International Journal of Research and Advanced Development (IJRAD)

A Hybrid Multi -Resource Dynamic Scheduling and Mapping in IaaS Cloud Systems

P. Suganya[1], M. Vinodha[2], S. Vishnu Priya[3], Mrs. S. N. Sangeetha[4]

[1, 2, 3]IV Year B. E., CSE, [4]Assistant Professor,
[1, 2, 3, 4]Department of Computer Science & Engineering, Velalar College of Engineering and Technology, TN, India.
[1]sibi.suganya@gmail.com, [2]svinopriyas@gmail.com, [3]vishnupriyasridharan@gmail.com

Received Date: 15th March, 2017, Revised Date: 2nd April, 2017, Accepted Date: 12th April, 2017.

Abstract - The core functions of Infrastructure-as-a-Service (IaaS) cloud are Resource allocation and job scheduling. These are based on cloud systems adequate information of available resources. The overall performance of cloud system can be improved by acquiring dynamic resource status information by the user. A cloud system for analysing performance, diagnosing the fault, and maintaining dynamic load balancing, thus, desired computing can be efficiently obtained by the user through dynamic load balancing. Many previous studies are either to improve the performance or for it mainly concentrate on fairness, without considering the importance of both. Recent studies observe that there is a necessary to consider both performance and quality because of resource contention between users/jobs. However, scheduling algorithms for bi-criteria optimization between performance and fairness are static, without considering the impact of different workload characteristics of providers on the trade-off between performance and suitability. A distributed system for cloud resource scheduling and load balancing is a system that uses Dynamic Prediction Scheduling and Mapping (DPSM) technique to allocate resources dynamically based on application job demands. Through the design and implementation of this automated resource management system achieves a good balance between the load and resource scheduling in cloud systems. Finally, the proposed technique presents the better results with increasing number of tasks with good response time in customer stratification in contrast with other earlier system.

Keywords: IaaS, Dynamic Prediction Scheduling and Mapping, Cloud System, Resource Sharing, Virtual Machine,

I. INTRODUCTION

For elastic computing over the Internet, the Infrastructure-as-a-Service (IaaS) cloud system is used as an appealing paradigm. IaaS clouds allow users to access the resources in the form of virtual machines [1]. Today, most IaaS cloud lenders offer some of VM types (such as small, medium, large and extra-large). These virtual machines are allocated by middleware with a fixed amount of CPU, main memory, and disk. Tenants can only purchase fixed- size VMs. Then when the resource demands change the user can change the number of VMs. It can be termed as T-shirt and scale-out model by the cloud. However, this leads to inefficient allocation of cloud resource, which impacts to a higher capital expense and operating amount for cloud providers, and the increase of monetary expense for users. First, the granularity of resource acquisition/release is coarse in the sense that the fixed-sized VMs are not tailored by server for cloud applications with dynamic demands delicately. As a result, users need an over-provision resource (costly), or risk performance penalty and Service Level Agreement (SLA) violation. Second, scale-out model is termed as elastic scaling in clouds, is also costly due to the latencies involved in VM instantiating and software runtime [3].

These costs are ultimately borne by users in terms of performance penalty. Resource sharing is a classic and effective approach to resource efficiency. As adding/removing resource is directly performed on the allocated VMs, fine-grained resource allocation is supported which is also known as a scale-up model, and the cost tends to much smaller compared to the scale- out model. Unfortunately, current IaaS clouds have limitations in resource sharing among VMs, even if those VMs belong to the same tenant. There is a better resource model than the T-shirt model to enable resource sharing for better cost efficiency of users as well for the higher resource utilization of cloud providers [2].

It has witnessed the prosperousness of a group-buying mechanism in real product and service markets. The mechanism of Group-buying offers products or services at discounted prices when the item is bought in a minimum quantity or dollar amount. This market-based mechanism can benefit both cloud users and providers in IaaS cloud systems. This method binds different VMs to form a cooperative group, which even goes beyond a single tenant. This paper work is to propose a distributed system for cloud dynamic resource scheduling and load balancing which present a system that uses dynamic prediction scheduling technique and parallel processing and mapping technique to allocate resources dynamically based on application job demands. By considering dynamic parameters design, implementation and evaluation of an optimized automated resource management system for cloud computing services which achieve a good balance between the load and efficient resource scheduling and mapping in IaaS cloud systems [4].

The challenge is to develop a remote scheduling scheme that enables prediction based resource allocation by middleware to make autonomous decisions while producing a desirable emergent property in the remote system; that is, the two system-wide objectives are achieved simultaneously. The key issues for system implementation, including machine learning based methodologies for modelling are analysed. Then an optimization of resource prediction models is also discussed.

II. RELATED WORKS

Resource sharing is a classic and effective approach to resource efficiency. As applications with diversifying and heterogeneous resource requirements are already deployed in the cloud, there are vast opportunities for the resource. Recent work has shown that fine-grained and dynamic resource allocation techniques (e.g., resource multiplexing or over committing can significantly improve the resource utilization. Cloud computing service providers, make the large-scale network servers form the pool of large-scale virtual resources. In a cloud system, IaaS level is the delivery of hardware as well as the associated software (operating systems virtualization technology, file system), and provides a lot of computational capacities to service remote users. The resources provide at this level are flexible in nature and can be used efficient way.

In IaaS level,

1) The resources have provided in the form of Virtual Machines (VMs). This machines deployed within the cloud equipment consists of Data-centre, physical resources, etc. for fulfilling the requests of the user.
2) Resource management subsystems in IaaS cloud systems are used to schedule incoming tasks in getting the service.
3) Task scheduling is a major process for IaaS clouds. This is mapping the requests on resources by considering cloud characteristics in an efficient manner. It takes VMs as scheduling units for mapping physical heterogeneous resources to tasks.

A. Weighted Max-Min Fairness (WMMF) Algorithm

WMMF is widely used to solve the problem of allocating scarce resource among a set of users. WMMF algorithm defines three principles to allocate resources:

1) Resources are allocated in ascending order of demands normalized by the weight. Thus, if two users have the same weights, the user with a smaller resource demand is satisfied first.
2) No user obtains a resource share larger than its demand. Thus, the over-provisioned portion should be reallocated to other unsatisfied users.
3) Users with unsatisfied demands get resource shares in proportion to their weights. This policy defines how to distribute the over-provisioned resources to unsatisfied users. The outcome of running WMMF is that it maximizes the minimum share of a user whose demand is not completely satisfied.

B. Dominant Resource Fairness (DRF) Algorithm

DRF is used for multiple resource types. This is a generalization of max-min fairness algorithm. The core idea of DRF is that the resource allocated to a user should be determined by the user's share and response on the dominant resource type (or dominant share). The demand of users are satisfied in the ascending order of dominant shares by DRF and thus maximizes the smallest dominant share of users in a system.

1) T-shirt Model

With T-shirt Model, we allocate the total resources to tenants in proportion to their share values of CPU and memory separately. The T-shirt model guarantees that each user precisely receives the resource shares that the user pays for. However, it wastes scare resource because it may over allocate resource to VMs that has high shares but low demand; even other VMs have unsatisfied demand.

C. Problem Description

The communication cost affects the response time to requests as a major factor; these algorithms do not consider it.

1) When the size of the workflow is increased the processing time may become very long.
2) When there exists resource contention, for more benefit a user may lie about her resource demand. So dynamically sharing resources gives rise to resource contention.
3) By some reason such as heterogeneous and dynamic properties of resources, in addition to many numbers of entry tasks with different characteristics, this issue is known to be an NP-Complete problem.
4) The requests will run on all resources and would not support the optimal usage of resources and a proper load balancing.

III. PROPOSED METHOD

A. Reciprocal Resource Fairness (RRF) Algorithm

By considering the resource sharing model in multi-user cloud environments, where each provider may provide several VMs to perform applications, and each VM has multi-resource demands. The key term "multi-resource" specifies the different types of resource, instead of multiple units of the same resource type. This mainly considers the resource type: CPU. Tenants can have different weights (or CPU value) to the resource. The CPU value of a user reflects the load of user's priority about other users. Based on resource complementarily some users can form a resource pool. In proposed allocation model, each part of resource is represented by some shares. The virtual machines of these users then provides the same resource pool with limited resource shares, which are determined by tenants' incentive. The implementation details of key components are:

1) Virtual Machine Grouping: VM grouping algorithm co-locates tenants with complementary resource requirements to increase sharing opportunities. Cloud tenants submit their resource requirements, VM configurations, workload patterns and other preferences or constraints. The VM grouping algorithm then places users in a suitable coalition to enhance the resource multiplexing among different virtual machine users.
2) Demand Prediction: Resource demand is taken by the system as one of the inputs in our algorithm. At run-time estimated periodically at run-time.
3) Resource Allocator: For CPU resource allocation, a convenient interface is provided to configure VMs' CPU weights. CPU cycles allocated to different VMs are proportional to their weights are guaranteed by weighting mechanism.
4) Resource Pool Scaling: This model supports fine-grained (i.e., CPU) resource allocation to VMs via a dynamic schedule.
5) Load Balancing: To handle load imbalance of different resource provider/tenants, our prototype supports present CPU value based VM migration for load management. VM present migration can also facilitate resource trading between tenants.

B. Hybrid Multi-Resource Dynamic Scheduling and Mapping Technique

There are mainly two mechanisms for acquiring information of remote resources (Tenants): remote resource monitoring and remote resource prediction. Remote resource state monitoring cares about the running state, distribution, and system load in remote system by means of monitoring strategies. Remote resource state prediction focuses on the variation trend and running track of resources in remote system using modelling and analysing each resource provider's load i.e., CPU usage. Periodic updating by monitoring and future variation generated by prediction are combined to feed remote system for analysing performance, eliminating diagnosing fault, and maintaining dynamic load balancing, thus to help remote users obtain desired computing results by efficiently utilizing system resources regarding minimized cost, maximized performance or trade-offs between cost and performance. To reduce overhead, the goal of designing a remote resource monitoring and prediction system is to achieve seamless fusion between remote resource and efficient monitoring and prediction strategies. Finally, the scheduler of such a processing framework must be able to determine which task of a job single or multiple should be executed by which type of resource providers and, possibly, how many of those so by using efficient parallel data processing with dynamic resource allocation and load balancing. Figure 1 shows the architecture of the proposed method.

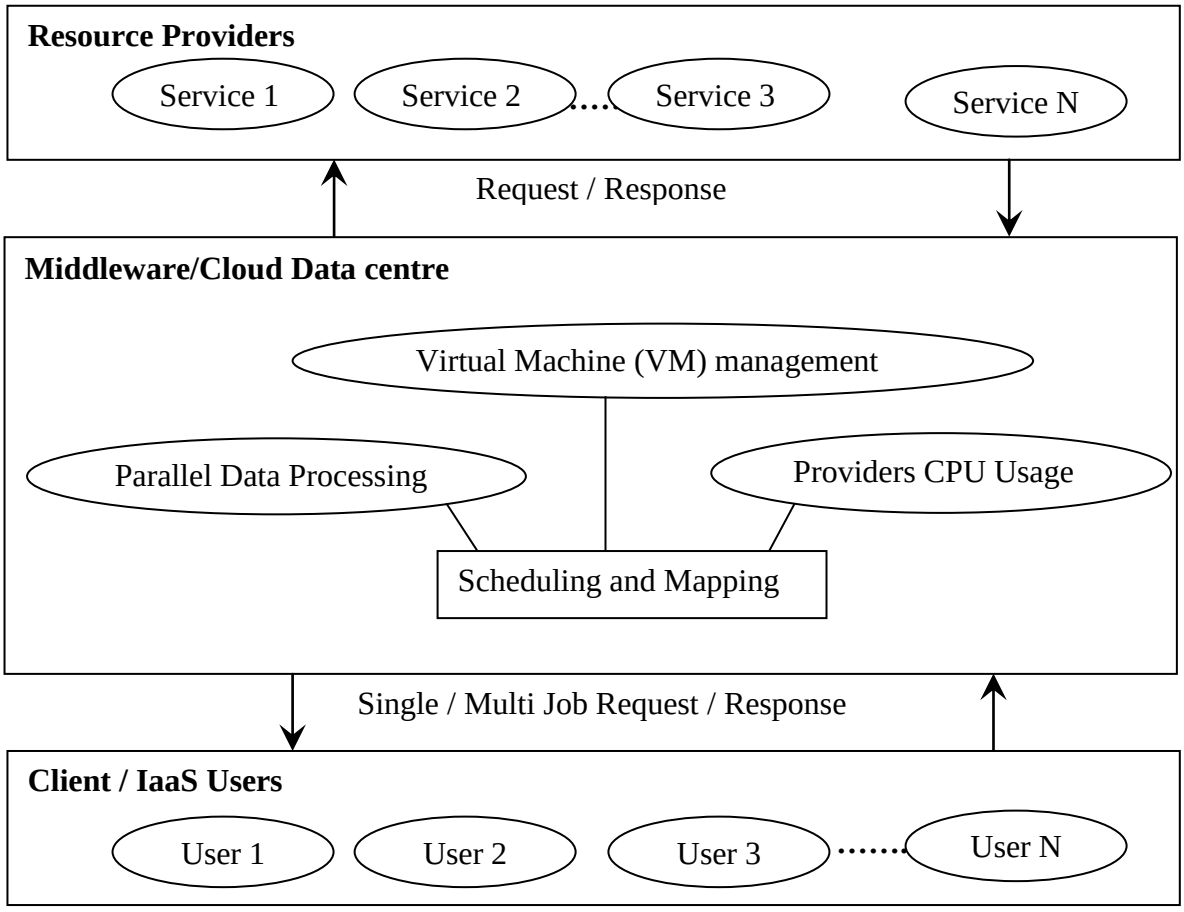

Fig. 1 Architecture of the Proposed Method

IV. RESULTS AND DISCUSSION

Implementation is a stage where the theoretical design is turned into the working system by the developers. The most crucial part is to give the users confidence that the proposed system will works fairly as well as efficiently. The performance of reliability of the system is tested by the middleware, and it gained acceptance.

The whole system divided into two subsystems which are evaluated individually due to the different evaluation criteria. The one among them is monitoring subsystem, which made a comparative study on accuracy and response time between our system and existing system. The design and implementation of an automated resource management system that achieves a good balance between the resource allocation and efficiency is resulted in the proposed method. A resource allocation system is developed that can avoid overload in the provider system effectively while minimizing the number of resources used. The performance of our algorithm is evaluated using real-time work. Note that our execution procedure uses the same code base for the algorithm as the implementation in the experiments. It ensures the fidelity of our experimental results. Both prediction service and evaluation services are used to control the overall prediction method and also used for a multi or long job in parallel.

V. CONCLUSION

Distributed efficient resource scheduling and prediction architecture is proposed that seamlessly combines cloud technologies, resource monitoring and machine learning based resource identification. The proposed system consists of a set of distributed services to provide required resource monitoring, data gathering, and state prediction functions. The challenges for efficient parallel data processing in cloud environments and the first data processing framework to exploit the dynamic resource provisioning offered by today's cloud system were discussed. The performance evaluation describes on how the ability to assign specific virtual machine types to specific tasks of a processing job, and the possibility to automatically allocate or deallocate virtual machines based on the need of a job execution, can help to improve the overall resource utilization and, consequently, reduce the processing cost. The output of this work will contribute to building and advance of computing cloud infrastructure.

REFERENCES

[1] Hadoop Fair Scheduler [Online]. Available: http:// hadoop.apache.org/docs/r1.2.1/fair scheduler.html.

[2] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in Proc. ACM SIGCOMM Conf., 2011, pp. 242–253.

[3] H. Ballani, K. Jang, T. Karagiannis, C. Kim, D. Gunawardena, and G. O'Shea, "Chatty tenants and the cloud network sharing problem," in Proc. 10th USENIX Conf. Netw. Syst. Des. Implementation, pp. 171-184, 2013.

[4] N. Bansal, J. R. Correa, C. Kenyon, and M. Sviridenko, "Bin packing in multiple dimensions: Inapproximability results and approximation schemes," Math. Oper. Res., Vol. 31, No. 1, pp. 31-49, 2006.