Model Identification for Energy-Aware Management of Web Service Systems

Mara Tanelli[1,2], Danilo Ardagna[1], Marco Lovera[1], and Li Zhang[3,⋆]

[1] Dipartimento di Elettronica e Informazione,
Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy
{tanelli,ardagna,lovera}@elet.polimi.it
[2] Dipartimento di Ingegneria dell'Informazione e Metodi Matematici,
Università degli studi di Bergamo, Via Marconi 5, 24044, Dalmine (BG), Italy
[3] IBM Research, T.J. Watson Research Center, Yorktown Heights, NY 10598
zhangli@us.ibm.com

Abstract. In SOA environments, service providers need to comply with the service level objectives stipulated in contracts with their customers while minimizing the operating costs of the physical infrastructure, mainly related to energy costs. The problem can be effectively formalized by using system identification and control theory: the service levels are translated into set-points for the response times of the hosted applications, and performance are traded-off with energy saving objectives based on suitable models for server dynamics. As the behavior of the incoming workload changes significantly within a single business day, control-oriented system identification approaches are very promising to model such systems, especially at a very fine grained time scales and in transient conditions. In this paper Linear Parameter Varying (LPV) state space system identification algorithms are analyzed for modeling Web services systems. The suitability of LPV models is investigated and their performance assessed by experimental data.

1 Introduction

Energy management is rapidly becoming a priority in the design and operation of complex service-based information systems, as the impact of energy consumption associated with IT infrastructures increases. The growth in the number of servers and the increasing complexity of the network infrastructure have caused an enormous spike in electricity usage. IT analysts predict that, by 2012, up to 40% of IT budget will be consumed by energy costs, [12]. This trend is striving green computing activities in the industry research agenda (see for example IBM's project *Big Green* [8] and HP's *Green up* initiative [7]).

In the context of Web services and SOA based systems, service centers need to comply also to the Service Level Agreements (SLAs) stipulated with their customers. At run

[⋆] The work of M. Tanelli and M. Lovera has been partially supported by MIUR project "New methods for Identification and Adaptive Control for Industrial Systems." The work of D. Ardagna has been partially supported by the EU FP7 Q-ImPrESS project.

A. Bouguettaya, I. Krueger, and T. Margaria (Eds.): ICSOC 2008, LNCS 5364, pp. 599–606, 2008.

time, service requestors address their invocation to the most suitable provider according to their Quality of Service (QoS) preferences. QoS requirements are difficult to satisfy because of the high variability of Internet workloads. It is difficult to estimate workload requirements in advance, as they may vary by several orders of magnitude within the same business day, [6]. To handle workload variations, many service centers employ autonomic techniques [4,18] such that resources are dynamically allocated among running Web services based on short-term demand estimates. The goal is to meet the application requirements while adapting the IT system. This leads to the study of how to efficiently use resources and reduce energy consumption.

Early autonomic techniques switched servers on and off based on the service center workload, [4]. More recent proposals, see e.g. [13,10], have started reducing the frequency of operation of servers by exploiting the Dynamic Voltage Scaling (DVS) mechanisms implemented in new servers. DVS varies both CPU supply voltage and operating frequency. The adoption of DVS is very promising, as power consumption is proportional to the cube of the operating frequency, while servers performance varies linearly with the operating frequency. Furthermore, DVS does not introduce any system overhead (vice versa, hibernating and restoring a server require time and energy).

Several research contributions have proposed autonomic self-managing techniques and can be classified mainly in two categories: (i) utility-based optimization techniques, and (ii) feedback control-theoretic approaches. Utility-based approaches have been introduced to optimize the degree of user satisfaction by expressing their goals in terms of user-level performance metrics. Typically, the system is described by means of a performance model based on queueing theory, embedded within an optimization framework. Utility based approaches can handle multiple decision variables (e.g., admission control, application placement, load balancing, etc.) but are based on the assumption that the system is at *steady state*. Hence, these techniques are effective on a medium term control time horizon, e.g., half an hour, [4], [13]. Vice versa, genuine control-theoretic approaches can accurately model system transients and can adjust the system configuration within a very short time frame. Thus, control-theoretic approaches are effective over fine grained time horizons, e.g., minutes and, furthermore, can effectively employ DVS as control variable and formally guarantee both closed-loop stability and performance specifications.

In this paper, the adoption of Linear Parametrically Varying (LPV) models for the performance control of Web services will be addressed. A LPV model is linear in the parameters and a vector of scheduling variables enters the system matrices in an affine or linear fractional way ([11,19,16]). Such a representation for general nonlinear systems can be useful in view of control design using modern robust control and gain-scheduling control techniques [3]. Models are identified from experimental data measured on a micro-benchmarking Web service application, adopting DVS of CPUs as control variable.

The structure of the paper is as follows. Section 2 provides a review of the literature, while Section 3 formally states the problem addressed in this paper and introduces the needed notation. Section 4 briefly describes discrete time state space dynamical models and illustrates the LPV models adopted in this work. Experimental results are presented in Section 5. Conclusions are finally drawn in Section 6.

2 Related Work

Autonomic management of service center infrastructure is receiving great interest by the control theory research community. The first control-oriented contributions applied to the management of Web services are reported in [1,15], and use feedback control to limit the utilization of bottleneck resources by means of admission control and resource allocation. In the practice of control engineering, when a single control system must be designed to guarantee closed-loop operation of a given plant in many different operating conditions, two broad classes of methods are available: gain scheduling and adaptive control.

The gain scheduling approach to the problem can be summarised as follows: find one or more *scheduling variables* which can completely parameterise the operating space of interest for the system to be controlled; define a parametric *family* of linearised models for the plant associated with the set of operating points of interest; finally, design a *parametric* controller which can both ensure the desired control objectives in each operating point and an acceptable behaviour during (slow) transients between one operating condition and the other. A wide body of design techniques is now available for this problem (see, e.g., [3]), which can be reliably solved provided that a suitable model in parameter-dependent form has been derived. This modelling problem, however, raises a number of significant issues. While the literature on non-linear identification can now provide advanced tools for the estimation of a wide variety of model classes, in such a case it would be useful to separate conventional input variables from scheduling variables (i.e., variables defining the operating point of the plant), by letting them enter the model in distinct ways. LPV models have been recently proposed as a way of dealing with this kind of problems and have been adopted recently in [13] to implement an autonomic controller able to provide performance guarantees by means of DVS.

With respect to that approach, where input/output (I/O) LPV identification was considered, the method adopted here is more appropriate to provide system models tailored to LPV control design, as they are directly identified in state space form and avoid all the issues - not addressed in [13] - related with equivalence notions between I/O and state space LPV realizations, [17]. Furthermore, state space LPV identification allows a straightforward extension to the multiple input, multiple output case, which is needed if more than one class of customers needs to be taken into account.

Control theoretic approaches are suitable to model Web systems both in stationary and transient conditions. In the queueing theory literature, some recent proposals address the problem of modelling queue network transient behaviour by means of Markov models in order to study burstiness and long range dependency in system workloads [5,14]. The main limitation of Markovian models is their complexity, which makes, even for very simple systems, e.g., a first come first served (FCFS) single server queue, the number of parameters to be determined quite large. These models suffer for high computation overhead and, hence, are presently not suitable for the implementation of real-time controllers.

3 Problem Statement

In this paper, a CPU bounded Web service application will be considered where, without loss of generality, the resource scheduler implements a FCFS policy (LPV models can consider also other scheduling policies, e.g., processor sharing or generalized processor sharing). In the remainder of the paper the following notation will be adopted:

- Δt: sampling interval;
- k: discrete time index over the interval $[k\Delta t,\ (k+1)\Delta t]$;
- λ_k: requests arrival rate at the server in the k-th interval;
- s_k: requests service time, i.e., overall CPU time needed to process a request in the k-th interval;
- R_k: average server response time in the k-th interval, i.e., the overall time a request stays in the system.

We assume that s_k is inversely proportional to the physical server CPU frequency. As such, when physical servers are endowed with DVS capabilities, the effect of - say - lowering the CPU frequency when a light workload is present in the system causes an increase of the effective CPU time needed to serve a request [10]. This assumption is supported by current technology trends, since in modern systems (e.g., AMD Operon 2347HE Barcelona core) CPUs and RAM clock can be scaled independently (otherwise, RAM access could become a bottleneck and CPUs could stall for memory accesses). If we denote with f_k the ratio of the frequency applied during the time interval k to the physical server maximum frequency, the *effective* service time can be defined as $s_{f,k} = s_k/f_k$.

The goal of this paper is to derive a dynamic model of an application server capable of capturing system behaviour at a very fine-grained time resolution (seconds), with an accuracy suitable for control purposes. This identification process provides a control-oriented dynamical description of the server behavior and it is the first step to be achieved in order to design a closed-loop controller for service center infrastructures able to meet SLAs requirements while minimizing energy costs. The design of the closed-loop controller is the focus of our ongoing work. The adoption of control oriented techniques is motivated since the workload of SOA systems is characterized by highly varying conditions [18]. The LPV framework is adopted since it seems very promising for modeling such systems. Furthermore once the modelling and control problem in the LPV framework is solved, the closed-loop system will not require to be complemented with workload predictors, whose design is hard to carry out, as the best models have been shown to require nonlinear and non-stationary workload description [2]. In fact, the workload variability is embedded in the LPV system representation, which tunes on-line both the model and the control action taking into explicit account the current workload condition.

4 Identification of Discrete-Time State Space Models

The problem of model identification can be formulated as the one of deriving a mathematical representation for the behaviour of a physical system on the basis of input-output data collected in dedicated experiments. As far as linear models are concerned, the main “ingredients” of an identification problem are essentially: 1) a definition of the class of models to be considered and 2) a suitable algorithm for the estimation of the model parameters on the basis of the available data. Classical model identification problems for Single-Input Single-Output time-invariant systems are formulated using models in input-output form (i.e., difference equations relating the measured input and output variables in a direct way), the parameters of which are estimated using least squares techniques. Whenever Multiple-Input Multiple-Output and/or time-varying systems must be dealt with, state-space representations turn out to be a more flexible and reliable model class. In this work discrete-time linear state-space models will be considered, in the *innovation form*:

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + K_k e_k \\ y_k &= C_k x_k + D_k u_k + e_k, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the vector of control inputs and $y \in \mathbb{R}^l$ is the vector of measured outputs and e is a white process noise. More precisely, with reference to the specific modelling problem considered in this study, $u_k = s_{f,k}$ and $y_k = R_k$, i.e., the goal of the model identification problem considered in this paper is to derive a state-space model describing the dynamic relationship between the effective service time and the server response time. In (1) generically time-varying state space matrices $\{A_k, B_k, C_k, D_k\}$ have been considered. In what follows we will introduce different, additional assumptions on the time-variability of the model dynamics, suitably tailored for the model identification problems associated with the management of Web services.

More precisely, the generic time-variability will be restricted to the LPV class. LPV systems are linear time-varying plants whose state space matrices are fixed functions of some vector of varying parameters. LPV model identification algorithms are available in the literature both for input-output and state-space representations of parametrically-varying dynamics. In this work state-space LPV models will be adopted:

$$\begin{aligned} x_{k+1} &= A(\delta_k) x_k + B(\delta_k) u_k \\ y_k &= C(\delta_k) x_k + D(\delta_k) u_k, \end{aligned} \tag{2}$$

where $\delta \in \mathbb{R}^s$ is the parameter vector. For the sake of simplicity, in the following we will deliberately focus on purely deterministic models, i.e., we will ignore the presence of the noise terms in the model representation. It is important to point out, however, that the theory underlying the parameter estimation algorithms used in this work can effectively deal with the presence of process and measurement noise [19]. It is often necessary to introduce additional assumptions regarding the way in which δ_k enters the system matrices. The most common are the following:

1. Affine parameter dependence (LPV-A), where $A(\delta_k) = A_0 + A_1\delta_{1,k} + \ldots + A_s\delta_{s,k}$ and similarly for B, C and D. $\delta_{i,k}, i = 1, \ldots, s$ denotes the i-th component of vector δ_k. This form can be immediately generalised to polynomial parameter dependence.
2. Input-affine parameter dependence (LPV-IA): this is a particular case of the LPV-A parameter dependence in which only the B and D matrices are considered as parametrically-varying, while A and C are assumed to be constant: $A = A_0$, $C = C_0$.

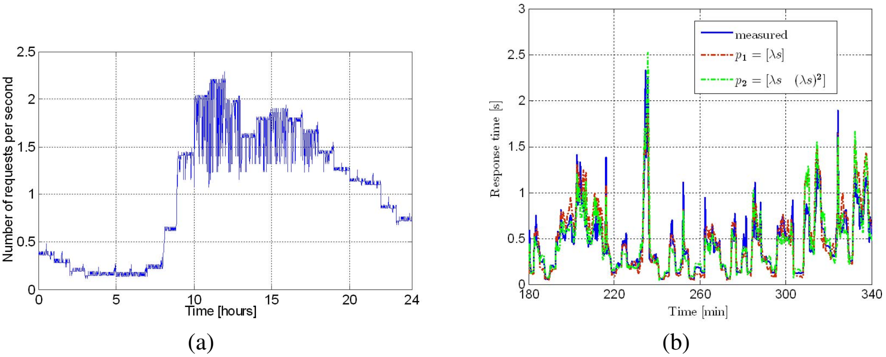
Identifying LPV models in general state space form is a difficult task. It is usually convenient to consider first the simplest form, i.e., the LPV-IA one, as its parameters can be retrieved by using subspace model identification (SMI) algorithms for linear time invariant systems (which are significantly easier to use and available in commercial software packages) by suitably extending the input vector. In this work the MOESP class of SMI algorithms [20] has been considered. The classical way to perform linear system identification is by minimizing the error between the real output and the predicted output of the model. A similar approach can be used for LPV state-space systems of the form (2). Letting the system matrices of (2) be completely described by a set of parameters θ, identification can be carried out by minimizing the cost function $V_N(\theta) := \sum_{k=1}^{N} ||y_k - \widehat{y}_k(\theta)||_2^2 = E_N^T(\theta)E_N(\theta)$ with respect to θ, where $E_N^T(\theta) = \left[\left(y_1 - \hat{y}_1(\theta)\right)^T \cdots \left(y_N - \hat{y}_N(\theta)\right)^T\right]$, while y_k denotes the measured output and $\widehat{y}_k(\theta)$ denotes the output of the LPV model to be identified. In general, the minimization of $V_N(\theta)$ is a nonlinear, nonconvex optimization problem. Many algorithms have been proposed, in this work the gradient search method based on the Levenberg-Marquardt algorithm has been adopted [11].

5 Experimental Results

In the experimental framework, a workload generator and a micro-benchmarking Web service application have been used. The workload generator is based on a custom extension of the Apache *JMeter* 2.3.1 workload injector, which allows to generate workload according to an open model [9] with a Poisson arrival process. The Web service is a Java servlet designed to consume a fixed amount of CPU time generated according to deterministic (for identification purposes), Poisson, Pareto and log-normal distributions (for validation). The adoption of a micro benchmarking application allows the validation of the effectiveness of our approach both for workload intensive and for computationally intensive applications. Furthermore, the CPU time standard deviation of the micro benchmarking application has been varied in order to verify if LPV models performance depends on the variability of the CPU time distribution: the standard deviation $\sigma[s]$ has been chosen as q times the average of the service time distribution $E[s]$, i.e., $\sigma[s] = q\,E[s]$, where q was set equal to 2, 4 and 6. For model validation, the incoming workload reproduces a 24 hour trace obtained from a large Internet Web site. The log was collected on a hourly basis. The workload injector is configured to follow a Poisson process with request rate changing every minute where the requests rate is obtained from the log trace superimposed with a Gaussian noise proportional to the workload intensity, as in [10].

To quantitatively evaluate the models, two metrics have been considered: the percentage Variance Accounted For (VAF), defined as $VAF = \left(1 - \frac{Var[y_k - \widehat{y}_k(\theta)]}{Var[y(k)]}\right)$, where y_k is the measured signal (i.e., application response time), and $\widehat{y}_k(\theta)$ is the output obtained from the simulation of the identified model, and the percentage average simulation error e_{avg}, computed as $e_{avg} = \left(\frac{E[|y_k - \widehat{y}_k(\theta)|]}{E[|y_k|]}\right)$.

The identification data were processed to extract the average values over a sampling interval $\Delta t = 10$s and two LPV second order models, one with $p_1 = \lambda\, s$ and the other with $p_2 = [\lambda\, s \quad (\lambda\, s)^2]$ have been identified (see Figure 1(b) for a plot of a detail of the results). The plot shows that the models are capable of providing a response time which correctly follows the peaks of the measured one. Results reported in Table 1 also show that the performance of LPV models are almost independent on the value of q, i.e., the models are robust to the variability of the service time distribution.



(a)

(b)

Fig. 1. (a) Time history of the request rate applied during a validation test; (b) Detail of the measured (solid line) and the response time obtained with $\Delta t = 10\,\text{s}$ an LPV model with $p_1 = \lambda\, s$ (dashed line) and $p_2 = [\lambda\, s\; (\lambda\, s)^2]$ (dash-dotted line) on identification data in with $q = 4$

Table 1. Performance of the identified models with $\Delta t = 10\text{s}$ on validation data

Valid. Performance - LPV	q=2		q=4		q=6	
Δt = 10 s	(p_1)	(p_2)	(p_1)	(p_2)	(p_1)	(p_2)
VAF	58.31%	74.14%	54.01%	71.50%	58.85%	74.52%
e_{avg}	25.70%	18.36%	20.30%	7.40%	31.87%	31.67%

6 Concluding Remarks and Future Work

This paper presents the results obtained in the application of LPV model identification techniques for the performance control of Web services. Specifically, the suitability of subspace LPV methods has been checked against experimental data measured on a custom implementation of a workload generator and a micro-benchmarking Web service application. Future work will develop along two directions: on the modelling side, we aim to further validate our approach on real applications and to extend the models considering a multi-class framework in virtualized environments, whereas on the control design side, work will be devoted to devise both admission control policies and response time regulation in the LPV framework.

References

1. Abdelzaher, T., Shin, K.G., Bhatti, N.: Performance Guarantees for Web Server End-Systems: A Control-Theoretical Approach. IEEE Trans. on Parallel and Distributed Systems 15(2) (March 2002)
2. Andreolini, M., Casolari, S.: Load prediction models in web-based systems. In: Proc. of the 1st international conference on Perf. evaluation methodologies and tools, Pisa, Italy (2006)
3. Apkarian, P., Adams, R.J.: Advanced Gain-Scheduling Techniques for Uncertain Systems. IEEE Trans. on Control System Technology 6, 21–32 (1998)
4. Ardagna, D., Trubian, M., Zhang, L.: SLA based resource allocation policies in autonomic environments. Journal of Parallel and Distributed Computing 67(3), 259–270 (2007)

5. Casale, G., Mi, N., Smirni, E.: Bound Analysis of Closed Queueing Networks with Workload Burstiness. In: Proc. of SIGMETRICS (2008)
6. Chase, J.S., Anderson, D.C.: Managing Energy and Server Resources in Hosting Centers. In: ACM Symposium on Operating Systems principles (2001)
7. HP. Green up initiative,
   http://www.hp.com/hpinfo/newsroom/feature-stories/2007/07-360-greenup.html
8. IBM. Project Big Green,
   http://www-03.ibm.com/press/us/en/photo/21514.wss
9. Kleinrock, L.: Queueing Systems. John Wiley and Sons, Chichester (1975)
10. Kusic, D., Kandasamy, N.: Risk-Aware Limited Lookahead Control for Dynamic Resource Provisioning in Enterprise Computing Systems. In: ICSOC 2006 Proc. (2006)
11. Lee, L.H., Poolla, K.: Identification of linear parameter-varying systems using nonlinear programming. ASME J. of Dynamic Systems, Measurement and Control 121(1), 71–78 (1999)
12. Metha, V.: A Holistic Solution to the IT Energy Crisis (2007),
    http://greenercomputing.com/
13. Qin, W., Wang, Q.: Modeling and control design for performance management of web servers via an LPV approach. IEEE Trans. on Control Systems Tech. 15(2), 259–275 (2007)
14. Riska, A., Squillante, M., Yu, S.Z., Liu, Z., Zhang, L.: Matrix-Analytic Analysis of a MAP/PH/1 Queue Fitted to Web Server Data. In: Latouche, G., Taylor, P. (eds.) Matrix-Analytic Methods: Theory and Applications, pp. 335–356. World Scientific, Singapore (2002)
15. Robertsson, A., Wittenmark, B., Kihl, M., Andersson, M.: Admission control for web server systems - design and experimental evaluation. In: 43rd IEEE Conference on Decision and Control (2004)
16. Tanelli, M., Ardagna, D., Lovera, M.: LPV model identification for power management of web service systems. In: 2008 IEEE Multi-conference on Systems and Control, San Antonio, USA (2008)
17. Toth, R., Felici, F., Heuberger, P.S.C., Van den Hof, P.M.J.: Discrete time LPV I/O and state space representations, differences of behavior and pitfalls of interpolation. In: Proc. of the 2007 European Control Conference, Kos, Greece (2007)
18. Urgaonkar, B., Pacifici, G., Shenoy, P.J., Spreitzer, M., Tantawi, A.N.: Analytic modeling of multitier Internet applications. ACM Transaction on Web 1(1) (January 2007)
19. Verdult, V.: Nonlinear System Identification: A State-Space Approach. PhD thesis, University of Twente, Faculty of Applied Physics, Enschede, The Netherlands (2002)
20. Verhaegen, M.: Identification of the deterministic part of MIMO state space models given in innovations form from input output data. Automatica 30, 61–74 (1994)