Resource Monitoring and Provisioning in SaaS in Public Cloud

Syed Manzoor Ali & Sandeeep Sahu

P.G Department of computer science SRIT jabalpur
E-mail : manzoor2001in@gmail.com, sandeep.sahu12@gmail.com
SRIT , Jabalpur ,MP, India.

Abstract - **Load balancing of non-preemptive independent tasks on virtual machines (VMs) is an important aspect of task scheduling in clouds. Whenever certain VMs are overloaded and remaining VMs are under loaded with tasks for processing, the load has to be balanced to achieve optimal machine utilization. In this paper, we propose an algorithm named honey bee behavior inspired load balancing (HBB-LB), which aims to achieve well balanced load across virtual machines for maximizing the throughput. The proposed algorithm also balances the priorities of tasks on the machines in such a way that the amount of waiting time of the tasks in the queue is minimal. We have compared the proposed algorithm with existing load balancing and scheduling algorithms. The experimental results show that the algorithm is effective when compared with existing algorithms. Our approach illustrates that there is a significant improvement in average execution time and reduction in waiting time of tasks on queue.**

Keywords: Agent,Codes , Load, ConvergentCoherence , HBB-LB

I. INTRODUCTION

Cloud computing is a paradigm that focuses on sharing data and computations over a scalable network of nodes [1, 2]. Examples of such nodes include end user computers, data centers, and web services. Such a scalable network of nodes is called cloud. An application based on such clouds is taken as a cloud application.

Cloud computing is a model for enabling ubiquitous ,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 (Mell and Grance,2011). Deployments of cloud computing are expected to be a $160 billion market in 2012 (Crossman,2009) in a growth of 25% of all incremental investment in technology in 2012 – a growth indicated to be the largest since the Internet (Hamm, 2009).

Platform as a Service (PaaS) is one of the service model of cloud computing. The capability provided through PaaS to the consumer is to deploy onto the cloud infrastructure consumer- created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network , servers ,operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting Cloud Computing and PaaS in Information Systems education

Cloud computing is clearly considered to be developing as an enabling model for improving the processes of firms. Schools of computer science and information systems may benefit by having students cognizant of the cloud computing technologies and strategies (Lawler,2010 and 2011). In effect, several studies have pioneered in this front.

Among these studies, one describes an introductory lesson that builds on student familiarity with Google Docs to illustrate the concepts of Infrastructure, Software, Data, and Platform as a Service (Frydenberg 2011). The study found that while many first-year college students have heard of the term Cloud Computing and used Google Docs, a popular web-based office suite of applications for collaboration, their knowledge of how the Cloud is used in a business environment is limited (Frydenberg, 2011).

Hollingsworth and Powell (2010) introduce Cloud Computing topics in a Web programming course where students write and configure applications deployed to Google's Cloud (App Engine). Malan (2010) created a network of virtual machines hosted on Amazon's Elastic Cloud platform for students to explore cloud concepts and build applications in an introductory computer science course. Rehman and Sakr (2010) chose Microsoft's Azure platform to design an undergraduate-level course in Cloud Computing where students learned about public and private clouds, virtualization, MapReduce and Hadoop parallel processing technologies, and other cloud technology concepts. Each of these courses requires programming knowledge in order to complete the projects (Frydenberg, 2011).

II. RESEARCH PURPOSE

The cloud has become a widely used term in academia and the industry. Education has not remained unaware of this trend, and several educational solutions based on cloud technologies are already in place, especially for software as a service cloud. However, despite some of the early research efforts, an evaluation Of the educational potential of infrastructure and platform clouds has not been explored yet (Vaquero, 2011).

In particular, Platform-as-a-Service (PaaS) has gained great popularity among developers in recent years. It has the potential to significantly reduce IT administration and management complexities associated with developing and deploying web applications.

To incorporate PaaS in the IS curriculum, it is important to understand the strengths, weaknesses, and constraints of the platform,especially security,compliance and perhaps legal constraints (Roggio, 2011).

As suggested by extant research, it is not recommended to develop the platform in house. Establishing a set of development tools (solution stack) in a platform can be very time consuming, and keeping these plug-ins up to date is daunting administrative work. Administrators of such a platform require a significant competence in a variety of skills (Roggio, 2011).

Therefore, our research focuses on examining the commercial PaaS providers and their potential to be utilized in IS education. Traditionally, PaaS providers have focused on Ruby, Python and PHP platforms. The Java platform has so far been under-served by PaaS providers due to the complexity of managing and scaling Java application servers that can be very resource hungry. On the other hand, the Java platform is widely used in most Information System Education curricula. For IS educators, finding the right resource to teach clouding computing on Java platform could be challenging.

Since 2010, several companies have tried to tackle the Java PaaS problem, and among them, CloudBees has emerged as an early leader of this front, as noted in several independent comparison studies such as InfoQ's "A Java Developer's Guide to PaaS" (Yuan, 2011).

In this research, we will discuss key concepts behind Java PaaS and the key features of CloudBees. We argue that as a leader in the Java Platform-as-a-Service (PaaS) sector, CloudBees provides Information System educators a standard-based,integrated, and scalable platform for students who want to develop and deploy web applications in a cloud environment without administration headaches.

III. ADVANTAGES OF CLOUDBEES

A standard-based Java platform

The Java platform has a strong tradition of standardization – both in the programming language itself as well as in the libraries for the platform (aka "write once, run anywhere"). In the world of enterprise Java, much of the innovation in the platform happens in application frameworks and application servers. The ability to use any framework on a variety of different application servers is a core requirement for a Java PaaS.

In the past, we have seen Java PaaS efforts that are based on non-standard technology. A very good example is Google App Engine for Java (GAE). GAE provides a Java runtime that limits access to file system, networking, threading and other Application Programming Interface (API) methods. It provides a non-relational database, and supports a limited set of outdated Java Data Object (JDO) API on top of it. While Google provided great technical rationale on why they did this, the non-standard approach created many problems and gotchas for Java developers. Fast forward to 2011, when Google decided to move GAE out of beta and dramatically raised the price, many GAE developers found themselves stuck in that PaaS environment not being able to move their application and data elsewhere because their applications were written specifically to GAE's non-standard Java runtime platform.

By comparison, CloudBees strongly believes in the benefit of standardization and portability of the Java platform.As a result,the CloudBees PaaS environment supports plain old Tomcat,Java Platform Enterprise Edition (Java EE) certified JBoss 6 Application Server (JBoss 6 AS),and MySQL database.You can deploy just any standard-based Java web application to CloudBees, and easily take your CloudBees application to any other PaaS or in-house providers of Tomcat /Java EE application servers.

With the CloudBees Software Development Kit (SDK), you can easily test and validate your application on their local computer inside an Eclipse IDE. But, because of the support for standard-based technology, you do not need the SDK or Eclipse to be a

CloudBees developer. You can use your favourite tools any IDE or even text editors such as EMACS or VIM.

Since CloudBees deploys both the application server and the database for you,it provides a tightly integrated experience just as an in-house IT team might provide for you. The database is simply available as a Java Naming and Directory Interface (JNDI) resource as soon as itis deployed – there is no need to chase down Java Database

IV. RESEARCH PURPOSE

The cloud has become a widely used term in academia and the industry. Education has not remained unaware of this trend, and several educational solutions based on cloud technologies are already in place, especially for software as a service cloud. However, despite some of the early research efforts, an evaluation Of the educational potential of infrastructure and platform clouds has not been explored yet (Vaquero, 2011).

In particular, Platform-as-a-Service (PaaS) has gained great popularity among developers in recent years. It has the potential to significantly reduce IT administration and management complexities associated with developing and deploying web applications.

To incorporate PaaS in the IS curriculum, it is important to understand the strengths, weaknesses and constraints of the platform, especially security, compliance, and perhaps legal constraints (Roggio, 2011).

As suggested by extant research, it is not recommended to develop the platform in house. Establishing a set of development tools (solution stack) in a platform can be very time consuming, and keeping these plug-ins up to date daunting administrative work. Administrators of such a platform require a significant competence in a variety of skills (Roggio, 2011).

Therefore, our research focuses on examining the commercial PaaS providers and their potential to be utilized in IS education. Traditionally, PaaS providers have focused on Ruby, Python and PHP platforms. The Java platform has so far been under-served by PaaS providers due to the complexity of managing and scaling Java application servers that can be very resource hungry. On the other hand, the Java platform is widely used in most Information System Education curricula. For IS educators, finding the right resource to teach clouding computing on a Java platform could be challenging.

Since 2010, several companies have tried to tackle the Java PaaS problem, and among them, CloudBees has emerged as an early leader of this front, as noted in several independent comparison studies such as InfoQ's "A Java Developer's Guide to PaaS" (Yuan, 2011).

In this research, we will discuss key concepts behind Java PaaS and the key features of CloudBees. We argue that as a leader in the Java Platform-as-a-Service (PaaS)sector, CloudBees provides Information System educators a standard-based, integrated, and scalable platform for students who want to develop and deploy web applications in a cloud environment without administration headaches.

V. ADVANTAGES OF CLOUDBEES

A standard-based Java platform

The Java platform has a strong tradition of standardization – both in the programming language itself as well as in the libraries for the platform (aka "write once, run anywhere"). In the world of enterprise Java, much of the innovation in the platform happens in application frameworks and application servers. The ability to use any framework on a variety of different application servers is a core requirement for a Java PaaS.

In the past, we have seen Java PaaS efforts that are based on non-standard technology. A very good example is Google App Engine for Java (GAE). GAE provides a Java runtime that limits access to file system, networking, threading and other Application Programming Interface (API) methods. It provides a non-relational database, and supports a limited set of outdated Java Data Object (JDO) API on top of it. While Google provided great technical rationale on why they did this, the non-standard approach created many problems and gotchas for Java developers. Fast forward to 2011, when Google decided to move GAE out of beta and dramatically raised the price,many GAE developers found themselves stuck in that PaaS environment not being able to move their application and data elsewhere because their applications were written specifically to GAE's non-standard Java runtime platform.

By comparison, CloudBees strongly believes in the benefit of standardization and portability of the Java platform. As a result, the CloudBees PaaS environment supports plain old Tomcat, Java Platform Enterprise Edition (Java EE) certified JBoss 6 Application Server (JBoss 6 AS), and MySQL database.You can deploy just any standard-based Java web application to CloudBees, and easily take your CloudBees application to any other PaaS or in-house providers of Tomcat / Java EE application servers.

With the CloudBees Software Development Kit (SDK), you can easily test and validate your application on their local computer inside an Eclipse IDE. But, because of the support for standard-based technology, you do not need the SDK or Eclipse to be a CloudBees developer. You can use your favorite tools – any IDE or even text editors such as EMACS or VIM.

Since CloudBees deploys both the application server and the database for you, it provides a tightly integrated experience just as an in-house IT team might provide for you. The database is simply available as a Java Naming and Directory Interface (JNDI) resource as soon as it is deployed – there is no need to chase down Java Database Connectivity(JDBC) drivers, connection strings, and configure JNDI.The same goes for clustering. The CloudBees PaaS deploys application servers running your application behind a load balancer. Of course, your application does need to be "cluster aware" by saving HTTP sessions in a database or a shared in-memory cache, but all the hard work of managing the cluster and load balancer has been take care of for you.

A) No vender lock-in

The CloudBees PaaS supports Java standards like servlets, Java EE and others, and is based on industry standard application servers such as Tomcat and JBoss. As such, the CloudBees PaaS does not require you to use proprietary APIs or other non-standard Java libraries, ensuring that your Java code remains portable. The CloudBees PaaS simply runs your Java code. If you are unhappy with CloudBees for any reason, you can take your code and run it elsewhere.

B) Easy Continuous Integration

While all other PaaS providers focus on providing a hosted runtime environment for applications, CloudBees take the "platform" concept further to support the entire development, testing, and deployment life cycle of Java applications. Specifically, CloudBees supports private source control repositories, private Maven repositories for team's internal use,and on-demand continuous build servers managed by Jenkins.

Maven is a widely used build tool for managing dependencies in a Java project. A typical enterprise Java project has hundreds of external dependencies for libraries and frameworks. The exact version of each of those dependency Java Archive (JARs) must be carefully managed to make sure that they all work together in a particular version of the application server. Maven helps us automatically and declaratively manage those dependencies via configuration files.

While Maven is very useful, not all JARs are available in public Maven repositories. That is especially true for internal library JARs you might build. Without internal Maven repositories, you would fall back to manually copying JARs and putting those JARs into their individual computer's local Maven repositories – a process that is not only error prone, but forgoes much of the benefits Maven brings. CloudBees provides you with private Maven repositories for both development and production use. It is a great feature to take full advantage of Maven.

Perhaps even more significantly, CloudBees is the first company to provide "Jenkins in the cloud" services to support continuous integration processes required by many Agile development teams. The concept of continuous integration is that software products needs to be built and validated against an array of tests every time a developer checks code into the source control repository, to make sure that the software is always in the "ready to be released" state. If the build fails or any test fails, appropriate members of the team need to be notified. Jenkins is a very popular tool to manage the continuous build and testing process.

However, the challenge of using Jenkins in house is relatively high infrastructure and IT administration overheads. You have to run Jenkins on an in-house server or a cluster of servers, and have an IT administrator to manage it. That barrier often prevents continuous integration. CloudBees provides an on-demand Jenkins service that spawns a new Jenkins build server instance every time the code changes in the source control (or at fixed times during the day). The build server runs the Maven script against your private Maven repository to build the software application, and then automatically runs all the unit tests and end-user tests via a script. CloudBees then reports the results back to you via a dashboard and notifications.

If you have always wanted to apply Agile methods in your curriculum to build more reliable software faster, but did not have time / resources to set up the required technical infrastructure, CloudBees could be the ideal choice.

VI. PROPOSED METHOD:

Provisions of service and resources in cloud PaaS is an important function that provides analytical statistics about the current view of cloud (running instance for a user or group of users).

Author of [1] digging on it and found that, An Amazon EC2 instance [2] is a virtual processing resource (VM) in the Amazon cloud. The process of instantiating new VMs could take as long as few minutes. The new VMs originate either as fresh boots or replicas of a template VM, unaware of the current application state.

A set of instances is monitored by a web service called CloudWatch [3], and are automatically scaled in or out by Auto Scaling [4] according to user-defined conditions. For example, it is possible to set a condition to add new instances when the average CPU utilization of the current VMs exceeds 70%; and similarly, remove instances in the same increments when the CPU utilization falls below 10%. It takes an action based on metrics exposed by CloudWatch contrary to our work. These metrics take a purely system view such as utilization, but not the application view such as average response time of a request, or an associated Service Level Objective (SLO). Further, Auto Scaling is agnostic to the need for provisioning data resources needed for workload execution.

Amazon claims that the latency and throughput of the volumes are designed to be significantly better than the instance's local store. However, a volume can only be attached to only one instance.

Microsoft Windows Azure does not offer automatic scaling, but it is the primary tool for provisioning [5]. Users can provision any number of instances that they wish to have available for their application. Like Amazon EC2, the instances are virtual processing resources. Effectively, Azure provides provisioning mechanisms which can be used by a management function to improve application and system metrics.

Systems that jointly employ scheduling and provisioning techniques have been explored in grids. The Falkon [6] scheduler triggers a provisioner component for host increase or decrease. This host variation has also been explored during the execution of a workload, hence providing dynamic provisioning.

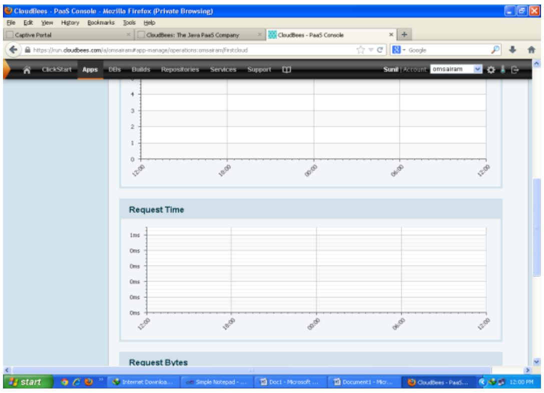
The biggest problem raised during research in the field of cloud computing is to deploy the cloud IaaS and PaaS. There are many alternatives to like amazon, windows azure etc that provides users to deploy the cloud and validate their area of interest on applying them. But they charge for that. That means user has to pay for using their services. Cost is not more worthy if the deployable application services fulfill the organization needs but for a individual it is costly to pay per month.

Second approach is to configure and deploy private cloud again the cost of the required resources is to higher for an individual or a group (research group). Therefore the biggest challenge to evaluate our proposed method is the IaaS and PaaS of the cloud which has been solved by using ClouBee infrastructure that provides public PaaS service to the registered user without paying any thing.

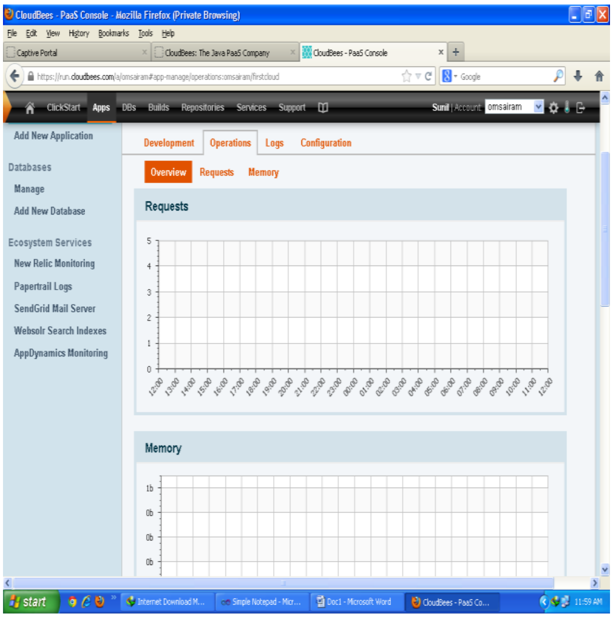
Now the second and most important point is to provisioned of resources and application on the cloud, to achieve this we have deployed the New Relics tool on our cloudBee for better evaluation of the proposed work for each request with QoS constraints:

```
resources←available_resources for the requested application;
  Jobs_pending←number of jobs in the queue;
  effort ← (Jobs_pending /resources)× averageJobsRuntime;
   if  effort > Remaining_Time_application
        then
 additionalResources ←(Jobs_pending×averageJobsRuntime) /
Remaining_Time_application;
 CALL_New_Releics(job_Id);  // for resource provisioning
      else
          toRelease ← 0;
            if Jobs_pending < resources then
             toRelease ← Jobs_pending − resources;
  end   else
       Jobs_pending ← Jobs_pending + Jobs_running;
  effort←(Jobs_pending /resource× averageJobsRuntime;
      if( effort < Remaining_Time_application) then
  toRelease←resource–(Jobs_pendin×averageJobsRuntime)   /
Remaining_Time_application;
          end
                 end
  CALL_New_Relics_Relaese_resources(job_Id);
                    End  end
```

(a) RequestTime



(b) Memory usage



VII. CONCLUSION

The cloud computing system can not only work in Intranet but also work in Internet. For the unstable bandwidth and long transfer delay of Internet, the mobile agent is a better underlying facility to implement the software and data migration in wide area cloud computing system. In this paper, The Mobile Agent based Service for Cloud Computing in Internet Environment (SaaAS) is presented. The code and data of service load mechanism based mobile agent and divided- cloud and convergent coherence mechanism of SaaAS are also discussion in this paper.

By using the agent rather than RPC/RMI as the underlying facility to implement the software and data migration, SaaAS is more suitable to the cloud computing system which works in Internet.

In conclusion, the application of mobile agent enables SaaAS to have good flexibility, adaptability and usability and to be more suitable to work in Internet than conventional cloud computing system.

VIII. REFERENCES

[1]. Rizwan Mian, Patrick Martin and Jose Luis Vazquez-Poletti “Provisioning data analytic workloads in a cloud”, Elsevier, Future Generation Computer Systems, 2012.

[2]. Amazon, Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/.

[3]. Amazon,CloudWatch.http://aws.amazon.com/cloudwatch/.

[4]. Amazon,AutoScaling.http://aws.amazon.com/autoscaling/.

[5]. D. Chappell, Introducing Windows Azure, David Chappell & Associates, 2009.

[6]. I. Raicu, Y.Zhao, C. Dumitrescu, I. Foster, M. Wilde, Falkon: a fast and light-weight tasK executiON framework, in: Proceedings of the 2007 ACM/IEEE Conference on Supercomputing, ACM, Reno, Nevada, 2007.

◈◈◈