International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2017, pp. 471-475
e-ISSN:2278-621X

A SURVEY ON ENERGY EFFICIENT RESOURCE SCHEDULING ALGORITHMS FOR CLOUD DATA CENTERS

Mr. Lazaras Lobo[1], Ms. Nausheen Mulla[2] & Mr Santhosh B[3]

Abstract – Cloud computing is utility oriented. The primary concern in data centers is Energy Efficiency. The cloud computing data centers consume huge amount of energy and operational cost. This paper will illustrate about the techniques of saving energy and operational cost data centers. In this paper we focus on energy efficient resource allocation fundamentals, scheduling algorithms and software technologies for energy efficient computing. We have considered ONWID, OFWID, MBFD, Round-Robin algorithms.
Keywords –Energy Efficiency, ONWID, OFWID, MBFD, Round-Robin.

1. INTRODUCTION

Energy efficiency in cloud computing is becoming very important due to the use of cloud computing in every field. As it is on-demand computing which is an Internet-based computing which also provides data to computers and devices on demand. Because of its low cost of services cloud computing has become highly in demand .It focuses in managing and configuring the application online at point of time and offer load obtaining that makes easier and reliable. With the emergence of Cloud computing many companies are shifting from traditional to online services and they are using the services of DCs. So the major intention of energy efficient cloud computing is to reduce the waste of energy by the servers.

1.1 Energy efficiency challenges–

In this section we present the most important power management and energy efficient algorithms. Data centers generates more carbon emissions and high energy costs incurred due to large amounts of electricity needed to power and cool servers hosted in that data centers. Minimizing the energy usage of data centers is challenging and complicated issue because computing applications and data are increasing where large servers and disks needed to process them fast enough within the required time. To achieve an efficient processing and utilization of computing infrastructure green cloud computing is achieved successfully to minimize energy consumption. Data centers need to be maintained in an energy-efficient manner to save energy in cloud computing.

1.2 Problem–

The virtual machine on physical machine shares CPU, memory and network bandwidth. Allocating and adapting the virtual machines and taking out the features of the hosting physical machines are challenging in scheduling the resources in cloud data centers. The problem is defined as the given the m identical machines PM1,PM2,….PMm and a set of n request make it possible in saving energy efficient, the objective is to provide efficiency of data center ,while delivering the quality of Service.

2. LITERATURE SURVEY

In this Section we consider various energy efficient scheduling algorithms.

2.1 Round-Robin:

Round-Robin is an algorithm which is used commonly for scheduling the tasks in datacenters. This algorithm allocates VM requests to each PM And Round-Robin is simple and easy to implement.

2.2 Modified Best Fit Decreasing (MBFD):

MBFD is a bin-packing algorithm. This algorithm sorts all VM in decreasing order depends upon the CPU current utilization and allocates each VM to host which provides the maximum least power consumption. This helps in leveling and lower energy consumption by choosing most power- efficient nodes first.

For same resources (PM), the VM can be allocated to any running PM that can still host because the power increasing is the same for same resources. The complexity of the allocation part of the algorithm is nm, where n is the number of VMs that must be allocated and m is the number of hosts. MBFD needs sorting requests so that it is only suitable for offline (or semi-offline) scheduling.

---

[1] Department of Computer Applications, AIMIT, St Aloysius college (Autonomous), Mangalore, Karnataka, India
[2] Department of Computer Applications, AIMIT, St Aloysius college (Autonomous), Mangalore, Karnataka, India
[3] Department of Computer Applications, AIMIT, St Aloysius college (Autonomous), Mangalore, Karnataka, India

2.3 Online Without Delay (ONWID):

This algorithm is also known as one request each time. In increasing order it allocates the tasks to PMs. If all the PMs are busy with another tasks a new PM will be powered on. When the total number of PMs are fixed and still the request cannot hosted then the request will be blocked.

Input: VM requests (each indicated by their required VM type ID, start time, finish time, and requested capacity), the interval of start time and finish time of request Ii
Output: Assign a PM ID to each request and allocate an interval for each request.

1. **d=0;**
2. **for j = from 1 to n do**
3. **foreach Ii that precedes Ij**
4. **if they are not overlapped or overlapped but still can share resources of an allocated PM do**
5. **allocateIj to the PM**
6. **else**
7. **start a new PM;**
8. **d=d+1;**
9. **allocateIj to PM d;**
10. **endif**
11. **endforeach**
12. **endfor**

2.4 Offline without Delay (OFWID) algorithm:

This algorithm knows the entire request in advance and without delay follows the request. According to the request start-time it sorts and allocates to the PMs in an increasing order of their IDs. If all PMs are running busy other tasks then a new PM will be turned on to execute the request.

1. **Sort intervals in the increasing order of their start time, breaking ties arbitrarily;**
2. **Let I1, I2, …, In denote the intervals in this order;**
3. **d=0;**
4. **for j = from 1 to n do**
5. **foreach Ii that precedes Ij in sorted order**
6. **if they are not overlapped or overlapped but still can share resources of an allocated PM do**
7. **allocateIj to the PM hosting Ii**
8. **else**
9. **start a new PM;**
10. **d=d+1;**
11. **allocateIj to PM d;**
12. **endif**
13. **endforeach**
14. **endfor**

3. EXPERIMENT AND RESULT

Algorithms are tested using the cloud tool –Cloudsched. Cloudsched simulator generates different requests as follows: the total numbersof return (requests) can be randomly set; all requests follow Poisson arrival process and have exponential length distribution; Therefore to test the algorithm, it is executed five times and its average has been taken. Here test results are compared with executing the simulators with the duration 50,100,200 and 400 respectively. The results are showed in table 1 and the configuration settings is as showed in Figure 1.

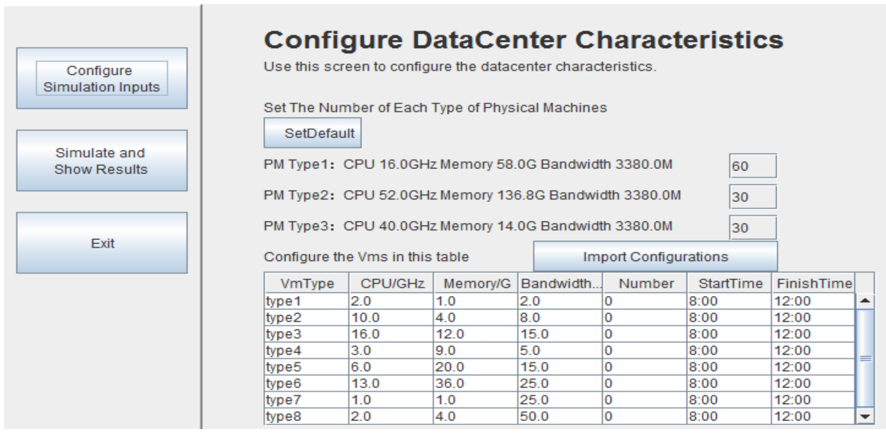

Figure 1: data set user for the experiment (Data Set 1)

The experiment is executed 5 times and average of total energy consumption is taken. These values are represented in table 1. The energy consumption is recorded at 50ms,100ms,200ms and 400 ms.

	Duration=50ms	Duration=100ms	Duration=200ms	Duration=400ms
RR	585	1104	2090	7005
OFWID	500	700	1489	4850
ONWID	520	800	1500	5410
MBFD	515	780	1480	5190

Table 1: Total energy consumption

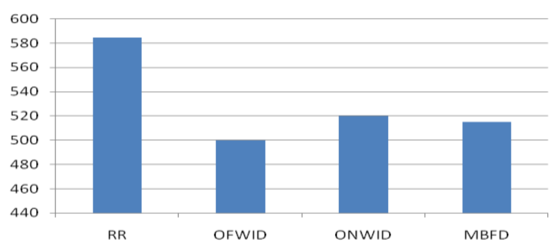

Figure 2: Energy consumption (in kilowatt hours) of Dataset 1and duration 50 ms

The Figure 2 shows that OFWID algorithm consumes Minimum Energy when the duration time is 50Ms. Hence the OFWID is energy efficient algorithm.

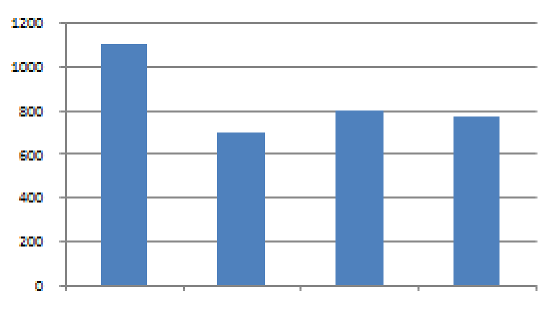

Figure 3: Energy consumption (in kilowatt hours) of Dataset 1and duration 50 ms

The Figure 3 shows that OFWID algorithm consumes minimum amount of energy when the duration time is 100Ms. Hence the OFWID is energy efficient algorithm when the duration time is 100Ms.

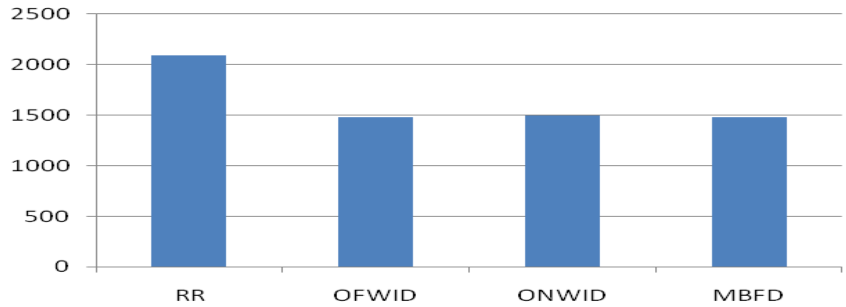

Figure 4: Energy consumption (in kilowatt hours) of Dataset 1and duration 50 ms

Figure 4 shows that OFWID algorithm consumes low amount of energy when the duration time is 200Ms.

Figure 5: Energy consumption (in kilowatt hours) of Dataset 1and duration 50 ms

Figure 5: shows that still OFWID algorithm consumes low amount of energy when the duration time is 400Ms.

Fig5 shows the average output of the algorithms when the duration is 50Ms, 100Ms, 200Ms, 400Ms.

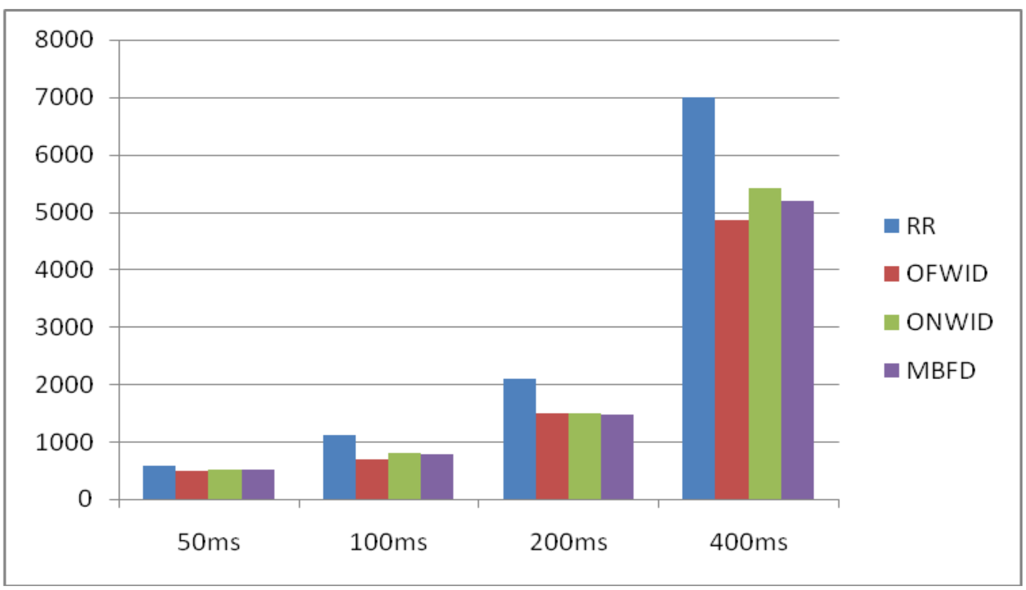

Figure 6 Energy consumption (in kilowatt hours) of 50ms,100 ms,200ms and 400 ms

Hence the Figure 6 itself indicates that OFWID is minimum energy consumption algorithm. When we compare all the algorithms experimental results shows that OFWID algorithm consumes less amount of energy for all the durations mentioned above.

4. CONCLUSION

One of the important requirements of a dynamic resource scheduler is to minimize energy consumption of the datacenter. Simulation results shows that OFWID algorithm has the minimum energy consumption in compare with the other scheduling algorithms like RR,ONWID and MBFD.

5. REFERENCES

[1] Armbrust M, Fox A, Griffith R, Joseph A, Katz R, Konwinski A, et al. Above theclouds: a Berkeley view of cloud computing. Technical Report No. UCB/EECS-2009. University of California at Berkley, CA; February 10, 2009.

[2] L. Andre et al., the Datacenter as a Computer: An Introduction to the Design of Warehouse-ScaleMachines, Ebook, 2009

[3] M. Armbrust et al., Above the Coulds: A Berkeley View of Cloud Computing, Technical Report,2009.

[4] Beloglazov, A., Buyya, R., Lee, Y. C., and Zomaya, A. Y., A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems, Advances in Computers, vol. 82, pp. 47-111, M. Zelkowitz(editor), Elsevier, Amsterdam, The Netherlands, 2011.

[5] Jing, S., Ali, S., She, K., Zhong, Y., State-of-the-art Research Study for Green Cloud Computing. Journal of Supercomputing, Special Issue on Cloud Computing, 2011.

[6] A. Beloglazov, J. H. Abawajy, R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing", Future Generation Computer Systems, Volume 28(5), pp755-768, 2012.

[7] Lee, Y. C., Zomaya, A. Y., Energy Efficient Utilization of Resources in Cloud Computing Systems, Journal of Supercomputing, vol. 60, no. 2, pp. 268-280, 2012.

[8] W. Tian, C. S. Yeo, R. Xue and Y. Zhong, "Power-aware scheduling of real-time virtual machines in cloud data centers considering fixed processing intervals," 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, Hangzhou, 2012, pp. 269-273.doi: 10.1109/CCIS.2012.6664410.