ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Selecting Nodes Based on Resource Availability in Planetlab System

[I]M.Vidhya, [II]D.Gomathi
[I]PG Student-CSE, [II]Asst. Professor-MCA
[I,II]Krishnasamy College of Engineering and Technology, Tamilnadu, India.

Abstract

To developing techniques for selecting nodes for 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 work is focused on developing node selection techniques; we first study the resource utilization characteristics of PlanetLab 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.

Keywords

Availability, Fault Tolerance, Prediction, Large-scale Systems, Resource Management, Scheduling.

I. Introduction

Cooperatively pooled resources in large scale platforms are generally spread over a wide area and loosely managed by the participating organizations. In this project it involved computing platforms such as PlanetLab [1] demonstrate the feasibility of using cooperatively pooled distributed resources for deploying experimental distributed systems and applications. PlanetLab is a geographically distributed overlay network designed to support the deployment and evaluation of planetary-scale network services. PlanetLab 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 [14] 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 [15]. 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 consider here cooperatively pooled shared computing platforms with the following characteristics, as exemplified by the PlanetLab platform. Existing system did 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. Allocate resources to competing applications on fair share basis and do not provision guaranteed levels of resource capacities to an application. In these platforms are time-shared among competing applications, available resources vary across nodes and over time. An application can use the idle resource capacity as long as others do not utilize that resource. Furthermore, as is the case in any large-scale computing infrastructure, there is no guarantee of node availability, no competing user, can use the idle resource capacity in such platforms [15]. In such computing environments, the owner of a node has autonomous control to shut it down at any time. Therefore, a node can become unavailable due to crashes as well as shutdowns. The available resource capacity [12] at a node may change significantly within 30-60 minutes. For deployment of an application, We focused on select nodes 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.

II. System Architecture

The architecture for selecting nodes on resource availability in large computing platforms is included in cloud monitoring and scheduling systems. It first utilizes the resource availability and then if the resource is not available, they have to monitoring the resource availability and there will be discovering the new available services into the client. If there is scheduled process, because so many resources are available in it, and discovery system introduces many more services, 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. The node have an autonomous control of the owner, so they can be shut it down when they crashes occurred so we use discovering, monitoring, scheduling, predicting node and fault tolerance service to recovering the system and reserving the resource for deploying their application.

A. Application

Generally an application means that application is a computer program designed to help people perform an activity. Here we use an application is a workflow that is used to giving an input by the user interface.

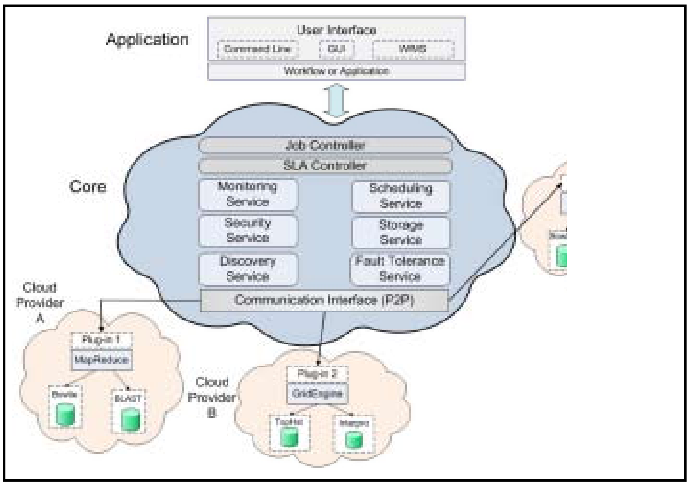

Fig. 1: System Architecture

B. Core

The core is the central part of architecture and it having all the process, services and techniques. It contains monitoring service, scheduling service, discovery service, storage service and fault tolerance services and the communication interface.

C. Cloud providers

A cloud providers is a person or organization, it is an entity that is responsible for making a service available for interested parties. For example Amazon, Grid Engine, Map Reduce, Vcloud etc…

III. Related Work

Characterizing resource availability is more appropriate for the platforms like PlanetLab because unlike Desktop Grids [7] or volunteer computing systems, the resources on a Planet- Lab node are allocated in fair-share manner to all competing users [13]. A great deal of previous work [3], [5-8], [11] has provided characterization and statistical models for resource availability in Desktop Grid systems and public resource computing systems such as SETI@home [6]. For all these previous work, the resource availability is based on activities of user only. But here the resource availability characteristics based on a node's eligibility [15] for a particular requirement based on the idle resource capacities available on that node. Here we used profile based Profiling Based Node Selection technique and Node-level resource capacity prediction.

1. Implementation of job controller

In this work 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.

2. Cloud monitoring

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.

3. Discovery services

Discovery service identifies the cloud providers integrating the federation, and consolidates information about storage and processing capabilities, network latency, availability of resources, available bioinformatics tools, details of parameters and input and output files. To realize this, the discovery service waits for information published by providers about their infrastructure and available tools. To consolidate these data, the discovery service maintains a data structure that is updated whenever new data is received. Furthermore, the discovery service has a policy of controlling each provider, removing from the federation those providers not regularly sending updated information, which guarantees the correct and update task execution on the federated cloud.

4. 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.

5. Predicting node and fault tolerance

This service guarantees that all the core services are always available. In a cloud environment, machine failures occur, and it is well known among the cloud community that those failures are the norm rather than the exception. Thus, any federated cloud should be designed for fault recovering and system availability. Therefore, a fault tolerance service is an essential part of our federated cloud, and has the objective of providing high availability and resiliency against periodic or transient failures. They have to be recovered from system failures like crashes and shutdown suddenly.

IV. System Techniques

In these work there are two techniques are used one of them is profiling based node selection and another one is node-level resource capacity prediction. These two techniques are explained below.

A. Profiling based node selection

For selecting high quality nodes, we investigated the methods for discriminating nodes based on their eligibility periods. The set of nodes satisfying a given resource capacity requirement as its eligibility set [15]. The eligibility period of a node is defined as the contiguous period for which it remains in the eligibility set. Based on these methods, we develop profiling approaches for selecting nodes for a given requirement.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

1. The node has remained eligible for certain duration.
2. The previous eligibility period of the node was greater than certain threshold.

B. Node-level resource capacity prediction

Prediction method is based on observing the fluctuations in the available resource capacities over time. It predicts the idle resource capacity. Prediction for a given resource at a node the amount of its idle capacity that is likely to be available in the near future with some given probability. The prediction of available capacity can be useful for an application to estimate how much additional capacity is likely to be available beyond its resource requirements [13].

V. Conclusion

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 deployed would need to monitor nodes only for short time before selecting nodes for deployment. 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.

VI. Acknowledgment

Thanks to my Guide and family member who always support, help and guide me during my study. Special thanks to my husband who always support my innovative ideas and encourage me to doing that.

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.*

[11] *Mechanisms for Building Autonomically Scalable Services on Cooperatively Shared Computing Platforms, Padhye V., Tripathi A., Software Practice and Experience (SPE), june -2013.*

[12] *D. Oppenheimer, B. Chun, D. Patterson, A.C. Snoeren, and A.Vahdat, "Service Placement in a Shared Wide-Area Platform,"Proc. Ann. Conf. USENIX '06 Annual Technical Conf. (USENIXATC), pp. 273-288, 2006.*

[13] *V. Padhye and A. Tripathi, "Building Autonomically Scalable Services on Wide-Area Shared computing Platforms," Proc. IEEE Network Computing and Applications,pp.314-319,2011.*

[14] *Resource-Aware Migratory Services in Wide-Area Shared Computing Environments - A Tripathi, V Padhye, D Kulkarni -- Reliable Distributed Systems, 2009.*

[15] *Resource availability characteristics and node selection in cooperatively shared computing platforms V Padhye, A Tripathi – 2013.*

[16] *X. Ren, S. Lee, R. Eigenmann, and S. Bagchi, "Resource Availability Prediction in Fine-Grained Cycle Sharing Systems," Proc. 15th IEEE Int'l Conf. High Performance Distributed Computing (HPDC),pp. 93-104, 2006.*

[17] *D. Stutzbach and R. Rejaie, "Understanding Churn in Peer-to-Peer Networks," Proc. Sixth ACM SIGCOMM Conf. Internet Measurement, pp. 189-202, 2006.*

[18] *T. Warns, C. Storm, and W. Hasselbring, "Availability of Globally Distributed Nodes," Proc. IEEE Symp. Reliable Distributed Systems, pp. 279-284, 2008.*

[19] *R. Wolski, N. Spring, and J. Hayes, "Predicting the CPU Availability of Time-Shared Unix Systems on the Computational Grid," Cluster Computing, vol. 3, no. 4, pp. 293-301, 2000.*

[20] *R. Wolski, N.T. Spring, and J. Hayes, "The Network Weather Service:A Distributed Resource Performance Forecasting Service for Metacomputing," Future Generation Computer Systems, vol. 15, no. 5/6, pp. 757-768, 1999.*

Author's Profile

M.Vidhya pursuing M.E, Computer Science and Engineering from Krishnasamy college of engineering and Technology, Cuddalore, Tamilnadu, India.

D.Gomathi working as Asst.Professor in Krishnasamy college of engineering and Technology, Cuddalore, Tamilnadu, India