



**HAL**  
open science

## Towards Non-Stationary Grid Models

Tamas Elteto, Cecile Germain-Renaud, Pascal Bondon, Michèle Sebag

► **To cite this version:**

Tamas Elteto, Cecile Germain-Renaud, Pascal Bondon, Michèle Sebag. Towards Non-Stationary Grid Models. *Journal of Grid Computing*, 2011, 9 (4), pp.423-440. 10.1007/s10723-011-9194-z . inria-00616279

**HAL Id: inria-00616279**

**<https://inria.hal.science/inria-00616279>**

Submitted on 21 Aug 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Towards non-stationary Grid models

Tamás Éltető · Cécile Germain-Renaud ·  
Pascal Bondon · Michèle Sebag

Received: date / Accepted: date

**Abstract** Despite intense research on grid scheduling, differentiated quality of service remains an open question, and no consensus has emerged on the most promising strategy. The difficulties of experimentation might be one of the root causes of this stalling. An alternative to experimenting on real, large, and complex data is to look for well-founded and parsimonious representations, which may also contribute to the a-priori knowledge required for operational Autonomics. The goal of this paper is thus to explore explanatory and generative models rather than predictive ones. As a test case, we address the following issue: is it possible to exhibit and validate consistent models of the grid workload? Most existing work on modeling the dynamics of grid behavior describes grids as complex systems, but assumes a steady-state system (technically stationarity) and concludes to some form of long-range dependence (slowly decaying correlation) in the associated time-series. But the physical (economic and sociologic) processes governing the grid behavior dispel the stationarity hypothesis. This paper considers an appealing different class of models: a sequence of stationary processes separated by breakpoints. The model selection question is now defined as identifying the breakpoints and fitting the processes in each segment. Experimenting with data from the EGEE/EGI grid, we found that a non-stationary model can consistently be identified from empirical data, and that limiting the range of models to piecewise affine (autoregressive) time series is sufficiently powerful. We propose and experiment a validation methodology that empirically addresses the current lack of theoretical results concerning the quality of the estimated model parameters. Finally, we present a

---

Tamás Éltető  
University Paris-Sud, INRIA-Saclay and CNRS  
E-mail: eltetot@lri.fr

Cécile Germain-Renaud (corresponding author)  
University Paris-Sud, INRIA-Saclay and CNRS  
E-mail: cecile.germain@lri.fr

Pascal Bondon  
Laboratoire des Signaux et Systèmes, Supelec, CNRS  
E-mail: Pascal.Bondon@lss.supelec.fr

Michèle Sebag  
University Paris-Sud, INRIA-Saclay and CNRS  
E-mail: michele.sebag@lri.fr

---

bootstrapping strategy for building more robust models from the limited samples at hand.

## 1 Introduction

Large-scale distributed computing systems, such as EGEE/EGI [14], allocate computing resources following the matchmaking principle pioneered by Livny [53]: the providers publish the characteristics of their resources, and these are matched with the users' requests. The fundamental motivation for the matchmaking principle is the federative nature of real-world grids. On the other hand, grid users, or grid market participants, seek for differentiated Quality of Service (QoS): in the e-science context, physicists ask for a different service for interactive analysis tasks and for long running simulations; TeraGrid users exploit the Batch Queue Predictor [4] capacities. An extensive body of research e.g. to cite only a few [31, 35, 38, 51] focuses on economic and intelligent models of resource provisioning for QoS, which sophisticate, but do not contradict, the matchmaking principle. Despite this intense activity, no consensus has emerged on the most promising strategies. For instance, gLite [13], the major EGEE/EGI middleware, adopts an agnostic approach derived from the Copernican principle [22] ("job is not special"); even research grids are quite conservative when production is concerned.

Scheduling for large-scale distributed systems explores a very complicated landscape. Any job dispatcher has to integrate a feedback loop with the resource provider; the usage involve *externalities*, decisions which affects users and resources beyond the resource consumer and producer; QoS should not result in under-utilization, thus even the more constrained models should state scheduling as a multi-objective optimization problem. On top of these intrinsic difficulties, two operational issues contribute to challenge the researcher. First real world experimentation is hardly possible. Second, significant experiments with simulators or analysis require large data sets. These data sets may be publicly available [12], but comparative experiments are rare in the grid community and experiments on high level concepts such as autonomic programming models [24, 32], are extremely difficult to conduct. One of the reasons is probably to be found in the well-known data-mining ratio: 80% of the effort goes to pre-processing.

An alternative to experimenting on real, large, and complex data is to look for well-founded and parsimonious representations, with the unavoidable approximations implied. Autonomics calls for models too. In principle, the knowledge at the core of the MAPE-K loop (monitor-analyze-plan-execute) [21] might be built from *tabula rasa*. In fact, real use cases have been repeatedly proven to require a priori knowledge [51], for which inferred models are an attractive alternative to human expertise.

The goal of this paper is thus to explore explanatory and generative models rather than predictive ones. As a test case, we address the following issues. Is it possible to exhibit consistent models of the grid workload? If such models do exist, which classes of models are more appropriate considering both simplicity and descriptive power? How can we actually fit such models? And finally, how can we rigorously assess the quality of these models?

This alternative path immediately raises fundamental issues in statistical inference. In a nutshell, these are related to stationarity. Most existing work on modeling grid behavior describes grids as complex systems. Accordingly, when the target data to model are limited to time-series, most existing work under the stationarity hypothesis

concludes to some form of *long-range dependence*. This paper considers an appealing different class of models: *piecewise stationary* models, that is a sequence of stationary processes separated by breakpoints. The model selection question is now defined by the identification of the breakpoints and the fitting of the processes in each segment.

Our main contributions are as follows.

- To the best of our knowledge, the recognized complexity of grid behavior is generally addressed through equivalently complex models assuming stationarity. We propose to explore a fully different class of models, where non-stationarity is a first-class component.
- We found that a non-stationary model can consistently be fitted from empirical data, and that limiting the range of models to piecewise affine (autoregressive) time series is sufficiently powerful. We propose and experiment a validation methodology that empirically addresses the current lack of theoretical results concerning the quality of the estimated model parameters.
- Finally, we present a bootstrapping strategy for building more robust models from the limited samples at hand.

The experimental part of this paper is based on exhaustive information covering more than a year of the flagship EU grid infrastructure EGEE/EGI, and is representative of a significant fraction of e-science computing activity in Europe. The focus is on a particular category of grid time-series, the *workload*, which is the product of the users' requests (arrival and request process) and the middleware stack activity through scheduling and mapping. This paper extends our previous work [11] by explicating the theoretical hypotheses underlying the mathematical model, improved bootstrapping, and a much more developed analysis of the experimental results.

The rest of the paper is organized as follows. Section 2 surveys the alternative models for complex time-series, and defines the general scope and interest of piecewise linear ones. Related work in grid modeling is also discussed here. Section 3 describes the data set, its grid context, and the derivation of the times-series workload process from the empirical data. Section 4 describes the model selection procedure, and the validation methodology. Section 5 discusses the experimental results. The stability analysis, and the exploitation of bootstrap for use in model reconciliation are presented in section 6, before the conclusion.

## 2 Grid models

### 2.1 Empirical motivation for piecewise models

Almost all existing work on the characterization of the dynamics of the grid (and previously, clusters) behavior implicitly assumes stationarity. Formally, stationarity means that the joint probability distribution does not change when shifted in time or space. Empirically, this amounts to formulating the hypothesis that, at least at some relevant scale, the underlying "physical" process (including resources as well as workload) that creates the observed time-series either does not change at all, or evolves along a slow drift. Even more informally, the hypothesis would be the one of a mature, steady-state system. We advocate that this is not true.

The "physical process" is the combination of the requirements of the e-science actors (requests), and the grid middleware (services). But both do experiment unexpected and abrupt changes. Important events in the life of scientists (e.g. conferences)

create massive peaks of activity; more importantly, experimental events in the science itself (e.g. data taking at the Large Hadron Collider) shift the focus of users and correlatively the usage of the infrastructure at non-predictable times; finally, the grid middleware significantly evolves at major releases, triggering complex interactions with and amongst the users adapting to the new service infrastructure. Striking examples of such events, and their impact on workload, are presented in [15].

Nonetheless, a slow drift is a component of the long-term behavior. In a given scientific community, the usage of a new infrastructure qualitatively follows Roger's Diffusion of Innovation curve [44]: the overall usage is increasing, and at a non-constant rate. However, at the scales relevant for the goal presented in the introduction i.e. explanatory and generative models for scientific and technical research (instead of e.g. long-term capacity planning), the dominating effects are the abrupt changes, and the drift can be to the first order approximated as linear.

## 2.2 Formal motivation for piecewise models

We consider the problem of modeling a non-stationary time series by segmenting the series into blocks of stationary processes. More precisely, the time series is represented by a parametric model in which the parameters change values at fixed unknown times called *breakpoints*. The problem is in estimating the breakpoint number and the locations as well as in selecting an appropriate model and then in estimating its parameters for each stationary regime. This problem is addressed by [25, 17, 6] when the pieces are autoregressive (AR) processes and we shall follow the same line: as explained in the introduction, AR models are both powerful and computationally tractable.

A commonly adopted alternative strategy is to consider that the series is stationary but exhibits *long range dependence*, i.e. slowly decaying correlation. This type of series appear in many areas, including hydrology, meteorology, economics, finance and telecommunications; see for instance, [1] and [49]. In all these examples, there is some empirical motivation for assuming that the distribution of the random variables (here the workload), including their correlation, is constant over time: at a certain scale, the root causes of the observed phenomenon are supposed invariant. A useful model for long range dependence processes is the fractional autoregressive integrated moving-average process, introduced by [19] and [20]. The main feature of this process is that its covariance function decays hyperbolically, while the covariance function of a short memory process decays exponentially.

Nevertheless, structural changes and long range dependence are easy to confuse. Indeed, short-memory processes which are affected by a regime change or by a smoothly varying trend may exhibit features that make standard tools incorrectly conclude that long-memory is present, leading to spurious long-memory; see e.g. [2, 52, 7, 18]. In particular, [2] gave necessary and sufficient conditions for a weakly dependent stationary sequence of random variables perturbed by some trend to produce spurious long memory; [52] showed that a variance-type estimator of the long memory parameter can be misleading when the series has either a jump in the mean or a slow trend; [7] and [18] gave both theoretical justification and Monte-Carlo evidence to support the claim that long memory and structural change are easily confused in simple econometric models and financial data.

Our strategy here is to fit a piecewise AR model to the data. More precisely, let  $m$  denote the unknown breakpoint number and  $n$  the length of the time series. For

$j = 1, \dots, m$ , let  $\tau_j$  be the breakpoint between the  $j$ th and  $(j + 1)$ th AR regime, and set  $\tau_0 = 1$  and  $\tau_{m+1} = n + 1$ . For  $j = 1, \dots, m + 1$ , the  $j$ th piece of the observed time series  $\{Y_t\}$  is modeled by

$$Y_t = X_{t,j}, \quad \tau_{j-1} \leq t < \tau_j, \quad (1)$$

where  $\{X_{t,j}\}$ ,  $t \in \mathbb{Z}$ , is the AR( $p_j$ ) process defined by the difference equation

$$X_{t,j} = \gamma_j + \phi_{j,1}X_{t-1,j} + \dots + \phi_{j,p_j}X_{t-p_j,j} + \sigma_j\epsilon_t, \quad (2)$$

$\{\epsilon_t\}$  is a sequence of independent identically distributed (iid) zero-mean Gaussian random variables with unit variance,  $\sigma_j > 0$ , and the polynomial  $\Phi_j(z) = 1 - \phi_{j,1}z - \dots - \phi_{j,p_j}z^{p_j}$  has no zeros in the closed unit disk  $\{z \in \mathbb{C} : |z| \leq 1\}$ . An important argument for focusing on piecewise AR models is that they are dense in the class of locally stationary processes with regular time varying spectrum; more intuitively, they are good approximations of a fairly general class of processes. As efficient algorithms exist for fitting an AR model, most of the computational complexity of the fit lies in the identification of the changes of regime, as shown in section 4.

### 2.3 Related work

*Explanatory* models of the workload in HPC systems [9, 46] characterize the distributional properties of the quantities of interest for job behavior (e.g. inter-arrival time, queuing delay, or execution time) using different parametric models. More recently, efforts [23, 29, 30, 33, 47] address grid systems along the same path.

A different approach targets *predictive* models, either by time series analysis methods [8, 29, 36, 54] or statistical ones [4, 27, 55]. This direction of research selects a specific view of the system, such as short time range for time-series, or features of the job and target execution support, in order to improve the predictive accuracy at the expense of a general model.

Our work shares the explanatory goal of the first approach, and the techniques of time-series analysis of some of the second one. It differs in two significant ways, which as far as we know have not yet been explored. First, we aim at identifying the structural breaks in the model, and we exploit an unified method for fitting both the segments model and its ruptures, rather than assuming stationary processes or decoupling the identification techniques for changes of regime and intra-regime behavior. Second, the bootstrapping strategy addresses the lack of confidence associated with the uncertainties and non-reproducibility of the acquisition process.

The need for *generative* models for scientific research has been identified more recently as a separate requirement from the two previous ones. Epema advocates for this goal [34], and specifically presents an integrated model of parallel bag-of-tasks (BoT) workload. The proposed model is built on four a-priori concepts (long-range dependence, temporal and spatial burstiness, BoT clustering, and correlation between parallelism and runtime), which implicitly define an ontology of parallel workloads (e.g. by implicit rules about the mapping of a parallel computation on a given resource). A multivariate approach might be required for modeling systems where coupled computations are dominant. However, more concepts (realized as sub-model parameters) increase the risk of an arbitrary choice inspired by the considered samples. Specifically, [34] validates the generative model only in a narrow sense: it checks the coincidence of

the previously described four concepts as fitted from experimental data with those fitted from a generated realization. In other words, if e.g. burstiness does not fully capture an important property of the empirical data, such as non-stationarity, the validation procedure might not show the bias in model selection. Actually, [34] estimates all parameters by statistical procedures taking as inputs multi-years samples, thus implicitly assuming stationarity.

Considering models as capturing knowledge for Autonomics, the alternative was stated by G. Tesauro in [42], and explored in [50] addressing dynamic resource provisioning through Reinforcement Learning (RL). At least for RL, the conclusion is clear: *tabula rasa* learning is very likely to be defeated, because it requires an exceedingly risky exploration/exploitation tradeoff as all learning happens on-line and on real data. This conclusion might seem paradoxical, because RL is the paradigm of model-free policies [48]. In fact, limiting exploration by pre-training the learner on a generative model off-line, has been shown to be critical for a large range of applications of RL; [50,39] exemplify this feature for the specific case of resource provisioning in large scale distributed systems. The focus of this paper on non-stationary processes is particularly relevant for RL: at rupture points, by definition the statistical distribution changes and the learned value function becomes (partially) obsolete.

### 3 The Workload Process

#### 3.1 EGEE and gLite

For the sake of precision and because the experimental data set come from EGEE, this section will describe its scheduling under gLite [13], its major middleware. gLite integrates the sites' computing resources through a set of middleware-level services; this set is called the Workload Management System (WMS). The WMS accepts jobs from users and dispatches them to computational resources based on the users requirements on one hand, and the characteristics (*e.g.* hardware, software, localization) and state of the resources on the other hand. Thus, the WMS follows the matchmaking principle pioneered by Livny [41]. As other high performance space-shared systems, most EGEE sites implement their scheduling policies through multiple FIFO queues and complex tuning of configuration files. The matchmaking operates as follows: each site publishes structural (eg number of machines, OS) or dynamic (eg Expected Response Time) information related to Computing Elements through the glite Information System. As a rough approximation, it can be considered that a Computing Elements (CE) stands for a site's queue in the operational version of the Grid Information Model (we skip here the fundamental issues about the semantics of a CE analyzed in [16]). The set of CEs fulfilling the job requirements is first selected, then the ties are arbitrated along the lowest Expected Response Time. As shown for instance in [37,26], a wide spectrum of scientific communities make use of EGEE/EGI, each with its specificities. A common feature is that the EGEE/EGI infrastructure targets High Throughput Computing, not large scale parallelism (intra node parallelism such as multicore and GPU, as well as intra-cluster parallelism, are out of the scope of this discussion). As this workload is indeed representative of grid usage in the flagship european infrastructure, the associated generative models, and specifically the rupture process, might be of interest as a new category of inputs for the models of classical scheduling of parallel jobs such as proposed in [40].

### 3.2 Workload Definition

In grid context, workload is the equivalent of *backlog* in queuing systems terminology. Backlog at time  $t$  has two definitions a) the amount of unfinished work in the system and b) the delay that a job arriving at time  $t$  would experience before starting execution. Our interpretation is the first one. Formally, let  $T_a(j)$  be the arrival date of job  $j$  at a CE,  $T_s(j)$  the date where job  $j$  starts running, and  $T_e(j)$  the date where job  $j$  finishes. The cumulative running time of jobs that are accepted by the CE up to time  $t$  is

$$C^{\text{RA}}(t) = \sum_{j:T_a(j)<t} T_e(j) - T_s(j).$$

The cumulative running time of jobs that are started by the system up to time  $t$  is

$$C^{\text{RS}}(t) = \sum_{j:T_s(j)<t} T_e(j) - T_s(j).$$

The remaining running time of jobs that are started by the system and not yet finished is

$$R^{\text{R}}(t) = \sum_{j:(T_s(j)<t) \wedge (T_e(j)>t)} T_e(j) - t.$$

The workload at time  $t$ ,  $W(t)$ , is the total running time of jobs that were accepted by CE and waiting to start plus the remaining running time of jobs already running and not finished yet.

$$W(t) = C^{\text{RA}}(t) - C^{\text{RS}}(t) + R^{\text{R}}(t).$$

This definition implicitly assumes a homogeneous intra-CE system, by not referencing the dispatch algorithm. In fact, the actual running time of jobs, as observed in the logs, depends on the capacities of the machine on which they ran, thus on the dispatch system, except if the machine panel is fully homogeneous. The homogeneity assumption would be grossly erroneous at the grid scale; as we will consider the time series individually for each CE, it is acceptable: the grid sites are institutional ones, with reasonable coherency.

### 3.3 The data set

This study is based on exhaustive information covering all the gLite monitored jobs in the EGEE grid, from August 2008 to March 2009. The data is collected by the Real Time Monitor project, and is available through the Grid Observatory portal [12].

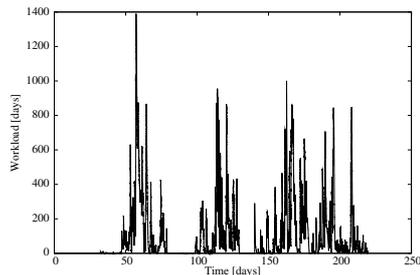
Significant preprocessing was required for building the workload process. First, jobs that fail to be dispatched are reported with a zero time-stamp, and were excluded. Second, and more importantly, as in any real-world large-scale experiment, measurements may in exceptional cases not be accurate. For instance, [57] reports a situation where the LogMonitor service become clogged, and is not consistent with the time-stamps provided by the Local Resource Management System (LRMS) service. However, as LRMS information for the entrance in the queue is not available, we choose to use the uniform reporting system provided by LogMonitor. Therefore, an outlier detection procedure had to be applied in order to remove artifacts. Common knowledge in

**Table 1** Descriptive statistics for the top CEs

	Total [years]	Jobs K	Average [hours]	StdDev [hours]	percentile [hours]		
					$q_{25\%}$	$q_{50\%}$	$q_{90\%}$
CE-A	151.4	551	2.8	10.0	0	10.0	303
CE-B	103.8	87	11.8	1331.0	16.0	1331.0	3999
CE-C	81.9	205	3.6	26.0	0	26.0	408
CE-D	58.4	336	2.1	0.2	0	0.2	203
CE-E	51.6	184	2.9	2.8	0	2.8	150
CE-F	49.1	155	3.0	0.6	0	0.6	87
CE-G	44.7	209	2.8	0.0	0	0.0	73
CE-H	44.6	217	2.5	0.1	0	0.1	78
CE-I	42.9	132	3.5	3.6	0	3.6	83

the EGEE community is that execution times longer than one day should be considered suspicious. Comparison of the LRMS data and LogMonitor data confirmed this intuition, leading to an exclusion threshold of one day.

Table 1 presents the statistics of the nine CEs featuring the largest total load (the real names of the CEs are omitted for staying on the safe side with respect of privacy and legal issues). The first column is the total execution time summed over the considered sample. The average, standard deviation, and percentiles refer to the workload (i.e.  $W(t)$ ). All criteria for very high variability (variance, interquartile range, maximum) are met. For instance, the standard deviation is between 1 and 3.5 times as large as the mean. Moreover, variability as expressed by the standard deviation is positively correlated with the median (correlation coefficient 0.98) and the mean (correlation coefficient 0.99). Similar results are true for the interquartile range.

**Fig. 1** Time series of the workload for CE-A

Visual inspection of the workload time series indicates that this variability is not uniform, but corresponds to different regimes. Fig. 1 shows a 1,400-day burst at day 60. Similar but lower, peaks repeat afterwards, and the trace shows an irregular alternation of quiet and loaded segments.

The discrete time series of the workload,  $W_t$ , is obtained by sampling  $W(t)$ . The sampling frequency should be high enough to make it possible for the analysis to provide practically useful output, but should remain close to the operational timescale of the analyzed system. The average number of job arrivals within an hour is in the range of 22 – 138 for the four top CEs. This suggests a sampling frequency in the order of  $10^{-3}$  –  $10^{-4}$  Hz because 1) each value of the differenced series cover several hundred jobs on

average, and 2) the practical timescale of interest was the behavior of the system in the order of hours and above.

The time series  $W_t$  cannot be reasonably modeled by a piecewise stationary process. The trend of  $W_t$  for instance is not piecewise constant. In this case, a standard methodology in time series analysis consists in differentiating  $W_t$  to remove the trend. Hence, in the following, our series of interest is defined by  $Y_t = W_t - W_{t-1}$ .

## 4 Fitting and validating piecewise models

### 4.1 Model fitting using MDL

Model (1)–(2) is characterized by the breakpoint number  $m$ , the locations  $\tau_1, \dots, \tau_m$  and the parameters  $\alpha_1, \dots, \alpha_{m+1}$  where  $\alpha_j = (\gamma_j, \phi_{j,1}, \dots, \phi_{j,p_j}, \sigma_j)$ .

Fitting model (1)–(2) to the data  $y = (y_1, \dots, y_n)$  consists in finding the “best” vector  $(m, \tau_1, \dots, \tau_m, \alpha_1, \dots, \alpha_{m+1})$ . This is a model selection problem in which possible models may have different number of parameters. [6] applied the Minimum Description Length (MDL) principle of [43] to solve this problem. The main goal of their procedure, referred to as Auto-PARM, is to find a model that can be used to perform lossless encoding of the data with as few bits as possible. The code applied in Auto-PARM encodes the data in two additive parts, one corresponding to the parameters of the fitted piecewise AR model, the other being related to the residuals of the corresponding AR segments. The best model is the one that minimizes with respect to  $(m, \tau_1, \dots, \tau_m, p_1, \dots, p_{m+1})$  the code length defined by

$$CL = \log m + (m+1) \log n + \sum_{j=1}^{m+1} \left\{ \log p_j + \frac{p_j+2}{2} \log n_j + \frac{n_j}{2} \log(2\pi) + \frac{1}{2} \log(\det \widehat{V}_j) + \frac{1}{2} (\underline{y}_j - \widehat{\underline{m}}_j)' \widehat{V}_j^{-1} (\underline{y}_j - \widehat{\underline{m}}_j) \right\},$$

where  $n_j = \tau_j - \tau_{j-1}$  is the number of observations in the  $j$ th regime,  $\widehat{V}_j$  is the covariance matrix with size  $n_j$  of the process  $\{X_{t,j}\}$  in (2) where the vector of parameters  $\alpha_j$  is replaced by the Gaussian maximum likelihood estimate  $\widehat{\alpha}_j$ ,  $\underline{y}_j = (y_{\tau_{j-1}}, \dots, y_{\tau_j-1})'$  is the vector of observations in the  $j$ th piece in (1),  $\widehat{\underline{m}}_j$  is the vector of size  $n_j$  with all elements equal to  $\widehat{m}_j$ , the maximum likelihood estimate of the mean  $m_j$  of the  $j$ th piece. Note, that here the natural logarithm is used instead of the base 2 logarithm, that is, the code length corresponds to nats and not bits.

Using the standard approximation to the likelihood for Gaussian AR models, the code length becomes

$$CL = \log m + (m+1) \log n + \sum_{j=1}^{m+1} \left\{ \log p_j + \frac{p_j+2}{2} \log n_j + \frac{n_j}{2} \log(2\pi \widehat{\sigma}_j^2) \right\},$$

where  $\widehat{\sigma}_j^2$  is the Yule-Walker estimate of  $\sigma_j^2$ . The interest of using Yule-Walker estimate is that an efficient implementation through the Levinson algorithm is available. Minimizing the objective function  $CL$  requires a tradeoff between the number of breakpoints and the complexity of the segments: segments that extend over different regimes will tend to require higher order AR models, and more variability.

The search space for breakpoints is very large, and the optimization problem is ill conditioned. [6] proposed to tackle the optimization problem by a genetic algorithm, which encodes a solution as a set of chromosomes of size  $n$  bearing the orders of the AR pieces at the selected breakpoints. This encoding is further constrained, so that the length of the segment is large enough to provide good estimates to the parameter of the related AR process (*min\_span* parameter), and to limit the order of the process. Termination is decided by empirical convergence (identical best chromosome along a fixed number of generations) or when a pre-defined number of iterations is reached. To limit the computational complexity, crossover is allowed only inside sub-populations, with periodic migration across the islands. The running time is characterized by  $M$ , the number of migrations.

## 4.2 Validation Methodology

The MDL-based AutoPARM procedure optimizes a target function that captures both the segmentation (number and locations of the breakpoints), and AR models inside each segment. More generally, while MDL-based segmentation methods consistently produce excellent experimental results, their theoretical properties are still largely unknown. The validation methodology is thus largely empirical. In particular, AutoPARM has been shown experimentally to be able to correctly detect change of regimes in series which are piecewise, but not AR on each segment. This fact gives two objectives for the validation methodology. First, the segment models and the breakpoints locations should be checked independently. Second, we seek indicators that are detailed enough to capture the potentially differentiated correctness of the model in various locations. For instance, the Mean Squared Error, or any other cumulative indicator, including the code length, does not reveal which segments are correctly modeled. The indicators should also be concise enough to provide a quantitative measure of correctness.

### *Asymptotic properties*

A few general results provide a guarantee about the consistency of estimators minimizing the code length  $CL$ . When a process satisfies model (1)–(2) and the number  $m$  of breakpoints is known, [6] established that the estimated break fraction  $\hat{\lambda}_j = \hat{\tau}_j/n$  converge a.s. to its true value for  $j = 1, \dots, m$ . The consistency of the estimate of  $m$  seems difficult to establish, since even in the independent case, the consistency is established in only some special cases, see e.g. [56]. Even when  $m$  is known, the asymptotic distribution of  $\hat{\lambda}_j$  is unknown, and thus no confidence intervals for the breakpoint locations are available.

### *Model Accuracy*

In model (1)–(2), the residuals in each segment should be iid. Several tests are available, see e.g. [5, 45], and here we use the classical Ljung-Box test. Thus, the indicator is simply the p-values of the test on each segment. The null hypothesis is that the neighboring residuals are uncorrelated, thus the larger (null hypothesis not rejected) the better. The Ljung-Box test is a portmanteau test: it tests the existence of autocorrelation in a time series at some lags. Being an asymptotic test, it is reliable when the size of the data set is large enough. The estimates of the correlation are asymptotically

Gaussian, whether the residuals are Gaussian or not. As we have no reason to restrict the distribution of the residuals, this is a good point. However, for small segments, the asymptotic behavior might not be relevant.

### *Model Stability*

The next question is the stability of the fitted model: is the model robust - i.e. more or less invariant - against the various sources of randomness involved? The internal randomization of the genetic algorithm is the first source. The second one is intrinsic to the data, which should be considered as a particular realization of a random process, for multiple reasons: the grid schedulers randomly break ties; measurements very likely include possible transient errors; and finally, within their general goals and objectives, the users' behavior cannot be considered fully deterministic. This intrinsic randomness is captured by the innovations of the model (the  $\epsilon_t$  in (2)).

The first issue is the definition of the distance in the models space. In principle, a classical distance between distributions such as the Kullback-Leibler one should be computed. However, first, a formal computation is intractable in practice, and would thus have to be performed numerically; second, more information can be gained by separately checking the components of the model. The synthetic indicators will be  $m$  itself and its standard deviation, and the average and standard deviation of the orders and code length. As our main interest is the segmentation, we will examine in more details the breakpoints, through a complete distribution of their distances.

To go beyond the synthetic indicators, we evaluate the stability of the segmentation w.r.t the intrinsic randomization through *bootstrapping* [10]. This procedure creates  $k$  samples of the piecewise AR model, namely the breakpoint locations and the parameter vectors; the size of each sample is  $n$ , the size of the original series. Then, *Auto-PARM* is applied to each sample. These  $k$  segmentations provide a distribution and associated statistics for the model parameters.

## 5 Experimental results

### 5.1 Experimental setting

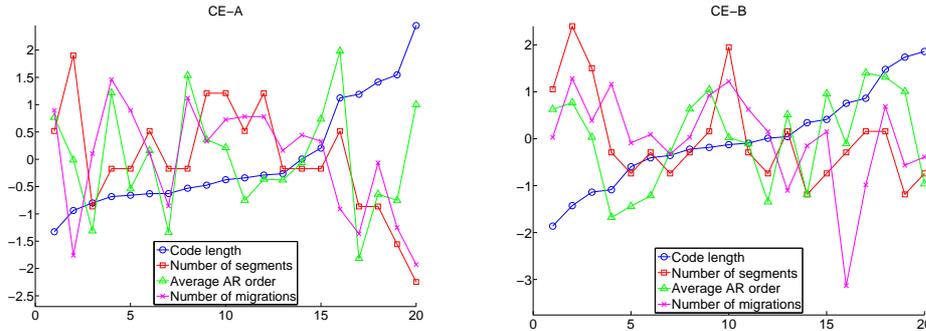
AutoParm features internal randomization (decision on mutation etc.). Thus, for each experiment, the procedure is repeated 20 times (*restarts*) and the results providing the smallest description length is selected. The parameters of the Genetic Algorithm (GA) are as follows: 100 islands of size 50, the 2 best chromosomes on island  $n$  migrates to island  $(n + 1) \bmod 100$  at every 5<sup>th</sup> offspring. The convergence criterion is the stability of the overall best chromosome for 10 consecutive migrations, and was met in all experiments. The complexity of the optimization landscape translates to a high computational complexity: one hour is typically required for one model selection.

### 5.2 The optimization landscape

The repeated runs (restarts) of AutoParm provide a first approximation of the optimization landscape for each data set. A complete sensitivity analysis would have to

**Table 2** Mean and standard deviation (bracketed) of the model parameters and algorithm indicators over the restarts of the GA.  $m$  is the Number of Segments,  $\bar{p}$  is the average AR order,  $CL$  is the Code Length,  $M$  is the number of migrations.

	CE-A	CE-B	CE-C	CE-D
$m$	20.25 (1.41)	27.65 ( 2.17)	29.60 (1.96)	20.65 (1.68)
$\bar{p}$	1.57 (0.31)	2.12 (0.40)	1.95 (0.35)	1.49 (0.45)
$CL$	2.04E+04 (4.45E+01)	2.17E+04 (6.56E+01)	1.84E+04 (7.55E+01)	1.75E+04 (3.93E+01)
$M$	128 (17.7)	150 (16.8)	156 (26.8)	118 (16.8)



**Fig. 2** Results of each restart for CE-A and B . The horizontal axis is the restart number ordered by increasing  $CL$ , the vertical axis corresponds to all parameters after rescaling

run experiments with different initialization values; due to the high computational cost of the method, we focus on the internal randomization.

Table 2 presents the summary statistics, over the restarts, for the four CEs. The values both for the algorithm indicators (code length and number of migrations) and for the model parameters (number of breakpoints and AR order) are clearly consistent for each experiment, and this holds for the four experiments.

Fig. 2 plots the detailed results for CE-A and B, together with the number of migrations. The values have been translated (to zero average) and rescaled (to unit variance), in order to visualize the trends on the same graph. The restarts have been ordered by increasing  $CL$ , thus the leftmost points are the best fits. The rightmost (worst) five restarts for CE-A show a significantly larger code length, together with a smaller number of segments and a smaller number of migrations. In these cases, AutoParm gets soon stuck into sub-optimal solutions where the variance of the noise is high. This confirms the need for the restart procedure. From this point, the results are reported only for the best restart.

### 5.3 The models

#### 5.4 The piecewise AR models

In this section, we go through the results of one run of AutoParm on CE-A and CE-B, which correspond to two different modes of grid usage. Table 3 gives the parameters of the models. For CE-A, the first result is that low-order AR models are the most frequent: seven segments are white noises (*i.e.* AR(0)) and six are AR(1). White noises

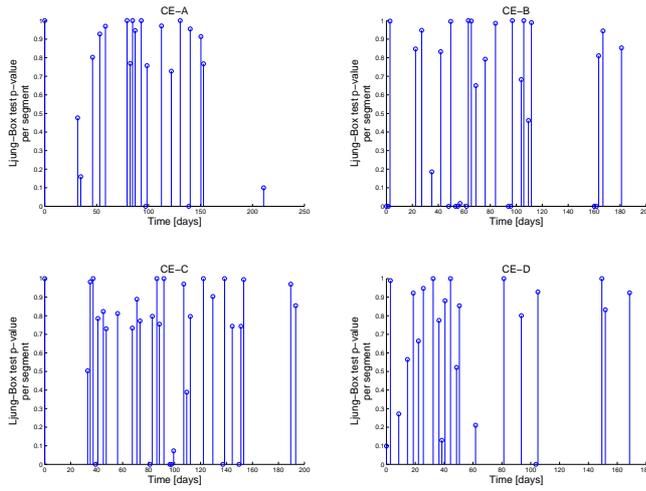
totalize 49% of the whole measurements. These weak correlations, and the fact that the estimated variance for most segments is very high, typically twenty times larger than the mean, can be interpreted as the result of a poor, but effectively mixing, load balancing policy, or as an intrinsic feature of the job arrival process. It is important to notice that the size of the corresponding segments is large enough to have authorized for a much higher order (e.g. the *min\_span* parameter is 20 for order 6). Segments 18, 19 and 20 actually exhibit higher orders (respectively 6, 5 and 5), showing that the procedure is able to fit more correlated models when adequate. CE-B involves much more long jobs than CE-A and the resulting workload model is more complex, both with respect to the number of breakpoints (30 instead of 21), and to the AR orders: for instance, the third segment is AR(13), indicating a correlation with three days old load. Nonetheless, the order of half of the segments is 0 or 1, comforting the diagnostic of a weakly correlated load.

**Table 3** The model parameters for CE-A, B, C and D  $j$ : segment index,  $n_j$ : segment length,  $p_j$ : AR order,  $\gamma_j$ : segment mean

CE-A				CE-B			
$j$	$n_j$	$p_j$	$\gamma_j$	$j$	$n_j$	$p_j$	$\gamma_j$
1	274	0	0.00E+00	1	14	0	4.98E+05
2	26	0	-5.98E+02	2	12	0	0.00E+00
3	98	0	5.98E+01	3	171	13	2.96E+05
4	60	2	2.93E+04	4	42	2	-1.07E+06
5	47	1	1.69E+05	5	68	2	-2.71E+05
6	180	3	-3.18E+04	6	60	0	3.86E+05
7	26	0	0.00E+00	7	54	3	-3.96E+05
8	20	2	-2.40E+01	8	15	0	0.00E+00
9	21	0	0.00E+00	9	33	4	4.78E+03
10	51	7	5.68E+01	10	16	0	0.00E+00
11	36	2	0.00E+00	11	16	3	2.41E+04
12	12	1	-6.99E+00	12	44	1	-3.85E+03
13	120	1	3.18E+03	13	13	1	1.35E+05
14	82	1	4.82E+04	14	21	2	0.00E+00
15	74	1	-3.66E+04	15	31	5	8.92E+05
16	71	0	0.00E+00	16	63	4	-3.16E+05
17	12	0	-3.94E+01	17	70	1	3.36E+05
18	89	6	3.61E+02	18	86	2	-2.52E+05
19	22	5	3.17E+03	19	17	3	-6.82E+05
20	500	5	-4.68E+03	20	12	1	-4.22E+05
21	74	1	-2.55E+03	21	60	0	-1.56E-01
				22	18	0	6.08E+05
				23	32	5	0.00E+00
				24	21	1	-3.73E+04
				25	418	5	2.66E+04
				26	17	1	-1.82E+05
				27	15	1	2.10E+01
				28	30	8	-2.98E+05
				29	124	1	6.97E+04
				30	49	2	-2.95E+05

## 5.5 Accuracy

Fig. 3 shows the p-values of the Ljung-Box test for the whiteness of the residuals considering the whole segments. Larger p-value indicate that the hypothesis of independence (which we hope to be confirmed) is more likely not to be rejected. The segments too short to run the tests are omitted (1% of the measurement time for CE-A and 9% for CE-B). The p-values are typically far from 0 therefore the whiteness hypothesis cannot be rejected for most of the segments. Nevertheless, the results for a number of segments lead to the rejection of the whiteness hypothesis at the 5% significance level. In these cases, the AR model might be an approximation of a more complex model, or the test might be limited by the small size of the segment.



**Fig. 3** Independence of the residuals: the Ljung-Box test

**Table 4** Fraction of the trace covered by segments with hypothesis of uncorrelated residuals not rejected at significance level  $1 - \alpha$

$\alpha$	5%	10%	20%	$\geq 50\%$
CE-A	41.1%	47.4%	50.6%	11.7%
CE-B	52.3%	64.1%	74.7%	16.1%
CE-C	45.2%	48.7%	68.0%	9.1%
CE-D	16.7%	50.2%	74.6%	18.7%

As there is no obvious relationship between the AR order of the segments and the test results, it is unlikely that the MDL method is in this case biased against high order models. There is some relationship with the length of the segments. In fact, a detailed analysis shows that the segments where the p-value is below 5% are in all cases except one those of small size (12 to 17 data points). With so few data, the results given by a portmanteau test are likely to be unreliable. The exception is segment 12 in CE-B, with 44 data points.

In Table 4 column 5% (resp. 10% and 20%) contains the fraction of the total length of the time-series formed by the segments for which the p-value is above or equal .95 (resp. .90 and .80); column  $\geq 50\%$  contains the sum of the length of the segments for which the p-value is less than .50. Except for CE-A, the p-value of the test results is over 0.80 for the largest part of the traces.

Overall, the whiteness of residual is clearly confirmed for the large segments, with only one exception, where the hypothesis of an approximation of a more complex model is valid.

## 6 Bootstrapping

The previous results show that the piecewise AR model adequately describes a significant part of the experimental data. The question is now if the descriptions are not exceedingly accurate: would a small change in the experimental data induce significant

changes in the model? In this case, the procedure would have over-fitted the data. Yet the motivations for possible variability are multiple, as described in 4.2.

Evaluating stability would require other samples of the load process but, of course, no other realization of the experimental data is available. To cope with this difficulty, we used the classical *bootstrapping* technique, which creates a sample of mock realizations of the process. In general, bootstrapping [10] is the technique which resamples from original data with replacement, assuming that the experimental data faithfully describe the population.

## 6.1 Method

Given the size, lack of homogeneity, and intrinsic correlation structure of the series (which is precisely the motivation for the piecewise model), naive resampling (randomly selection data points within the time series) would not create a valid realization, because it would destroy these inherent correlations.

Two methods have been considered for bootstrap: random generated iid Gaussian innovations and residual resampling. Often, the noise terms in an AR model are assumed to be iid Gaussian random variables. In this case, the bootstrapping would be fully parametric: after extracting the model parameters, the random model is generated by using normally distributed random noise. However, as there is no empirical reason to restrict the residuals to be Gaussian, the risk of creating artificial errors is real. We thus bootstrap through *resampling residuals*, where the residuals are resampled from the original ones. More precisely, the procedure is as follows

1. Extract the parameters (mean, AR parameters, noise variance) of an AR model with a given order from the workload measurement.
2. Compute the residuals.
3. Randomly resample the innovations from the set of residuals on each segment, and generate a new realizations from the AR model and the resampled innovations.
4. Each of the bootstrapped realizations is then segmented with the AutoParm procedure, with restarts.

The final result is an ensemble of models

$$\mathcal{S} = \{m_i, (\tau_i^j), (p_i^j), 1 \leq i \leq k, 1 \leq j \leq m_i\}$$

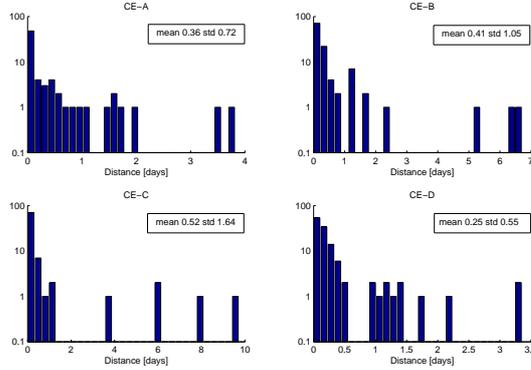
where  $k$  is the number of samples,  $m_i$  is the number of breakpoints in sample  $i$ ,  $\tau_i^j$  the  $j$ -th breakpoint in sample  $i$  and so on.

More precisely, ten series are bootstrapped from the initial model. Then, each of these series is in turn segmented 20 times, and for each of them, the model providing the smallest code length is selected. The output is thus ten "best fit" models, for which the numbers and locations of the breakpoints, and the AR orders, are compared. These ten best fits will be called *samples* in the following.

The ten best models correspond to ten different randomly generated series, and these model share only the piecewise AR model that was used to produce the input samples. Thus the conclusions from the bootstrapping analysis focus on the intrinsic sensitivity of the piecewise AR model to perturbations, and not on the sensitivity of the model fitting procedure.

**Table 5** Mean and standard deviation (bracketed) of the model parameters over the bootstrapped samples:  $m$  is the Number of Segments,  $\bar{p}$  is the average AR order.  $c$  is the cutoff parameter for hierarchical clustering, see text.

	CE-A	CE-B	CE-C	CE-D
$m$	19.70 (0.95)	25.50 (1.51)	20.60 (1.50)	20.50 (1.08)
$\bar{p}$	1.22 (1.48)	1.24 (1.22)	0.88(1.22)	1.16 (1.56)
$c$	25	25	25	30



**Fig. 4** Bootstrap with residual resampling: histogram of the distances between nearest neighbor breakpoints in the samples from the four CEs. Note the log scale on the vertical axis.

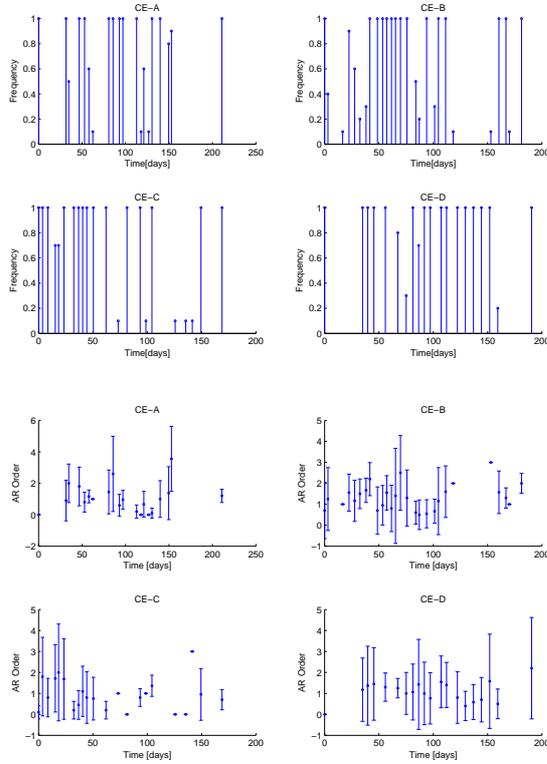
## 6.2 Stability

The first two lines in table 5 show the basic statistics of the model parameter over the bootstrapped samples. The segmentation itself (number of breakpoints) appears remarkably stable, with a standard deviation in the order of 5% of the mean. Bootstrapping also confirms the low values of the AR orders, but with a larger variability.

Breakpoints defined by  $\mathcal{S}$  can be very close, but not identical across the bootstrapped samples. For instance, in CE-A, some samples provide the segment [86.57–92.94], while other provide [86.80, 92.94] (the unit is the day). These segments should be considered as variants of the same one. This is by far the most frequent case. On the other hand some samples find a continuous segment (93 data points) between 112.38 and 121.42, while a few other samples feature a breakpoint in the range 117.82- 119.79, roughly halving the number of data points, and denoting a true disagreement between the segmentation results.

In order to evaluate more precisely the stability of the segmentation, we compute the distances between each breakpoint and its closest neighbor. This distance is 0 if two or more samples feature exactly the same breakpoint, or strictly positive if a breakpoint is present in only one sample. Fig. 4 shows the distribution of these distances when the unit is the day. The distribution is strongly dominated by low values of the distance. When the unit is the sampling period, more than 97% of the distances are less than 10 points, which is in fact the resolution of the method, as 10 points are the lower bound for fitting the simplest AR models, and the mean is close to 1, with a standard deviation less than 10 in all cases. The segmentation can be thus considered stable.

## 6.3 Clustered models



**Fig. 5** The clustered models. Upper graphs: frequency of the breakpoints. Lower graphs: AR order  $\pm$  one standard deviation.

In order to define a final model for each CE, firstly, the close breakpoints, which are variants, must be clustered. The method is hierarchical clustering with centroid linkage; the cutoff is defined so as to produce the number of clusters closest to the average (across samples) number of breakpoints for this CE, as shown in table 5. The median of the breakpoints (*m-breakpoint*) in each cluster represents the cluster.

Fig. 5 (upper graphs) displays the frequency of the *m*-breakpoints after clustering, as the fraction of the samples that feature a breakpoint within the cluster represented by this *m*-breakpoint. Most *m*-breakpoints are recognized in all samples, as expected from the distance distributions. Fig. 5 (lower graph) shows the mean and standard deviation of the AR order. In many cases, the consensual *m*-breakpoints (those selected by all samples) are associated with small AR-orders variability: for instance, the last *m*-breakpoint in CE-A, -B, and -C, or the first consensual *m*-breakpoint after 100 in CE-A. However, some AR-orders exhibit significant variability.

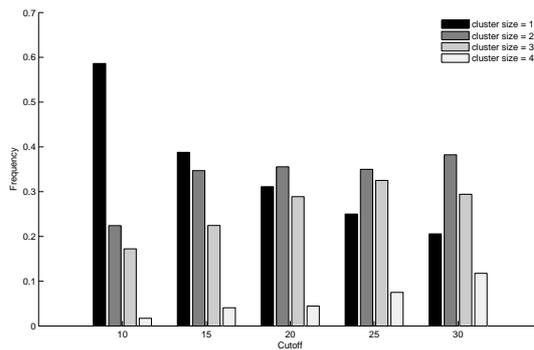
The limits of the robustness of the models derived from MDL principle regarding the AR order selection (the wide confidence intervals in fig. 5) can be explained by

the relative flatness of the optimization landscape in this direction. The part of the description length formula considering the AR orders has a rather small contribution to the overall code length.

In order to build a more robust model AR orders have to be reconciliated. Bootstrapped aggregation, or *bagging* [3] gives theoretical foundations to model reconciliation, either by averaging or voting. Here, the number of models is bound to be small due to computational time, thus voting should be preferred [28]. The choice of the best voting strategy is (and is likely to remain) an open question; in our case, the simple majority voting may be used, with a random choice for breaking ties.

#### 6.4 A global segmentation

Intuitively, the grid behavior should exhibit some level of coupling, due to the emergent collective behavior of its users and the complex middleware feedbacks. Do the segmented models bear witness of this coupling? Here too, some clustering must first be performed, to take into account the local features of the CEs, e.g. time zone. There is not much guide here to decide on the cutoff  $c'$ , except of course that it should not re-cluster breakpoints within each CE. The resulting clusters of breakpoints will be called *global clusters*. Fig 6 shows the distribution of the sizes of the global clusters for various values of the cutoff  $c'$ . At  $c' = 25$ , 75% of the breakpoints are shared by at least two CEs, and 40% by at least three, showing a correlated dynamics.



**Fig. 6** Distribution of the sizes of global clusters along the cutoff.

## 7 Conclusion

This paper makes a first step to more realistic modeling strategies for large-scale distributed systems: relinquish the stationarity hypothesis. Physical explanations support stationarity for some phenomena, e.g. in hydrology, the Nile floods that stemmed the concepts and analysis of long-range dependence. But stationarity should not be taken as a given, and appears in fact as an implausible feature for e-science infrastructures,

still in the ramp-up phase, and which support a so-called chaotic component. Segmenting the data flow into stationary segments is thus a preliminary for principled statistical analysis and detailed modeling.

A possible approach for segmentation, that has been explored here, is MDL-based model selection. We modeled the workload of the four most heavily loaded CEs in the EGEE/EGI grid over nearly a year. Because little is known theoretically besides basic convergence guarantees concerning the quality of the estimated model parameters, we proposed and illustrated the successive steps of empirical validation. It includes non-parametric bootstrapping for analyzing the stability of the model. Considering the specific case of EGEE/EGI, we show that its workload can be explained by piecewise autoregressive models to a large extent, that the orders of the models are mostly low to moderate, and that they show a significant level of inter-sites correlated activity.

The most significant limitation of the method is the complexity of the objective function to optimize: it includes not only the segmentation itself, but the details of the model (in our case the AR orders). This complexity, coupled with the poor scalability of the genetic algorithm with respect to the length of the time series, makes the segmentation analysis a heavyweight computing task. The substitution of an alternative evolutionary optimization algorithm (CMA-ES), associated to the same MDL principle, is explored in another paper and shows some promise. However, an agnostic (w.r.t. intra-segment model) approach, in the spirit of clustering, would be required to achieve real-time analysis.

**Acknowledgements** This work was partially funded by the DIM program of Region Ile de France and the Digiteo Foundation and by the France-Grilles initiative. The data sets have been provided by the Grid Observatory. The Grid Observatory is part of the European Grid Initiative.

## References

1. J. Beran. *Statistics for long-memory processes*, volume 61 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, New York, 1994.
2. R. N. Bhattacharya, V. K. Gupta, and E. Waymire. The Hurst effect under trends. *J. Appl. Probab.*, 20(3):649–662, 1983.
3. L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2), 1996.
4. J. Brevik, D. Nurmi, and R. Wolski. Predicting bounds on queuing delay in space-shared computing environments. In *IISWC*, pages 213–224, 2006.
5. P. Burns. Robustness of the Ljung-Box test and its rank equivalent. In *The Journal of Derivatives*, pages 7–18, 2002.
6. R. A. Davis, T. Lee, and G. Rodriguez-Yam. Structural break estimation for nonstationary time series models. *J. American Statist. Assoc.*, 101:229–239, 2006.
7. F. X. Diebold and A. Inoue. Long memory and regime switching. *J. Econometrics*, 105(1):131–159, 2001.
8. P. A. Dinda and D. R. O’Hallaron. Host load prediction using linear models. *Cluster Computing*, 3(4):265–280, 2000.
9. A. B. Downey. Using queue time predictions for processor allocation. In *IPPS’97, JSSPP’97*, pages 35–57, 1997.
10. B. Efron. Bootstrap. another look at jackknife. *Ann. Statist.*, 7(1):1–26, 1979.
11. T. Elteto, C. Germain Renaud, P. Bondon, and M. Sebag. Discovering Piecewise Linear Models of Grid Workload. In *10th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing*, pages 474–484, 2010.
12. C. Germain-Renaud et al. The Grid Observatory. In *11th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing*, 2011.
13. E. Laure et al. Programming the Grid with gLite. *Computational Methods in Science and Technology*, 12(1):33–45, 2006.

14. F. Gagliardi et al. Building an Infrastructure for scientific Grid computing: status and goals of the EGEE project. *Phil. Trans. of the Royal Society A*, 1833, 2005.
15. M. Lassnig et al. Identification, Modelling and Prediction of Non-periodic Bursts in Workloads. In *10th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing*, pages 485–494, 2010.
16. S. Andreozzi et al. Glue Schema Specification, V1.3. Technical report, Open Grid Forum, 2008.
17. P. Fearnhead. Exact Bayesian curve fitting and signal segmentation. *IEEE Trans. Signal Process.*, 53(6):2160–2166, 2005.
18. C. W. J. Granger and N. Hyung. Occasional structural breaks and long memory with an application to the S&P 500 absolute stock returns. *J. Empirical Finance*, 11(3):399–421, 2004.
19. C. W. J. Granger and R. Joyeux. An introduction to long-memory time series models and fractional differencing. *J. Time Ser. Anal.*, 1(1):15–29, 1980.
20. J. R. M. Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, 1981.
21. M. C. Huebscher and J. A. McCann. A survey of autonomic computing: degrees, models, and applications. *ACM Comput. Surv.*, 40:7:1–7:28, August 2008.
22. R. Gott III. Implications of the copernican principle for our future prospects. *Nature*, (363):315–319, 1993.
23. L. Ilijašić and L. Saitta. Characterization of a computational grid as a complex system. In *Procs. of GMAC '09*, pages 9–18, 2009.
24. S. Jha, M. Parashar, and O. Rana. Investigating autonomic behaviours in grid-based computational science applications. In *Proceedings of GMAC'09*, pages 29–38, 2009.
25. G. Kitagawa and H. Akaike. A procedure for the modeling of non-stationary time series. *Ann. Inst. Statist. Math.*, 30(1):351–363, 1978.
26. A. Lagana and al. COMPCHEM: Progress towards gems a grid empowered molecular simulator and beyond. *Journal of Grid Computing*, 8:571–586, 2010.
27. B.-D. Lee and J. M. Schopf. Run-time prediction of parallel applications on shared environments. In *CLUSTER*, pages 487–491, 2003.
28. T-W Lee and Y. Yang. Bagging binary and quantile predictors for times series. *Journal of econometrics*, 135(1-2):465–497, 2006.
29. H. Li and M. Muskulus. Analysis and modeling of job arrivals in a production grid. *SIGMETRICS Perform. Eval. Rev.*, 34(4):59–70, 2007.
30. D. Lingrand, T. Glatard, and J. Montagnat. Modeling the latency on production grids with respect to the execution context. *Parallel Computing*, 35(2009):493–511, 2009.
31. M. Macias, O. Rana, G. Smith, J. Guitart, and J. Torres. Maximising revenue in grid markets using an economically enhanced resource manager. *Concurrency and Computation: Practice and Experience*, 2008.
32. J. Meng, S. T. Chakradhar, and A. Raghunathan. Best-effort parallel execution framework for recognition and mining applications. In *IPDPS*, pages 1–12, 2009.
33. N. Mi, G. Casale, L. Cherkasova, and E. Smirni. Injecting realistic burstiness to a traditional client-server benchmark. In *Proceedings of ICAC '09*, pages 149–158, 2009.
34. T. N. Minh, L. Wolters, and D. Epema. A Realistic Integrated Model of Parallel System Workloads. In *10th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing*, pages 464–473, 2010.
35. A. Mutz, R. Wolski, and J. Brevik. Eliciting honest value information in a batch-queue environment. In *Proceedings of GRID '07*, pages 291–297, 2007.
36. F. Nadeem, M. M. Yousaf, R. Prodan, and T. Fahringer. Soft benchmarks-based application performance prediction using a minimum training set. In *e-science'06*, 2006.
37. Special issue:EGEE applications and supporting grid technologies. *Journal of Grid Computing*, 8(3), 2010.
38. J. Perez, C. Germain-Renaud, B. Kégl, and C. Loomis. Utility-based reinforcement learning for reactive grids. In *The 5th IEEE ICAC Autonomic Computing*, 2008.
39. J. Perez, C. Germain-Renaud, B. Kégl, and C. Loomis. Multi-objective reinforcement learning for responsive grids. *Journal of Grid Computing*, 8(3):473–492, 2010.
40. A. Pugliese, D. Talia, and R. Yahyapour. Modeling and supporting grid scheduling. *Journal of Grid Computing*, 6(2):195–213, 2008.
41. R. Raman, M. Livny, and M. Solomon. Matchmaking: Distributed resource management for high throughput computing. In *Procs 7th IEEE Int. Symp. on High Performance Distributed Computing*, pages 140–147, 1998.

- 
42. I. Rish, R. Das, G. Tesauro, and J. Kephart. Autonomic computing: A new challenge for machine learning, 2006. ECML/PKDD tutorial, available at [www.ecmlpkdd2006.org/tutorials.html](http://www.ecmlpkdd2006.org/tutorials.html).
  43. J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, 1989.
  44. E. Rogers. *Diffusion of innovations*. New York: Free Press, 1983.
  45. H.J. Skaug and D. Tjostheim. Testing for serial independence using measures of distance between densities. In P.M. Robinson and M. Rosenblatt, editors, *Athens Conference on Applied Probability and Time Series*, Springer Lecture Notes in Statistics 115, 1996.
  46. W. Smith, V. E. Taylor, and I. T. Foster. Using run-time predictions to estimate queue wait times and improve scheduler performance. In *IPPS/SPDP '99, JSSPP'99*, pages 202–219, 1999.
  47. O. Sonmez, N. Yigitbasi, A. Iosup, and D. Epema. Trace-based evaluation of job runtime and queue wait time predictions in grids. In *Proceedings of HPDC '09*, 2009.
  48. R. S. Sutton, A. G. Barto, and R. J. Williams. Reinforcement Learning is Direct Adaptive Optimal Control. In *American Control Conference*, pages 2143–2146, 1992.
  49. M. S. Taqqu and V. Teverovsky. Robustness of Whittle-type estimators for time series with long-range dependence. *Comm. Statist. Stochastic Models*, 13(4):723–757, 1997.
  50. G. Tesauro, N. K. Jong, R. Das, and M. N. Bennani. On the use of hybrid reinforcement learning for autonomic resource allocation. *Cluster Computing*, 10(3):287–299, 2007.
  51. Gerald Tesauro. Reinforcement Learning in Autonomic Computing: A Manifesto and Case Studies. *IEEE Internet Computing*, 11:22–30, 2007.
  52. V. Teverovsky and M. Taqqu. Testing for long-range dependence in the presence of shifting means or a slowly declining trend, using a variance-type estimator. *J. Time Ser. Anal.*, 18(3):279–304, 1997.
  53. D. Thain, J. Bent, A. Arpacı-Dusseau, R. Arpacı-Dusseau, and M. Livny. Gathering at the well: Creating communities for grid i/o. In *Procs of Supercomputing*, 2001.
  54. R. Wolski, N. T. Spring, and J. Hayes. Predicting the cpu availability of time-shared unix systems on the computational grid. *Cluster Computing*, 3(4):293–301, 2000.
  55. L. Yang, J. M. Schopf, and I. Foster. Conservative scheduling: Using predicted variance to improve scheduling decisions in dynamic environments. In *SC '03: Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, page 31, 2003.
  56. Y.-C. Yao. Estimating the number of change-points via Schwarz' criterion. *Statistics & Probability Letters*, 6(3):181–189, 1988.
  57. X. Zhang, C. Furtlehner, J. Perez, C. Germain-Renaud, and M. Sebag. Toward autonomic grids: analyzing the job flow with affinity streaming. In *Proc. of the 15th ACM SIGKDD*, pages 987–996, 2009.