

Presenting Migration Scheduling Algorithm for Virtual Machines to Optimize Energy Consumption Simultaneously and Generate Pollutants in super Computing Networks

Maryam Moeentaghavi^{1*} Maryam Khademi²

1.Instructor of Department of Computer Engineering ,Islamic Azad University South Tehran Branch,Iran

2.Assistant Professor of Department of Applied Mathematics,Islamic Azad University South Tehran Branch,Iran

Abstract

In this research, we focus on the migration of virtual machines in the cloud data center using the inheritance algorithm. The simulation results confirm the feasibility and effectiveness of this timing algorithm and lead to a significant reduction in total energy consumption compared to other strategies. And since our focus is on the operational energy of the centers and with reduction in operational energy, producing Carbon biodegradation has also declined, which plays a significant role in reducing user costs.

Keywords: migration, car, super computing energy consumption.

DOI: 10.7176/CEIS/10-6-03

Publication date: July 31st 2019

Introduction

In 1969, Leonard Kleinrock, one of the leading scientists in the Network for Advanced Research Projects Agency (ARPANET), the founder of the Internet said networks are still in the first stages but when they develop or be complicated we will see that we will probably expand the computer industry ", which, like the electricity and telephone, currently provide services to homes and offices in the country. This insight from the general computing industry based on the service provider model predicts a massive transformation from the entire 21st century computing industry, whereby computer services will be demand-driven, such as other utility services present in today's society that they can be easily accessible. Similarly, users (consumers) pay only when they have access to the computing services of providers. In addition, consumers no longer need to invest heavily in building and maintaining complex IT infrastructure. In such a model, users access their services regardless of where those hosting services are. This model is referred to as general computing or recent cloud computing [2]. The second one represents the infrastructure as a "cloud" from which users can access applications as services from anywhere in the world in terms of demand. Hence, cloud computing can be introduced as a new paradigm for providing dynamic services for computer-supported datacenter support, which usually uses virtual machine technology for environmental fixation and goals. [3].

Until recently, the only concern was the deployment of high-performance super data centers without any attention to energy consumption. Data centers use an average energy of as much as 25,000 households [4]. A significant portion of the electrical energy consumed by computing resources is converted into heat. High temperature leads to a number of problems, such as reducing system reliability and availability, as well as reducing the life of the devices. Data centers are not only costly, but also seriously harm the environment. The carbon produced by these centers is now more than that of a country with all industrial facilities [5]. The high carbon content is due to the large amount of power required to cool multiple host servers in data centers. super service providers to ensure their profits are not drastically reduced due to high energy costs, while also providing high-quality, high-speed services, as well as increasing pressure from governments around the world to reduce carbon footprint. Which has a significant impact on climate change, needs to be taken. [6].

Meanwhile, virtualization solutions, in addition to reducing energy costs in data center infrastructures, dramatically exploit and increase server performance. Virtualization using Machine Immigration Technique Virtual, with the integration of load balances between physical servers and the implementation of ten applications or more virtual machines on an x86 server, can transfer virtual resources with a high degree of flexibility between physical servers [7].

What is being done in this research is the presentation of a migration virtual machine algorithm to solve the energy problem and pollution in the computational environment.

Research Methodology

Suggested algorithm

We set the p variable to display the physical machines in the system.

$P = \{p_1, p_2, \dots, p_n\}$, and n is the number of physical machines we consider here to compare with other algorithms 800. P_i refers to the unique physical machine, which is the number of the physical machine and is in the following range. $1 \leq i \leq n$. We also show virtual machines on each host with the letter V: $V_i = \{v_1, v_2, \dots, v_m\}$, where m

is the number of virtual machines on the physical server I . The total number of virtual machines is set to 1052.

We intend to allocate virtual machines in some way to physical machines to minimize the energy consumption of servers and to ensure the quality of the user's program execution. To ensure the quality of our applications, we use the virtual machines migration. A number of proposed energy-saving algorithms shut off the physical machine when the physical machine arrives at the critical point of energy consumption, in order to reduce energy consumption, and naturally all virtual machines that are based on the physical machine are also disabled and suspended by users. But in our work, with using the virtual machine migration technology, the user's program is running in anyway, and this means that when the energy of a physical machine reaches the critical point, here it is up to 90% and 10% using the physical machine processor will be adjust and the additional virtual machines migrate to other machines using the live migration method.

The proposed timing algorithm ensures the quality of user program execution because it uses live migration to migrate virtual machines. In this type of immigration, the immigrant machine operating program continues to migrate and does not stop there. Therefore, the user program continues at its own time and the user will not notice this change and will not be delayed.

This means that the normal immigration is used , programs are completely disconnected before the migration and they are resumed at destination, which will waste user time. On the other hand, shutting down the device, all programs on it will stop and run again. However, during live migration while migrating, all commands and the status of the system are being transport to the destination and will continue to run.

Based on the algorithm presented by Mr. Buya et al. [73] and its presentation in the simulators, we selected the critical point of 90 percent and 10 percent for migration.

Based on this algorithm, if a physical machine processor uses more than 90 percent of its capacity, a number of virtual machines based on it will emigrate altogether, due to an increase in the energy consumption of the physical machine after 90 percent of the use of the processor , this will reduce energy consumption.

On the other hand, when using the processor to reach less than 10% of its capacity, we decide that all virtual machines based on it will migrate to other physical machines and the physical machine will be turned off to minimize energy consumption. This is because of when the physical machine turns on, even if no virtual machine is on it , it consumes a lot of initial energy, so it's adjusted to 10 percent, so the physical machine does not consume much energy at a low cost.

In this algorithm, we use the genetic optimization algorithm to solve the problem.

Genetic algorithms have a lot of differences from the usual optimization methods and search procedures:

1. The genetic algorithms answers the encoded set of algorithms, not the answers themselves.
2. Genetic algorithms work with a population of answers, not with the answer itself.
3. Genetic algorithms use the final result information (matching function), not from derivation or other auxiliary knowledge.
4. Genetic algorithms use probability transfer rules, not definitive rules.
5. Genetic algorithms operate in an atmosphere independent from the problem space.

As stated in the previous chapter, there are five main functions in the genetic algorithm:

- 1- Initial Population Initialization
2. Evaluation of the population ()
- 3- Selection ()
- 4- Crossover ()
- 5- mutation ()

In the proposed algorithm, with the change we make in the genetic algorithm, we added the sixth function to the routine that checks the load on the servers, and then, in the event of need, the migration process takes place. And then re-evaluated and the string is selected with minimum energy consumption

- 6- Overload / underload

- 7- Migration

Primary population:

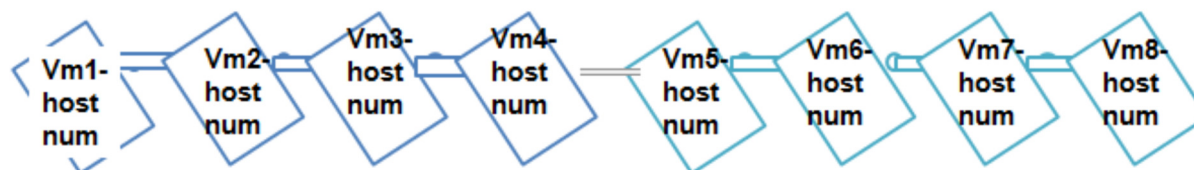


Figure 3-2 The proposed chromosome string

The above table is an example of a population group or the same chromosome. There are 200 chromosomes that randomly arranges the location of VM on hosts. Each chromosome is a solution that randomly allocates virtual machines to the servers.

Based on this, we generate 200 chromosomes (chromosomes) that randomly determine each virtual machine site (VM) in hosts. We chose each chromosome to a length of 1052. 1052 is the number of virtual machines that we have to assign to 800 hosts. Each gene takes up one of 800 servers in a way that minimizes energy consumption. So, we consider each chromosome to be a single-row array that has 1052 columns. Each node specifies which machine is deployed on which host. So, any gene in the genetic expression is the physical machine number. For example, in the first one, if the value is 275, it means that the first virtual machine is located on the 275M server. Each virtual machine can only be on one host at a time. On the other hand, each host has a limited capacity.

We apply these conditions to the production of primary populations so that our population is well placed.

We repeatedly changed the number of repetitions and the best result was repeated to 1000. Therefore, each of the 200 chromosomes will be updated 1000 times and will be optimized during the steps of the genetic algorithm (evaluation-combination-mutation).

Evaluation function: Power (chromosome) Σ , at first we calculate the host evaluated energy and then the total energy of the hosts of a string. At the end, the energy of the strings is compared to the minimum energy string selected.

In this algorithm, using a table in paper [74] two of the cloud computing pioneers, namely Buyya and Belaglazov, tried to evaluate the function selection. The table below is based on two types of processors in data centers, super data are used and simulations are used in this model. In the first line, the load on the server is expressed in percentages, and in the second and third lines, respectively, the energy consumption of these processors is given according to the workload.

Server	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	116	121	125	129	133	135

Table 3-1 Energy Consumption of Processors Due to Workload

The specifications of the two processors are as follows:

HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores _ 1860 MHz, 4 GB),

HP ProLiant ML110 G5 (Intel Xeon 3075, (2 cores _ 2660 MHz, 4 GB)

Therefore, with these numbers, when the first type server is turned on but no program is running on it, its power consumption is 86 watts, and as much as 10 percent of the processor's capacity is loaded, the power consumption is 89.4 watts per hour. The rest are similarly achieved, and at the end of the day we see that when the processor is 100 percent, its power consumption is 117 watts per hour.

The second type processor is the same as when the server is turned on and idle, its power consumption is 93.7 watts and when the server is 100 percent, the power consumption is 135 watts per hour.

Since both types of these servers are dual-core, we consider the full dual-processor capacity.

With regard to the energy consumed per server workload, a linear function is obtained. Because each type of processor has a steady slope, the power consumption is higher.

we use the following formula:

If the program goes to the first type processor, then we calculate the 31% $86 + \text{workload per host}$.

This is the energy used to turn on the device plus the difference of completeness and the empty server multiplied by percentage of first type server's fullness.

If the program goes on the second type processor, we calculate $93.7 + 46.3 * \text{Percentage of work per host}$.

This is the energy used to turn on the device plus the difference between completeness and the empty server multiplied by the percentage of the second type server fullness. By doing so, we will determine whether the workload has been reached by the host to the designated critical point or not

If the host's workload is less than 10%, the virtual machines will transfer from it to other hosts on this condition so that the destination server does not reach to this critical point.

Also, if the host's workload reaches over 90% of its capacity, we will ship several virtual machines from this host to other servers to achieve a normal percentage load.

Selection

In this section, check the chromosome evaluation function. The population is considered to be 100 in each repetition, and each chromosome is evaluated, we select five to the lowest of the parent and we consider straightforward as a child about 95 others we go on in the process of combining and jumping operators. In fact, each population consists of five parents with the least value and 95 genes obtained from the genetic algorithm.

The combination

For the composition process in this algorithm we use a one-point and two-point routine

Single Point Procedure:

This method is the most common combination method. In this method, a point is randomly assigned to chromosomes

The parent is selected and combined with each other, in a way that from the beginning of the first parent to that point the parent genes of one inherits from that point to the end of the second parent's inheritance.

Two-point routine:

In this method, two points are randomly selected in the parent chromosomes, and then the genes between the selected points exchange with each other alternately. It inherits from the beginning of the first parent to the first point and from the second to the end of the first parent and inherits between the points from the second parent.

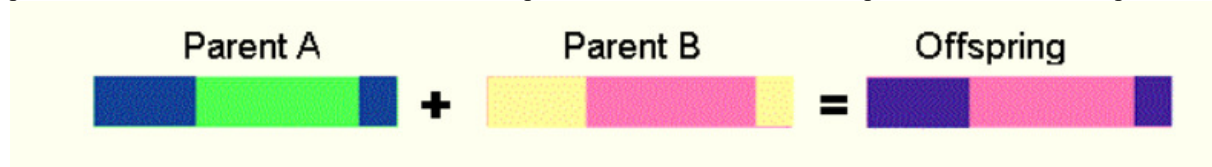
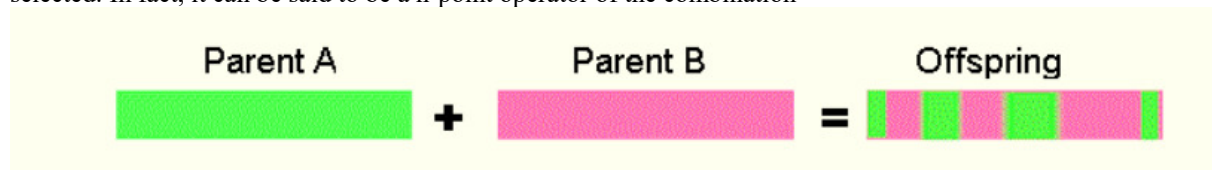


Figure 3-5 Composition - Two-point routine

The following example shows an example of this method

Uniform routine

In this routine, randomly, parts of the child are selected from the first parent and parts of the second parent are selected. In fact, it can be said to be a n-point operator of the combination



Combination rate

In practice, we use the combination of the probable rate. For example, based on our experience, we suggest a seventy percent rate. On the other hand, in each repetition, a random number is generated between one and one hundred. If the replication number is between one and seventy, then the combinator is applied to that string, otherwise it goes to the next step of the algorithm where the mutation operator is applied.

Leap

For the mutation process, we use a bitwise method. This method randomly changes a bit in the string.

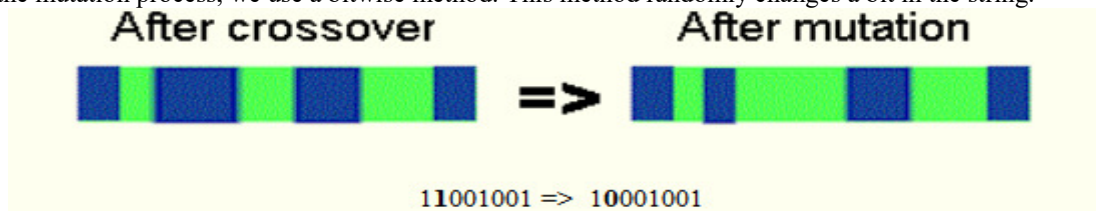


Figure 3-8 mutation-bit

In this process we use a leap rate of 0.05. This means that a random number is generated between one and 100 if the number is between one and five, then the mutation occurs, otherwise we go to the next step. This is because the mutation should happen very little, until a little change occurs. Not in our fields. On the other hand, the mutation is used to ensure that if the optimal global point is outside of the current search range, it will allow an algorithm to scan those points. If the value obtained from the jump routine is better than the current search points, the algorithm moves to that range and otherwise it continues to search in the same range.

In the next step, we examine the overload and underload on each chromosome.

If the use of the host CPU exceeds ninety percent of its capacity, overload or overflow has occurred.

And if the use of the host CPU is less than 10% of its capacity, then underload or minimal labor has taken place. During overloads, large amounts of workloads are consuming a lot of power and power, and this lack of energy has a direct impact on the cost of paying customers to use super services and reduce the quality of service, and has caused problems such as reduced availability and reduced lifetime of the device. It also lowers the reliability of the system. It is therefore decided to migrate a number of virtual machines from this host to other hosts based on the machine selection algorithm for migrating MMT to return the host state to normal.

During underload, because there is a small number of virtual machines or programs on a host that works with great expense and power, it is decided that the host is shut down and resting and virtual machines on the host also Other active hosts migrate.

In this section, our fields have reached their final status after passing through the stages of combination, mutation and migration. Here, we evaluate the strings using the objective function. The string that consumes the least amount of energy is sent to the output for a graceful string.

Results

2.4 Characteristics of Virtualization Assignment and Migration Simulation

Definition of problem: The goal is to introduce simulation for academic and industrial purposes. We test new systems for better performance than super infrastructure. By simulating, efficiency and resource efficiency for resource allocation policies and resource scheduling are measured. And developers are able to test the overall efficiency of systems for migrating policies. The simulator helps analyze and model various components of super computing. Using this simulator, cloud service providers can measure the performance of applications from customers. Here is a list of the ideal features of cloud computing simulation.

A: Physical machine:

This feature is able to simulate physical machines with different hardware features, which is a high level of abstraction from a physical machine. Abstraction means that the physical machine is a set of resources that these resources are shared through several virtual machines, which is done by virtualization techniques. The interface allows the creation, start, stop and destruction of virtual machines within a physical machine. The interface also provides different types of applications for configuration and feature assignment among different virtual machines.

B: Virtual Machine:

We will be able to simulate a virtual machine with specific resource requirements and virtualization technologies. This abstraction is a high level of virtual machines. The virtual machine exploitation executed on the host of the physical machine, which shares the resources of the underlying layers to complete an application. And the interface allows us to approve and take apps in a virtual machine.

C) Different virtualization technology:

For the requirements and characteristics of each virtualization technology, users should be able to point out which virtualization techniques will be used and depending on the technology, different features provided by that technology will be created, such as gene technology.

One of the main benefits of virtualization is the ability to migrate to virtual machines. The migration process will affect the use of resources from source and destination machines. In both machines, the use of CPU and bandwidth increases dramatically.

D) Applications:

Ability to simulate applications with different configuration of applications on a specific machine (all requirements) and at a specific time. We see the use of a set of resources that can be independent and dependent on external concepts but depend on hardware resources and can be implemented sooner or later. Time to display the features of an application there should be many choices for the user on how the resources needed can be displayed.

E) Allocation of resources:

Resource allocation policies decide on the amount of resources assigned to a particular virtual machine or a set of them. These policies can also dynamically allocate resources. For our prioritization implementation, we can prepare more resources for a virtual machine more than a different virtual machine ratio. Non-functionalities can interfere or be shared with each other. Storage areas can use different technologies, similar to direct or network storage. Different technologies, different delays, and different CPU chips. Storage technology can have a significant impact on the system's entire system.

F) storage systems

Storage systems are one of the most important parts of data centers. Which can be divided into 1 network storage area (SAN) 2- storage area. Most network storage areas are used in data centers. The storage system contains the image of the virtual machine, which is the virtual machine's instant display. The use of SAN leads to challenges for the placement of virtual machines, but instead it provides flexibility.

G) Position policies

The placement policies are addressed both to the virtual machine and to the application placement sites. The placement of virtual machines is about the destination of the physical hosts, which decide on a number of parameters of the parameters. The context of the placement of the applications into existing resources and the number of running applications depends on the virtual machine.

Positioning policies can be categorized by:

1) Initial placement policies (fixed) 2) Dynamic placement policies

Initial placement policies are implemented at the time of the launch of an application or virtual machine placement, in the context of dynamic placement policies at runtime, which depends on different parameters, similar to the amounts of different sources available with the full SLA (service approvals).

H) Immigration policies

Immigration policies decide which and where a virtual machine migrates. These policies limit the circumstances that make migration to a virtual machine.

The destination machine that has been tampered with must have a number of conditions. For example, a virtual machine must be migrated when the load generated by the running application exceeds 90% of the virtual machine's capacity.

T) virtualization techniques

Different virtualization techniques implement different resource allocation methods.

I) price model

In practical terms, providers of physical cloud service providers rent or run user apps. Based on a number of pricing models, the price model can be based on specific hardware features or resources to be used. Types of rents, pricing models, and the use of the model can be implemented. The price model helps us to get the values that can be saved using super computing tools. [77]

4-3 MATLAB software

Matlab (MATLAB) is the name of one of the computer software for mathematical computing. The term MATLAB also means the digital computing environment, and also the meaning of the corresponding programming language, which is made up of the combination of MATRIX and LABORATORY. This name implies a matrix-oriented approach to the programs, in which even single numbers are considered as matrices.

MATLAB, which is a product of METWOKS Co., is used by various groups of engineers in a variety of fields, including electrical engineering, mechanics, computers, and so on. The core of MATLAB has been written in C for high speed and high performance, but its graphical interface is implemented in Java.

MATLAB's programs are mostly open source, and in fact MATLAB (like BASIC) is a commentator, not a compiler. The power of MATLAB is due to its flexibility and ease of operation, as well as a variety of companies and groups, including universities around the world, and some engineering companies annually add custom-utility gadgets that enhance their efficiency and popularity

A list of these toolbars are listed below:

- MATLAB Telecommunications Toolbox, Telecommunications Engineering Functions and Computing Tools
- MATLAB control toolbox, control functions and calculating tools
- Fuzzy MATLAB toolkit, functions and fuzzy computing tools
- MATLAB calculator tool, numerical calculator functions and tools
- Simulink MATLAB toolkit, a toolkit for simulating single systems
- Toolkit for estimating mathematics, functions and computing tools. System estimation in control engineering
- MATLAB statistic toolkit, functions and tools for statistical calculations.
- Toolkit for collecting data from MATLAB, functions and data collection tools
- MATLAB neural network tools, functions, and neural network computing tools
- Toolkit for image processing, functions and image processing tools
- MATLAB sound processing tools, audio processing functions and tools
- Toolbox for MATLAB's probabilities

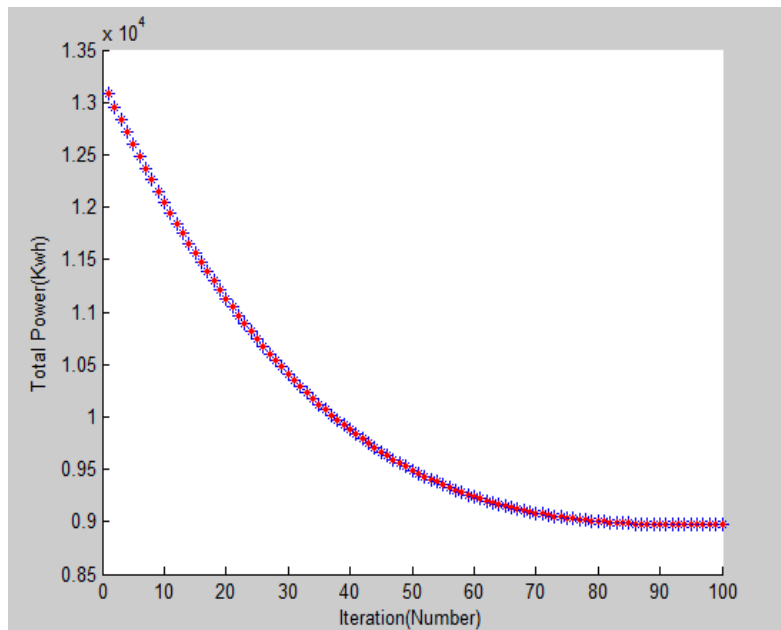
Real-time workstation toolkit, real-time system functions and tools

4-4 Simulation Results

We used different methods of checking the workload and choosing a migrating virtual machine to integrate into the proposed scheduling algorithm. It also uses a one-point, two-point, multiplication method, ... for the scheduling phase, and bitwise methods, etc. for the jump scheduler stage.

Finally, the best results were obtained after frequent changes in the variables with respect to the change of the combinator operator and the number of repetitions. Here are some charts that are more relevant to the graph and change of other variables:

Repeat for 100 with single point operator and bitwise mutation and the presented selection operator with the population of 100 per replication, the result is obtained just like below.

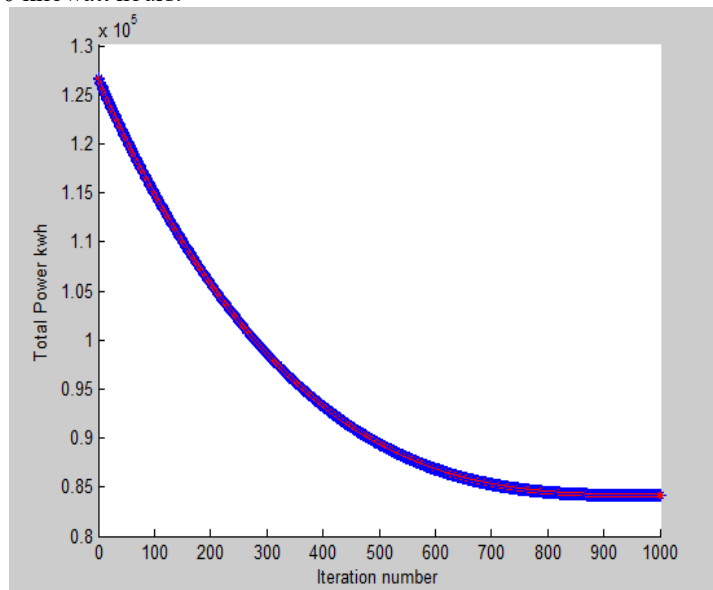


Graph 1-4 - The scheduler provided with the number of repetitions 100 and the one-point combination operator.

As you can see, energy consumption in this way has reached about 90 kWh.

Then we chose the Scatter function operator, which determines the number of points with the probability number that it generates and then the combination is made. In fact, it can be called the infinite point combination.

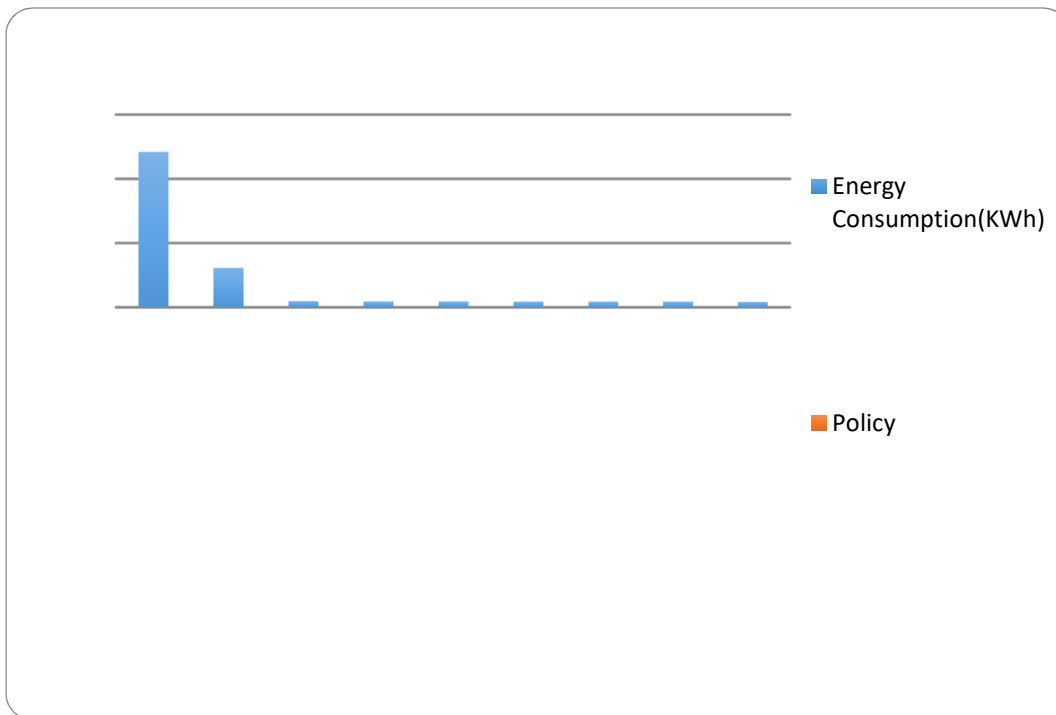
With a repeat shift to a thousand and dispersed combiners, the result was improved and the minimum energy consumption was 84.40 kilowatt hours.



Compared with other proposed methods, the proposed method shows energy efficiency improvement. The table below shows this comparison.

<i>Policy</i>	<i>Energy kwh</i>
NPA	2419.2
DVFS	613.6
THR-MMT-1.0	95.36
THR-MMT-0.8	89.92
IQR-MMT-1.5	90.13
MAD-MMT-2.5	87.67
LRR-MMT-1.2	87.93
LR-MMT-1.2	88.17
<u>Suggested Algorithm</u>	84.40

Table 4-1 Comparison of Different Algorithms Based on Energy Consumption Servers (Kwh)

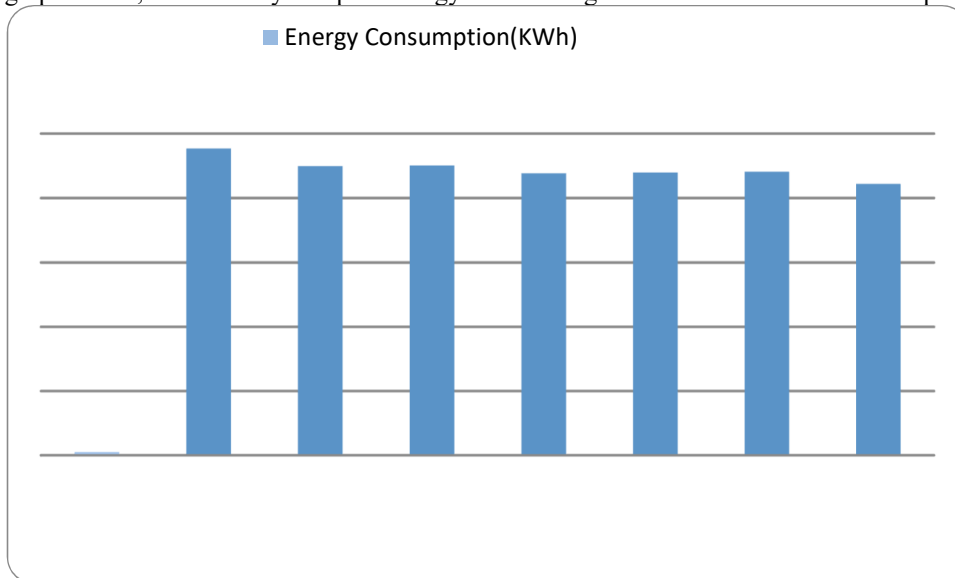


Graph 4-1 Comparison of Different Algorithms Based on Energy Consumption of Servers (Kwh)

The NPA algorithm is actually an algorithm that does not attempt to optimize the energy consumption of the data centers, and considers other factors such as the quality of the work.

We plotted this algorithm to provide a general comparison between energy consumption based on energy-oriented algorithms and other algorithms.

In the graph below, we will only compare energy-oriented algorithms. To further see these optimizations.



Graph 4-4 Comparison of energy-conscious energy sources based on energy consumption of servers (Kwh)

As we can see, the proposed algorithm reaches a minimum of 84.40, which is a very good result among the previous algorithms. As can be seen, this algorithm is even more accurate than the algorithm presented by Mr. Buyya et al. (LR-MMT) which has reduced energy consumption. The LR-MMT algorithm worked on the critical point survey with the LR algorithm and the allocation of additional resources using the MMT algorithm.

And based on the comparison of the rectangular diagram below, it is clearly obvious that the proposed algorithm based on the genetic algorithm improves the energy consumption of super data centers as compared to the previous methods. As a result, less electricity is consumed and the produced carbon decreases. The result of the bio-pollutants that has become very challenging in the world in recent years is reduced.

Conclusion

As you can see in the results section, we measured the effectiveness of our method by performing different experiments, and in order to better identification in this performance, we compared these results with the results obtained from different methods. The results of the comparison showed that our proposed method reduces the energy consumption of the super data centers and thus reduces the production of biological pollutants. And while it ensures the quality of the user's services, and with the live migration method used in this algorithm, user programs are run without any disruption even during migration.

In the next chapter we will summarize what is discussed in this thesis, and also suggest some of the things that can be done to improve our approach to a more complete approach.

References

- [1] A. Gandhi, M. Harchol-Balter, R. Das, C. Lefurgy, "Optimal Power Allocation in Server Farms", in proc: The Eleventh International Joint Conference on Measurement and Modeling of Computer Systems, pp: 157-168,2009.
- [2] A. Weiss Computing in the Clouds. networker ", ACM Press, Vol. 11, pp. 16-25, 2007.
- [3] A.Bloglazov, R..Buyya, "Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers," In Proc: 8th International Workshop on Middleware for Grids, Clouds and e-Science, No 4 , 2010
- [4] Amazon Elastic Compute Cloud (EC2), <http://www.amazon.com/ec2/>, July 2008.
- [5] Azure Service Platform, Microsoft Corporation, <http://www.microsoft.com/azure/services.msp>
- [6] Azure Service Platform, Wikipedia, http://en.wikipedia.org/wiki/Microsoft_Azure
- [7] B Lin,, and P. A. Dind, "Mixing Batch And Interactive Virtual Machines Using Periodic Real-Time Scheduling". In Proc: the 2005 ACM / IEEE Conference on Super Computing, pp: 1-8, 2005.
- [8] B. Sotomayor, R.S. Montero, I.M. Llorente and I.Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," in Proc: IEEE Internet Computing, pp: 14-22, 2009.
- [9] C L. Karr and L.M.Freeman, "Industrial Applications of Genetic Algorithms", CRC Press, 1999.
- [10] C P. Sapuntzakis, R. Chandra, B.Pfaff, J. Chow, M.S. Lam, M.Rosenblum, "Optimizing the Migration of Virtual Computers". In Proc: The 5th Symposium on Operating Systems Design and Implementation (OSDI) vol 36, pp. 377-390, 2002.
- [11] C.P. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, M.S. Lam, and M. Rosenblum, "Optimization of Virtual Computers", In Proc: 5th USENIX Symposium on Operating Systems Design and Implementation (OSDI-02) vol. 11, pp. 377-390, 2002.
- [12] C.Scheffy, "Virtualization For Dummies", Amd Special Editors, Publisher: Wiley, ISBN: 978-0-470-14831-0, 2007.
- [13] D Meisner, B. T. Gold and T. F. Wenisch, "PowerNap: Eliminating Server Idle Power," in Proc: Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp: 205-216, 009.
- [14] D Nurmi, R. Wolski, C. Grzegorzcyk, G. Obertelli, S. Soman, L. Youseff and D. Zagorodnov, "The Eucalyptus Open-Source Cloud-Computing System". In Proc: 9th IEEE / ACM International Symposium on Cluster Computing and the Grid (CCGRID), pp: 124-131, 2009.
- [15] D.Ongaro, A.L. Cox, and S.Rixner, "Scheduling I / O in Virtual Machine Monitors," in Proc: ACM SIGPLAN / SIGOPS International Conference on Virtual Execution Environments, pp: 1-10,2008.
- [16] E Harney, S. Goasguen, J. Martin, M. Murphy, and M. Westall, "The Efficacy of Live Virtual Machine Migrations Over the Internet," in proc: Second International Workshop on Virtualization Technology in Distributed Computing,, 2007.
- [17] F.M.Aymerich, G.Fen, S.Surcis, "An approach to a cloud computing network, Applications of Digital Information and Web Technologies," First International Conference ICADIWT, pp. 113-118,2008.
- [18] G. Dhiman, G. Marchetti and T. Rosing, "vGreen: A System for Energy Efficient Management of Virtual Machines", Journal of Transaction in Design, Automation of Electronic System, Vol. 16, pp: 6-32, 2010.
- [19] G. Lovasz, F. Niedermeier and H. De Meer. "Performance Compositions of Energy-Aware Virtual Machine Consolidation," Cluster Computing Journal, vol 15, pp. 36-42, 2012.
- [20] Google App Engine, <http://appengine.google.com>
- [21] H Kim, H.Lim, J. Jeong, H. Jo, and J. Lee, "Task-aware Virtual Machine Scheduling for I / O Performance", In Proc: ACM SIGPLAN / SIGOPS International Conference on Virtual Execution Environments, pp : 101-110, 009.
- [22] I.Goiri, F. Julià, R.Nou, J.L. Berral, J.Guitart, J. Torres, "Energy-Aware Scheduling in Virtualized Datacenters," IEEE International Conference on Cluster Computing, pp. 58-67, 2010.
- [23] I. Menken, G. Blokdijk, "Cloud Computing Virtualization Specialist Full Certification Kit" Publisher: Emereo, ISBN: 9781921644047, 2008.
- [24] J. Bali, R. Ayre, K. Hinton, R.S. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage

- and Transport", in proc: IEEE Press, Vol. 99, pp: 149-167, 2011.
- [25] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle. Managing Energy and Server Resources in Hosting Servers. In Proc: Symposium on Operating Systems Principles (SOSP), pp: 103-116, 2001.
- [26] J. Heo, D. Henriksson, X. Liu, T. Abdelzaher, "Integrating Adaptive Components: An Emerging Challenge in Performance-Adaptive Systems and a Server Farm Case Study," in proc: the 28th IEEE Real-Time Systems Symposium (RTSS) , pp: 61-72,2007.
- [27] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making a Scheduling Cool: Temperature-Aware Workload in a Data Center." In Proc: USENIX Annual Technical Conference, pp: 5, 2005.
- [28] J. Stoess, C. Lang and F. Bellosa. "Energy Management for Hypervisor-based Virtual Machines". In Proc: USENIX Annual Technical Conference, pp: 54-60, 2007
- [29] J. Stoess, C. Lang, F. Bellosa, "Energy Management for Hypervisor-based Virtual Machines", in Proc the USENIX Annual Technical Conference, pp: 1-14, 2007.