Journal of Physics: Conference Series **664** (2015) 052024 doi:10.1088/1742-6596/664/5/052024

Automation of Large-scale Computer Cluster Monitoring Information Analysis

Erekle Magradze[1], Jordi Nadal[1], Arnulf Quadt[1], Gen Kawamura[1], Haykuhi Musheghyan[1]

[1] II. Physikalisches Institut, Georg-August-Universität Göttingen, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany

E-mail: erekle.magradze@phys.uni-goettingen.de

Abstract. High-throughput computing platforms consist of a complex infrastructure and provide a number of services apt to failures. To mitigate the impact of failures on the quality of the provided services, a constant monitoring and in time reaction is required, which is impossible without automation of the system administration processes. This paper introduces a way of automation of the process of monitoring information analysis to provide the long and short term predictions of the service response time (SRT) for a mass storage and batch systems and to identify the status of a service at a given time. The approach for the SRT predictions is based on Adaptive Neuro Fuzzy Inference System (ANFIS). An evaluation of the approaches is performed on real monitoring data from the WLCG Tier 2 center GoeGrid. Ten fold cross validation results demonstrate high efficiency of both approaches in comparison to known methods.

1. Introduction

High throughput computing platforms as well as other large scale computing facilities provide a number of services mainly to access, store and process the data of a particular scientific or commercial importance. The demand on these services is growing alongside with the development of technologies and growing importance of the interconnected digital systems.

Most of the services are provided via the standard, or in rare cases by means of specific protocols. The service layer of the computing facility forms the front-end for the consumers who expect to have high Quality of Services (QoS) on a constant basis. Provisioning of high QoS is a big challenge and up to now there is no solid and fully verified general solution to it.

Behind the front-end there is a complex, heterogeneous variety of hardware and software systems, with hundreds of tuning parameters, system fail-over implementations, monitoring tools and the personnel. However, systems and services are apt to failures, which leaves a big space for research in the area of computing facilities performance and management improvement.

The main objective of this work is to provide the approach for the automation of the information extraction from the monitoring data. In particular the goal is to identify the service failure patterns based on the monitoring metrics and also to predict its degradation using one of its main characterizing metrics, e.g. the Service Response Time (SRT). The paper is organized as follows. In the next section the method and the case study for the service status identification is described. Section three contains the techniques for the SRT prediction as well as the results of the conducted case studies. The final section concludes and summarize the work.

2. The Service Status Identification

In time service failure detection is very important for the stable functioning of a computing infrastructure, even if the service is for an internal usage, guaranteeing availability and serviceability of the computing infrastructure front-end. The main indication of the failed service is a specific message providing the information about its malfunction, or the delayed response to a valid request. The applications used in the large scale computing systems record the problematic events in a corresponding log file. Tools like logstash [1] together with kibana [2] and also hadoop [3] based technologies are employed actively for the log files and monitoring data analysis and visualisation. The main limitation of these tools are the requirement of the large amount of log data and frequently clear specification of the failure messages. Also it is rarely possible to identify the root cause of the service failure and degradation in terms of the trivial monitoring metrics such as, "CPU Load", "Memory Usage", number of established network connections etc.

The approach described in this section does not require large amount of monitoring data. It provides the failure root cause analysis by ranking of the trivial monitoring metrics according to their influence level on a service functionality. Based on the approach it is possible to automatically define the general model of the service availability using the monitoring metrics and log information. The general model for the service status identification makes it possible to quickly detect the service degradation or failure.

The case study for the approach validation was performed on the GoeGrid storage system. The storage infrastructure of the GoeGrid computing facility consists of multiple, heterogeneous hardware and virtual servers, which are deployed as one monolithic system based on the dCache [4] storage management software.

Although, the service under observation ("Pnfs Manager") is not part of the front-end services exposed to the users, it manages the namespace information about all the data objects stored at the GoeGrid hard drive systems, hence it is a critical component for the entire storage software functionality. The Pnfs Manager stores and provides, changes and manages all the namespace information in a dedicated database, which is called "Chimera". The Chimera component of the Pnfs Manager has a big number of tuning parameters and its proper functionality requires a non-negligible amount of memory and cpu resources. During the period of the case study, 15 disabilities of the Pnfs Manager service were observed that caused malfunction of the entire storage system. The server, hosting the service, was under the monitoring of the ganglia monitoring tool [5]. As a source containing the service failure information with the time stamps the corresponding log file has been used. According to the monitoring data aggregation and preprocessing workflow, more than 2600 datasets were collected and synchronized. Each dataset represented a vector with the 14 variables, where the initial 13 corresponded to the monitoring metrics from the ganglia and the last was a binary value of 0 or 1, indicating disability or availability of the service correspondingly.

As a technique for the Pnfs Manager service failure root cause analysis the linear Support Vector Machine (SVM) [6] was employed. The SVM method is actively used for the feature selection and ranking in multidimensional space, especially if there is mapping of multiple components to a binary variable. The implementation of the SVM method in the Rapidminer tool [7] assign weights to each of the attributes of the dataset except the last one - the service status identifier element. Thus it is possible to rank the monitoring metrics according to their correlation strength to the service status. Ranking information is important for the infrastructure administrators to detect the bottlenecks in the system i.e. it makes possible to perform the service failure root cause analysis.

Alongside with the root cause detection of service disability the corresponding model for the service state identification has been developed. The approach for this development is based on a Fuzzy Inference System (FIS) [8], which relies on a Fuzzy Sets Theory [9]. Application of the

Fuzzy Sets and in particular FIS in the process of the complex, non-linear system modelling has an advantage of taking into account not only the quantitative aspects of statistical data describing the system, but also the qualitative characteristics.
Two key components, the fuzzy value and the inference rules, define the FIS. The fuzzy value is a pair of crisp number and corresponding Membership Function (MF). The MF allows to map the particular measurement represented in a crisp number to the certain qualitative characteristics e.g. term used in everyday life ("Tall", "Small", "Too big", "Important", "Critical" etc.). The inference rule also consist of two parts, the "if" or the premise and "then" the consequent parts. The premise part checks the meaning of the membership function for the particular crisp input variable i.e. identifies at which level the particular value belongs to the predefined qualitative term. The number of input variables and qualitative terms for each input variable needs to be defined in advance in order to identify the structure of the model. In case of the non-linear systems modelling, such as the Pnfs Manager service state identification it is required to have at least two input variables. In case of more than one input variable in the premise part of the rule the logical "and" or "or" operation acts as a connection. Analytically, the "and" operation can be any function satisfying the *t-norm* [10] conditions, while the "or" operation should correspond to the *t-conorm* [10] requirements. For the consequent part of the inference rule the corresponding membership functions should be defined in order to map the outcome of the premise part to the particular qualitative term characterizing a certain state of the entire system under the observation. According to the type of analytical operations for the premise part outcome identification, as well as the type and structure of the membership functions of the consequent part, one can distinguish three types of the Fuzzy Inference Systems [11]. The general structure of the FIS (see Fig. 1) for all three types is common and requires involvement of an expert in order to generate the rules and define the MF's structure. Due to the goal of the research, to decrease the human factor involvement in the computing infrastructure management automation process and in particular the service availability modelling, the general attention was concentrated on Takagi-Sugeno type of FIS [12].

The Takagi-Sugeno Fuzzy Inference System was integrated with the Feed Forward Artificial

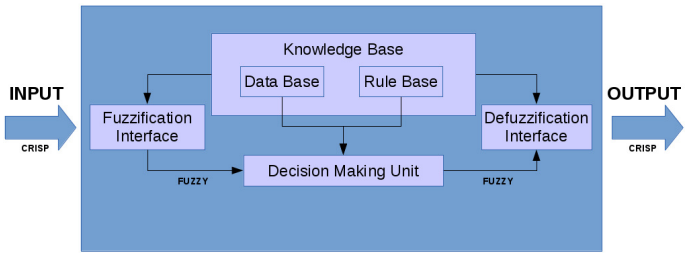

Figure 1: The general structure of the Fuzzy Inference System (FIS).

Neural Networks (ANN) [13] and derived as the Adaptive-Network-Based Fuzzy Inference System (ANFIS). The ANFIS is a generic approach to automatically create the inference rules, define the parameters of the membership functions and derive the model identifying the state of the process of interest or even predicting its further behaviour. Only few components need to be defined in advance - the membership functions and the number input variables. The model identification is performed by the ANN training process with the backpropagation learning algorithm [14], that completely excludes the need for an expert involvement and hence reduces the subjective information influence on the modelling process.
Due to the structure of the Takagi-Sugeno FIS, the "and" operation was employed in the premise part of the inference rule. The number of functions are satisfying the *t-norm* conditions but due to the complex non-linear process modelling in terms of the Pnfs Manager state identification

the "Gaussian" membership function, $\mu(x;\sigma,c) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$, was considered to be appropriate. The backpropagation learning algorithm automatically identifies the σ and c parameters that characterize the qualitative aspect of the measured monitoring metric x being the component of the input dataset vector. Together with the membership function the number and the structure of the input data vector and its elements needs to be defined. For this purpose, the feature selection results derived from the SVM algorithm were employed (see Fig. 2). According to the results, the two monitoring metrics "load fifteen" and "tcp established" have the highest distinguished weights i.e. have the largest influence on the Pnfs Manager service state. In this case study, the "load fifteen" monitoring component shows the fifteen minutes average load of the system, which hosts the target service, while the "tcp established" shows the average number of established TCP connections in six minutes period of time. These components were selected as an input data to ANFIS. The output of the system was the binary value, indicating the target service disability "0" or availability "1". The entire monitoring data - more than 2600 datasets

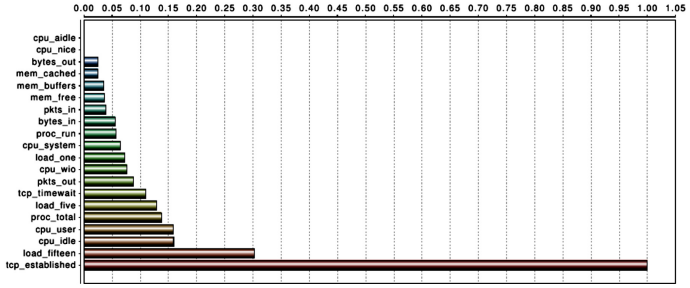

Figure 2: Ranked components (on the vertical axes) of the input dataset vector according to the weights (horizontal axes) defined by the SVM algorithm.

were split in two major parts for training and checking. The ANFIS training procedure, for 1000 datasets, took 40 iterations and less than a minute of time. The accuracy of the approach was checked according to the contingency table method described in [15]. The final result (see Table 1) of 99% accuracy in terms of the service status identification, proved the validity and high efficiency of the chosen method for the adequate service state identification.

Metric	**Value**	**Details**
Precision	99%	Positively Identified Values
Recall	99%	True-Positive Rate
Specificity	88%	True-Negative Rate
Accuracy	99%	-

Table 1: The contingency table for the Pnfs Manager service status identification using ANFIS.

3. The Service State Forecasting

One aspect of the monitoring data analysis is to identify the service status and in case of its failure provide the root cause of the problem. This aspect was covered in the previous section. Another outcome, which can be derived from the processing of the monitoring information is the prediction of the service degradation.

This section is dedicated to the description of the approach for the prediction of the Service Response Time (SRT), which is the key feature for the service availability. The approach is

based on the time series analysis techniques [16], which is used for modelling of dynamic processes evolution in time. The time series information is the sequence of a single or multidimensional data sorted by date in the ascending or descending order. In general, data should contain the measurements of the process characteristics. The advantage of this analysis method is possibility to apply different techniques for the information extraction from the sequential data. Similarly to the previous section, the core technique here is ANFIS. However, in this case the input to ANFIS is a data vector of the same monitoring metrics (SRT) measured in different periods of time and the output is the continuous values of SRT unlike to the technique discussed in the section two.

The key importance in efficient time series analysis for the service state prediction, has the adequate identification of the monitoring metrics, which characterize the service availability or disability. Here it is the Service Response Time, which is the fundamental metrics for the efficient service delivery measurement in the systems with Service Oriented Architecture (SOA), e.g. WLCG sites. The SRT is the period of time required for getting the response from a service, after the request is sent. Thus, efficient service delivery assumes that in spite of the number of requests, i.e. the load on the system or infrastructure, the response time should be stably short. However, this condition is not always achieved and if the SRT is beyond the certain threshold, the violation of the Service Level Agreement (SLA) occurs. The SLA, in addition to the number of conditions, contains the information about the failure thresholds in terms of SRT (in sec. or ms.). Thus, the SRT thresholds for the service failure or availability identification is specific for each particular service provider. Therefore, prediction of such a critical metrics as SRT allows to identify the SLA violation risks in advance and plan the corresponding proactive management actions for avoiding the foreseen problems. Another advantage of using the SRT as an object for the time series analysis is its complex dependency on the internal and external factors affecting the service host, i.e. the hardware server and its environment. Hence, the SRT reflects not only the service performance but it also shows how efficiently the infrastructure is configured and is able to handle the high load.

Thus, the objective of this section is to provide the time series and ANFIS based approach for the SRT prediction in the range of minutes to hours. Despite to the previous section SRT is not a binary value and for the evaluation of the approach efficiency the Regression Error Characteristics (REC) curves [17] and the Area Over Curve (AOC) metrics is used. The case studies applying the technique, covers two key services running at the Tier 2 center GoeGrid. The dCap service [4] and the "pbs server", which corresponds to the server daemon of the TORQUE batch system [18].

We decided to concentrate on these services because they are actively used and their malfunctioning is critical for an entire system. In particular, the dCap service is one of the most actively used methods for fetching and returning the data from and to the storage system for its analysis. Of course this service is used for the WLCG sites, which run the dCache as the storage system. If the response time of the dCap service is high, the corresponding computational job for the data analysis may fail, because of the large CPU wall time [19]. Another problem possibly caused by the long dCap SRT is the small number of running computational jobs, which might not correspond to the available resources, hence the risk of the waste of provided computational power is high. The long SRT of the "pbs server" daemon can also lead to the miss-usage of the computing resources. Because of its key importance for the batch system functionality, the long SRT of the "pbs server" increases the SRT of the entire batch system performance. In particular, if the "pbs server" is responding with delays the time for acceptance, scheduling and managing of the computational jobs will increase, which can lead to the drop-down of the reliability and serviceability of the entire infrastructure. Thus, both selected services for the case studies are critical in terms of the efficient usage of the available computational resources and provision of highly reliable and serviceable computing platform.

The time series analysis is impossible without the historical data containing the general patterns of the service dynamics. Let us denote the SRT historical information, as D_i, i.e. it is the sequential measurements of the SRT for a service. The first element of the SRT time series data corresponds to the time moment t_1 and the final value is measured at the moment t_i. The step size between each SRT measurement in our case is six minutes according to the best practice from the ganglia monitoring tool, i.e. $t_{j+1} - t_j = p$, $\forall j, 1 \leq j \leq i-1$, where $p = 6$ min. Under the assumption that the D_i is available, the aim of the time series prediction model, M, is to assess the d_{i+n}, where $n \times p$ represents the prediction time window, i.e. the time series problem can be formulated as follows $d_{i+n} = M(D_i) \pm e_{i,n}$, where $e_{i,n}$ is the prediction error, which increases with n.
Similarly to the service state identification, for the SRT prediction we employ ANFIS with Gaussian membership functions but with the specifically structured input data $l(\frac{d_i - 2d_{i-1} + d_{i-2}}{4}, \frac{d_i - d_{i-1}}{2}, d_i)$, where d_i corresponds to the SRT measurement at time t_i and the rest of the components of the l vector are defined accordingly. The advantage of the specific structure of the l vector was proved by the comparison of ANFIS prediction accuracy on $l^{'}(d_{i-2}, d_{i-1}, d_i)$ (see Fig. 3).
Besides the input data structure, the important parameter for the ANFIS, is the membership

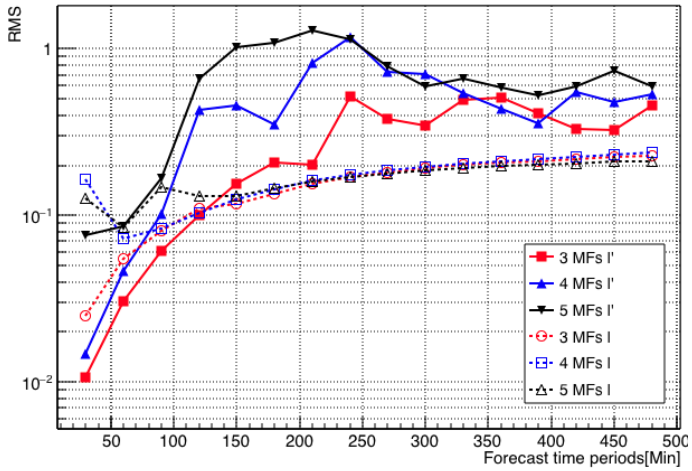

Figure 3: Tenfold cross validation results of Root Mean Squared (RMS) Error of the ANFIS prediction with l and $l^{'}$ vectors, in case of different number of MFs for each input component.

function (MF) as it is already mentioned in the previous section. The number of MFs per input variable identify the amount of parameters that need to be calculated, hence it needs to be identified the correct trade off between the model accuracy and the computation time for the parameters calculation. Therefore, in parallel we have conducted small analyses for the identification of the best number of MFs in each case studies. The number of MFs for both case studies varies from three to five. Less than three membership functions do not provide the adequate accuracy and even ANFIS is facing the convergence problem in the training phase, while more than five MFs require long training period, which makes the entire prediction procedure useless. In addition to the MF related analysis another time series prediction technique, the Nonlinear Auto Regressive Neural NETwork (NARNET) [20] was compared to ANFIS.
According to the results (see Fig. 4), for the dCap case study, the ANFIS with five MFs shows the best results, while for the "pbs server" case study the best results are achieved with the four MFs. In both studies the ANFIS has demonstrated better prediction accuracy than NARNET in terms of Mean Absolute Percentage Error (MAPE) (see Table 2).

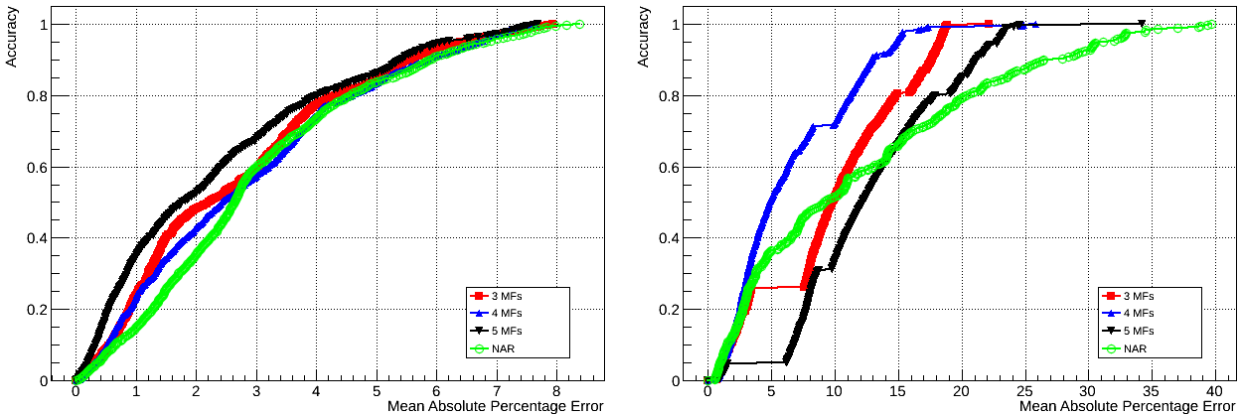

Figure 4: Mean Absolute Percentage Error (MAPE) for "dCap" (left) and "pbs server" (right) SRT prediction up to eight hours.

Method	**100%-MAPE**	
	dCap	PBS Server
ANFIS 3MF	97.3%	90.4%
ANFIS 4MF	97.2%	**93.6%**
ANFIS 5MF	**97.7%**	84.6%
NARNET	97.0%	88.2%

Table 2: Mean Absolute Percentage Error (MAPE) for "dCap" and "pbs server" services.

4. Summary and Conclusions

Nowadays it is impossible to imagine the process of administration and management of the large scale computing facilities, without the automation and configuration management tools like CFEngine [21], Puppet [22], Chef [23] etc. Actions, notifications and changes issued by these tools are based on the policies and instructions defined by the system administrators and IT professionals. Efficiency of such automation and configuration management applications, depends on the proper statement of the required actions. In particular the rules for the correct inference making should be provided based on the adequate system related knowledge. The described approaches for the service failure and its root cause detection as well as for the SRT prediction, can be used for the automated identification of the knowledge base, with the focus on the service related problems detection. Automated service failure detection and prediction approach allows to describe the policies and inference rules objectively. Thus, automatically generated, system related knowledge in the form of inference rules and policies can be provided to the management and configuration automation tools. Therefore it is possible to decrease the number of improper or inadequate actions caused by the biased knowledge of a particular system administrator or IT specialist. Thus there is a room for service availability and reliability improvement.

The entire implementation of the approach is based on the RapidMiner (for SVM) and MatLab Fuzzy Logic Toolbox [24] packages (for ANFIS) wrapped by the Python [25] script. The only step which requires human interaction is required, for the described approaches, is the provisioning of the monitoring data in a suitable, CSV format. Hence the ANFIS based approach can be considered as an automatic, service related knowledge extraction tool from the available monitoring information. The integration of the automated, ANFIS based approach, in the

action taking and configuration management systems can be considered as a next step for the further implementations, which can be considered to be successful due to the similarities in the SOA based systems.

References

[1] logstash tool `http://logstash.net/docs/1.4.2`
[2] Kibana - Data Visualisation Tool `http://www.elastic.co/products/kibana`
[3] Apache Hadoop `http://hadoop.apache.org/`
[4] The dCache Storage Management System `http://www.dcache.org/manuals/index.shtml`
[5] Massie M, Li B, Nicholes B, Vuksan V, Alexander R, Buchbinder J, Costa F, Dean A, Josephsen D, Phaal P *et al.* 2012 *Monitoring with Ganglia* (" O'Reilly Media, Inc.")
[6] Hsu C W, Chang C C, Lin C J *et al.* 2003 A practical guide to support vector classification
[7] Jungermann F 2009 *Proceedings of the GSCL SymposiumSprachtechnologie und eHumanities* (Citeseer) pp 50–61
[8] Lata S and Ayyub M 2014 *International Journal of Applied Engineering Research* **9** 805–813
[9] Dubois D J 1980 *Fuzzy sets and systems: theory and applications* vol 144 (Academic press)
[10] Gupta M and Qi J 1991 *Fuzzy sets and systems* **40** 431–450
[11] Lee C C 1990 *Systems, Man and Cybernetics, IEEE Transactions on* **20** 419–435
[12] Lohani A, Goel N and Bhatia K 2006 *Journal of Hydrology* **331** 146–160
[13] Jang J S 1993 *Systems, Man and Cybernetics, IEEE Transactions on* **23** 665–685
[14] Aizenberg I and Moraga C 2007 *Soft Computing* **11** 169–183
[15] Salfner F, Lenk M and Malek M 2010 *ACM Computing Surveys (CSUR)* **42** 10
[16] Madsen H 2007 *Time series analysis* (CRC Press)
[17] BIJ J B, EDU R and BENNEK K P B 2003 *Twentieth International Conference on Machine Learning (ICML-2003). Washington, DC*
[18] TORQUE Batch System `http://www.adaptivecomputing.com/products/open-source/torque/`
[19] Jones M B, Roşu D and Roşu M C 1997 *CPU reservations and time constraints: Efficient, predictable scheduling of independent activities* vol 5-31 (ACM)
[20] Chow T and Leung C 1996 *Generation, Transmission and Distribution, IEE Proceedings-* vol 5-143 (IET) pp 500–506
[21] Burgess M *et al.* 1995 *in USENIX Computing systems, Vol* (Citeseer)
[22] Turnbull J 2008 *Pulling strings with puppet: configuration management made easy* (Springer)
[23] Nelson-Smith S 2013 *Test-Driven Infrastructure with Chef: Bring Behavior-Driven Development to Infrastructure as Code* (" O'Reilly Media, Inc.")
[24] Sivanandam S, Sumathi S, Deepa S *et al.* 2007 *Introduction to fuzzy logic using MATLAB* vol 1 (Springer)
[25] Van Rossum G *et al.* 2007 *USENIX Annual Technical Conference* vol 41