International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 1, Issue 1, May-June 2012 ISSN 2278-6856

Resource Allocation and Scheduling in the Cloud

Ms. Shubhangi D. Patil[1], Dr. S. C. Mehrotra[2]

[1]Department of IT
Government Polytechnic, Jalgaon
sdkirange@gmail.com

[2]Department of Computer Science
Dr. Babasaheb Ambedkar Marathwada University,
Aurangabaad

Abstract: ***Recently, there has been a dramatic increase in the popularity of cloud computing systems that rent computing resources on-demand, bill on a pay-as-you-go basis, and multiplex many users on the same physical infrastructure. These cloud computing environments provide an illusion of infinite computing resources to cloud users so that they can increase or decrease their resource consumption rate according to the demands. At the same time, the cloud environment poses a number of challenges. Two players in cloud computing environments, cloud providers and cloud users, pursue different goals; providers want to maximize revenue by achieving high resource utilization, while users want to minimize expenses while meeting their performance requirements. However, it is difficult to allocate resources in a mutually optimal way due to the lack of information sharing between them. Moreover, ever-increasing heterogeneity and variability of the environment poses even harder challenges for both parties.This paper reviews certain papers on resource management and job scheduling in cloud computing.***

Keywords: cloud computing, resource management.

1. INTRODUCTION

"A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers."

The computing resources, either software or hardware, are virtualized and allocated as services from providers to users. The computing resources can be allocated dynamically upon the requirements and preferences of consumers. Traditional system-centric resource management architecture cannot process the resource assignment task and dynamically allocate the available resources in a cloud computing environment. Since the consumers may access applications and data of the "Cloud" from anywhere at any time, it is difficult for the cloud service providers to allocate the cloud resources dynamically and efficiently.

2. PROBLEMS OF RESOURCE MANAGEMENT IN THE CLOUD

In cloud computing, the underlying large-scale computing infrastructure is often heterogeneous, not only because it's not economic and reliable to procure all the servers, network devices and power supply devices in one size and one time, but because different application requires different computer hardware, e.g. workflow extensive computing might need standard and cheep hardware; scientific computing might need specific hardware other than CPU like GPU or ASIC.

There are kinds of resources in the large-scale computing infrastructure need to be managed, CPU load, network bandwidth, disk quota, and even type of operating systems. To provide better quality of service, resources are provisioned to the users or applications, via load balancing mechanism, high availability mechanism and security and authority mechanism. To maximize cloud utilization, the capacity of application requirements shall be calculated so that minimal cloud computing infrastructure devices shall be procured and maintained. Given access to the cloud computing infrastructure, applications shall allocate proper resources to perform the computation with time cost and infrastructure cost minimized. Proper resources shall be selected for specific applications.

1. RESOURCE ALLOCATION AND JOB SCHEDULING IN THE CLOUD: A LITERATURE STUDY

1.1 Agent-based elastic Cloud bag-of-tasks concurrent scheduling [1]

In [1] J. Octavio Gutierrez-Garcia, a family of 14 Cloud scheduling heuristics based on the remaining allocation times of Cloud resources is proposed. The scheduling heuristics consist of two phases: task ordering, where tasks are ordered prior to execution (when possible), and task mapping, where tasks are mapped to available (unoccupied) Cloud resources.

The scheduling and execution of bag-of-tasks applications (BoTs) in Clouds is performed on sets of virtualized Cloud resources that start being exhausted right after their allocation disregarding whether tasks are being executed. In addition, BoTs may be executed in potentially heterogeneous sets of Cloud resources, which may be either previously allocated for a different and fixed number of hours or dynamically reallocated as needed.

The Cloud scheduling heuristics are adapted to the resource allocation settings (e.g., 1-hour time slots) of Clouds by focusing on maximizing Cloud resource utilization based on the remaining allocation times of Cloud resources. Cloud scheduling heuristics supported by information about BoT tasks (e.g., task size) and/or Cloud resource performances are proposed. Additionally, scheduling heuristics that require no information of either Cloud resources or tasks are also proposed. The Cloud scheduling heuristics support the dynamic inclusion of new Cloud resources while scheduling and executing a given BoT without rescheduling. Furthermore, an elastic Cloud resource allocation mechanism that autonomously and dynamically reallocates Cloud resources on demand to BoT executions is proposed. Moreover, an agent-based Cloud BoT scheduling approach that supports concurrent and parallel scheduling and execution of BoTs, and concurrent and parallel dynamic selection and composition of Cloud resources (by making use of the well-known contract net protocol) from multiple and distributed Cloud providers is designed and implemented.

1.2 The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment [2]

Resource allocation is a complicated task in cloud computing environment because there are many alternative computers with varying capacities. In [2] Daji Ergu and Gang Kou proposed a model for task-oriented resource allocation in a cloud computing environment. Resource allocation task is ranked by the pairwise comparison matrix technique and the Analytic Hierarchy Process giving the available resources and user preferences. The computing resources can be allocated according to the rank of tasks. Furthermore, an induced bias matrix is further used to identify the inconsistent elements and improve the consistency ratio when conflicting weights in various tasks are assigned.

1.3 Swarm scheduling approaches for work-flow applications with security constraints in distributed data-intensive computing environments [3]

The scheduling problem in distributed data-intensive computing environments has become an active research topic due to the tremendous growth in grid and cloud computing environments. As an innovative distributed intelligent paradigm, swarm intelligence provides a novel approach to solving these potentially intractable problems. In [3], the scheduling problem for work-flow applications with security constraints in distributed data-intensive computing environments is formulated and a novel security constraint model is presented. Several meta-heuristic adaptations to the particle swarm optimization algorithm are introduced to deal with the formulation of efficient schedules. A variable neighborhood particle swarm optimization algorithm is compared with a multi-start particle swarm optimization and multi-start genetic algorithm.

1.4 Scheduling strategies for optimal service deployment across multiple clouds [4]

Figure 1 shows the Cloud brokering architecture proposed in [4].The architecture components' functionality is the following: the Cloud manager periodically collects information about instances availability and instances price for each instance in the database. It obtains this information from each particular cloud provider and acts as a pricing interface for users, updating the database when new information is available. This is specially useful in dynamic price case, in which it is necessary to have these prices updated. The Scheduler is responsible for making the placement decision.

The architecture has two main actors: the administrator and the user of the cloud broker. The former adjusts the broker configuration options (available clouds, instances types from each cloud, pricing information, etc.) before the execution's beginning; and the latter receives information from the broker and specifies a new service to deploy among available clouds, describing it through a service description file. A service is a set of components each one composed by a number of virtual machines, a scheduling strategy, an optimization criteria, and some particular restrictions

Figure 1 The Cloud brokering architecture

Contributions:

- A novel cloud broker architecture adapted to multi-cloud environments, which acts as a cloud management software and is aware of different cloud features. This broker is aimed to deploy multi-tier services among available cloud providers.
- One of the main components of the broker architecture is the cloud scheduler, which is responsible for making autonomously scheduling decisions based on dynamic pricing schemes,

dynamic user demands, and different instance type performance.

- The scheduler can be configured to work with different scheduling policies based on different optimization criteria, such as service cost, service performance, etc. According to these policies, the scheduler performs an optimal deployment of the service components among different cloud providers trying to optimize a particular cost function.

1.5 Profit-driven scheduling for cloud services with data access awareness [5]

Resource sharing between multiple tenants is a key rationale behind the cost effectiveness in the cloud. While this resource sharing greatly helps service providers improve resource utilization and increase profit, it impacts on the service quality (e.g., the performance of consumer applications). In [5] the reconciliation of these conflicting objectives by scheduling service requests with the dynamic creation of service instances is addressed. Specifically, the scheduling algorithms attempt to maximize profit within the satisfactory level of service quality specified by the service consumer.

Contributions:

The major contributions include

- the development of a pricing model using processor-sharing for clouds (i.e., queuing delay is embedded in processing time),
- the application of this pricing model to composite services with dependency consideration,
- the development of two sets of service request scheduling algorithms, and
- the development of a prioritization policy for data service aiming to maximize the profit of data service.

1.6 Policy based resource allocation in IaaS cloud [6]

Most of the Infrastructure as a Service (IaaS) clouds use simple resource allocation policies like immediate and best effort. Immediate allocation policy allocates the resources if available, otherwise the request is rejected. Best-effort policy also allocates the requested resources if available otherwise the request is placed in a FIFO queue. It is not possible for a cloud provider to satisfy all the requests due to finite resources at a time. Haizea is a resource lease manager that tries to address these issues by introducing complex resource allocation policies. Haizea uses resource leases as resource allocation abstraction and implements these leases by allocating Virtual Machines (VMs). Haizea supports four kinds of resource allocation policies: immediate, best effort, advanced reservation and deadline sensitive. The work in [6] provides a better way to support deadline sensitive leases in Haizea while minimizing the total number of leases rejected by it. Proposed dynamic planning based scheduling algorithm is implemented in Haizea that can admit new leases and prepare the schedule whenever a new lease can be accommodated. Experiments results show that it maximizes resource utilization and acceptance of leases compared to the existing algorithm of Haizea.

Contributions:

- The proposed algorithm finds multiple slots in addition to finding single slot while scheduling a deadline sensitive lease.
- It also applies two concepts (swapping and backfilling) in addition to preemption, while rescheduling already accommodated leases to make space for a newly arrived lease. Swapping uses the information available about leases to be rescheduled, to decide the order in which they should be rescheduled.
- When swapping and preemption both fails to schedule a lease, the algorithm [6] applies the concept of backfilling. Backfilling can fill up the idle resources, which cannot be filled up by consecutive lease. Backfilling has a disadvantage of requiring more preemption, which increases overall overhead of the system.
- By applying rescheduling when an advance reservation or immediate lease gets rejected, it tries to increase acceptance of leases.

1.7 Dynamic Combination of Genetic Algorithm and Ant Colony Algorithm [7]

For the cloud database route scheduling problem, [7] designed a cloud database route scheduling algorithm according to the dynamic combination of the genetic algorithm and ant colony algorithm. The initial solution got by the Genetic Algorithm was transformed into the pheromone initial value, which was needed by ant colony algorithm, then the optimal solution by the ant colony algorithm was obtained .Genetic control function was set up to control the opportunity of two algorithm's fusion .[7] proposed a reasonable algorithm which could find the required database rapidly and effectively, reduce the dynamical load of cloud database routing, and improve the efficiency of cloud computing.

Contributions:

- In [7], the author combines the dynamic combination of genetic algorithm and ant colony algorithm with the searching in cloud database, and good results are obtained.
- The bigger the nodes scale is, the better the algorithm performs.
- After comparing the fusion algorithm with the ant colony algorithm, the author obtains good simulation result.

1.8 Job scheduling algorithm based on Berger model in cloud environment [8]

In cloud computing, entities are mainly users, resource providers, and scheduling system. The main body that corresponds with them is user tasks, the resources and the

International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 1, Issue 1, May-June 2012 ISSN 2278-6856

scheduling strategy. As shown in Figure 2 , in order to be able to map the theory of distributive justice in Berger model to resource allocation model in cloud computing, it is need to carry on the task classification, fairness function definition of user tasks, the task and resource parameterization, the task and resource mapping, and etc.

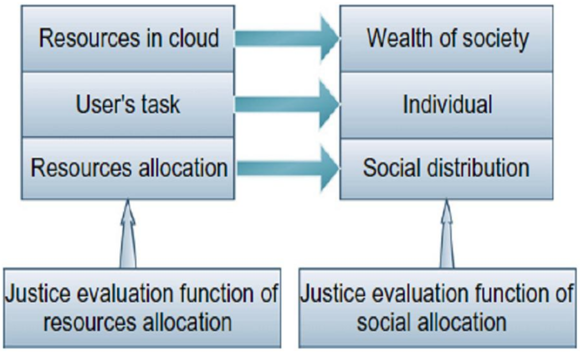

Figure 2 Mapping between cloud and Bergh model

Basic working mechanism of the algorithm provided by [8] is:

(1) According to QoS classification, the general expectation of tasks, which acts as the fairness constraints for selection and allocation of resource, is established.

(2) According to parameterized task characteristics and their corresponding general expectation constraint, select the better resources of virtual machine to run the task.

(3) Calculate the value of fairness justice function based on the results of resource allocation. Statistics user satisfaction and adjust model.

Contributions:

- In this paper, Berger model theory on distributive justice in the field of social distribution was first introduced into the job scheduling algorithm in cloud computing. Through the expansion of CloudSim platform, job scheduling algorithm based on Berger model is implemented.
- The validity of the algorithm is verified on the extended simulation platform. By comparing of simulation results with the optimal completion time algorithm, the proposed algorithm in [8] is effective implementation of user tasks, and with better fairness.

1.9 Community-aware scheduling algorithm [9]

[9] propose a novel decentralized dynamic scheduling approach named the community-aware scheduling algorithm (CASA). In this work, based on the previously proposed two-phase scheduling protocol, a set of heuristic algorithms are designed to efficiently distribute jobs amongst participating nodes without asking for detailed node real-time processing information nor control authorities of remote nodes. The CASA indicates a collection of implemented interfaces and heuristics used to facilitate job scheduling across decentralized distributed nodes. A variety of variables, such as resource heterogeneity and unpredictable job characteristics, are considered. The design of CASA yields a dynamic, adaptive scheduling algorithm to promote job sharing/execution efficiency and improves the job owners' experience.

Contributions:

- The observed experimental results presented show that CASA is able to accomplish the same number of jobs as a centralized approach in a decentralized manner.
- Furthermore, the CASA is able to improve both the average job slowdown and average job waiting time dramatically without asking for neither detailed information of participating nodes nor centralized control of the grid.
- In addition, while applying the CASA with a local perspective information system, both the performance benefit (e.g., job slowdown, job waiting time) and the overhead (e.g., generated/transferred messages) are significantly improved compared to the use of CASA upon a global perspective information system. This phenomenon illustrates that the proposed community-aware scheduling algorithm (CASA) has better performance while working with an information system with faster response time, instead of an information system with integrated global knowledge but slow reaction speed.
- The dynamic scheduling phase of the CASA can compensate for the fault of a local perspective information system by continuous job rescheduling with up-to-date ''local knowledge''.

1.10 Dynamic trusted scheduling for Cloud computing [10]

Figure 3 shows a basic integrated framework based on Cloud-DLS. There are four tiers in this framework:

- the first one is the resource and infrastructure tier, including network, cluster or Cloud computing infrastructure
- the second one is the basic middleware for Cloud computing,
- the third one is the trustworthy scheduler;
- the last one is the client.

In trustworthy scheduler, Schedule Advisor is developed based on Cloud-DLS, and the Trust Model is based on the Bayesian trust evaluation model.

In the trust scheduling based system framework, the whole process of the task submission and execution is in the following:

1. tasks are submitted to the task queue;
2. the task scheduler fetches tasks from the queue and communicates with the schedule advisor
3. the schedule advisor communicates with the trust model;
4. the trust model analyzes the local transactions, communicates with the trust middleware, obtains the detail trust resource

information of task, and transfers them to task scheduler;

5. the task scheduler executes the task on the most trustworthy resource node in the Cloud.

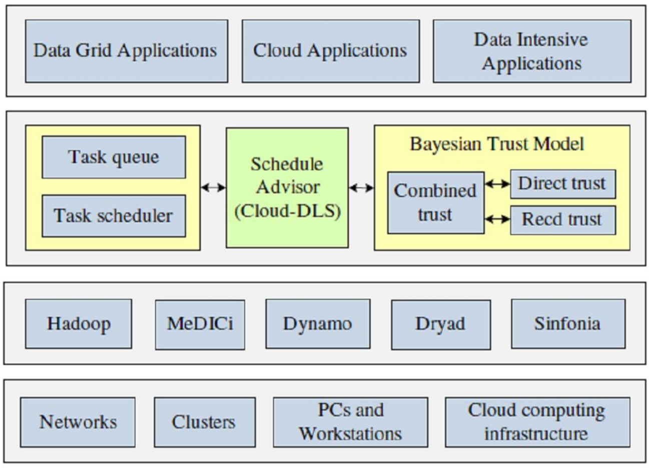

Figure 3 Trusted dynamic scheduling framework.

Contributions:

- A kind of trust mechanism-based trusted dynamic level scheduling algorithm is proposed to decrease the failure probability of the task assignments, and assurance of the execution of tasks in a security environment.
- A comprehensive Bayesian trust model is proposed to evaluate the trust degree of resource for general purpose Cloud computing.
- A benchmark for evaluating the proposed mechanism is developed with various computing patterns which can be utilized in general purpose Cloud environment.
- Simulation experiments proved that the proposed Cloud-DLS algorithm can efficiently reduce the ratio of task failure exertion with a little more time cost.

1.11 Adaptive energy-efficient scheduling [11]

The scheduler in Figure 4 shows the architecture of an energy efficient real-time scheduling mechanism proposed in [11]. The real-time controller and adaptive voltage controller work together and determine if an arriving task in the global queue can be admitted or not. Once the task is accepted, a voltage level will be assigned by the scheduler.

Each node in the cluster maintains a local queue in which admitted tasks are queuing up for execution on the node. The local voltage controller in each node aims at minimizing the voltage levels for admitted tasks to reduce energy consumption. When a new task arrives, the scheduler follows three steps below to allocate and schedule the task.

Step 1: The scheduler checks system status information (e.g.,voltage levels of the nodes, tasks running on the nodes, tasks waiting in the local queues, and actual execution times of finished tasks).

Step 2: The scheduler decides whether or not the new task can be allocated to a node and completed within its deadline by the energy-efficient global scheduling algorithm or EEGS. In the process of scheduling, EEGS makes the best effort to schedule a new task to a node with the possibly lowest voltages. If the new task's deadline is not guaranteed, it will be dropped to the rejected queue. Otherwise the task will be transferred to a destination node.

Step 3: The status information including node voltage, sequence of the new task, execution time of tasks waiting in this node are passed on to this destination node. After a task in one node is executed, the local voltage adjuster relies on the local voltage adjusting algorithm or LVA to dynamically reduce the node voltage subject to the timing constraints of tasks waiting in the local queue. The dynamic voltage scaling approach can achieve high energy efficiency of the heterogeneous clusters.

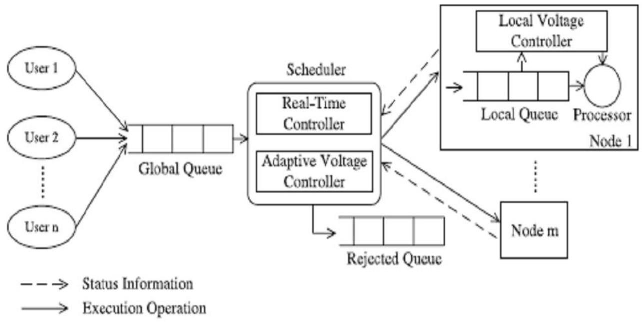

Figure 4 Energy-efficient real-time scheduling architecture

Contributions:

- AEES seamlessly integrates two algorithms—EEGS and LVA. EEGS is implemented in the scheduler that is able to adaptively adjust voltages according to system load to guarantee deadlines of all waiting tasks in local queues. LVA, implemented in the local adjuster, can decrease voltage levels of waiting tasks to conserve energy when a task is scheduled and dispatched to a computing node.
- With EEGS and LVA in place, AEES efficiently improves the adaptivity and schedulability of realtime heterogeneous clusters. The extensive simulation studies using practical system parameters show that AEES is an excellent energy-efficient scheduling strategy designed for DVS-enabled heterogeneous clusters in dynamic environments.
- The AEES algorithm and simulation studies are the first step toward the development of energy-efficient real-time scheduling mechanisms for heterogeneous clusters.

References

1. J. Octavio, Gutierrez-Garcia, Kwang Mong Sim, “A family of heuristics for agent-based elastic Cloud bag-of-tasksconcurrent scheduling”, Future Generation Computer Systems, SciVerse ScienceDirect, doi:10.1016/j.future.2012.01.005
2. Daji Ergu, Gang Kou ,Yi Peng ,Yong Shi, Yu Shi, “The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment”, Springer Science+Business Media, LLC 2011
3. Hongbo Liu, Ajith Abraham, Vaclav Snanel, Seán McLoone, “Swarm scheduling approaches for work-flow applications with security constraints in distributed data-intensive computing environments”, Information Sciences 192 (2012) 228–243
4. Jose Luis Lucas-Simarro, Rafael Moreno-Vozmediano, Ruben S. Montero, Ignacio M. Llorente, " Scheduling strategies for optimal service deployment across multiple clouds”, Future Generation Computer Systems, SciVerse ScienceDirect 2012
5. Young Choon Lee, Chen Wang, Albert Y. Zomaya, Bing Bing Zhoua, “Profit-driven scheduling for cloud services with data access awareness”, J. Parallel Distrib. Comput. 72 (2012) 591–602 ScienceDirect
6. Amit Nathani, Sanjay Chaudharya, Gaurav Somani, “Policy based resource allocation in IaaS cloud”, Future Generation Computer Systems 28 (2012) 94–103 doi:10.1016/j.future.2011.05.016
7. ZHANG Yan-huaa, Feng Leia, Yang Zhia, “Optimization of Cloud Database Route Scheduling Based on Combination of Genetic Algorithm and Ant Colony Algorithm”, Procedia Engineering 15 (2011) 3341 – 3345 2011 Published by Elsevier Ltd.
8. Baomin Xu, Chunyan Zhao, Enzhao Hua, Bin Hu, “Job scheduling algorithm based on Berger model in cloud environment”, ScienceDirect , Advances in Engineering Software 42 (2011) 419–425
9. Ye Huanga, Nik Bessis, Peter Norringto, Pierre Kuonend, Beat Hirsbrunner, “Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm”, Future Generation Computer Systems, 2011 Elsevier B.V. All rights reserved. doi:10.1016/j.future.2011.05.006
10. Wei Wang, Guosun Zeng, Daizhong Tang, Jing Yao, “Cloud-DLS: Dynamic trusted scheduling for Cloud computing”, SciVerse ScienceDirect , Expert Systems with Applications 39 (2012) 2321–2329
11. Xiaomin Zhua, Chuan Hea, Kenli Li, Xiao Qin, “Adaptive energy-efficient scheduling for real-time tasks on DVS-enabled heterogeneous clusters”, J. Parallel Distrib. Comput, SciVerse ScienceDirect, 2012 Elsevier Inc.

Ms Shubhangi Patil received the M.E. degree in Computer Engineering from Swami Ramanand Teerth Marathwada University, Nanded in 2009. During 2003-2011, he stayed in J. T. Mahajan College of Engineering, Faizpur, She is now working as a lecturer in Governemt Ploytechnic, Jalgaon.