An Efficient Cloud Resource Allocation and Job Scheduling Using Firefly Algorithm based on Resource Monitoring and Prediction

S.LEENA
PG Scholar
KSR Institute for Engineering and Technology
Tiruchengode-637215
ssrileena@gmail.com

T.SOUNDER RAJAN
AP / CSE
KSR Institute for Engineering and Technology
Tiruchengode-637215
tsounderrajan@gmail.com

M.SELVAM
Principal
ACT College of Engineering and Technology
Kancheepuram

Abstract- **Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. Resource allocation and job scheduling are the core functions of cloud computing. These functions are based on adequate information about available resources. Thus timely acquiring resource status information is of great importance in ensuring overall performance of cloud computing. A distributed computing system consists of multiple software components that are on multiple computers, but run as a single system. As the web, social networking, and smart phone application have been popular, the data has grown drastically every day. Such effectively grown data is termed as resource monitoring information. This resource monitoring information allows clients to have exclusive access to cloud resources and additionally it delivers higher levels of quality of services. This work aims to analyze current resource monitoring strategies and resource prediction in cloud node.**

Keywords- **Resource Monitoring, Resource Allocation, Job Scheduling.**

I. INTRODUCTION

The resource performance is the most influencing factor in cloud computing. The basic problem in cloud computing is allocation of resources and scheduling the task. Two mechanism are used to achieve the performance of resources in cloud computing. They are resource monitoring and prediction system. Resource monitoring in the cloud computing gather the information about the resources. Fault resources in the cloud are identify by means of monitoring strategies.

Resource prediction identify the variation trend based on the historical information gathered by the resource monitoring. In cloud computing resource monitoring provide historical information and current information by means of resource sensor while the prediction system generate variation trend based on the historical information collected by the resource sensor. The monitoring and prediction data in the cloud focuses on

1. Detect the fault resources and avoid the bottleneck
2. With minimized cost and time, cloud user can effectively compute the task
3. Ensure overall performance of cloud computing by effective resource allocation and job scheduling with dynamic load balancing

II. REPRESENTATIVE SYSTEM

A) Monitoring System

In order to monitor the resources the number of monitoring tool is used. But these tools are used for individual personal computer or single cluster so they cannot used in the cloud system. However monitoring system employed and resource sensor realized are reusable for cloud system. Monitoring system consist of resource sensor. Resource sensor collect the information about the resources. It provided whether the resources are idle or it is allocated to the task. Monitoring the resources by resource sensor collects the historical information and current status information in the cloud. This gathered information helps resource allocation and job scheduling in the cloud. During monitoring of resources it predict the efficient resources and fault resource in the cloud. Task are allocated to the efficient resources in the cloud.

Let S_-=(s(1),s(2),…s(t)) this set generated by the resource sensor which represent the historical information about the resources in the cloud. Where S represents the state of the resources at particular time. S_- Represents the historical data collected by the resource sensor. Based on the historical data gathered by the monitoring system the prediction system is to be performed.

B) Prediction System

A historical data set is divided into three parts: training, validation and test sets. Prediction model can be built by using training set. Validation set is used to optimize the outcome of training set. Test set is used to evaluate the historical data. Finally prediction model is used to generate the variation trend. This variation trend predict whether there is a demand for the particular type of resources or not.

Let S_+=(s(t+1),s(t+2),…) this set generated by the prediction model which represent the future set. Where S represents the state of the resources in future. S_+ represents state of the particular resources in future based on the historical data gathered by the monitoring system.

III. SYSTEM ARCHITECTURE DESIGN

A) System Design Principles

Monitoring the resources will timely update the information of resources in the cloud. For user the resource sensor and prediction models are periodically executed to

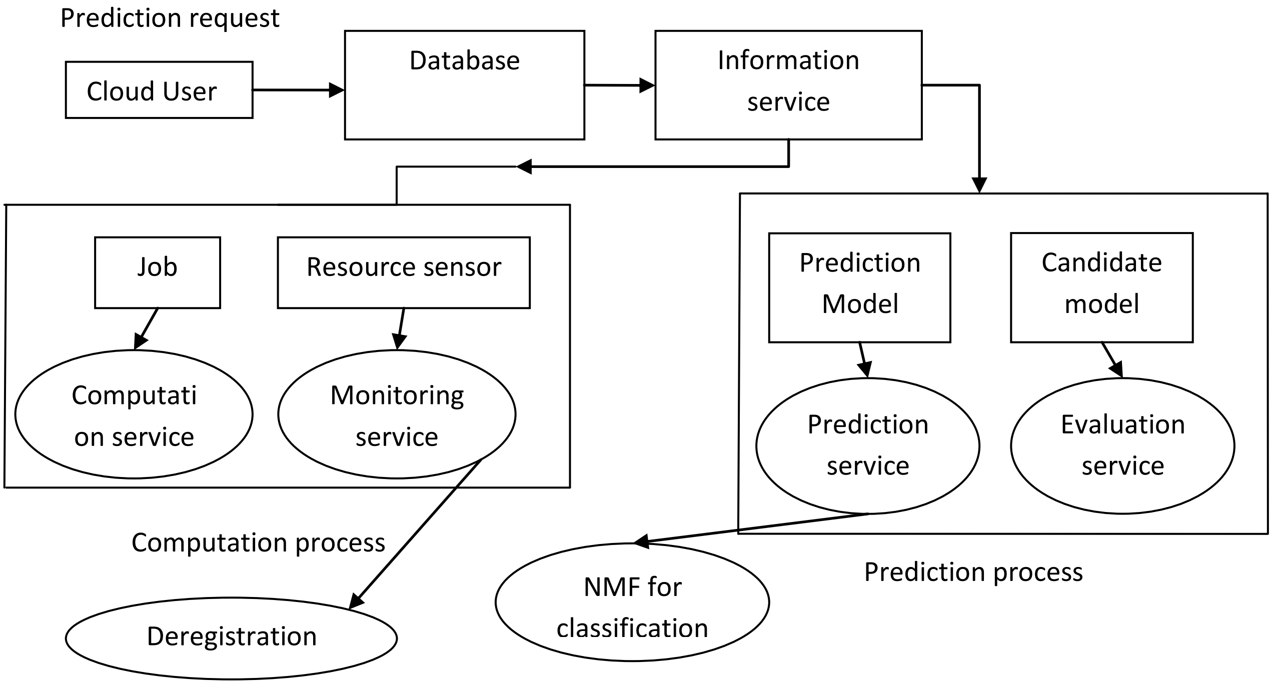

Fig.1. System Architecture

generate up-to-date information. This leads to increases responsiveness in cloud. Each computing node is deployed with resource sensor. This resource sensor dynamically sleep and run in order to reduce the overhead. Monitoring the resources by means of resource sensor are user friendly. This increases the efficiency of resources in the cloud computing.

Information gathered by the resource sensor are in the form of XML format. So that it can easily interact with other component and support with other resource type. This achieves the extensibility and modularity in the cloud.

B) Service Description

System design consist of two subsystem: monitoring system and prediction system. Monitoring system consist of number of computing system and task scheduler. Task scheduler assign the task to the computing resources in the monitoring system. Each computing services is deployed with the computing service and the monitoring service. The responsibility of computing service is to run the job on the computing services. Monitoring services consist of resource sensor. This resource sensor dynamically run and sleep in order to reduce the overhead. Resource sensor collect the information about the historical data and current state information of resources in the cloud. The gathered information of resources in the cloud are stored in the database. This database are maintained by the information service.

Prediction subsystem consist of number of prediction node. Each prediction node is deployed with prediction service and evaluation service. Prediction service retrieve the historical data from the database and generate the variation trend. Evaluation service based on candidate model. This evaluation services predict whether the resource satisfies the quality of constraints (QoS) or not.

Information gathered by the monitoring system are in the form of XML format. XML format is mainly used in order to support new resource types and interact with other components.

C) System Work Flow

Cloud user generate the prediction request based on the user needs. Once the request is generated by the cloud user then monitoring request is send by the information service in order to activate the resource sensor. Resource sensor gather and store the historical information in the database. Prediction request made by the information service to the prediction service. Prediction service then request for historical data to the information service.

Information service retrieve the historical data from the database and provide it to the prediction service. Based on the historical data evaluation result is generated by the evaluation service. Prediction service also request for updated monitoring data in order to evaluate the evaluation result. The result of prediction is finally send to the cloud user.

In order to deactivate the resource sensor the terminal request is send to the monitoring service and resource sensor become sleep mode. Resource sensor send deregister to the information service. In order to deactivate the prediction model terminal request is send to the prediction service and prediction model become sleep mode. Prediction model send the deregister to the information service.

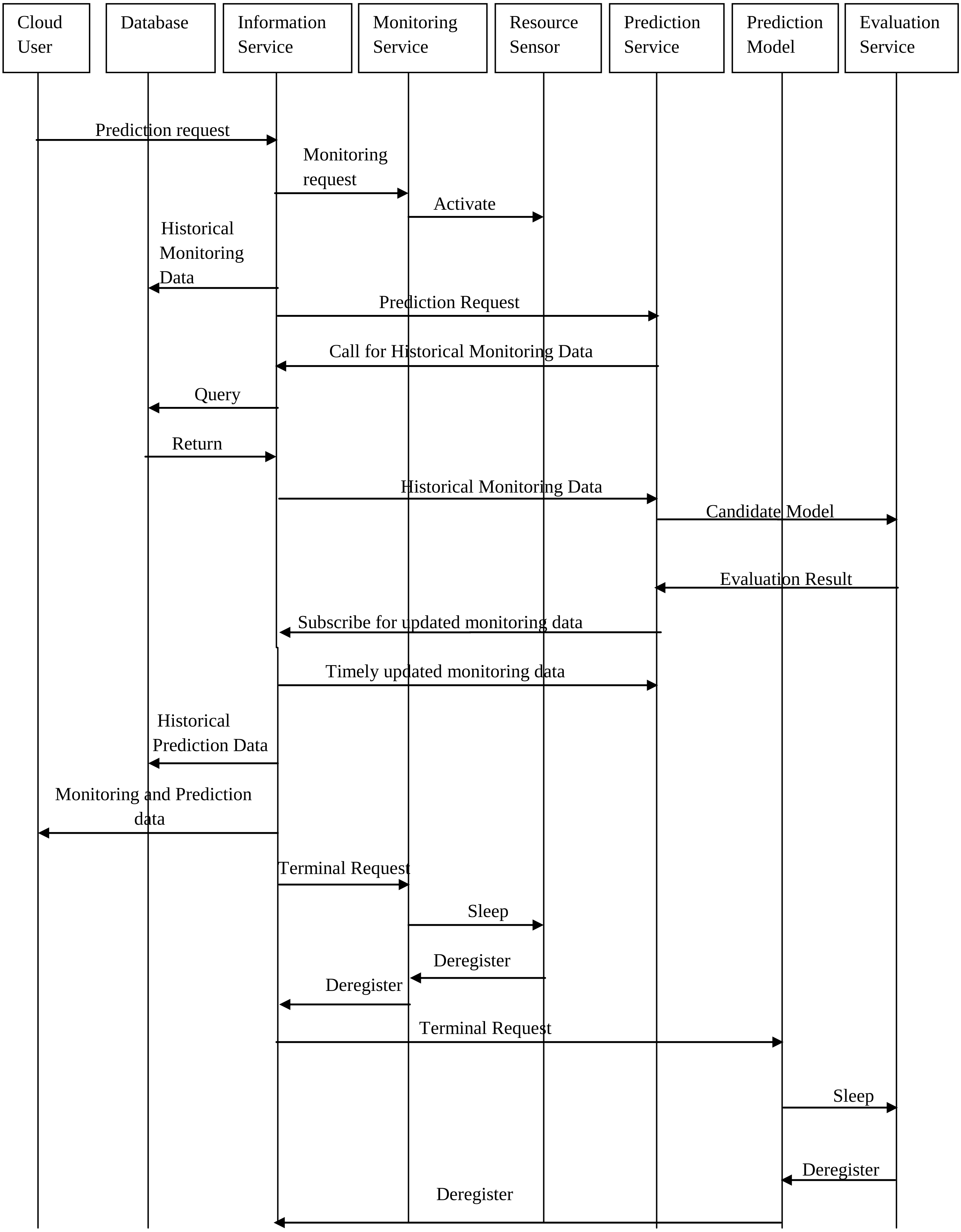

Fig.2. System Work Flow

IV. PROPOSED WORK

The optimized prediction model is built using the parallel hybrid particle swarm optimization method. However this work will consume more time complexity when there is a more possible alternatives allocation present. It will be overcome by introducing the new optimization methodology called the firefly algorithm. This algorithm aims to reduce the time complexity which is occurred by using PSO optimization algorithm. The experimental tests conducted proves that the proposed methodology can provide the better result than the existing approaches.

In order to provide better results, firefly algorithm is being introduced. Firefly Optimization algorithm is used for resource prediction and it uses the subsystem problem. Combinational optimization problem with a hybrid vector, which consists of real numbers and binary numbers is coded. The firefly algorithm (FA) is inspired by the flashing behavior of fireflies whose primary purpose is to act as a signal system to attract other fireflies. All fireflies are unisexual, so that one firefly will be attracted to all other fireflies as stated. It encounters that the attractiveness is proportional to their brightness, and for any two fireflies, the less bright one will be attracted by the brighter one. If there are no fireflies brighter than a given firefly, it will move randomly. Firefly algorithm is a nature-inspired metaheuristic optimization algorithm.

ALGORITHM

Begin
Objective function: $f(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, ..., x_d)$;
An initial population of fireflies is generated by:
$\mathbf{x}_i \quad (i = 1, 2, \ldots, n)$;.
Light intensity I is associated with $f(\mathbf{x})$
Absorption coefficient γ
While (t < MaxGeneration)
For a = 1 : n (all n fireflies)
For b = 1 : n (n fireflies)
If ($I_b > I_a$)
Move firefly i towards j;
End if
Attractiveness varies with distance r $\exp(-\gamma\, r)$;
New solutions and update light intensity are evaluated,
End for b
End for a
Rank fireflies and find the current best;
End while
Results are post processed and visualized
End

V. CONCLUSION

Therefore proposed system for distributed resource monitoring architecture concludes that it seamlessly combines cloud technologies together with resource monitoring as well as prediction systems. This integrated system consists of a set of distributed services to accomplish all required resource monitoring, prediction with data gathering functions and evaluation functions respectively. The analysis of current resource monitoring is achieved by means of resource sensor. A prediction model has been developed which evaluates the prediction system in an effective manner by the proposal of Firefly algorithm.

REFERENCES

[1] Bittencourt, L.F. and Madeira, E.R.M. (2005) 'A Performance-Oriented Adaptive Scheduler for Dependent Tasks on Cloud,' Concurrency and Computation: Practice and Experience.

[2] Caron, E. Chis, A. Desprez, F. and Su, A. 'Design of Plug-in Schedulers for a CLOUDRPC Environment,' Future Generation Computer Systems, vol. 24, no. 1, pp. 46-57.

[3] Dinda, P.A. and O'Hallaron, D.R. (2010) 'Host Load Prediction Using Linear Models,' Cluster Computing, vol. 3, no. 4, pp. 265-280.

[4] David R, (2011) 'Design, Implementation, and Performance of an Extensible Toolkit for Resource Monitoring and Prediction in Distributed Systems,' IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 2,b pp. 160-173.

[5] Eddy Caron, Andreea Chis, Frederic Desprez, Alan Su (November 2011) 'Plug-in Scheduler Design for a Distributed Cloud Environment'.

[6] Liang Hu, Xi-Long Che, (2012) 'Online System for Cloud Resource Monitoring and Machine Learning-Based Prediction' IEEE transactions on parallel and distributed systems, vol. 23.

[7] Massie, M.L. Chun, B.N. and Culler, D.E. (2009) 'The Ganglia Distributed Monitoring System: Design, Implementation, and Experience,' Parallel Computing, vol. 30, no. 7, pp. 817-840.

[8] Peter Dinda, A. and David R. O'Hallaron (July 2012) 'An Extensible Toolkit for Resource Monitoring and Prediction in Distributed Systems' School of Computer Science Carnegie Mellon University Pittsburgh, PA ,15213.

[9] Sam Verboven, Peter Hellinckx, FransArickx and Jan Broeckhove 'Runtime monitoring and Prediction based Cloud Scheduling of Parameter Sweep Jobs' University of Antwerp Antwerp, Belgium.

[10] Sodan, A.C. Gupta, G. Han, L. Liu, L. and Lafreniere, B. (2011) 'Time and Space Adaptation for Computational Cloud with the ATOP Cloud Middleware,' Future Generation Computer Systems, vol. 24, no. 6, pp. 561-581.