ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

A Different Load Processing and Resource Provisioning Mechanism to Maintain Flash and Base Load In Cloud

M.Mithra[1], G.Thiagarajan[2], M.Jaffer[3]

ME Student, Dept. of Computer Science, CSI College of Engineering, Ketti, The Nilgiris, Tamil Nadu, India[1]

Asst. Professor, Dept. of Computer Science, CSI College of Engineering, Ketti, The Nilgiris, Tamil Nadu, India [2]

ME Student, Dept. of Computer Science, CSI College of Engineering, Ketti, The Nilgiris, Tamil Nadu, India[3]

ABSTRACT: Cloud computing allows business customers to scale up and down their resource usage based on needs. Many of the touted gains in the cloud model come from resource multiplexing through virtualization technology. In this paper, we present a system that uses virtualization technology to allocate data center resources dynamically based on application demands and support user specified constraints by optimizing the number of servers in use. This also considers various parameters for service selection. This introduces the concept of "Heuristic approach" to measure the unevenness in the multi-dimensional resource utilization of a server. By minimizing workload, we can combine different types of workloads nicely and improve the overall utilization of server resources. The system proposes a new algorithm named as **flow fairness algorithm.** We develop a set of heuristics that prevent overload in the system effectively while saving energy used. The system has been developed using Java Swing. The experiments have been developed to show the efficiency of the proposed system.

KEYWORDS: Virtual machine, Cloud, Load, flash crowd, Base crowd

I. INTRODUCTION

The elasticity and the lack of upfront capital investment offered by cloud computing is appealing to many businesses. There is a lot of discussion on the benefits and costs of the cloud model and on how to move legacy applications onto the cloud platform. Here we study a different problem: how can a cloud service provider best multiplex its virtual resources onto the physical hardware. This is important because much of the touted gains in the cloud model come from such multiplexing. Studies have found that servers in many existing data centers are often severely underutilized due to over provisioning for the peak demand. The cloud model is expected to make such practice unnecessary by offering automatic scale up and down in response to load variation. Besides reducing the hardware cost, it also saves on electricity which contributes to a significant portion of the operational expenses in large data centers. Virtual machine monitors (VMMs) like Xen provide a mechanism for mapping virtual machines (VMs) to physical resources. This mapping is largely hidden from the cloud users. Users with the Amazon EC2 service, for example, do not know where their VM instances run. It is up to the cloud provider to make sure the underlying physical machines (PMs) have sufficient resources to meet their needs. VM live migration technology makes it possible to change the mapping between VMs and PMs While applications are running. However, a policy issue remains as how to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized. This is challenging when the resource needs of VMs are heterogeneous due to the diverse set of applications they run and vary with time as the workloads grow and shrink.

II. EXISTING SYSTEM

The main drawback of the existing load balancing technique is time latency in accuracy. The cloud load could not be balanced due to unpredictable request of the clients. In existing system, they have used to develop the system using Round Robin model and Fluid flow model. Those models are not effective. Those models are not able to give the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

output in time and the thorough put also lesser than that their expected output. The models had made the Latency problem and minimal through put. Some existing designed an efficient truthful-in-expectation mechanism for resource allocation in clouds where only one type of resource was considered. Some others designed a stochastic mechanism to allocate resources among selfish VMs in a non-cooperative cloud environment. In such existing the author showed that system heterogeneity plays an important role in determining the dynamics of truthful mechanisms. the proposed mechanisms take into account the heterogeneity of the systems and that of user requests when making allocation decisions In their model, the objective of the SaaS is to maximize its revenue satisfying the service level agreement, while the objective of the IaaS is to maximize the profit by determining the spot instances price. However, both studies considered only one type of VM instances, thus, the problem they solved is a one dimensional provisioning problem.

III. LITERATURE SURVEY

Zhang, Yan, and Nayeem Ansari proposed in [1] major mechanisms involved in DRE techniques, including fingerprinting, cache management, chunk matching, and decoding error recovery, have been discussed. For each mechanism, different approaches have been reviewed. In [2] presents LBFS, a network file system designed for low-bandwidth networks. LBFS exploits similarities between files or versions of the same file to save bandwidth. It avoids sending data over the network when the same data can already be found in the server's file system or the client's cache. [3] Describes a new idea called extreme Binning, for scalable and parallel deduplication, which is especially suited for workloads consisting of individual files with low locality. Existing approaches which require locality to ensure reasonable throughput perform poorly with such a workload. Extreme Binning exploits file similarity instead of locality to make only one disk access for chunk lookup per file instead of per chunk, thus alleviating the disk bottleneck problem. Eshghi, Kave, and Hsiu Khuern Tang given a new algorithm, TTTD, which performs much better than all the existing algorithms, and also puts an absolute size limit on chunk sizes. Using this algorithm can lead to a real improvement in the performance of applications that use content based chunking.

In [5] Zhu, Benjamin, Kai Li, and R. Hugo Patterson proposed three techniques employed in the production Data Domain deduplication file system to relieve the disk bottleneck. These techniques include:
(1) the Summary Vector, a compact in-memory data structure for identifying new segments;
(2) Stream-Informed Segment Layout, a data layout method to improve on-disk locality for sequentially accessed segments; and
(3) Locality Preserved Caching, which maintains the locality of the fingerprints of duplicate segments to achieve high cache hit ratios.

[6] Provides an ADMAD: an Application-Driven Metadata Aware De-duplication Archival Storage System, which makes use of certain meta-data information of different levels in the I/O path to direct the file partitioning into more Meaningful data Chunks (MC) to maximally reduce the inter-file level duplications. Puzio, Pasquale, et al presented
[7] ClouDedup, a secure and efficient storage service which assures block-level deduplication and data confidentiality at the same time. Although based on convergent encryption, ClouDedup remains secure thanks to the definition of a component those implements an additional encryption operation and an access control mechanism.

[8] Is proposed by Yuan, Jiawei, and Shucheng Yu and they provide a Proof of Retrievability (POR) and Proof of Data Possession (PDP) techniques assure data integrity for cloud storage. Proof of Ownership (POW) improves storage efficiency by securely removing unnecessarily duplicated data on the storage server. However, trivial combination of the two techniques, in order to achieve both data integrity and storage efficiency, results in non-trivial duplication of metadata (i.e., authentication tags), which contradicts the objectives of POW.

[9] Proposed a cryptographic primitive called proofs of ownership (PoW) to enhance security of client-side deduplication in cloud storage. In a proof of ownership scheme, any owner of the same file F can prove to the cloud storage that he/she owns file F in a robust and efficient way, in the bounded leakage setting where a certain amount of efficiently-extractable information about file F is leaked. Zhang, Kehuan, et al proposed in [10] Sedic, leverages the special features of MapReduce to automatically partition a computing job according to the security levels of the data it

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

works on, and arrange the computation across a hybrid cloud. Specifically, we modified MapReduce's distributed file system to strategically replicate data, moving sanitized data blocks to the public cloud. Over this data placement, map tasks are carefully scheduled to outsource as much workload to the public cloud as possible, given sensitive data always stay on the private cloud.

IV. PROPOSED SYSTEM

For this problem they introduced the Heuristic Flow Equilibrium (Load Balancing) model is to overcome the existing problems. This research is going to implement FF (Flow Fairness) model in the proposed system. The existing approach created a NP hard and resource provisioning problem. In this section, this formally defines the VM selection problem with personal need consideration. The original VM balancing problem is to distribute the subscripting users to different VMs so that all the VMs are kept balanced in all the time. In practice, this problem has no optimal solution because the optimal solution requires the exact leaving time for each user. Such information is of the future and can never be obtained.

This addressed the problem of dynamic VM provisioning and allocation in clouds by designing truthful mechanisms that give incentives to the users to reveal their true valuations for their requested bundles of VM instances. The proposed truthful optimal and greedy mechanisms for solving the VMPAC problem consider the presence of resources of multiple types. We determined the approximation ratio of the proposed greedy mechanisms and investigated their properties by performing extensive experiments. The results showed that the proposed greedy mechanisms determine near optimal solutions while effectively capturing the dynamic market demand, provisioning the computing resources to match the demand, and generating high revenue. In addition, the execution time of the proposed greedy mechanisms is very small. As a recommendation, G-VMPAC-II is the best choice for the cloud providers since it yields the highest revenue among the proposed greedy mechanisms.

- The main goal of the proposed system is to schedule the jobs of sub servers with the aim of reducing the server load imbalance.
- And this aims to support user constraint based VM selection with different parameters.
- Prediction of demands helps to avoid unexpected problems.
- Creating a cloud environment with different virtual machine and making the load sharable by performance based job scheduling. The overall objective is to enhance the QOS in the cloud environment.
- Overload avoidance: the capacity of a VM should be sufficient to satisfy the resource needs of all VMs running on it. Otherwise, the VM is overloaded and can lead to degraded performance of its VMs.
- User requirements based VM selection helps to satisfy user needs
- Customization possible for dynamic workloads.
- Easy and effective

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

System Flow Diagram:

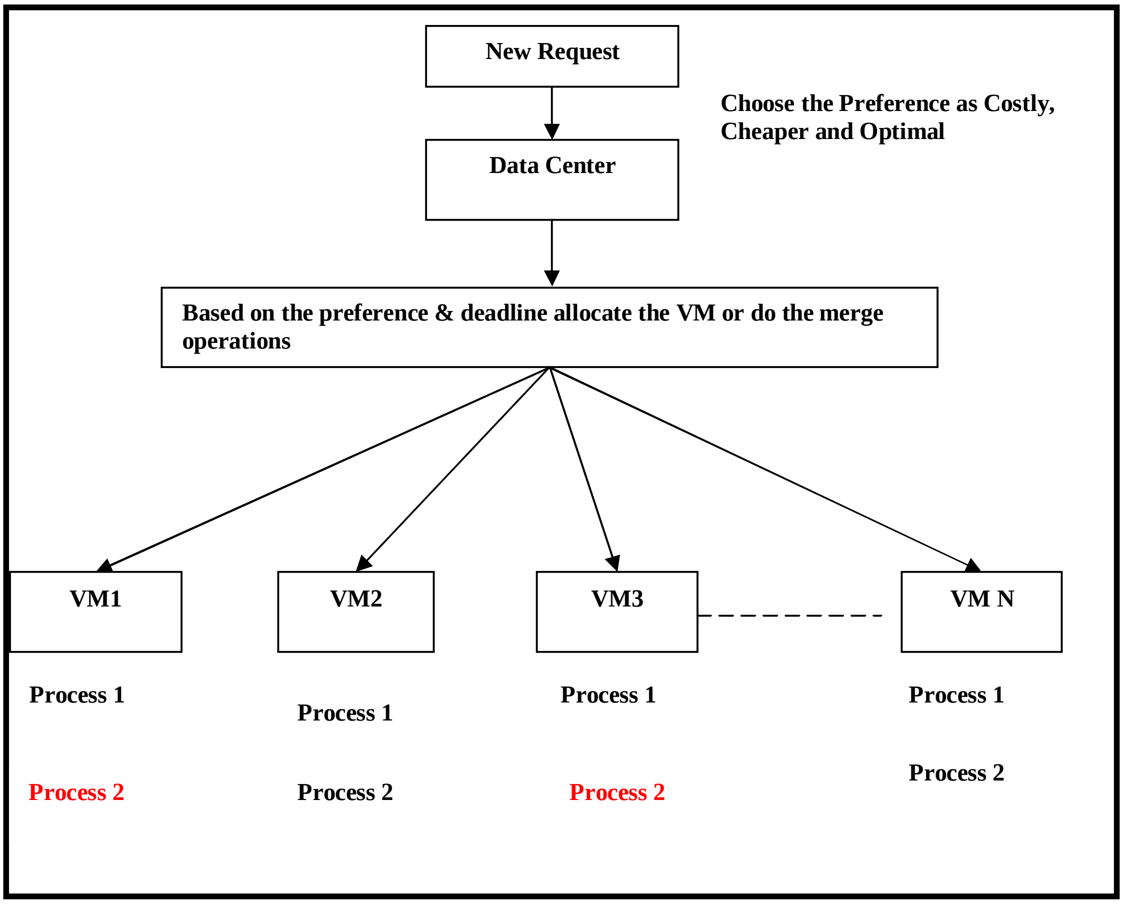

Fig 3.1: System Flow Diagram

The system flow diagram clearly explains the flow followed in the cloud request processing between virtual machines. Whenerver a new request received by the main server the requets will be forwarded to any virtual machines. The virtual machines are with different hardware and quality configurations so based on the user preference the service will be given. When the flash growd request received then the virual machines needs resources to handle the arriving requests that time mainserver provide the extra resources to handle the requestes.

Heuristic approach:
Heuristic planner has two heuristics to reduce the searching space. First of all, it uses main schemes and auxiliary schemes alternatively during the optimization process. Second of all, the planner uses the cost model to prune the “unpromising” transformations. Third, the planner is rule-based. The rule is defined to consist of two components: condition and action, where the condition is usually defined based on the cost and performance optimization goal (e.g., the estimated monetary cost can be reduced by 20 percent) and the action consists of transformations on the workflow.

Cost:
Our flow fairness algorithm does not assume any specific initial instance assignment. We present a number of heuristic based methods for initial instance assignment. If the priority is a costlier machine then the cost should be bear by the user (costlier in the sense like extra bandwidth, speed and memory etc.,).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Time:
Application Service period is an important tuning parameter in the VM schedule planner. If the period is long, more workflows are buffered in the queue. When the period parameter is short, the optimization space gets smaller and the chance for operating transformations to reduce cost decreases.

Optimal: (Quality)
To make the optimization plan, more combinations of tasks need to be checked for transformations and the optimization space becomes much larger.

Architecture Diagram:
Architecture diagram describes about the proposed system process. This architecture diagram shows how the requests are arrived from the user and the load predictor predict the number of request are arrived and find the user preferences specified in the requests after that the virtual machines which are ready to accept the request are assigned based on the preferences. If the surplus requests are received and virtual machines are in need of resources to process at that time main server allots the configurations to all the virtual machines.

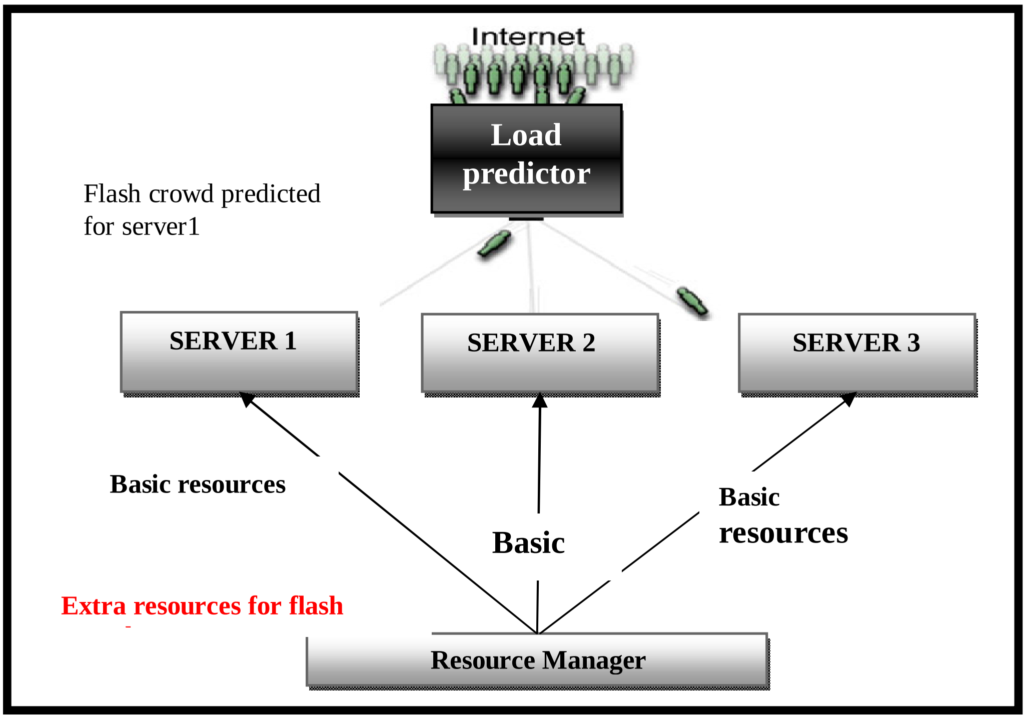

Fig 3.2: Overall Architecture Diagram

IV. EXPERIMENTAL RESULTS

Performance Comparison
The below chart shows the performance of the existing and proposed system. Different user request timings are analyzed and deployed as a chart. Using this chart everyone can analyze that the proposed system is better than the existing.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

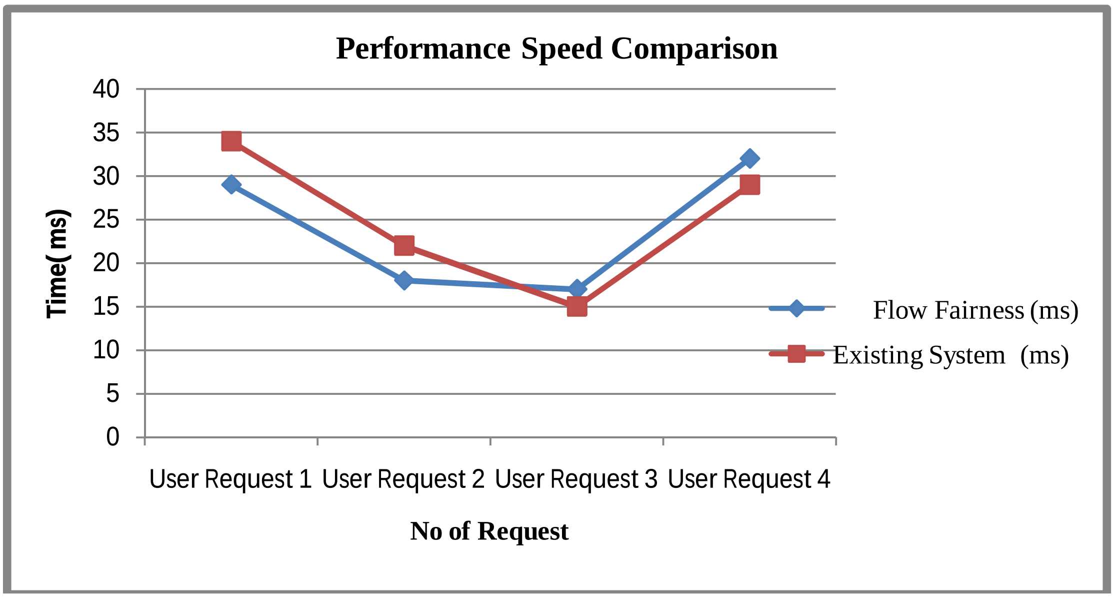

Fig 4.1: Speed Comparison Chart

V. CONCLUSION

In the world of intelligent workload management, there is a need to be able to deal with multiple environments, not just virtual and not just CDN. From a build perspective, that means we need to provide portability and reduced complexity. As part of the Workload solution, the proposed system reduced complexity and provides quick and easy transformation between virtual machines.

FUTURE ENHANCEMENT

To Enhance the current proposed system, the study utilizes a temporal R++tree as well as a Inverted key index and hash table for quickly identifying a candidate set of uncertain trajectories, by dynamically computing the rank for each point and service. This helps to satisfy a given query condition. This process does not affect by any uncertainty object queries and results. The process does not a part of information loss during probabilistic query processing.

V1. REFERENCES

[1] Windows Azure, http://www.windowsazure.com/en-us/pricing/ calculator/, 2014.
[2] Amazon EC2 Instance Types, http://aws.amazon.com/ec2/ instance-types/, 2014.
[3] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, Algorithmic Game Theory. Cambridge Univ. Press, 2007.
[4] G. Wei, A. Vasilakos, Y. Zheng, and N. Xiong, "A Game-Theoretic Method of Fair Resource Allocation for Cloud Computing Services," The J. Supercomputing, vol. 54, no. 2, pp. 252-269, 2010.
[5] N. Jain, I. Menache, J. Naor, and J. Yaniv, "A Truthful Mechanism for Value-Based Scheduling in Cloud Computing," Theory of Computing Systems, pp. 1-19, http://dx.doi.org/10.1007/s00224-013- 9449-0, 2013.
[6] Z. Kong, C.-Z. Xu, and M. Guo, "Mechanism Design for Stochastic Virtual Resource Allocation in Non-Cooperative Cloud Systems," Proc. IEEE Fourth Int'l Conf. Cloud Computing, pp. 614-621, 2011.
[7] Y. Wang, A. Nakao, and A.V. Vasilakos, "Heterogeneity Playing Key Role: Modeling and Analyzing the Dynamics of Incentive Mechanisms in Autonomous Networks," ACM Trans. Autonomous and Adaptive Systems, vol. 7, no. 3, article 31, 2012.
[8] D. Ardagna, B. Panicucci, and M. Passacantando, "Generalized Nash Equilibria for the Service Provisioning Problem in Cloud Systems," IEEE Trans. Services Computing, vol. 6, no. 4, pp. 429- 442, Oct.-Dec. 2013.
[9] V. Di Valerio, V. Cardellini, and F. Lo Presti, "Optimal Pricing and Service Provisioning Strategies in Cloud Systems: A Stackelberg Game Approach," Proc. IEEE Sixth Int'l Conf. Cloud Computing, pp. 115-122, 2013.
[10] X. Zhou, S. Gandhi, S. Suri, and H. Zheng, "ebay in the Sky: Strategy-Proof Wireless Spectrum Auctions," Proc. ACM Mobi- Com, pp. 2-13, 2008.