BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **12**, No 2

Sofia • 2012

Secured Multi-Cloud Virtual Infrastructure with Improved Performance

R. Thandeeswaran[1], *S. Subhashini*[1], *N. Jeyanthi*[1], *M. A. Saleem Durai*[2]

[1] *School of Information Technology and Engineering, VIT University, Vellore – 632014, Tamilnadu, India*
[2] *School of Computing Sciences and Engineering, VIT University, Vellore – 632014, Tamilnadu, India*
Emails: rthandeeswaran@vit.ac.in suba.shni@yahoo.com

Abstract: *Cloud computing is a model where software applications and computing resources are accessed over Internet with minimal cost and effort by interacting with the service provider. Along with these benefits there are also some significant security concerns that need to be addressed for handling sensitive data and critical applications.*

The simultaneous use of multiple clouds can provide several potential benefits, such as high availability, fault tolerance and reduced infrastructural cost. The model proposed which is the implementation of a secured multi-cloud virtual infrastructure consists of a grid engine on top of the multi-cloud infrastructure to distribute the task among the worker nodes that are supplied with various resources from different clouds to enhance cost efficiency of the infrastructure set up and also to implement high availability feature. The Oracle grid engine is used to schedule the jobs to the worker nodes (in-house and cloud). Worker nodes will be acting like listeners to receive the job from the oracle grid engine master node. High security is provided at this point for data using AES algorithm and also a password protection key for privileged user's access. Performance analysis, cost analysis and cost-performance ratio analysis are done by comparing different cluster configurations.

Keywords: *Grid gain engine, map reduce, worker nodes, passport protection key, Monte-Carlo simulation.*

1. Introduction

In recent years IT professionals try to come up with cloud technology that will not only help their business to reduce expenses, but also provides support to compete and get ahead in the market [1]. Organizations that are experimenting with cloud computing carefully choose projects that benefit from the cloud's resources and take a crawl, walk, and run approach to build an efficient cloud implementation.

The term cloud computing can be defined as "a system that is concerned with the integration, virtualization, standardization, and management of services and resources". The benefits of cloud computing include minimized capital expenditure, utilization and efficiency improvement, high computing power, location and device independence and finally very high scalability. The various concerns about storage, data privacy, interoperation, integrity, availability, risk management and trust management.

Multi-cloud infrastructure [2] can be exposed to the public as utility computing. The use of multiple clouds enables the following advantages:

- import and export data from various clouds;
- enables "choice" ability to move clouds easily based on price and Service Level Agreement;
- stops vendor lock-in;
- automated synchronization of different clouds;
- fault tolerance with primary back and high availability of data;
- infrastructure cost reduction.

2. Issues in the existing methodology

The main challenges in multi-cloud deployments, regarding cloud interfacing, image management, network management and fault tolerance [4].

- *Cloud Interfacing*: Cloud interoperability is probably one of the aspects that are receiving more attention by community. The need for interoperable clouds is two folded: first, the ability to easily move a virtualized infrastructure among different providers would prevent a vendor lock-in and secondly, the simultaneous use of multiple clouds that are geographically distributed can also improve the cost-effectiveness, high availability or efficiency of the virtualized service.
- *Network Management*: The resources running on various cloud providers are located in different networks and may use different addressing schemes (public, private, NAT). But some services need all their components to have a uniform IP address so it is necessary to build an overlay network above a physical network for the communication purpose with different service components.
- *Image Management*: One or more virtual machines support one or more components in a virtualized service. Cloning of the master images for a component which consists of a basic OS installation is done to obtain the instance. High maintenance cost is required for distribution to each cloud.

- *Fault Tolerance and HA*: If there is a failure in one node, then the entire process fails. The number of nodes cannot be changed dynamically. Also the cloud cannot be accessed easily and some time is needed due to the network traffic and other issues.
- *Absence of Map Reduce*: A special feature of the grid gain engine which is concerned for dynamic scheduling of the jobs among the available nodes.
- *Security*: There is no secured interaction between the local grid and the cloud.

3. Proposed approach: Enhanced multi-cloud virtual infrastructure with security

The Client after proper authentication procedure submits the job to the grid gain engine as displayed in Fig. 1. The access control is a key concern as attacks by hackers are of great risk. A potential hacker can be someone with approved access to the cloud. The job specified in this implementation is Monte-Carlo simulation for calculating the credit risk.

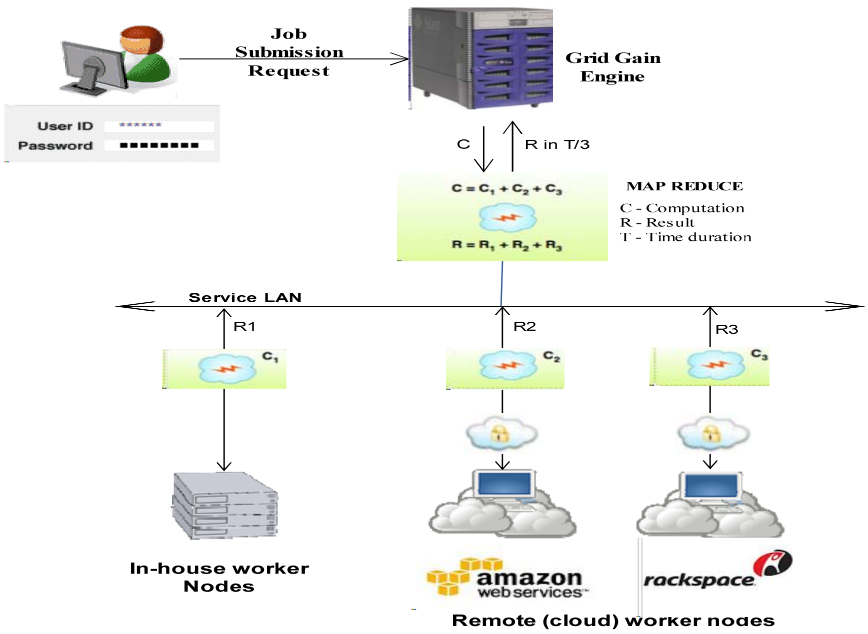

Fig. 1. Framework of the multi-cloud implementation

Monte-Carlo simulation performs risk analysis by generating models of the possible results, substituting a range of values using probability distribution. Based on the range of estimations, a random value is selected for each task. The calculation is done on this random value. The result is recorded and the process is repeated. The calculations are performed hundreds or thousands of times with each time using different randomly selected values. On completion the simulation yields a large pool of results that are used to describe the probability of reaching various results in the model.

Oracle Grid Engine is used to schedule [7] and dispatch jobs to the cloud with the help of its special feature, known as Map Reduce. When the client submits a job request, it is handled by the master node in the Grid Engine. In case of failure of the master node, one shadow master will take charge. A monitor is always analyzing the performance of the Master host and Execution host.

The Worker nodes act as listeners for receiving jobs submitted by clients through OGE, where in the jobs are scheduled in the order of priority and job types. The simultaneous tasks are executed in different nodes which are deployed in-house or remotely, as it is in Fig. 2. A different cluster configuration is maintained. The cloud providers or the external sites in this work are Amazon EC2 and Rackspace. The client must need an account with these cloud providers and the users have to pay on need basis for using the cloud infrastructure. Advanced Encryption Standard (AES) algorithm [8] is used for secured data transfer [9, 10] while accessing the cloud.

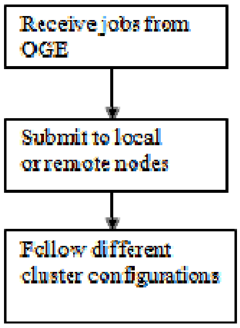

Fig. 2. Worker Node implementation

From the worker node, the user will be programmatically accessing the Rackspace (through RMI) and Amazon (through HTTP request) cloud computing infrastructure, in which the cloud exposes API.

The steps following for accessing the cloud are Launch and control cloud servers, programmatically using a RESTful API, Assign server instances custom metadata using own key/value pairs, Reboot servers from any image specified, Create custom images and schedule backups of the cloud servers.

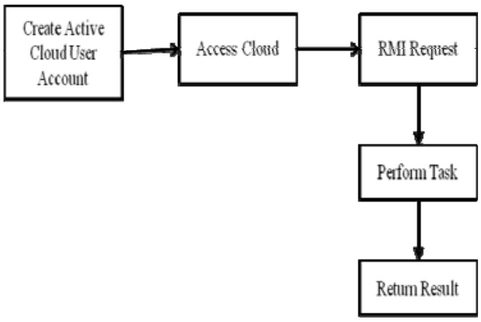

Fig. 3. API Access to a Rack space Cloud

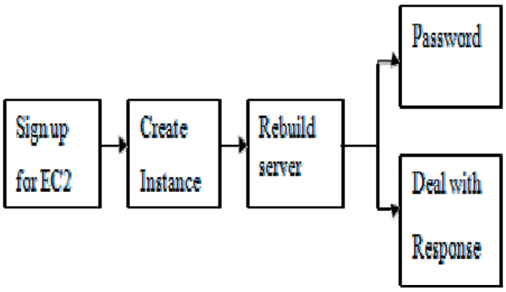

Fig. 4. API Access to Amazon Cloud

4. Experimental setup and analysis

Amazon, Google, IBM, Flexscale, Icloud, Rackspace, VMware, Sun, Microsoft, etc., are some of the companies that provide cloud computing services. Amazon, provided by Amazon Web Services [13] and Rackspace [14] are used in this implementation. Both are public clouds. Table 1 shows the main characteristics of in-house node and cloud nodes used in our experiment which is based on the computer society digital library [15].

Table 1. Characteristics of various nodes

Site	Architecture	Processor	Memory (GB)	Cost USD/Hr
In-house	i686 32 bit	Pentium dual core	4.0	0.01
Rackspace	AMD 64 bit	Opteron 2.1 Ghz	0.512	0.03
Amazon	i686 64 bit	Xeon 1.7 Ghz	0.752	0.04

Multiple grid engines are available worldwide at heavy cost and most are apt for a particular platform. The features that each engine provides vary accordingly. But the world is moving to open the source and people rely on it. The oracle grid engine is distributed resource management software that provides high utilization, better throughput and high end-user productivity from existing computer resources [3, 5]. By effectively selecting the resources, the engine distributes the workload efficiently among the nodes shielding the end users from inner working of the cluster. The map reduce is the key feature of this engine.

A grid engine consists of execution machines, a master machine and a zero or more shadow master machines, shown in Fig. 5. The grid engine execution daemon is run by execution machines. The shadow daemon is run by the shadow master machines. If the master machine fails, the shadow daemon on one of the shadow master machines will make the machine a new master machine. The master daemon is the heart of the cluster without which no jobs can be submitted or scheduled. The execution daemons are the workhorses which run the job one by one.

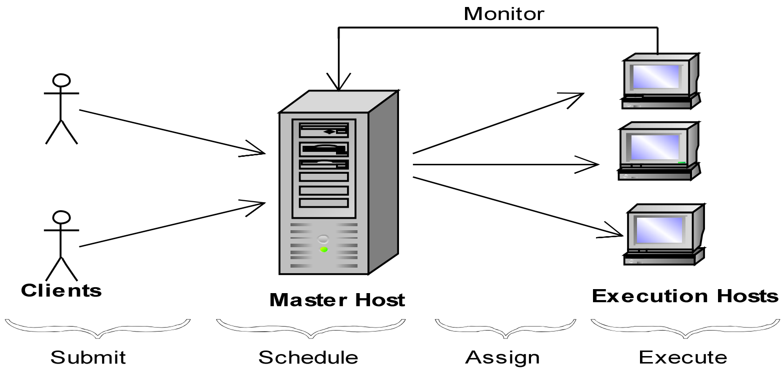

Fig. 5. Oracle Grid Engine

Why to choose Grid Engine?

- *Scalability* – Multiple customers running tens of millions of jobs with thousands of machines.
- *Flexibility* – Easy customization of the system to fit the needs exactly.
- *Advanced Scheduler* – Scheduling policies configure the Grid Engine to make its scheduling decisions match the enterprise business rules.
- *Reliability* – Requires only a little effort to maintain to provide high performance.

The distributed cluster test bed is deployed on top of a multi-cloud infrastructure [4]. The test bed includes a virtual cluster [6] deployed in a local data centre, with a queuing system managed by Oracle Grid Engine (OGE) software, and an OGE master host, a front end and a fixed number of virtual worker nodes. When the user submits the job request to OGE, it schedules the task and distributes it to the worker nodes which in turn process it and the result back to the user.

4.1. Benefits of the proposed system

- Cloud interoperability management through API prevents the vendor lock-in and provides high availability and fault tolerance.
- 10000 jobs of Monte-Carlo simulation.
- Use of Virtual Private Network (VPN) technology to interconnect the different cloud resources with the in-house data centre infrastructure in a secure way.
- Map-Reduce to schedule jobs to various worker nodes.
- Dynamic node identification in a network.
- The in-house data centre infrastructure is connected to the clouds through VPN. The VPN server with a bridge and a tap interface will be at in-house node and a VPN client at each cloud. The server and clients are connected by open VPN tunnels. A new network interface which is directly connected to the data centre LAN is enabled by each client.
- Password protection to the cloud access.

4.2. Algorithm for worker node implementation

```
Split number of iterations
Specify number of iterations per node using
Math. round (iterations/ (float) grid size)
Specify number of iteration for the last node using formula
Iterations-(gridsize-1)*iterPerNode
Add job references to the split
IF (nodes == 0) -→      Call Rackspace
IF (nodes == 1) →       Call Amazon
IF (nodes >1)→Call in-house node
```

(ii) Algorithm for Amazon Service Client

```
Create Method instance
Send HTTP request
Execute Method and get Status Code
Read the response body
Deal with the response
Use caution: ensure correct character encoding and is not binary data
Perform task and return results to user
```

(iii) Algorithm for Rack space Service Client

```
        Send RMI request
        Perform task
        Return result to user
```

(iv) AES algorithm for secured data transfer

```
Generate plain text
Add Round Key
Expansion key – using Rijindael’s key
Perform bitwise XOR
Round 1
      Substitute Bytes - using non-linear substitution according
to lookup table
        Shift Rows
        Mix Columns – Combines four bytes in each column
        Add Round Key
Round 2
        //No Mix Columns in case of final round
        Substitute Bytes
        Shift Rows
        Add Round Key
```

5. Results and discussion

Once the task is completed by each worker node it is then gathered and given back to the user. Table 2 clearly shows the results for the implementation of multi-cloud virtual infrastructure. It denotes the node type, the time taken for simulation by each node in milliseconds (msecs), the number of iterations for each node and finally the total time taken for the given 10000 iterations in milliseconds. This resultant values will not be constant every time since it depends upon the network and dynamic scheduling done by the Map Reduce of the grid gain engine. AES also provides high security to data and password protection key prevents invalid access.

Table 2: Time taken to execute Monte-Carlo Simulation in cluster nodes

Node Type	Time taken for simulation (msecs)	Number of iterations	Total time taken for 10000 iterations (msecs)
Rackspace	3619	3333	22,439
Amazon	2380.0	3333	22,439
In-house node	1934	3334	22,439

5.1. Performance analysis

Performance analysis [11] is been done for individual nodes such as in-house node – Fig. 6, Amazon – Fig. 7 and Rack space for three rounds, Fig. 8, and also with the combination of these three nodes – Fig. 9. The study reveals that the throughput, i.e., the number of jobs executed per second varies for each round and also the throughput is high in the case of the combination of in-house, Amazon and Rackspace nodes indicating that performance is high in the case of multi-cloud deployment rather than using a single cloud.

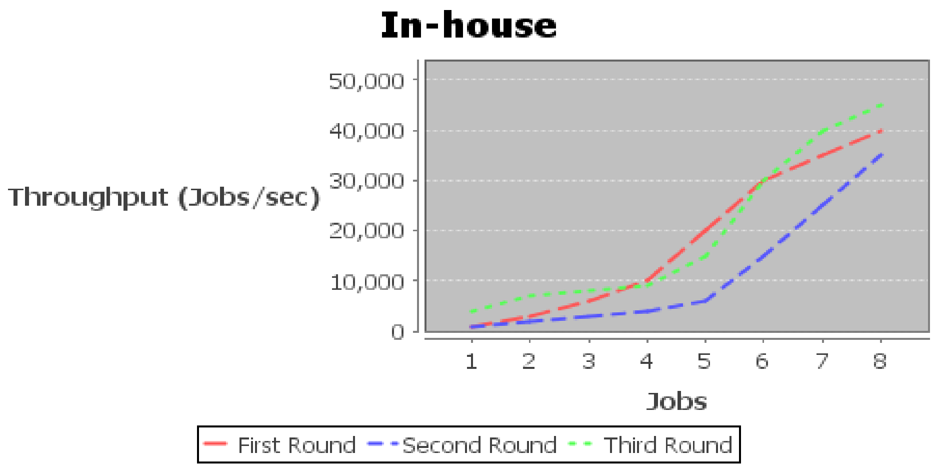

Fig. 6. Performance graph for in-house node

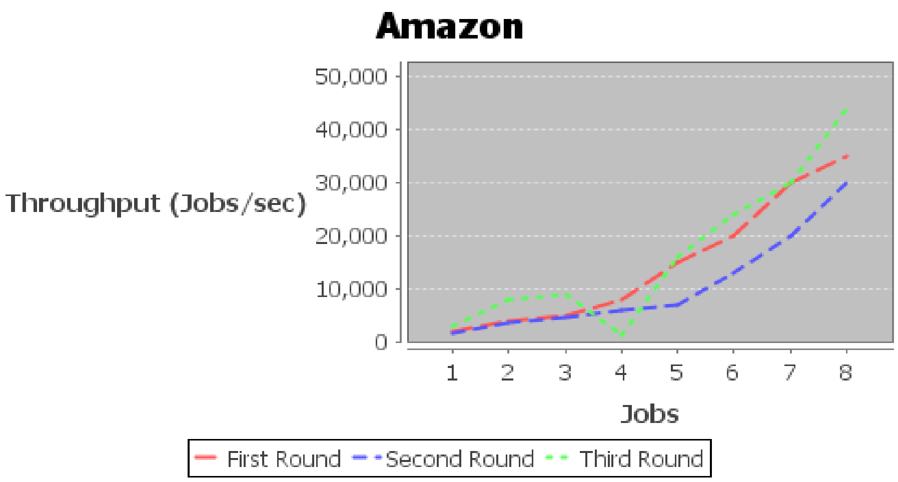

Fig. 7. Performance graph for Amazon Cloud

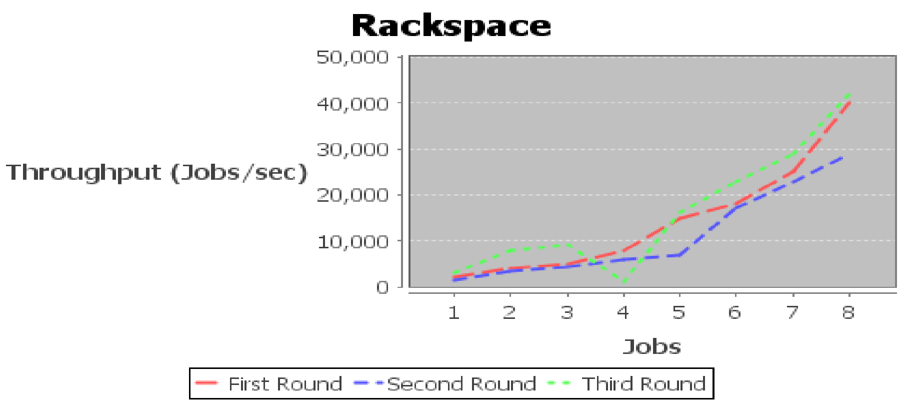

Fig. 8. Performance graph for Rackspace Cloud

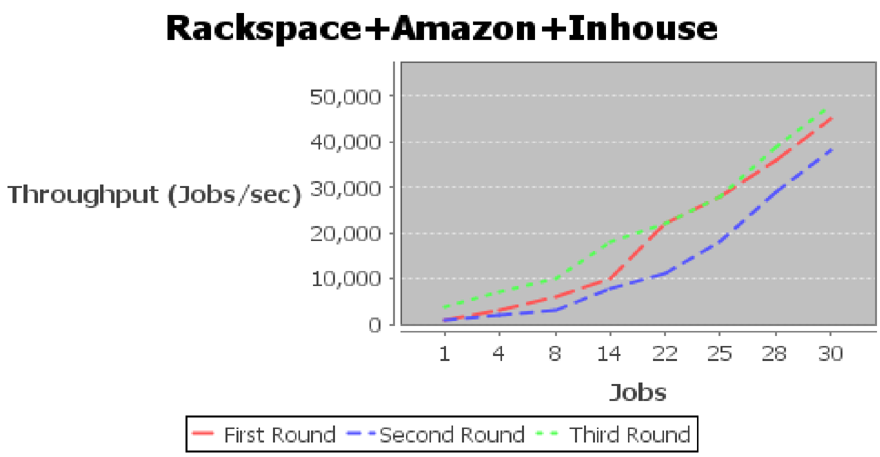

Fig. 9. Performance graph for all nodes together

5.2. Cost analysis

The cost analysis [12] is also been done along with the performance analysis since cost plays a vital impact for setting up a multi-cloud infrastructure. A comparative study is done with the three nodes namely In-House Node (IHN), Amazon Cloud (AC) and Rackspace Cloud (RC). The result of the study states that performance increases by using multi node deployment rather than using a single node. The higher the number of nodes, the higher the performance is, shown in Figs. 10, 11 and 12.

Number of node α performance. 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. In spite of the higher cost of cloud resources with respect to in-house nodes, the other configurations exhibit better performance-cost ratio than in-house node. This proves that the proposed multi-cloud implementation is an efficient and effective solution, not only from the performance point of view, but also from the cost perspective.

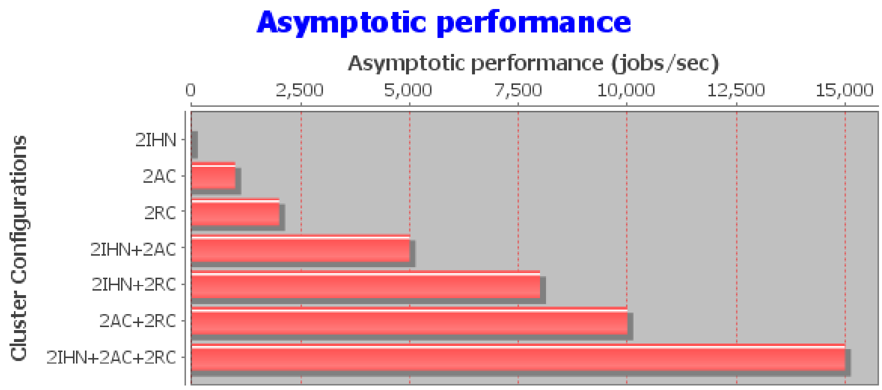

Fig. 10. Graph representing Asymptotic Performance

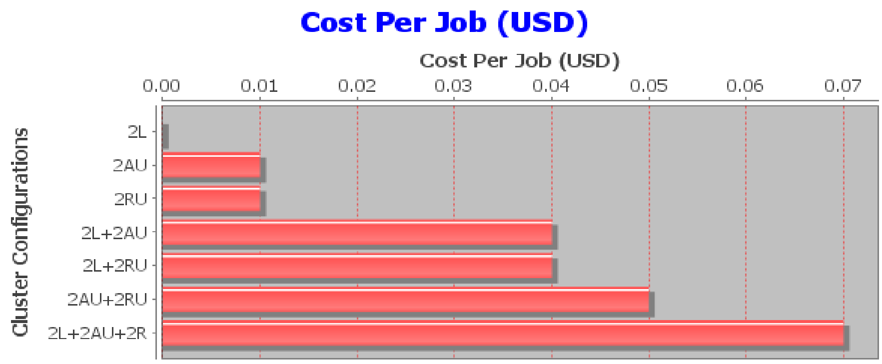

Fig. 11. Graph representing Cost for each nodes

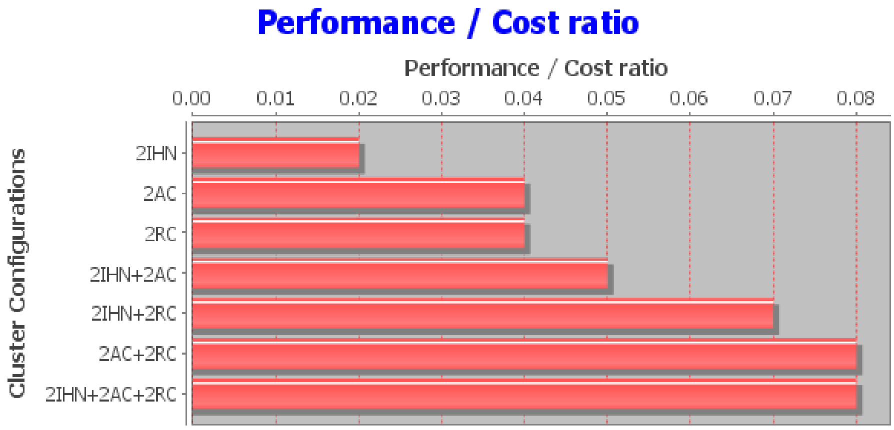

Fig. 12. Graph representing Performance-cost ratio

6. Conclusion and future work

A secured multi-cloud virtual infrastructure was successfully implemented using the oracle Grid Engine. We implemented a real test-bed comprising of in-house infrastructure and external resources from different clouds Amazon and Rackspace. The Map Reduce feature of Grid Gain Engine executed the optimizing criteria and scheduling policy efficiently through API support.

High security for data transfer was provided through Advanced Encryption Standard and security to cloud access was provided through the generation of password protection key.

The features, such as high availability and fault tolerance were provided through dynamic allocation of the nodes. The numbers of iterations were equally divided among the nodes and also the architecture supports interoperability. The throughput increases linearly as the number of nodes of clouds inside the cluster increases. As the cloud nodes cause no overhead, it doesn't cause any performance degradation. The hybrid configurations of in-house and remote nodes exhibit better performance-cost ratio proving multi-cloud solution is good in cost perspective also. Thus the infrastructure cost is also reduced.

The paper is successfully completed and implemented with all the basic needs and requirements that are described. This work can be further enhanced in future by adding more additional features like providing a generic job template in cloud so that it can process any kind of jobs effectively and efficiently with minimum time.

References

1. Wright, P., T. Harmer, J. Hawkins, Y. L. Sun. A Commodity-Focused Multi-Cloud Marketplace Exemplar Application. – In: IEEE 4th International Conference on Cloud Computing, 2011, 590-597.
2. Rochwerger, B., D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P.

M a s s o n e t, H. M u ñ o z, T o f e t t i. Reservoir – When One Cloud is Not Enough. – IEEE Journals & Magazines, Vol. **44**, 2011, Issue 3, 44-51.
3. C a r y e r, G., J. G a l l o p, J. G r a b o w s k i, T. K o v a c i k o v a, T. R i n g s, I. S t o k e s-R e e s. Grid/ Cloud Computing Interoperability, Standardization and the Next Generation Network (NGN). – In: IEEE Conference, October 2009, 1-6.
4. M o r e n o-V o z m e d i a n o, R., R. S. M o n t e r o, I. M. L l o r e n t e. Multi-Cloud Deployment of Computing Clusters for Loosely Coupled MTC Applications. – IEEE Computer Society, IEEE Transactions on Parallel and Distributed Systems, Vol. **22**, June 2011, No 6, 924-930.
5. D e m c h e n k o,Y., J. v a n der H a m, V. Y a k o v e n k o. On-Demand Provisioning of Cloud and Grid Based Infrastructure Services for Collaborative Projects and Groups. – In: IEEE Conferences, Collaboration Technologies and Systems (CTS), 2011, 134-142.
6. M u r p h y, M., B. K a g e y, M. F e n n, S. G o a s g u e n. Dynamic Provisioning of Virtual Organization Clusters. – In: Proc. 9th IEEE Int’l Symp. Cluster Computing and the Grid, 2009.
7. L o v e s u m, D. D., S. P. J. A Novel Approach for Scheduling Service Request in Cloud with Trust Monitor**. –** Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011, 509-513.
8. H w a n g, J.-J., H.-K. C h u a n g, Y.-C. H s u, C.-H. W u. A Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service**. –** In: ICISA International Conference, 2011, 1-7.
9. R a y, C., U. G a n g u l y. An Approach for Data Privacy in Hybrid Cloud Environment**. –** In: ICCCT 2nd International Conference, 2011, 316-320.
10. H a n, S., J. X i n g. Ensuring Data Storage Security Through a Novel Third Party Auditor Scheme in Cloud Computing**. –** In: Cloud Computing and Intelligence Systems (CCIS), IEEE International Conference, 2011, 264-268.
11. M o n t e r o, R. S., R. M o r e n o-V o z m e d i a n o, I. M. L l o r e n t e. An Elasticity Model for High Throughput Computing Clusters. – Journal of Parallel and Distributed Computing, MAY 2010.
12. S i m a r r o, J. L. L., R. M o r e n o-V o z m e d i a n o, R. S. M o n t e r o, I. M. L l o r e n t e. Dynamic Placement of Virtual Machines for Cost Optimization in Multi-Cloud Environments. – IEEE, July 2011, 1-7.
13. Amazon Cloud, 2011.
http://www.amazonaws.com
14. Rackspace Cloud, 2011.
http://www.rackspace.com
15. **http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.186**