



A Dynamic Resources Allocation by Data Migration within Cloud Resources by Gossip Protocol in Cloud Environments

Murgesh V Jambigi, S G Maknur, Arun Kumar G, C M Parameshwarappa, Srinivasa Rao Udara

PG Student, Dept of CS&E, STJIT, Ranebennur, Karnataka, India.

Professor, Dept of IS&E, STJIT, Ranebennur, Karnataka, India.

Assistant Professor, Dept of E&CE, STJIT, Ranebennur, Karnataka, India.

Professor, Dept of CS&E, STJIT, Ranebennur, Karnataka, India.

Assistant Professor, Dept of E&CE, STJIT, Ranebennur, Karnataka, India.

ABSTRACT: We describe the process of data migration within cloud resources by including cloud middleware component Gossip protocol which provides 1) Fair memory allocation 2) Fast computation and reallocation 3) Reduces processing memory. We describe the problem of data migration in cloud under CPU and Memory constraints 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.

KEYWORDS: Data Migration, Cloud Middleware, CPU and Memory Constraints

I. INTRODUCTION

In every day fast growing technical world 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 is shown in figure 1.a. This paper explains the new concept of dynamic data migration within cloud resources as 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 [3] 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.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

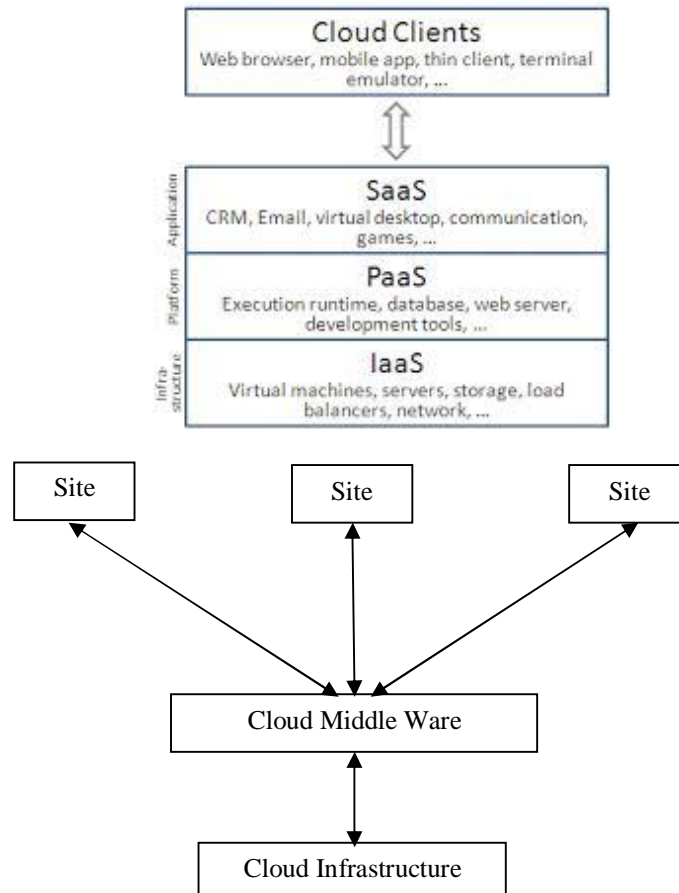


Figure 1: (a) In traction of clients with cloud services. (b) Architecture of cloud.

So far as recent technologies cloud computing have emerged a good performances for services namely SAS,PAS and IAS and many companies have been providing a many of these services through their own cloud to many clients so it's one of the cost effective technology in the current technical market. This cloud provides a beautiful architecture in which all sites in cloud are allocated resources based of the infrastructure available in the particular cloud with the help of cloud middle ware which allocates resources for all sites in a cloud as shown in figure 1(b). So as far this technology of cloud computing is executed by many companies it reduces the cost of using computing utilities and whenever the client can access these utilities they have to be paid for that particular usable period and we are concentrating here on the allocation of resources to the client in an efficient way dynamically. As there are various ways to dynamically allocating resources for a sites or application one such new method is executed in this paper that is data migration for efficient resources allocation for site or process.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

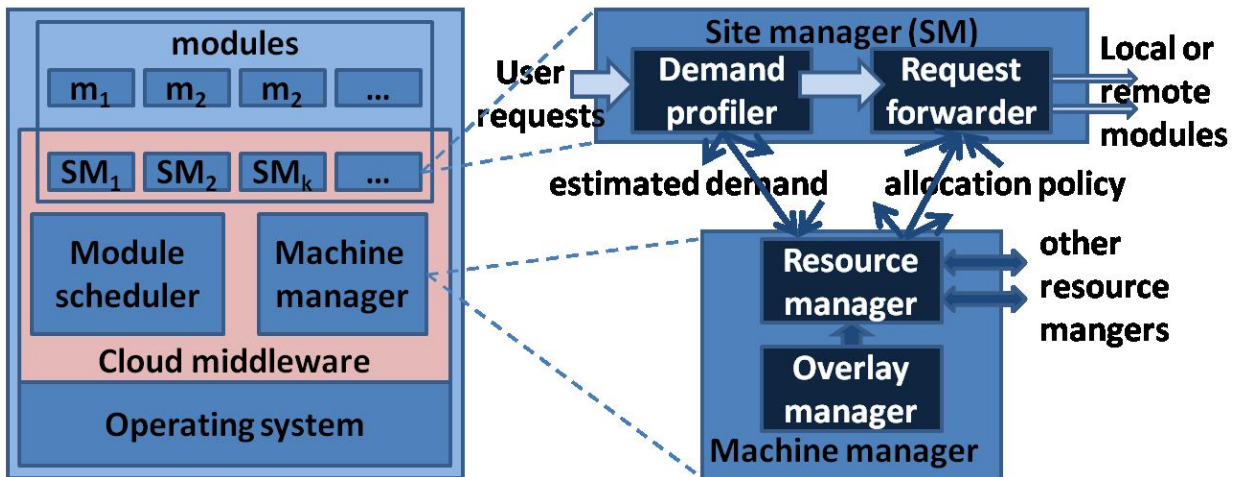


Figure 2: Cloud middleware architecture and its components [3].

The main technique behind dynamic allocation of resources by data migration lies in the middle ware of cloud architecture and its components helps in allocation of resources dynamically. The cloud middle ware consists of two components that are module scheduler and machine manager ,modules component consists of many modules as shown in fig 2(left side) such as m_1, m_2, \dots, m_n in which each module m_i runs a part of service logic or its instances of application or sites.

The figure 2(right side) shows the components of machine manager that is resource manager which takes a estimated demand from the demand profiler which is a component of site manager(SM) and same management of resources is done which decides which instance of module to be run this is calculated by gossip protocol P^* . The allocation policy is sent to both module scheduler which schedules the modules to be run and request forwarder which forward the request by site manager. The overlay manager which interacts resource manager which tells that the set of machines to interact with modules. Each modules is associated with their site manager which consists of demand profiler which is interacted with the resources manager which gives estimated demand of each module for its resources allocation and based on the allocation policy the request is forwarded. This middleware architecture in figure 2 and its components is important for allocation of resources as the load of the site or application changes dynamically but this concept of dynamic allocation of resources does not give the answer when the allocated resources reaches its maximum threshold memory so in way to introduce the data migration concept to get rid of this problem.

II. LITERATURE SURVEY

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.

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.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

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.

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.

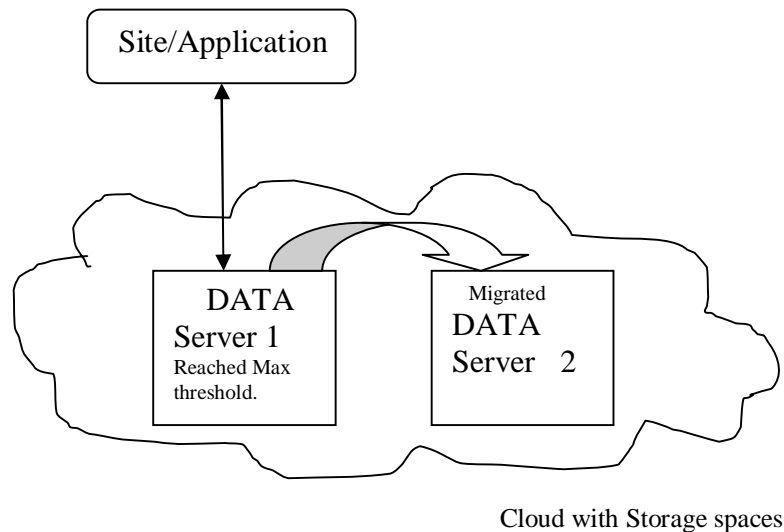


Figure 3: Data migration within cloud storage

IV. METHODOLOGY AND ALGORITHM FOR DATA MIGRATION

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 $s \in S$ which contains set of modules M .

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

Each site $s \in S$ composed of set of modules M_s and memory demand is given by $\gamma = [\gamma_1, \gamma_2, \dots, \gamma_{|S|}]^T$, the machine with CPU capacity is Ωn and memory capacity is Γn

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

- 1: Let S be the set of all sites or applications stored in array γ_s .
- 2: Let N be the set of all machines.
- 3: Let function $\text{freeMemNode}(N)$ return a machine $n \in N$ with the largest free memory, namely, $\Gamma_n - \sum_s \gamma_s \text{sign}(\alpha_s, n)$.
- 4: $\alpha_{n,s} = 0, \forall n, s$
- 5: **for** $i = 1$ to $|S|$ **do**
- 6: $s = S[i]; n = \text{freeMemNode}(N)$
- 7: **if** $\Gamma_n - \sum_i \gamma_i \text{sign}(\alpha_{n,i}) \geq \gamma_s$ **then**
- 8: $\alpha_{n,s} = 1$.

Algorithm 1.2: protocol P** for data migration

- 1: Let T be the maximum threshold memory capacity of all machines $n \in N$.
- 2: Let S be the set of all sites $s \in S$.
- 3: Let N be the set of machines $n \in N$.
- 4: Let $N = \{n_1, n_2, \dots, n_n\}$ be set of all machines.
- 5: $\alpha_{n_1, s} = 1$
- 6: **for** $i=1$ to $|N|$ **do**
- 7: $n = N[i]$
- 8: **if** $\alpha_{n_1, s} > \text{Maximum threshold capacity of memory } (T) \text{ of } n_1$.
- 9: $\alpha_{n_1, s} = 0$.
- 10: Let function $\text{freeMemNode}(N)$ return a machine $n \in N$ with the largest free memory, namely, $\Gamma_n - \sum_s \gamma_s \text{sign}(\alpha_s, n_i)$.
- 11: $\Gamma_n - \sum_s \gamma_s \text{sign}(\alpha_{n_i, s}) \geq \gamma_s$ **then**
- 12: $\alpha_{n_i, s} = 1$.

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

VI. 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 data migration. Our data migration 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. Montessor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," ACM Trans. Computer Syst., vol. 23, no. 3, pp. 219–252, 2005.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

- [5] A. Verma, G. Dasgupta, T. K. Nayak, P. De, and R. Kothari, "Server workload analysis for power minimization using consolidation," in USENIX'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 28–28.
- [6] F. Wuhib, R. Stadler, and M. Spreitzer, "Gossip-based resource management for cloud environments," in CNSM 2010, October, pp. 1–8.
- [7] F. Wuhib, R. Stadler, and M. Spreitzer, "Gossip-based resource management 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: eliminating 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 interface specification," 2010.
- [11] D. Carrera, M. Steinder, I. Whalley, J. Torres, and E. Ayguade, "Utility-based placement of dynamic web applications with fairness 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.
- [15] V. Petrucci, O. Loques, and D. Mossé, "Dynamic optimization of power and performance for virtualized server clusters," in ACM SAC 2010, 2010, pp. 263–264.
- [16] G. Jung, M. Hiltunen, K. Joshi, R. Schlichting, and C. Pu, "Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures," in ICDCS2010, 2010, pp. 62–73.
- [17] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," IEEE TSC, vol. 3, no. 4, pp. 266–278, 2010.