ISSN 2229-5518

Fuzzy-GA Optimized Multi-Cloud Multi-Task Scheduler For Cloud Storage And Service Applications

Suresh Kumar V.S ,Chandra Sekar. A

ABSTRACT

Computing clusters have been one of the most popular platforms for solving Many Task Computing(MTC) problems, especially in the case of loosely coupled tasks. However, building and managing physical clusters exhibits several drawbacks:1)Major investments in hardware, specialized installations, and qualified personal; 2) Long periods of cluster under-utilization; 3)Cluster overloading and insufficient computational resources during peak demand periods. Regarding these limitations, cloud computing technology has been proposed to complement the in-house data-center infrastructure to satisfy peak workloads.In this paper, we explore this scenario to deploy a computing cluster on the top of a multi cloud infrastructure, for solving loosely coupled MTC Application.

The benefits of Quality of Service (QoS) aware service selection is undisputed. The selection process based on QoS allows the user to specify their requirements not only based on functional attributes but also on non-functional attributes. The automation of this selection process can be done via optimization. Genetic algorithm is one such method that cans find approximate solutions during the service selection task

Index terms: Multi cloud infrastructure, Many-Task Computing (MTC), Quality of Service (QoS).

1. INTRODUCTION

Cloud computing is gaining acceptance in many IT organizations, as an elastic, flexible, and variable-cost way to deploy their service platforms using outsourced resources. Unlike traditional utilities where a single provider scheme is a common practice, the ubiquitous access to cloud resources easily enables the simultaneous use of different clouds.

We explore this scenario to deploy a computing cluster on the top of a multicloud infrastructure, for solving loosely coupled Many-Task Computing (MTC) applications. In this way, the cluster nodes can be provisioned with resources from different clouds to improve the cost effectiveness of the deployment, or to implement high-availability strategies.

*V.S.suresh Kumar,*PG Student Department of computer Science and Engineering, Nandha College of Technology, Erode-638052. suresh_acp@sify.com,8220455007

A.Chandrasekar, Associate professor, Department of computer Science and Engineering, Nandha College of Technology,Erode638052,chandru.as76@gmail.com

We prove the viability of this kind of solutions by evaluating the scalability, performance, and cost of different configurations of a Sun Grid Engine cluster, deployed on a multicloud infrastructure spanning a local data center and three different cloud sites: Amazon EC2 Europe, Amazon EC2 US, and Elastic Hosts. Although the testbed deployed in this work is limited to a reduced number of computing resources (due to hardware and budget limitations), we have complemented our analysis with a simulated infrastructure model, which includes a larger number of resources, and runs larger problem sizes. Data obtained by simulation show that performance and cost results can be extrapolated to large-scale problems and cluster infrastructures
This cluster can be scaled-out by deploying new virtual worker nodes on remote clouds. MTC paradigm embraces different types of high-performance applications involving many different tasks, and requiring large number of computational resources over short periods of time. These tasks can be of very different nature, with sizes from small to large, loosely coupled or tightly coupled, compute-intensive or data-intensive. Cloud computing is gaining acceptance in many IT organizations, as an elastic, flexible, and variable-cost way to deploy their service platforms using outsourced resources. Unlike traditional utilities where a single provider scheme is a common practice, the

ubiquitous access to cloud resources easily enables the simultaneous use of different clouds.

2. LITERATURE REVIEW

In WS GRAM (Grid Resource Allocation and Management) when a job is submitted, the request is sent to the Managed Job Factory service of the remote computer. The Managed Job Factory and Managed Job are two services running on every node of a Globus Grid. The Managed Job Factory handles each request and creates a Managed Job resource for each job. Authentication is performed via Web Service mechanisms and some operations are mapped to a local user via the sudo UNIX command. The Managed Job service uses a Job Manager to start and control the job according to its RSL specification, mapping the request to a local user and communicating state changes back to the GRAM client via WSNotifications .When the job terminates, either normally or by failing, the Managed Job resource is destroyed, ending the life cycle of the Grid job [1]. A key component in private/hybrid clouds will be virtual infrastructure (VI) mangement, the dynamic orchestration of VMs that meets the requirements we've just outlined. Here, we discuss VI management's relevance not just for creating private/hybrid clouds but also within the emerging cloud ecosystem [2]

ViNe (Virtual Netwrk) must use a space that is not routed in the Internet namely Unused/reserved class A IPv4 spaces, Private address spaces, Some range of IPv6 space. Hosts are added by assigning an address in ViNe space, Unused/additional NIC. IP aliasing is the one where Software installation is NOT required in hosts and Sub-space allocated for each institution [3]. The Cloud Container product set makes it possible to extend your data center into one or more IaaS provider's clouds while retaining complete securely & control. Security & control accomplished by literally extending your enterprise firewall to enclose, isolate, and control each of the servers running in the public cloud, and all of the data in motion as well as when at rest. Fully integrated into existing ALM, monitoring, cost accounting processes, and operating procedures the a Cloud Container based Disaster Readiness solution streamlines migration and simplifies operation of stand-by clone Data Center(s). Miniaturized clones ready to instantly scale to full power on-demand, but only if and when required [4].

Clouds give workflow developers several advantages over traditional HPC systems, such as root access to the operating system and control over the entire software environment, reproducibility of results through the use of VM images to store computational environments, and on-demand provisioning capabilities [5]. Montage is a data-intensive application. The input images, the intermediate files produced during the execution of the workflow and the output mosaic are of considerable size and require significant storage resources. The tasks on the other hand have a small runtime of at most a few minutes [6].

Understanding this tradeoff through modeling lets IT organizations justify purchase decisions quantitatively. In addition, funding agencies can objectively evaluate

alternative equipment proposals, and policymakers can fashion guidelines that ensure the most efficient outcome for all. Most importantly, if consumers understand the real cost of a CPU hour, an efficient market will eventually reflect this real cost, enabling a fair, competitive market for the benefit of all consumers [7]. Grids constitute a promising platform to execute loosely coupled, high-throughput parameter sweep applications, which arise naturally in many scientific and engineering fields like bio-informatics, computational fluid dynamics, and particle physics, etc. In spite of the simple computational structure of these applications, its efficient execution and scheduling are challenging because of the dynamic and heterogeneous nature of Grids [8].

Support for heterogeneous configurations, usually applications from different domains needs different software environments (e.g. libraries, compilers or scientific databases). The installation and maintenance of these different environments, which sometimes are conflicting, increases the operational cost of the cluster [9]. The isoefficiency function is one of many parallel performance metrics that measure the scalability. It relates problem size to the number of processors required in maintaining a system's efficiency, and it lets us determine scalability with respect to the number of processors, their speed, and the communication bandwidth of the interconnection network. The isoefficiency function also succinctly captures the characteristics of a particular algorithm/architecture combination in a single expression, letting us compare various combinations for a range of problem sizes and numbers of processors [10].

3. METHODOLOGY

Present an Optimized Multi-cloud Multi-task Scheduler for cloud storage and services infrastructure. Cluster the service being rendered by cloud systems. Optimization criteria are arrived for multi-tasking in the cloud storage space. Cluster objects are formed with the optimal threshold arrived on the multi-tasks based on user demand. Implement scheduling policies of FIFO by master node to distribute waiting tasks to computing nodes. Achieve optimal task scheduling threshold for the demanded

multi-task cluster. Fuzzy GA optimization is applied to make scheduling decision by evaluating entire group of task in the job queue. The phases of Multi-cloud Multi-task Scheduler are:

- Multi-cloud Infrastructure
- Virtual Cluster
- Many Task Computing

- Fuzzy GA Cluster
- Optimal Multi-task Scheduler

3.1. MULTI-CLOUD INFRA STRUCTURE

Multi-Cloud infrastructure builds with data-centers running Many Task Computing applications. Enable to increase or decrease computing resources according to task demand. Allow to get exact amount of resources require at any specific time. Reduce or even eliminate obsolete or unused cloud resources. This gives access to potentially unlimited resources. Cloud providers deploy required number of servers' instances simultaneously. User deploys cloud instances with different hardware configurations, operating systems, and software package.

3.2 VIRTUAL CLUSTER

Virtual clusters platforms used for solving MTC loosely coupled tasks high-throughput computing applications. Deploy elastic computing clusters satisfy peak workloads. Problem sizes in terms of mesh size, iterations, and number of jobs are classified as set of class attributes. Chose a problem size of specific class appropriate for middle-class resources used as cluster worker nodes and measured of computing time. Submitted higher number of jobs depending on the cluster configuration to saturate the cluster and obtain realistic throughput measures. On executing loosely coupled high throughput computing applications cluster performance depends on. Number of jobs completed in maximum performance rate of cluster in jobs executed per second half-performance length. Multi-cloud cluster implementations do not incur in performance slowdowns compared to single-site implementations, cluster performance scales linearly local cluster is complemented with external cloud nodes. Quantify the cost of cluster configurations measured as cost of infrastructure per time unit.

3.3 MANY TASK COMPUTING

Many-Task Computing (MTC) contains different types of high-performance applications involving many different tasks require large number of computational resources over short periods of time. Tasks are different in nature with sizes from small to large loosely coupled or tightly coupled compute-intensive or data-intensive. Workload comprises execution of a set of independent jobs with no communication among them. Data transfer takes place when front-end submit a new job to a worker node read the executable binary file, read caching is activated in the client side binary file is the same for all the runs client has access to the server only once in first job submission. For all the subsequent runs client has the binary file locally cached no not need to access to the server. For number of clients over 16 file size over 100KB, server peak throughput for read operations is identified. When clients' access is simultaneously to the server identified throughput is shared among them. Server read bandwidth allocated to each client for different number of clients display read time for different file sizes.

3.4 FUZZY-GA CLUSTER

Fuzzification comprises process of transforming values into grades of membership for linguistic terms of fuzzy sets. Membership function is used to associate a grade to each linguistic term. For each input and output variable selected, define two or more membership functions (MF), define a qualitative category for each one of them. Shape of the functions work with is triangles with three parameters for triangular function. They are: execution time, work load and objective function. It has three variable input for fuzzification. After fuzzify all these parameter using triangular MF of fuzzy logic to get the optimized task scheduling by apply GA to it. Genome representation is for job scheduling. The feasible solutions are encoded in vector (schedule) of size n tasks and numbers indicates the slot where task i is assigned by the schedule. Values of the vector are natural numbers included in the range (1, m slots).

Genetic Operators has the following process. In initialization process the slot number (node number) which has data for the task is chosen as the initial schedule. If there are several choices for one task slot number is randomly selected among these slots lead us faster to an optimal schedule. Selection operator used is based on roulette wheel method. Probabilities are calculated by dividing fitness values of individuals by sum of the fitness values. Crossover operator is the simple one-point crossover. Mutation operator used is Flip Mutator.

3.5 OPTIMAL MULTI-TASK SCHEDULER

Multitask scheduling scheme comprises of processor units in multi-cloud, collecting all tasks and take charge of dispatching them to other process units. Each process unit has its own dispatch queue (DQ). Master unit communicates with other process units through these dispatch queues. Ensure that processor units always find some tasks in the dispatch queue. Process units works in parallel with other units scheduling the newly arrived tasks, periodically updating dispatch queues. Tasks are sorted ascending by the value of deadlines. GA as optimization algorithm is the simplicity of operation and power of effect.

4. RESULTS AND DISCUSSIONS OF FUZZY-GA OPTIMIZED MULTI-CLOUD MULTI-TACK SCHEDULER

In this section we evaluate performance of Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme for Wireless Ad hoc Network through NS2 simulation. Proposed Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme has been compared to Existing computing cluster on top of a multi-cloud infrastructure that is very latest and the most similar to our proposed model.
In order to construct performance evaluations and comparisons, we have simulated Proposed Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme and existing computing cluster on top of a multi-cloud infrastructure using the NS2 simulator. In order to compare these methods in the similar conditions, we have considered the identical simulation scenarios as described. The simulation is conducted with a NS2 simulator which simulates several broadcast algorithms on random ad hoc networks. To generate a random ad hoc network, n hosts are randomly placed in a restricted 1000 m × 1000 m area. The performance of Proposed Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme has been compared to existing computing cluster on top of a multi-cloud infrastructure is evaluated by the following metrics.

- Cluster size
- Number of cloud sources
- Multi-task density

4.1 Cluster size

Cluster size is the number of virtual multi-cloud services. Results show that the proposed Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme is higher than that of the Existing computing cluster on top of a multi-cloud infrastructure.

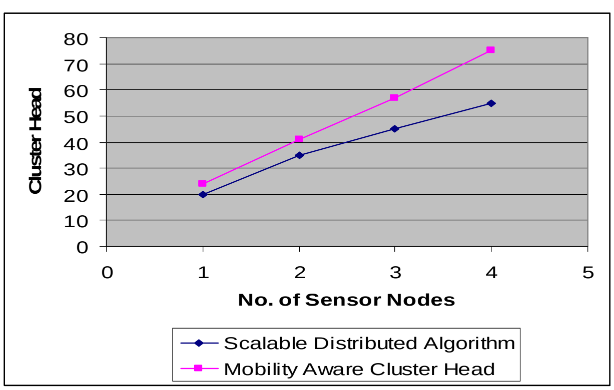

Fig.1 cluster size

We notice that for the uppermost curve in Performance of Proposed Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme in terms of cluster size contribution attains 2-6 % higher than the existing computing cluster on top of a multi-cloud infrastructure.

4.2 Number of cloud sources

Cloud sources are the one which involves in the cloud services. Fig. 2 shows the number of cloud sources in cloud services. Results show that the proposed Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme cloud service participation is higher than that of the Existing computing cluster on top of a multi-cloud infrastructure. We notice that for the uppermost curve in Performance of Proposed Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme in terms of cloud service contribution attains 30-60% higher than Existing computing cluster on top of a multi-cloud infrastructure.

4.3. Multi-task density

Multi-task density means number of multi-task executed simultaneously. Results show that the proposed Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme multi-task participation is higher than that of the Existing computing cluster on top of a multi-cloud-infrastructure.

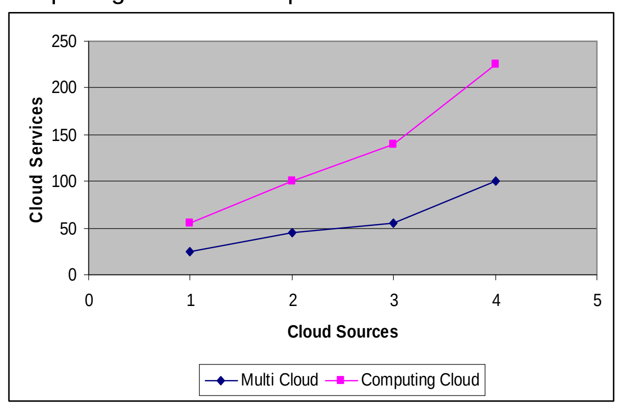

Fig.2 cloud sources

We notice that for the uppermost curve in Performance of Proposed Fuzzy-GA Optimized Multi-Task Multi-Cloud scheme in terms of multi-tasking contribution attains 30% to 80% higher than Existing computing cluster on top of a multi-cloud infrastructure.

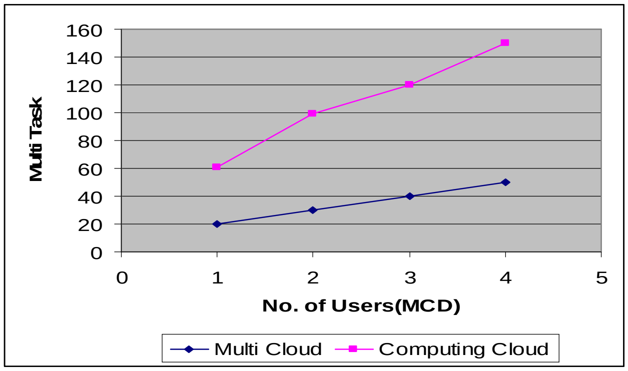

Fig. 3 multi-task density

5. CONCLUSION

In this paper we have analyzed the challenges and viability of deploying a computing cluster on top of a multi-cloud infrastructure spanning four different sites for solving loosely coupled MTC applications. We have implemented a real test bed cluster (based on a SGE queuing system) that comprises computing resources from our in-house infrastructure, and external resources from three different clouds: Amazon EC2 (Europe and USA zones) and Elastic Hosts. This fact proves that the multi-cloud implementation of a computing cluster is viable from the point of view of scalability, and does not introduce important overheads, which could cause significant performance degradation. On the other hand, the cost analysis shows that, for the workload considered, some hybrid configurations (including local and cloud nodes) exhibit better performance-cost ratio than the local setup, so proving that the multi-cloud solution is also appealing from a cost perspective. Fuzzy-GA optimization is carried out to evaluate task optimal value on user demand. Cluster evaluation efficacy is improved with Fuzzy-GA optimality. Simulations are conducted to evaluate the performance of proposal in terms of Cluster size, Number of cloud sources, Infrastructure services, and Multi-task density

REFERENCES

[1] E. Huedo, R. Montero, I. Llorente, A Modular Meta-Scheduling Architecture for Interfacing with Pre-WS and WS Grid Resource Management Services, Future Generation Computer Systems 23 (2): 252–261, 2007.

[2] B. Sotomayor, R.S. Montero, I.M. Llorente, I. Foster, Virtual Infrastructure Management in Private and Hybrid Clouds, IEEE Internet Computing 13(5): 14–22, 2009.

[3] M. Tsugawa, J. Fortes, A Virtual Network (ViNe) Architecture for Grid Computing, in: In Proc. IEEE Intl. Parallel and Distributed Processing Symposium (IPDPS06), 2006, pp. 1–10.

[4]VPN-Cubed, http://www.cohesiveft.com/vpncubed/

[5] G. Juve, E. Deelman, K. Vahi, B. P. Berman, B. Berriman, and P. Maechling, Scientific Workflow Applications on Amazon EC2. Workshop on Cloud-based Services and Applications in conjunction with 5th IEEE Int. Conference on e-Science, 2009.

[6] E. Deelman , G. Singh , M. Livny, B. Berriman, and J. Good, The cost of doing science on the cloud: the Montage example. Proceedings of the 2008 ACM/IEEE conference on Supercomputing, 2008.

[7] E. Walker, The Real Cost of a CPU Hour. Computer 42(4): 35–41, 2009.

[8] R. Montero, E. Huedo, I. Llorente, Benchmarking of High Throughput Computing Applications on Grids, Parallel Computing 32 (4): 267–279, 2006.

[9] R.S. Montero, R. Moreno-Vozmediano. I.M. Llorente, An Elasticity Model for High Throughput Computing Clusters. Journal of Parallel and Distributed Computing (in press, DOI: 10.1016/j.jpdc.2010.05.005), 2010.

[10] A.Y. Grama, A. Gupta, and V. Kumar. Isoefficiency: Measuring the Scalability of Parallel Algorithms and Architectures. IEEE Parallel and Distributed Technology: Systems and Applications, 1(3): 12–21, 1993.