International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.2, 2015

RESEARCH ARTICLE

ISSN: 2321-7758

NODE SELECTION AND RESOURCE PREDICTION IN DISTRIBUTED COMPUTING PLATFORMS

KAVITHA RANI.T[1], C.VIJAYALAKSHMI[2], K.SANGEETHA[3]

[1,] Student, [2,3]Faculty of CSE

Department of Computer Science and Engineering,

Panimalar Engineering College, Chennai.

Article Received: 11/03/2015 | Article Revised on; 15 /03/2015 | Article Accepted on:19/03/2015

KAVITHA RANI.T

ABSTRACT

To develops techniques for selecting nodes and scheduling applications in large-scale, cooperatively pooled shared computing platforms. In such platforms, resources at a node are allocated to competing users on fair-share basis, without any reserved resource capacities for any user. There is no resource manager; the users independently select nodes for their applications. Our project is focused on developing node selection techniques; we first study the resource utilization characteristics of Planet Lab nodes. Our approach uses the notion of eligibility period, which represents a contiguous duration for which a node satisfies a given resource requirement. Based on this study we develop models for identifying nodes that are likely to satisfy a given requirement for long durations. We also develop an online model for predicting the idle resource capacity that is likely to be available on a node over a short term. We evaluate and demonstrate the performance benefits of the node selection techniques and the prediction model.

I.INTRODUCTION

Cooperatively pooled resources in large scale platforms are generally spread over a wide area and loosely managed by the participating organizations. Platforms such as Planet Lab do not utilize any centralized resource management and scheduling mechanisms, thereby putting the responsibility of node selection for application deployment and scheduling on the users. Different user applications may be co-hosted on a node and they compete for the resources available on that node. Such platforms allocate resources to competing applications on fair-share basis and do not provision guaranteed levels of resource capacities to an application. The focus of our work is to observe the resource availability characteristics of nodes in platforms. Based on this observation, we would develop an intelligent method to select nodes on such environment. In order to select nodes those are likely to satisfy the given requirement for a long time. The requirements of a task could be stated in terms of CPU capacity, memory, and network bandwidth. We also observe ineligibility period that might very less time within 60 sec, it can tolerate before take any remedial actions such as relocating and terminate. After getting eligible the application runs normal. We also develop idle resource predicting model that might useful for replicated services or any distributed application. We evaluate and demonstrate the benefits of our new node selection technique and prediction model.

2. Existing Work

In an existing system, cooperatively pooled resources in Platforms such as Planet Lab do
not utilize any centralized resource management and scheduling mechanisms, Different user applications may be co-hosted on a node and they compete for the resources available on that node. Such platforms

allocate resources to competing applications on fair share basis and do not
provision guaranteed levels of resource capacities to an application.

Techniques in Existing Work

Fair-share based resource allocation

The resources on a single node are allocated on fair-share basis to the competing users. An application can use the idle resource capacity as long as others do not compete for it.

Time based predictive

Concerned with node availability in terms of the MTTF and MTTR of the nodes.

3. Proposed Work

We focused on developing node selection techniques, that nodes are likely to satisfy a
given requirement for long durations. Based on the characteristics of the resource utilization Planet lab nodes for various resource capacity requirements. We also develop an online model for predicting the idle resource capacity that is likely to be available on a node over a short term. We
evaluate and demonstrate the performance benefits of the node selection techniques and the prediction model using the Planet Lab node utilization data traces collected at different intervals. We are going to achieve dynamic community cloud computing
environments. The resource discovery
system focused to observe the resource
availability on dynamically at the rate of
95 percent accuracy.The reserved capacity is utilized and this information is sent to all cloud providers to increase their resource
capacity.

4. Technology Used

JSP: In our project we are using Jsp to design the application process. JSP pages are using to develop the form pages like login and user registration pages. it means it is mainly useful for user Interaction development. And some static content of html pages to jsp pages for dynamic content.

Servlet: In our project we are using Servlet to control the application process. Servlet is the center of our application because all the controlling part will be monitoring by the Servlet only. It means Servlet takes requests and matches for suitable jsp's and it is also useful for database controlling.

Collections:

The Java Collections, Java API's provide Java developers with a set of classes and
interfaces that makes it easier to handle collections of objects. Collections are useful to storing the values of a page as input to the backend and from backend to the front end carrying through network. Some of them are ArrayList, HashMap.

Thread: In this project threading concept is very important. Why because, to increase the speed of the service process when more number of users interact. Thread is a independent part of the same program it may execute independently.

MY SQL Database: In our project we are using a backend as My SQL 5.5. Here we are create and maintaining the tables which are having values used for resource management.

Testing: **Software testing** is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

Interfaces: An interface is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface. An interface is not a class. Writing an interface is similar to writing a class, but they are two different concepts. A class describes the attributes and behaviors of an object. An interface contains behaviors that a class implements.

Bean Classes Java Beans are using for less code and more work approach... Java Beans are used throughout Java EE as a universal contract for runtime discovery and access. For example, Java Server Pages (JSP) uses Java Beans as data transfer objects between pages or between servlets and JSPs. Java EE's JavaBeans Activation Framework uses Java Beans for integrating support for MIME data types into Java EE. The Java EE Management API uses JavaBeans as the
foundation for the instrumentation of resources to be managed in a Java EE environment.

Techniques used in Proposed Work

Profiling Based Node Selection

For selecting high quality nodes, we investigated the methods for discriminating
nodes based on their eligibility periods. Based on these methods, we develop profiling approaches for selecting nodes for a given requirement.

Node-level resource capacity prediction

Prediction method is based on observing the fluctuations in the available resource capacities over time.

KEYWORDS

MTTR: Mean Time To Repair (MTTR) is the time needed to repair a failed hardware module. In an operational system, repair generally means replacing a failed hardware part. Thus, hardware MTTR could be viewed

as mean time to replace a failed hardware module. Taking too long to
repair product drives up the cost of the installation in the long run, due to down time until the new part arrives and the possible window of time required scheduling the installation. To avoid
MTTR, many companies purchase spare products so that a replacement can be installed quickly. Generally, however, customers will inquire about the turn-around time of repairing a product, and indirectly, that can fall into the MTTR category.

ARCHITECTURE DIAGRAM

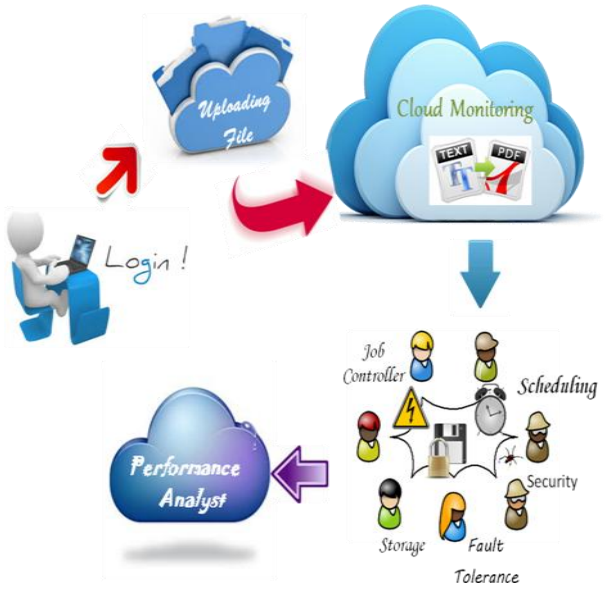

Figure 1: Architecture Diagram

MTTF:

Mean Time To Failure (MTTF) is a basic measure of reliability for non-repairable systems. It is the mean time expected until the first failure of a piece of equipment. MTTF is a statistical value and is meant to be the mean over a long period of time and a large number of units. Technically, MTBF should be used only in reference to a repairable item, while MTTF should be used for non-repairable items. However, MTBF is commonly used for both repairable and non-repairable items.

5. METHODOLOGY DESCRIPTION

❖ ***User Interface and Security Services***

In this module we design the webpage for the project. These webpage are used to interact from user to server. We use the Jsp and servlet available in Java to design the User Interface. JSP pages are using to develop the form pages like login and user registration pages. it means it is mainly useful for user Interaction development. And some static content of html pages to jsp pages for dynamic content. In this module mainly we are focusing the login design page with the Partial knowledge information. Application Users need to view the application they need to login through the User Interface GUI is the media to connect User and Media Database and login screen where user can input his/her user name, password and password will check in database, if that will be a valid username and password then he/she can access the database.

❖ ***Implementation of Job Controller***

In this project we are going to implement the protocol and the proposed protocol involves only adjustment steps, the job controller links the core and the application layers. It first calls the security service to verify if a user has permission (authentication) to execute jobs in cloud computing platform and what are the credentials of this user. Moreover, the job controller's main function is to manage distinct and simultaneously running workflows, noting that the workflows may belong to the same or to different users. Thus, for each accepted workflow, the job controller generates an associated ID and controls each workflow execution using this ID.

❖ ***Cloud Monitoring and Discovery Services***

Monitoring service verifies if a requested service is available in a cloud provider, searching for another cloud in the federation if it is not; receives the tasks to be executed from the job controller, and sends them to the scheduling service that will distribute them, guaranteeing that all the tasks of a process are correctly executed; and informs the job controller when a task successfully finishes its execution. To ensure the monitoring of all the requested tasks, this service periodically sends messages to the clouds that are executing tasks, and informs the user the current status of each submitted task.

❖ ***Scheduling and Storage Services***

In cloud computing, the monitoring service requests the scheduling service to find the best suitable provider by matching the gathered resource properties to the service requirements by applying predefined scheduling algorithms. If none of the providers can be matched, the monitoring service enables the discovery service, which must seek new cloud providers to be integrated into the federated environment. To realize this, the storage service can communicate with the discovery service to access information about the federation, since the discovery service knows the actual storage conditions of each provider integrating the federation.

6. REAL TIME EXAMPLES

Soft Layer cloud services:
http:// www.softlayer.com/
Rackspace cloud deployment services:
http:// www.rackspace.com/
Network World:

http://www.networkworld.com/

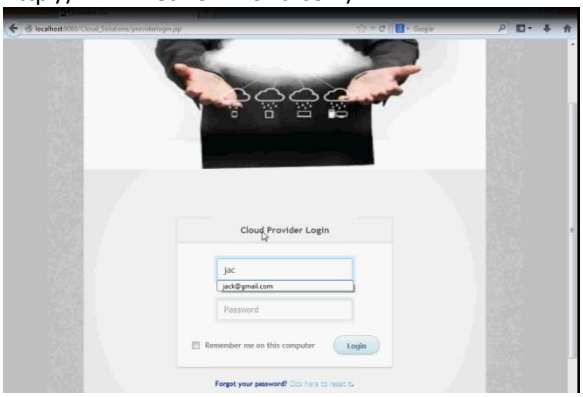

Figure 2:User Interface Design & Security Services

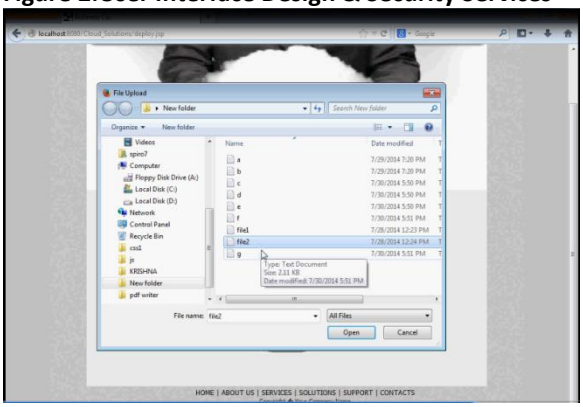

Figure 3: File Uploading

Figure 4: Cloud Monitoring & Discovery Services

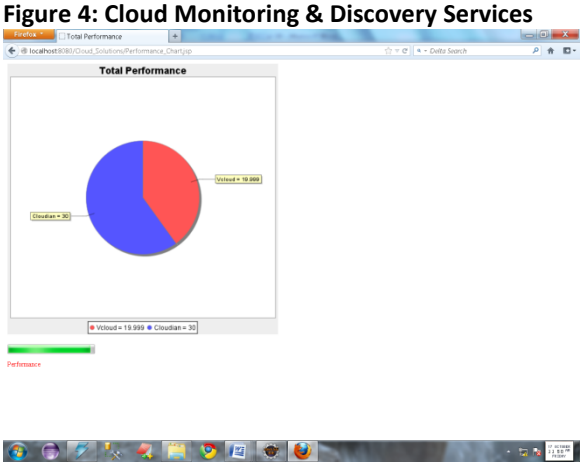

Figure 5: Performance Analyst

7. CONCLUSION

The scope of our project is selecting nodes on a cloud computing environments to deploy their user's application. Here we are forming the centralized resource manager by observing the characteristics of resource availability in long duration. The selecting nodes are likely to satisfy the given requirements. Scheduling the users to deploy their application based on resource availability. Additionally we are develop prediction model for predict the idle resource on such environments. we develop an online prediction model. Such prediction of available resource capacity can be useful for applications such as replicated services or any distributed application to load-balance requests or schedule their computation based on the estimated available resource capacity. We have presented method for intelligent selection of nodes to assist the platform users in selecting nodes for application deployment. For this, we first observed the nodes in satisfying different resource requirements. We used these observations to develop a profiling approach for selecting nodes. Thus, an application deployer would need to monitor nodes only for short time before selecting nodes for deployment. Finally our project shows the performance benefits.

ACKNOWLEDGEMENTS

I, Kavitha Rani acknowledge my gratitude for the sincere support provided to me by the management for providing all the help and facilities. I also express my sincere thanks to my guide who has been an integral part in helping us throughout the entire project

REFERENCES

[1]. A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L.Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak, "Operating System Support for Planetary-Scale Network Services," Proc. First Symp. Networked Systems Design and Implementation (NSDI '04), pp. 253-266, 2004.

[2]. P. Dinda, "Design, Implementation, and Performance of an Extensible Toolkit for Resource Prediction in Distributed Systems," IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 2, pp. 160- 173, Feb. 2006.

[3]. P.A. Dinda, "The Statistical Properties of Host Load," Scientific Programming, vol. 7, pp. 211-229, Aug. 1999.

[4]. L. Gong, X.-H. Sun, and E.F. Watson, "Performance Modeling and Prediction of Non-Dedicated Network Computing," IEEE Trans. Computers, vol. 51, pp. 1041-1055, Sept. 2002.

[5]. R. Iosup, M. Jan, O. Sonmez, and D.H.J. Epema, "On the Dynamic Resource Availability in

Grids," Proc. Eighth IEEE/ACM Int'l Conf. Grid Computing, 2007.

[6]. B. Javadi, D. Kondo, J.-M. Vincent, and D.P. Anderson, "Discovering Statistical Models of Availability in Large Distributed Systems: An Empirical Study of SETI@home," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 11, pp. 1896-1903, Nov. 2011.

[7]. D. Kondo, G. Fedak, F. Cappello, A.A. Chien, and H. Casanova, "Characterizing Resource Availability in Enterprise Desktop Grids," Future Generation Computing Systems, vol. 23, pp. 888-903, Aug. 2007.

[8]. D. Kondo, M. Taufer, C.L. Brooks III, H. Casanova, I. Henri, C. Andrew, and A.A. Chien, "Characterizing and Evaluating Desktop Grids: An Empirical Study," Proc. 18th Int'l Parallel and Distributed Processing Symp. (IPDPS '04), 2004.

[9]. M. Mutka and M. Livny, "The Available Capacity of a Privately Owned Workstation Environment," Performance Evaluation, vol. 12, no. 4, pp. 269-284, 1991.

[10]. M. Mutka, "Estimating Capacity for Sharing in a Privately Owned Workstation Environment," IEEE Trans. Software Eng., vol. 18, no. 4, pp. 319-328, Apr. 1992.