International Research Journal of Human Resource and Social Sciences
ISSN(O): (2349-4085) ISSN(P): (2394-4218)
Impact Factor 6.924 Volume 9, Issue 09, September 2022
Website- www.aarf.asia, Email : editoraarf@gmail.com

A STUDY OF VIRTUAL MACHINE AND DATA CENTERSIN CLOUD COMPUTING ENVIRONMENT

K.SHANMUGAM
RESEARCH SCHOLAR SUNRISE UNIVERSITY ALWAR
DR. PARVIN PRAKASH ADIVAREKAR
ASSOCIATE PROFESSOR SCHOLAR SUNRISE UNIVERSITY ALWAR

ABSTRACT

The placement component is designed to be both traffic aware and load aware. During non-rush hours (low traffic), as the number of requests is minimum, the VMP-LR uses a simple enhanced round robin method for placing VMs to PMs. During rush hours (heavy traffic), the three queues created are handled using three separate hybrid scheduling and load balancing algorithms in order to perform placement operation efficiently with accommodating high resource demands.The requests in high request queue are placed using an enhanced Max-Min, Ant Colony Optimization and Artificial Bee Colony. The medium request queue mapping is performed using an algorithm that combines First Fit, Best Fit and multi-level grouping genetic algorithm. Finally, the low request queue is handled using an algorithm that combines Enhanced Max-Min with enhanced Particle Swarm Optimization Algorithm.As the consequence of more and more virtual machines is packed onto a physical machine, the load imbalance factor increases, leading to the degradation of the performance of the cloud system. In order to solve this issue, the load monitoring component is used. The load monitoring component uses a load rebalancing algorithm to efficiently maintain load among VMs and PMs after placement. For this purpose, a hybrid algorithm that combines ant colony optimization with artificial bee colony algorithm is used.The proposed algorithms are implemented using Cloud Sim Simulator and evaluated using seven performance metrics. They are throughput, response time, SLA violation rate, resource utilization rate, power usage, load imbalance rate and migration rate. The

experimental results prove that the proposed VMP-LR is efficient in mapping VMs to PMs effectively in terms of cloud service response time and can save energy and increase resource utilization in a positive manner.

KEYWORDS: Virtual Machine, Data Centers, Cloud Computing Environment, traffic aware, Swarm Optimization Algorithm

INTRODUCTION

The application performance must not dig up negatively exaggerated as a result of the additional applications running on the identical hardware physical configuration. (Diwakar et al., 2007) explained performance separation as the situation in which application performance remains the same despite of kind, quantity of the process load of remaining applications giving out the identical set of computing sources. Performance separation is a significant objective in several collective hosting environments like virtualized computing environment. This can be justified with the example described in the preceding section; processor capacity allotted to the application has a key force on the performance of the application. Performance separation can be realized with suitable computing source allotment method between the contending virtual machines. To come to a decision on the resource shares for an application, we have to recognize how the computing resource scheduling procedure functions during Virtual Machine Monitor's. A Virtual Machine Monitor assigns distribution of resources like processor, RAM to every virtual machine. Consider an example; Processor scheduler in XEN called credit scheduler receives two arguments cap and weight related to every virtual-machine. cap indicates upper limit on processor consumption by a virtual machine, whereas weight describes qualified allocation of a virtual machine. Rate of cap places boundary on processor consumption through virtual computing machine. If summation of all cap values belongs virtual machines implementing on given processor are less than processor capacity subsequently processor residue in inoperative even if some executable process present in the virtualization system. Application performance placed in Virtual Machine is susceptible to the cap or weight set to the domain where the program (application) is executing. Conversely, accurate association between the value of the cap or weight with reference to domain, and the application performance measured values like throughput or response time is unclear. Hence, finding the suitable arguments that would

make available a definite Quality of Service in favor of program is a complicated issue. To put together belongings poorer, we have numerous bases of dynamics that craft the job of bringing Quality of Service to the programs placed in virtual computing machines. Consider an example the dynamic character of the workload, or varying client SLAs. Adding new customers and removing existing customers also a constant procedure. There is situation in the company of essential physical sources that regularly scaled up or improved by way of fresh hardware and software workings. Through exist these dynamic events, the correct association between application / program performance and the quantity of computing sources assigned to the program is unclear and is not static. As of this situation it is deduced that performance separation be capable of realized by supervising the executing system and fine-tuning the suitable configuration parameter values at dynamic instance.

VIRTUAL MACHINE AND DATA CENTERS

The job of predicting and continuing the system efficiency and performing capacity forecast is fetching complicated outstanding to augmented convolution in the Information Technology applications and communications. Service providers host programs/applications from diverse enterprise customers on a public collection of computing sources in data center environments. Customers agree a bond in the shape of a Service level agreement through service providers that comprise a narration of performance assurance. The performance assurance might comprise Quality of Service necessities like preferred time of response or amount of work done by the application. Dishonoured show result into Service level agreement breach that yet again results in fine in favor of the service providers. The outcome is frustrated clients and eventually fallout in monetary defeat for the service providers. Excess prerequisite of computing resources has all the time is the simplest selection on behalf of service providers to keep away from those performance issues. Nevertheless this action results into inefficient resource management and costlier infrastructure set up. The system need perform the resource allocation dynamically among applications so that they reuse shared resources more efficiently. One attractive condition happens once there are no pre-mentioned preferred values of performance measured arguments. The customers may not state the preferred value; as alternative customers need the take full advantage of performance at low price. Let us see an case, response- time of an application decline with raise in processor capacity through precise rate for a quantity of capacity and rate

begins to fall after certain processor capacity. Consequently make use of more processor does not offer performance at the similar speed, therefore the price to profit ratio enlarge. Let us illustrate this situation with an example. We performed an experiment with an application deployed in an Operating System running in a virtual machine. CPU capacity allocated to OS is varied and its effect on response-time and throughput of application is observed. The experiment was carried out on dual core processor; hence maximum CPU capacity can be 200%. The graph in the Figure 1.4 shows the experiment results. The values of CPU capacity, response time and throughput are plotted on Y-axis with the iterations plotted on X-axis. It can be observed from the graph that the response time of the application decreases by around 15 msec per 10% of CPU capacity in the capacity range of 10-90%. This rate of decrease drops down to 1msec per 10% above the CPU capacity of 120%. So increase in CPU capacity beyond 90% does not improve the response time by a significant amount. In case of throughput, it increases by 8 req/sec per 10% CPU capacity in the CPU capacity range of 10-110%. After capacity of 120%, rate of increase drops down to 1% req/sec per 10% of CPU. Hence it is clear that after certain value of CPU capacity, increasing CPU is costly for the client, as the same amount of benefit yield per CPU capacity is not there. An example of ratio of client interest can be given as: $\frac{Throughput}{Response\ \ Time + Allocated\ \ CPU\ Time}$. Clients want to operate their applications just before the point where this ratio starts decreasing rapidly. Hence in above example, the best region to operate the application is around 90-120% of CPU capacity. This region gives the best possible response time & throughput values at minimal possible CPU capacity. This ratio specified above is in its simplest form, and can include coefficients representing the relative weights of each of the metrics.

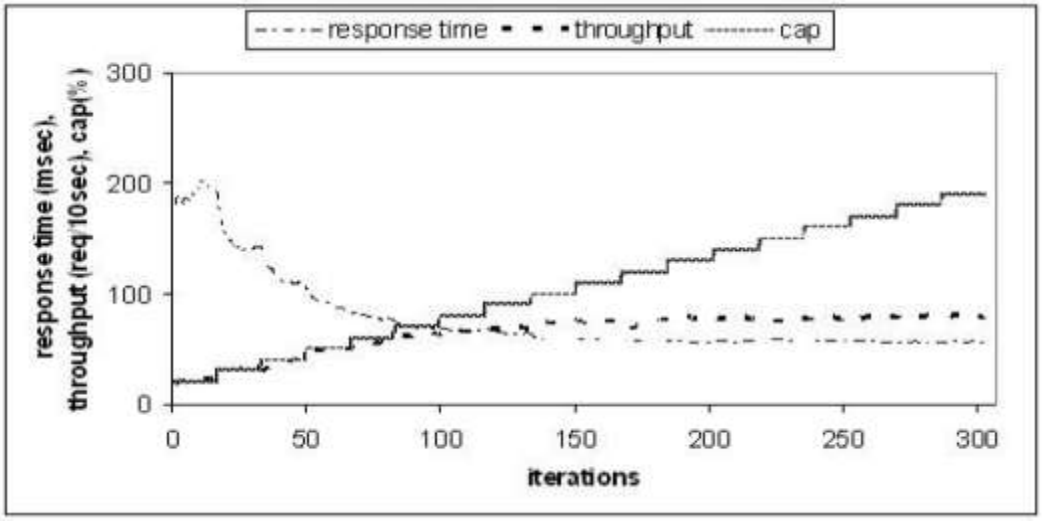

Figure 1. Optimal cost for the application

PERFORMANCE ISOLATION AND APPLICATION QOS

The concept of Resource sharing in cloud computing environment can guarantee significant cost savings, this is because the reduced per customer overheads and fiscal level. The most considerable barrier for potential cloud users, as well data isolation and security aspects, is undependable performance. The National Institute of Standards and Technology called as NIST identifies three service paradigms for cloud computing. The Infrastructure as a Service designated as IaaS paradigm influence virtualization to split physical computing resources among customers. The Platform as a Service designated as PaaS paradigm places applications of diverse customers within one middle-ware imagination. Software as a Service designated as SaaS is the final paradigm that offers a prepared located application. Separating cloud consumers during requisites of the performance they observe is a significant worry in these situations. Performance isolation is central issue for a variety of stake holders includes service providers and customers. Once a software practitioner or designer has to build up a method that guarantee performance isolation connecting consumers they entitle to authenticate the usefulness of the method to make sure the eminence of the creation. In addition, to progress on hand method they require separation metrics to contrast dissimilar alternatives of the response. The predictable questions like the disconnection of hardware and software workings on various nodes are of substance or not arise once a proprietor has to make a decision for one exacting deployment in a virtual computing environment. As a result ensuring performance guarantees is a principal

research subject in domain of cloud computing. Inside a virtualized computing environment, many software systems are hosted mutually on particular common stage. Every software system may belong to divergent proprietor. For each server, the Quality-of-Service necessities are articulated by customer by Service level agreement. Service level agreement defiance encompasses pre described punishment connected with customers. Quality of Service cross-talk take place in a condition when uphold service quality for some customers outcome into dishonored service quality for another customer. Performance assurance for the programs executing within the virtual computing machines be satisfied merely if here exists performance isolation transversely several virtual computing machines.

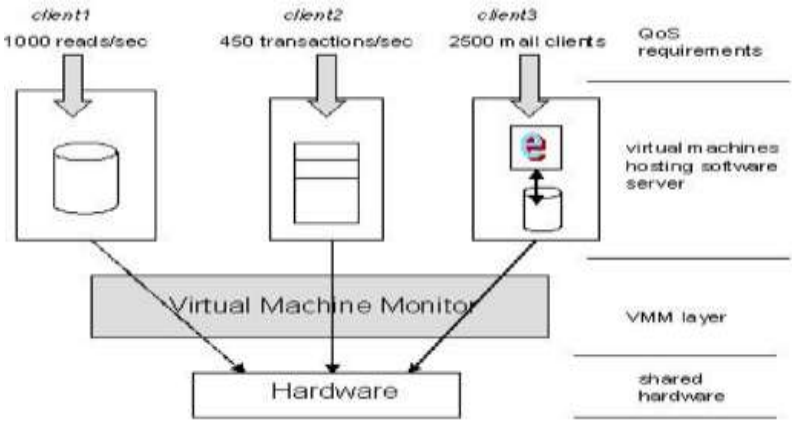

Figure 2 Virtualized computing environment and applications running inside environment

As shown in the Figure, the client 1 have expressed QoS requirements in terms of throughput requirements, client 2 expressed QoS in terms of required transaction rate and client 3 expressed QoS in terms of required mail client capacity. Every client application is deployed in different virtual machines all of which are running on same shared set of hardware. The job of the service provider is preserving performance such that SLA of customer cannot be dishonored. SLA contravention has pre described fine associated with them. Quality of service crosstalk arises in a state when preserving Quality of service for some client results into degraded Quality of service for another client. Performance promise for the applications running within the virtual machines can be satisfied merely if here exist performance isolation transversely virtual computing machines. Performance Isolation as described by is as follows:" Consumption of computing sources by any of the virtual computing machines must not influence the assured performance

guarantees to all other virtual computing machines running on the identical physical components". A system can be defined as performance isolated, if for customers running applications contained by their proportion the performance is not distressed when supplementary customers go beyond their proportion. A reduced performance meant for customers working surrounded by their portion are acceptable as long as portion is within corresponding customer SLA.Provisioning of resources in excess is straightforward answer to realize performance isolation nevertheless it vanish complete spirit of exercising virtualization. The definitive plan is in fact to boost the advantage of service provider through enhanced deployment of computing sources with restriction of carrying Quality of Service for each of the customer. For this reason enhanced solution is necessary.

Table 1 Effect on Mixed Load in Evaluating the Performance of Applications in Virtualized Computing Environment

Statistics of webserver running in virtualized environment						
	Weight	CAP	Load	CPU usage	Requests per sec	Transfer rate (Kbytes per sec)
Experiment1: With Web Server running						
Domain0	256	400	-	-	NA	
VM2	256	400	-	-	NA	
VM3	256	400	Web Server	180	797.61	1035.17
Experiment1: Mixed Load 1 VM CPU Load,1 With Web Server running						
Domain0	256	400	-	-	NA	
VM2	256	400	CPU	100	NA	

VM3	256	400	Web Server	180		

Consider an example which depicts following problem. During prior research it is known for us about conduct of the programs executing in the interior of the virtual computing machines residue as volatile when there is Input Output weight placed on at least one virtual computing machine. The experiments were completed to investigate the effect of mixed-load applications on performance of every one. In this work, one application is processor concentrated application and extra application is I/O intensive application deployed on a webserver. We carried out test 1 with only the webserver operation in virtual machine VM3. The test 2 was carried out with processor concentrated application implementing in VM2, webserver is running on VM3.

In both the experiments we set equal weights for all the virtual machines. We have performed the experiment without setting the value of cap for any virtual machine which means there is no upper limit on processor consumption by any virtual machine. As described in the table 1, in either case, processor consumption by Virtual Machine VM3 is identical which 180-percent is whereas in second test Virtual Machine VM2 consumed 100 percent processor. Test bed contains of four cores of processor; therefore we have still some processor capacity left. However, if we note the throughput readings, there is drastic change in the webserver throughput in the 2nd experiment. Even though processor exploitation is similar in two trials, the service quality carried has been exaggerated by means of the existence of the extra virtual computing machine. The test depicted above was prepared with a straightforward arrangement. When we see as a practical picture, the position can get poorer in the company of number of virtual computing machines using the collection of resources. Each one of the virtual computing machines may be placing dissimilar variety programs with diverse sort of process load prototype and having unusual stages of most wanted service quality. Modification of software process like application characteristic or the virtual computing machine, or variation physical configuration source capable of influence performance unfavourably. Consequently, the resource allocation for the virtual machines should not be done statically.

CONCLUSION

The VMP-LR has three main components to perform VM placement. They are, Resource Request Handling Component, Placement Component and Load Monitoring Component. VMP-LR is designed using a two-phase methodology, where the first phase handles the tasks involved with Resource Request Handling and Placement Components, while Phase II handles the tasks of Load Monitoring Component. The resource request handling component in Phase I of the research work, manages the multi-dimensional resource requests from users in the form of VM requests. VMP-LR considers three resources, namely, CPU, RAM and Bandwidth as relevant server resources in the context of VM placement, load balancing and rebalancing and the requests are structured as 3-dimensional cubic vector. This component interacts with the resource repository and VM manager of the cloud system along with the other two components of the VMPLR to aid in improving the process of VM placement. The main aim of the request handling component is to group the request into three queues, namely, high resource request queue, medium resource request queue and low resource request queue. This grouping is performed using a simple rule-based algorithm that is based on the current load and resource availability. The join shortest queue algorithm is used to attach new incoming requests to appropriate queue when the queues are being processed by VMP-LR. The placement component in Phase I of the research work performs the actual mapping of VMs to PMs. Four enhanced hybrid VM placement algorithms are proposed, and designed to be both traffic sensitive and load aware. The four algorithms are matched to effectively handle the heavy (rush hour) requests and light (non-rush hour) requests. During rush hours (heavy traffic), in order to accommodate high resource demands, the three queues created are handled using three separate hybrid scheduling and load balancing algorithms in order to perform placement operation efficiently. The three proposed algorithms are designed to handle requests of any one of the three queues.

REFERENCES

1. Beloglazov, A., Abawajy, J. and Buyya, R. (2012) Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing, Future Gener. Comput. Syst., Vol. 28, No. 5, Pp. 755-768

2. Beloglazov, A., Buyya, R., Lee, Y.C. and Zomaya, A. (2011) A taxonomy and survey of energy-efficient data centers and cloud computing systems, Advances in computers, Vol. 82, No. 2, Pp.47-111.

3. Beni, G. and Wang, J. (1989) Swarm intelligence in cellular robotic systems, NATO Advanced Workshop on Robots and Biological Systems, Il Ciocco, Tuscany, Italy.

4. Bichler, M., Setzer, T. and Speitkamp, B. (2006) Capacity planning for virtualized servers, Workshop on Information Technologies and Systems, Pp. 1-7.

5. Bing, W., Chuang, L. and Xiangzhen, K. (2011) Energy optimized modeling for live migration in virtual data center, Book Energy optimized modeling for live migration in virtual data center, 2011 Edition, Pp. 2311-2315.

6. Bitam, S. (2012) Bees life algorithm for job scheduling in cloud computing, Conf. on Computing and Information Technology, Pp. 186-191.

7. Blackburn, M. (Ed.) (2010) The Green Grid Data Center Compute Efficiency Metric :DCcE, White Paper #34, The Green Grid Publications, Pp. 1-15.

8. Bobroff, N., Kochut, A. and Beaty, K. (2007) Dynamic placement of virtual machines for managing sla violations, 10th IFIP/IEEE International Symposium on Integrated Network Management. Pp. 119-128.

9. Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999) Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, Oxford.

10. Botran, L.T., Alanso, J.M. and Lozano, J.A. (2014) Auto-scaling Techniques for Elastic Applications in Cloud Environments, Journal of Grid Computing, pages 1-34.

11. Brugger, B., Doerner, K., Hartl, R. and Reimann, M. (2004) Antpacking-an ant colony optimization approach for the one-dimensional bin packing problem, International Conference on Evolutionary Computation in Combinatorial Optimization, Pp.41-50.

12. Buyya, R., Beloglazov, A. and Abawajy, J. (2010) Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. arXiv preprint arXiv:1006.0308.

13. Buyya, R., Vecchiola, C. and Selvi, S.T. (2013) Mastering Cloud Computing Foundations and Applications Programming, Tata McGraw Hill Education Private Limited, New Delhi.

14. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I. (2009) Cloud computing and emerging IT platforms : Vision. Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems, Vol. 25, No. 6, Pp. 599-616.