ISSN:0975-9646

Effective Resource Management for Virtualized Data centers

G.Ramakrishna[*1], Dr.A.Mallareddy[*2]

M.Tech Student, Dept of CSE, Sri Indu Institute of Engineering & Technology,
Ibrahimpatnam, Ranga Reddy, Telangana
Professor, Dept of CSE, Sri Indu Institute of Engineering & Technology,
Ibrahimpatnam, Ranga Reddy, Telangana

Abstract:Time and Technology has its significance made us to think and switching from classical to modern era of Information technology. If we look forward to the best of the Industrial world where Technology is the use of Information technology organization especially service based organization, provides a solution telling so and so high approaches. But we know the crackers and hackers may be ethical or any other which in this paper we have given the cryptographic acknowledgement based solution generated randomly; leads to the Data centers where Data is crucial. Cloud Infrastructure which needs to be optimized for the fault tolerant and performance evaluation is the measure metrics in the current trend of the IT Industry. Hence; in the context we have pulled the concept of the Pick time and load the data center with its alternative to the geographical location in the map reduce programming approach.

Keywords: Cloud computing, virtualization, performance evaluation, fault-tolerance, metrics/measurement, resource management.

I.INTRODUCTION

Virtualization refers to many different concepts in computer science, and is often used to describe many types of abstractions. In this work, we are primarily concerned with Platform Virtualization, which separates an operating system from the underlying hardware resources. Virtual Machine (VM) refers to the abstracted machine that gives the illusion of “real machine” .The earliest experiments with virtualization date back to 1960s, when IBM built VM/370 and operating system that gives the illusion of multiple independent machines. VM/370 is built for System/370 mainframe computers built by IBM, and the virtualization features are used to maintain backward compatibility with the instruction set in System/360 mainframes (precursor to System/370 mainframes). Similar attempts were made to provide virtual machines on DEC PDP-10.

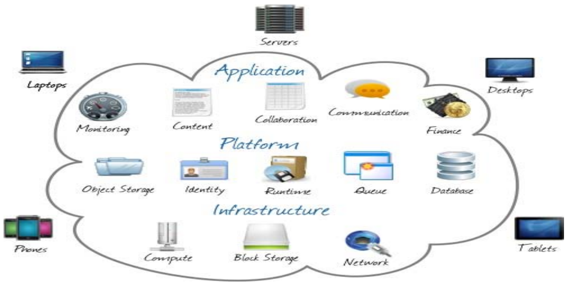

Fig.1.1. Illustration of the Cloud Network

II.RELATED WORK

Virtualization Proportional share schedulers allow reserving CPU capacity for applications. While these can enforce the desired CPU shares, our controller also dynamically adjusts these share values based on application-level metrics. It is similar to the feedback controller in that allocates CPU to threads based on an estimate of each thread’s progress, but our controller operates at a much higher layer based on end-to-end application performance that spans multiple tiers in a given application. In the past few years, there has been a great amount of research in improving scheduling in virtualization mechanisms. . The cap allows one to set a hard limit on the amount of CPU used by a VM. The share knob is expected to be used for proportional sharing. However, in our practice we found out those using caps as the single knob for enforcing proportions works better than trying to use both knobs together.

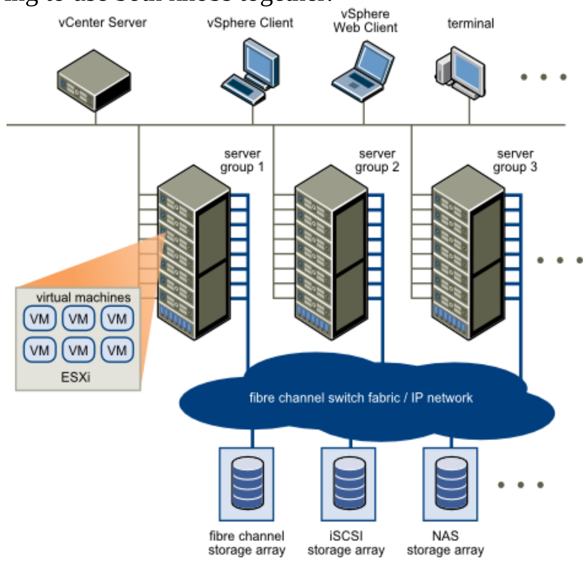

Fig.2.1. Datacenter w.r.t. VM

Virtualization technologies including VMware and, offer proportional share schedulers for CPU in the hypervisor layer that can be used to set the allocation for a particular VM. However, these schedulers only provide mechanisms for controlling resources. One also has to provide the right parameters to the schedulers in order to achieve desired application-level goals. For example CPU credit scheduler provides two knobs: cap and share.

III. PROPOSED METHODOLOGY

Virtualization design a two-layer, multi-input, multi-output controller to automatically allocate multiple types of resources to enterprise applications to achieve their SLOs. The first layer consists of a set of application controllers that automatically determine the amount of resources necessary to achieve individual application SLOs, using the estimated models and a feedback-based approach. The second layer is comprised of a set of node controllers that detect resource bottlenecks on the shared nodes and properly allocate multiple types of resources to individual applications. Under overload, the node controllers provide service differentiation according to the priorities of individual applications. The applications can be isolated by running them in different virtual machine. Different virtualization technologies differing levels of performance isolation, but most of them offer safety, correctness and fault-tolerance isolation. Though many virtualization technologies can be used for isolation, we choose hypervisor-based virtualization. The reasoning for choosing this method is described though physical machine can offer isolation by running multiple applications in different physical machines, such scenario would waste great amount of resources. Similar to storage, knobs for network resources are not yet fully developed in virtualization environments. Our initial efforts in adding network resource control have failed, because of inaccuracies in network actuators. Since native network control is not fully implemented, we tried to use Linux's existing traffic controller to allocate network resources to VMs. We found that the network bandwidth setting in is not enforced correctly when heavy network workloads are run.

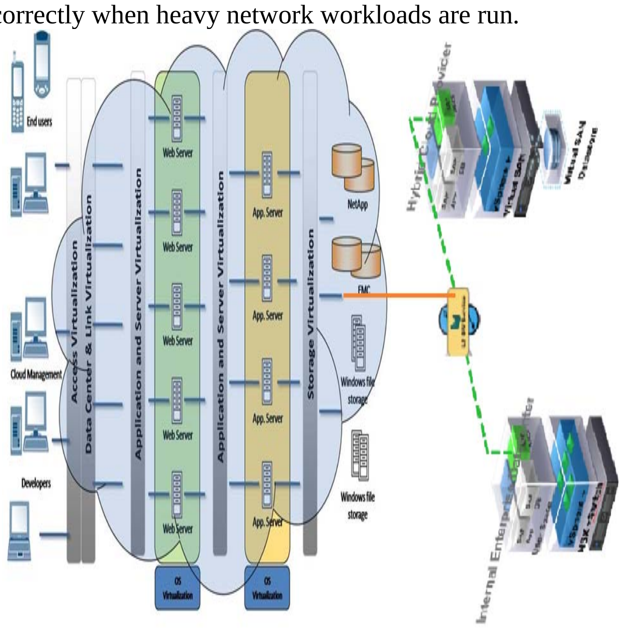

Fig.3.1. Architecture Flow Model of the VM Data center

However, the theory we developed in this work is directly applicable to any number of resources. The memory ballooning supported in VMware provides a way of controlling the memory required by a VM. However, the ballooning algorithm does not know about application goals or multiple tiers, and only uses the memory pressure as seen by the operating system. We have done preliminary work in controlling CPU and memory together with other researchers. In many consolidation scenarios, memory is the limiting factor in achieving high consolidation. In order for each App Controller to decide how much resource is needed for the application to meet its performance target, it first needs to determine the quantitative and dynamic relationship between the application's resource allocation and its performance. Such a relationship is captured in the notion of "transfer function" in traditional control theory for modeling of physical systems. However, most computing systems, such as the one considered in this paper, cannot be represented by a single, linear transfer function (or model) because their behavior is often nonlinear and workload-dependent. We assume, however, that the behavior of the system can be approximately characterized locally by a linear model. We periodically re-estimate the model online based on real-time measurements of the relevant variables and metrics, allowing the model to adapt to different operating regimes and workload conditions.

IV. EVOLUTION AND ANALYSIS

For traditional control systems, such models are often based on first principles. For computer systems, although there is queueing theory that allows for analysis of aggregate statistical measures of quantities such as utilization and latency, it may not be fine-grained enough for run-time control over short time scales, and its assumption about the arrival process or service time distribution may not be met by certain applications and systems. Therefore, most prior work on applying control theory to computer systems employs an empirical and "black box" approach to system modeling by varying the inputs in the operating region and observing the corresponding outputs.

V. CONCLUSION & FUTURE WORK

In this work, only fixed models were used to capture the input-output relationship in the steady state, which simplifies both the modeling process and the controller design. We explore the use of a dynamic model to capture more transient behavior of the system and use it as the basis for better controller design. All nodes in the data center are connected with a high speed network, so that sensor and actuation delays within Auto Control are small compared to the control interval. We also require accurate system sensors and actuators, and assume that the underlying system schedulers provide a rich enough interface to dynamically adjust resource shares for VMs.

REFERENCES

[1] N. Shang, M. Nabeel, F. Paci, and E. Bertino, "A Privacy-Preserving Approach to Policy-Based Content Dissemination," Proc. IEEE 26th Int'l Conf. Data Eng. (ICDE '10), 2010.
[2] "LibertyAlliance," http://www.projectliberty.org/, 2013.
[3] "OpenID," http://openid.net/, 2013.
[4] "Microsoft Windows CardSpace," http://msdn.microsoft.com/ en-us/library/aa480189.aspx, 2013.
[5] "Higgins Open Source Identity Framework," http://www.eclipse.org/higgins/, 2013.

[6] R. Richardson, "CSI Computer Crime and Security Survey," ttp://www.ppclub.org/CSIsurvey2008.pdf, technical report, Computer Security Inst., 2008.
[7] Y. Challal and H. Seba, "Group key Management Protocols: A Novel Taxonomy," Int'l J. Information Technology, vol. 2, no. 2, pp. 105-118, 2006.
[8] H. Harney and C. Muckenhirn, "Group key Management Protocol (GKMP) Specification," technical report, Network Working Group, United States, 1997.
[9] H. Chu, L. Qiao, K. Nahrstedt, H. Wang, and R. Jain, "A Secure Multicast Protocol with Copyright Protection," SIGCOMM Computer Comm. Rev., vol. 32, no. 2, pp. 42-60, 2002.
[10] C. Wong and S. Lam, "Keystone: A Group Key Management Service," Proc. Int'l Conf. Telecomm. (ICT), 2000.
[11] A. Sherman and D. McGrew, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," IEEE Trans. Software Eng., vol. 29, no. 5, pp. 444-458, May 2003.
[12] G. Chiou and W. Chen, "Secure Broadcasting Using the Secure Lock," IEEE Trans. Software Eng., vol. 15, no. 8, pp. 929-934, Aug. 1989.
[13] S. Berkovits, "How to Broadcast a Secret," Proc. 10th Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT '91), pp. 535-541, 1991.
[14] X. Zou, Y. Dai, and E. Bertino, "A Practical and Flexible Key Management Mechanism for Trusted Collaborative Computing," Proc. IEEE INFOCOM, pp. 538-546, Apr. 2008.
[15] A. Shamir, "How to Share a Secret," Comm. ACM, vol. 22, no. 11, pp. 612-613, 1979.
[16] E.F. Brickell, "Some Ideal Secret Sharing Schemes," Proc. Workshop the Theory and Application of Cryptographic Techniques on Advances in Cryptology (EUROCRYPT '89), pp. 468-475, 1990.
[17] N. Shang, M. Nabeel, F. Paci, and E. Bertino, "A Privacy-Preserving Approach to Policy-Based Content Dissemination," Proc. IEEE 26th Int'l Conf. Data Eng. (ICDE '10), 2010.
[18] O. Goldreich, Foundations of Cryptography: Basic Tools. Cambridge Univ. Press.

AUTHORS PROFILE:

Gottipati Ramakrishna M.Tech (computer science) pursuing in sri indu institute of engineering & technology B.Tech (it) from brillient inistitute of engineering & technology abdhulapurmet (v), hayathnagar(md),rr(dt) .

A. MALLAREDDFY He is currently working as a Professor, at Sri indu institute of engineering & Technology, Sheriguda (V), ibrahimptnam (MD), RR (D). He guided various projects in UG&PG level for CSE & CS Departments.