Dynamic Resource Parallel Processing and Scheduling by Using Virtual Machine in the Cloud Environment

[1] V. Sriram
Final Student M.E: CSE
M.A.M College of Engineering
Trichy, India

[2] S. Ravimaran
Professor and Head: CSE
M.A.M College of Engineering
Trichy, India

Abstract— ***Resource allocation and job scheduling are cloud computing core functions. These functions are based on abundant information of accessible resources. To ensure overall performance of cloud computing, timely acquiring dynamic resource status information is more important. A cloud method to analyze performance, bottleneck elimination, fault diagnose and dynamic load balancing maintenance. The desired computing results should be obtained to help cloud users by efficiently utilizing the resources in the system in terms of reduced cost, exploited performance or trade-offs among cost also performance. The objective of designing a dynamic resource allocation and prediction system is to achieve seamless fusion between cloud technologies and efficient resource scheduling and prediction strategies is to reduce the overhead. This work aims at building a distributed system for cloud resource scheduling and prediction as a state-of-the-art parallel data processing algorithm. In this virtual machine learning-based methodologies for modeling and optimization of resource prediction models and the efficiency, accuracy of our system meet the demand of online system for dynamic resource utilization and prediction.***

Keywords— ***Cloud Computing, Resource Management, Virtualization, Green Computing, Parallel Processing and scheduling Method.***

I. INTRODUCTION

In cloud environment, skewness algorithm was used in single job execution process in enhance for this project “parallel data processing & scheduling” is used multiple job operation method. Today a growing number of companies have to process huge amounts of data in a cost-efficient way. In order to streamline the development of distributed applications on top of such architectures, several real time enterprises have also built customized data handling frameworks. Today’s handling frameworks usually assume the resources that consist of a static set of reliable nodes. Though it is designed to deal with failure of discrete node, considers the number of existing machines to be constant, specifically when forecast the job execution process. Hence, the compute resources available in a cloud are highly dynamic and possibly heterogeneous. The challenge is to develop a cloud scheduling scheme that enables prediction based resource allocation by middleware to make autonomous decisions while producing a desirable emergent property in the cloud system; that is, the two system wide objectives are achieved concurrently. The key issues for system implementation are discussed, as well as machine learning based methodologies for modeling and optimization of resource prediction models.

There are mainly two mechanisms for acquiring information of cloud resources: cloud monitoring resource and cloud prediction resource. The resource state monitoring of cloud maintains the running state, distribution, and load in the cloud system by means of monitoring strategies. The resource state prediction of cloud focuses on the variation trend and running track of resources in cloud system by means of modeling and analyzing each provider’s load CPU usage. To reduce overhead, the goal of designing a cloud resource monitoring and prediction system is to achieve faultless combination between cloud technologies and efficient resource monitoring and prediction strategies.

A. *Monitoring Systems*

At present, there are a lot of monitoring tools that have been widely used. These tools are usually designed for monitoring individual personal computer or single group, so that they can't be used directly in grid systems. However, monitoring technologies hired and the resource sensors recognized are refillable for grid schemes. Besides, these tools are being developed to support grid system, along with the rapid development and wide spread application of grid computing. In addition, many grid projects have designed the monitoring module of their own. Several representative monitoring systems are introduced.

B. *Prediction Systems*

Resource monitoring can only provide the instantaneous information of grid node however; it cannot afford to generalize the dynamic variation principle of resources. Such gap can be filled by resource prediction. Lots of Grid middle-wares are born with prediction component, while many efforts are dedicated on integrating prediction tools with projects without such factors. Some descriptive prediction efforts are introduced as follows: RPS (Resource Prediction System) project is a resources-oriented system for online prediction and scheduling.

IJERT

It carries on explicit prediction based on the resource signal, and realizes time series models to predict resource information of hosts. NWS (Network Weather Service) project is a distributed system for generation and publication of computing resources prediction, periodically and dynamically. It maintains a group of distributed performance sensors, such as CPU sensors, network sensors, etc. NWS collects information from these sensors on calculating nodes and predicts usage of resources in firm time interval in advance, using several models such as mean, median centred one and autoregressive method.

II. RELATED WORK

In recent years, growing number of companies have to process huge amounts of data in a cost-efficient manner. In order to simplify the development of distributed applications on top of such architectures, many of real time companies have also built customized data processing frameworks.

It is desirable to avoid wasting resources as a result of under-utilization and to avoid lengthy response times as a result of over-utilization. The main contribution of this effort stands two-fold. The distributed architecture is adopted where resource management is decomposed into self-determining tasks, each one of which is executed by Independent Node Agents that are tightly coupled with the physical machines in a data center. Second, the Autonomous Node Agents carry out configurations in parallel through Multiple Criteria Decision Analysis using the PROMETHEE method. In this paper, a new approach for dynamic autonomous resource management in computing clouds is introduced. This approach consists of a distributed architecture of NAs that perform resource configurations using MCDA with the PROMETHEE method. This approach comes from its ability to easily change the weights of criteria and adding/removing criteria in order to change configuration goals.

Cloud computing technology is increasingly being used in enterprises and business markets. In cloud paradigm, an effective resource allocation strategy is required for achieving user satisfaction and maximizing the profit for cloud service providers. It summarizes the classification of RAS and its impacts in cloud system. The mainly focus on CPU memory resources but are lacking in some factors Hence this paper will hopefully motivate future researchers to come up with smarter and secured optimal resource allocation algorithms and framework to strengthen the cloud computing paradigm.

The focus on grid resource monitoring and prediction, representative monitoring and prediction systems are analyzed and evaluated, then monitoring and prediction strategies for grid resources are summarized and discussed, recommendations are also given for building monitoring sensors and prediction models. During problem definition, one-step-ahead prediction is extended to multi-step-ahead prediction, which is then modeled with computational intelligence algorithms such as neural network and support vector regression. The efforts can be utilized as direction for building online monitoring and prediction system for grid resources. The basic problems in grid computing, and there is no doubt that resource performance is the most influencing factor within such area then how to measure the resource monitoring and prediction.

In a grid environment, resources are independent, wide-area disseminated, and they are usually not free. The two events in the grid are resource consumers and resource providers. The goal of the work is to build a global computational grid that every participant has enough incentive to stay and play in it. Thus the performance objective of scheduling is two-fold: for customers, great effective execution speed of jobs also for providers and allocation of benefits in fairly manner. For resource consumers, they are satisfied with the quality of the job execution service and will continue to buy computational resources in the grid for jobs. For resource providers, they make reasonable amount of profit and are willing to deploy their computational resources into the grid for sale.

The performance objective of scheduling is to guarantee the incentive of the two parties, thus two-fold. To resource consumers, the high successful execution rate of jobs can be adopted as a performance metric. A successful execution means that a job is finished without missing its deadline. To resource providers, fair allocation of benefits is a reasonable performance metric. Fair allocation of benefits means that the profit of every provider is proportional to its investment or cost.

The scheduling problem in markets is like computational grids. Many previous research projects focused on optimizing traditional performance metrics, like system utilization, system load balance, and application response time in controlled grids. They did not consider market-like grids, where providing sufficient incentives for participants is a key issue. In existing system all user must need to buy a every software what they want to develop or run. Some software's are very costly sometimes system capacity not enough still we need to buy those software's. The main challenge, phrased as a scheduling problem is to schedule jobs of consumers to resources of providers to optimize incentives for both parties.

IJERT

III. PROPOSED SYSTEM MODEL

In cloud environment, the designing and evaluating the system architecture for cloud resource scheduling and prediction. The challenge is to develop a cloud scheduling scheme that enables prediction based resource allocation by middleware to make autonomous decisions while producing a desirable emergent property in the cloud system; that is, the two system wide objectives are achieved instantaneously. The key issues for the implementation of the system include machine learning based methods for modeling and optimization of resource prediction models.

Advantages

- The dynamic allocation of resource and its implementation in cloud with good performance metric values.
- It improves the overall utilization of server resources.
- The evaluation of performance gives a first impression on how the ability to assign specific

virtual machine types to specific tasks for processing job as well as the possibility to automatically allocate/de-allocate virtual machines in the course of a job execution.

- It aids to improve the overall utilization of resource consequently, reduce the processing cost.

A. Modules Description

1) *Cloud Client or Consumer*
2) *Cloud Virtual Machine or Middleware*
3) *Cloud Resource Provider*
4) *Job Parallel Scheduling on Multi Subtask*
5) *Job Mapping and Virtual Machine Response*

1) Cloud Client or Consumer

Client is the system having service or job request with the server. Client need to make a connection with the server or middleware. Using that connection client submit a job to the server and waits until get the response from the server. A client does not share any of its possessions, but it requests content of the server or function of service. Thus the client inductee communication sessions with servers which await (*listen* to) incoming request?

2) Cloud Virtual Machine or Middleware

Client-server cloud computing or networking is a distributed application architecture that partitions tasks or workloads between resource providers and requesters of service called consumers. The details of the client and the providers are maintained by the system called virtual machine or middleware. It also maintains the information of each connected providers of resource, which is used for scheduling of job, selection of resource provider and also for splitting of task.

3) Cloud Resource Provider

Providers are the process to satisfy the request from the clients and it is varied in configuration of system, workloads, and performance. Provider need to register their profile with the virtual machine, thus virtual machine can able to allocate the job under the request of client. Then the machine gets the providers short list, current CPU usage which is used for prediction to allocate the task.

4) Job Parallel Scheduling on Multi Subtask

Virtual machine collects the job request from client and split the job into sub tasks then the respective providers are selected through their prediction method based on requested resource provider's availability. To the selected providers; each subtask is allocated parallel by the virtual machine. After the allocation of task, the process execution in each provider is monitored by the virtual machine and sends the response to client.

5) Job Mapping and Virtual Machine Response

After completion of sub task, server maps into task and deliver the task output to the particular client. Hence through this method inclusive utilization of cloud as well as the performance is measured perfectly and profitably.

IV. SYSTEM OVERVIEW

The architecture of the system is presented in Fig.1. In the above figure, the set of users to requesting multiple services to virtual machine this request submitted to VM. Then VM Checks the status of service providers whether it is free .This jobs are allocated to Service Providers who is ready to process this are executed parallel and it may reduce the CPU usage.

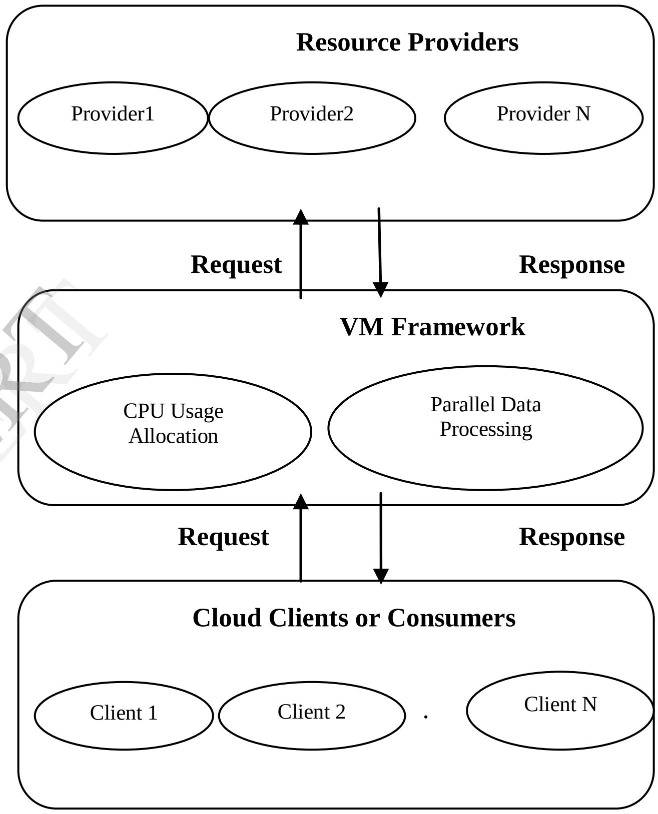

Fig.1. System Architecture

Server or virtual machine plays a role between client and the resource providers. Before client job request, server gets the resource providers details like resource name and related details with its current system connectivity which is used for prediction based dynamic resource allocation.

Client submits job to the middleware instead of provider selection done by client, then middleware or server or virtual machine schedule the job through predicting system load (which as system CPU usage) and providers of resource short list. The server allocates the job to the provider based on system low CPU usage of various providers whose capacity i.e. the current usage of CPU is lower.

Then server collects the result from the providers and sends to the respective client. Thus client waiting time or response time must be reduce and also each provider

IJERT

utilization time and energy must be less so overall performance must be better than the existing work.

V. VIRTUAL MACHINE BASED DYNAMIC SCHEDULING TECHNIQUE

Schedulers that effectively predict the availability of constituent resources, and use these predictions to make decision on scheduling, can progress performance of application. Disregarding resource availability characteristics can lead to longer application make spans due to wasted operations. Even more directly, it can adversely affect application reliability by favoring faster but less available resources that cannot complete jobs before becoming unavailable or being reclaimed from the cloud by their owners. Unfortunately, performance and reliability vary inversely; favoring one necessarily undermines the other. For this reason, Cloud schedulers must consider both reliability and performance in making scheduling decisions. This requirement is true for current Clouds, and will become more important as resource characteristics become progressively miscellaneous.

To determine resource reliability the resource availability predictors can be used and this in turn can be combined with measures of resource performance while decide on resources. Subsequently performance depends on challenging resource load during the application lifetime, the load monitors and predictors make use of by the schedulers, which can be centralized, or can be distributed using various dissemination approaches. In scheduling for performance and consistency, it is perilous to examine the effect on both application make span and the number of wasted operations due to application evictions, which can occur when a resource executing an application becomes unavailable. Evictions and wasted operations are important because of their direct “cost” within a Cloud economy, or simply because they essentially deny the use of the resource by another local or Cloud application.

The proposed approach to virtual machine scheduling involves analyzing resource availability and predicting future resource (un)accessibility, observing also current load bearing, storing static resource capability information, and considering all of these factors when placing applications. For scheduling, the best availability predictor is used to investigate the effects of weighing resource load, and reliability in a variety of ways, to decrease makes span and increase application reliability. This project reports the effort aiming at building a system for cloud resource monitoring and prediction.

A. *Procedure*

1. Initialize A = 1,2,… k to hold all provider details
2. For each client request c_i get the matched providers among the providers in A set.
3. if job. equals (single)
4. Compute the CPU usage be c_p= a1, a2… k as request to providers in the set A.
5. Order the provider a1 to k CPU usage present in the hash table.
6. Allocate the job to the low cpu usage provider as p1 then get the output O_t and display it in client C_i
7. Else
8. Compute the resource name as R_n from client request and download the multiple files in the request C_i
9. if R_n. equals(Resource)
10. Count the active providers as A_p having java resource and allocate the files parallel to the providers in the list of hash table.
11. Map the output from the providers and generate the single output as O_i
12. Connection request to client C_i and sends O_i to the client and displays it.
13. else if(other resource)
14. Count the active providers as A_p having requested resource and allocate the files parallel to the providers in the list of hash table.
15. Map the output from the providers and generate the single output as O_i
16. Connection request to client C_i and sends O_i to the client and displays it.
17. end for
18. Output the window in the client C_i

VI. CONCLUSION

In the cloud environment the challenges and opportunities for efficient dynamic resource processing in cloud environments are discussed. The performance evaluation gives a first impression on how the ability to assign specific virtual machine types to specific tasks of a handling job, also there is a possibility to automatically allocate/ de-allocate virtual machines in the course of execution of a job. It can help to increase the inclusive utilization of resource subsequently, reduces the cost of process and capacity. The variety of open research issues in which the plan is to address the future work is to improve the resource overload or underutilization adaption ability during job execution. The present reporting approach builds a valuable basis yet; the system still requires a realistic amount of user annotations. In general, the work represents an important contribution to the growing field of Cloud computing services and points out exciting new opportunities in the field of parallel data processing.

IJERT

REFERENCES

[1] Yagiz Onat Yazir, Chris Matthews, Roozbeh Farahbod, "Dynamic Resource Allocation in Computing Clouds Using Distributed Multiple Criteria Decision Analysis," 2013 IEEE Sixth International Conference on Cloud Computing, pp. 91-98, 2010 IEEE 3rd International Conference on Cloud Computing, 2010

[2] V V. Vinothina, Dr.R.Sridaran and Dr.Padmavathi Ganapathi, “A Survey on Resource Allocation Strategies in Cloud Computing” International Journal of Advanced Computer Science and Applications (IJACSA) , 3(6), 2012.

[3] Si-Qing Zheng, Xi-Long Che, Liang Hu, "Online System for Grid Resource Monitoring and Prediction Strategies," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 1, pp. 134-145, January, 2012

[4] Lijuan Xiao, Yanmin Zhu, Lionel M. Ni, "Incentive-Based Scheduling for Market-Like Computational Grids," IEEE Transactions on Parallel and Distributed Systems, vol. 19, no. 7, pp. 903-913, July, 2008.

[5] Danak.A, Montreal, Mannor.S "Efficient Bidding in Dynamic Grid Markets" IEEE Transactions on Parallel and Distributed Systems, vol.22, no.9 pp. 1045-9219, Sep, 2011.

[6] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast Transparent Migration for Virtual Machines," Proc. USENIX Ann. Technical Conf. 2005.

[7] C.A. Waldspurger, "Memory Resource Management in VMware ESX Server, "Proc. Symp. Operating Systems Design and Implementation (OSDI '02), Aug. 2002.

IJERT