International Journal On Engineering Technology and Sciences – IJETS™
ISSN(P): 2349-3968, ISSN (O): 2349-3976
Volume III, Issue X, October- 2016

Resource Allocation for Web Applications under Integrated Cloud Architecture

Ms. C. Deepika, MPhil., Research Scholar,
Dr. N. Chandrakala, MSc., MPhil., PhD., Head/Department of Computer Science.,
SSM College of Arts & Science, Komarapalayam, Tamilnadu, India

Abstract

The web applications are installed in the Internet environment to manage commercial and scientific workloads. The web applications are executed with the support of the web client tools. Request based resource utilization is highly increased in the web applications. Public and private cloud resources are combined to form integrated cloud architecture. The hardware and software resources under the cloud environment are provided for the web applications. The web applications are migrated from the web server to the cloud architecture. Service load and data sharing operations are redirected to the cloud resources. Economical, spatial and time parameters are adapted for the workload migration process.

Cloud services are employed to share the hardware and software resources for the web workloads with reference to the cost and time factors. CloudGenius middleware is adapted to handle the web application migration process. Workload movement is carried out with multiple component based model. The CloudGenius middleware is constructed with Analytic Hierarchy Process (AHP). The workload movement decisions are finalized with Parallel Genetic Algorithm (PGA). Web workloads are spited into various components to reduce the complexity in resource sharing process.

The Enhanced CloudGenius mechanism is build to manage resource allocation under integrated cloud architecture. Customized parameter based resource and data sharing operations are supported in the system. Location of the data and operational sequence are considered in the workload movement process. Proximity based data center optimization is utilized in the workload movement process. Clustering techniques are integrated with the system to group up the resources and workload information for the resource sharing operations.

1. Introduction

The continuing growth of cloud computing and its immediate uptake by the industry have yielded a large number of cloud-based services spanning the infrastructure, platform, and software levels. As the demand for cloud services continue to increase at a global scale, so does the energy consumption of the service providers' data centers and, ultimately, their negative impact on the environment. Nowadays, enterprise-scale cloud computing infrastructures consume huge amounts of electrical energy, contributing to high operational costs and carbon footprints to the environment. Still, the majority of enterprise cloud data centers utilize only a fraction of their available resources, while a considerable part of their power consumption is lost due to both over-provisioned and idle resources. It therefore becomes important for cloud service providers to adopt appropriate measures in order to attain energy-efficient processing and utilization of their computing infrastructure.

In computation–intended data centers, the workload gets essentially translated into a number of provisioned virtual machine (VM) instances. To address the aforementioned problems in such settings, the technology of dynamic VM consolidation has been devised, widely studied, and applied. In a nutshell, dynamic VM consolidation continuously strives to reduce the energy consumption of the data center by packing the running VM instances to as few physical machines as possible, and consequently switching off the unnecessary resources. Combined with the use of live VM migration, which refers to the process of moving a running VM instance between different physical compute nodes without disconnecting the client, VM consolidation has become feasible in terms of cost and it can considerably improve the energy footprint of cloud data centers.

Still, in the presence of enterprise clouds consisting of hundreds to even thousands of physical machines utilized for the provision of large numbers of VM instances, energyefficient load balancing through VM consolidation becomes a challenging task. Indeed, the problem of VM consolidation is an applied form of bin packing, which is by nature a combinatorial NP-hard problem and therefore, expensive to compute when large numbers of physical machines and thousands of VM instances are involved. To date, most of the existing approaches rely on centralized, hierarchical, or ring topologies, all of which exhibit certain performance limitations as both data centers and their workload scale out. Consequently, it is critical for cloud service providers to select an appropriate and scalable data center architecture in order to carry out the VM consolidation process in an efficient way.

2. Related Work

There are a lot of works related to our study, and we focus on the most relevant ones on cost optimizations and cloud performance dynamics.

Cost-aware optimizations. Workflow scheduling with deadline and budget constraints has been widely studied. Yu *et al*. proposed deadline assignment for the tasks within a job and used genetic algorithms to find optimal scheduling plans. Multi-objective methods such as evolutionary algorithms have been adopted to study the tradeoff between monetary cost and performance optimizations for workflow executions. Those studies only consider a single workflow with on-demand instances only. Malawski *et al*. proposed dynamic scheduling strategies for workflow ensembles. The previous studies proposed auto-scaling techniques based on static execution time of individual tasks. In comparison with the previous works, the unique feature of Dyna is that it targets at offering probabilistic performance guarantees as QoS, instead of deterministic deadlines. Dyna schedules the workflow by explicitly capturing the performance dynamics in the cloud. Buyya *et al*. proposed an algorithm with task replications to increase the likelihood of meeting deadlines. Due to their ability on reducing monetary cost, Amazon EC2 spot instances have recently received a lot of interests. Related work can be roughly divided into two categories: modeling spot prices and leveraging spot instances. For modeling spot prices, Yehuda *et al*. conducted reverse engineering on the spot price and figured out a model consistent with existing price traces. Javadi *et al*. developed statistical models for different spot instance types.

Those models can be adopted to our hybrid execution. For leveraging spot instances, Yi *et al*. introduced some checkpointing mechanisms for reducing cost of spot instances. Further studies used spot instances with different bidding strategies and incorporating with fault tolerance techniques such as checkpointing, task duplication and migration. Those studies are with spot instance only, without offering any guarantee on meeting the workflow deadline like Dyna. Similar to Dyna, Chu *et al*. proposed a hybrid method to use both on-demand and spot instances for minimizing total cost while satisfying deadline constraint. However, they did not consider the cloud performance dynamics.

Cloud performance dynamics. There have been some proposals to reduce the performance interference and unpredictability in the cloud, such as network performance and I/O performance. This paper offers a probabilistic notion to capture the performance and cost dynamics, and further develop a probabilistic scheduling system to minimize the monetary cost with the consideration of those dynamics.

3. Web Workload Migration in Clouds

Migrations of web applications to the cloud are expected to be linear transitions from the state of "not migrated" to "migrated". Web applications and related software stacks introduce complexity. Migrations should involve multiple repetitions and reconsiderations of past actions. CloudGenius accommodates the vicissitudes within a cloud migration by embedding decision support methods into an evolutionary migration process model. Also, existing migration strategies can be employed within the process model.

CloudGenius' migration process model describes the states of a web application as depicted in a finite state machine. From an initial state a web application is migrated by an application engineer into the cloud with an optional cancellation path. From its migrated state a web application can be migrated within the cloud, be it between providers or services, or due to changes in the software stack. A web application might be replaced or needs to be disposed and moves to an inactive state. CloudGenius provides decision support in the transition phases to a migrated web application, except reactivation. The migration process model describes the steps within a transition and embeds the decision support. model for clusters of web applications in Business Process Model and Notation (BPMN) 2.0. The process is divided into two lanes: (1) "user input" lane representing application engineers and domain experts and (2) "CloudGenius" lane which represents an implementation of the framework.

The process begins with an initial decision between a cloud and non-cloud infrastructure. Subsequently, an engineer states preferences and requirements and lets CloudGenius recommend a list of feasible cluster solutions. Every solution comprises a mapping of a cluster component to a target platform constituted by a VM image and a compute service. In case no satisfying solution has been identified, the process allows ending an infeasible cloud migration. Otherwise, the process continues with migration steps still offering the chance to return to an earlier stage in the process. Thereby, CloudGenius becomes an evolutionary approach that facilitates cycles in a migration. In case of an user-initiated abort in the process, an intentional reset to the fundamental cloud decision activity is forced. This gives an engineer the chance to reconsider the fundamental infrastructure decision or to skip forward and modify retained preference and requirement statements.

International Journal On Engineering Technology and Sciences – IJETS™
ISSN(P): 2349-3968, ISSN (O): 2349-3976
Volume III, Issue X, October- 2016

Eventually, an engineer ends up with a successfully migrated web application cluster. In case of discontinuation of the process, a validation that cloud infrastructures are yet an unsatisfying choice have been attained. In conclusion, the process model supports an evolutionary approach to a migration. The evolutionary nature is facilitated through the chances for reconsiderations by returning to earlier steps within the process or by reapplying the whole process model.

CloudGenius' formal model is extended with the notion of components (C), compatibilities (D, E, F) and network traffic (Nin, Nout) to facilitate clusters. The extended formal model of CloudGenius incorporates l components c_h which is part of a cluster F. Furthermore, the model includes m images a_i and n cloud infrastructure services s_j of o providers p_k. C is the corresponding set of software components in a cluster, A the set of VM images, S the set of cloud infrastructure services, P the set of cloud providers and I the set of component connections. Every VM image a_i, compute service s_j and any combination thereof (x_l) owns numerical and non-numerical attributes noted in the sets A_{ai} , A_{sj} , B_{ai} and B_{sj} . x represents a value connected with a numerical attribute a or non-numerical attribute b. Moreover, the model introduces r_A , r_s r_X requirements:

Based on the model, CloudGenius recommends a best solution for a cluster F with best combinations (a_i, s_j) in X_{ch} for every component c_h. A best solution for a cluster has the highest value according to total benefit versus network traffic costs tradeoff evaluated in a function c(.) for network costs and a utility function u(.) that considers an engineer's preferences. In addition, a best solution needs to conform with set D. In D viable combinations of VM images a_i deployable on computer services s_j are marked. Compatibilities between components are held in sets E and F. Compatible VM images a_i and compute services s_j are marked in E and F respectively. The sets Nout and Nin hold the expected network traffic for component relations. In sum, the problem can be expressed as an optimization problem to find F^* with the highest value.

4. Problem Statement

Cloud service providers' offers computational services and Virtual Machine (VM) images for information systems. Throughput and cost factors are considered in the service selection process. CloudGenius framework is constructed to handle process migration from web applications into public cloud resources. CloudGenius provides migration support for multi-component web applications. Evolutionary migration process for web application clusters is distributed over multiple locations. A multi-criteria-based selection algorithm on Analytic Hierarchy Process (AHP) is employed in CloudGenius model. Parallel Genetic Algorithm (PGA) is applied to select migration solutions. CumulusGenius is an implementation support for CloudGenius framework. The following problems are identified from the current workload migration systems. Hybrid cloud architecture is not supported. Migration cycles delay is high. Provider Customization is not supported. Control and data flow dependencies are not considered.

5. Enhanced CloudGenius framework

Cloud services are deployed to provide resources and service components. CloudGenius framework is adapted to handle workload migration for web applications. CloudGenius is enhanced to support migration under hybrid cloud environment. Component dependency analysis, customization and middleware service integration features are added to the system. The CloudGenius framework is enhanced to support migration under public and private cloud environment. User selection criteria based migration scheme is integrated with the system. Control flow and data dependency analysis mechanism is integrated with the migration system. Middleware services are adapted to support migration tasks.

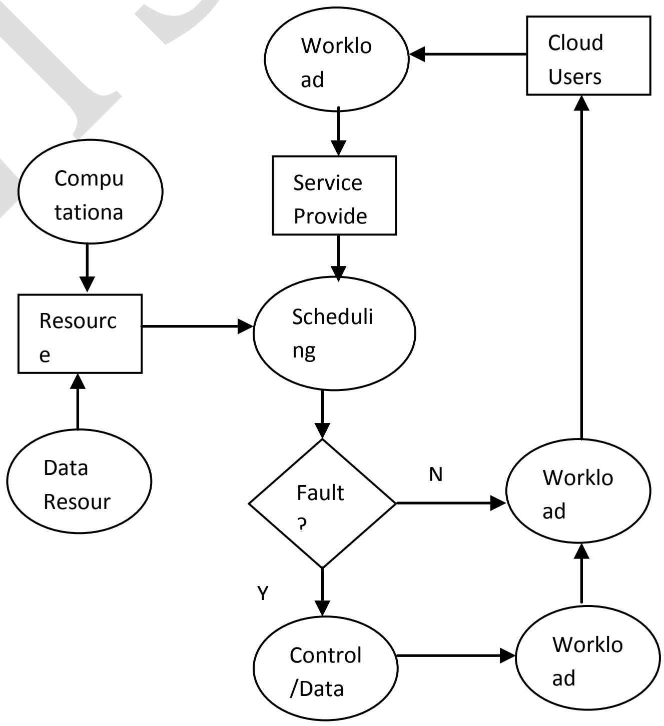

Figure. No: 5.1. Enhanced CloudGenius framework

The web workload migration scheme is constructed to move the workloads from web application to the cloud resource providers. The CloudGenius middleware supports migration on public cloud environment only. The cloud services are provided in the public cloud environment. The data centers and storage services are provided in the private cloud environment. The workloads can be the cloud resources from the public cloud and data values from the private cloud environment. The Enhanced CloudGenius framework is formed to migrate web workloads to the public and private cloud resources. The cloud computational services utilize the data from the private cloud data centers.

The web workload migration is carried out with user selected parameters. The data center data values are transferred to the public cloud resources to execute the workloads that are redirected from the web applications. Control flow and data dependancy measures are verified in the migration process. The migration can be initiated with feasible data center selection process. The data center selection is also carried out with its load and temporal factors. The K-means clustering algorithm is applied to group up the similar resource providers and tasks. Resources and tasks are consolidated in the clustering process. The consolidated resource and task information is analyzed in the scheduling process. The computational resource allocation is carried out with data center selection process. The Parallel Genetic Algorithm is applied to identify the suitable cloud resource providers. Migrated workload results are redirected to the web applications.

6. Experimental Analysis

The cloud resource sharing methods are used to share the resources and services to execute the workloads. The workloads are collected from web applications. The workloads are redirected to the cloud resources. CloudGenius (CG) middleware is designed to handle the workload management process. The Enhanced CloudGenius (ECG) is designed to support workload migration on resource failure conditions. The system is tested different resource and workload levels.

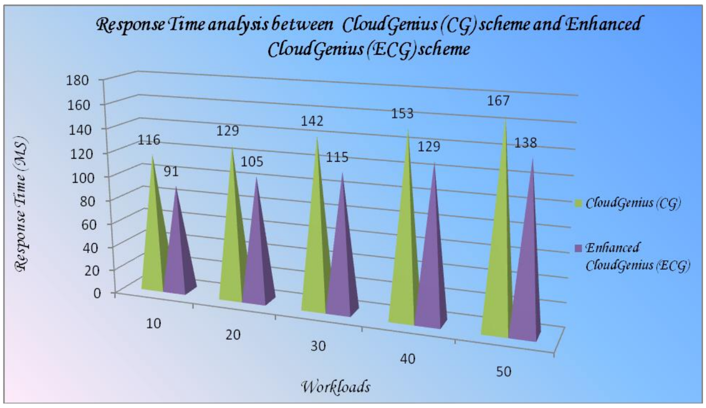

Figure No: 6.1. Response Time Analysis between CloudGenius (CG) scheme and Enhanced CloudGenius (ECG) scheme

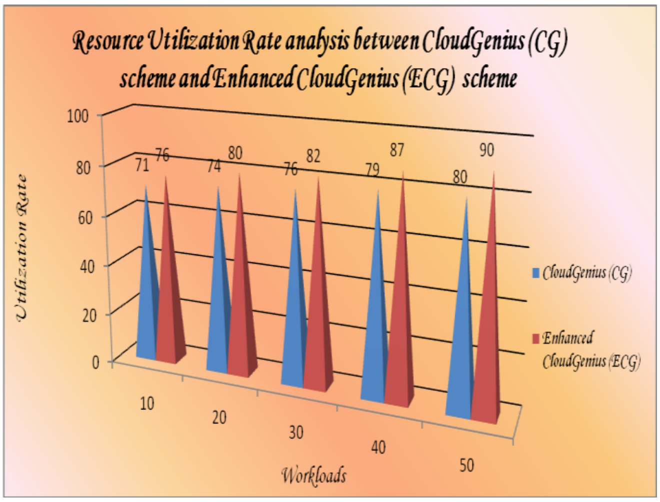

Figure No: 6.2. Resource Utilization Rate Analysis between CloudGenius (CG) scheme and Enhanced CloudGenius (ECG) scheme

The system uses the response time, resource utilization rate and power consumption ratio analysis for the performance process. The response time analysis for the CloudGenius (CG) and Enhanced CloudGenius (ECG) techniques are shown in figure 6.1. The Enhanced CloudGenius (ECG) reduces the average response time 20% than the CloudGenius (CG) model. The resource utilization rate analysis for the CloudGenius (CG) and Enhanced CloudGenius (ECG) techniques are shown in figure 6.2. The resource utilization rate in Enhanced CloudGenius (ECG) is 10% increased than the CloudGenius (CG) mechanism. The power saving analysis for the CloudGenius (CG) and Enhanced CloudGenius (ECG) techniques are shown in figure 6.3. The energy saving ratio in Enhanced CloudGenius (ECG) is 35% reduced than the CloudGenius (CG) model.

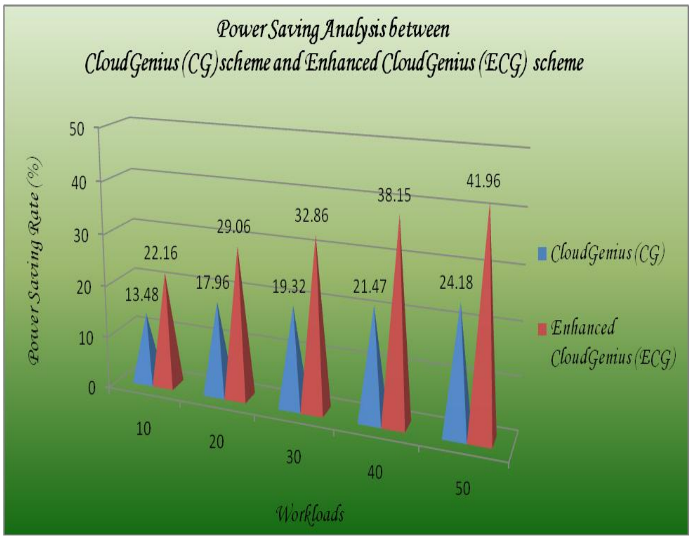

Figure No: 6.3. Power Saving Analysis between CloudGenius (CG) scheme and Enhanced CloudGenius (ECG) scheme

7. Conclusion and Future Work

The web application migration scheme is adapted for the public and private cloud environment. Control flow and data dependencies are analyzed in the migration process. Customization features are adapted in the CloudGenius framework. Workload performances are increased with minimum cost and time. The system can be enhanced to support security features. The system can be adapted to handle the commercial cloud operations.

References

1. A. C. Zhou, B. He and S. Ibrahim, "A taxonomy and survey on escience as a service in the cloud," *arXiv preprint arXiv:1407.7360*, 2014.
2. S. Yi, A. Andrzejak and D. Kondo, "Monetary cost-aware checkpointing and migration on amazon cloud spot instances," *IEEE TSC*, 2011.
3. R. N. Calheiros and R. Buyya, "Meeting deadlines of scientific workflows in public clouds with tasks replication," *IEEE TPDS*, 2013.
4. H. Kloh, B. Schulze, R. Pinto, and A. Mury, "A bi-criteria scheduling process with cos support on grids and clouds," *Concurrency Computat. Pract. Exper.*, 2012.
5. S. Di, C.-L.Wang, and F. Cappello, "Adaptive Algorithm for Minimizing Cloud Task Length with Prediction Errors," *IEEE TCC*, 2014.
6. M. Rodriguez and R. Buyya, "Deadline Based Resource Provisioningand Scheduling Algorithm for Scientific Workflows on Clouds," *IEEE TCC*, 2014.
7. D. Oliveira, K. A. Oca˜na, F. Bai˜ao, and M. Mattoso, "A Provenance- based Adaptive Scheduling Heuristic for Parallel Scientific Workflows in Clouds," *J. Grid Comput.*, 2012.
8. N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in *CLOUD '11*.
9. A. C. Zhou and B. He, "Transformation-based monetary cost optimiza- tions for workows in the cloud," *IEEE TCC*, 2013.
10. B. Javadi, R. K. Thulasiram, and R. Buyya, "Characterizing spot price dynamics in public cloud environments," *FGCS*, 2013.
11. "Iperf," http://iperf.sourceforge.net.
12. Amelie Chi Zhou, Bingsheng He and Cheng Liu, "Monetary Cost Optimizations for Hosting Workflow-as-a-Service in IaaS Clouds", IEEE transactions on cloud computing, 2016
13. Bing Luo, Shinan Wang, Weisong Shi and Yanfeng He, "eCope: Workload-aware Elastic Customization for Power Efficiency of High-End Servers", IEEE Transactions on Cloud Computing, June 2016