IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661 Volume 4, Issue 3 (Sep-Oct. 2012), PP 05-12
www.iosrjournals.org

Multicloud Deployment of Computing Clusters for Loosely Coupled Multi Task Computing (MTC) Applications

[1]A. Karthik, [2]P. Abdul Wahid, [3]D. Nagarani
[1, 2, 3](M Tech(Software Engineering) Dept.of Computer Science & Technology,(Student) Sreenidhi Institute of Science & Technology/An Autonomous Institution,Hyderabad,AP, India)

Abstract: *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. In this project, we explore this scenario to deploy a computing cluster on the top of a multicloud infrastructure, for solving loosely coupled Multi-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. 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 test bed 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.*

Index Terms*—Cloud computing, computing cluster, multicloud infrastructure, loosely coupled applications.*

I. Introduction

MULTI-TASK COMPUTING (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 technologies can offer important benefits for IT organizations and data centers running MTC applications: elasticity and rapid provisioning, enabling the organization to increase or decrease its infrastructure capacity within minutes, according to the computing necessities; pay as-you-go model, allowing organizations to purchase and pay for the exact amount of infrastructure they require at any specific time; reduced capital costs, since organizations can reduce or even eliminate their in-house infrastructures, resulting on a reduction in capital investment and personnel costs; access to potentially “unlimited” resources, as most cloud providers allow to deploy hundreds or even thousands of server instances simultaneously; and flexibility, because the user can deploy cloud instances with different hardware configurations, operating systems, and software packages.

Computing Clusters have been one of the most popular platforms for solving MTC problems, especially in the case of loosely coupled tasks (e.g., high-throughput computing applications). However, building and managing physical clusters exhibits several drawbacks:
1) major investments in hardware, specialized installations (cooling, power, etc.), and qualified personal;
2) long periods of cluster underutilization; and
3) cluster overloading and insufficient computational resources during peak demand periods.

Regarding these limitations, cloud computing technology has been proposed as a viable solution to deploy elastic computing clusters, or to complement the in-house data center infrastructure to satisfy peak workloads. For example, the BioTeam has deployed the Univa UD UniCluster Express in an hybrid setup, which combines local physical nodes with virtual nodes deployed in the Amazon EC2.we extend this hybrid solution by including virtualization in the local site, so providing a flexible and agile management of the whole infrastructure, that may include resources from remote providers. However, all these cluster proposals are deployed using a single cloud, while multicloud cluster deployments are yet to be studied. The simultaneous use of different cloud providers to deploy a computing cluster spanning different clouds can provide several benefits:

- **High availability and fault tolerance:** the cluster worker nodes can be spread on different cloud sites, so in the case of cloud downtime or failure, the cluster operation will not be disrupted. Furthermore, in this situation, we can dynamically deploy new cluster nodes in a different cloud to avoid the degradation of the cluster performance.

- **Infrastructure cost reduction:** since different cloud providers can follow different pricing strategies, and even variable pricing models (based on the level of demand of a particular resource type, daytime versus nighttime, weekdays versus weekends, spot prices, and so forth), the different cluster nodes can change dynamically their locations, from one cloud provider to another one, in order to reduce the overall infrastructure cost.

The main goal of this work is to analyze the viability, from the point of view of scalability, performance, and cost of deploying large virtual cluster infrastructures distributed over different cloud providers for solving loosely coupled MTC applications. This work is conducted in a real experimental test bed that comprises resources from our in-house infrastructure, and external resources from three different cloud sites: Amazon EC2 (Europe and US zones) and Elastic Hosts. On top of this distributed cloud infrastructure, we have implemented a Sun Grid Engine(SGE) cluster, consisting of a front end and a variable number of worker nodes, which can be deployed on different sites (either locally or in different remote clouds).We analyze the performance of different cluster configurations, using the cluster throughput (i.e., completed jobs per second) as performance metric, proving that multicloud cluster implementations do not incur in performance slowdowns, compared to single-site implementations, and showing that the cluster performance (i.e., throughput) scales linearly when the local cluster infrastructure is complemented with external cloud nodes. In addition, we quantify the cost of these cluster configurations, measured as the cost of the infrastructure per time unit, and we also analyze the performance/cost ratio, showing that some cloud-based configurations exhibit similar performance/ cost ratio than local clusters.Due to hardware limitations of our local infrastructure, and the high cost of renting many cloud resources for long periods, the tested cluster configurations are limited to a reduced number of computing resources (up to 16 worker nodes), running a reduced number of tasks (up to 128 tasks). However, as typical MTC applications can involve much more tasks, we have implemented a simulated infrastructure model, that includes a larger number of computing resources (up to 256 worker nodes), and runs a larger number of tasks (up to 5,000). The simulation of different cluster configurations shows that the performance and cost results can be extrapolated to large-scale problems and cluster infrastructures.More specifically, the contributions of this work are the following:

1. Deployment of a multicloud virtual infrastructure spanning four different sites: our local data center, Amazon EC2 Europe, Amazon EC2 US, and Elastic Hosts; and implementation of a real computing cluster test bed on top of this multicloud infrastructure.
2. Performance analysis of the cluster test bed for solving loosely coupled MTC applications (in particular, an embarrassingly parallel problem), proving the scalability of the multicloud solution for this kind of workloads.
3. Cost and cost-performance ratio analysis of the experimental setup, to compare the different cluster configurations, and proving the viability of the multicloud solution also from a cost perspective.
4. Implementation of a simulated infrastructure model to test larger sized clusters and workloads, proving that results obtained in the real testbed can be extrapolated to large-scale multicloud infrastructures..

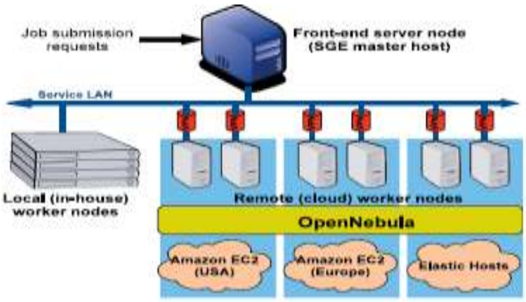
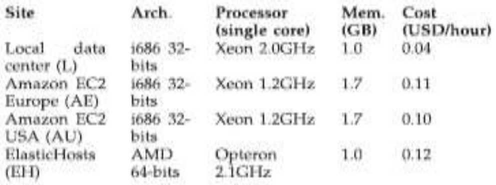

Job submission requests
Front-end server node (SGE master host)
Service LAN
Local (in-house) worker nodes
Remote (cloud) worker nodes
OpenNebula
Amazon EC2 (USA)
Amazon EC2 (Europe)
Elastic Hosts

FIG 1: EXPERIMENTAL FRAMEWORK

II. Deployment Of A Multicloud Virtual Cluster

Fig. 1 shows the distributed cluster testbed used in this work deployed of top of a multicloud infrastructure. This kind of multicloud deployment involves several challenges,related to the lack of a cloud interface standard; the distribution and management of the service master images;and the interconnection links between the service components.A brief discussion of these issues, and the main design decisions adopted in this work to face up these challenges are included in Appendix A of the supplemental material, which can be foundontheComputer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.186.Our experimental testbed starts from a virtual cluster deployed in our local data center, with a queuing system managed by SGE software, and consisting of a cluster front end (SGE master) and a fixed number of virtual worker nodes (four nodes in this setup). This cluster can be scaled out by deploying new virtual worker nodes on remote clouds. The cloud providers considered in this work are Amazon EC2 (Europe and US zones) and ElasticHosts. Table 1 shows the main characteristics of in-house nodes and cloud nodes used in our experimental testbed. Besides the hardware characteristics of different testbed nodes, Table 1 also displays the cost per time unit of these resources. In the case of cloud resources, this cost represents the hourly cost charged by the cloud provider for the use of its resources. Appendix B.1 of the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.186, gives more details about the cost model used for cloud resources. On the other hand, the cost of the local resources is an estimation based on the model proposed by Walker that takes into account the cost of the computer hardware, the cooling and power expenses, and support personnel. For more information about the application of this cost model to our local data center, see Appendix B.2, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.186.

TABLE 1
Characteristics of Different Cluster Nodes



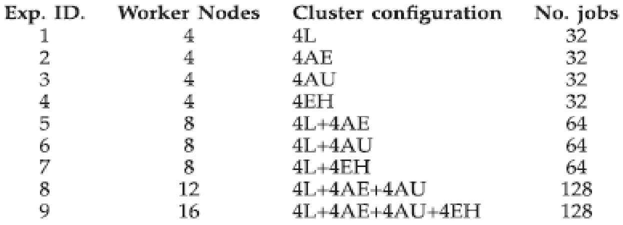
Site	Arch.	Processor (single core)	Mem. (GB)	Cost (USD/hour)
Local data center (L)	i686 32-bits	Xeon 2.0GHz	1.0	0.04
Amazon EC2 Europe (AE)	i686 32-bits	Xeon 1.2GHz	1.7	0.11
Amazon EC2 USA (AU)	i686 32-bits	Xeon 1.2GHz	1.7	0.10
ElasticHosts (EH)	AMD 64-bits	Opteron 2.1GHz	1.0	0.12

2.1PERFORMANCE ANALYSIS

In this section, we analyze and compare the performance offered by different configurations of the computing cluster, focused in the execution of loosely coupled applications. In particular, we have chosen nine different cluster configurations (with different number of worker nodes from the three cloud providers), and different number of jobs (depending on the cluster size), as shown in Table 2. In the definition of the different cluster configurations, we use the following acronyms: L: local infrastructure; AE: Amazon EC2 Europe cloud; AU: Amazon EC2 US cloud; and EH: Elastic Hosts cloud. The number preceding the site acronym represents the number of worker nodes. For example, 4L is a cluster with four worker nodes deployed in the local infrastructure; and 4L+ 4AE is a eight-node cluster, four deployed in the local infrastructure and four in Amazon EC2 Europe. To represent the execution profile of loosely coupled applications, we will use the Embarrassingly Distributed (ED) benchmark from the Numerical Aerodynamic Simulation(NAS) Grid (NGB) suite. The ED benchmark consists of multiple independent runs of a flow solver, each one with a different initialization constant for the flow field.

NGB defines several problem sizes (in terms of mesh size, iterations, and number of jobs) as classes S, W, A, B, C, D, and E. We have chosen a problem size of class B, since it is appropriate (in terms of computing time) for middle-class resources used as cluster worker nodes. However, instead of submitting 18 jobs, as ED class B defines, we have submitted a higher number of jobs (depending on the cluster configuration, see Table 1 below) in order to saturate the cluster and obtain realistic throughput measures.

Cluster Configurations



Exp. ID.	Worker Nodes	Cluster configuration	No. jobs
1	4	4L	32
2	4	4AE	32
3	4	4AU	32
4	4	4EH	32
5	8	4L+4AE	64
6	8	4L+4AU	64
7	8	4L+4EH	64
8	12	4L+4AE+4AU	128
9	16	4L+4AE+4AU+4EH	128

Table 2: Cluster Configurations

As we have proven in a previous work, when executing loosely coupled high-throughput computing applications, the cluster performance (in jobs completed per second) can be easily modeled using the following equation:

$$r(n) = \frac{r_{max}}{1 + n_{1/2}/n},$$

Equation 1: Cluster Performance

where n is the number of jobs completed, r_{max} is the asymptotic performance (maximum rate of performance of the cluster in jobs executed per second), and $n_{1/2}$ is the half-performance length.

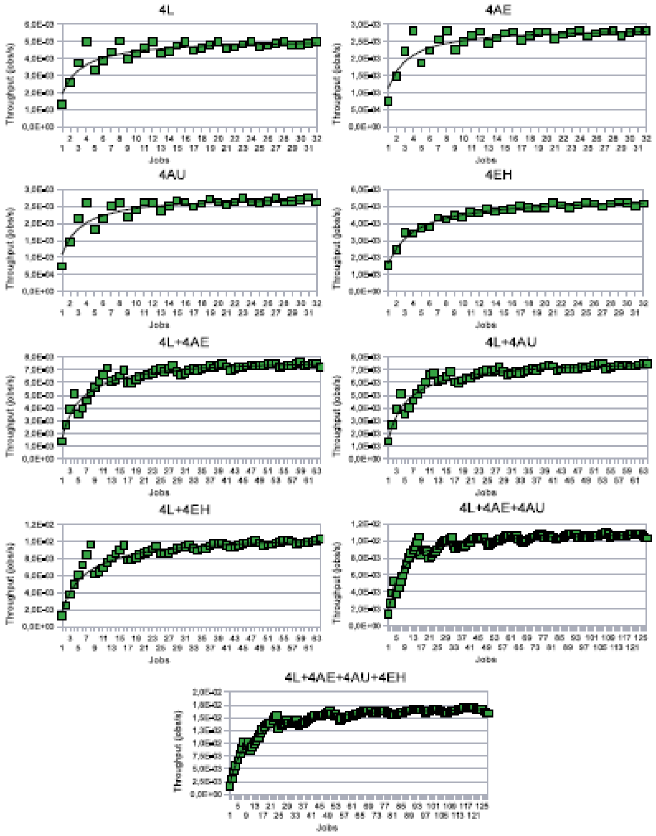
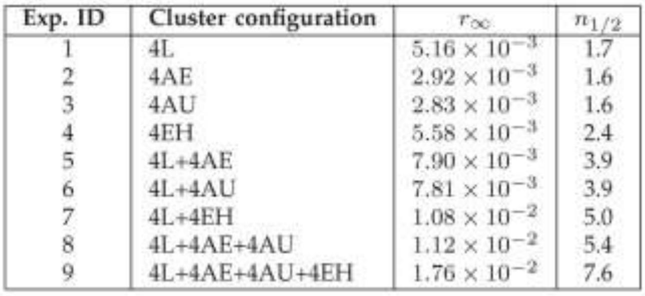


Figure2: Throughput for Different Cluster Configurations



Exp. ID	Cluster configuration	r_∞	$n_{1/2}$
1	4L	5.16×10^{-3}	1.7
2	4AE	2.92×10^{-3}	1.6
3	4AU	2.83×10^{-3}	1.6
4	4EH	5.58×10^{-3}	2.4
5	4L+4AE	7.90×10^{-3}	3.9
6	4L+4AU	7.81×10^{-3}	3.9
7	4L+4EH	1.08×10^{-2}	5.0
8	4L+4AE+4AU	1.12×10^{-2}	5.4
9	4L+4AE+4AU+4EH	1.76×10^{-2}	7.6

Table3: Performance Model Parameters

The above table shows the r_{max} and $n_{1/2}$ parameters of the performance model for each cluster configuration. The parameter r_{max} can be used as a measure of cluster throughput, in order to compare the different cluster configurations, since it is an accurate approximation for the maximum performance (in jobs per second) of the cluster in saturation. Please, note that we have achieved five runs of each experiment, so data in above table represents the mean values of r_{max} and $n_{1/2}$.

Table 3 shows the r1 and n1=2 parameters of the performance model for each cluster configuration. The parameter r1 can be used as a measure of cluster throughput, in order to compare the different cluster configurations, since it is an accurate approximation for the maximum performance (in jobs per second) of the cluster in saturation. Please, note that we have achieved five runs of each experiment, so data in Table 3 represent the mean values of r1 and n1=2 (standard deviations can be found in Appendix C, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.186).If we compare 4L and 4EH configurations, we observe that they exhibit very similar performance. This is because of two main reasons: first, worker nodes from both sites have similar CPU capacity (see Table 1); and second, communication latencies for this kind of loosely coupled applications do not cause significant performance degradation, since data transfer delays are negligible compared to execution times, mainly thanks to the NFS file data caching implemented on the NFS clients (worker nodes), which notably reduces the latency of NFS read operations. On the other hand, the lower performance of 4AE and 4AU configurations is mainly due to the lower CPU capacity of the Amazon EC2 worker nodes (see Table 1).

An important observation is that cluster performance for hybrid configurations scales linearly. For example, if we observe the performance of 4L+4AE configuration, and we compare it with the performance of 4L and 4AE configurations separately, we find that the sum of performances of these two individual configurations is almost similar to the performance of 4L + 4AE configuration. This observation is applicable to all of the hybrid configurations, as shown in Fig.3. This fact proves that, for the particular workload considered in this work, the use of a multicloud infrastructure spanning different cloud providers is totally viable from the point of view of performance and scalability, and does not introduce important overheads, which could cause significant performance degradation.

2.2 COST ANALYSIS

Besides the performance analysis, the cost of cloud resources also has an important impact on the viability of the multicloud solution. From this point of view, it is important to analyze, not only the total cost of the infrastructure, but also the ratio between performance and cost, in order to find the most optimal configurations.

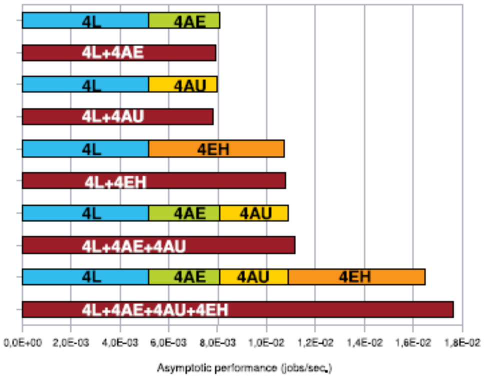
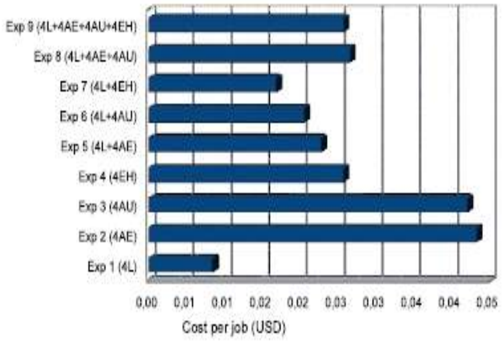


FIGURE 3: ASYMPTOTIC PERFORMANCE COMPARISION

The average cost of each instance per time unit is gathered in Table 1. Based on these costs, and using the cost model detailed in Appendix B of the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.186, we can estimate the cost of every experiment. However, this cost is not suitable to compare the different cluster configurations, since we are running different number of jobs for every configuration. So, in order to normalize the cost of different configurations, we have computed the cost per job, shown in Fig. 4, by dividing the cost of each experiment by the number of jobs in the experiment. As the cost of local resources is lower than the cloud resources, it is obvious that 4L configuration results in the experiment with the lowest cost per job. Similarly, those experiments including only cloud nodes (e.g., 4AE,4AU, and 4EH) exhibit higher price per job than those hybrid configurations including local and cloud nodes (e.g.,4L + 4AE, 4L + 4AU, and 4L + 4EH). We also observe that, for the particular workload used in this experiment,configurations including EH nodes result in a lower cost per job than those configurations including Amazon nodes(e.g., 4EH compared to 4AE and 4AU, or 4L +4EH compared to 4L + 4AE and 4L + 4AU). Regarding these comparative cost results, it is obvious that, for large organizations that make intensive use of Computational resources, the investment on a right-sized local data center can be more cost



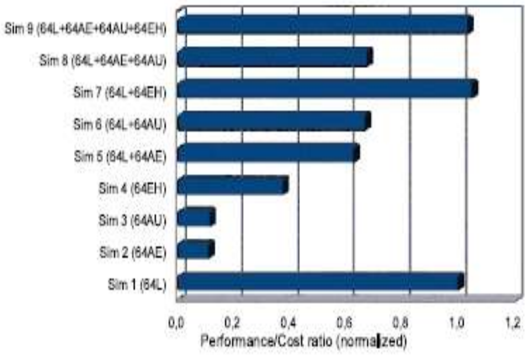


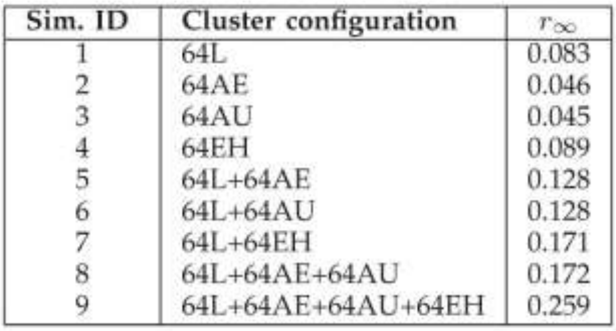
Figure 4: C**ost per job for different configurations configurations.**

Figure 5: Performance/cost ratio for different

effective than leasing resources to external cloud providers. However, for small companies or small research communities, the use of cloudbased computing infrastructures can be much more useful, from a practical point of view, than setting up their own computing infrastructure.Finally, to decide what configurations are optimal from the point of view of performance and cost, we have computed the ratio between cluster performance (r1) and cost per job, as shown in Fig. 5 (we show the normalized values with respect to the 4L case). We can observe that, in spite of the higher cost of cloud resources with respect to the local nodes, there are some hybrid configurations (4L + 4EH, and 4L +4AE + 4AU + 4EH) that exhibit better performance-cost ratio than the local setup (4L configuration).This fact proves that the proposed multicloud implementation of a computing cluster can be an interesting solution, not only from the performance point of view, but also from a cost perspective.

III. Simulation Results

The previous section shows the performance and cost analysis for different configurations of a real implementation of a multicloud cluster infrastructure running a real workload. However, due to hardware limitations in our local infrastructure, and the high cost of renting many cloud resources for long periods, the tested cluster configurations are limited to a reduced number of computing resources (up to 16 worker nodes in the cluster), running a reduced number of tasks (up to 128 tasks). However, as the goal of this paper is to show the viability of these multicloud cluster solutions for solving MTC problems, which can involve much more tasks, we have implemented a simulated infrastructure model, conducted by the real results obtained with the experimental testbed, which includes a larger number of computing resources (up to 256 worker nodes in the simulated infrastructure), and runs a larger number of tasks (up to 5,000). The details and validation of the simulator used in this section can be found in Appendix D of the supplemental material, which can be found on the Computer Society Digital Library at http:// doi.ieeecomputersociety.org/10.1109/TPDS.2010.186. In addition, Appendix E, analyzes the NFS server performance for the cluster sizes used in our simulation experiments, and proves that, for loosely coupled workloads, the NFS server is not a significant bottleneck, and it has little influence on cluster performance degradation.



Sim. ID	Cluster configuration	r_∞
1	64L	0.083
2	64AE	0.046
3	64AU	0.045
4	64EH	0.089
5	64L+64AE	0.128
6	64L+64AU	0.128
7	64L+64EH	0.171
8	64L+64AE+64AU	0.172
9	64L+64AE+64AU+64EH	0.259

TABLE 4:Simulation Results for Different Cluster Configurations

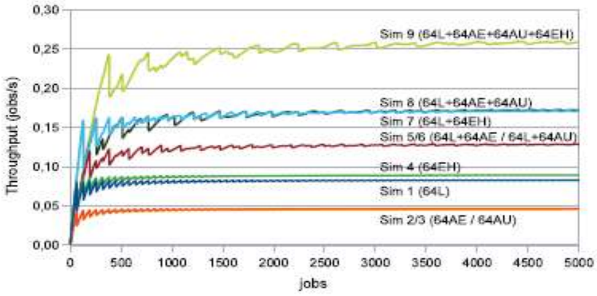


Figure 6: Throughput for simulated cluster configurations.

We have simulated different cluster configurations (see Table 4), each one running 5,000 independent tasks belonging to the NGB ED class B benchmark. The throughput results of these simulations are displayed in Fig. 6. Table 4 summarizes the asymptotic performance (r1,expressed in jobs/s) obtained for each configuration.

Looking at throughput results, we can conclude that the experimental results obtained for the real testbed can be extrapolated to larger sized clusters and higher number of tasks. In fact, we observe in Table 4 that the asymptotic performance for the simulated model scales linearly when we add cloud nodes to the local cluster configuration. For example, we find that the asymptotic performance of 64L+64AE configuration is almost similar to sum of the performances of 64L and 64AE configurations separately. Similar observations can be made for the rest of cluster configurations. Regarding the cost analysis for simulated infrastructures, summarized in Fig. 7, we observe that cost per job results for the real testbed can also be extrapolated to larger clusters, namely, hybrid configurations including local and cloud nodes (e.g., 64L + 64AE, 64L + 64AU, and 64L+ 64EH) exhibit lower cost per job than those configurations with only cloud nodes (e.g., 64AE, 64AU, and 64EH); and configurations including EH nodes result in a lower cost per job than configurations including Amazon nodes (e.g., 64EH compared to 64AE and 64AU, or 64L + 64EH compared to 64L + 64AE and 64L+ 64AU). Finally, analyzing the performance-cost ratio of the simulated infrastructures in Fig. 8, we see again a similar behavior to that observed for the real testbed, with some hybrid configurations (64L + 64EH,and 64L + 64AE+ 64AU+ 64EH) exhibiting better performance-cost ratio than the local setup (64L).

IV. Related Work

Efficient management of large-scale cluster infrastructures has been explored for years, and different techniques for on-demand provisioning, dynamic partitioning, or cluster virtualization have been proposed. Traditional methods forthe on-demand provision of computational services consist in overlaying a custom software stack on top of an existing middleware layer. For example, the MyCluster Project creates a Condor or a SGE cluster on top of TeraGrid services. The Falkon system provides a light highthroughput execution environment on top of the Globus GRAM service. Finally, the GridWay metascheduler has been used to deploy BOINC networks on top of the EGEE middleware. The dynamic partitioning of the capacity of a computational cluster has also been addressed by several projects. For example, the Cluster On Demand software enables rapid, automated, on-the-fly partitioning of a physical cluster into multiple independent virtual clusters. Similarly, the VIOcluster project enables to dynamically adjust the capacity of a computing cluster by sharing resources between peer domains.

Several studies have explored the use of virtual machines to provide custom cluster environments. In this case, the clusters are usually completely build up of virtualized resources, as in the Globus Nimbus project ,or the Virtual Organization Clusters (VOC) proposed in . Some recent works, have explored the use of cloud resources to deploy hybrid computing clusters, so the cluster combines physical, virtualized, and cloud resources. There are many other different experiences on deploying different kind of ultitier services on cloud infrastructures, such as web servers, database appliances, or web service platforms, among others. However, all these deployments only consider a single cloud, and they do not take advantage of the potential benefits of multicloud deployments. Regarding the use of multiple clouds, Keahey et al.introduce in the concept of “Sky Computing,” which enables the dynamic provisioning of distributed domains over several clouds, and discusses the current shortcomings of this approach, such as image compatibility among providers, need of standards at API level, need of trusted networking environments, etc. This work also compares the performance of two virtual cluster deployed in two settings:a single-site deployment and a three-site deployment, and concludes that the performance of a single-site cluster can be sustained using a cluster across three sites. However, this work lacks a cost analysis and the performance analysis is limited to small size infrastructures (up to 15 computer instances, equivalent to 30 processors).

V. Conclusions And Future Work

In this paper, we have analyzed the challenges and viability of deploying a computing cluster on top of a multicloud infrastructure spanning four different sites for solving loosely coupled MTC applications. We have implemented a real testbed 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 US zones) and ElasticHosts.

Performance results prove that, for the MTC workload under consideration (loosely coupled parameter sweep applications), cluster throughput scales linearly when the cluster includes a growing number of nodes from cloud providers. This fact proves that the multicloud 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 multicloud solution is also appealing from a cost perspective.

In addition, we have also implemented a model for simulating larger cluster infrastructures. The simulation of different cluster configurations shows that performance and cost results can be extrapolated to large-scale problems and clusters. It is important to point out that, although the results obtained are very promising, they can differ greatly for other MTC applications with a different data pattern, synchronization requirement, or computational profile.

The different cluster configurations considered in this work have been selected manually, without considering any scheduling policy or optimization criteria, with the main goal of analyzing the viability of the multicloud solution from the points of view of performance and cost. Although a detailed analysis and comparison of different scheduling strategies is out of the scope of this paper, and it is planned for further research, for the sake of completeness, Appendix F of the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.186, presents some preliminary results on dynamic resource provisioning, in order to highlight the main benefits of multicloud deployments, such asinfrastructure cost reduction, high availability and fault tolerance capabilities.

References

[1] I. Raicu, I. Foster, and Y. Zhao, “Many-Task Computing for Grids and Supercomputers,” Proc. Workshop Many-Task Computing on Grids and Supercomputers, pp. 1-11, 2008.

[2] BioTeam “How to Unicluster and Amazon EC2,” technical report, BioTeam Lab Summary, 2008.

[3] I. Llorente, R. Moreno-Vozmediano, and R. Montero, “Cloud Computing for On-Demand Grid Resource Provisioning,” Advances in Parallel Computing, vol. 18, pp. 177-191, IOS Press, 2009.

[4] Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2,2010.

[5] Elastic Hosts, http://www.elastichosts.com/, 2010.

[6] E. Walker, “The Real Cost of a CPU Hour,” Computer, vol. 42, no. 4, pp. 35-41, Apr. 2009.

[7] M.A. Frumkin and R.F. Van der Wijngaart, “NAS Grid Benchmarks: A Tool for Grid Space Exploration,” J. Cluster Computing, vol. 5, no. 3, pp. 247-255, 2002.

[8] E. Walker, J. Gardner, V. Litvin, and E. Turner, “Creating Personal Adaptive Clusters for Managing Scientific Jobs in a Distributed Computing Environment,” Proc. IEEE Second Int’l Workshop Challenges of Large Applications in Distributed Environments (CLADE ’06).

[9] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, and M. Wilde, “Falkon:A Fast and Light-Weight TasK ExecutiON Farmework,” Proc.IEEE/ACM Conf. SuperComputing, 2007.

[10] E. Huedo, R.S. Montero, and I.M. Llorente, “The GridWay Framework for Adaptive Scheduling and Execution on Grids,” Scalable Computing—Practice and Experience, vol. 6, pp. 1-8, 2006.

[11] J. Chase, D. Irwin, L. Grit, J. Moore, and S. Sprenkle, “Dynamic Virtual Clusters in a Grid Site Manager,” Proc. 12th IEEE Symp.High Performance Distributed Computing, 2003.

[12] P. Ruth, P. McGachey, and D. Xu, “VioCluster: Virtualization for Dynamic Computational Domains,” Proc. IEEE Int’l Conf. Cluster Computing, 2005.