Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 11, November 2014, pg.564 – 569

SURVEY ARTICLE

A Survey on Auto Scaling of Internet Application in a Cloud Environment

Sasipriya.S[1], Ms.Kavitha[2]

Department of Computer Science and Engineering, Sri Eshwar College of Engineering, Coimbatore

mailtosasipriya@gmail.com [1], kaviktg@gmail.com [2]

Abstract: Automatic scaling and dynamic resource management in cloud server provisioning has become an active area of research in the Cloud Computing paradigm. Cost of resources varies significantly depending on configuration for using them. Hence efficient management of resources is of prime interest to both Cloud Providers and Cloud Users. In this work we suggest a probabilistic resource provisioning approach that can be exploited as the input of a dynamic resource management scheme. Using resource on Demand use case to justify our claims, we propose a class constraint bin packing technique with online colour set algorithm inspired from standard models developed to represent sudden and intense workload variations. We show that the resulting model verifies Deterministic algorithms that statistically characterize extreme rare events, such as the ones produced by varying resource demands that may cause workload overflow in the resource on demand context. This analysis provides valuable insight on expectable abnormal behaviors of systems. We exploit the information obtained using Deterministic algorithms for the proposed on Demand use-case for defining policies (Service Level Agreements). We believe these policies for elastic resource provisioning and usage may be of some interest to all stakeholders in the emerging context of cloud networking.

Keywords: a class constraint bin packing (CCBP), Automatic scaling, Cloud computing

1. Xen and art of Virtualization

This paper presents Xen, an X86 virtual machine monitor allows multiple commodity operating systems to share conventional hardware in safe and resource managed fashion but without sacrificing either performance or functionality over design targeted at hosting up to

100 virtual instances simultaneously on modern server. The virtualization approach taken by Xen is extremely efficient.

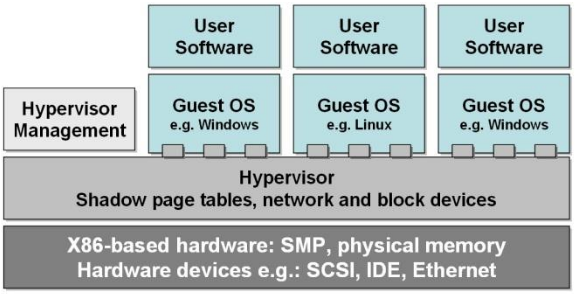

Figure: Xen, virtual machine monitor

Successful portioning of machine to support the concurrent execution of multiple operating systems poses several challenges.

1. Virtual machine must be isolated from one another.

2. It is necessary to support a variety of different operating system to accommodate the heterogeneity of popular application.

3. The performance overhead introduced by virtualization should be small.

The basic approach to build Xen, which multiplexes physical resources at the granularity of an entire operating system and is able to provide performance isolation between them. In contrast to process, level multiplexing this allows a range of guest operating system to gracefully coexist rather than mandating a specific application binary interface. There is a price to pay for this flexibility. Running a full OS is more heavy weight than running a process both in terms of initialization and in terms of resource consumption.

For our target of up to 100 hosted OS instances.paravirtualized X86 interface factored into 3 broad aspects of system: memory management, CPU and device I/O.

Xen provides an excellent platform for deploying a wide variety of network centric services such as local mirroring of dynamic web content, media stream transcending and distribution, multiplayer game and virtual reality servers and smart proxies to provide ephermence network presence for transiently connected devices.

2. Multi-Agent Learning Approach to Online Distributed Resource Allocation

Efficient resource allocation in a network of computing clusters may enable building large computing infrastructure. We consider this problem as a novel application for multi agent learning (MAL).

The MAL algorithms applies for optimizing online resource allocation in a cluster networks.

To build a larger shared computing infrastructure, one common model is to organize a set of shared clusters into network an enable resource sharing across shared clusters. The resource allocation decision is now distributed to each shared clusters. Each cluster (referred to as agent) still uses a cluster-wide technique for managing its local resources. Here task is referred as application services. All agents make decision depends not only on its local state but also on other agent states and policies. To simplify learning the learning, we decompose each agent's decisions into two connected learning problem:

Local allocation problem-deciding what tasks to be allocated locally.

Task routing problem-deciding where to forward a task.

To avoid poor initial policies during learning, heuristic strategies are developed to speed up the learning.

3. Tight bounds for online class-constrained packing

The vector bin packing problem considers multi-dimensional constraints when packing items into a minimum number of bins for the Automatic scaling of the internet application for resource management. CPU demand and the memory requirement of an Internet application as individual elements in the vector and use vector bin packing .The Class Constrained Multiple Knapsack problem (CCMK) aims to maximize the total number of packed items under the restriction that each knapsack has a limited capacity and a bound on the number of different types of items it can hold. Process migration for multiple application is modelled and achieved a best results. Unlike virtualization technology, it does not capture the execution environment of the running processes. Nor does it support the auto scaling of the processes based on the observed demand along with load balancing strategies. Performance Factor Considered for System evaluation

- Demand satisfaction ratio
- Computing Capacity
- Average server utilization
- Response time

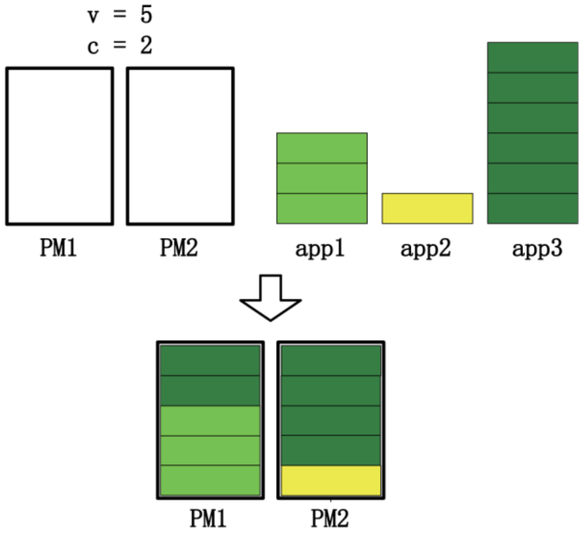

Figure Class constrained bin packing (CCBP).

4. Resource Allocation in a Network-Based Cloud Computing Environment: Design Challenges

Cloud computing is an increasingly popular computing example, now proving a necessity for utility computing services. Each provider offers a unique set of services with a range of resource configurations. Resource provisioning for cloud services in a wide-ranging way is essential to any resource allocation model. The resource allocation is to accurately represent computational resources and network resources.

Another aspect that should be considered while provisioning resources is energy consumption. With that in mind, resource allocation algorithms aim to achieve the task of scheduling virtual machines on data center servers and then scheduling connection requests on the network paths available while complying with the problem constraints.

Since it is a relatively new paradigm, the community still has to tackle deeply these issues regarding SDN:

A-Reliability - Using centralized SDN controller affects reliability. Although solutions like stand by controllers or using multiple controllers for the network are suggested, practical investigation is needed to reveal the problems and analyze the trade-offs of using such solutions.

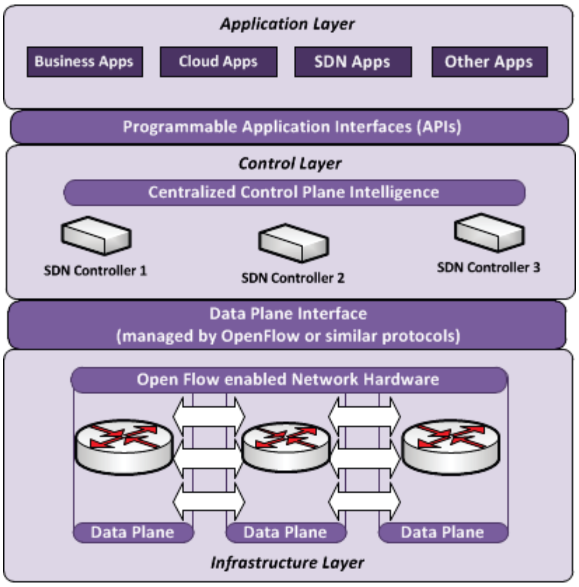

figure: SDN architecture

B- Scalability - When the network scales up in the number of switches and the number of end hosts, the SDN controller becomes a key bottleneck

Visibility - for example, that a backup was slowing the network. The solution would then be to simply reschedule it. Unfortunately with SDN, only a tunnel source and a tunnel endpoint with User Datagram Protocol (UDP) traffic are visible.

5. Dynamic Placement for Clustered Web Applications

The evaluation of middleware clustering technology capable of allocating resources to web applications through dynamic application instance placement. The application instance placement as the problem of placing application instances on a given set of server machines to adjust the amount of resources available to applications in response to varying resource demands of application clusters. The objective is to maximize the amount of demand that may be satisfied using a configured placement. To limit the disturbance to the system caused by starting and stopping application instances, the placement algorithm attempts to minimize the number of placement changes. It also strives to keep resource Utilization balanced across all server machines. Two types of resources are managed, one load-dependent and one loadindependent. When putting the chosen placement in effect our controller schedules placement changes in a manner that limits the disruption to the system.

6. A Scalable Application Placement Controller for Enterprise Data Centers

Given a set of machines and a set of Web applications with dynamically changing demands, an online application placement controller decides how many instances to run for each application and where to put them, while observing all kinds of resource constraints. In this paper, a new algorithm that can produce within 30 seconds high-quality solutions for hard placement problems with thousands of machines and thousands of applications. This scalability is crucial for dynamic resource provisioning in large-scale enterprise data centers. Our algorithm allows multiple applications to share a single machine, and strives to maximize the total satisfied application demand, to minimize the number of application starts and stops, and to balance the load across machines. Our algorithm has been implemented and adopted in a leading commercial middleware product for managing the performance of Web applications.

Conclusion

System has been designed and implemented to work against the dynamic work load in the cloud data centers .In this work , we established a class constrained bin packing technique by utilizing the online color set algorithm to the system that can scale up and down the number of application instances automatically based on demand. We developed a color set algorithm to decide the application placement and the load distribution. Process migration is also carried out due to load variation above and below the threshold values .Our system achieves high satisfaction ratio of application demand even when the load demand is very high and very low. It saves energy by reducing the number of running instances when the load is low which achieve the low computation cost and low VM utilization in the cloud environments.

References

[1] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/. Accessed on May 10, 2012.

[2] 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,” in Proc. ACM Symp. Oper. Syst. Princ. (SOSP’03),Oct. 2003, pp. 164–177.

[3] M. McNett, D. Gupta, A. Vahdat, and G. M. Voelker, “Usher: An extensible framework for managing clusters of virtual machines,” in Proc. Large Install. Syst. Admin. Conf. (LISA’07), Nov. 2007, pp. 1–15.

[4] J. Zhu, Z. Jiang, Z. Xiao, and X. Li, “Optimizing the performance of virtual machine synchronization for fault tolerance,” IEEE Trans. Comput., vol. 60, no. 12, pp. 1718–1729, Dec. 2011.

[5] Linux Documentation. http://www.kernel.org/doc/documentation/power/states.txt. Accessed on May 10, 2012.

[6] H. Shachnai and T. Tamir, “Tight bounds for online class constrained packing,” Theor. Comput. Sci., vol. 321, no. 1, pp. 103–123, 2004.

[7] L. Epstein, C. Imreh, and A. Levin, “Class constrained bin packing revisited,” Theor. Comput. Sci., vol. 411, no. 34–36, pp. 3073–3089,2010.

[8] B. Urgaonkar, P. Shenoy, and T. Roscoe, “Resource overbooking and application profiling in shared hosting platforms,” SIGOPS Oper.Syst. Rev., vol. 36, no. SI, pp. 239–254, 2002

[9] C. Zhang, V. Lesser, and P. Shenoy, “A multi-agent learning approach to online distributed resource allocation,” in Proc. Int. Joint Conf. Artif. Intell. (IJCAI’09), 2009, pp. 361–366.
[10] J. L. Wolf and P. S. Yu, “On balancing the load in a clustered web farm,” ACM Trans. Internet Technol., vol. 1, no. 2, pp. 231–261, 2001.
[11]A. Karve, T. Kimbrel, G. Pacifici, M. Spreitzer, M. Steinder,M. Sviridenko, and A. Tantawi, “Dynamic placement for clustered web applications,” in Proc. Int. World Wide Web Conf. (WWW’06),May 2006, pp. 595–604.
[12] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, “A scalable application placement controller for enterprise data centers,” in Proc. Int. World Wide Web Conf. (WWW’07), May 2007, pp. 331–340.
[13] J. Famaey, W. D. Cock, T. Wauters, F. D. Turck, B. Dhoedt, and P. Demeester, “A latency-aware algorithm for dynamic service placement in large-scale overlays,” in Proc. IFIP/IEEE Int. Conf. Symp.Integrat. Netw. Manage. (IM’09), 2009, pp. 414–421.