iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** III **Month of publication:** March 2015

DOI:

Volume 3 Issue III, March 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Fine Grained Interference and Deadline Aware Scheduling For Data-Intensive Applications in Virtualized Environments

P.Sujatha[#1], Dr.V.Venkatesakumar[*2], S.Ayyasamy[#3], A.Nelson[#4]
[#2] *Assistant Professor, Dept of CSE, Anna University Regional Center, Coimbatore*
[#134] *P.G Scholars, Dept of CSE, Anna University Regional Center, Coimbatore*

Abstract— Nowadays many of the organization make use of virtualized environment of cloud computing to improve their resource sharing. When sharing the resources for task scheduling particularly for data intensive applications interference becomes major issue. In the existing work TRACON framework is developed to attain the interference aware scheduling of information intensive applications. Once the task is submitted by the user, the user also will specify the Quality of Service needs. It will be difficult to realize associate user such Quality of Service constraints once the user submitted jobs are knowledge intensive application. The data intensive application will consume more time for execution when it is submitted by the cloud service provider. It is overcome in our work by proposing a novel algorithm called interference aware group based job scheduling. This algorithm aims to group the tasks with the similar deadline and that will assigned based on the priority with the consideration of interference. The experimental result demonstrates that our proposed algorithm is better than the existing algorithms.
Keywords— Cloud Computing, Scheduling, TRACON,, Cloud service provider.

I. INTRODUCTION

Cloud computing is a recent trends in IT that moves computing and data away from desktop and portable PCs into large data center. It refers to applications delivered as services over the Internet to the actual cloud infrastructure, hardware and systems software in data center that provides these services. The key driving forces behind cloud computing is the ubiquity of broadband and wireless networking, falling storage costs, and progressive improvements in Internet computing software. Cloud-service clients will be able to add more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers will increase utilization via multiplexing, and allow for larger investments in software and hardware. Currently, the main technical underpinnings of cloud computing infrastructures and services include virtualization, service-oriented software, grid computing technologies, management of large facilities, and potter efficiency. Consumers purchase such services in the form of infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), or software-as-a-service (SaaS) and sell value-added services to users. Organizations can provide hardware for clouds internally, or a third party can provide it externally. Within the cloud, the laws of probability give service providers great leverage through statistical multiplexing of varying workloads and easier management as a single software installation can cover many users' needs. Two differ-ent architectural models are used for clouds: the first one is designed to scale out by providing additional computing instances on demand. Clouds can use these instances to supply services in the form of SaaS and PaaS. The second architectural model is designed to provide data and compute-intensive applications via scaling capacity. In most cases, clouds provide on-demand computing instances or capacities with a "pay-as-you-go" economic model.

A cloud comprises processing, network, and storage elements, and cloud architecture consists of three abstract layers. Infrastructure is the layer and is a means of delivering basic storage and compute capabilities as standardized services over the network. Servers, storage systems, switches, routers, and other systems handle specific types of workloads, from batch processing to server or storage augmentation during peak loads. The middle platform layer provides higher abstractions and services to develop, test, deploy, host, and maintain applications in the same integrated development environment. The application layer is the highest layer and features a complete application offered as a service. Cloud computing is an umbrella term used to refer to Internet based development and services.

Cloud Services can also be used to hold structured data. There has been some discussion of this being a potentially useful notion possibly aligned with the Semantic Web, though concerns, such as this resulting in data becomes undifferentiated, have been raised.

II. BACKGROUND

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Virtualized data centres are the most common cloud computing platforms. In this work, we focus on Xen and its notable paravirtualization technique, where the Xen VMM works as a hardware abstraction layer to guest operating systems with the modified kernels. Note that Xen also supports hardware-assisted full virtualization that emulates the host hardware for unmodified operating systems. In paravirtualization, the VMM is in charge of resource control and management, including CPU time scheduling, routing hardware interrupt events, allocating memory space, etc. In addition, a driver domain (Dom0) that has the native drivers of host hardware performs the I/O operations on behalf of guest domains (DomU). Fig. 1 depicts a typical Xen I/O architecture, where each guest domain uses a virtual device driver (frontend) in cooperation with a corresponding module (backend) and the native driver in the driver domain to accomplish I/O operations.

III.VIRTUAL MACHINE (VM)

A virtual machine (VM) could be a computer software implementation of a machine is executing programs sort of a physical machine. Virtual machines are a unit separated into 2 major categories, supported their use and degree of correspondence to any real machine. A system virtual machine provides a whole system platform that supports the execution of a whole software package (OS).These typically emulate associate existing design, and area unit engineered with the aim of either providing a platform to run programs wherever the important hardware isn't on the market to be used (for example, death penalty on otherwise obsolete platforms), or of getting multiple instances of virtual machines resulting in additional economical use of computing resources, each in terms of energy consumption and price effectiveness and also known as hardware virtualization. A process virtual machine (also known as language virtual machine) is meant to run one program, which suggests that it supports one method. Such virtual machines are a unit typically closely suited to one or additional programming languages and engineered with the aim of providing program movability and adaptability (amongst different things). A necessary characteristic of a virtual machine is that the computer code running within is restricted to the resources and abstractions provided by the virtual machine cannot flee of its virtual surroundings. Benefits of Virtual Machines are tempting and live migration, template is nothing but create an OS with application VM, use it to create multiple instances of that combination and Live migration is the movement of a running VM from one host to another and therefore no interruption of user access can occur. The various virtual machine scheduling mechanisms are discussed below in the evaluation of performance and to improve network utilization. It has advantages as better address I/O bound applications, understand how different application mixes affect the overall benefits of our solution, and if workload heterogeneity can be exploited to make better use of cloud hardware under QoS constraints. VMM will observe scheduling order and CPU consumption for every task from mixed overloads, then it schedules the task to IO task and takes minimum CPU time as an IO bound task.

IV.THE CHALLENGES WITH EXISTING TECHNIQUES

In the existing work, TRACON framework is implemented to provide the interference aware scheduling for the both the task and resources. This framework consists of three steps: the interference prediction model infers the application performance from the resource consumption observed from multiple VMs.The interference-aware schedulers is designed to utilize the model and generate optimized assignments of tasks and physical resources; and the task and resource monitor collects application characteristics at the runtime and feeds to both the model and scheduler. The existing system possesses several limitations and they are given as follows,

It cannot complete the execution within the user specified deadline and Increases the system overhead.

V. TRACON SYSTEM ARCHITECTURE

Gh Cloud technology provides the QOS to the user with less response time with heterogeneous systems. To improve the features of this environment, in this work we propose a new heuristic algorithm interference aware group scheduling (Priority Based on Deadline and Size) which increases the throughput of the overall system by providing the immediate response to the jobs based on the priority. Jobs with earliest deadline with short burst time will be executed by the resources first. Short jobs are benefited with this method. It also decreases the system overhead by preventing the task migration between the processors and system power is not wasted unnecessarily in context switching between the processes.

CSP: Cloud Service Provider keeps the information of user jobs which includes QOS like granule size of process, deadline and maximum processing capacity needed for the job.

Resource manager: Major role of resource manager is to schedule the user job to the available resources in resource pool environment based on the QOS maintained by CSP submitted by user. Resource managers also called as brokers of the network.

Bag of tasks submitted by user maintained in CSP. Resource manager checks the needed service of user and allocate the resources .Deadline is considered as one of QOS of user. Task set ŕi ,0<i<=n tasks , Di,0<i<=n(deadline) and Ěi,0<i<=n(execution time) are taken as a input from the user. Then tasks with the same deadline are grouped as the task set. In a group of the tasks with the similar deadline, tasks are sorted in ascending order based on execution time and arrival time. Task set group also sorted and the tasks in the group1 assigned to the resources first and till the resources availability is >0 tasks will be allotted. When resource availability is = 0 task will be in waiting stage. After the resources finished the allotted task next group allotted and the process continued till all the groups of tasks are executed successfully. Here response time of the tasks is reduced and missed deadline ratio also reduced which increase the overall system throughput. The key advantages of proposed system is Key generation is distributed manner, Product sensitive data, Noisy data upload is controlled and Resource and task allocation is distributed

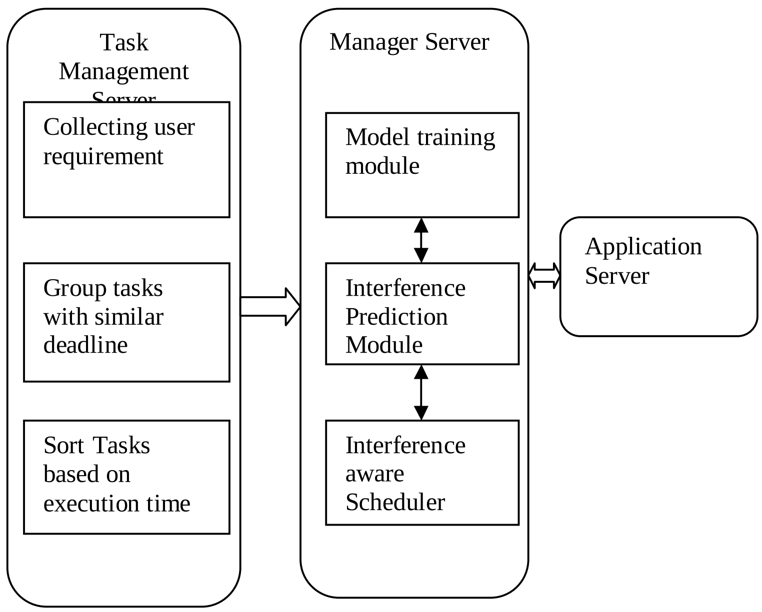


Fig. 1 TRACON system architecture

VI. INTERFERENCE PREDICTION MODEL

On the high level, the interference can be perceived as the changes in the performance, including the total runtime as used in prior work , and I/O throughput that we have shown is critical to data-intensive applications. In this work, we construct the interference prediction models in order to extrapolate the application performance as a function of the resource consumption by virtual machines. We apply several regression analysis techniques that are commonly used for modeling the relationship between an observed response and controlled variables. For the data-intensive applications that consume a significant amount of I/O bandwidth and CPU cycles, we choose to characterize each application by four key parameters (con-trolled variables), the read throughput, the write throughput, the local CPU utilization in the current guest VM domain (DomU), and the global CPU utilization in the virtual machine monitor (Dom0). The first two parameters measure the I/O workload from the target application in terms of the number of requests per second, and the third is used to model the CPU consumption from data processing of the application. While a model with these three parameters is a straightforward approach to reason about an application, such a model is not sufficient to achieve high accuracy for a virtualized environment, which is the reason that we introduce the fourth parameter, the global CPU utilization in the virtual machine monitor. Intuitively, because all the requests from guest VMs are routed through Dom0, it is crucial to properly account for the CPU consumption from the I/O handling tasks that are performed in DomU, as well as in Dom0 that acts on physical devices on behalf of DomU.

It reduces the running time and increase the IO throughput for data incentive applications in a virtualized environment. The performance of the co-located data incentive applications will be analyzed for task and resource allocation control (TRACON) which is a framework that will mitigate the interference effects between data incentive concurrent applications. TRACON will use modeling and control techniques on the basis of statistical machine learning techniques. It contains three components that are an interference prediction model, interference aware scheduling and task and resource monitor. To identify interference effects from concurrent data intensive applications and improve overall system aware scheduling and task and resource monitor [3]

A. The Interference Prediction Model

This model will automatically infer effects when various levels of concurrent IO operations occur and it controls the situation of low overhead system tools and reduce unnecessary interference from system monitoring. It collects the application characteristics of performance and resource utilizations of various virtual machines.

B. The Interference Aware Scheduler

Interference aware scheduler will assign the tasks to physical resources based on the interference prediction model which infers the application performance based on resource consumption on multiple VMs. It will generate a number of possible assignments depending on incoming task and the number of available which can communicate to an interference prediction module that will use constructed modules, application profiles and machine status to find interference effects for a given assignment [3].

This model is designed to utilize effective resource management Interference aware scheduling is designed to utilize efficient resource management for that various scheduling strategies and machine learning techniques can be used. The machine learning technique avoids large interference and reduces total interference. There are two machine learning based schedulers can be used in this design which are k-means++ and Doubling where both techniques are based on clustering algorithm in a virtual environment.

C. Task and resource monitor

The task is assigned to each VM can be managed by the task and resource allocation monitor after that application characteristics and performance interference are collected these information will be used as feedback to updated interference prediction model. This module observes CPU and resource utilizations and requirements of the different client customer to perform scheduling operations.

VII. FRAMEWORK

A. Cloud Setup

In this module, the cloud initialization will be done. The Host setup, Virtual Machine setup with different requirements will do in order toallcate the user submitted task efficiently. The number of jobs to be executed will be gathered from the user with the user specified requirements. The deadline and task capacity requirements will also be gathered from the users. Manger server will be build in this module who will be responsible for monitoring and predict the interference states of the cloud service providers.

B. Grouping Jobs and assigning priority

In this module, the jobs with the same deadline requirement will be grouped together. And the jobs in the group will be aligned based on the arrival time and the execution time. After sorting the tasks based on the execution time, the task set itself will be sorted in order to give preference to the tasks which is having minimum deadline and execution time. And then the tasks in the group which is having minimum deadline will be allocated to the resources first.

- Input: ŕ1 …ŕn =Bag of tasks,
- Ď1…Ďn, =Deadline
- Ě1 …Ěn = Execution Time,
- Þ1… Þn = Processors
- Sort Task set ŕ based on Deadline Ď
- Group tasks with same deadline
- Sort task in group based on execution time.

C. Finding possibility of resource allocation

Once the jobs are grouped together, the scheduler will generate a number of possible assignments based on the incoming tasks and the list of available VMs, which will then be communicated to the interference prediction module. That is in this module, the possible assignments are calculated. Consider 3 tasks and 2 VM. Tasks 1 can be allocated to either one of this VM’s like VM1, VM2, and VM3. This scheduling group the VMs and their application in order to share cache miss. This dynamic VM scheduling use live migration to minimizing shared caches and memory controllers.

D. Interference prediction module

This number of possible assignment will be send to the interference prediction module. This module uses the constructed models, the application profiles, and the machine status to predict the interference effects for the given assignments. For the data-intensive applications that consume a significant amount of I/O bandwidth and CPU cycles, each application is characterized by four key parameters (controlled variables): the read throughput, the write throughput, the CPU utilization in the current guest VM domain, and the CPU utilization in the virtual machine monitor. The first two parameters measure the I/O workload from the target application in terms of the number of requests per second, and the third is used to model the CPU consumption from data processing of the application. While a model with these three parameters is a straightforward approach to reason about an application, such a model is not sufficient to achieve high accuracy for a virtualized environment, which is why introduce the fourth parameter, the CPU utilization in the virtual machine monitor.

E. Interference aware scheduling

Once the interference prediction is done, this module will schedule the incoming tasks to different virtual machines in a way that minimizes the interference effects from co-located applications. In general, optimally mapping tasks to machines in parallel and distributed computing environments has been shown to be an NP-complete problem. Specifically, our proposed model aims to meet the user specified deadline in the fine grained manner as well as reduce the runtime and improve the I/O throughput for data-intensive applications in a virtualized environment.

F. Task/Resource monitoring

In module the task and resource monitors will manage the on-going tasks that are assigned to each VM and collect application characteristics and the performance interference. In a simple example where two VMs are to share the host hardware, the monitor measures the resource utilization of both VMs via existing system tools (e.g., xentop and iostat) while the assigned tasks are running. Xentop is used to monitor and record the physical CPU utilization of each domain, because a top command in each domain, including the driver domain, can only get the domain's virtual CPU utilization. In addition, because a driver domain (Dom0) performs the I/O operations on behalf of each guest domain, we use iostat in Dom0 to monitor and record the resource utilization of physical storage devices. Note that the resource utilization monitoring in Dom0 is preferable because system administrators generally do not have access to each guest domain for security and privacy reasons. The collected information is used as feedback to update the interference prediction model. It will identify the cache behaviour of the VMs, based on that VMs can be relocated to improve the memory behaviour.

VIII. CONCLUSION

Cloud computing provides the quality of service to the user with the available heterogeneous systems which are loosely coupled and can be reconstructed or scalable in nature. Managing the resources and deploying the jobs submitted by user to concern resource and complete the task set with maximum bandwidth and speed is the essential criteria of cloud. Virtualization has become the key for data center to achieve excellent resource utilization, scalability, and high availability. The interference aware scheduling is concerned to provide the efficient task assignment on the cloud providers in which the tasks can be allocated without interferences. The deadline aware provisioning also concentrated when allocating tasks without violating the interference effects.

REFERENCES

[1] Qian Zhu, Jiedan Zhu, Gagan Agrawal, "Power-aware Consolidation of Scientific Workflows in Virtualized Environments", High Performance Computing, Networking, Storage and Analysis (SC), PP: 1 – 12, 13-19 Nov. 2010

[2] Jing Xu and José Fortes, "A Multi-Objective approach to Virtual Machine Management in Datacenters", Advanced computing and information system laboratory, 2010

[3] Timothy Wood, Ludmila Cherkasova, Kivanc Ozonat, and Prashant Shenoy, "Profiling and Modeling Resource Usage of Virtualized Applications", Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, Pages 366-387, 2008

[4] Chuliang Weng, Zhigang Wang, Minglu Li, and Xinda Lu, "The Hybrid Scheduling Framework for Virtual Machine Systems", Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, Pages 111-120, 2009

[5] Xiaodan Wang, Eric Perlman, Randal Burns, Tanu Malik, Tamas Budav´ariz, Charles Meneveaux and Alexander Szalay, "JAWS: Job-Aware Workload Scheduling for the Exploration of Turbulence Simulations", Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, Pages 1-11, 2010

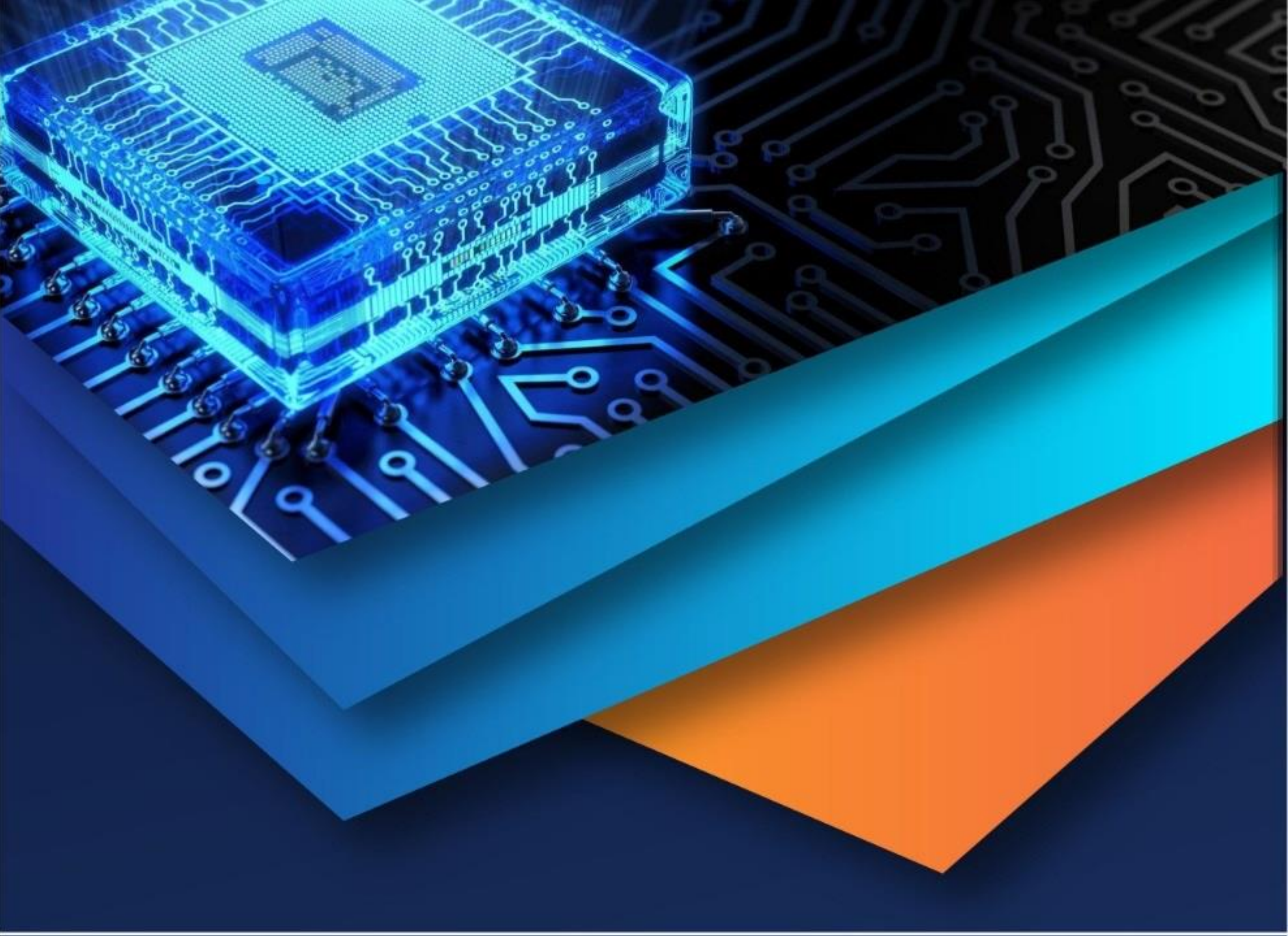
[6] Xiaorui Wang and Ming Chen, "Cluster-level Feedback Power Control for Performance Optimization", The 14th IEEE International Symposium on High-Performance Computer Architecture, 2008

[7] Fengguang Song, Asim YarKhan, Jack Dongarra, "Dynamic Task Scheduling for Linear Algebra Algorithms on Distributed-Memory Multicore Systems", Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, 2009

[8] Will Sobel, Shanti Subramanyam, Akara Sucharitakul, Jimmy Nguyen, Hubert Wong, Sheetal Patil, Armando Fox, David Patterson, "Cloudstone: Multi-Platform, Multi-Language Benchmark and Measurement Tools for Web 2.0", Lecture Notes in Computer Science Volume 5931, 2009, pp 254-265

[9] Pradeep Padala, Kai-Yuan Hou, Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant , "Automated Control of Multiple Virtualized Resources", Proceedings of the 4th ACM European conference on Computer systems, Pages 13-26, 2009

[10] Alireza Ghaffarkhah, Ripal Nathuji and Aman Kansal, "Q-Clouds: Managing Performance Interference Effects for QoS-Aware Clouds", Proceedings of the 5th European conference on Computer systems, Pages 237-250, 2010



10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY