Novel Resource Allocation Algorithm for Cloud System That Supports VM-Multiplexing Technology

P.Hema venkataramana, Beebi Sufia Sheik
Dept. of CSE, Kakinada Institute of Engineering & Technology for Women, Korangi, e.g.dt, AP, India

ABSTRACT:
Formulation of a deadline-driven resource allocation problem based on the cloud environment assist with VM resource isolation technology and also suggests a novel solution with polynomial time which could minimize user’s payment in terms of their predictable deadlines. By studying the upper bound of task execution length based on the probably imprecise workload prediction. We further recommend an error-tolerant method to assurance task’s conclusion within its limit. We authenticate its efficiency over a real VM-facilitated cluster environment under special levels of competition. By alteration algorithmic input deadline based on our derived bound, task completion length can at all times be incomplete within its deadline in the sufficient-supply situation. The mean execution length still keeps 70 percent as high as user specified deadline under the severe competition.

KEYWORDS: VM multiplexing, resource allocation, convex optimization, prediction error tolerance, payment minimization.

INTRODUCTION:
The resource allocation in cloud computing is much more compound than in other dispersed systems like Grid computing platform. In a Grid system it is unseemly to share the compute resources among the numerous applications concurrently running atop it due to the predictable mutual presentation interference among them. Whereas cloud systems generally do not stipulation physical hosts directly to users but influence effective resources isolated by VM technology. Not only can such an expandable resource usage way acclimatize to user’s specific demand but it can also maximize resource utilization in fine granularity and isolate the abnormal environments for safety purpose. On the other hand with quick development of scientific research users may suggest quite complicated demands. For instance users may desire to minimize their payments when guaranteeing their service level such that their tasks can be finished before deadlines. Furthermore inevitable errors in predicting task workloads will absolutely make the problem harder. Based on the elastic resource usage model we aim to design a resource allocation algorithm with high prediction error tolerance ability and also minimizing user’s payments subject to their expected deadlines.

RELATED WORK:
With the VM resource isolation technology being established recently it is feasible to plan more proficient resource allocation due to the fledged performance isolation among VMs running on the same machines. Proposal of a VM multiplexing-based resource allocation approach which can productively analyze the compatibility of any two different VMs each with an application running atop it on the same physical machines and rearrange the combination of the VMs to get better the overall performance. Though it cannot assurance high compatibility among more than two VMs on the same machine. Q-Clouds is another well-known system which can understand high consolidation of multiple VM-hosted applications focusing on how to prevent inevitable performance interference among VMs from degrading user’s QoS or enhancing corresponding users payment unexpectedly. This work will certainly benefit and motivate many cloud users or service providers who wish to reduce the infrastructure cost with the guaranteed QoS actually already endeavoured by many researchers.

EXISTING METHOD:
Traditional optimization issues are frequently subject to the precise prediction of task’s characteristic or execution property which is nontrivial to realize in practice. Traditional job scheduling is often originated as a kind of combinatorial optimization
problem or queue-based multiprocessor scheduling problem due to the nonguaranteed performance isolation for multiple tasks running on the same machines. That is most of the existing deadline-driven task scheduling solutions from single cluster environment limited in LAN to the Grid computing environment appropriate for WAN are also severely subject to the queuing model under which a single machine’s multiple resources cannot be further split to smaller fractions at will. This will finally cause the raw-grained resource allocation, relatively low resource utilization and suboptimal task execution efficiency.

DISADVANTAGES:

Users may desire to reduce their payments when guaranteeing their service level such that their errands can be completed before deadlines. Such a deadline-guaranteed resource allocation with minimized payment is rarely studied in literatures. Furthermore inevitable errors in predicting task workloads will absolutely make the problem harder.

PROPOSED METHOD:

We expand our work in and propose an approach to comprehend general fault acceptance mechanisms as independent modules such that each module can obviously function on users’ applications. We then augment each module with a set of metadata that distinguish its fault tolerance properties and use the metadata to select mechanisms that satisfy user’s requirements.

ADVANTAGES:

Delivers an inclusive fault tolerance solution to user’s applications by merging selected fault tolerance mechanisms. Establishes the properties of a fault tolerance solution by means of runtime monitoring.

The Fault Tolerance Manager (FTM) that provides the basis for a service provider to realize the delivery scheme presented in the previous section and hence to offer fault tolerance as a service. The objective is to insert our framework as a dedicated service layer between the client’s applications and the hardware that works directly on the top of the virtual machine manager at the level of VM instances. The Fault Tolerance Manager must deal with the issue of heterogeneity in computing resources fulfill the target of evidently providing fault tolerance support to user’s applications against node failures and convince scalability and interoperability goals. To overcome these challenges we propose to build the Fault Tolerance Manager using the principles of service-oriented architecture where each ft−unit is realized as an individual web service and an ft−sol is formed using the business process execution language (BPEL). We include a resource manager within FTM that initially coordinates with the cloud manager to produce the resource graph and the database.

SYSTEM ARCHITECTURE:

Fig. 1. Resource allocation in cloud system.

TASK ASSIGNMENT:

In cloud systems the cloud proxy frequently obtains and reacts to user requests or tasks with modified requirements or virtual machines. All errands will be handled based on their priorities like Google task scheduler or in terms of First-Come-First-Serve (FCFS) strategy when the tasks are of the same priorities. Each task’s implementation may engage multidimensional resources such as CPU and disk I/O. A data mining task for example typically desires to load a large set of data from disk before or in the middle of its computation. Ultimately such a task
may store its computation results onto the local disk or a public server through network.

**PHYSICAL NODES:**

A physical node is characteristically a server or a virtual machine but it can be any active device attached to a network that is capable of sending, receiving and forwarding information over a communications channel. In other words a physical node is any active device attached to a network that can run a chef-client and also allow that chef-client to communicate with a server.

**VIRTUAL MACHINES:**

In the cloud model any task will be executed on one or more virtual machines with user-reserved resources and the payment is planned based on the customized resource as pay-by-reserve policy. Assuming such a pricing policy is driven by three reasons. First the efficiencies of many applications typically depend on multiple resources but it is nontrivial to precisely evaluate the exact amount of their consumption separately on individual resources. Secondly a few users desire to reserving resources for tolerating usage burst and guaranteeing their service levels. Lastly the alternative pricing policy, pay-as-you-consume, is rather simple due to its payment is always fixed regardless of the resource allocation.

**CLOUD SERVER:**

The users can upload their data in the cloud. We develop this module where the cloud storage can be made secure. Conversely the cloud is not fully trusted by users since the CSPs are very likely to be outside of the cloud users’ trusted domain. Comparably we assume that the cloud server is sincere but inquisitive. That is the cloud server will not unkindly remove or adapt user data due to the protection of data auditing schemes but will try to learn the content of the stored data and the identities of cloud users.

**ALLOCATED AND AVAILABLE RESOURCE:**

Demonstrated optimal for minimizing the payment cost within user-defined deadline for the task. The deadline still may not be guaranteed due to two factors whichever bounded available resources or imprecise workload vector information about the task. We propose effective technique which provides a essential and adequate condition of guaranteeing the task’s deadline given accurate prediction and relatively sufficient re-sources.

---

**ALGORITHM USED:**

**Algorithm 1. Optimal Allocation Algorithm**

Input: $D(t_i)$; Output: execution node $p_x$, $r^*(t_i)$

1. $\Gamma = \Pi, C = D(t_i), r^* = \Phi$ (empty set);
2. repeat
3. $r^*_x(t_i, p_x) = CO-STEP(\Gamma, C); /^{*} \text{Compute optimal } r^*_x \text{ on } \Gamma^* /$
4. $\Omega = d_k|d_k \in \Gamma \& r^*_x(t_i, p_x) > a_k(p_x))$;
5. $\Gamma = \Gamma \setminus \Omega; /^{*} \text{Takes away } \Omega^* /$
6. $C = C - \sum_{d_k \in \Omega} (r^*_x(t_i, p_x), \forall r^*_x(t_i, p_x) \in C^*$
7. $r^*(t_i, p_x) = r^*(t_i, p_x) \cup r^*_x(t_i, p_x);$
8. until ($\Omega = \Phi$);
9. Select the smallest $P(t_i)$ by traversing the candidate solution set;
10. Output the selected node $p_x$ and resource allocation $r^*(t_i, p_x);$

**EXPERIMENTAL RESULTS:**

![Experimental Results](image)
The prediction method we used can make sure that the lower bound of the workload predicted is always lower than the real workload that is considered after its execution. We evaluate our designed algorithm with and without the prediction-error-tolerant support. That is the system will test the Algorithm with the tuned stricter deadline or the original one. We use Deadline Extension Ratio defined as the ratio of task’s final execution time to its deadline to evaluate the statistical task execution lengths compared to their expected deadlines. We run 40 separate cases each with different number 1-40 of tasks and show the lowest/average/highest level of DER for each case.

**FUTURE WORK:**

In the future we plan to integrate our algorithms with stricter/original deadlines into some excellent management tools like Open Nebula, for maximizing the system-wide performance. Some queuing policies like earliest deadline first (EDF) will be studied to further reduce user payment especially in the short supply situation. More complex scheduling constraints like the compatibility and security issue will also be taken into account. When the resources provisioned are relatively sufficient, we can guarantee task’s execution time always within its deadline even under the wrong prediction about task’s workload characteristic.

**CONCLUSION:**

The main objective is to diminish user’s payment on his/her task and also attempt to guarantee its execution deadline meanwhile. We can prove that the output of our algorithm is optimal based on the KKT condition which means any other solutions would absolutely cause larger payment cost. In addition we analyze the estimate ratio for the expanded execution time generated by our algorithm to the user-expected deadline under the probably erroneous task property prediction. When the resources provisioned are comparatively adequate. We can guarantee task’s execution time always within its deadline even under the wrong prediction about task’s workload characteristic. The future strategy involves integrating our algorithms with stricter/original deadlines into some excellent management tools for maximizing the system-wide performance. Some queuing policies like earliest deadline first (EDF) will be considered to further reduce user payment especially in the short supply situation. More complex scheduling constraints like the compatibility and security issue will also be taken into account.

**REFERENCES:**


Ms. P. Hema Venkata Ramana is a student of Kakinada Institute Of Engineering & Technology for Women, Korangai. Presently she is pursuing her M.Tech [Computer Science Engineering] from this college and she received her B.Tech from Kakinada Institute of Engineering & Technology for Women affiliated to JNT University, Kakinada in the year 2012. Her area of interest includes Cloud computing and Object Oriented Programming languages, all current trends and techniques in Computer Science.

Ms. Beebi Sufia Sheik, well known Author and excellent teacher Received M.Tech (SE) from JNT university is working as Assistant Professor, Department of Computer science engineering, Kakinada Institute of Engineering & Technology for women (kiet-w), Korangai. She has 2 years of teaching experience. Her areas of Interest includes, information security, cloud computing Data mining and other advances in computer Applications.