

Task Scheduling Algorithm based on Resources Segregation in Cloud Environment

Prof. Shailendra Raghuvanshi, Priyanka Dubey

Takshila Institute of Engineering and Technology, Jabalpur, Madhya Pradesh, 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, 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 using workflowsim simulator in JAVA.

Keywords : Workflowsim Simulator, Java, Enhanced Max-Min, Load; Convergent Coherence

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, 2017). 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 environment (Mell and Grance, 2017).

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)

For PaaS, it is probably the most abstract cloud concept for students without programming background to understand. It is also the least accessible concept because most consumer cloud applications do not rely on PaaS (Frydenberg, 2011). One study proposed teaching PaaS in a capstone course. It compares two different options, a

commercial platform or developing a private platform with technical support locally (Roggio, 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 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.

III. ADVANTAGES OF PUBLIC CLOUD

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 taken care of for you.

No vendor 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.

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 need 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.

Proposed Algorithm

Following represents proposed algorithm :

Input: T is task set, R is Resource set with MIPS speed, and N is total no of Resources

Algorithm Proposed (Task, Resource, N)

```

{
For ti in Task do
    {
Sort resource set in ascending order of execution
speed in MIPS.
    }
SG= Resource/2, resources having MIPS less than 500.
FG= Resource/2, resources having MIPS greater than
500.
While ti ∈ TS and TS is not empty do
    {
Find resource (Ri) with minimum execution time
If Ri ∈ SG than
Assign average length task to Ri
Else
Assign maximum length task to Ri
Delete ti from T
    }
}
    
```

By using the above algorithm, we are trying to schedule the task properly. The proposed algorithm will minimize the wastage of resource and also will minimize the execution time of the task on resources. In the proposed algorithm the Cloud providers has the ability to assign mips randomly while creating the virtual machines in cloud environments.

The proposed solution is given below and it shows working of proposed method step by step.

Step-1 Arrange N resources in their ascending order of MIPS.

Step-2 Classify resources in two sets:

If (MIPS <500) go to SG.

Else

Go to FG.

Step-3 Select a task from task set.

Step-4 Find available resource with minimum execution time.

Step-5 If resource belongs to slow set of resources than select average length task otherwise select maximum length task.

Step-6 Assign task to resource.

Step-7 Delete assign task from task set.

Step-8 Repeat steps from 3 to 7 until no more tasks left in task set.

IV. RESULTS AND DISCUSSION

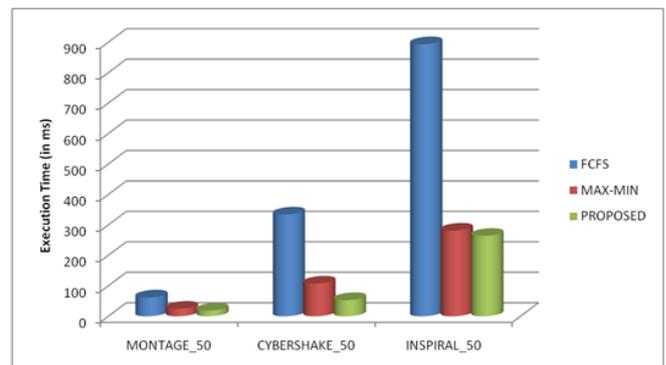


Fig.4.1 : Make span Comparison of Scheduling Algorithms for VM=20.

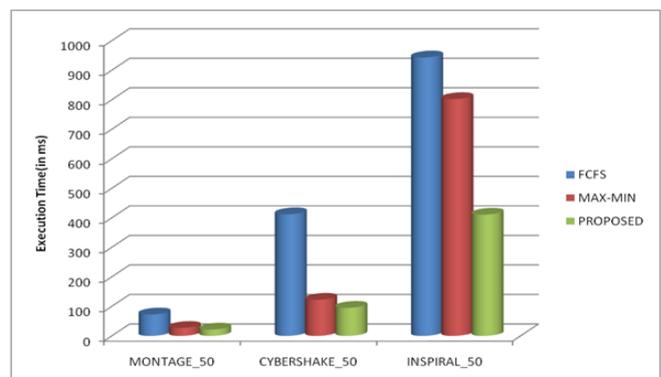


Fig.4.2 : Makespan Comparison of Scheduling Algorithms for VM=30.

V. CONCLUSION

Cloud computing is a huge source of computing resources, storages, etc., that are provided through the network platform based on the user's requests. The main focus of task scheduling is the cloud data centres and allocates the resources available in the cloud so that the total time of tasks execution (make span) is minimized. In case of having many resources and receiving many requests for these resources, it is difficult to process the requests and task scheduling process for resources allocations. Therefore, the tasks have to be scheduled through an appropriate framework for resources allocation to the users' requests.

In this thesis, using two traditional Min-Min and Max-Min algorithms, an effective scheduling algorithm was proposed to reduce the wait time and Make span for tasks in the cloud computing environment. After selecting the priority for all tasks by the corresponding algorithm, it selects the first jobs; and according to the number of tasks whose priority is more or less than the mean number, it selects one of the two traditional algorithms Min-Min or Max-Min.

The results of the experiments indicate that the proposed algorithm is superior to other algorithms. Also, given that the overhead of the two traditional algorithms mentioned is low and the proposed algorithm uses the advantages of the two algorithms mentioned above, the proposed algorithm overhead is too low.

VI. REFERENCES

- [1]. Global Cloud Computing Research Test Bed Wiki [URL]. <http://cloudtestbed.org/>.
- [2]. IBM Blue Cloud project [URL]. <http://www-03.ibm.com/press/us/en/pressrelease/22613.wss/>.
- [3]. Lijun Mei, W.K. Chan, and T.H. Tse, "A tale of clouds: paradigm comparisons and some thoughts on research issues", in Proceedings of 2017 IEEE Asia-Pacific Services Computing Conference (APSCC 2017), pp.464-469.
- [4]. Cloud computing. Wikipedia. Available at http://en.wikipedia.org/wiki/Cloud_computing.
- [5]. K Hartig. What is cloud computing? SOA World Magazine. Available at <http://soa.sys-con.com/read/579826.htm>.
- [6]. R Wang, T. Anderson and M. Dalin, "Experience with a distributed file system implementation", Technical report, University of California, Berkeley, Computer Science Division, June 2017.
- [7]. L Ismail and D. Hagimont, "A Performance Evaluation of the Mobile Agent Paradigm", ACM SIGPLAN Notices, October 2018, 34(10), pp.306-313.
- [8]. PS. Narayanan. From grid computing to cloud computing: the IBM approach. Garuda Partner Meet, Bangalore, India, March 4, 2018.
- [9]. Web services business process execution language version 2.0. Available at <http://docs.oasis-open.org/wsBPEL/2.0/wsBPEL-v2.0.html>.
- [10]. K. Hartig. What is cloud computing? SOA World Magazine. Available at <http://soa.sys-con.com/read/579826.htm>.
- [11]. C. Lee, S. Ko, S. Lee, W. Lee, and S. Helal. Context-aware service composition for mobile network environments. In Ubiquitous Intelligence and Computing, volume 4611 of Lecture Notes in Computer Science, pages 941–952. Springer, Berlin, Germany, 2017.
- [12]. L. Ismail and D. Hagimont, "A Performance Evaluation of the Mobile Agent Paradigm", ACM SIGPLAN Notices, October 2016, 34(10), pp.306-313.