Fuzzy Subtractive Clustering Based Prediction Approach for CPU Load Availability

Kadda Beghdad Bey
Laboratoire de Systèmes Informatiques
Ecole Militaire Polytechnique
Algiers, Algeria
bey_kadda@yahoo.fr

Farid Benhammadi
Laboratoire de Systèmes Informatiques
Ecole Militaire Polytechnique
Algiers, Algeria
benhammadif@yahoo.fr

Faouzi Sebbak
Laboratoire de Systèmes Informatiques
Ecole Militaire Polytechnique
Algiers, Algeria
faouzi.sebbak@gmail.com

Abstract—Distributed processing environment has emerged as a new vision for future network based calculation, allowing the federation of heterogeneous computing resources to incorporate the power. Cloud computing is a new computing paradigm composed of a combination of grid computing and utility computing concepts. In cloud computing, the prediction methods play a key role in managing large scale of computation capacity. In this paper, a modelling approach to predict the future CPU load value is presented. The proposed approach employs a computational intelligence technique to classify the CPU load time series into similarity component group. This technique is based on the Fuzzy Subtractive Clustering algorithm and a combination of local Adaptive Network-based Fuzzy Inference System. The results of an exhaustive set of experiments are reported to validate the proposed prediction model and to evaluate the accuracy of their prediction. Experimental results demonstrate both feasibility and effectiveness of our approach that achieves important improvement with respect to the existing CPU load prediction models.

Keywords-Subtractive clustering; CPU load prediction; cloud computing; system modelling; ANFIS.

I. INTRODUCTION

Heterogeneous computer network environments involve effective utilization of the distributed resources to achieve high performance computing. Cloud computing is a new computing paradigm composed of a combination of grid computing and utility computing concepts. Cloud promises high scalability, flexibility and cost-effectiveness to satisfy emerging computing requirements; therefore, they can treat task scheduling and resource allocation over the virtual clusters [1]. In the literature, various architectures have been proposed to satisfy the user's needs in terms of computational power through the use of distributed computing resources [2]. In distributed environments, resources monitoring needs continual parameters monitoring in terms of CPU load, memory size, bandwidth and latency. Irrespective of the nature and the type of the used distributed processing environment, the creation of resource pools should satisfy several requirements for each parameter quality during the computation service. To efficiently provision computing resources in the cloud, the ability to accurately predict resource capabilities is of great importance since it permits to determine how to use time-shared resources.

Many interesting modelling strategies have been proposed to predict available CPU load in a grid computing environment [3,4,5]. The main contribution of the present paper relies on the integration of the subtractive clustering technique and the Fuzzy Inference System (FIS) to make short and medium-term predictions of CPU availability on time-shared environment systems. The proposed approach predicts the future value of CPU load based on a set of local Adaptive Network-based Fuzzy Inference System (ANFIS) predictors to perform short-term accurate and mid-term reliable prediction using the selection instances in several past steps. We also propose a deterministic approach for k-folds cross-validation that constructs representative rather random folds. Through this approach, we attempt to reduce the effects of using only a few instances for training.

The rest of this paper is organized as follows. Section 2 reviews the related works about CPU load prediction approaches in time-shared systems. Section 3 presents the proposed subtractive clustering-based ANFIS prediction model. This section also describes how this software is used to carry out experiments. Experimental results are reported in Section 4. Conclusions and directions for future work end the paper.

II. RELATED WORK

A Cloud computing platform offers to users a virtualized distributed system, where computing resources are dynamically allocated to satisfy a user's Service Level Agreement. Predicting the processor availability for a new process or task in computer network systems is a basic problem arising in many important contexts. Making such predictions is not easy because of the dynamic nature of current computer systems and their workload.

The Network Weather Service (NWS) [3] is the most famous system designed to provide dynamic resource performance forecasting. The predictive methods currently used in NWS include running average, sliding window average, last measurement, adaptive window average, median filter, adaptive window median, α-trimmed mean, stochastic gradient, and auto-regression (AR). Dida [6] studied different linear series models including autoregressive, moving average, autoregressive moving average, autoregressive integrated moving average and autoregressive fractionally integrated moving average models, for predicting future loads from 1 to 30 seconds. Huo et al. [7] evaluated four criteria to determine the

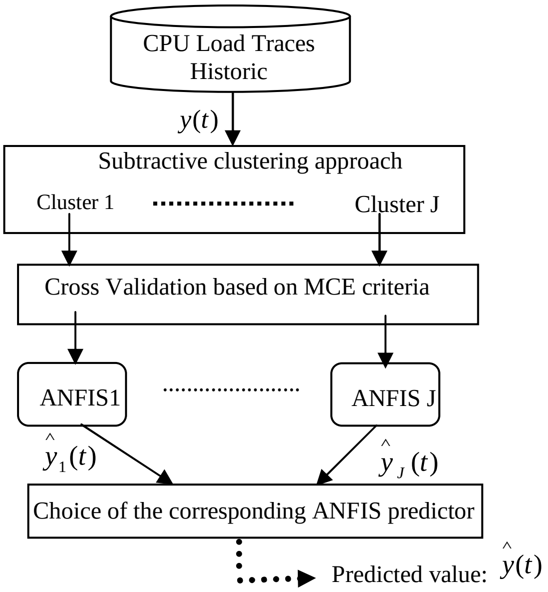
ISBN: 978-1-61208-271-4

optimal order of AR models: Final Prediction Error (FPE), Akaike's Information Criterion (AIC), Minimum Description Length (MDL) and Bayesian Information Criterion (BIC). The authors claimed that the BIC criterion performs better than other criteria. An approach based on the Tendency-Based and Polynomial fitting method predictor is proposed by Yang et al. [8]. Liang et al. [9], presented a more-generic prediction scheme using both the autocorrelation of CPU load and the cross correlation between CPU load and free memory to achieve higher CPU load prediction accuracy. In [10], Zhang et al. tackled the problem of predicting available CPU performance in a time-shared grid system. Their strategy forecasts the future CPU load based on the variety tendency in several past steps and in previous similar patterns. Recently, non linear models have been tried for time series prediction [11,12,13]. Liu et al. [13] proposed a hybrid non-linear time-series segmentation algorithm to discover duration-series pattern. In the experiment, they compared six approaches including LAST, MEAN, Exponential Smoothing, Moving Average, AR and Network Weather Service.

The present framework is related to our prior efforts in CPU load prediction and complements the existing performance CPU load prediction schemes [11, 12] with a modification of the soft computing algorithm using a subtractive clustering method. The new prediction system combines the subtractive clustering method and ANFIS. A strong point of our model is that it contains the same set of predictors which are able to deliver accurate prediction in peaks, switch level and regular situations.

III. SUBTRACTIVE CLUSTERING-BASED ANFIS PREDICTION

Cloud computing has become a great solution for providing a flexible and dynamically scalable computing infrastructure for many applications. Cloud computing presents a significant technology trends, and it is already obvious that it is reshaping information technology process [19]. To realize the next generation of distributed computing, we need to be able to accurately predict resource utilization. In this work, we proposed a novel model to predict the behavior of computing resources. Fuzzy models have been shown to be very effective techniques for the modelling of nonlinear, uncertain and complex systems. Subtractive Clustering is a fast one-pass algorithm for estimating the number of clusters and determining the cluster centres in a set of data [14]. We use the subtractive clustering if we do not have a clear idea about how many clusters should be used for a given data set. After clustering the data set, the number of fuzzy rules and premise fuzzy membership function are determined. Then, the linear squares estimate is used to determine the consequent in the output membership function, which provides a valid fuzzy inference system (FIS). The proposed approach includes three major steps: CPU load time series clustering, the ANFIS clusters model prediction and the combination of local ANFIS prediction model. As shown in Fig. 1, before making predictions about future CPU load values, subtractive clustering is applied to divide the historic CPU load data into sub-clusters and generate more homogeneous data.


CPU Load Traces Historic
$y(t)$
Subtractive clustering approach
Cluster 1
Cluster J
Cross Validation based on MCE criteria
ANFIS1
ANFIS J
$\hat{y}_1(t)$
$\hat{y}_J(t)$
Choice of the corresponding ANFIS predictor
Predicted value: $\hat{y}(t)$

$y(t)$: *CPU load time series.*

$\hat{y}_J(t)$: *Output from ANFIS J at time t.*

MCE : Minimum Checking Error

Figure 1. Subtractive clustering-based ANFIS prediction.

A. *CPU load time series clustering*

The purpose of this step is to identify natural groupings of CPU time series from a large set of historic traces, and to produce a concise representation of the system's behaviour. For our problem, one does not have a clear idea about the number of clusters to be used for a given set of data. Subtractive clustering technique, proposed by Chiu [14], has been shown to be a fast way of estimating the number of clusters and their centres positions. This technique calculates the density function based on the positions of data points, which leads to a significant reduction of the number of calculations. Each data point is a candidate to become a cluster centre. A density measure at data point x_i is defined as:

$$D_i = \sum_{j=1}^{n} \exp\left(-\frac{\|x_i - x_j\|^2}{(r_a/2)^2}\right) \qquad (1)$$

where r_a is a positive constant representing a neighbourhood radius. Hence, the more neighbouring points a data point has, the higher is its density. The density measure of each data point x_i is defined as follows:

$$D_i = D_i - D_{c_1} \exp\left(-\frac{\|x_i - x_j\|^2}{(r_b/2)^2}\right) \qquad (2)$$

where r_b is a positive constant that defines a neighbourhood that has measurable reductions in density measure. Thus, the data points near the first cluster centre x_{c_1} will have significantly reduced density measure.

After updating the density function, the next cluster centre is selected as the point having the highest density value. This process continues until a sufficient clusters number is attainted. Fig. 2 shows an example of CPU time series clustering based on the subtractive clustering method.

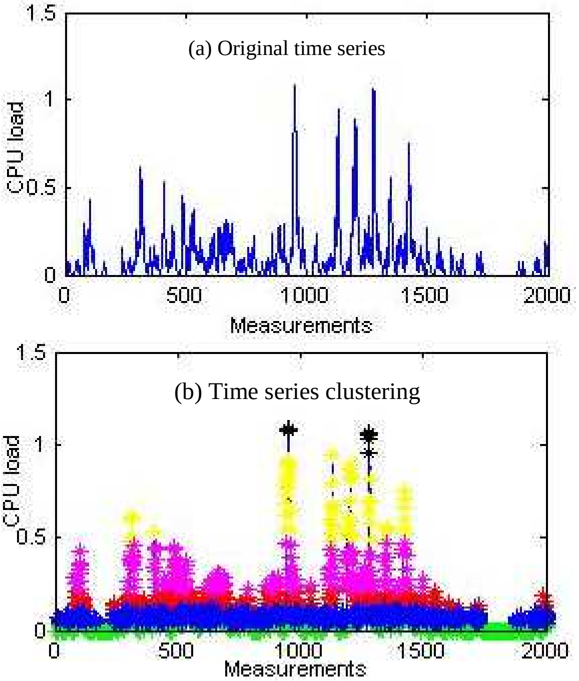


Figure 2. An example of CPU loads time series clustering.

For CPU load time series clustering, we use known values of the dynamical situation of the historic data up to time *t*. Let Y*(t)={y$_1$,y$_2$,…,y$_t$}* be the time series at time t. The dynamical situation Δy_t at time t is defined as follows:

$$\Delta Y_t = \{y_t - y_{t-1}, y_{t-1} - y_{t-2}, \cdots, y_2 - y_1\} \quad (3)$$

The Subtractive clustering technique is used to cluster all time series y_t into clusters. It estimates the number of clusters and the cluster centres. This process assigns the CPU load data y_t using the cluster membership degree μ_j that represents the degree to which y_t belongs to cluster c_j. This assignment is computed using the following objective minimization function:

$$O = \sum_{j=1}^{J} \sum_{t=1}^{t=T} (\mu_j(t))^2 \left\| \Delta y_t - \nu_j \right\| \quad (4)$$

where ν_j is the centre of cluster j and J is the number of clusters.

B. ANFIS Predictor

The Adaptive Network-based Fuzzy Inference System (ANFIS) proposed by Roger Jang [15] is one of the most commonly used fuzzy inference systems. It is a universal approximator used in various applications of predictions. Moreover, it has been proven to be more powerful than other models for short term prediction. ANFIS is a 5-layer feed-forward network in which each node performs a particular function in incoming signals, as well as a set of parameters pertaining to that node. Similarly to ANFIS, the compensatory neural fuzzy network with n-dimensional input-data vector x_p and one-dimensional output-data vector y_p has 5 functional layers: input layer, fuzzification layer, pessimistic-optimistic operation layer, compensatory operation layer (fuzzy reasoning method) and defuzzification layer.

Let us suppose that the fuzzy inference system under consideration has four inputs and one output. If two fuzzy sets are associated with each entry variable, then the system presents 16 inferences rules Rj (24), that are of the first-order Sugeno fuzzy type:

Rj : if (x1 is A1j) and (x2 is A2j)
and (x3 is A3j) and (x4 is A4j) (5)
Then yj=fi(x)=c1jx1+ c2jx2 + c3jx3+ c1jx1+ c4jx4=Bj

These rules correspond to the third category of fuzzy inference systems mentioned in [16]. One of the most important stages of the Neuro-fuzzy TSK (Takagi-Sugeno-Kang) network generation is the establishment of inference rules (Takagi and Sugeno 1985) [17] often used is the so-called grid method, in which the rules are defined as the combinations of the membership functions for each input variable.

C. Future CPU load Prediction

In this study, Adaptive Network-based Fuzzy Inference System based subtractive clustering has been used to predict availability of the CPU load. In our previous works [11, 12], a simple method for accuracy estimation is used. The dataset is randomly portioned in two disjoint subsets of N/2 instances. The first subset serves as the training set and the second one as the test set. The drawback of this method is that it makes inefficient use of data since typically a relatively large proportion of the instances is used for testing [18]. Cross-validation attempts to resolve this drawback by successively removing some instances from the initial set, treating them as a test set. In *k-fold cross-validation*, the dataset is randomly partitioned into *k* disjoint blocks (folds), of approximately equal size *d (d ≈ N / k)*. The learning algorithm runs *k* times. In the *i*th iteration, the *i*th training set is formed by the initial dataset without the *i*th fold, while the test set is formed using the *i*th fold alone [18]. The aim of directing similar instances to different folds is to reduce the pessimistic effects caused by the removal of instances from the dataset. The principle for constructing representative folds in unsupervised stratification is to channel similar instances to different folds in order to reduce the effects of using fewer instances for training.

For the final decision of CPU load time series prediction, we have used cluster predictor to select the adequate ANFIS predictor. After the application of the subtractive clustering method above the dataset, the instance space is partitioned into clusters. The next step is to determine the appropriate cluster, which aims at predicting future CPU load cluster based upon the observed history. The appropriate cluster for final decision of CPU load prediction is defined by the largest similarity between the cluster centres and the input times series points, as show in Fig. 3.

```
For each time series point $X_i$
    Find the cluster centres $C_j$
    $C_c$= the closest centre to $X_i$
      For j=1 to J        J: number of cluster
          /*Calculate the similarity Sim between
                        the centre $C_k$ and $X_i$
          S =Sim ( $X_i$, $C_k$)
      End
      /* Find the largest similarity $S_L$ between
                        $X_i$ and all other centres
      $S_L$= Max(Sim( $X_i$, $C_k$))
      $C_c$=$C_k$
End
```

Figure 3. Selection of appropriate cluster

IV. EXPERIMENTAL RESULTS

In the previous section, we have presented a new prediction approach for CPU load availability. In the present one, we assess its performance with respect to other methods. For this purpose, we carry out series of experiments on different CPU load time series with a variety of statistical properties collected by Dinda [19]. These CPU load traces were collected for two time periods on roughly the same group of machines. The traces used are in two column whitespace-delimited ASCII format. The first column gives the time stamp in seconds whereas the second one provides the floating point measured load value.

A. *Prediction model validation*

To generate a FIS using ANFIS, it is important to select the number of Membership Functions (MF) and the proper parameters for the learning and refining process. For training and testing data sets, we analyse the effect of these parameters on the final ANFIS performance including the training and testing minimum checking error (MCE). We evaluate and compare our prediction model with previous approaches using the Normalized Mean Square Error (NMSE) defined by:

$$NMSE = \frac{\sum_{t=1}^{T} \left(y_t - \hat{y}_t\right)^2}{\sum_{t=1}^{T} \left(y_t - \frac{1}{T} \sum_{t=1}^{T} y_t \right)^2} \qquad (6)$$

where $\hat{y}_t$ represents the CPU prediction value, y_t the actual measurement, and T the number of time series points.

The proposed ANFIS prediction model is based on the subtractive clustering process that resolves the problem of clusters number used for each CPU load time series. Though, this method determines the optimal number of cluster for each CPU load traces. Table 1 summarizes the prediction results of the CPU load time series from the proposed prediction model for four different machines traces collected by Yang [20]. This table shows that the Subtractive Clustering-based ANFIS model achieves better performance than other strategies for the same four load traces. The converged RMSE is much smaller than for the models reported in [11,12].

TABLE 1. NMSE FOR DIVERSE CPU LOAD PREDICTION

Prediction of future value x (t+1)	***NMSE***	***Error %*** ***(min/max)***
axp0Aug.180	0.056297	9,44 / 10,7
Abyss.1000	0.031459	1,13 / 3,06
Mystere.10000	0.26987	6,18 / 10,02
axp1Aug.120	1.185	6,38 / 45,99

We also tested some other prediction models including ours, ANFIS without clustering and Mixture of ANFIS. Fig. 3 illustrates a comparison between these three prediction models for five machines using different CPU load time series. The Mean Error Prediction of the proposed subtractive clustering based-model is smaller than that of other models. The predictive results of one traces machines using the Subtractive Clustering-based ANFIS model are shown in Fig. 4. The obtained prediction mean error was 0.08% whereas the RMSE is less than 0.15%. This shows again the consistent improvements of the proposed approach on the prediction quality over the corresponding time series collected on these machines.

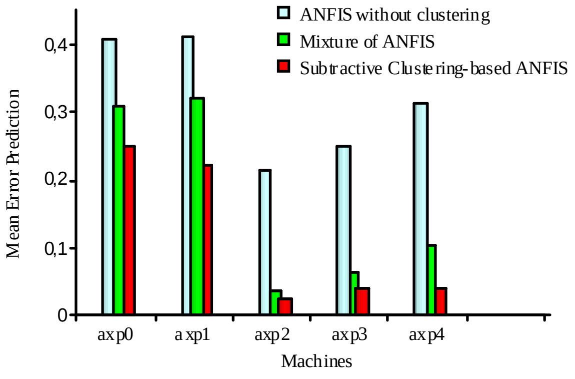


Figure 3. Comparison of three CPU load prediction models

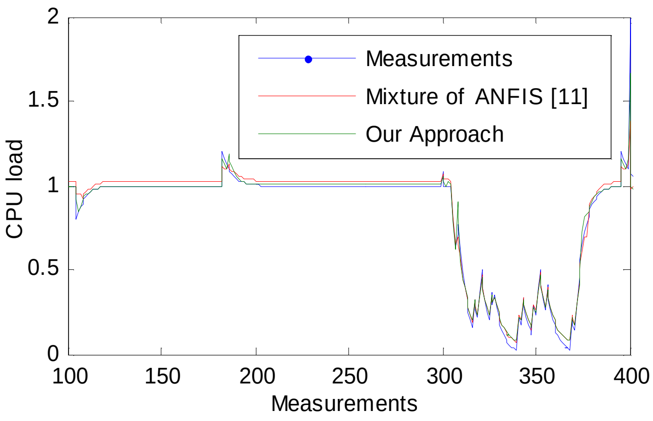


Figure 4. Comparative results of our predictor with Mixture of ANFIS [11].

B. Comparisons with other models

To evaluate the performances of the proposed prediction approach with respect to the existing ones, we have assembled test data from multiple datasets. The results of the subtractive clustering-based ANFIS prediction model on all the test time series are illustrated in Fig.5. These results show that the proposed prediction model performs well in general. The results of the approach based on Mixture of ANFIS [12] are better for various host traces. Therefore, it can be concluded that our model gives a good prediction on most of the host's time series and outperforms then the models reported in [10,12].

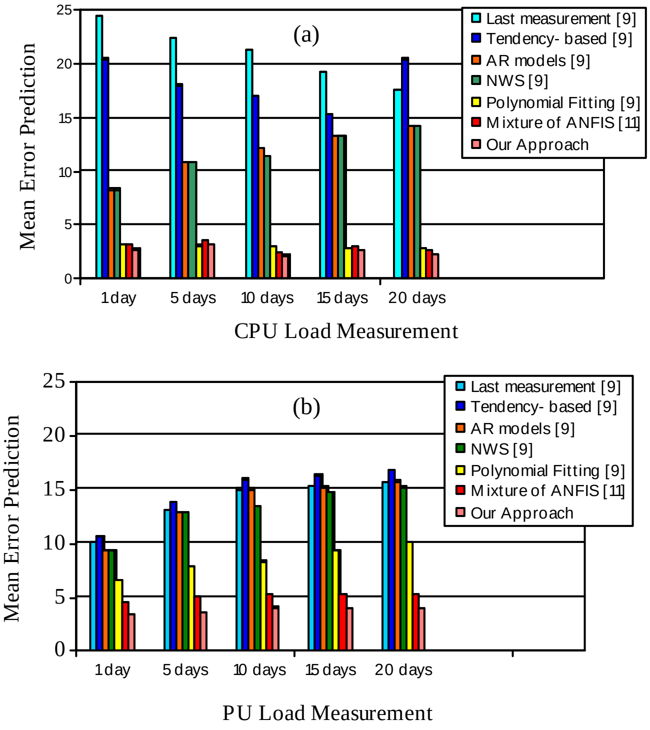


Figure 5. Mean error prediction of several models

V. CONCLUSION AND FUTURE WORK

Performance prediction is set to play a significant role in the resource management and distributed systems. Clouds computing are designed to provide services to external users, providers need to be compensated for sharing their resources and capabilities. The contribution of this paper is a new modelling approach to predict CPU load future value in distributed computing. This approach employs subtractive clustering technique to classify the CPU traces into similarity component group and a combination of local ANFIS. The proposed prediction model is validated and checked with a set of exhaustive experiments performed on a set of real and representative CPU load traces. In addition, we have shown that a significant reduction in prediction errors is experienced using the subtractive clustering-based ANFIS model since it always computes accurate predictions.

Predicting resource utilization is a fundamental need when running a virtualized system. It is necessary because cloud infrastructures use virtual resources on demand. As future work directions we will be building model considering virtualization and cloud environment. Furthermore, we will be developing prediction models based on monitoring metrics of application and services.

REFERENCES

[1] A. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared", in Proc. of the Grid Computing Environments Workshop, pp. 1-10, December 2008.

[2] Z. Shi, H. Huang, J. Luo, F. Lin, and H. Zhang, "Agent-based grid computing", Applied Mathematical Modelling 30, pp. 629–640, July 2006.

[3] R. Wolski and N. Spring, "The Network Weather Service: A distributed resource performance forecasting service for metacomputing", Future Generations of Computer Systems 15, pp. 757-768, October 1999.

[4] P. A.Dinda and D. R. O'Hallaron, "Host load prediction using linear models", J. Cluster Computing Volume 3, Issue 4, pp. 265–280, 2000.

[5] H. Koide, N. Yamagishi, H. Takemiya, and H. Kasahara, "Evaluation of the resource information prediction in the resource information server", IPSJ Transactions on Programming, Vol. 42, SIG3(PRO10), pp. 65–73, 2001.

[6] P. A. Dinda and D. R. O'Hallaron, "An Evaluation of Linear Models for Host Load Prediction", the Eighth IEEE International Symposium on High Performance Distributed Computing, pp. 87–96, August 1999.

[7] J. Huo, L. Liu, L. Liu, Y. Yang, and L. Li "Selection of the Order of Autoregressive Models for host Load Prediction in Grid", Eighth International Conference on Software Engineering, Parallel/Distributed Computing, IEEE, pp. 516-521, July 2007.

[8] L. Yang, I. Foster, and J.M. Schopf, "Homeostatic and tendency based CPU load predictions", Proc. 17th Int'l Parallel and Distributed Processing Symp, pp.42-50, April 2003.

[9] J. Liang, K. Nahrstedt, and Y. Zhou, "Adaptive multi-resource prediction in distributed resource sharing environment", In: IEEE International Symposium on Cluster Computing and the Grid, pp. 293–300, April 2004.

[10] Y. Zhang, W. Sun, and Y. Inoguchi, "CPU Load Predictions on the Computational Grid", IEICE Trans. Inf. & Syst., Vol. E90–D, No.1, pp. 40- 47, January 2007.

[11] K. Beghdad-Bey, F. Benhammadi, Z. Guessoum, and A. Mokhtari, "CPU Load Prediction Using Neuro-Fuzzy and Bayesian Inferences", Neurocomputing journal 74, pp. 1606-1616, May 2011.

[12] K. Beghdad-Bey, F. Benhammadi, A. Mokhtari, and Z. Guessoum, "Mixture of ANFIS systems for CPU load prediction in metacomputing environment", Future Generation Computer Systems 26, pp. 1003-1011, July 2010.

[13] X. Liu, Z. Ni, D. Yuan, Z. Wu, Y. Jiang, J. Chen, and Y. Yang, "A novel statistical time-series pattern based interval forecasting strategy for activity duration in workflow system", Journal of systems and software 84, pp. 354-376, March 2001.

[14] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation", Journal of Intelligent & Fuzzy Systems, Vol. 2, No. 3, pp. 267–278, September 1994.

[15] R. Jang, "ANFIS: Adaptive network-based fuzzy inference system", IEEE Transactions on Systems, Man and Cybernetics, 23 (3), pp. 665-685, June 1993.

[16] L. Yang, J. M. Schopf, and I. Foster, “Conservative scheduling: Using predicted variance to improve scheduling decisions in dynamic environments”, In Proc. Supercomputing *2003*, vol. 11, pp 15–11, April 2003.

[17] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its application to modeling and control”, IEEE T. Syst. Man Cyb., SMC-15(1), pp. 116–132, January 1985.

[18] N. Diamantidis, D. Karlis, and EA Giakoumakis, “Unsupervised stratification of cross-validation for accuracy estimation”, Artificial Intelligence 116, pp. 1-16, January 2000.

[19] F. Borko and E. Armando, “Handbook of Cloud Computing”, Springer Book, 2010.

[20] http://www.cs.cmu.edu/~pdinda/LoadTraces [April 2013].

[21] http://people.cs.uchicago.edu/˜lyang/Load [April 2013].