Sensor Selection for IT Infrastructure Monitoring

Gergely János Paljak, Imre Kocsis, Zoltán Égel, Dániel Tóth, András Pataricza
Department of Measurement and Information Systems
Budapest University of Technology and Economics
Magyar tudósok krt. 2, H-1117 Budapest, Hungary
paljakg@sauron.inf.mit.bme.hu, {ikocsis,egel,dtoth,pataric}@mit.bme.hu

Abstract. Supervisory control is the main means to assure a high level performance and availability of large IT infrastructures. Applied control theory is used in physical and virtualization based clustering, autonomic-, self-healing and cloud computing, but similar problems arise in any distributed environment.

The selection of a compact, but sufficiently characteristic set of control variables is one of the core problems both for design and run-time complexity. Most results in the literature are based on a single algorithm for variable selection, but our measurements indicate that no single algorithm can generate faithful estimates for all the different operational domains.

We propose to use a combination of different model extraction techniques on benchmark-like data logs. The main advantages of this multi-paradigm approach are twofold: it provides good parameter estimators for predictive control in a simple way; and supports the identification of the actual operational domain facilitating context-aware adaptive control, diagnostics and repair.

Keywords. Autonomic computing, control theory, signal processing, artificial intelligence, benchmarking, performance and performability control

1. Introduction

Modern system management aims at guaranteeing a high service level in terms of all operational aspects, primarily of performance and availability by applying a feedback control loop scheme. Feedback control in autonomic computing continuously monitors the service level and upon an unwanted deviance triggers optimization/health maintenance actions according to a predefined control policy.

Trustworthy autonomic performability management necessitates establishing a formal relationship between certain *monitored* and *influenced* attributes of a system even for rough granular control; for fine granular approaches utilizing classic control theory it is even more so (for some examples, see [1]). However, when first principles based modeling is infeasible – what is rather the rule, than the exception for IT systems in general – a prerequisite of system identification is establishing the set of underlying attributes for the model. While this naturally occuring task of autonomic performability control design seems to be quite neglected, it is by no means a trivial one. This paper proposes an AI inspired approach to address this requirement.

A monitored configuration consists of the application and its runtime platform instrumented with additional sensor and actuator agents *Sensors* report on the run and health state of the application and its run time platform The monitoring node processes these sensor values and initiates active diagnostic probing, repair actions, like dynamic allocation or reallocation of resources or even a reconfiguration of the application deployment in the system to be executed by actuator agents.

The monitoring scheme covers on the one hand functional and extra-functional discrete state change events (like beginning or termination of a job or detection of an error manifestation, respectively) and on the other hand platform, application, component and system service level quantitative performance and dependability measures.

While this overall control algorithm problem appears in a general form in all large scale systems and the principle of our approach remains valid for this more general context, we confine our subsequent discussion to datacenter-like infrastructures (and cloud environments).

Large IT infrastructures and even the monitoring functions are large-scale distributed systems. The objects of the control in server farms and clouds, the applications, their deployment with the monitoring and control agents and the local control functions in the application nodes are all distributed. The control functionality has typically a hierarchical structure composed of domain and system controller nodes processing the raw sensor data and preprocessed data from the subordinate monitor nodes.

A *monitoring and supervisory control node*

- collects the raw information directly incoming from the sensors and the potential preprocessed data from subordinate monitoring nodes,
- *correlates the events*, estimates the metrics and
- identifies (and possibly predicts) the *situation*
- compares them with those in an anticipated use case (e.g. prediction of a potential overload of a particular resource or diagnoses fault in a component) based on estimates
- decides on the reactions to be executed by the actuators according to a predefined *control policy* usually formulated in a rule-based manner.

  The candidate actions triggered by the actuator agent deployed into the monitored infrastructure consist of *tuning the resources* available for the individual application tasks (priority reassignment in multitasking, modification of the resource arbitration in virtualization) or *structural reconfiguration* (dropping non-critical tasks, task replication and/or migration).

The prevailing industrial approach is still dominated by the former age of manual control for configuring system supervision. It deploys and activates a very wide set of sensor agents onto the platform under control, as the operator may select the relevant ones and simply ignore all the others. On the processing side, some application and infrastructure specific, quite ad-hoc thresholds and simple empirical rules are provided, aiming primarily at defining “normal” operational intervals on a per metric basis and raising an “out of range” type of alarm in the case of a deviance.

However, the automated identification (prediction), diagnosis and reaction on problems needs precisely formulated rules of metric aggregation and correlation.

Over-instrumentation, i.e. monitoring too many metrics of a system poses significant problems, as a large number of threshold estimation, quantification, aggregation, situation identification and diagnostic rules exclude reliable manual design and maintenance, especially in evolving applications. On the other hand monitoring too many metrics also causes unnecessary performance overhead on the monitored systems, and data collection nodes especially in case of historic data collection.

Under-instrumentation, i.e. the improper reduction of the set of monitored metrics, on the other hand can significantly compromise the capabilities of supervision, manifesting in large reaction times to workload changes, significantly reduced availability due to late error detection and diagnosis.

Heuristic manual control based monitoring does not scale well for large, heterogeneous IT systems from many aspects; as emphasized by industry initiatives as IBM Autonomic Computing [2] or the evolving “cloud computing”. Such systems increasingly employ almost fully automated structural reconfiguration and other adaptive techniques borrowed from control theory to guarantee the performance and dependability of services.

Consequently, a theoretically well-founded approach is needed for selecting a minimal or sufficiently small set of metrics and associated points of measurement out of the technically measurable ones, which characterizes the system “adequately”. Selecting such a metric set is certainly only the precursor to setting up e.g. diagnostic rules. More precisely, given a *control objective metric* (e.g. throughput of a particular service as an influenced attribute), we seek a corresponding, near-minimal subset of metrics and an appropriate approximation function delivering enough information to assure the fulfillment of the control objective.

We illustrate the core problem and our approach by a simple example of curve approximation:

Given a series of observations y co-recorded with all the parameters potentially forming its cause, we have to select the principal factors in an estimator of this objective function sufficiently closely matching it.

After selecting the set of independent variables for the estimator, we have to select a single function or a family of functions for a best fit estimator (for the sake of simplicity we assume that a single independent variable is sufficient to create a faithful estimator of the observation series in Figure 1.). Here the “best fit” is measured by means of an approximation error metrics quantifying the deviance of the estimator from the observation.

Given the set of independent variables, one option is to use a single function for curve fitting (curve 1 and 2 in Figure 1.) An additional degree of freedom is given when a family of functions is offered for fitting instead of a single one. Here the accuracy of the approximation can be further improved by piecewise fitting, i.e. by selecting a particular function for a given interval resulting in the best match (the partially linear/non-linear/linear curve 3 in Figure 1.).

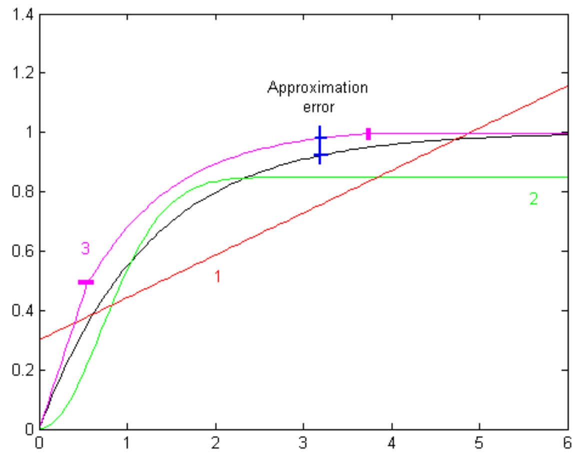

Figure 1. Curve fitting examples with respect to the objective function in black

The online approximation delivers as byproduct the identification of the best matching estimator function in addition to the expected value of the objective variable, as well.

As the splitting of the domain of the independent variable corresponds to the different operation domains, the information on the best fitting function identifies the actual state at a rough granular level at the resolution of the operational domain. An adaptive control policy may fine granular evaluation of the causal variables for diagnostics after the appearance of degradation, as indicated by the best fit of the corresponding non-linear approximation function of a phenomenological variable.

This paper proposes to combine linear estimators with the powerful minimum-Redundancy-Maximum-Relevance (mRMR) nonlinear feature selection scheme for the selection of such a small set of metrics that still adequately characterizes an objective metric.

Measurements on a testbed implementing an industrial OLTP performance benchmark equipped with a fully instrumented, commercial enterprise system monitoring product were used for the experimental validation of the approach.

The remainder of this paper is organized as follows. We first briefly describe prior research underlying our experiment and introduce existing approaches and then discuss monitoring instrumentation issues. Section 4 describes the test-bed, subject of our investigations that are described in Section 5. Section 6 highlights the most important results which we then evaluate and conclude.

The experiment was partially funded by the AMBER EU FP7 project, “Assessing, Measuring and Benchmarking Resilience” [3].

2. Related Work

Evaluating large amounts of measured metrics by statistical methods and methods from artificial intelligence can be effectively utilized to improve enterprise systems’ dependability by allowing fault detection [4, 5] and the forecasting of the system’s behavior [6, 7]. Unfortunately, the dimensional problem of such approaches has not been sufficiently addressed by the community. Either simple, linear methods are used either in forward selection or backward elimination fashion or a wrapper approach [8] is utilized that requires the presence of a learning algorithm and thus the speed of the process is significantly reduced while results are dependent on their further usage. Munawar et al. [9] are suggesting a Mutual Information based method and show that non-linear correlations exist between metrics and those can be effectively utilized to enhance fault diagnosis.

Our approach applies all the measures offered by the large set of sensors in industrial tools (e.g. IBM Tivoli Monitoring) in a benchmark-like experiment. The set of the measures to be monitored in the operational environment will be reduced by intelligent log analysis to those few ones which sufficiently characterize the system by themselves for fast reaction or early error detection. We use the systematic, well-tunable mRMR algorithm for variable selection. It is also based on mutual information and has been shown to scale well for large problem spaces [10, 11].

There are numerous approaches to utilize the data obtained this way. [12] shows an entropy based – like mutual information in case of mRMR – fault detection method that is dependent on the window size and thus may not always be sufficient for early fault-detection. [13] presents collected data from a web server under overload and builds time series ARMA (autoregressive moving average) models to detect aging, and estimate resource exhaustion times. [14] presents a way of on-line discovery of quantitative models, based on linear least-squares regression and shows its application for a database system. However, no established investigation is known that would validate these approaches across different operational domains and evaluate their performance.

3. Instrumentation Support of Metric Selection

We apply the mRMR feature selection scheme for multi-tiered online transaction processing systems. All major components of the system (operating system instances, middleware, server software, network interfaces and components) are instrumented with sensors in the initial data log acquisition phase.

Commercial off-the-shelf monitoring products offer a large selection of candidate *sensor agents* out-of-the-box for each major component type. IT infrastructure components and services have typically the option to be associated with a wide set of *metrics* and emitted *events* delivered in a raw form by the *instrumentation* of the controlled node.

Local metrics measured by the sensor agents and derived metrics used in the control nodes can be grouped jointly, independently of their source into two main classes:

- *Phenomenological metrics* deliver the measured or derived results in an implementation independent form in the terms of some standard (logic) units (for instance the average transaction time in a database).
  Such metrics are typically used to characterize the extra-functional characteristics of the services delivered by the individual software components and applications, computing nodes and the entire system.
  As the objective of the feedback control is keeping the overall service level characteristics within the range allowed by the specification (frequently expressed in the form of an SLA), these are the primary control variables at the topmost level of control.
- *Causal metrics* are able to “look inside” the component internals (for instance buffer pool attributes for the version x of type y of a database). The main advantage of their use is the high level of observability and controllability provided at a price of high maintenance and version control costs originating in the strong implementation platform dependence.

Their typical use is on the one hand the reduction of error latency in critical applications, as monitoring and checking the internal state may detect an error in a component prior of the degradation of the services delivered by it; on the other hand they are used in fine granular diagnostics.

The components of the target system are treated as providing either a “resource service” or a “request-response” service for other components. These two service types have some associated metrics that are meaningful in all cases, regardless of the specific service provided or its implementation. The former category is typically associated with quantitative metrics that are utilization aggregates, originating from the behavior of multiple clients. The latter case can be characterized by workload (faultload) and output performability metrics.

As a rule of thumb, all these “implementation independent” metrics (arguably of a phenomenological nature) should be recorded for each component. This guideline is to ensure that there is a uniform set of metrics that applies for all components, comprising at least a black-box characterization of all the individual system components.

Additionally, in most cases the COTS instrumentation of the components offers insight into the internals of the component implementation supporting a much earlier problem detection and actual fault diagnosis, like their manifestation in the services. However, our approach should not solely rely on these, as the behavioral coverage they provide is quite hard to reliably assess.

As part of the necessary system instrumentation, the examined objective metrics forming the core factors of the service level agreement offered to the end user are also to be chosen and their measurement implemented. For OLTP systems, service response time and throughput are the most natural choices for selecting sensors for managing service performability, which is our current focus.

4. Experimental Setup

Our experimental testbed is a small, three-tier virtualized server architecture having two additional nodes: one for workload generation, the other for monitoring and processing the captured data (Figure 2.). The infrastructure contains 6 virtual servers, each of them running the CentOS 5 GNU/Linux distribution with Apache, Tomcat, MySQL and Sequoia (a database clustering middleware) installed, respectively. All servers are deployed on a single VMware ESX host.

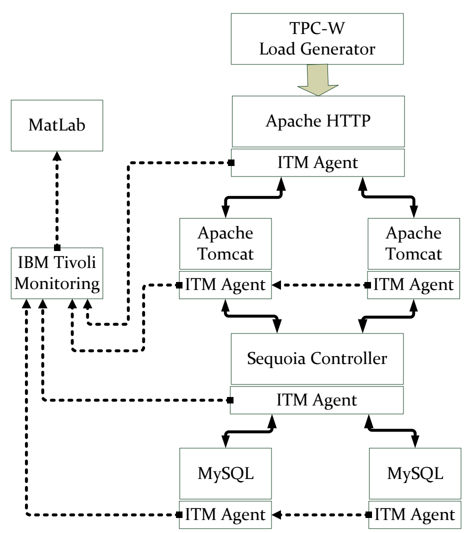

Figure 2. Architecture of the experimental setup.

The environment runs an implementation of the TPC-W standard benchmark [14]. The workload used in the experiment is the TPC-W “Shopping Mix”.

Two objective metrics were chosen: response time and throughput, using Web Interaction Response Time (WIRT) and Web Interactions Per Second (WIPS) metrics of the TPC-W specification for exact definition.

IBM Tivoli Monitoring 6.1 (ITM) is used for monitoring purposes. ITM is a centralized, agent-based monitoring solution: central monitoring server(s) collect measurement data and event notifications provided by monitoring agents running on the supervised hosts. On a single host multiple agents may be deployed, as every platform and software component covered by the product is supported by a separate agent. Numerous platforms, software components and devices that are not supported by the product out-of-the box (or by product extensions) have agents freely available on the Tivoli Open Process Automation Library (OPAL) site. Most of these utilize the Tivoli Universal Agent, a special agent type with the purpose of enabling the development of custom sensors against documented interfaces. Altogether over 1000 metrics were measured by the agents, with a sampling interval of 30 seconds.

A Java importer has been implemented for in-MATLAB execution that queries data samples from the central ITM server and transforms them to MATLAB-format time series for further processing.

5. Experimental Methodology

As our goal is early fault-detection and pro-active prevention we opted for the regression of the selected objective metrics i.e. throughput and response time. First of all, we have to reduce the number of sensors/metrics considered in order to avoid over-instrumentation and to simplify the regression problem.

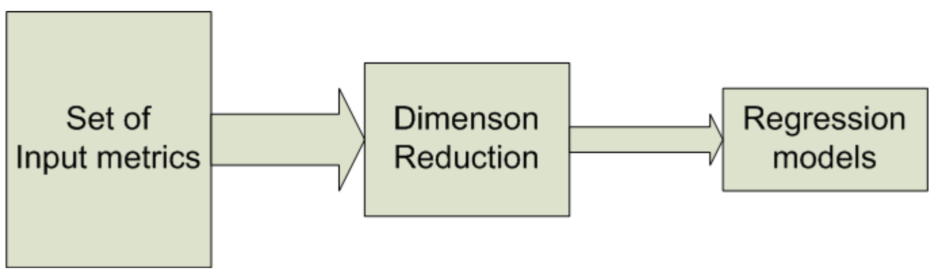

Figure 3. Experimental methodology

Dimension reduction is the generic term for methods that aim at reducing the number of considered variables in the mathematical model of a given problem. The dimension of the task at hand is the number of variables to be measured for some further action. The problem is well-known in the statistical and machine-learning communities who were the pioneers facing the problem of high-dimensional datasets. Here the impact of a given variable can frequently not be determined on sole human expertise; all and any of them can be “important” for the understanding of the examined process/system.

Dimension reduction methods are traditionally divided into two groups: feature selection- and feature extraction approaches. Feature selection aims at finding a subset of the measured variables while feature extraction is applying a projection of the multidimensional problem space into a space of fewer dimensions thus resulting in aggregate measures that did not exist in the measured environment [16, 17].

So the dimension of the problem – i.e. the number of attributes processed – shall be reduced. A selection of few attributes is required: finding those that mostly influence system-level metrics (e.g. throughput, response time) and thus enable the construction of a control algorithm with relatively unambiguous rules. As a basic approach we implemented a greedy forward selection method that uses linear regression as an evaluative measure in the incremental process. We also selected the relatively new mRMR algorithm [18], a feature-selection method to identify candidates that are likely to have influence on high-level performance metrics for its high accuracy and fast speed [19], presenting a promising approach to grab a descriptive set of metrics considering various aspects of the system.

mRMR is based on the concept of mutual information, that for two probabilistic variables x, y, can be calculated as:

$$I(x;y) = \iint p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{1}$$

In our case we want to select a set of variables, S, so that the mutual information between each element of S and the objective metric c is maximal (maximum relevance):

$$\max R(S,c), \qquad R = \frac{1}{|S|} \sum_{x_i \in S} I(x_i;c) \tag{2}$$

and the redundancy is minimal inside S, which means the mutual information between the elements is minimal:

$$\min r(S), \quad r = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i;x_j) \tag{3}$$

We intend to find those attributes that have the highest mutual information against an objective metric, and keep the mutual information low among the set of the identified attributes in order to find signs of distinct performance issues.

In practice an iterative algorithm optimizes the following condition:

$$\max_{x_j \in X - S_{m-1}} \left[I(x_j;c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j;x_i) \right] \tag{4}$$

where $m = |S|$, the size of S in the current iteration and S_{m-1} being the set of metrics selected prior to the current iteration.

So the algorithm in the first step selects the variable $x \in X$ with the greatest mutual information with respect to the objective metric. In the second step it selects the variable $y \in X - x$ with the smallest mutual information with respect to x while maximizing the

S_2 subset's mutual information according to the objective metric c. It carries on iteratively until the pre-determined number of variables is reached.

Please note that calculating the mutual information for a set of time series is a very computation intensive task. The mRMR algorithm is incremental, gradually selecting the target variables by choosing the next best fitting one for extending the variable set. This way, it is only optimal in a local sense for each iteration step but does not ensure global optimality.

As for the regression part we decided to utilize two different methods: linear regression that aims at approximating the objective metric as the weighted sum of the selected variables and two-layer feed-forward neural network that works similarly but has non-linear capabilities. Traditionally the linear regression equation is as follows:

$$Y(t) = \sum_{i=1}^{K} w_i X_i(t) + \varepsilon(t)$$

Assuming that we selected *K* variables *X* the method computes the weights *w* to calculate the objective value *Y* in the given time *t* with error *ε(t)*.In case of prediction, the right hand side is shifted back in time and thus the result is estimated based on the available values of the past i.e. using the values (t-k), k = 1..N.

6. Experimental Results

In order to gain an insight into the setups internal relations, we stressed the system with different load scenarios, including normal and extreme loads and some abrupt changes as well and then evaluated the acquired time-series with the methods introduced above.

First we examined the available features and those selected. Calculating the correlation matrix we find a lot of high coefficients, clearly confirming the base assumptions in [20] of lower dimensions. On the other hand it is also suggests that due to that and the large number of measured metrics we are unlikely to find matchings in the individual scenarios between the features selected by mRMR and those by the greedy algorithm. However, that is not the case. By selecting 50 features we find that 17% of them are present in both cases and in general the simpler the case (practically: the lower the load) the more matches are present. Finally, the approaches tend to select the same metrics (although with different ranks) across different load scenarios (around 40% of the selected metrics) thus highlighting those that should be considered under most circumstances.

To evaluate the methodology we selected 6 different load scenarios, performed the feature selection and executed the approximation with constantly growing number of features. A typical curve is depicted in Figure 4. while the Mean Square Error results are shown in Table 1. where 'R' stands for Linear Regression, 'N' for Neural Network, 'F' for the Forward Selection and M for the mRMR feature selection respectively.

Table 1.

	MSE - RF	MSE - RM	MSE - NF	MSE – NM
LOW	0.0233	0.0326	1e-30	1e-4
MID	0.0510	0.0887	1.86e-4	1e-29
HIGH	0.2361	0.3139	1e-25	1e-26
VHIGH	0.9309	1.0020	0.7746	0.8111
DROP	0.2806	0.4990	0.0227	0.0516
STEP	0.1908	0.2300	0.0961	0.1818

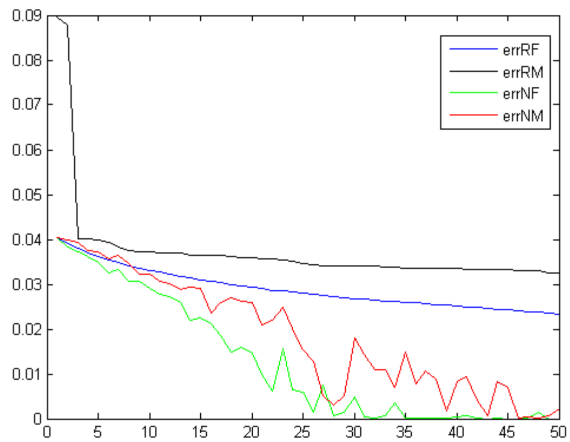

Figure 4. MSE as the function of selected features

The overall results seem to show that despite the successes of using mRMR in bioinformatics applications, it is inappropriate in our case where a simple greedy algorithm can outperform it.

If we take a closer look at the targeted throughput (see Figure 5.), our objective metric, we can discover the intervals where the system begins to saturate. Note the abrupt falls in performance at time instants 33, 74, 110 and 170. Those are the typical times indicating that non-linear phenomenon like resource pooling, swapping and caching do occur and the mutual information based non-linear capabilities of mRMR come in handy.

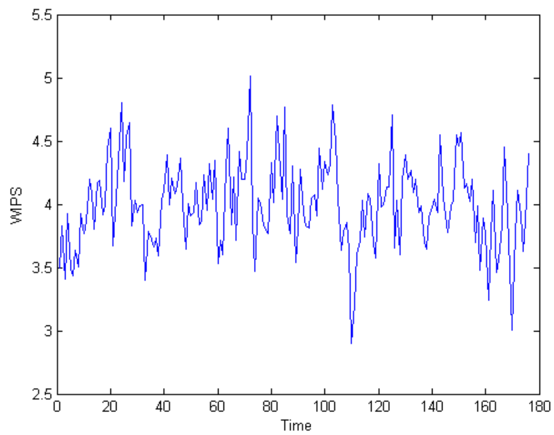

Figure 5. The 'Throughput' objective metric in the MID scenario

Considering this we can assume the following about the system:

- NORMAL operational states can be adequately approximated by means of linear methods, providing good approximations in a simple and computationally inexpensive way.
- DEGRADATION system states however can be more effectively treated by non-linear means that can provide their earlier detection and thus a more time for pro-active actuation.
- SATURATION (over-loaded) system states can also be approximated by linear methods as the systems performance will be mainly influenced by its physical parameters and limits rather than the internal relations of its metrics.

7. Conclusion and Future Work

The most important conclusion of our work is that no single approach is sufficient for system management and early diagnostics, but a combination of the approaches best fitting to the individual qualitatively different operational domains is needed for this purpose. Related efforts [12, 13, 14] all seem to lack this consideration and while [13, 14] show convincing results their impact is limited to the normal and saturated operation states, disregarding degradation, and thus seem insufficient from the pro-active actuation's point of view. [12] on the other hand may lack the benefit of early diagnosis in some cases where linear methods could raise alarm in a more prescient way.

All of the solutions above exploit implicitly some mutual interdependency between the operation domain and the best fitting function. However, our measurements indicate that the behavior of a system is so much different in the individual operational domains that a homogenous approach using a single kind of function fails to faithfully approximate it.

The solution should be a heterogeneous monitoring and control system that utilizes linear and non-linear methods in parallel and switches them according to the current system behavior (Figure 6.).

An additional benefit of this approach is that the growing error between the estimate delivered by the model in use and the observed value indicates simultaneously the necessity of a switchover from one approximation function to another one and correspondingly detects a transition in the operation domain.

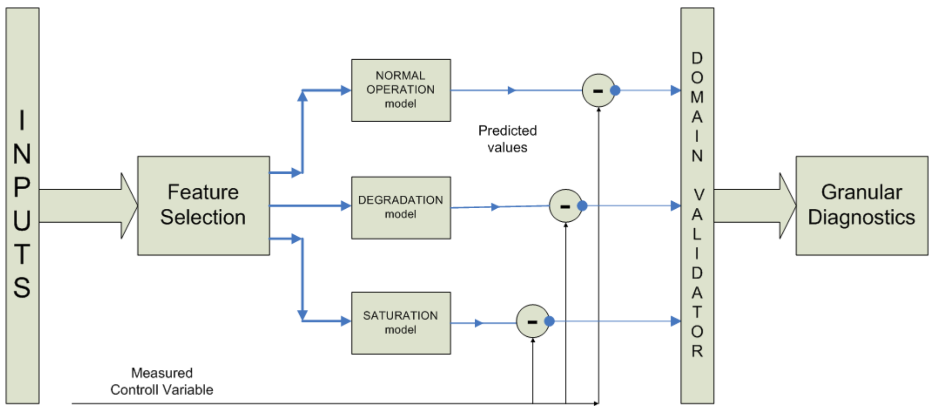

Figure 6. Blockscheme

While our measurements validated the soundness of the multi approximate based approach further questions are raised for checking the practical usefulness:

- As in each case in benchmark-based methods, the representativeness of the benchmark scenarios and measurement setup drastically influence the results, thus determine whether they can be used in a generalized form under all circumstances.
- This approach is expected to scale well with respect to the number of attributes to be processed. Still, a monolithic approach for large systems seems to be disadvantageous. The two main arguments supporting hierarchical modeling and feature selection are the following.
  On the one hand, it is reasonable to believe that the number of distinct "operating points" – the sets of significant variables as the function of system state, current load and time – will become so big that it becomes unpractical for system management design. To counter this, systems can be subdivided into a hierarchy of subsystems so that a higher level (and the feature selection of that) sees only specific, service access related attributes of the subsystems.
  On the other hand, in sizeable heterogeneous, distributed systems the compensation and repair mechanisms usually operate on multiple levels of

granularity; thus, a hierarchical approach with intra-subsystem feature selection is also of paramount importance, not only for localized early warning, but also to support decisions of compensation or repair.
Refinement and experimental validation of the hierarchical approach sketched above shall be performed.

- Our measurements indicated that the linear approximation fits well to the normal operation mode and saturation, mRMR is flexible enough to support a good match to the behavior in the degradation phase. Further experiments are needed to identify the best fitting functions for abrupt changes in the system, like those caused by critical resource faults.
- The number of sufficient features should be determined in a methodical way, e.g. using the Lipschitz-index [21] or some other approach. Here the robustness of the control and its impacts has to be assessed both in the terms of selecting a low number of input variables and the sensitivity to errors in the parameter estimation (this second question is a traditional topic in control theory).

In this paper, we have shown in a methodology experiment that the mRMR feature selection scheme combined with linear approximation can be employed for selecting the few, "most significant" quantitative aspects of a system for the purpose of supervisory instrumentation.

We also have to address the question that how can be the results systematically used for configuring simple rule-based supervision and even more importantly, helping the design of autonomic control schemes.

References

[1] Y. Diao, J. L. Hellerstein, G. Kaiser, S. Parekh, and D. Phung, "Self-managing systems: A control theory foundation," Engineering of Computer-Based Systems., pp. 441–448, Apr. 2005.

[2] IBM Autonomic Computing Initiative , http://www.research.ibm.com/autonomic/

[3] Assessing, Measuring and Benchmarking Resilience, FP7 ICT CA 216295, http://amber.dei.uc.pt/

[4] I. Cohen, M. Goldszmidt, T. Kelly, J. Symons, and J. S. Chase. "Correlating instrumentation data to system states: A building block for automated diagnosis and control". In Proc. 6th USENIX OSDI, San Francisco, CA, Dec. 2004.

[5] S. Zhang, I. Cohen, M. Goldszmidt, J. Symons, and A. Fox. "Ensembles of models for automated diagnosis of system performance problems". Technical Report HPL-2005-3, Hewlett-Packard, January 2005.

[6] R. Powers, M. Goldszmidt, and I. Cohen. "Short term performance forecasting in enterprise systems." ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, Aug. 2005.

[7] Hoffmann, G.A., Trivedi, K.S., Malek, M. "A best practice guide to resource forecasting for computing systems." IEEE Transactions on Reliability, Vol 56, pp 615–628, 2007.

[8] R. Kohavi and G. John. "The wrapper approach". In H. Liu and H. Motoda, editors, Feature Extraction, Construction and Selection: A Data Mining Perspective. Springer Verlag, 1998.

[9] M. Jiang, M. A. Munawar, T. Reidemeister, and P. A. S. Ward. "Information-theoretic modeling for tracking the health of complex software systems". In the Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds, ACM New York, 2008.

[10] C. Ding and H. Peng. "Minimum redundancy feature selection from microarray gene expression data." In Proceedings of the Computational Systems Bioinformatics Conference, pp 523-529, 2003.

[11] Zhou, J., Peng, H.: "Automatic recognition and annotation of gene expression patterns of fly embryos." Bioinformatics 23(5), pp 589–596, 2007.

[12] M. Jiang, M. A. Munawar, T. Reidemeister, and P. A. S. Ward. "Automatic Fault Detection and Diagnosis in Complex Software Systems by Information-Theoretic Monitoring", will appear in IEEE International Conference on Dependable Systems and Networks 2009

[13] M. Grottke, L. Lie, K. Vaidyanathan, and K. Trivedi, "Analysis of software aging in a web server," IEEE Trans. Reliability, vol. 55, no. 3, pp. 411–420, September 2006

[14] Keller, A.; Diao, Y.; Eskesen, F.; Froehlich, S.; Hellerstein, J.l.; Surendra, M.; Spainhower, L.F., "Generic On-Line Discovery of Quantitative Models," Network and Service Management, IEEE Transactions on , vol.1, no.1, pp.39-48, April 2004

[15] TPC-W official page: http://www.tpc.org/tpcw/default.asp

[16] Fodor, I. K. "A survey of dimension reduction techniques". Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory, Center for Applied Scientific Computing, 2002.

[17] L. Molina, L. Belanche, and A. Nebot. "Feature selection algorithms: a survey and experimental evaluation." International conference on data mining, Maebashi City, Japan, 2002.

[18] Hanchuan Peng; Fuhui Long; Ding, C.: "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol . 27, pp. 1226-1238, 2005.

[19] Li, S., Zhu, Y., Feng, J., Ai, P., and Chen, X. "Comparative Study of Three Feature Selection Methods for Regional Land Cover Classification Using MODIS Data". In Proceedings of the 2008 Congress on Image and Signal Processing, Vol. 4, 2008.

[20] H. Chen, G. Jiang, K. Yoshihira: "Monitoring High-Dimensional Data for Failure Detection and Localization in Large-Scale Computing Systems". IEEE Trans. Knowl. Data Eng. (TKDE) 20., 2008

[21] X. He and H. Asada, "A new method for identifying orders of input–output models for nonlinear dynamic systems," in Proc. Autom. Contr. Conf., 1993, pp. 2520–2523