www.ijiarec.com

ISSN: 2348-2079

MAR-2015

International Journal of Intellectual Advancements and Research in Engineering Computations

AN IMPROVED RESOURCE SCHEDULING FOR VIRTUAL MACHINES IN CLOUD DATA CENTERS USING QUALITY OF SERVICE

[1]V.Manimaran, [2]S.Prabhu, [3]Dr.N.Sengottaiyan

ABSTRACT

Cloud computing grants business clients to scale up and down their resource usage based on their needs. Many of the touted benefits in the cloud model come from resource multiplexing through virtualization concepts. Dynamic consolidation of virtual machines (VMs) is an effective approach to improve the utilization of resources and energy efficiency in cloud data centers. Finding out when it is best to reallocate VMs from an overfull host is an aspect of dynamic VM consolidation that directly influences the reserve exploitation and quality of service (QoS) delivered by the system In this paper, we introduces VM Migration and optimal precedence, a technique that obviously migrates only the working set of an idle VM and support green computing by optimizing the number of servers in use. We use the maximum precedence algorithm to reduce the trouble in virtual machine. We develop a set of heuristics that put off trouble in the system efficiently while saving energy used.

KEYWORDS: Cloud Computing, Resource Management, Virtualization, Green Computing.

.

INTRODUCTION

Cloud computing is attracting many business customers since it is providing infinite computing resources which is available on demand. Today, we have an ability to utilize scalable, distributed computing environments within the confines of the Internet, a practice known as cloud computing. In this new world of computing, users are generally required to accept the underlying hypothesis of trust. Within the cloud computing world, the virtual environment lets users' access computing power that exceeds that restricted within their own corporal worlds. Cloud computing is the process of providing computer conveniences via internet. And it's provided us better and efficient way to access information in timely manner and also increases storage of capacity for user in.

Cloud computing enables a new dealing model that supports on-demand, pay for-use, and economies-of-scale IT services over the Internet. Cloud platforms are dynamically built through virtualization with provisioned hardware, software, networks, and datasets. The idea is to migrate desktop computing to a service-oriented platform using virtual server clusters at data centers. However, a lack of trust between cloud users and providers has held up the universal approval of clouds as outsourced computing services. To promote multi tenancy, we must design the cloud ecological unit to be secure, rust precious, and dependable. In reality, trust is a social problem, not a purely technical issue. However, we believe that technology can enhance trust, justice, standing, credibility, and assurance in Internet applications. To increase the adoption of Web and cloud services, cloud service providers

Author for Correspondence:

[1]PG Scholar, Dept of CSE, Nandha Engineering College, Erode, Tamilnadu, India. Email: mani.nec2007@gmail.com.

[2]Assistant Professor, Dept of CSE, Nandha Engineering College, Erode, Tamilnadu, India. Email: prabhuvelli@gmail.com.

[3]Principal, Indira Institute of Engineering & Technology, Padur, Thiruvallur, Tamilnadu, India.

(CSPs) must first launch trust and security to ease the uncertainties of a large number of users.

We aim to achieve two goals in this paper

- Overload avoidance. The capacity of a PM should be sufficient to satisfy the resource needs of all VMs running on it. Otherwise, the PM is overloaded and can lead to degraded performance of its VMs.
- Green computing. The number of PMs used should be minimized as long as they can still satisfy the needs of all VMs. Idle PMs can be turned off to save energy.

In this paper, we present the design and implementation of resource management system that achieves a good balance between the two goals. We make the following contributions:

- We develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used.
- We introduce the concept of "VM Migration" to mitigate the load from one VM to another. With this, we could improve the overall utilization of servers in the face of multidimensional resource constraints.
- We design optimal precedence algorithm in order to reduce the burden in virtual machines.

The rest of the paper is organized as follows. Section 2 provides an overview of our system and Section 3 describes problem formulation. The details of our algorithm are presented in Section 4. Sections 5 and 6 present simulation and experiment results, respectively. Section 7 concludes.

BACKGROUND AND MODEL

There have been a lot of work done on resource scheduling and allocation in cloud computing. New algorithms and management techniques and different methods for resource scheduling in cloud computing are being preferred to make cloud computing a best experience for providers as well as customers. The surveys on scheduling strategies, techniques, methods have been done and a lot of task/job scheduling algorithms are introduced. The resource scheduling is been a tough job in cloud especially as it is the one which decides which process will be allocated to which resource and for how much time [2]. There are also resource allocation strategies that take into consideration the input parameters and on the basis if whether they are related to either of provider and customer. These parameters are execution time, policy, virtual machine, application, auction, utility function, gossip, hardware resource dependency, SLAs. While making a strategy the allocation methods should keep into consideration resource contention, fragmentation, under provisioning and over provisioning [6]. The different task scheduling methods in cloud computing are Cloud Service, User Level, Dynamic and Static, Heuristic, Workflow and Real Time scheduling. Some of the scheduling algorithms in cloud whether or job or task or resources or workflow are Compromised-Time-Cost, Particle Swarm Optimization based Heuristic, Improved cost based for tasks, RASA workflow, SHEFT workflow, Innovative transaction intensive cost constraint, Multiple QoS Constrained for Multi- Workflows. There are also the workflow scheduling algorithms that are described some of which are deadline constrained, ant colony, market oriented hierarchical etc. These surveys concluded that there is still a need for reliable and available resource scheduling algorithms as none of them focuses on both parameters [5].

FORMULATION OF PROBLEM

The problem defined presented design, implementation, and evaluation of a resource management system for cloud computing services. System multiplexes virtual to physical resources adaptively based on the changing demand. 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 used. It used the skewness metric to combine VMs with different resource characteristics appropriately so that the capacities of servers are well utilized. The algorithm achieves both overload avoidance and green computing for systems with multi resource constraints. We develop a set of heuristics that prevent overload in the system effectively while saving energy used.

OPTIMAL RESOURCE ALLOCATION

In this paper, we aim to allocate the resources based on optimal resource allocation techniques. Maximum precedence algorithm includes following steps. .

1. Begin to login after account creation.

2. Generate an account with possible rule from given items;

3. Compute confidence of all the rules for each hidden item, compute confidence of rule.

4. For each rule in client which server is in RHS

    4.1 If confidence (Items) < Rules, then

    Go to next 2-itemset;

    Else go to step 5

5. Decrease Support of RHS item H. (T=DataTransaction,t=Message of the data)

    5.1 Find T=t in cloud fully supports R;

    5.2 While (T is not empty)

    5.3 Choose the first transaction t from T;

    5.4 Modify t by putting 0 instead of 1 for RHS item;

    5.5 Remove and save the first transaction t from T; End While

6. Compute confidence of R;

7. If T is empty, then t cannot be hidden;

8. For each rule in which client is in LHS

9. Increase Support of LHS;

10. Find T=t in cloud| t does not support Rule;

11. While (T is not empty)

12. Modify t by putting 1 instead of 0 for LHS item;

13. Remove and save the first transaction t from T; End While

14. Compute confidence of R;

15. If T is empty, then item cannot be hidden;

    End For;

    End Else;

    End For;

16. Output updates cloud, as the transformed cloud server;

Finally it is finished after completion of desired action by the user. For update, the above derived algorithms are taken into the account. For new data update, each data block should be updated routinely from already existing value to new updated value. So if we consider this data block as one array formation then, the result is depending on the users operation on the data. It may be positive or negative depending on the update operation to be performed.

EXPERIMENTS

Our experiments are conducted using Netbeans IDE where cloudsim is incorporated. Since working in real cloud is difficult, our concepts are simulated and results are compared then. We use QoS parameter in the simulation.

ALGORITHM EFFECTIVENESS

We evaluate the effectiveness of our algorithm in overload mitigation and green computing. We start with a small scale experiment consisting of three PMs and five VMs so that we can present the results for all servers in Fig. 1. Different shades are used for each VM.

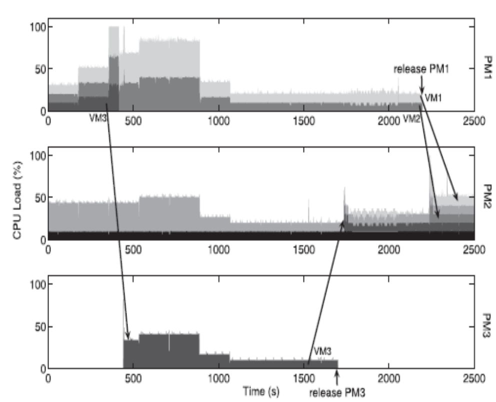

Fig 1: Algorithm Effectiveness

Fig. 2 shows how the number of APMs varies with the average number of requests to each VM over time. We keep the load on each VM low at the beginning. As a result, green computing takes effect and consolidates the VMs onto a smaller number of servers.

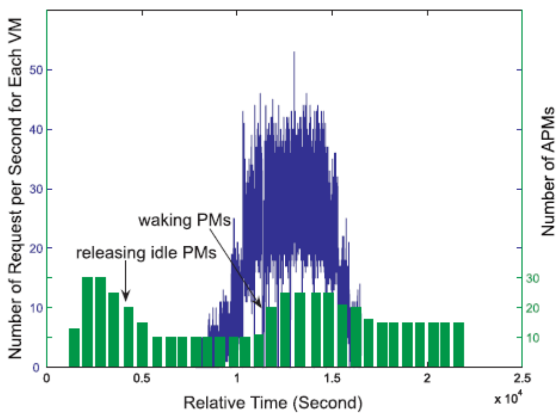

Fig 2: APMs varies with load

Fig. 2 shows that over the course of the experiment, the number of APM rises much faster than it falls. This is due to the effect of our FUSD load prediction. The figure also shows that the number of APMs remains at a slightly elevated level after the flash crowd. This is because the TPC-W servers maintain some data in cache and hence its memory usage never goes back to its original level.

SUMMARY OF RESULTS

DYNAMIC NETWORK RESOURCE ALLOCATION ON HOMOGENOUS NODE

The large data files are stored in the cloud and rely on the cloud for data safeguarding and multiplication, can be either entity consumers or organizations. An individual, which is managed by Cloud Service Provider (CSP), has major storage space and multiplication reserve to maintain the clients' data.

Server - Client computing or networking is a distributed application structural design that partitions tasks or workloads between service providers and service requesters, called clients. Often clients and servers operate over a computer network on separate hardware. A server machine is a high-performance host that is running one or more server programs which share its resources with clients. A client also shares any of its resources. Clients therefore initiate communication sessions with servers which await incoming requests. The processing frameworks typically assume the possessions they manage consist of a static set of harmonized compute nodes. Although designed to deal with individual nodes failures, they consider the number of available machines to be constant, especially when scheduling the processing job's execution. While IaaS clouds can certainly be used to create such cluster-like setups, much of their plasticity leftovers unused.

SERVICE AS LINEAR SCHEDULING TASKS

Even though a user may create a fake ID to enter the tune-up, his location alone may disclose his actual identity. Linking a position to an individual is possible by various means, such as publicly available information city maps. When a user wishes to pose a query, he sends his location to a trusted server, the anonymizer through a secure connection. The LS retrieves a candidate set that is guaranteed to contain the query results for any possible user location inside the process. We egg on the public auditing system of data storage security in cloud computing, and propose a protocol supporting for fully dynamic data operations, especially to support block insertion, which is missing in most on hand schemes. Considering the processing time, resource utilization based on CPU usage, memory usage and throughput, the cloud environment with the service node to control all clients request, could provide maximum service to all clients. Scheduling the resource and tasks separately involves more waiting time and response time. A scheduling algorithm named as Linear forecast for Tasks and Resources (LSTR) is designed, which performs tasks and resources scheduling respectively. Here, the combination of Nimbus and Cumulus services are imported to a server node to establish the IaaS cloud environment and KVM/Xen virtualization along with LSTR scheduling is used to allocate resources which maximize the system throughput and resource utilization.

RESOURCE PROVISION TO WORKLOAD SERVICE

In modern virtualization based subtract clouds, applications share the underlying hardware by running in isolated Virtual Machines (VMs). Each VM, during its initial creation, is configured with a certain amount of computing resources (such as CPU, memory and I/O). A key factor for achieving economies of scale in a compute cloud is resource provisioning, which refers to allocating resources to VMs to match their workload. Typically, efficient provisioning is achieved by two operations: (1) Static resource provisioning. VMs are created with specified size and then consolidated onto a set of physical servers. The VM capacity does not change; and (2) Dynamic resource provisioning. VM capacity is with dynamism adjusted to match workload fluctuations. Static provisioning often applies to the initial stage of capacity planning. It is usually conducted in offline and occurs on monthly or seasonal timescales. Such provisioning functionality has been included in much commercial cloud executive software.

SCHEDULED TASK ON PROVISIONING UTILIZATION

Recently, significant explore attention has focused on preventing distinctiveness inference in location-based services. At the end, these proposed cover alternative location privacy approaches and discuss why they are unfortunate to our problem setting. This offers privacy protection in the sense that the actual user position cannot be distinguished from others. Priority-based machine collocation incurs trivial performance impact to packet running in the high priority machines. The average performance loss of packet running in the foreground tier is between 0.0% and 3.7% compared to those running in the nodes exclusively (one-tier machine). We simply model the loss as a uniform distribution. When a foreground machine runs a packet with CPU exploitation higher than 96%, collocating a machine to run in background does not benefit either the foreground or the background packet due to that context switching incurs overhead and the background machine has very small chance to get corporeal resource to run. When a foreground machine runs a packet with low CPU utilization, the packet running in the collocated background machine can get significant share of physical resources to run.

To reduce cloud system resource cost, application consolidation is a must. In this paper, we present a novel pattern driven application consolidation (PAC) system to achieve efficient resource sharing in virtualized cloud computing infrastructures. PAC employs signal meting out techniques to dynamically discover significant patterns called signatures of different applications and hosts. PAC then performs dynamic application consolidation based on the extracted signatures. Cloud computing allows users to lease computing resources in a pay-as-you-go fashion without maintaining complex infrastructures themselves. Different from conventional distributed resource provisioning infrastructures, cloud systems grant users with *direct* but *shared* accesses to system resources and charge users for the exact resources and services they use (e.g., in terms of resource usage time). To reduce the resource cost of the cloud system, application consolidation is a must in order to host a large number of applications on a common physical computing transportation. However, without considering fine-grained resource usage patterns of different applications, cloud systems are forced to either abundance or under-provision resources. Resource over-provisioning will incur resource waste to the cloud system while resource under-provisioning will cause service level objective (SLO) violations to cloud applications. Thus, we need to provide more efficient resource control to minimize both resources over-provisioning and under-provisioning in cloud systems.

QUERY PROCESSING AT EXECUTION TIME OPTIMIZATION

Dealing out is based on implementation of the theorem that uses search operations as off the projection building blocks. Thus, it can be easily adapted to different network storage schemes. In this case, the queries are evaluated in a batch. This proposes the network-based anonymization and processing framework, the first system for query processing in road networks. Using this autonomy to construct the most efficient Execution Graph (in terms of processing time or monetary cost) is currently a major focus of our research. Unless the user provides any job annotation which contains more specific instructions we currently pursue a simple default strategy: Each vertex of the Job Graph is transformed into one Execution Vertex. The default channel types are network channels. Each carrying out Vertex is by default assigned to its own Execution Instance unless the user's annotations or other scheduling restrictions (e.g., the usage of in-memory channels) prohibit it. The default instance type to be used is the one with the lowest price per time unit available in the IaaS cloud.

MAXIMUM PRECEDENCE ALGORITHM AND VIRTUAL ALLOCATION TASK

Maximum precedence algorithm is used to reduce the burden in virtual machine. Set of heuristics that prevent burden in the system effectively while saving energy used. Virtualization technology is used to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. Resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used.

Server virtualization is the partitioning of a physical server into smaller virtual servers to help maximize your server resources. In server virtualization the assets of the server itself are hidden, or masked, from users, and software is used to divide the physical server into multiple virtual environments, called virtual or secretive servers. Applications run on virtual servers that are constructed using virtual machines, and one or more virtual servers are mapped onto each physical server in the system.

CONCLUSION

We have introduced a maximal precedence and optimal algorithm to solve for the problem of host overkill detection as a part of dynamic VM consolidation. The model allows a system to unambiguously set a QoS goal in terms of the optimal parameter, which is a workload independent QoS metric. For a known stationary workload and a given state configuration, the control policy obtained from the model optimally solves the host overload detection problem in the online setting by maximizing the mean intermigration time, while meeting the QoS goal. Using the Multi size descending Window workload estimation approach, we have heuristically adapted the model to handle unknown at a standstill workloads. We have also proposed an optimal for the problem of host overload detection to evaluate the efficiency algorithm. The future work focuses primarily on the server memory resource. Extending our work to multi-resource VM allocation that takes into account memory, CPU, and network resources is another important direction for future work.

REFERENCES

[1] Olivier Beaumont, Lionel Eyraud-Dubois, Christopher Thraves Caro, and Hejer Rejeb, "Heterogeneous Resource Allocation under Degree Constraints", IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 5, May 2013.

[2] Mohamed Abu Sharkh, Manar Jammal, Abdallah Shami, and Abdelkader Ouda, "Resource Allocation in a Network-Based Cloud Computing Environment: Design Challenges", IEEE Communications Magazine, 0163-6804/13, November 2013.

[3] Bernardetta Addis, Danilo Ardagna, Barbara Panicucci, Mark S. Squillante, and Li Zhang, "A Hierarchical Approach for the Resource Management of Very Large Cloud Platforms", IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 5, September/October 2013.

[4] Qian Zhu, and Gagan Agrawal, "Resource Provisioning with Budget Constraints for Adaptive Applications in Cloud Environments", IEEE Transactions on Services Computing, vol. 5, no. 4, October-December 2012.

[5] Imad M. Abbadi, and Anbang Ruan, “Towards Trustworthy Resource Scheduling in Clouds”, IEEE Transactions on Information Forensics and Security, vol. 8, no. 6, June 2013.

[6] Hamzeh Khazaei, Jelena Mi_si_c, Vojislav B. Mi_si_c, and Saeed Rashwand, “Analysis of a Pool Management Scheme for Cloud Computing Centers”, IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 5, May 2013.

[7] Sheng Di, and Cho-Li Wang, “Dynamic Optimization of Multiattribute Resource Allocation in Self-Organizing Clouds”, IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 3, March 2013.

[8] Zhen Xiao, Weijia Song, and Qi Chen, “Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment”, IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, June 2013.

[9] Ying Song, Yuzhong Sun, and Weisong Shi, “A Two-Tiered On-Demand Resource Allocation Mechanism for VM-Based Data Centers”, IEEE Transactions on Services Computing, vol. 6, no. 1, January-March 2013.

[10] Kyle Chard, and Kris Bubendorfer, “High Performance Resource Allocation Strategies for Computational Economies”, IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 1, January 2013.

[11] Hong Xu, and Baochun Li, “Anchor: A Versatile and Efficient Framework for Resource Management in the Cloud”, IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, June 2013.

[12] Mayank Mishra, Anwesha Das, Purushottam Kulkarni, and Anirudha Sahoo, “Dynamic Resource Management Using Virtual Machine Migrations”, IEEE Communications Magazine, 0163-6804/12, September 2012.

[13] Sharrukh Zaman, and Daniel Grosu, “A Combinatorial Auction-Based Mechanism for Dynamic VM Provisioning and Allocation in Clouds”, IEEE Transactions on Cloud Computing, vol. x, no. x, xxxx, 10.1109/TCC.2013.9.

[14] Wenhong Tian, Yong Zhao, Minxian Xu, Yuanliang Zhong, and Xiashuang Sun, “A Toolkit for Modeling and Simulation of Real-Time Virtual Machine Allocation in a Cloud Data Center”, IEEE Transactions on Automation Science and Engineering, 10.1109/TASE.2013.2266338.

[15] Luiz F. Bittencourt, Edmundo R. M. Madeira, and Nelson L. S. da Fonseca, “Scheduling in Hybrid Clouds”, IEEE Communications Magazine, 0163-6804/12, September 2012.

[16] Kyle Chard, Kris Bubendorfer, Simon Caton, and Omer F. Rana, “Social Cloud Computing: A Vision for Socially Motivated Resource Sharing”, IEEE Transactions on Services Computing, vol. 5, no. 4, October-December 2012.

[17] M. Reza Hoseiny Farahabady, Young Choon Lee, and Albert Y. Zomaya, “Pareto-Optimal Cloud Bursting, IEEE Transactions on Parallel and Distributed Systems”, 10.1109/TPDS.2013.218.

[18] Chun-Wei Tsai, and Joel J. P. C. Rodrigues, “Metaheuristic Scheduling for Cloud: A Survey”, IEEE Systems Journal, 10.1109/JSYST.2013.2256731.

[19] Jens Buysse, Konstantinos Georgakilas, Anna Tzanakaki, Marc De Leenheer, Bart Dhoedt, and Chris Develder, “Energy-Efficient Resource-Provisioning Algorithms for Optical Clouds”, J. OPT. COMMUN. NETW, vol. 5, no. 3, March 2013.

[20] Yu Hua, and Xue Liu, “Scheduling Heterogeneous Flows with Delay-Aware Deduplication for Avionics Applications”, IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 9, September 2012.

Author Profile

V. Manimaran completed his B.E. degree in Computer Science and Engineering from Nandha Engineering College, Erode, India in 2011. He is currently doing his M.E (Computer Science and Engineering) in Nandha Engineering College (Autonomous), Erode, India.

S. Prabhu completed his B.E. degree in Computer Science and Engineering from SSM College of Engineering, Komarapalayam, Salem, India in 2008. He completed his M.E degree in Computer Science and Engineering from Kongu Engineering College (Autonomous), Erode, India in 2010. He is currently pursuing his Ph.D. Programme on the area of cloud computing. Presently he is working as Assistant Professor in Computer Science and Engineering Department in Nandha Engineering College (Autonomous), Erode, India.