Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 11, November 2015, pg.233 – 240

SURVEY ARTICLE

A Survey on PMML based Platform Aware Resource Management in Cloud-PaaS

K.SIVA KUMAR[1], MS.N.SENTHAMARAI, M.E, (PH.D)[2]
[1]PG STUDENT, [2]ASSISTANT PROFESSOR (SR.GR)
[1,2] Easwari Engineering College, TAMILNADU, INDIA
[1] sivakumarme22@gmail.com; [2] senthamaraivijay@gmail.com

Abstract— Popularity of cloud computing has increased many times in the last few years. One major driving force behind this rapid increase in adoption of cloud is the economic benefits that the cloud provides. The benefits imply the economies of scale that go with the pool of configurable computing resources which together constitute the cloud. Cloud frees the user from the job of setting up and maintaining the computational infrastructure and helps him to focus on developing and perfecting his application. Also the cloud provides the benefit of scaling (manual/real-time) so that the application continues to work even under heavy load. However moving onto cloud is not an easy process and requires planning. The proposed work resolves the limitations of static resource allocation policy which reserves the provisioned resources to an instance irrespective of its utilization status. Though such an approach may ensure high availability, the scope for minimizing wastage of resources exists. To overcome this limitation, a solution for optimised resource utilization based on the application's requirements is proposed where the provisioning of resources is based on forecasted application dependency.

Keywords— Cloud Computing; PaaS; Application deployment; Predictive allocation; Resource Management

I. INTRODUCTION

Cloud computing encompasses virtualization and other associated techniques so as to provide computing as a utility model which facilitates scalability and on-demand serviceability to name a few. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The NIST definition of cloud computing describes five three service models and four deployment models.

The cloud service models are generalised to three categories namely: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

Software as a Service model provides cloud user, a ready to use application delivered over the Internet. The software licence may also limit the number of users and/or devices where the software can be deployed. Software as a Service users, however, subscribe to the software rather than purchase it, usually on a monthly basis. Applications are purchased and used online with files saved in the cloud rather than on individual computers.

Platform as a service is a cloud computing model that delivers applications over the Internet. In a PaaS model, a cloud provider provides hardware and software tools usually those needed for application development to its users as a service. A PaaS provider hosts the hardware and software on its own organization. PaaS frees users from having to install in-house hardware and software to develop or run a new application. An appropriate example of PaaS is Google AppEngine, App Engine will scale your application automatically in response to the amount of traffic it receives so you only pay for the resources you use. Just upload your code and Google will manage your app's availability. There are no servers for you to provision or maintain.

Infrastructure as a Service is a Provisioning processing, storage, networks, and other fundamental computing resources means the consumer of those resources does not manage or control the underlying cloud physical infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components. The cloud deployment models are generalised to four categories namely: Private Cloud, Public Cloud, Community Cloud, Hybrid Cloud.

Private Clouds – For exclusive use by a single organization and typically controlled, managed and hosted in private data centers. The hosting and operation of private clouds may also be outsourced to a third party service provider, but a private cloud remains for the exclusive use of one organization.

Public Clouds – For use by multiple organizations (tenants) on a shared basis and hosted and managed by a third party service provider.

Community Clouds – For use by a group of related organizations who wish to make use of a common cloud computing environment. For example, a community might consist of the different branches of the military, all the universities in a given region, or all the suppliers to a large manufacturer.

Hybrid Clouds – When a single organization adopts both private and public clouds for a single application in order to take advantage of the benefits of both. For example, in a cloud bursting scenario, an organization might run the steady-state workload of an application on a private cloud, but when a spike in workload occurs, such as at the end of the financial quarter or during the holiday season, they can burst out to use computing capacity from a public cloud, then return those resources to the public pool when they are no longer needed.

II. LITERATURE SURVEY

2.1 CRESS: A Platform of Infrastructure Resource Sharing for Educational Cloud Computing

A platform of sharing infrastructure resources in the virtual Cloud computing lab was developed. The system had good expandability and can improve resource sharing and utilization. There are still few research issues needed further consideration: providing performance evaluation against other platforms such as Amazon EC2; developing more scheduling strategies and algorithms, providing performance indices for different scheduling algorithms, conducting load and pressure tests in distributed environment etc. CRESS (Cloud Resource Sharing System) currently relies on the underlying virtualization platform such as VMWare to do migration when VM failures and network interruptions happen.

All current available cloud computing platforms are either proprietary or their software infrastructure is invisible to the research community except for a few open-source platforms. A platform of infrastructure resource sharing system (Platform as a Service (PaaS)) is developed in virtual Cloud computing environment. Its architecture, core modules, main functions, design and operational environment and applications are good expandability and can improve resource sharing and utilization and is applied to regular computer science teaching and research process.

Major contributions of CRESS:

1) Software and hardware platforms provided from real and virtualized servers;

2) Resource management node;

3) Database servers and users who access resources through Internet or Intranet.

Cloud computing makes elastic computing (scale out or in) possible. Students/researchers/researchers can deploy Web service on CRESS depends on the traffic load to increase or decrease VMs. open source Cloud computing platforms need medium or large size computing resource but students/researchers normally cannot access. However, through virtualization, CRESS provides these resources for students/researchers to remotely access seven days a week. A platform of infrastructure resource sharing system (CRESS) is presented to improve resource sharing and utilization in virtual Cloud computing environment.

2.2 A Method for Solving Performance Isolation Problem in PaaS Based on Forecast and Dynamic Programming

An application may not use resources in upper bound all the time which is set in SLA. So one of the basic idea is to allocate more resources to virtual machines (VM) in a real machine host and total of these resources in all VMs are more than the real machine host has. To ensure a service provider won’t breach SLA, one of VMs performance may not be influenced. This problem is called Performance Isolation problem. Proposed solution to this problem, such kind of VMs to other hosts which have more idle resource. As moving a VM to another host may cost lots of time, VM workload may increase fast in advance. This forecast needs not to be most accurate because the increasing risk or trend of VMs resources demands. This strategy to move VMs is another problem. If move a VM with high workload to a host which has fewer resources, the performance isolation problem still exists. If the VM to a host which has lots of resources with none other VMs, it’s very wasteful, design and implement the method of solving performance isolation problem in PaaS. Curve-fitting forecast method is used to calculate resource usage of VMs in advance and determine which VMs need to migrate. Dynamic programming method is used to decide a best strategy of VM migrating. Curve-fitting forecast and Dynamic Programming methods solved the performance isolation problem in PaaS is right.

2.3 Model Driven provisioning in multi-tenant clouds

The crucial issue is provisioning appropriate resources for multi-tenant cloud environments. Traditional provisioning techniques are not sufficient to solve this problem, as provisioned resources undergo dynamic changes, which can affect tenancy SLO. Presented an algorithm for grading and monitoring resource health, thereby determining affected provisioned resource. This approach is based on mapping of functional and non-functional tenancy requirements with appropriate resources, their parameters, and health monitoring policy. An approach via a proof-of-concept prototype, using a realistic example of cloud resources and applications. A novel approach towards dynamic provisioning and define a cloud provisioning system based on what define as Health Grading Model (HGM) and Tenancy Requirement Model (TRM), and then provide algorithms to dynamically provision the resources based on tenancy matching to health metrics based on HGM, and underpinned by tenancy requirements as specified in TRM.

The HGM helps in defining, quantifying and monitoring the health of a resource, along with critical levels for each of the monitored parameters. Similarly, TRM helps in defining and quantifying the requirements of a specific tenancy. Dynamic resource provisioning algorithm ensures that the set of provisioned resources are able to meet the changing requirements of tenancy, without affecting the given tenancy SLOs, using the best available set of healthy resources. The health of a resource is constituted by a set of critical parameters. Any change in the value of these parameters affect the functioning of that resource, in turn affecting the SLO of the tenancy provisioned on that resource. It may be possible to augment a resource to increase certain of its parameter values, either by tuning or by physically choosing another similar resource with enhanced capability, so as to increase its health. Likewise, any decrease in such a parameter value would result into deterioration of the resource health, and have a negative impact on the SLO of the tenancy provisioned. For example, health of a Server System, Could be best attributed to the number of CPU cores and CPU utilization over a period of time. Any variations in these values will have an impact on the SLO of provisioned tenancy.

Proposed key contributions are:

1) Enable tenancy requirements to be captured and represented in a formalized model both for future reuse considerations and also for continuous conformance of provisioning
2) Enable certification and guarantee of accurate resource provisioning based on TRM and HGM

3) Flexible switching between health monitoring for provisioned resources and health monitoring for the candidate resource pool, thus supporting both published catalogue based tenancy provisioning and made-to-order tenancy provisioning

4) Continuous monitoring and replacement of provisioned systems based on changing health grading

5) Providing the best-fit resource for current tenancy requirements and thus maximizing resource utilization and reducing cost of hosting

2.4 Automatic Generation of Platforms in Cloud Computing

Environment for Automatic Generation of Platforms in Cloud Computing using a model-based approach. It discusses the challenges pertaining to building these systems, as well as the adoption of SLA as a mechanism to regulate the use of resources. This proposal differs from other existing services, by merging the negotiation of SLA guarantees, during the procurement of services process, and how it serve as a basis for resource allocation and for verification and assurance of this guarantees. Using this solution has great advantages because it proposes an automated process of creating virtual machines, consider aspects related to both the hardware and software, a fact that contributes to the increased use of cloud computing for users who do not have much knowledge of the cloud functioning, and actually making its use easier even for in experienced users, saving time with the automatic creation of a platform.

From the point of view of the user, the following contributions regarding automated software development, in the context of cloud computing, can be highlighted:

• The user does not need to have knowledge of how the cloud environment works, or even aspects related to maintenance of the infrastructure.

• The work platform is specified with a simple selection of parameters, without the need of installations and configurations of robust hardware in the local environment of the user.

• The time to release the application or work platform is significantly reduced, since the models for each feature will be available and pre-tested making the whole process virtually automatic, except when there is the need for negotiation of specific aspects between the parties.

• The user can make a clear and objective forecasting of costs related to the application or platform in order to accurately assess the cost / benefit of hiring services.

A major challenge in the environment of clouds is efficiently control the use of computational resources, and make sure that he was hired by clients is actually being

Offered, so the use of SLA, can clearly define the responsibility of each part, and define the requirements that are being hired, to be able to check the contracts and resource efficiency. The negotiation of SLA guarantees during the procurement of services process and how it serve as a basis for resource allocation and for verification and assurance of this guarantees. Automatic generation of Platforms is to combine these paradigms, cloud, models and SLA, in order to use the resources of a computational cloud to provide automatically a platform as a service - PaaS. It will be developed in a web environment, using a model-driven approach to create the platforms and the management of resources and quality will be made by SLA.

2.5 Design Support for Performance Aware Dynamic Application (Re-) Distribution in the Cloud

The need to combine both top-down and bottom-up application workload analysis approaches in order to proactively enable the (re-)distribution of the application componenets to cope with fluctuating resources demands. The first rounds of experiments positioned this work on the application database layer, and used the TPC-H benchmark as the basis. The TPC-H workload according to its computational demands. This characterization was used as the basis to generate representative workloads with different behavioural characteristics, which emulated the business logic of an application. Evaluated different deployment approaches of the application database and analyzed the perceived performance and its variation on a daily basis. There is a dependency between the workload distribution, the concrete distribution of the application components, and the performance variability observed in virtualized environments. Such a performance variation mostly increases in off-premise virtualized environments.

The need for an application distribution process which can be used to enable the application (re-)distribution based on a dynamic analysis of the workload and the resources demands evolution. The process introduced the concept of the Collaborative Loop as an approach to assist the application developer in efficiently selecting the distribution of the application components and the selection of cloud services to cope with the evolution of the application performance demands.

III. CONCLUSION

Cloud frees the user from the job of setting up and maintaining the computational infrastructure and helps him to focus on developing and perfecting his application. Also the cloud provides the benefit of scaling (manual/real-time) so that the application continues to work even under heavy load. However moving onto cloud is not an easy process and requires planning. The proposed work resolves the limitations of static resource allocation policy which reserves the provisioned resources to an instance irrespective of its utilization status. Though such an approach may ensure high availability, the scope for minimizing wastage of resources exists. The objective function and the predictive engine along with the network monitoring allocates the resources based on the user requests. The application dependency for the user request is predicted by the predic-tive engine that uses the PMML. The objective function significantly creates an improvement in the performance of the cloud network. Thus, the resource are allocated in such a way that the allocation is network aware and reduces infrastructure cost, latency and gains performance enhancement.

REFERENCES

[1] TIAN Wenhong, SUN Xiashuang, JIANG Yaqiu, WANG Haoyan "CRESS: A Platform of Infrastructure Resource Sharing for Educational Cloud Computing" China Communications, 43-52, 2013.
[2] Jiayang Yu, Ruonan Rao. "A Method for Solving Performance Isolation Problem in PaaS Based on Forecast and Dynamic Programming"ICCIS, 947-950, 2012.
[3] Atul Gohad, Karthikeyan Ponnalagu, Nanjangud C. Narendra. "Model Driven provisioning in multi-tenant clouds"SRIIGC, 11-20, 2012.
[4] Helder Pereira Borges, Bruno Schulze "Automatic Generation of Platforms in Cloud Computing"IEEE, 1311-1318, 2012.
[5] Santiago Gomez saez, Vasilios Andrikopoulos, Frank Leymann, Steve Strauch. "Design Support for Performance Aware Dynamic Application (Re-) Distribution in the Cloud"IEEE, 225-239, 2015.