Allocating memory dynamically for resource in cloud environment using gossip protocol

Iranna S Amargol, Prof. Vidya R Kulkarni

***Abstract*— We describe the process of allocating memory for cloud resources by using layer called cloud middleware. Gossip protocol which provides 1)performs memory allocation fairly 2) Fast computation and reallocation 3) Reduces processing memory .We describe the problem of allocating memory dynamically for resources which needs gossip protocol, for best utility of resources in cloud which takes less CPU time. management of resources is important part of data migration as the site or application allocated to one resources had reached their maximum threshold memory so to gain more performance and to get reduced processing memory we applying the concept of data migration which gives the fast computation and reallocation as the load of allocated resources changes dynamically**

***Index Terms*— Memory management, Cloud Middleware, CPU and Memory Constraints.**

I. INTRODUCTION

Now a day's technical world growing very fast it is very difficult to manage the resources in large scale cloud environment so as the elements of cloud which gives platform as service (PAS) by cloud service provider which give us platform as services to many users based on various payments and with contribution to it a Service provider also provide as a infrastructure as service(IAS) that let various users to use many infrastructures as services which is being hosted by service provider. This paper explains the new concept of allocating memory dynamically for resources using gossip protocol. cloud resource management is done by middleware and one of its elements called gossip protocol which contributes the performance of fair allocation, scalability, Adaptability. 1) Performance of fair allocation: contribution to dynamic resources management in cloud computing. our data migration within cloud resources gives better performance by considerations of memory constraints. 2) Scalability: we can achieve better performance with scalable both in data and in machines. 3) Adaptability: as the load changes to the allocated process we can manage dynamic resource allocation by dynamically adapting the changes by sites.

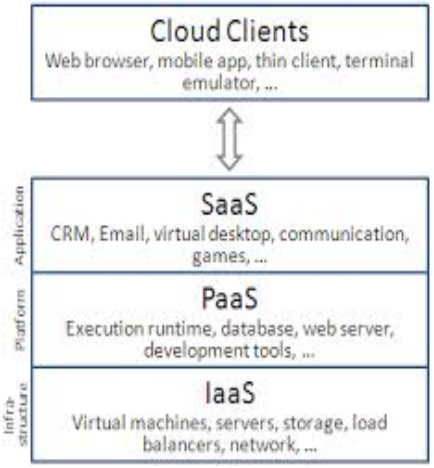
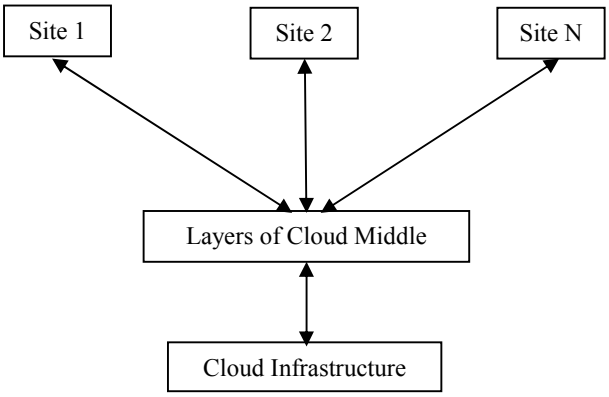


Figure 1: (a) In traction of clients with cloud services.



(b) Architecture of cloud.

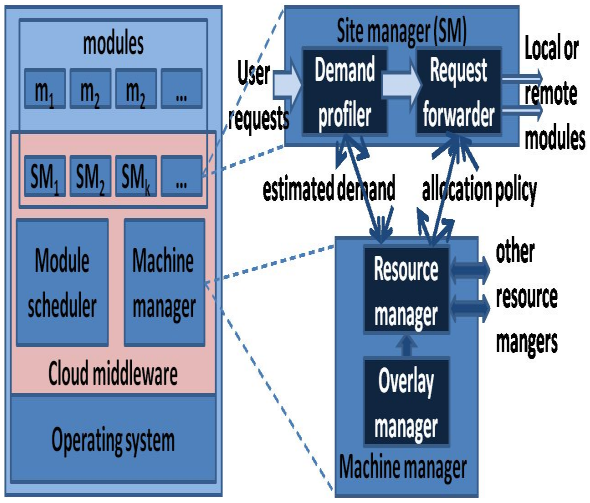


Figure 2: Cloud middleware architecture and its components

Manuscript received May 12, 2015

Iranna S Amargol, PG Student, Dept of Computer Science &Engineering, KLS Gogte Institute of Technology and Management, belagavi, Karnataka, India

Prof. Vidya R Kulkarni, Dept of Computer Science &Engineering, KLS Gogte Institute of Technology and Management, belagavi, Karnataka, India

II. LITERATURE SURVEY

Márk Jelasity [2]: Aggregation is a key functional building block for very large scale applications: it refers to a set of functions that provide components of a distributed system access to global information including network size, average load, and so on. A gossip-based protocol is proposed for computing aggregate values over network components in a fully decentralized fashion. The protocol is suitable for extremely large and highly dynamic systems due to its proactive structure---all nodes receive the aggregate value continuously to track any changes in the system. The protocol is extremely lightweight, and suitable for many distributed applications including peer-to-peer and grid computing systems. The efficiency and robustness of our gossip-based protocol is shown.

Fetahi Wuhib [3]: Gossip protocol used in the proposed system executes in a middleware platform. The protocol ensures three design goals namely fairness, adaptability and scalability. The protocol continuously executes while it's input and consequently its output dynamically changes. Global synchronization can be avoided, as there is a single continuous execution instead of a sequence of executions with restarts. The system can continuously adapt to changes in local input. Continuously executes and dynamically solves the problem of optimally placing applications in a cloud, achieving fair resource allocation.

R. Yanggratoke [1]: Formalization of dynamically optimizing a cloud configuration for green computing objectives under CPU and memory constraints. A generic protocol is used for resource allocation which aims at minimizing power consumption through server consolidation, while satisfying a changing load pattern. Under overload, the protocol gives a fair allocation of CPU resources to clients. The effectiveness of the protocol in achieving its objective increases with increasing memory capacity in the servers.

III. OBJECTIVE

The main objective of this technique is to reduce processing memory and to increase performance as compared to dynamic resources allocation in cloud environment [3]. The Idea behind this is to manage the resources dynamically by data migration , when compared with resources management by middleware in cloud which consist of gossip protocol for resources allocation here when the new process is sent is sent to cloud is allocated to machine based on its load, after allocation of resources for site and application the owner of site or application wants to modify the contents of site or application the same process that is site or application is modified with new load for allocation at this stage the allocated resources is reached its maximum threshold level of memory due to modified load by allocated process of site or application so we have to migrate the whole data to dynamically allocating new resources within cloud to the application or site such that the allocated resources should not reach its maximum threshold level as shown in fig 3. By dynamically calculating for allocation of resources by data migration reduces the processing memory and increases the availability of performance.

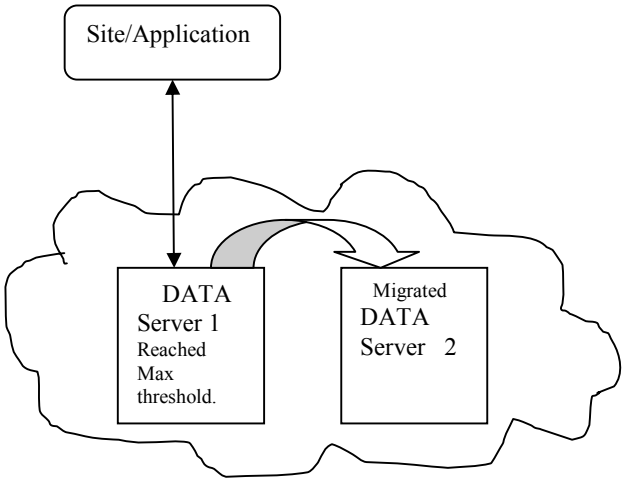


Figure 3: Data migration within cloud storage

IV. METHODOLOGY AND ALGORITHM FOR RESOURCE

We consider here the algorithm for data migration before that we are considering the algorithm for dynamic resources allocation for all sites.

Let set of sites be S such that each s∈ S which contains set of modules M.

Let set of machines be N such that n∈ N which can allocate sites to any machine.

Each site s∈ S composed of set of modules $_{Ms}$ and memory demand is given by $\boldsymbol{\gamma}$ = [γ1, γ2. . . γ|S|]T , the machine with CPU capacity is Ωn and memory capacity is Γn

Algorithm 1.1: protocol P* for resource allocation for sites s

1: Let S be the set of all sites or applications stored in array γs.
2: Let N be the set of all machines.
3: Let function freeMemNode(N) return a machine n ∈ N with the largest _ free memory, namely, Γn −∑s γs**sign**(αs,n).
4: αn,s = 0, ∀n,s
5: **for** i = 1 to |S| **do**
6: s = S[i]; n = freeMemNode(N)
7: **if** Γn −∑i∈S γi **sign**(αn,i) ≥ γs **then**
8: αn,s = 1.

Algorithm 1.2: protocol P** for data migration

1: Let T be the maximum threshold memory capacity of all machines n∈ N.
2: Let S be the set of all sites s∈ S.
3: Let N be the set of machines n∈ N.
4: Let N={n1,n2,……nn} be set of all machines.
5: αn1,s = 1
6: for i=1 to |N| do
7: n=N[i]
8: if αn1,s > Maximum threshold capacity of memory (T) of n1.
9: αn1,s = 0.
10: Let function freeMemNode(N) return a machine n ∈ N with the largest_free memory, namely, Γn −∑s γs **sign**(αs,ni).
11: Γn −∑s∈S γs **sign**(αni,s) ≥ γs then
12: αni,s = 1.

V. EXPECTED RESULT

So far what we gained performance for dynamic resources allocation for process or sites is better when we compared it to the static resources allocation for process or sites by using virtual machine or simulator but when we compare same with our technique that is data migration in within cloud server gives better performance than all previous technique because when we modify the contents of allocated sites or process it reaches the maximum threshold level of the allocated resources so because of this the performance is degraded and processing memory is not reduced and reallocation of resources is not possible so our technique of data migration within cloud gives better performance and reallocation can be done easily and processing is reduced

CONCLUSION

As the cloud technology added a greater advantage to the new computing utility techniques as its added here also out new technique added more advantage to the dynamic resources management in cloud. Before concluding this paper we have to see that we had got lot to do with this new technique by the improvement of performance in cloud services and which gives more security for data which is being taken by memory management. Our memory management concept by gossip protocol gives a scalability, adaptability and fairness of resources of resources allocation as load of sites or application changes dynamically. The results by simulation or by using virtual machines we can even prove that this method is better which gives more performance and processing memory is reduced which also gives fast computation and reallocation of resources as the modified load reaches the maximum threshold memory of allocated resources.

REFERENCES

[1] R. Yanggratoke, F. Wuhib, and R. Stadler, "Gossip-based resource allocation for green computing in large clouds," in 2011 International Conference on Network and Service Management.

[2] Márk Jelasity" Gossip based aggregation in large dynamic networks" .

[3] Fetahi Wuhib, Rolf Stadler, and Mike Spreitzer " A Gossip Protocol for Dynamic Resource Management in Large Cloud Environments".

[4] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," ACM Trans. Computer Syst., vol. 23, no. 3, pp. 219–252, 2005

[5] A. Verma, G. Dasgupta, T. K. Nayak, P. De, and R. Kothari,"Server workload analysis for power minimization using consolidation," inUSENIX'09. Berkeley, CA, USA: USENIXAssociation, 2009, pp. 28–28.

[6] F. Wuhib, R. Stadler, and M. Spreitzer, "Gossip-based resourcemanagement for cloud environments," in CNSM 2010 , October, pp. 1 –8.

[7] F. Wuhib, R. Stadler, and M. Spreitzer, "Gossip-based resourcemanagement for cloud environments," in CNSM 2010 , October, pp. 1 –8.

[8] Z. Gong, X. Gu, and J. Wilkes, "PRESS: PRedictive Elastic ReSource Scaling for cloud systems," in CNSM 2010 , October, pp. 9 –16.

[9] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: elimi- nating server idle power," SIGPLAN Not. , vol. 44, pp. 205–216, March 2009.

[10] Hewlett-Packard, Intel, Microsoft, Phoenix Technologies Ltd., Toshiba Corporations, "Advanced configuration and power in- terface specification," 2010.

[11] D. Carrera, M. Steinder, I. Whalley, J. Torres, and E. Ayguade, "Utility-based placement of dynamic web applications with fair- ness goals," in IEEE NOMS , April 2008, pp. 9 –16.

[12] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A scalable application placement controller for enterprise data centers," in WWW2007 . New York, NY, USA: ACM, 2007, pp. 331–340.

[13] R. Yanggratoke, F. Wuhib, and R. Stadler, "Gossip-based resource allocation for green computing in large clouds (long version)," KTH Royal Institute of Technology, https://eeweb01.ee.kth.se/upload/publications/reports/2011/ TRITA-EE 2011 036.pdf, Tech. Rep. TRITA-EE 2011:036, April 2011.

[14] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in INFOCOM , vol. 1, 1999, pp. 126 –134.