Resource Grid Architecture for Multi Cloud Resource Management in Cloud Computing

Chittineni Aruna[1] and R. Siva Ram Prasad[2]

[1] Department of CSE, KKR & KSR Institute of Technology and Sciences,
Acharya Nagarjuna University,
Guntur, Andhra Pradesh, India
chittineni.aruna@gmail.com
[2] Department of CSE and Head, Department of IBS,
Acharya Nagarjuna University, Guntur, Andhra Pradesh, India

Abstract. Multi Cloud Architecture is an emerging technology in cloud computing to improve the security of data and process management in single clouds. Dynamic Resource Allocation is a proven technique for efficient Resource Management in Single Cloud Architecture to mitigate resource overflow and underflow problems. Resource Management in Single Cloud Architecture is an improving area and still it has to address many problems of resource management like resource bottlenecks. Resource Management at single cloud always recommends the additional resources adoption due to frequent resource underflow problems. In this paper we proposed a Resource Grid Architecture for Multi Cloud Environment to allocate and manage the resources dynamically in virtual manner. This architecture introduces a resource management layer with logical resource grid and uses the virtual machines to map the resources against physical systems of single cloud. Experiments are proving that our Resource Grid Architecture is an efficient architecture to manage the resources against multiple clouds and supports the green computing.

Keywords: Resource Grid Architecture, Dynamic resource management, Multi Cloud Environment, Cloud Computing, Green Computing.

1 Introduction

Cloud computing is the new age technology offers on-demand, pay-per-use, reliable and QoS (Quality of Service) resources as IaaS, PaaS and SaaS over internet. Cloud migration is increasing rapidly in cloud computing to achieve the cloud benefits from non-cloud based applications. Day to Day improving features and advancements of cloud computing are assuring the future and attracting the organizations and stake holders towards cloud adoption. This rapid growth of cloud computing needs high level security and huge resources like memory, process and storage to satisfy the SLA [1].

Cloud data security, Process tampering prevention[3] and resource management are the three important aspects of cloud computing to concern. Recent researches[4, 5 and 6] raised numerous issues on cloud data security and trustworthiness of cloud

S.C. Satapathy et al. (eds.), *Emerging ICT for Bridging the Future – Volume 1*,
Advances in Intelligent Systems and Computing 337, DOI: 10.1007/978-3-319-13728-5_71

service providers at cloud data center level. George and Pavlos[6] et al, assumes that present cloud is following the unsecured Honest but Curious model for data and process management. To mitigate the security issues and to prevent the process tampering in single cloud architecture, recently multi cloud architecture [7, 8] emerged as an alternative. This approach distributes the encrypted data and process among multiple clouds to resist from disclose of complete data by compromising of any single cloud. Federating multiple clouds not only improves the security, but also provides the additional benefits to cloud like high availability, auto elasticity, transparency and trustworthiness. Efficient resource management in cloud architecture can save the 15 to 25 percent of resources from additional resource budget. Present multi cloud architectures are still suffering from resource bottlenecks [2] and scalability issues of SLA due to decentralized resource management. This inefficient resource management needs frequent extension and wastage of process, memory and storage resources at every single cloud level and causes to resource overflow and underflow problems.

In this paper to address the above discussed problems, we proposed a Resource Grid Architecture (RGA) for Multi Cloud Environment to allocate and manage the resources dynamically in a virtual manner. RGA federates the resources of each single cloud of multi cloud environment to create the centralized virtual resource pool. This pool is called as Resource Grid (RG), is a collection of virtual machines (VM's) controlled by Virtual Resource Manager (VRM). Every virtual machine is a representation of Physical System (PM) at single cloud operation level to identify the resource allocation needs and management. This architecture introduces a new resource management layer with logical resource grid and uses the virtual machines to map the resources against physical systems of single cloud. RGA includes the load management system as an integral part to assess the resource requirements of multiple clouds to mitigate resource overflow and underflow problems. Experiments are proving that our RGA is an efficient architecture to manage the resources against multiple clouds and supports the green computing.

The rest of the paper is organized as follows. Section 2 discusses about multi cloud architecture, resource management techniques and section 3 provides the implementation of RGA architecture. Section 4 depicts the experimental setup, simulations and section 5 concludes this paper. Section 7 lists the main references of this paper to design.

2 Related Work

In this section we discuss about the need of multi cloud architecture and resource management efficiency in cloud.

Today public clouds are offering the three prominent cloud service layers SaaS, IaaS and PaaS over internet, which made the user sensitive data asset availability from intranet environment to internet environment for remote access flexibility and data globalization. This aspect raises various data and process security issues [7, 8], due to the less control on data to data owner and full control on data to service provider. Cloud service provider may be honest but curious [6] also under some circumstances. Several researches were introduced the Third Party Auditing [9] and Client Side Data Encryption [10] to mitigate the unauthorized cloud data access and process tampering in cloud. These encryption techniques are having scalability problems and complex to manage to satisfy SLA. To avoid these problems in cloud, Bohli and Jensen [7] et al proposed multi cloud architecture by federating the single clouds of cloud computing environment. This architecture becomes an alternative to single cloud to improve the security for data and application by using multiple clouds simultaneously. In order to resist the adversary attacks this architecture uses the data and process distribution among multiple clouds. The single cloud and multi cloud architectures are shown in Figure 1.

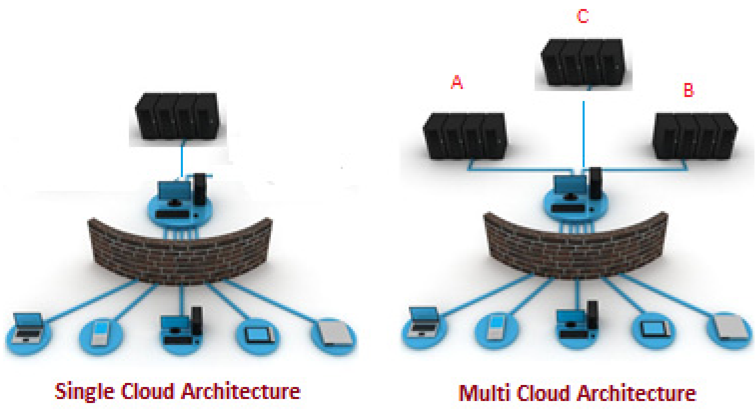


Fig. 1. Single and Multi Cloud Architectures

Resource management is an important concern in cloud computing to manage the primary resources like memory, process and storage etc. Rapid growth of cloud computing demands high amount of resources to rent cloud users and to run the cloud applications. This is a frequent requirement for every single cloud depends on the resource usage and prediction. Static allocation of resources [12] in cloud cannot be used effectively because of resource need variations for an application over time. These variations of resource usage create overflow and underflow problems in resource allocation. To avoid these problems resources should be adjusted dynamically for efficient resource management in cloud architecture. Dynamic resource management [11] reduces the resource contention, scarcity of resources, resource fragmentation, and overflow, underflow problems. In this research we create a resource pool to manage multiple cloud resources effectively through corresponding virtual machines.

3 Resource Grid Architecture

This section gives the comprehensive explanation and implementation of our Resource Grid Architecture for Multi Cloud Environment. RGA is logically cross cut into four layers for efficient resource management in multi cloud environment as shown in figure 2.The layers are i) Cloud Layer ii) Network layer iii) VM layer iv) Resource Grid Layer.

3.1 Cloud Layer

This layer is a collection of multiple remote clouds with available physical resources like storage, network, process, memory. Resource Scheduler, Utility Tracker, Local Resource Manager and physical resources are the parts of this layer. Local Resource Manager is the head of Resource Scheduler and Utility Tracker. Utility Tracker is a tool to track the resource utilization at single cloud level based on certain interval period of time. This tool enquires the allocated resources to cloud application to track the record of resource utilization and sends this information to cloud Resource Prediction Unit for hotspot and cold spot detection. Utility Tracker is a prescheduled program which runs automatically over the specified period of time to track usage and to upload this data to Local Resource Manager. In order to allocate the additional resources to any application of cloud, Local Resource Manager gets the tracker report from utility tracker and gives the instructions to resource scheduler to allocate resources and starts the new job. Resource scheduler is not only allocating the required resource to running application but also working as a job scheduler to start and stop the jobs. This is simply an extension to job scheduler program of OS. Local Resource Manager is a module to verify the resource allocation and utilization at single cloud level and is an agent of Global Resource Manager. Extracted tracking information periodically updated by each Local Resource Manager to Resource Grid Layer for allocation analysis. This module is responsible to periodically generate a Resource Occupancy Chart (ROC), which summarize the available resources, allocated resources, over flow and underflow problems. Local resource manager always works as per the Global Resource Manager instructions and works like as a virtual node in this architecture.

3.2 Network Layer

This layer is the conjunction between cloud layer and the resource grid layer and having collection of routers, firewalls, HDMI interfaces etc. Secured communication relevant all aspects are implemented in this layer along with https like secured protocols. This layer contains the high speed wired and wireless network components to support high speed commutation among clouds. This layer is the backbone for this architecture because intra cloud communication is highly demanded in this implementation for resource sharing and process sharing.

3.3 VM Layer

This layer is a list of virtual machine groups where each group is a set of virtual machines for physical system mapping. In this layer we proposed a separate virtual machine group for single cloud architecture to achieve the operational feasibility. Each VM is a virtual representation of a cloud physical machine at VM Layer level. This is the prominent way to perform mapping between logical resources usage with physical requirements. Every VM is having the OS part for virtual process management and the data part for data manipulations. Virtual machines are always available to resource grid layer monitoring and resource mapping.

3.4 Resource Grid Layer (RGL)

RGL is a collection of management modules of this architecture like Global Resource Manager (GRM), Load Balancing System (LBS), Monitoring System, Resource Allocation Tracker, Resource Prediction Unit (RPU) etc as shown in figure 2. All of these modules are interlinked together to monitor, analyze, allocate and track the resource information of multiple clouds at a centralized virtual location. This layer is called as a grid because of this is a centralized pool of logical resources, which predicts the requirement of resources based on past statistics. In this architecture, we introduce the logical federation of resources from multiple single clouds instead of physical collaboration to achieve the resource management. With the inspiration of water Grid Management System from Indian government, that collaborate all available small water plants together to adjust the water to avoid drinking water problem in all seasons. Single cloud environments are not good enough to manage the resources, due to the problems of limited resources, inefficient resource management system and frequently occurring resource overflow and under flow problems. To mitigate all these problems at single cloud level we are implemented a multi cloud resource management layer to share the resources over clouds. In this case, our RGL connects all individual clouds and adjusts one cloud resources with another cloud via network layer to support Green Computing [13].

Global Resource Manager is the vital role in this layer and to make decisions of resource allocation and to mitigate resource congestion problem in this architecture. Monitoring System is the management level monitoring module, which collects the periodic reports from local resource manager of each single cloud. This system prepares the Cloud Health Charts for each cloud of multi cloud architecture to represent the current status of resource usage and availability. Resource prediction unit is the integral part of the Resource Grid Layer to assess the future requirements of individual cloud based on their recent utilization statistics at cloud applications and physical machines level. In the same way this will calculate at all clouds level and generates a Resource Occupancy Chart (ROC) to give the clarity about every cloud level available resources, allocated resources, overflow of resources and underflow of resources. This module prepares another report is Future Resource Prediction Chart (FRPC) by estimating future requirements at each cloud level and handover this to Global Resource Manager. By considering this report GRM suggests the additional resources requirement for future to individual clouds. Load balancing System takes the ROC as input from Resource Prediction Unit to level the overflow and underflow of resources at single clouds. This is a knowledgeable system with an efficient Dynamic Resource Allocation Unit to adjust the resources and processes among multiple clouds by mapping with respective virtual machines. Load Balancing System finds the resource hot spots and cold spots at every cloud level to balance the resources to mitigate resource overflow and underflow problems. After mapping the resources to VM's, allocation information will send to allocation tracker at resource grid level. This tracker is having a dashboard to display utilization ratio at cloud level, grid level and displays the resource savings of each cloud by becoming a part at Resource Grid.

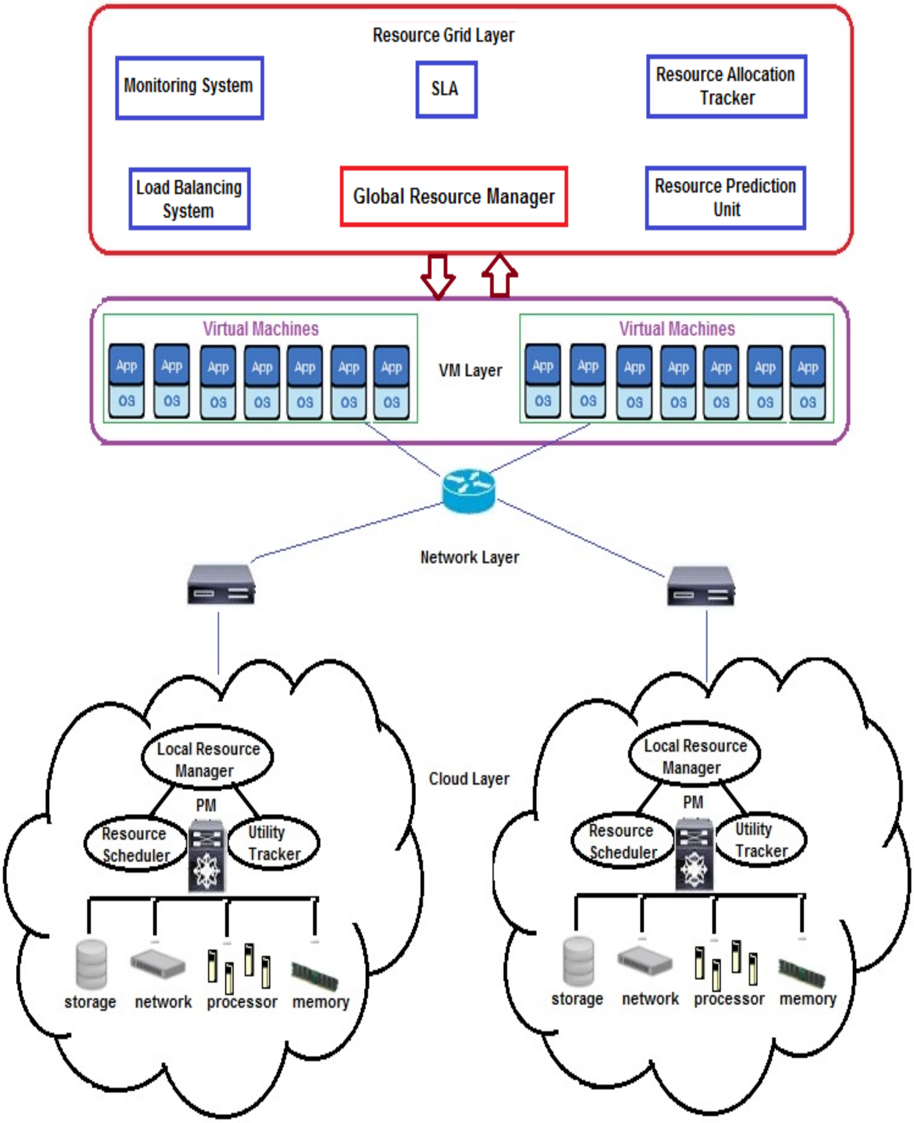


Fig. 2. Resource Grid Architecture (RGA) for Multi Cloud Environment

4 RGA Experimental Setup and Process Flow at a Glance

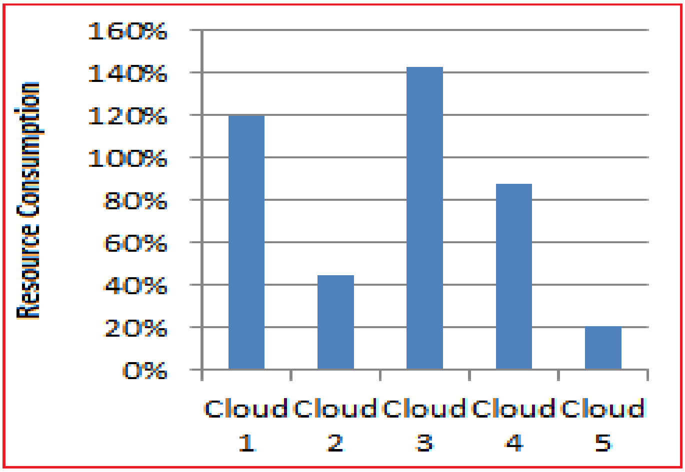
In this section we discuss about the experimental setup of RGA and the process flow along with the obtained result information at glance.

4.1 Experimental Setup

As the architecture is having multiple clouds as a part of this, it is important to create a large scale clouds for experiments. However it is extremely complex to setup the large scale clouds for experiments, we created 5 different single clouds with each cloud having more than 100 physical nodes with commodity hardware at every system level. These 5 clouds are complete heterogeneous and connected through a high speed wireless network system for intra communication to each other and the resource grid layer. We implemented the resource grid layer as a part of one cloud of 5 clouds and allocated an independent processor and memory pool for that module. Every commodity (Physical Machine) system is having the Linux (Ubuntu), 8 GB RAM, 1 TB hard drive and AMD8 Processor with 3.5 GHZ speed. To simulate the cloud platform we used the CloudSim Toolkit[14], which creates a Virtual Machine (VM) environment to map with the physical systems of cloud. Each VM is running an application with various workloads and having the individual process execution environments mapped with available static resources. A set of VM's of same cloud are forming a VM cluster to deviate from cloud to cloud. After a trial run of individual cloud with static resources, resources are adjusted as per resource requirement. This allocation may need to update at runtime depends on application workload leads to resource overflow or under flow. We run this experimental setup for 3 days to find hidden flaws to make this architecture more reliable.

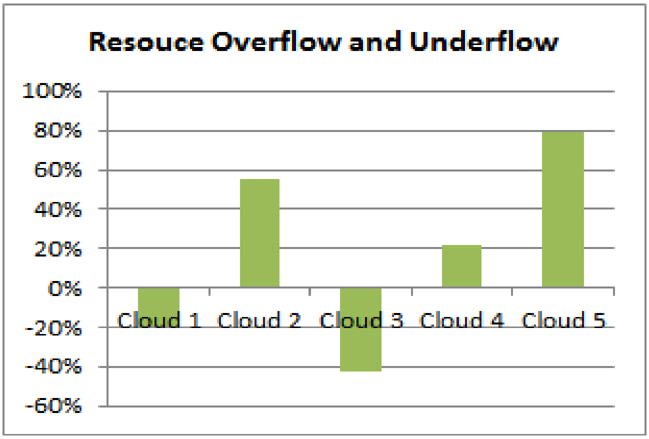
4.2 RGA Process Flow

While running the experiments the process flow of this architecture is described in this section. As per the above discussion every cloud is deployed with physical resources to map with VM's which are running the user applications. After the specific period amount of time the report generation and updating to next higher level will start at every possible module of this architecture. This time vary from module to module subject to complexity and need. This reporting is divided to multi hop environment, where at each hop is connected to its next level and having various modules to generate different reports for process. Initially the utility tracker generate the resource availability and allocation at cloud individual physical machine level and submits to the first hop manager is Local Resource Manager. After trail run execution we set the periodical reporting time difference as 5 min. This report will generate at every cloud level automatically and sends to Local Resource Manager for every 5min. At this stage, Local Resource Manager will connect to the second hop and updates the report to monitoring system through network layer. This system will get the similar report from multiple clouds and forwards this to Resource Prediction Unit to generate ROC and FRPC reports. Resource Prediction Unit forwards the ROC to Load Balancing System and FROC to GRM. As per the information of ROC, Load Balancing System adjusts the resources of one cloud with another by considering the overflow and underflow of resources at each cloud level. At the same time, concurrently this will updates the adjusted information to Resource Allocation Tracker, which maintains a dash board to display the statistics of resource utilization. For every 15 min this dash board will update with the available resources at grid, allocated resources at grid and overflow, underflow problems at every individual cloud level as shown in below graphs.



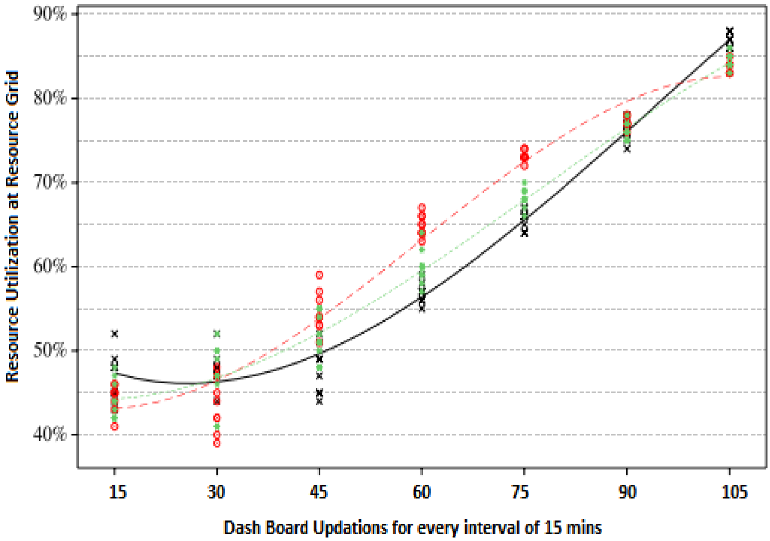
Resource Occupancy Chart by RPU

Graph 1(a). Resource Occupancy Chart by RPU



Resource Overflow and Underflow chart from RAT

Graph 1(b). Overflow and underflow graph by RAT at multi cloud level



Graph 2. Dashboard resource utilization updates at Resource Grid Level

5 Conclusion

In this paper we are introducing an efficient new architecture to implement the resource management in cloud computing by creating a Resource Grid at Multi Cloud level. Logical integration of cloud resources are proven to mitigate the resource overflow and underflow problems at every individual cloud level, without adopting new resources. Virtual Machines layer with executing processes are mapped with physical resources to adjust resources dynamically as per the load balancing system. This approach is having multi hop reporting mechanism to send the various analysis and utilization statistics to next level for efficient resource management at multiple clouds level. Experimental results are showing that RGA is having the high scalability than resource management at individual cloud environments. Our future work concentrates on how to recommend the additional resources adoption to a cloud owner and updating the billing system for an individual cloud to pay for usage of other cloud resource usage.

References

1. Bobroff, N., Kochut, A., Beaty, K.: Dynamic placement of virtual machines for managing sla violations. In: Proc. of the IFIP/IEEE International Symposium on Integrated Network Management (IM 2007) (2007)
2. Chase, J.S., Anderson, D.C., Thakar, P.N., Vahdat, A.M., Doyle, R.P.: Managing energy and server resources in hosting centers. In: Proc. of the ACM Symposium on Operating System Principles (SOSP 2001) (October 2001)

3. Jensen, M., Schwenk, J., Gruschka, N., Lo Iacono, L.: On Technical Security Issues in Cloud Computing. In: Proc. IEEE Int'l Conf. Cloud Computing, CLOUD-II (2009)
4. Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., Molina, J.: Controlling data in the cloud:outsourcing computation without outsourcing control. In: Proceedings of the 2009 ACM Workshop on Cloud Computing Security, pp. 85–90. ACM (2009)
5. Vijayan, J.: Vendors tap into cloud security concerns with new encryption tools, http://www.cio.com.au/article/376252/vendorstapintocloudsecurityconcernsnewencryptiontools/
6. Drosatos, G., Efraimidis, P.S., Athanasiadis, I.N.: A privacy-preserving cloud computing system for creating participatory noise maps. In: IEEE 36th International Conference on Computer Software and Applications, 0730-3157/12. IEEE (2012)
7. Bohli, J.-M., Jensen, M., Gruschka, N., Schwenk, J., Iacono, L.L.L.: Security Prospects through Cloud Computing by Adopting Multiple Clouds. In: Proc. IEEE Fourth Int'l Conf. Cloud Computing (CLOUD) (2011)
8. Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., Morrow, M.: Blueprint for the Intercloud—Protocols and Formats forCloud Computing Interoperability. In: Proc. Int'l Conf. Internet andWeb Applications and Services, pp. 328–336 (2009)
9. Hulawale, S.S.: Cloud Security Using Third Party Auditing and Encryption Service. M.S Thesis (July 2013)
10. Li, M., Yu, S., Zheng, Y., Ren, K.: Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption. IEEE Transactions on Parallel and Distributed Systems 56 (2012)
11. Song, Y., Wang, H., Li, Y., Feng, B., Sun, Y.: Multi-Tiered On-Demand resource scheduling for VM-Based data center. In: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, vol. 00, pp. 148–155 (2009)
12. Harchol-Balter, M., Downey, A.: Exploiting process lifetime distributions for load balancing. ACM Transactions on Computer Systems (TOCS) 15(3), 253–285 (1997)
13. http://searchdatacenter.techtarget.com/definition/green-computing
14. Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F.D., Buyya, R.: CloudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience 41(1), 23–50 (2011)