Volume 4, Issue 5, May 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Migration Improved Scheduling Approach In Cloud Computing

Devinder Saini
M.Tech CSE Department
Kurukshetra University

Er. Ravi Malik
HOD, ECE Department
Geeta Engineering College

Abstract— The cloud is a next generation platform that provides dynamic resource pools, virtualization, and high availability. Cloud computing is a construct that allows you to access applications that actually reside at a location other than your computer or other internet-connected device. With the increasing use of cloud computing concepts, the load over the cloud server is also increasing. To achieve the effective use of cloud resources and to improve the service execution and retrieval effectively some efficient scheduling mechanism is required.

Keywords: Cloud Computing, Scheduling, Process Switching, Deadline Orientation

I. INTRODUCTION

Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth. A simple example of cloud computing is Yahoo email, Gmail, or Hotmail etc. You don't need a software or a server to use them. All a consumer would need is just an internet connection and you can start sending emails. The server and email management software is all on the cloud (internet) and is totally managed by the cloud service provider Yahoo, Google etc. The US National Institute of Standards and Technology (NIST) has developed a working definition that covers the commonly agreed aspects of cloud computing. The NIST working definition summarizes cloud computing as:
"A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

II. ARCHITCTURE

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services.The architecture of Cloud computing can be categorized according to the three types of delivery models, namely Infrastructure as a service (IaaS), Software as a service (SaaS) and Platform as a service (PaaS).

A. Infrastructure as a Service (IaaS)

Infrastructure as a Service is a single tenant cloud layer where the Cloud computing vendor's dedicated resources are only shared with contracted clients at a pay-per-use fee. This greatly minimizes the need for huge initial investment in computing hardware such as servers, networking devices and processing power.

B. Software as a Service (SaaS)

Software as a service is where computer applications are accessed over the Internet rather than being installed on a local computing device or in a local data centre. SaaS is becoming an increasingly prevalent delivery model as underlying technologies that support Web services and service-oriented architecture (SOA) mature and new developmental approaches, such as Ajax, become popular. The availability of IaaS services is a key enabler of the SaaS model. Information security officers will need to consider various methods of securing SaaS applications. Web Services (WS) security, Extendable Markup Language (XML) encryption, Secure Socket Layer (SSL) and available options which are used in enforcing data protection transmitted over the Internet.

C. Platform as a Service (PaaS)

Platform as a service cloud layer works like IaaS but it provides an additional level of "rented" functionality. Clients using PaaS services transfer even more costs from capital investment to operational expenses but must acknowledge the additional constraints and possibly some degree of lock-in posed by the additional functionality layers

. Platform as a Service (PaaS) is an outgrowth of Software as a Service (SaaS), a software distribution model in which hosted software applications are made available to customers over the Internet.

III. CHARACTERISTICS OF CLOUD COMPUTING

A. Dynamic Scalability

A key benefit of cloud computing is the ability to add and remove capacity as and when it is required. This is also known as elasticity. Cloud computing is **dynamically scalable** because users only have to utilize the amount of online computing resources they actually required.

Cloud computing is task-centric

Cloud computing is task centric because it totally depends on the task the user wants to achieved and not on particular software, hardware or network infrastructure. Users do not have to purchase or install anything before using a cloud computing resource. They do not have to pay during the periods when they are not using the resources. It eliminates the need to install the software to achieve the task to be done.

B. Cloud computing is independent of devices used

Device Independence in the context of cloud computing means the freedom or power of cloud users to access their data or files in different devices or gadgets whether it is mobile phone, personal computer or any similar gadgets .Cloud computing is by now facilitate better device independence, excellent portability, and greater probability for interconnection and partnership. It is very simple and practical to make possible for users to access applications and information located in the cloud systems in spite of their place or what tool or machine they are utilizing.

C. Resources reuse, low cost

As cloud computing adopts statistical multiplexing technology of resources, IT physical resources are increasingly used in a large scale, thus, the cost of cloud business is greatly reduced.

D. Service-oriented

Cloud computing systems are all service oriented i.e. the systems are such that they are created out of other different services. Many such cloud computing different individual services combine together to form this service. Other such services can be created by using services that were just created. There are two keys to achieve the service oriented conception i.e. abstraction and accessibility.

Cloud computing is programmable

Many necessary task of cloud computing must be automated. for e.g., to protect the integrity of the data, information stored on a single computer in the cloud must be copied on another computer in the cloud . If that computer goes offline, the clouds programming automatically redistributes that computer's data to a new computer in the cloud.

IV. SCHEDULING

Scheduling is the basic concept used by any traditional computer system to line up the multiple processes in specific sequence for execution. The scheduling approach is fruitful if it provides the effective utilization of available resources as well as provides the management of different resources accordingly. The scheduling scheme is defined to design a dynamic system so that the process will be executed effectively and the waittime of these processes will be reduced. The scheduling is used to simplify the complexity of multiple processes by arranging them in specific sequence under different process analysis properties. Such as controlling the heterogeneity of processing, managing the prioritization of processes, provide the fault tolerance etc.

Scheduling mechanism is the most important component of a computer system. Scheduling is the strategy by which the system decides which task should be executed at any given time. There is difference between real-time scheduling and multiprogramming timesharing scheduling. It is because of the role of timing constraints in the evaluation of the system performance.

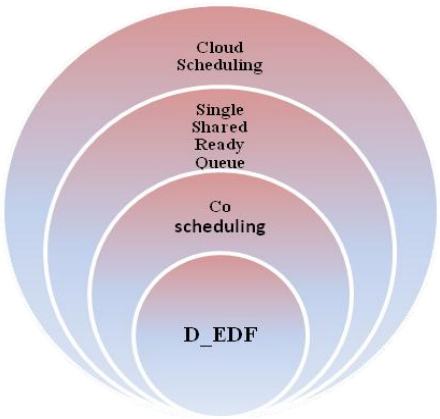

Figure : scheduling schemes

A. Cloud Scheduling

The main reason of using a Cloud system is to improve the performance of large-scale application programs, high scalability and fault tolerance. Therefore, it is essential to keep the cost of creating and managing parallelism as low as possible.
As for scheduling in a Cloud system, there are three goals to lower down the cost:

- Good processor utilization: all processors have work in their queues at all times. All processors, which have tasks assigned to them from the same program finish execution at the same time, thus the user gets the expected speedup. Processors spend most of their time doing useful work rather than coordination the division of work.
- Good synchronization effectiveness: Tasks are scheduled in such a way that interacting tasks across with fine-grained interaction should be running at the same time.
- Low communication/memory-access cost: Tasks are scheduled in such a way that communication time, either message passing or shared-memory latency is accounted for, and minimized. Scheduling data structures should be arranged so that they are no a source of contention.

B. Single Shared Ready Queue

It is a simple approach. A single global queue shared by all the processors in the system. Whenever a processor is available and the queue is not empty, the processor will be scheduled with a new task. The scheduling policies such as First Come First Serve (FCFS) and Shortest Job First (SJF) can be easily implemented. [9] It seems as if this approach may yield good processor utilization. However, this approach does not provide any mean to schedule fine-grained interacting tasks to run at the same time. Consider the following scenario: Task A already scheduled for a processor, but it mush wait for completion of Task B, which is still in the queue. Even though Task A keeps the processor busy, all it does is to wait. This approach also has potential high communication/memory access cost, because this global queue occupies a region of memory that can be accessed by all the processors simultaneously.

C. Co Scheduling

In co scheduling, all run able tasks of an application are scheduled on different processors simultaneously. Without coordinated scheduling, the processes constituting parallel tasks may suffer communication latencies because of processor thrashing. In recent years, researchers have developed parallel scheduling algorithms that can be loosely organized into three main classes, according to the degree of coordination between processors: explicit or static co scheduling, local scheduling, and implicit or dynamic co scheduling.

D. Explicit Co Scheduling

Explicit co scheduling ensures that the scheduling of communication jobs is coordinated by creating a static global list of the order in which jobs should be scheduled and then requiring a simultaneous context-switch across all processors. This may be accomplished statically by agreeing upon a global schedule in advance or dynamically by having a "master" local scheduler direct other schedulers by communicating with them at each context switch. However, this approach has quite a few drawbacks. Since it requires identifying parallel tasks in the application in advance, it complicates the implementation. Furthermore, it interacts poorly with interactive use and load imbalance.

E. Local Co Scheduling

Conversely, local scheduling allows each processor to independently schedule its processes. Although attractive due to its ease of construction, the performance of fine-grain communicating jobs degrades significantly because scheduling is not coordinated across processor

F. Implicit Co Scheduling

An intermediate approach, implicit co scheduling, allows each of the local schedulers to make decisions independently, but relies on local schedulers to take the communication behavior of local processes into account when making decisions. Local schedulers can converge on co scheduling behavior since each sees similar or related communication behavior by local processes that are part of parallel applications. There are many forms of implicit co scheduling developed over the years.

G. D_EDF

Deadline Monotonic is fixed-priority scheduling algorithm whereas EDF is dynamic-priority algorithm. A fixed-priority scheduler assigns the same priority to all instances of the same task, thus the priority of each task is fixed with respect to other tasks. However, a dynamic-priority scheduler may assign different priorities to different instances of the same task, thus the priority of each task may change with respect to other tasks as new task instances arrive and complete.

V. Present Work

A. Scope

Cloud demand and cloud resource utilization are factors that most of the IT industries and other organization will demand the most in future. Cloud computing technology is the future of computing and virtualizes and offers many services across the network .Cloud is purely a dynamic environment and the existing task scheduling algorithms are mostly static and considered various parameters like time, cost, make span, speed, scalability, throughput, resource utilization, scheduling success rate and so on. Available scheduling algorithms are mostly heuristic in nature and more complex, time consuming and does not consider reliability and availability of the cloud computing environment. Therefore there is a need to implement a scheduling

algorithm that can improve the availability and reliability in cloud environment. The virtualization technique along with the scheduling algorithm will yield higher resource utilization, system throughput, thus improving the performance of the cloud resources.

B. Problem formulation

A scheduling algorithm for resource allocation of tasks, which will take care of live migration and priority of various processes in cloud environment.

C. Objectives

1. To improve the availability and reliability in cloud environment.
2. The cloud infrastructure sample network will be formed with the server based services for an effective scheduling algorithm which schedules both the task and the resources.
3. Two important factors Priorities and migration of resource will be considered for designing the algorithm.

D. Research Methodology

Different methodologies and measures which are used in previous studies of cloud computing will be summarized. A scheduling algorithm for task scheduling and resource allocation will be designed which will perform task scheduling and allocation of tasks and will take care of live migration & priorities.

A scheduling algorithm for task scheduling and resource allocation will be designed which will perform task scheduling and allocation of tasks and will take care of live migration & check which cloud suits the modern technology and can be used to provide valuable information on the count performance.

VI. PROPOSED WORK

The proposed system is a middle layer architecture to perform the cloud allocation in case of under load and overload conditions. The over load conditions will be handled by using the concepts of process migration. The middle layer will exist between the clouds and the clients. As the request will be performed by the user this request will be accepted by the middle layer and the analysis of the cloud servers is performed by this middle layer. The middle layer is responsible for three main tasks

1. Scheduling the user requests
2. Monitor the cloud servers for its capabilities and to perform the process allocation
3. Process Migration in overload conditions

VII. CONCLUSION

In this present work, a resource allocation scheme on multiple clouds in both the under load and the over load conditions. As the request is performed by the user, certain parameters are defined with each user request, these parameters includes the arrival time, process time, deadline and the input output requirement of the processes. The cloud environment taken in this work is the public cloud environment with multiple clouds. Each cloud is here defined with some virtual machines. To perform the effective allocation, we have assigned some priority to each cloud. The virtual machines are here to perform the actual allocation. These are defined with certain limits in terms of memory, load etc. As the allocation begins, at first the scheduling of the processes is performed respective to the memory requirements. And along with it, the allocation of the process is done to the cloud based on the requirement and the availability analysis. If the allocated process cannot be executed in its required time slot, in such case the migration of the process is required. The migration of the processes is here defined in case of overload conditions.

The overload condition is defined in terms of simultaneous processes that are required to execute at particular instance of time. The analysis of the work is done in terms of wait time, process time of the processes. The obtain results shows the successful execution of all the processes within time limit. The work is performed on a generic system that can have n number of clouds.

REFERENCES

[1]. Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, "Cloud Computing, A Practical approach"

[2]. Ilango Sriram, Ali Khajeh-Hosseini ,"Research Agenda in Cloud Technologies", 2010.

[3]. Jeongseob Ahn, Changdae Kim, Jaeung Han, Young-ri Choi†, and Jaehyuk Huh,"Dynamic Virtual Machine Scheduling in Clouds for Architectural Shared Resources", 2011 IEEE.

[4]. Loganayagi.B, S.Sujatha,"Enhanced Cloud Security by Combining Virtualization and Policy Monitoring Techniques", 2011 Published by Elsevier Ltd.

[5]. Xun Xu, "From cloud computing to cloud manufacturing", 2011 Elsevier Ltd.

[6]. Masaya Yamada, Yuki Watanabe, Saneyasu Yamaguchi,"An Integrated 1/0 Analyzing System for Virtualized Environment, 2011.

[7]. Omer Khalid, Ivo Maljevic, Richard Anthony, Miltos Petridis, Kevin Parrott, Markus Schulz,"Deadline Aware Virtual Machine Scheduler for Grid and Cloud Computing",2010 IEEE.

[8]. Zheng Hu, Kaijun Wu, Jinsong Huang,"An Utility-Based Job Scheduling Algorithm for Current Computing Cloud Considering Reliability Factor",2012 IEEE.

[9]. Omer Khalid, Ivo Maljevic, Richard Anthony, Miltos Petridis, Kevin Parrott, Markus Schulz, "Dynamic Scheduling of Virtual Machines Running HPC Workloads in Scientific Grids" 2009 IEEE.

[10]. Chuliang Weng, Zhigang Wang, Minglu Li, and Xinda Lu,"The Hybrid Scheduling Framework for Virtual Machine Systems", 2009 ACM.

[11] Jie Zheng," Workload-Aware Live Storage Migration for Clouds", VEE'11, March 9–11, 2011, Newport Beach, California, USA. ACM 978-1-4503-0501-3/11/03.

