



تاریخچه مقاله:

تاریخ ارسال: ۹۷/۵/۱

تاریخ اصلاحات: ۹۷/۹/۱

تاریخ پذیرش: ۹۷/۱۲/۱۰

تاریخ انتشار: ۹۷/۱۲/۲۲

Keywords:

Load balancing,
Dragonfly-
optimization-
algorithm
Resources allocation
Cloud computing

Resources Allocation with Dragonfly Algorithm in Cloud Datacenters

Seyed Saleh Miri^{*1}, Faraneh Zarafshan¹, Mahdieh Nadi Senejani¹

¹Ashtian Branch, Islamic Azad University, Ashtian, Markazi, Iran

Abstract

Load balancing, resource allocation management, and scheduling tasks are the most important challenges in cloud computing. Load balancing is performed at virtual machine level in data centers, while scheduling and allocation of resources are applied on the hosts and virtual machines, respectively. Therefore, the process of allocation of resources to virtual machines in cloud computing and maintaining the load balanced are performed using dragonfly optimization algorithm in this research. The proposed method consists of initializing the algorithm, determining the number of virtual machines and the number of tasks, implementing the dragonfly optimization algorithm, allocating resources, and scheduling the tasks by maintaining the load balanced on virtual machines. Compared to the other methods, improvement of the dragonfly algorithm is approximately 1.34 times more at execution time, 20 milliseconds faster at response time to requests and 6% improvement in term of number of migrations.

روش ارجاع به
مقاله

S. Saleh Miri, F. Zarafshan, M. Nadi Senejani, Resources allocation with dragonfly algorithm in cloud datacenters, Journal of Distributed Computing and Systems (JDACS) Vol.1, No.2, PP.65-73, 2019

Email: Benjamin.miri@hotmail.com

1 Introduction

Cloud computing is an emerging technology to offer, use and deliver software and provides information technology services including information, hardware and other shared computational resources using the Internet. Internet provides the most suitable infrastructure for cloud computing, because it hides the technical details from the users and the user does not need any knowledge or control on the cloud infrastructure while is used [1].

Cloud computing focus on user access to a set of virtual shared resources through the Internet based on demand of service. Cloud service providers deliver the application through the Internet,

while scheduling, allocating the resources and storing the data are performed in a location far away from the users. One of the major challenges in cloud computing is resource allocation which refers to identify available resources, scheduling of the resources and allocation proper resources to applications in order to improve performance of the system.



Resource allocation ensures proper allocation of resources (e.g., processor, memory, peripheral devices) to applications that requested a specific service.

There is a direct relation between the energy consumption and resource allocation in cloud computing. As a result of optimal resource allocation, optimal number of servers and fewer communication is required to be directed from cloud servers to users and vice versa. Therefore, lower energy is consumed in cloud environment.

The performance of cloud computing is limited due to two constraints including physical capacity of machines and order of running tasks. The order of running tasks should be performed efficiently, with the lowest energy consumption and cost. Classical methods as used for resource allocation are time-consuming, while may not reach to optimal results. In virtual systems, resources and their availability change dynamically which result to different functions in time. Heuristic approaches are used for problem solving in conditions that optimal result may not be achieved by ordinary approaches. Furthermore, load balancing in virtual systems is affected by time and cost parameters.

Dragonfly algorithm can be used to solve multidimensional and one-dimensional problems. An important feature of this algorithm is the ability to search entire the problem space and jumping from local to global optimum. It also does not require any specific controller to run.

In this research, a heuristic method based on dragonfly life is utilized to ensure optimal allocation of resources and load balancing in cloud environment. The remainder of this study is organized as follows. In section 2, related works are overviewed. Section 3 presents the proposed method based on dragonfly heuristic

algorithm. The result of simulations is discussed in Section 4. Finally, the conclusion is provided.

2 Background

Ms. C. Wang [3] in 2012 proposed method of allocating resources based on a genetic algorithm that uses a fitness function and this fit function is divided into three subordinate fitness functions. This method uses a pre-migration algorithm. It performs Migration before the scheduling operation. To do this, it uses three criteria: the amount of CPU consumed, the operating power of system, and the disk entry/exit rate. The algorithm used in this method is based on genetics using several fitness functions, which is done with the approach of reducing energy consumption and increasing the productivity of the resources of this research.

Mr. Polepally in 2017 [4] focused on major challenges occur in cloud computing and proposed the load balancing algorithm depends on the constraint measure (CMLB) then evaluated it with the HBB-LB, DLB and HDLB. In their method the load balancing algorithm is introduced to allocate the tasks to the virtual machines. The load balancing algorithm takes the tasks from the overloaded virtual machine and reallocates them to the under loaded virtual machine. Thus, the tasks are executed in less time and the response time of the system is decreased. As observed results show that by using this method migrated only three tasks while seven tasks in HBB-LB.

In one new research in 2018 [5] Ms. Zahra Amini and her team presented a load balancing method based on dragonfly optimization algorithm for resource allocation in cloud computing. In this method based on Reynolds point of view [2] behavior of particles divided in separation, alignment and integrity. They compare results of execution time, response time



and number of migrated tasks with ACO, PSO and HBB-LB algorithms and observed by using dragonfly optimization algorithm because of its speed and precision in scheduling operations for maintaining load balance when allocating resources to virtual machines and scheduling them, provided better result.

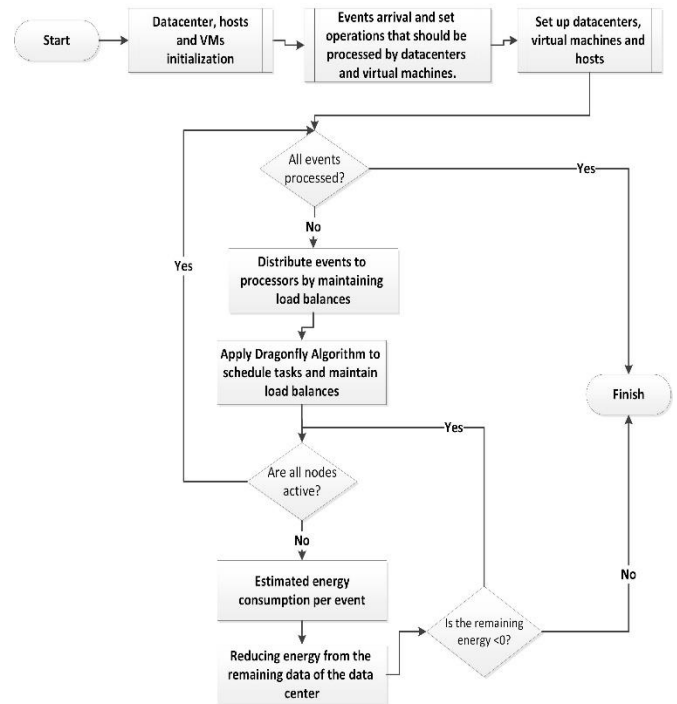
In 2018 Mr. C. Ashok Kumar and his team [6] presented technique for load balancing called Fractional dragonfly-based load balancing algorithm in cluster cloud model (FDLA). This algorithm combines DA and FC, which include exploration and exploitation and designed based on the interaction of dragonfly in navigating, search for foods and avoiding enemies, with the inclusion of FC in the update equation. The comparison of results in proposed FDLA technique performance with PSO, HBB-LB shows FDLA is effective with attain load of 0.2133 with number tasks reallocated as 14.

3 Methodology

In this research, the idea of scheduling tasks using dragonfly algorithm and load balancing approach has been used. In fixed or static masses, dragonflies form small groups and move in a small area forward and backward to hunt other flying insects like butterflies and mosquitoes. Environmental motions and sudden changes in the flight path are the main characteristics of the static mass. In dynamic masses, many dragonflies form a mass for migration in a long-distance direction [2].

The main idea of the dragonfly algorithm derives from static and dynamic congestion behaviors. These two congestive behaviors are very similar to the two main phases of optimization with metaheuristic algorithms consisting exploration and exploitation. Dragonflies form smaller groups and fly within a static group in different regions. This is the main purpose in the

exploration phase, while dragonflies fly in larger groups along one direction within a dynamic group which is a desirable behavior in the exploitation phase.



A general framework

3-1 Initializing datacenters in cloud computing

Initially, virtual data centers, virtual machines, hosts and physical machines are set. Then, input pattern of tasks and the operating set that should be performed by datacenters and virtual machines are determined by random distribution.

3-2 Launching the system components

Once initialization completed, it is necessary to activate so-called kernel or engine of the datacenters for starting the operation. At this stage, virtual datacenter, virtual machines, hosts and physical machines are launched as a software model. Afterwards, it is examined if all the events have been processed by the given processors in each datacenter.



3-3 managing the resources using the dragonfly algorithm

Once all the events processed, the algorithm goes to final step and ends. Otherwise, the following steps are performed:

- Input events are distributed among all processors and cores, and load balancing is maintained on every processor and datacenter. It should be noted that in such systems, it is possible to process of the Big Data, and the operation is performed with acceptable time and the use of minimum resources with the required productivity.
 - Processing the events one by one by processors and data centers
 - All energy consumption related to each event is calculated by datacenters and hosts and virtual machines.
 - Making scheduling of tasks using the dragonfly algorithm
- In in the next step, when all events are being processed, a minimize algorithm is used to check whether an event is ending. If events are ending, if all events are not processed, the previous steps are repeated, otherwise the following steps are performed.
 - Energy consumption of each event is calculated.
 - Reducing the amount of energy used of the entire remaining energy of data centers
- It evaluates whether energy has remained or not. If the energy has remained, it is dedicated to other

operations, and if there is no energy remained, the simulation will end.

As mentioned above, the dragonfly algorithm is used to balance load in cloud computing datacenters. Way of using this algorithm with the necessary details is presented below.

3-4 Load balancing method inspired from dragonfly's behavior

Cloud computation, by allocating computational tasks to a pool of dynamic resources, distributes online virtual machines according to the needs of different users and systems. Requesting customer services from different applications (software) in each datacenter is determined for each final server. Routing the request of services to various servers based on cloud management policies depends on the amount of load on individual servers, proximity to the database, and so on.

4 Simulation and Analysis of Results

The proposed method has been evaluated and simulated using MATLAB and random data. Table 1 shows the characteristics of the system used for simulation.

Table 1. Characteristics of the system for simulating and evaluating the results.

Hardware / Software	Characteristics
Operating system	Microsoft Windows 7 (32 bit)
RAM	4 GB (3.06 G Available)
Processor	Intel processor 7 cores (Core™) i7 CPU) with a frequency of 1.60GHz

Several scenarios have been defined for simulation. They are:



- 5 virtual machines and 40 input tasks
- 7 virtual machines and 40 input tasks
- 10 virtual machines and 40 input tasks
- 20 virtual machines and 40 input tasks
- 40 virtual machines and 40 input tasks

The result is further based on following criteria.

- Execution time of a task by virtual machines
- responding Time of virtual machines to a request
- Number of tasks migration
- Value of load balanced on the cloud computing system

The scenarios, initial values and distribution of tasks are chosen as the same for simulation of proposed and related methods.

Execution time of the tasks by virtual machines

As shows in Figure A the comparison of the execution time of the dragonfly optimization algorithm in order to schedule the tasks among virtual machines and allocate the resources with maintaining load balance in cloud computing with 40 tasks (considering the base paper [2]) and virtual machines from 5, 7, 10, 20 and 40 VMs in figures A-1 to A-5 respectively.

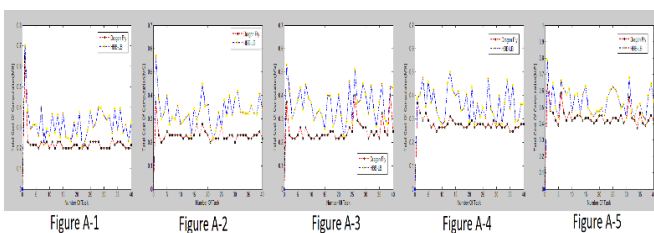


Fig. A. Comparing the execution time of the dragonfly optimization algorithm with other methods.

As observed, the execution time of the dragonfly algorithm in the proposed method is less than the

other methods and the tasks are run by virtual machines in the cloud computing at a more optimal time.

Response time to requests by virtual machines

The response time criterion represents the amount of time that a request, in the form of a task, is sent to the virtual machine until the virtual machine responds to that request. This criterion has also been used in many studies related to this research to prove the efficiency of its method. The lower the response time to requests in an algorithm, the higher the algorithm's efficiency, and virtual machines run tasks with a logical load balancing.

Figures B shows the comparison of response time to 40 tasks and virtual machines from 5, 7, 10, 20 and 40 VMs in figures B-1 to B-5 respectively in the proposed method compared to the method discussed in paper [2].

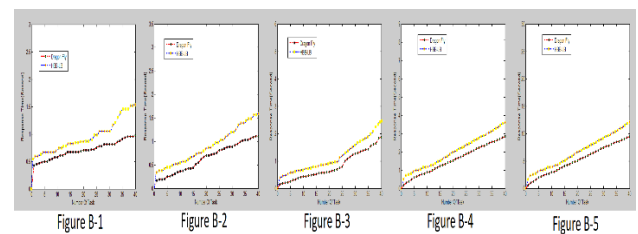


Fig. B. Comparison of the response time of the dragonfly optimization algorithm with other methods.

As shown in Figures B-1 to B-5, the total response time of the virtual machines to the tasks for running them is increasing to a great extent, but it will reach a stage that has a stability and the proposed algorithm has a specific stability and performs its activity with a uniform response time.



Number of tasks migrated

Its criterion is of great importance in scheduling and maintaining load balance and as a result allocating resources optimally. When the smaller number of tasks migrates during scheduling the tasks and allocating the resources, this means that load balance is maintained among the virtual machines and a smaller number of tasks are lost. The greater number of tasks migrates in the system, the greater the execution time and the response time in a great extent.

Figures C shows the comparison of the dragonfly method and the bee colony method, which is the newest research conducted in this field, for scheduling the tasks and allocating the resources with maintenance of load balance in cloud computing with 40 tasks and virtual machines from 5, 7, 10, 20 and 40 VMs in figures C-1 to C-5 respectively.

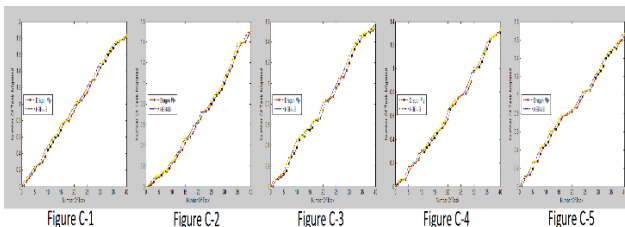


Fig. C. Comparison of the proposed method and other methods considering the number of tasks migrated.

4-3 Final comparison of the results of the proposed method

According to the results obtained in this section, the final comparison of the proposed method with other methods is investigated. Also simulated results compare with dragonfly optimization article [5]. Table 2 shows the comparison of different criteria and scenarios. It should be noted that the numbers listed in Table (2) are the average of the outputs for each

criterion. The unit for criteria of execution time is milliseconds and response time is seconds and migration of tasks is also based on the numbers.

Table 2. Comparison of criteria of 40 tasks for the proposed method compared to other methods.

#	Methods	5 VM	7 VM	10 VM s	20 VM s	40 VM s
1	Execution time of the dragonfly algorithm (MS)	0.21	0.22	0.24	0.27	0.42
2	Execution time of the bee algorithm (MS)	0.31	0.32	0.35	0.35	0.52
3	Execution Time of dragonfly optimization algorithm (MS)	0.31	-	-	0.31	0.31
4	Average response time of the dragonfly algorithm (S)	0.68	0.63	0.79	1.53	4.90
5	Average response time of the bee algorithm (S)	0.69	0.63	0.80	1.53	4.91
6	Average response time of dragonfly optimization algorithm (S)	0.81	-	-	0.81	0.81
7	Average number of migrations of dragonfly	0.92	0.65	0.77	0.61	0.83
8	Average number of migrations of bee	0.95	0.67	0.80	0.63	0.86
9	Average number of dragonfly optimization algorithm	2	-	-	1	1

As observed, Row 1 and 2 shows the comparison of execution time of tasks considering the



various number of virtual machines in the proposed method compared to the method discussed in paper [2]. Also, in row 3, 6 and 9 illustrated simulated results of comparing dragonfly optimization algorithm with presented algorithm.

As can be seen, considering the increase in virtual machines, the total retimes is increased to a great extent and the execution time of tasks is reduced.

Row 4 and 5 shows the comparison of response time to tasks considering the various number of virtual machines in the proposed method compared to the method discussed in paper [2].

As it's clear in row 4 and 5, the total response time of the virtual machines to the tasks for running them is increasing to a great extent, but it will reach a stage that reaches a stability and the proposed algorithm has a specific stability and performs its activity with a uniform response time. Graph 2 shows the comparison of the tasks migrated considering the various number of virtual machines in the proposed method compared to the method discussed in paper [2].

As it's shown in Row 7 and 8, considering the increase in the number of virtual machines, the number of tasks that will migrate will be reduced to a great extent, and this represents a balance between virtual machines.

In addition to the results and evaluations mentioned in the section above, and based on the fact that one of our goals is the reduction of energy consumption in cloud computing, due to the fact that the time of task is reduced, as a result the consumption of the processor and, mutually, the energy consumed by the Node is also reduced.

5 Conclusion and Suggestions

In this research, we have examined various criteria for assessing the efficiency and effectiveness of the dragonfly optimization algorithm, which are: execution time, response time, number of migratory tasks and load balancing. After finishing the simulation of the method proposed in this study, the following results were obtained:

- The execution time of the dragonfly algorithm to 4 to 40 virtual machines and 40 tasks is 0.278296 on average and it is 0.375082 for other algorithms. With the explanation, the improvement in the proposed algorithm is approximately 1.34 times more than other methods.
- The response time to requests in the dragonfly algorithm to 5 to 40 virtual machines and 40 tasks is 1.71274 on average and it is 1.71474 for other algorithms. With the explanation, the improvement in the proposed algorithm is approximately 20 milliseconds compared to other methods, this number is an acceptable value.
- According to the results from running the algorithm proposed, it is observed that the number of tasks in the mode which the dragonfly is run is 7.5 tasks on average for 5 to 40 virtual machines and 40 tasks. While, approximately 8 tasks emigrate on average in other methods. The improvement of the proposed method is approximately 6 percent.



5-1 Suggestions

- Use of whale optimization algorithms for scheduling tasks and allocating resources in cloud computing with maintaining the load balance
- Combination of particle swarm and genetic optimization algorithms for scheduling tasks and allocating resources in cloud computing with maintaining the load balance and comparing the results obtained with findings of this research
- Combination of dragonfly and genetic optimization algorithms for scheduling tasks and allocating resources in cloud computing with maintaining the load balance and comparing the results obtained with findings of this research
- Combination of shuffled frog leaping and genetic optimization algorithms for scheduling tasks and allocating resources in cloud computing with maintaining the load balance and comparing the results obtained with findings of this research.

6 References

1. Qian L., Luo Z., Du Y., Guo L.: *Cloud Computing: An Overview*. In: *1st International Conference on Cloud Computing*, pp. 626–631, (2009).
2. Mirjalili S., "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems", *Neural Compute & Applica*, 2015.
3. Wang C., Wang Q., Ren K., "Security Challenges for the Public Cloud", *IEEE Computer Society*, pp.96 -77, 2012.
4. Polepally, V., Shahu Chatrapati K., "Dragonfly optimization and constraint

measure-based load balancing in cloud computing." *Cluster Computing* (2017): pp.1-13.

5. Amini Z., Maeen M., Jahangir M. R., "Providing a load balancing method based on dragonfly optimization algorithm for resource allocation in cloud computing." (2018).

6. Kumar, C.A., Vimala, R., Britto, K.A. and Devi, S.S., *FDLA: Fractional Dragonfly based Load balancing Algorithm in cluster cloud model*. *Cluster Computing* (2018): pp.1-14.



سید صالح میری مدرک کارشناسی خود را در رشته مهندسی نرم افزار کامپیوتر در سال ۱۳۹۳ از دانشگاه آزاد اسلامی واحد رودهن و مدرک

کارشناسی ارشد خود را در سال ۱۳۹۷ در رشته مهندسی فناوری اطلاعات گرایش شبکه های کامپیوتری از دانشگاه آزاد واحد آشتیان اخذ کرده است. ایشان در حال حاضر به عنوان مدیر زیرساخت IT مراکز داده شرکت ابر آروان که ارائه کننده خدمات محاسبات ابری می باشد، مشغول به کار هست. زمینه های پژوهشی مورد علاقه ایشان عبارت اند از: محاسبات ابری، محاسبات توزیع شده و مجازی سازی.



فرانه زرافشان فارغ التحصیل ممتاز مقاطع کارشناسی و کارشناسی ارشد به ترتیب در رشته های مهندسی کامپیوتر گرایش سخت افزار و مهندسی کامپیوتر گرایش

معماری کامپیوتر از دانشگاه آزاد اسلامی اراک و فارغ



التحصیل مقطع دکتری رشته مهندسی سیستم های کامپیوتری از دانشگاه UPM است. ایشان در حال حاضر استادیار تمام وقت گروه مهندسی فناوری اطلاعات دانشگاه آزاد اسلامی واحد آشتیان و عضو هیأت مدیره شرکت دانش بنیان تحلیل پردازش ارقام هوشمند می باشد. وی استاد راهنما و مشاور بیش از ۴۰ پایان نامه مقطع کارشناسی ارشد و دکترا و استاد داور پایان نامه های ارشد در دانشگاه های مختلف بوده است. علاقه مند به پژوهش های ایشان عبارتند از اینترنت اشیا سیستم های توزیع شده و امنیت بحرانی، شبکه های موردی و بی سیم و امنیت شبکه.



مهدیه نادی سنجانی مدرک کارشناسی و کارشناسی ارشد خود را در رشته مهندسی کامپیوتر سخت افزار از دانشگاه آزاد اسلامی واحد اراک به ترتیب در سال های ۸۳ و ۸۵ دریافت کرد. مدرک دکتری را در رشته میکروالکترونیک از دانشگاه UTM مالزی دریافت کرد.

ایشان از سال ۱۳۸۶ عضو هیات علمی تمام وقت دانشگاه آزاد آشتیان است. تخصص تحقیقات ایشان در زمینه معماری شبکه های روی تراشه با توان مصرفی پایین و مدارات نانوکربنی است.