IJDCST @Sep, Issue- V-2, I-6, SW-03
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Dynamic Resource Allocation Using Green Computing Environment

[1]G.Swetha,[2]Sarat Chandra,[3]Ravi Mathey

[1]M.Tech Student- Vidya Jyothi Institute of Technology. Aziz nagar Hyderabad,

[2] Assistant Professor of CSE Department, Vidya Jyothi Institute of Technology, Hyderabad,

[3]Professor & Head of CSE Department, Vidya Jyothi Institute of Technology, Hyderabad.

ABSTRACT— Cloud computing allows business customers to scale up and down their resource usage based on needs., we present a system that uses virtualization technology to allocate data centre resources dynamically based on application demands and support green computing by optimizing the number of servers in use. We introduce the concept of “skewness” to measure the unevenness in the multidimensional resource utilization of a server. By minimizing skewness, we can combine different types of workloads nicely and improve the overall utilization of server resources. We develop a set of heuristics that prevent overload in the system effectively while saving energy used. Many of the touted gains in the cloud model come from resource multiplexing through virtualization technology. In this paper Trace driven simulation and experiment results demonstrate that our algorithm achieves good performance.

Index Terms—Cloud computing, resource management, virtualization, greencomputing

1 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 centres. 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.

BACKGROUND WORK:

SERVICE PERFORMANCE AND ANALYSIS IN CLOUD COMPUTING

Cloud computing is a new computing paradigm in which information and computer power can be accessed from a Web browser by customers. Understanding the characteristics of computer service performance has become critical for service applications in cloud computing. For the commercial success of this new computing paradigm, the ability to deliver Quality of Services (QoS) guaranteed services is crucial. In this paper, we present an approach for studying computer service performance in cloud computing

A BREAK IN THE CLOUDS: TOWARDS A CLOUD DEFINITION

This paper discusses the concept of Cloud Computing to achieve a complete definition of what a Cloud is, using the main characteristics typically associated with this paradigm in the literature. More than 20 definitions have been studied allowing for the extraction of a consensus definition as well as a minimum definition containing the essential characteristics. This paper pays much attention to the Grid paradigm, as it is often confused with Cloud technologies. We also de- scribe the relationships and distinctions between the Grid and Cloud approaches

CLOUD COMPUTING: A PERSPECTIVE STUDY

The Cloud computing emerges as a new computing paradigm which aims to provide reliable, customized and QoS guaranteed dynamic computing environments for end-users. In this paper, we study the Cloud computing paradigm from various aspects, such as definitions, distinct features, and enabling technologies. This paper brings an introduction review on the Cloud computing and provide the state-of-the-art of Cloud computing technologies.

MEGASTORE: PROVIDING SCALABLE, HIGHLY AVAILABLE STORAGE FOR INTERACTIVE SERVICES

Megastore is a storage system developed to meet the requirements of today's interactive online services. Megastore blends the scalability of a NoSQL data store with the convenience of a traditional RDBMS in a novel way, and provides both strong consistency guarantees and high availability. We provide fully serializable ACID semantics within ne-grained partitions of data. This partitioning allows us to synchronously replicate each write across a wide area net- work with reasonable latency and support seamless failover between datacenters. This paper describes Megastore's semantics and replication algorithm. It also describes our experience supporting a wide range of Google production services built with Megastore.

EXISTING SYSTEM

The number of servers is comparatively small, typically below 10, which makes them unsuitable for performance analysis of cloud computing data centers. Approximations are very sensitive to the probability distribution of task service times. User may submit many tasks at a time because of this bags-of-task will appear. Due to dynamic nature of cloud environments, diversity of user's requests and time dependency of load is high. The coefficient of variation of task service time is high.

LIMITATIONS

Traditional data mining techniques usually require entire data set to be present. Random access (or multiple access) to the data. Impractical to store the whole data. Simple calculation per data due to time and space constraints. Consist of 3PM and 5VM in a Single cloud. All VM's are Configured with 12MB of RAM.

PROPOSED SYSTEM

In Proposed system, the task is sent to the cloud center is serviced within a suitable facility node; upon finishing the service, the task leaves the center. A facility node may contain different computing resources such as web servers, database servers, directory servers, and others. A service level agreement, SLA, outlines all aspects of cloud service usage and the obligations of both service providers and clients, including various descriptors collectively referred to as Quality of Service (QoS). includes availability, throughput, reliability, security, and many other parameters, but also performance indicators such as response time, task blocking probability, probability of immediate service, and mean number of tasks in the system, all of which may be determined using the tools of queuing theory.

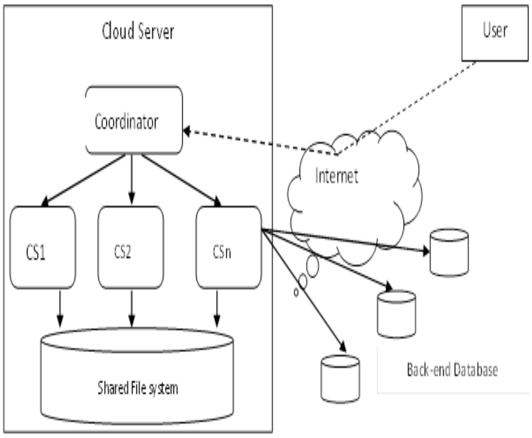

Fig 1: Architecture Diagram:

We model a cloud server system which indicates that the inter arrival time of requests is exponentially distributed, while task service times are independent and identically distributed random variables that follow a general distribution with mean value of u. The system under consideration contains m servers which render service in order of task request arrivals (FCFS).The capacity of system is m þ r which means the buffer size for incoming request is equal to r. As the population size of a typical cloud centre is relatively high while the probability that a given user will request service is relatively small, the arrival process can be modelled as a Markova process.

ADVANTAGES:

- Less Traffic Intensity.
- Analytical technique based on an approximate Markov chain model for best performance evaluation.
- General Service time for requests and large number of servers makes ourmodel flexible in terms of scalability and diversity of service time. High degree of accuracy for the mean number of tasks in the system, blocking probability, probability, response time.

HOT SPOT MITIGATION

We sort the list of hot spots in the system in descending temperature (i.e., we handle the hottest one first). Our goal is to eliminate all hot spots if possible. Otherwise, keep their temperature as low as possible. For each server p, we first decide which of its VMs should be migrated away. We sort its list of VMs based on the resulting temperature of the server if that VM is migrated away. We aim to migrate away the VM that can reduce the server's temperature the most. In case of ties, we select the VM whose removal can reduce the skewness of the server the most. For each VM in the list, we see if we can find a destination server to accommodate it. The server must not become a hot spot after accepting this VM. Among all such servers, we select one whose skewness can be reduced the most by accepting this VM. Note that this reduction can be negative which means we select the server whose skewness increases the least. If a destination server is found, we record the migration of the VM to that server and update the predicted load of related servers. Otherwise, we move onto the next VM in the list and try to find a destination server for it. As long as we can find a destination server for any of its VMs, we consider this run of the algorithm a success and then move onto the next hot spot. Note that each run of the algorithm migrates away at most one VM from the overloaded server

Green Computing

When the resource utilization of active servers is too low, some of them can be turned off to save energy. This is handled in our green computing algorithm. The challenge here is to reduce the number of active servers during low load without sacrificing performance either now or in the future. We need to avoid oscillation in the system. Our green computing algorithm is invoked when the average utilizations of all resources on active servers are below the green computing threshold. We sort the list of cold spots in the system based on the ascending order of their memory size. Since we need to migrate away all its VMs before we can shut down an underutilized server, we define the memory size of a cold spot as the aggregate memory size of all VMs running on it. Recall that our model assumes all VMs connect to share back-end storage.

CONCLUSION

We have presented the design, implementation, and evaluation of a resource management system for cloud computing services our system multiplexes virtual to physical resources adaptively based on the changing demand. We present a system that uses virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. We use the skewness metric to combine VMs with different resource characteristics appropriately so that the capacities of servers are well utilized. Our algorithm achieves both overload avoidance and green computing for systems with multi resource constraints. We have proposed a new strategy that can be included in the Cloud-Analyst to have cost effective results and development and we can conclude from the results that this strategy is able to do so. From the work done, we can conclude that the simulation process can be improved by modifying or adding new strategies for traffic routing, load balancing etc. to make researchers and developers able to do prediction of real implementation of cloud, easily. We develop a set of heuristics that prevent overload in the system effectively while saving energy used. Trace driven simulation and experiment results demonstrate that our algorithm achieves good performance. In the cloud model is expected to make such practice unnecessary by offering automatic scale up and down in response to load variation. It also saves on electricity which contributes to a significant portion of the operational expenses in large data centers.

FUTURE WORK

For the future work, scenario reduction techniques will be applied to reduce the number of scenarios. In addition, the optimal pricing scheme for cloud providers with the consideration of competition in the market will be investigated. Scenario reduction techniques will be applied to reduce the number of scenarios. In addition, the optimal pricing scheme for cloud providers with the consideration of competition in the market will be investigated. We need to predict the future resource needs of VMs. As said earlier, our focus is on Internet applications. One solution is to look inside a VM for application level statistics, e.g., by parsing logs of pending requests. Doing so requires modification of the VM which may not always be possible. Instead, we make our prediction based on the past external behaviors of VMs.

REFERENCES

[1] M. Armbrust et al., “Above the Clouds: A Berkeley View of Cloud Computing,” technical report, Univ. of California, Berkeley, Feb. 2009.
[2] L. Siegele, “Let It Rise: A Special Report on Corporate IT,” The Economist, vol. 389, pp. 3-16, Oct. 2008.
[3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the Art of Virtualization,”

Proc. ACM Symp. Operating Systems Principles (SOSP '03), Oct. 2003.
[4] "Amazon elastic compute cloud (Amazon EC2)," http://aws. amazon.com/ec2/, 2012.
[5] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," Proc. Symp. Networked Systems Design and Implementation (NSDI '05), May 2005.
[6] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast Transparent Migration for Virtual Machines," Proc. USENIX Ann. Technical Conf., 2005.
[7] M. McNett, D. Gupta, A. Vahdat, and G.M. Voelker, "Usher: An Extensible Framework for Managing Clusters of Virtual Machines," Proc. Large Installation System Administration Conf. (LISA '07), Nov. 2007.
[8] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-Box and Gray-Box Strategies for Virtual Machine Migration," Proc. Symp. Networked Systems Design and Implementation (NSDI '07), Apr. 2007.
[9] C.A. Waldspurger, "Memory Resource Management in VMware ESX Server," Proc. Symp. Operating Systems Design and Implementation (OSDI '02), Aug. 2002.
[10] G. Chen, H. Wenbo, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services," Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI '08), Apr. 2008.
[11] P. Padala, K.-Y. Hou, K.G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant, "Automated Control of Multiple Virtualized Resources," Proc. ACM European conf. Computer Systems (EuroSys '09), 2009.
[12] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," Proc. IFIP/IEEEInt'l Symp. Integrated Network Management (IM '07), 2007.
[13] "TPC-W: Transaction Processing Performance Council," http:// www.tpc.org/tpcw/, 2012.
[14] J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, and R.P. Doyle, "Managing Energy and Server Resources in Hosting Centers," Proc. ACM Symp. Operating System Principles (SOSP '01), Oct. 2001.
[15] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A Scalable Application Placement Controller for Enterprise Data Centers," Proc. Int'l World Wide Web Conf. (WWW '07), May 2007.
[16] M. Zaharia, A. Konwinski, A.D. Joseph, R.H. Katz, and I. Stoica, "Improving MapReduce Performance in Heterogeneous Environments," Proc. Symp. Operating Systems Design and Implementation (OSDI '08), 2008.
[17] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy: Fair Scheduling for Distributed Computing Clusters," Proc. ACM Symp. Operating System Principles (SOSP '09), Oct. 2009.
[18] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling," Proc. European Conf. Computer Systems (EuroSys '10), 2010.
[19] T. Sandholm and K. Lai, "Mapreduce Optimization Using Regulated Dynamic Prioritization," Proc. Int'l Joint Conf. Measurement and Modeling of Computer Systems (SIGMETRICS '09), 2009.

Author's Information

1. G.Swetha is M.Tech Student at Vidya Jyothi Institute of Technology (VJIT).

2. Sarat chandra.Jangam is a post-graduate specialized in Artificial Intelligence & Robotics from Andhra university, Vizag and he did Engineering in Computer Science Engineering from GEC, Gudlavalleru.He has 3 years experience in teaching. Presently he is working as Assistant professor in Vidya Jyothi Institute of Technology (VJIT).

3. Ravi Mathey is a post-graduate specialized in Computer Science from BIT -Ranchi and he did Instrumentation technology in Andhra University. He has more than 20 years of experience in Research and Development, and teaching presently he is working as a Professor and HOD of CSE Department at Vidya Jyothi Institute of Technology (VJIT). His area of research is wireless embedded application and image compression techniques by using fractals and wavelets, cloud computing