International Journal on Applications in Information and Communication Engineering
Volume 2: Issue 5: May 2016, pp 29-32. www.aetsjournal.com ISSN (Online) : 2394 - 6237

Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds

S.Mohana priya , R.Premkumar

Abstract — The need of a low latency analysis over high velocity data streams motivates the need for distributed continuous dataflow system. Contemporary streams processing systems use simple technique to scale on elastic cloud resources to handles variable data rates. However an application QoS is also impacted with variability in resource performance exhibited by clouds and hence necessitates automatic methods of provisioning elastic resources to support such applications on cloud infrastructures. We develop the concepts of "dynamic data flows" which utilize alternating tasks as additional control over the data flows cost and QoS. Further, we formalize an optimization problem to representing deployment and runtime resources provisioning that allows us to balance the application's QoS, value and the resource costs. We proposed two greedy heuristics, centralized and shared, based on the variable sized been packing algorithm and compare against a Genetic Algorithm (GA) based heuristics that gives a near optimal solution. A large scale simulation study, using the linear roads benchmark and VM performances trace from the AWS public cloud, shows that while GA based heuristic provides a better quality schedule, the greedy heuristics are more practical, and can intelligently utilize cloud elasticity to mitigate the effect of variability, both in input data rates and cloud resource performance, to meet the QoS of fast data applications.

Keywords: QOS, Genetic Algorithm, RSA Algorithm, Heuristic Approach.

I. INTRODUCTION

Cloud computing is a new and emerging trends in distributing computing that facilitate software application platforms, and hardware infrastructures as a service. Cloud service provider offers these services based on customized Service Level Agreements (SLAs), which defined user's required Quality of Service (QoS) parameters. Cloud computing reduces investments on various resource like hardware and software resource allows to be leased and released. It reduces initial investment, maintenance costs and operating cost. Cloud services are hosted on service provider's own infrastructures or on third parties cloud infrastructure providers [1]. Mainly three kinds of services are delivered; Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Services (SaaS). Cloud users using this service, whenever needed to a according to their demands using pay-per-use models [2]. Clouds provided the ability to adjust resources capacity according to the changing demands of the applications, often called auto scaling. However, giving users more controls also required the developments of new method for task scheduling and resource provisioning [3].

Resource management decision is required to cloud scenarios not only have to take into account performance related metrics such as workflows make spam or resource utilizations, but must also considers budget constraints, since the resources from commercial clouds, Usually have a monetary costs associated with them [4].

To gain insight to resource a management challenges, when executing a scientific workflow ensembles on clouds. We address a new and important problems of maximizing the numbers of completed workflows from an ensembles a under both budget and deadline constraint [5].

These systems supports scalability with respect to a high input data rates over static resource deployments, assuming the input rates are stables. When the input rates change, their static resources allocation causes over under provisioning, resulting in wasted resources during a low data rate periods and high data processing latency during high data rate periods. Storm's rebalance function allows to user monitor the incoming data rates and deploys the application on demand across a different set of resources, but requires the applications to be paused [6, 7].

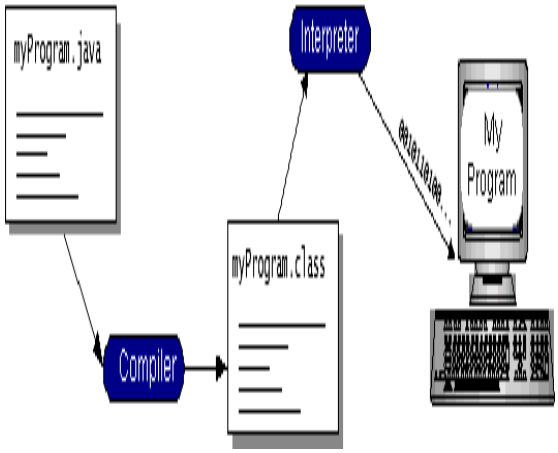

Fig.1 JAVA Working Model

S.Mohana priya , PG Scholar , Department of CSE, AMS College of Engineering and Technology, Namakkal, Tamilnadu .
(Email : mpriya429@gmail.com)
R.Premkumar , Assistant Professor , Department of CSE, AMS College of Engineering and Technology, Namakkal, Tamilnadu .

This can causes the message loss or processing delays during the redeployment. As a result of such systems offers limited in self manageability to changing a data rates, which we address in this articles. Recent SPS such as Esc [8] and stream cloud have a harnessed the cloud's elasticity to dynamically acquires and release resources based on application load, the model enables the system to account for resource interactions in a comprehensive ways, by predicting the effects of planned resource allotments and placement choices. For examples, the models can answer questions like: "how much a memory is needed to reduce this service's storage access to the rate by 20%? Figure 2 depicts the use of the models within the utility OS executives [9]. MBRP is a departure from a traditional resource management using a reactive heuristics with limited assumptions about application behaviour. Our premise is that MBRP is appropriate for a utility OS because it hosts a smaller number of distinct applications that are both heavily resource intensive and more predictable in their average prerequest resource demands [10].

II. WORK FLOWS IN CLOUD

Besides cloud workflow methods be an executed in grids also. But due to the complexity of the environment in grids, execute workflows in clouds is more promising as clouds offered less complex environments than grids. Cloud services likes storage, compute and bandwidth are available at much lower cost. Scalability is the prime benefit which is achieved if workflows are moved to cloud. Scalability allows real time provisioning of resources to meet workflows requirement.

This makes it easy to satisfy Quality of Service (QoS) requirement of application as an opposite to grid approach which require prior reservations of resource in global multiuser environments. Workflow applications typically required complex execution environments, which are difficult to implements on grid. Moreover, each grid sites may have different kinds of configuration which results in extra efforts each time when an application is ported to a new site. Virtual machines allowed to developers to create the fully customized execution environments, configured in specifically for the application in hand.

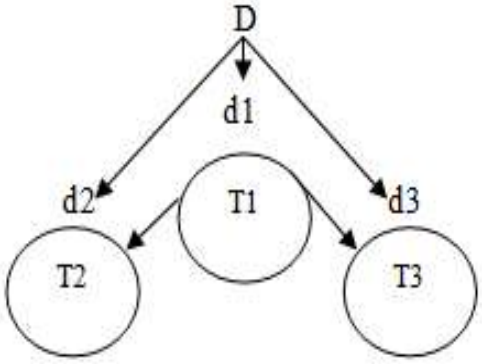

Fig.2 Deadline distribution

Workflows can be represented as a Directed Acyclic Graph (DAG) in which each node represented a tasks and the edge between corresponding nodes represents data dependency betweens tasks. Workflow scheduling is a key concern in workflow management systems. Workflows scheduling is the problem of mapping of tasks on suitable resource, while satisfying the constraints imposed by the users. Proper workflows scheduling can have a significant impact on the performance of the workflow application [11].

A typical example of work flow includes bank account verification, insurance claim processing, business scenarios and online banking. Here we describe the example of bank account verifications workflow processing to shows the characteristics of workflows. In bank account verification, first of all, user enters even passwords, then system retrieve original passwords associated with account [12]. User password is verified with original passwords match with is original password, system allows account access otherwise shows the warning message and request is denied. Fig 1 shows the simplified bank account verifications process which can be modeled as a simple workflows with several steps.

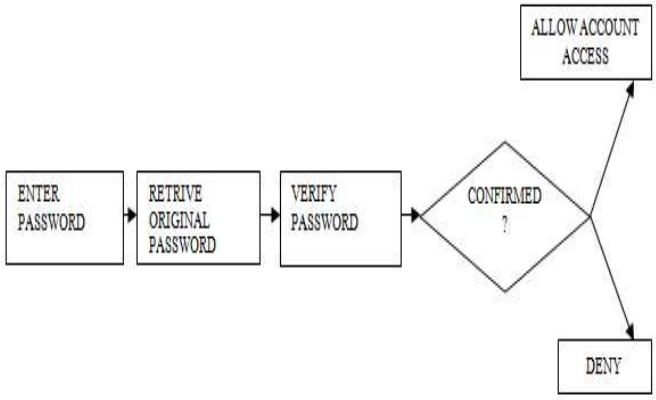

Fig.3 Simplified account verification workflow process

1) The Java Platform

A *platform* is a hardware or software's environment in which a programs run. We have already mentioned in a some of the most popular platforms likes Windows 2000, Linux, Solaris, and Mac OS. Most platform can be describes as a combination of the operating systems and hardware [13]. The Java platform differed from most other platforms in that a software only platform that runs on top of other hardware based platforms.

The Java platforms have two components:

- The *Java Virtual Machine* (Java VM)
- *Java Application Programming Interface* (Java API)

They already introduced to the Java VM. It's the based for the Java platforms and is ported onto a various hardware based platforms.

The Java API is a large collection of readymade software component that provide many usefully capabilities, such as a graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these

libraries are known as *packages*. The next section, The Java Technologies highlights what functionality some of the packages in the Java API provide.

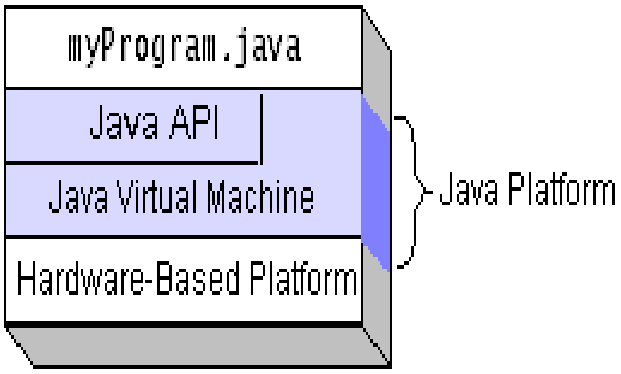

Fig.4 Program Runs In Java Platform

The following figure's depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulated the program from the hardware's.

III. PROPOSED APPROACH

The COATES Algorithm:

We will describe our algorithm for a text clustering with side information. We refer to these algorithms as a COATES throughout the projects, which corresponds to a fact that it is content and Auxiliary attribute based Text Clustering Algorithm. We assumed that an input to the algorithm is the number of clusters k. As in the case of all text clustering algorithm, it is assumed that stop words have been removed, and stemming has been performed in order to improve the discriminatory power of the attributes.

IV. RESULTS AND DISCUSSIONS

Comparing and our clustering classification methods against to the number of baseline technique on real and synthetics data set. We refer to our clustering approach as Content and Auxiliary attributes based Text clustering (COATES). As the baselines, we used two different methods: (1) an efficient projection based clustering approach which adapts the k means approached to text. This approach is widely known as to provide excellent clustering results in a very efficient way.

We referred to this algorithm as Schultz Silverstein in all figure legend in the experimental section.

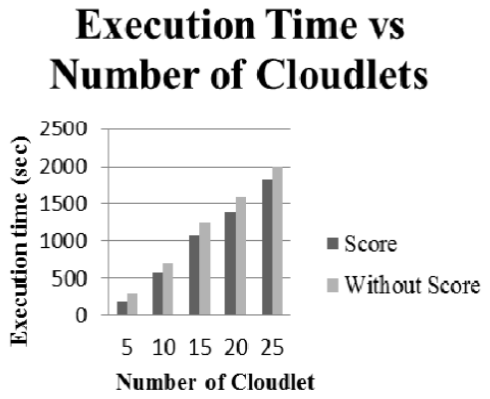

Fig.5 Execution cost vs. number of cloudlets

As future works, we would like to explore different option for the selections of the initial resource pool as it has a significant impact on the performance of the algorithms.

We would also like to an experiment with different optimizations strategies such as genetic algorithms and compare their performance with PSO.

Another future works is extending the resource model to consider the data transfer cost between data centers so that VMs can been deployed on different regions. Extending the algorithms to include heuristics that ensure a task is assigned to a VM with sufficient memory to execute it will be included in the algorithm. Finally, aim to implement our approach in a workflow engine so that it can be utilized for deploying applications in a real life environment.

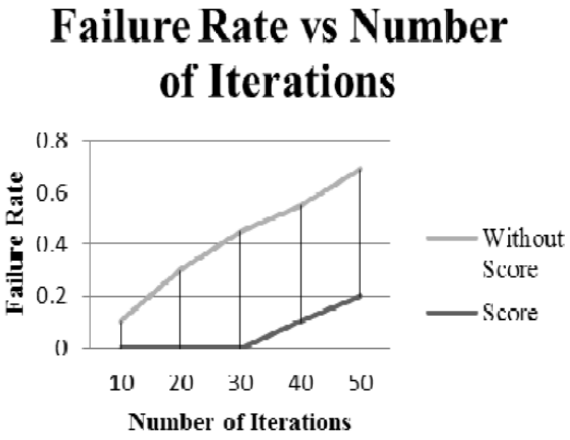

Fig.6 Failure rate vs. number of iterations

V. CONCLUSION

Our experimental results shown that continuous adaptation heuristics, which makes the user of applications dynamism can, reduces the execution costs by up to 27:5% on clouds while also meeting the QoS constraints. We have also studied the feasibility of GA based approach for optimizing executions of dynamics data flow and shows that although the GA based approach gives near optimal solutions its time complexity is proportional to the input data rates, making it unsuitable for high velocity applications. A hybrid approach

method which uses GA for initial deployment and the CE greedy heuristics for runtime adaptation may be more suitable. This is to be investigated as future works.

In addition, we planned to extend the concept of dynamic tasks which will further allow for alternate implementation sat coarser granularity such as "alternate paths", and provide end users with more sophisticated controls. Further, we planned to extend the resources mapping heuristics for an ensemble of data flows with a shared budget and address issues, such as fairness in addition to throughput constraints and application value.

REFERENCES

[1] S. Callaghan, P. Maechling, P. Small, K. Milner, G. Juve, T. Jordan, E. Deelman, G. Mehta, K. Vahi, D. Gunter, K. Beattie, and C. X. Brooks, "Metrics for heterogeneous scientific workflows: A case study of an earthquake science application," International Journal of High Performance Computing Applications, vol. 25, 2011.

[2] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: The montage example," in 2008 ACM/IEEE Conference on Supercomputing (SC 08), 2008.

[3] J. V¨ockler, G. Juve, E. Deelman, M. Rynge, and G. B. Berriman, "Experiences using cloud computing for a scientific workflow application," in 2nd Workshop on Scientific Cloud Computing 2011.

[4] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A performance analysis of EC2 cloud computing services for scientific computing," in Cloud Computing, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, O. Akan et al., Eds. Springer Berlin, 2010.

[5] K. Keahey, M. Tsugawa, A. Matsunaga, and J. Fortes, "Sky computing," IEEE Internet Computing, vol. 13, no. 5, 2009.

[6] D. Durkee, "Why cloud computing will never be free," Communications of the ACM, vol. 53, no. 5, May 2010.

[7] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, Jan. 2011.

[8] L. Fernando, B. Edmundo, M. Madeira M, "HCOC: A Cost Optimization Algorithm for Workflow Scheduling in Hybrid Clouds", Journal of Internet Services and Applications, Springer,2011.

[9] P. Marshall, K. Keahey, and T. Freeman, "Elastic site: Using clouds to elastically extend site resources," in 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010), 2010.

[10] H. Kim, Y. el-Khamra, I. Rodero, S. Jha, and M. Parashar, "Autonomic management of application workflows on hybrid computing infrastructure," Scientific Programming, vol. 19, April 2011.

[11] J. Yu, R. Buyya, and C. Tham, "Cost-Based scheduling of scientific workflow application on utility grids," in First International Conference on e-Science and Grid Computing, 2005.

[12] S. Abrishami, M. Naghibzadeh, and D. Epema, "Cost-driven scheduling of grid workflows using partial critical paths," in 11th IEEE/ACM International Conference on Grid Computing, 2010.