range between 1 and 5 process length is 92 processes, the process whose lengths between 6 and 8 are 28 processes, as well as, the process whose lengths extends from 9 to 10 is 24 processes. This means that the differences between the expected values and the real values are very small, this benefit is due to the use of the GA.

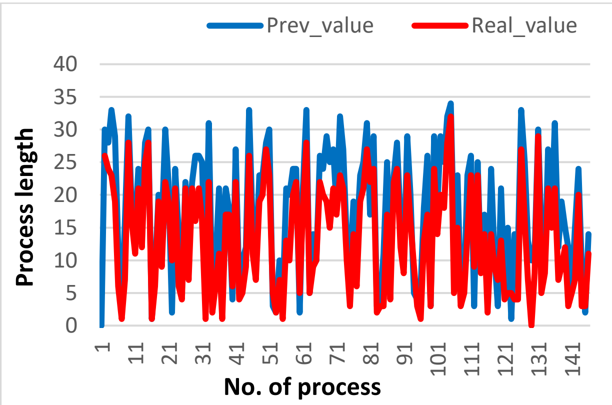

Figure 4. Differences between the process lengths of predictive and real values

In order to avoid a deadlock, which leads to system instability, a round-robin algorithm is used, which uses a time slice equal to 10 for each process. If the process length is less than or equal to 10 then the action is normal, else, if the process length is greater than 10. If the action is greater than 10 then the round-robin algorithm is applied, because if the process length reaches 10 the system enters a halt case, in order to reach to stability system.

Conclusion

There should always be a balance between not delaying tasks for processing and system stability. This balance is a problem for many researchers. In this paper, an intelligent technique for resource monitoring is suggested in a cloud computing environment depending on hybrid genetic and round-robin algorithms. The monitoring problem is important in cloud resources management. In this method, process length is utilized as a fitness condition for proving the fitness of the monitoring results to get the exact values for requests in order to decrease the delay that may be happened. The results from GA compared with the real process lengths and little differences are shown. Finally, a round-robin was applied for the requests whose process length is greater than 10 in order to avoid the deadlock state to obtain a stable system. For future work, the suggested hybrid algorithm can be replaced by another, and more parameters can be added based on the users' requirements.

Acknowledgment

The author would like to express her gratitude to everyone who contributed to the publication of this research paper.

Authors' Declaration

- Conflicts of Interest: None.
- I hereby confirm that all the Figures and Tables in the manuscript are mine. Furthermore, any Figures and images, that are not mine, have been included with the necessary permission for re-publication, which is attached to the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at University of Baghdad.

References

1. Fazlina MA, Latip R, Abdullah A, Ibrahim H, Alrshah MA. Crucial File Selection Strategy (CFSS) for Enhanced Download Response Time in Cloud Replication Environments. Baghdad Sci J. 2021 Dec 20; 18(4 Suppl.): 1356-1364. https://dx.doi.org/10.21123/bsj.2021.18.4 .
2. Yiqiu F, Xia X, Junwei G. Cloud computing task scheduling algorithm based on improved genetic algorithm. ITNEC. 2019 Mar 15 (pp. 852-856). https://dx.doi.org/10.1109/ITNEC.2019.8728996.
3. Abed MM, Younis MF. Developing load balancing for IoT-cloud computing based on advanced firefly and weighted round robin algorithms. Baghdad Sc. J. 2019 Mar; 16(1): 130-9. https://dx.doi.org/10.21123/bsj.2019.16.1.0130.
4. Salman SA, Dheyab SA, Salih QM, Hammood WA. Parallel Machine Learning Algorithms. MJBD. 2023 Jan 22: 13-17. https://dx.doi.org/10.58496/MJBD/2023/002.
5. Zhang Y, Yao J, Guan H. Intelligent cloud resource management with deep reinforcement learning. IEEE Cloud Comput. 2017 Nov; 4(6): 60-9. https://dx.doi.org/10.1109/MCC.2018.1081063.
6. Sharma C, Sharma S, Kautish S, Alsallami SA, Khalil EM, Mohamed AW. A new median-average round Robin scheduling algorithm: An optimal approach for reducing turnaround and waiting time. Alex Eng J. 2022 Dec 1; 61(12): 10527-38. https://dx.doi.org/10.1016/j.aej.2022.04.006.
7. Hasan RA, Sutikno T, Ismail MA. A Review on Big Data Sentiment Analysis Techniques. MJBD. 2021 Jan 15: 6-13. https://dx.doi.org/10.58496/MJBD/2021/002.
8. Shafiq DA, Jhanjhi NZ, Abdullah A. Load balancing techniques in cloud computing environment: A review. J King Saud Univ Comput Inf Sci. 2022 Jul 1; 34(7): 3910-33. https://dx.doi.org/10.1016/j.jksuci.2021.02.007.
9. Al-Arasi R, Saif A. Task scheduling in cloud computing based on metaheuristic techniques: A review paper. ToCS. 2020 Jan 30; 6(17): 1-19. https://dx.doi.org/10.4108/eai.13-7-2018.162829.
10. Kaur J., Bahl K. Job Scheduling in Cloud Computing Using Genetic Algorithm. IJISET. 2018 Apr; 5(4): 255-259. https://ijiset.com/articlesv5/articlesv5s4.html.

11. Varma SS. Task Scheduling for Cloud Computing Using Improved Genetic Algorithm. IJIRSET. June 2022; 11(6): 7859- 7868 https://dx.doi.org/10.15680/IJIRSET.2022.1106142.
12. Tseng FH, Wang X, Chou LD, Chao HC, Leung VC. Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm. ISJ. 2017 Jul 21; 12(2): 1688-99. https://dx.doi.org/10.1109/JSYST.2017.2722476.
13. Kousalya A, Radhakrishnan R. Hybrid algorithm based on genetic algorithm and PSO for task scheduling in cloud computing environment. IJNVO. 2017; 17(2-3): 149-57. https://dx.doi.org/10.1504/IJNVO.2017.085524.
14. Shishido HY, Estrella JC, Toledo CF, Arantes MS. Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds. Comput Electr Eng. 2018 Jul 1; 69: 378-94. https://dx.doi.org/10.1016/j.compeleceng.2017.12.004.
15. Gawali MB, Shinde SK. Task scheduling and resource allocation in cloud computing using a heuristic approach. J Cloud Comput. 2018 Dec; 7(1): 1-6. https://dx.doi.org/10.1186/s13677-018-0105-8.
16. Aziza H, Krichen S. A hybrid genetic algorithm for scientific workflow scheduling in cloud environment. Neural Comput Appl. 2020 Sep; 32(18): 15263-78. https://dx.doi.org/10.1007/s00521-020-04878-8.
17. Praveenchandar J, Tamilarasi A. Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. JAIHC. 2021 Mar; 12(3): 4147-59. https://dx.doi.org/10.1007/s12652-020-01794-6.
18. Singh AK, Swain SR, Saxena D, Lee CN. A bio-inspired virtual machine placement toward sustainable cloud resource management. ISJ. 2023 Mar 13: 1-12. https://dx.doi.org/10.1109/JSYST.2023.3248118.

تحسين إدارة موارد السحابة اعتماداً على نظام ذكي

منال فاضل يونس

قسم هندسة الحاسبات، كلية الهندسة، جامعة بغداد، بغداد، العراق.

الخلاصة

تعد الحوسبة السحابية نموذجًا تقنيًا مهمًا يتمتع بأقوى إمكانات لتغيير الطريقة التي يتم بها تنفيذ أنشطة تكنولوجيا المعلومات. توفر السحابة موارد تكنولوجيا معلومات منخفضة التكلفة وقابلة للتطوير، و يسعى مقدمو الخدمات السحابية إلى تحسين نتائج الأعمال بحيث تصبح الأنظمة السحابية أكثر تعقيدًا. تحتوي السحابة الذكية على عدد من المشكلات ، بما في ذلك تحسين تصميم الخدمة السحابية وتوزيع الموارد بشكل تكيفي واقتصادي. هناك اتجاه متزايد نحو اعتماد التقنيات الذكية لتحسين إدارة السحابة على وجه الخصوص. يقدم هذا البحث طريقة ذكية لإدارة الموارد السحابية دون تأخير أو مشاكل. بدءًا من تصميم نظام يعتمد على الخوارزمية الجينية (GA) كأدوات حسابية لتحقيق الهدف وتوضيح طلب تنبؤي ثم اعتمادًا على خوارزمية round robin (RR) لتخصيص طلبات المعالجة. لتقييم أداء الطريقة، تم التحقق من الخوارزمية المقترحة في 145 طلب. حيث حقق نظام التنفيذ نتيجة معقولة.

الكلمات المفتاحية: الحوسبة السحابية، الخوارزمية الجينية، إدارة الموارد، خوارزمية Round robin، الجدولة.