Modeling Job Arrivals in a Data-Intensive Grid

Hui Li[1,⋆], Michael Muskulus[2], and Lex Wolters[1]

[1] Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
hui.li@computer.org
[2] Mathematical Institute, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands

Abstract. In this paper we present an initial analysis of job arrivals in a production data-intensive Grid and investigate several traffic models to characterize the interarrival time processes. Our analysis focuses on the heavy-tail behavior and autocorrelation structures, and the modeling is carried out at three different levels: *Grid*, *Virtual Organization (VO)*, and *region*. A set of m*-state Markov modulated Poisson processes (MMPP)* is investigated, while *Poisson processes* and *hyperexponential renewal processes* are evaluated for comparison studies. We apply the *transportation distance* metric from dynamical systems theory to further characterize the differences between the data trace and the simulated time series, and estimate errors by *bootstrapping*. The experimental results show that MMPPs with a certain number of states are successful to a certain extent in simulating the job traffic at different levels, fitting both the interarrival time distribution and the autocorrelation function. However, MMPPs are not able to match the autocorrelations for certain VOs, in which strong deterministic semi-periodic patterns are observed. These patterns are further characterized using different representations. Future work is needed to model both deterministic and stochastic components in order to better capture the correlation structure in the series.

1 Introduction

Performance evaluation of computer systems, such as comparing different scheduling strategies on parallel supercomputers, requires the use of representative workloads to produce dependable results [9,13]. On single parallel machines, a significant amount of workload data has been collected [33], characterized [23,27], and modeled [7,25,41]. Benchmarks and standards are also proposed for workloads in evaluations of parallel job schedulers [6].

In a production Grid environment, however, few work has been done because the Grid infrastructure is still emerging and it is difficult to collect traces at the Grid level. Let us take the LHC Computing Grid (LCG) [21] as an example. The LCG testbed currently has approximately 180 active sites with a total number of 24,515 CPUs and 3 Petabytes storage, which is primarily used for high-energy physics data processing. Resource brokering or superscheduling in such an environment is challenging given the fact that Grid schedulers do not have control over the participating resources. In

[⋆] Corresponding author.

E. Frachtenberg and U. Schwiegelshohn (Eds.): JSSPP 2006, LNCS 4376, pp. 210–231, 2007.

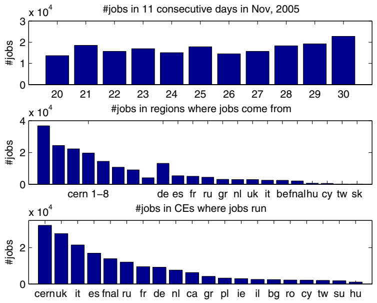

Fig. 1. Job distribution (cern - EU Center for Nuclear Research, fnal - Fermi Lab, the rest are country domain names)

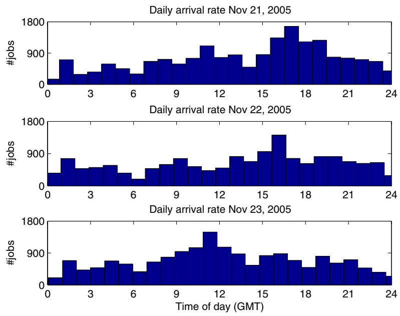

Fig. 2. Daily arrival rate in three consecutive days in November, 2005. The time is in Greenwich Mean Time (GMT).

such contexts different scheduling and resource management systems have been proposed [31]. The current scheduling system deployed in LCG is a distributed version of the centralized resource broker, which originated in the EU DataGrid. It has multiple resource broker instances distributed in different regions/countries [11]. The Virtual Organization (VO) based scheduler with usage SLAs is proposed in a similar computing environment with similar workloads [10]. The evaluations of these different superscheduling architectures and strategies require proper workload models at different levels.

In this paper we present an initial analysis and modeling of Grid job arrival patterns. Our data is obtained via the Real Time Monitor [36] in the LCG production Grid. Our analysis focuses on the heavy-tail behavior and autocorrelations of job arrival processes. The modeling is carried out at the Grid, VO, and region level for facilitating evaluations of different scheduling strategies. A set of m-state Markov modulated Poisson processes (MMPP) is investigated for modeling, while Poisson processes and hyperexponential renewal processes are also evaluated for comparison. We apply the transportation distance metric from dynamical systems theory [28] to further characterize the differences between the data trace and the simulated time series.

The rest of the paper is organized as follows. Section 2 describes the workload, analyzes the daily arrival rate and summary statistics from different VOs and users, and presents the self-similarity measurements in terms of the Hurst parameter and the autocorrelation function (ACF). Section 3 introduces the selected traffic models and describes how to estimate parameters for each model. The transportation distance metric as an analysis tool is also presented. Section 4 presents the detailed modeling of job arrivals at the Grid, VO, and region level. The goodness of models are evaluated by the interarrival time distribution, the autocorrelation function and transportation distance of simulated traces. Section 5 discusses related work in the analysis and modeling of arrival processes in a broader perspective. Conclusions and future work are presented in Section 6.

2 Statistical Analysis

2.1 Workload Description

As mentioned above, LCG is a worldwide production Grid developed and operated for physics data processing. Almost all the jobs are trivially parallel tasks, requiring one CPU to process certain amount of data. Most of the jobs come from multiple large-scale physics experiments, such as *lhcb*, *cms*, *atlas* and *alice*. These experiments are also named as Virtual Organizations (VOs), in which users worldwide participate. The computing and storage resources define local sharing policies based on VOs and users. At the meta level workloads are managed and routed to resources via resource brokers (RBs), which do the matchmaking for jobs and try to balance the load at a global level.

There are resource brokers distributed over the Grid by regions, such as one in Germany, one in the UK, and so on. A majority of jobs come from CERN in Switzerland and there are around eight RB instances at CERN to share the workloads. The Real Time Monitor developed by Imperial College London [36] monitors jobs from all the major RBs in the LCG testbed, therefore the trace data it collects is representative at the Grid level. The job characteristics includes VO name, user DN (Distinguished Name), RB name, UI (User Interface), CE (Computing Element), submission time, run time and status. These attributes enable us to categorize, analyze and model job arrivals at different levels.

The LCG Real Time Monitor was in operation since October, 2005 and we use a period of eleven consecutive days (from Nov 20th to 30th, 2005) without missing data[1] in this study. Figure 1 shows the number of jobs in each day, number of jobs coming from different regions, and number of jobs in CEs where jobs get executed. We can see that a total number of 188,041 jobs distributed quite evenly over the period. More than 75% of jobs come from User Interfaces at CERN while the rest originated in around twenty different countries. The workloads are routed by resource brokers to computing resources in more than twenty countries, in which jobs are distributed in quite different orders than job origins. Job turnaround times are frequently used as the metric for the resource brokers to rank resources after matchmaking.

2.2 Job Arrival Analysis

Figure 2 shows the daily arrival rate in three consecutive days (GMT) on LCG in November, 2005. As we can see at the Grid level there are no clearly observable daily patterns, which are evident on single parallel machines [7,23,25]. Jobs are scattered in daily hours more evenly with peaks in the middle day or in the afternoon. The even distribution of jobs is explainable by the fact that users are simultaneously active across different time zones in the Grid. The peaks in the middle day or in the afternoon are mainly attributed to users at CERN, who submit a majority of jobs during the period under study.

Figure 3 and 4 show the number of jobs submitted by VOs and users. There is an interesting pattern that the job distribution for VOs can be fitted by an exponential

[1] Only jobs submitted to RBs are recorded and those who directly go to the Computing Elements are not available.

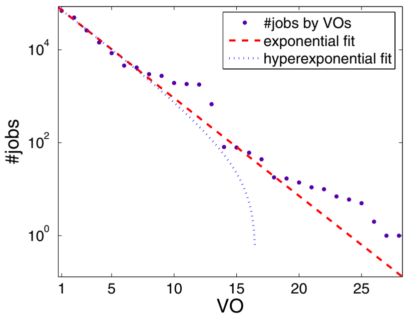

Fig. 3. Number of jobs submitted by VOs

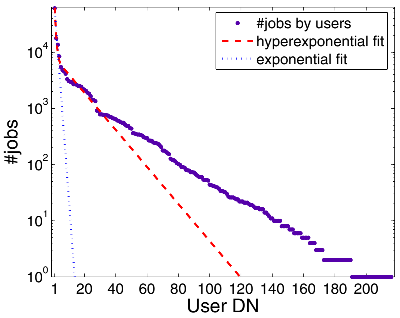

Fig. 4. Number of jobs submitted by users

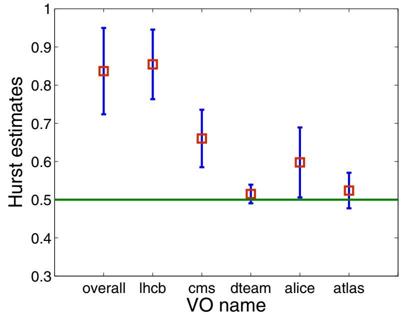

Fig. 5. Estimates of Hurst parameters for interarrival times

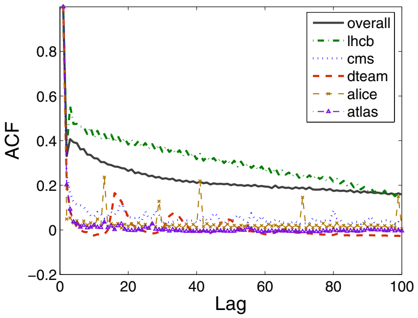

Fig. 6. Autocorrelation functions (ACFs) of interarrival times

function quite well. The top five VOs, namely *lhcb*, *cms*, *dteam*, *alice*, and *atlas*, submit almost 90% of the total number of jobs. The job distribution for users decreases even more sharply and a two-phase hyperexponential function has a better fit. The top 10% of users contribute to 90% of the whole workload and the top three account for 50%. This type of pattern is also observed in many social and physical phenomena, such as database transactions and Unix file sizes [13]. It is argued in [2] that it essentially originates in a priority selection mechanism between tasks and non-tasks waiting for execution. From a modeling perspective this pattern makes the VO an appropriate level for categorization since the limited number of main components represent most of the workloads.

2.3 Self-similarity

Self-similarity means that a process looks statistically the same over a wide range of different scales and is closely related to so-called "bursty" behavior and long range dependence [4]. The degree of the self-similarity of a stochastic process can be summarized by the Hurst parameter ($0 < H < 1$). A value of $H > 0.5$ indicates self-similarity with positive near neighbor correlation and the more H is close to 1, the more self-similar the process. As there is no consensus on how to best estimate the Hurst parameter, we use three estimation techniques, namely *R/S statistic, variance plot, Periodogram*, and try to find agreement among them[2]. Figure 5 shows the means and standard deviations of Hurst parameter estimates of the interarrival time processes for the Grid trace and different VOs. We can see that the overall Grid job arrivals are self-similar with $H \approx 0.84$. The VO *lhcb* is also strongly self-similar with the Hurst parameter reaching 0.85. The other VOs show moderate to weak self-similarity. These observations are also confirmed if we look at the autocorrelation function (ACF) of interarrival times, illustrated in Figure 6. Strongly self-similar processes (*overall, lhcb*) have a longer memory than the weakly self-similar counterparts (*dteam, atlas*), whose ACFs quickly approach zero as the lag increases. Autocorrelation is used as one of the statistical properties to measure the goodness of fit in the following sections.

3 Methodology

Job traffic can be mathematically described as a *point process*, which consists of a sequence of arrival instances. Two equivalent descriptions of point processes are *counting processes* and *interarrival time processes* [5,17]. In this paper we describe the traffic using the interarrival time process, sometimes also called the embedded process. Based on the analysis of job arrivals, several basic principles can be derived for model selection. Firstly, models should be parameterizable and flexible enough to represent the Grid job traffic at different levels. Secondly, models must be able to approximate both the interarrival time distribution (heavy-tail behavior) and the autocorrelation function. Thirdly, models should be analytically simple and there should exist proven methods to estimate their parameters from the data trace. Bearing these points in mind, we investigate a set of m-state Markov modulated Poisson processes to model job arrivals. Phase-type renewal processes and Poisson processes are also evaluated for comparison. We discuss the selected models and their corresponding parameter estimation methods in this section. The recently proposed transportation distance metric for the comparison of two time series is presented as a tool to further characterize the goodness of fit.

3.1 Markov Modulated Poisson Processes

A Markov modulated Poisson process (MMPP) is a doubly stochastic Poisson process whose intensity is controlled by a finite state continuous-time Markov chain (CTMC). Equivalently, an MMPP process can be regarded as a Poisson process varying its arrival rate according to an m-state irreducible continuous time Markov chain. Following the notations in [14], an MMPP parameterized by an m-state CTMC with infinitesimal generator Q and m Poisson arrival rates Λ can be described as

$$Q = \begin{bmatrix} -\sigma_1 & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & -\sigma_2 & \cdots & \sigma_{2m} \\ \cdot & \cdot & \cdots & \cdot \\ \sigma_{m1} & \sigma_{m2} & \cdots & -\sigma_m \end{bmatrix}, \qquad (1)$$

[2] Estimations of the Hurst parameters are calculated using a self-similarity analysis tool called SELFIS [19].

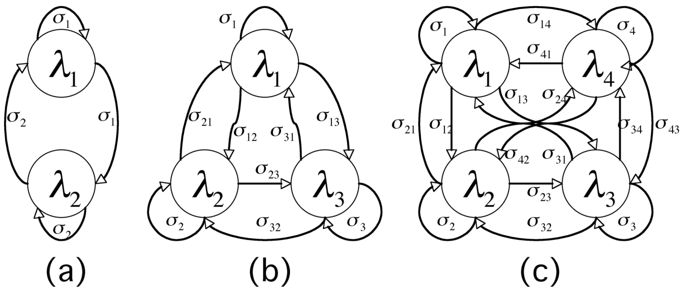

Fig. 7. MMPP models of state 2, 3, and 4, respectively

$$\sigma_i = \sum_{j=1, j \neq i}^{m} \sigma_{ij}, \tag{2}$$

$$\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_m). \tag{3}$$

MMPPs with state 2, 3, and 4 are illustrated in figure 7. The MMPP model is commonly used in telecommunication traffic modeling [16,17] and has several attractive properties, such as being able to capture correlations between interarrival times while still remaining analytically tractable. We refer to [14] for a thorough treatment of MMPP properties as well as its related queuing network models.

A natural problem which arises with the applications of MMPPs is how to estimate its parameters from the data trace. In [37] methods based on moment matching and maximum likelihood (MLE) are surveyed and it is proven that MLE methods are strongly consistent. In [38] Ryden proposed an EM algorithm to compute the MLE estimates of the parameters of a m-state MMPP. Recently, Roberts et al. improved Ryden's EM algorithm and extended its applicability in two important aspects [35]: firstly a scaling procedure is developed to circumvent the need for customized floating-point software, arising from the exponential increase of the likelihood function over time; secondly, evaluation of integrals of matrix exponentials is facilitated by a result of Van Loan, which achieves significant speedup. We implemented the improved version of Ryden's EM algorithm in Matlab and this is by far the best MLE estimator that we can find for m-state MMPPs. Given the difficult numerical issues involved, estimation errors could still be substantial, though. It should also be mentioned that the estimation for higher order MMPPs is increasingly difficult, since there are more parameters to take into account.

3.2 Hyperexponetial Renewal Processes

In a renewal process the interarrival times are independently and identically distributed but the distribution can be general. A Poisson process is characterized as a renewal process with exponentially distributed interarrival times. In phase-type renewal processes the interarrival times are distributed in so-called phase-type, e.g. as a n-phase hyperexponential distribution. In theory any interarrival distribution can be approximated by phase-type ones, including those which exhibit heavy-tail behavior [34].

However, a major modeling drawback of renewal processes is that the autocorrelation function (ACF) of the interarrival times vanishes for all non-zero lags so they cannot capture the temporal dependencies in time series. Unlike the renewal models, MMPPs introduce dependencies into the interarrival times so they can potentially simulate the traffic more realistically with non-zero autocorrelations.

There are special cases where an MMPP is a renewal process and the simplest one is the Interrupted Poisson Process (IPP). The IPP is defined as a 2-state MMPP with one arrival rate being zero. Stochastically, an IPP is equivalent to a 2-phase hyperexponential renewal process. Following the formulations in [14] the IPP can be described as

$$Q = \begin{bmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda & 0 \\ 0 & 0 \end{bmatrix}, \tag{4}$$

and the 2-phase hyperexponential distribution (H_2) has the density function

$$f_{H_2}(t) = p\mu_1 e^{-\mu_1 t} + (1-p)\mu_2 e^{-\mu_2 t}. \tag{5}$$

The parameters of H_2 can be transformed to parameters of IPP by

$$\lambda = p\mu_1 + (1-p)\mu_2, \tag{6}$$

$$\sigma_1 = \frac{p(1-p)(\mu_1-\mu_2)^2}{\lambda}, \tag{7}$$

$$\sigma_2 = \frac{\mu_1\mu_2}{\lambda}, \tag{8}$$

while the H_2 parameters (p, μ_1, μ_2) can be obtained from the data by applying an EM algorithm as described in [1], whose implementation is freely available [12].

3.3 Transportation Distance of Time Series

Coming from a dynamical systems theory background, Moeckel and Murray have given a measure of distance between two time series [28] that, from a time series perspective, excellently analyzes (short-time) correlations. It is based on recent research on nonlinear dynamics [18,3]. Given a time series, the data is first discretized, i.e. binned, with a certain resolution (a parameter of the method), and then transformed into points in a k–dimensional discrete space, referred to as the reconstruction space, using a unit-delay embedding. In dimension 2, for example, all $n-1$ consecutive pairs (x_i, x_{i+1}), $1 \leq i < n$, of n given data points thus constitute a point $y_i = (x_i, x_{i+1})$ in the reconstruction space. The idea is, that the essential dynamics of generic systems can usually be reconstructed sufficiently in a low dimensional space. The normalized k–dimensional probability distributions of these data points from the two series will then be considered as a transportation problem (also called a minimum cost flow problem): What is the optimal way, given the first probability distribution, to arrive at the second, just by transporting weight, i.e. probability, from some boxes to some others? With each movement a transportation cost is given, which is the normalized (by mass) taxi–cab distance from the first box to the second, measured in units of the discretization size[3], which is given by the resolution parameter of the method. The minimal such transportation cost can be computed by linear programming. We have written some code to generate a linear program from two time series which then will be fed into a specialized minimum-cost flow solver[4]. For details on linear programming, the transportation problem and algorithmic improvements, we refer to [39].

The transportation distance measures to which extent two given time series show the same k–correlation structure, and is thereby quite sensitive to (1) correlations, and (2) the underlying probability distributions. It is robust against small perturbations and outliers, too. A value of the transportation distance can be roughly interpreted as the average distance each data point of the first time series lies from a corresponding point in the second series.

Table 1. Transportation distances in dimension 1, i.e. for single interarrival times, between real data and simulated series of fitted Poisson, m-MMPP and IPP models. The time resolution is 10s intervals. All entries are normalized to mean taxi-cab distance (with a unit of 10s). Values depicted are bootstrap means and standard mean error, estimated by bootstrapping 50 times.

Level	Name	Poisson	IPP	MMPP2	MMPP3	MMPP4
Grid	lcg	0.039 ± 0.001	0.029 ± 0.001	0.024 ± 0.001	0.035 ± 0.001	0.058 ± 0.001
VO	lhcb	0.35 ± 0.01	0.35 ± 0.01	0.47 ± 0.01	0.50 ± 0.01	0.54 ± 0.01
	cms	1.35 ± 0.01	0.40 ± 0.01	0.81 ± 0.01	0.70 ± 0.01	5.34 ± 0.01
	dteam	4.57 ± 0.02	1.03 ± 0.02	17.07 ± 0.05	21.65 ± 0.06	N/A
	alice	1.57 ± 0.02	0.98 ± 0.02	1.21 ± 0.02	3.28 ± 0.02	3.36 ± 0.03
	atlas	16.38 ± 0.19	6.54 ± 0.15	56.94 ± 0.29	47.70 ± 0.50	5.49 ± 0.19
Region	cern	3.38 ± 0.02	0.78 ± 0.02	2.95 ± 0.02	2.53 ± 0.03	25.17 ± 0.08
	de	9.60 ± 0.09	3.77 ± 0.06	35.97 ± 0.14	43.73 ± 0.24	437.18 ± 0.95
	uk	28.91 ± 0.16	7.58 ± 0.10	95.83 ± 0.51	98.68 ± 0.46	N/A

[3] This is equivalent to considering all the points in each discrete box to be located at the center of their box.

[4] We use the *MCF* network simplex solver developed by Andreas Löbel [26], as well as the general purpose *lp_solve* linear programming solver [24] for comparing performance.

Table 2. Parameters of fitted Poisson, MMPP2 and IPP models as found by the EM algorithm

Level	Name	Poisson	MMPP2				IPP		
		λ	σ_1	σ_2	λ_1	λ_2	p	μ_1	μ_2
Grid	lcg	11.90	0.17	0.08	22.10	7.16	0.22	139.20	10.46
VO	lhcb	4.35	0.04	0.01	8.43	3.18	0.11	4.35	4.35
	cms	3.11	0.10	0.07	6.92	0.44	0.95	6.21	0.31
	dteam	1.64	0.83	0.08	17.86	0.10	0.91	18.31	0.17
	alice	2.38	0.16	0.06	6.67	0.73	0.78	6.79	0.71
	atlas	0.54	0.10	0.01	4.98	0.02	0.95	5.05	0.03
Region	cern	1.41	0.10	0.06	3.43	0.13	0.94	3.36	0.15
	de	0.83	0.17	0.03	4.98	0.03	0.94	5.08	0.06
	uk	0.19	0.36	0.01	4.93	0.03	0.75	5.82	0.05

Table 3. Transportation distances in dimension 2, i.e. comparing pairs of interarrival times, between real data and simulated series of fitted Poisson, m-MMPP and IPP models. The time resolution is 30 seconds. All entries are normalized to mean taxi-cab distance (with a unit of 30 seconds), and should therefore be about a factor of 3 smaller than the corresponding values in Table 1. Values depicted are bootstrap means and standard mean errors, estimated by bootstrapping 25 times.

Level	Name	Poisson	IPP	MMPP2	MMPP3	MMPP4
Grid	lcg	0.0038 ± 0.0001	0.0010 ± 0.0001	0.0139 ± 0.0001	0.0233 ± 0.0002	0.0035 ± 0.0001
VO	lhcb	0.179 ± 0.001	0.182 ± 0.001	0.244 ± 0.001	0.274 ± 0.001	0.295 ± 0.001
	cms	0.747 ± 0.004	0.394 ± 0.003	0.500 ± 0.004	0.458 ± 0.004	3.279 ± 0.008
	dteam	2.708 ± 0.012	1.141 ± 0.008	11.249 ± 0.029	14.285 ± 0.038	N/A
	alice	0.813 ± 0.011	0.661 ± 0.011	0.686 ± 0.011	1.936 ± 0.022	1.963 ± 0.018
	atlas	11.041 ± 0.123	5.601 ± 0.084	37.764 ± 0.175	31.906 ± 0.376	3.480 ± 0.099
Region	cern	2.174 ± 0.012	0.818 ± 0.010	1.917 ± 0.016	1.641 ± 0.023	16.674 ± 0.062
	de	6.080 ± 0.063	2.962 ± 0.039	24.007 ± 0.110	28.786 ± 0.196	290.859 ± 0.618
	uk	20.490 ± 0.108	8.504 ± 0.064	64.765 ± 0.370	65.414 ± 0.429	N/A

Unfortunately, the transportation distance is difficult to compute for higher lags, since the computational effort rises polynomially in the lag. We are working on approximation methods though, which might overcome this problem in the future [29].

Table 4. Error estimates for fitted MMPP2 model, standard mean errors have been estimated by bootstrapping 25 times with a geometrical blocksize distribution of mean length 100. Correlations between parameters have not been indicated.

Level	Name	MMPP2			
		σ_1	σ_2	λ_1	λ_2
Grid	lcg	0.262 ± 0.034	0.387 ± 0.064	17.300 ± 0.590	5.118 ± 0.291
VO	lhcb	0.632 ± 0.117	0.396 ± 0.153	9.051 ± 0.753	3.261 ± 0.093
	cms	0.106 ± 0.002	0.075 ± 0.001	6.833 ± 0.041	0.435 ± 0.015
	dteam	0.824 ± 0.016	0.079 ± 0.001	17.651 ± 0.211	0.100 ± 0.003
	alice	0.172 ± 0.004	0.069 ± 0.004	6.692 ± 0.022	0.728 ± 0.023
	atlas	0.102 ± 0.003	0.012 ± 0.001	5.020 ± 0.055	0.020 ± 0.001
Region	cern	0.099 ± 0.003	0.063 ± 0.002	3.436 ± 0.021	0.129 ± 0.003
	de	0.174 ± 0.006	0.035 ± 0.002	5.095 ± 0.064	0.032 ± 0.002
	uk	2.279 ± 0.261	0.128 ± 0.033	4.925 ± 0.530	0.054 ± 0.006

Table 5. Bootstrapped rate parameters of fitted MMPP3 model, standard mean errors have been estimated by bootstrapping 25 times with a geometrical blocksize distribution of mean length 100. Correlations between rates have not been indicated.

Level	Name	MMPP3		
		λ_1	λ_2	λ_3
Grid	lcg	1.979 ± 0.205	2.290 ± 0.209	13.812 ± 0.210
VO	lhcb	1.913 ± 0.135	2.092 ± 0.162	5.087 ± 0.150
	cms	0.257 ± 0.032	0.672 ± 0.099	7.098 ± 0.098
	dteam	0.046 ± 0.006	0.097 ± 0.040	15.901 ± 0.545
	alice	0.295 ± 0.038	0.537 ± 0.083	5.954 ± 0.152
	atlas	0.001 ± 0.054	0.163 ± 0.091	4.839 ± 0.174
Region	cern	0.094 ± 0.014	0.284 ± 0.052	4.050 ± 0.077
	de	0.015 ± 0.001	0.905 ± 0.124	6.321 ± 0.039
	uk	0.013 ± 0.002	0.039 ± 0.012	4.217 ± 0.334

3.4 Bootstrapping

Error estimates for arbitrary functions of stochastic variables can be produced by *bootstraping/resampling* [8] techniques. The finite data trace is thereby assumed to be a *realization* of an underlying probabilistic process, i.e. data points are assumed to be drawn randomly from a (usually unknown) probability density. Each data value is sampled with an empirical probability that converges to this density, in the limit of an infinite data trace. The size of the variations in finite traces can be estimated by looking at additional data traces of the same length, sampled from the same distribution. Bootstrapping methods achieve this by resampling from the observed data trace itself, i.e. instead of choosing data points randomly from the unknown true density, points are chosen by its approximation, the known empirical density.

Since the transportation distance compares two probability densities, error estimates for this measure can be produced by the bootstrap method easily. We have implemented

Table 6. Bootstrapped transition parameters of fitted MMPP3 model, standard mean errors have been estimated by bootstrapping 25 times with a geometrical blocksize distribution of mean length 100. Correlations between parameters have not been indicated.

Level	Name	MMPP3					
		σ_{12}	σ_{13}	σ_{21}	σ_{23}	σ_{31}	σ_{32}
Grid	lcg	1.25 ± 0.16	2.35 ± 0.40	0.24 ± 0.05	0.72 ± 0.10	0.06 ± 0.01	0.10 ± 0.02
VO	lhcb	1.07 ± 0.17	2.14 ± 0.30	0.29 ± 0.05	1.07 ± 0.17	0.12 ± 0.02	0.17 ± 0.03
	cms	0.33 ± 0.06	0.19 ± 0.03	0.53 ± 0.06	0.31 ± 0.06	0.13 ± 0.02	0.15 ± 0.02
	dteam	0.35 ± 0.05	0.13 ± 0.03	0.57 ± 0.07	0.18 ± 0.04	0.50 ± 0.06	0.51 ± 0.06
	alice	0.50 ± 0.06	0.27 ± 0.04	0.48 ± 0.05	0.31 ± 0.06	0.25 ± 0.04	0.27 ± 0.03
	atlas	0.35 ± 0.06	0.04 ± 0.01	0.71 ± 0.08	0.19 ± 0.05	0.22 ± 0.05	0.24 ± 0.03
Region	cern	0.43 ± 0.06	0.22 ± 0.05	0.56 ± 0.08	0.34 ± 0.06	0.19 ± 0.03	0.29 ± 0.04
	de	0.011 ± 0.001	0.016 ± 0.001	0.089 ± 0.006	0.053 ± 0.006	0.117 ± 0.001	0.049 ± 0.006
	uk	0.42 ± 0.05	0.10 ± 0.02	0.74 ± 0.08	0.17 ± 0.04	0.96 ± 0.13	0.95 ± 0.10

this method with 50 bootstraps of the same length in embedding dimension 1, and 25 in dimension 2, for each of the two time series fed into the distance algorithm. Results can be seen in Tables 1 and 3, where the bootstrap means and standard mean errors are shown. All of the results for the original series' distances lie scattered around the bootstrap means within one sampled standard deviation. This shows the appropriateness of the bootstrapping methodology, and we only give the bootstrap means in the tables for this reason.

For time series, where not only the distribution of values, but also the correlation structure is important, the simple bootstrap has to be replaced by more sophisticated methods. The block bootstrapping technique, developed by Künsch [20] and further analyzed in [32], instead of randomly choosing data points, randomly chooses sequences of consecutive points. The length of these *blocks* is again randomly chosen from a geometric distribution to smoothe boundary effects. We have applied this method with 25 bootstraps to the estimation of the MMPP model parameters by the EM algorithm. The mean block length has been chosen to be 100 interarrivals. Results can be seen in Table 4 for MMPP2, and Tables 5 and 6 for MMPP3, where we show bootstrap means and standard mean errors. Since there are strong correlations between parameters, these estimates have to be considered with some caution. This also explains the few discrepancies with the parameter estimation for the original data trace in Table 2.

4 Modeling

In a large-scale Grid environment different superscheduling architectures require modeling of job arrivals at different levels. By applying the methodology discussed above, we model the job traffic at the Grid, the Virtual Organization, and the region level, respectively in this section.

4.1 Grid Level

Figure 13 shows the fittings of the interarrival time in terms of complementary cumulative distribution function (CCDF) by five models, namely Poisson, IPP, MMPP2, MMPP3, and MMPP4. We can see that globally there is no heavy-tail behavior and all the models fit the job arrivals quite well. The transportation distances of dimension 1 given in Table 1 quantitatively measure the goodness of fit for interarrival time distributions. Since the values are all quite low, all models seem to reproduce correctly the probability distribution (1d), with MMPP2 being the best. The fittings of the autocorrelation function (ACF) of the interarrival time process are shown in Figure 14. As expected ACFs of Poisson and IPP vanish for all none-zero lags and they cannot capture the interdependencies of job arrivals. The MMPPs can introduce dependencies into the interarrival times, but they are not able to match the long memory of the original trace. By taking both CCDF and ACF into account we can conclude that MMPP2 is a better model for the Grid level job arrivals than the Poisson or IPP model. The transportation distances of dimension 2 given in Table 3 show the differences in pair correlations (2d), which are also quite small in value.

Figure 8 visually plots the sequences of interarrival times for the original trace and several models. We can see that both Poisson and IPP lack the kind of variability compared to the trace although their CCDFs fit quite well. MMPP2 looks more similar to the original data in terms of variability, therefore it can simulate the job traffic more realistically[5].

4.2 Virtual Organization Level

We model the five largest VOs, namely, *lhcb*, *cms*, *dteam*, *alice*, and *atlas*, in descendant order with respect to the number of jobs submitted. Figure 15 and 16 show the CCDFs and ACFs of the fitted models for the interarrival time process by *lhcb*. Being the largest VO in terms of the submitted jobs, *lhcb* has no heavy tail distribution of interarrivals and exhibits a long memory. It contributes significantly to the properties of overall Grid job arrivals shown in the last section. As to the models we can see that IPP produces identical fitting with Poisson. Both of them have slightly better results than MMPPs in terms of transportation distances of dimension 1 and 2. However, MMPP2 and MMPP3 have similar autocorrelations that come the closest to the original trace. Considering the tradeoffs, MMPP2 is selected as the best fit among the evaluated models. Clearly better

[5] This visual comparison should be replaced by objective, quantitative measures, of course, and this is exactly what the transportation distance achieves, when sufficiently high orders can be compared.

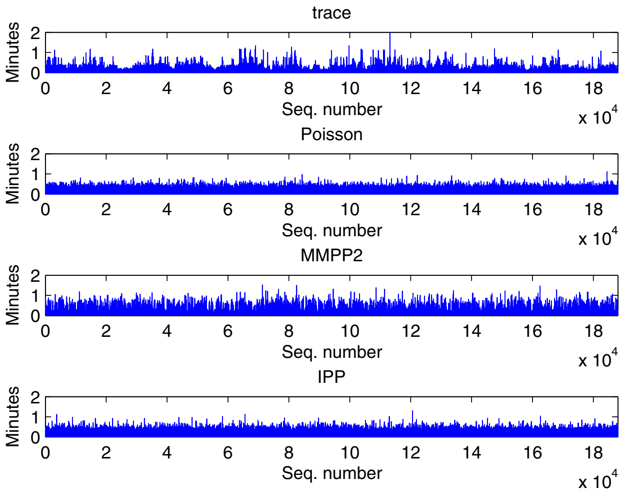

Fig. 8. Sequences of interarrival times of the Grid trace and the fitted models

models are needed to closely match the long memory in the series; we will elaborate why stochastic models fail to capture the autocorrelations in the coming sections, as well as indicate some future directions for research.

We observe that increasing the number of states in MMPPs would not necessarily improve the fitting. For instance, in the Grid and *lhcb* case MMPP4 is an overfitted model both in terms of CCDF and ACF. This phenomenon is seen with the transportation distance, too. It seems paradox at first, since MMPP4 is a more flexible model than MMPP3/MMPP2, but can be attributed to the following issues: (1) the parameter estimation by the EM algorithm does not easily give error estimates, so errors in the parameters could be substantial[6], (2) the data trace is finite, and actually rather small for fitting large interarrival times (which occur seldomly), (3) the compromise between fitting a lot of small interarrival times and some rarely occurring large events seems to favor the smaller times: there are too many large events generated by the higher order MMPPs, (4) there is a strong deterministic component in the *lhcb* data, as can be seen in Figure 9 and Figure 10 where we show the pair distribution for real data and simulated MMPP2 data. The large peak at about (24s, 24s) interarrival times is very difficult to model with a Poisson-based model, since waiting times in such models will always be from an exponential family, thereby monotonously decreasing with distance from the origin.

From Figure 17 to Figure 24 CCDF and ACF fittings are shown for the remaining four VOs. We can see that the less job submissions in the VO, the longer the tail the CCDF has. In those situations with heavy tails, the Poisson process fails to match the interarrival time distribution. For *cms* data with moderate interarrival time dependencies,

[6] In this respect, a Bayesian analysis by Monte-Carlo Markov Chain methods [40] would be desirable, since this would produce the *probability distribution* of the estimated parameters directly.

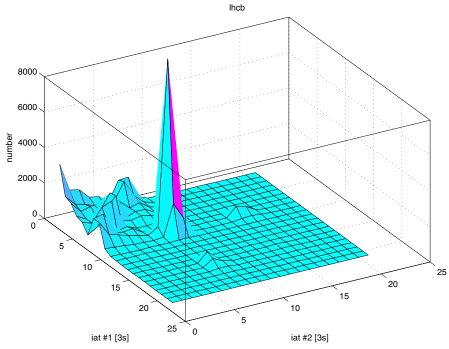

Fig. 9. Pairs of interarrival times for *lhcb*

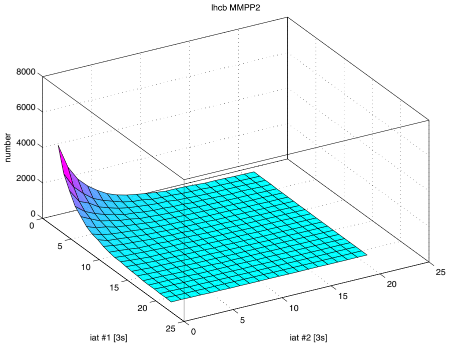

Fig. 10. Pairs of interarrival times for MMPP2

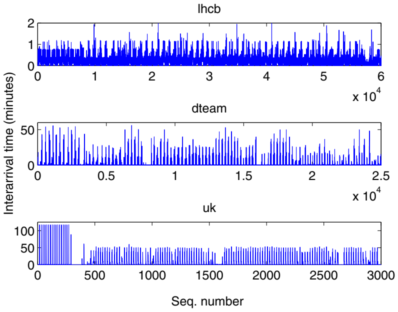

Fig. 11. Sequences of interarrival times for *lhcb*, *dteam*, and *uk*

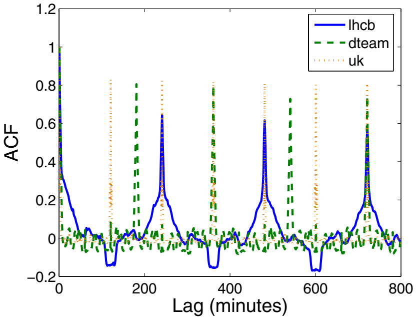

Fig. 12. Autocorrelation functions for *lhcb*, *dteam*, and *uk* from the binned count processes

we can see that MMPP3 has very good fittings for both CCDF and ACF (Figure 17 and 18). For *dteam*, MMPPs exhibit longer memory which is not present in the data and IPP is shown to be the most suitable model (Figure 19 and 20). For *alice* both MMPP3 and MMPP4 can model the interarrival process better than others (Figure 21 and 22), although they tend to generate too many large times[7]. In the last VO we studied, namely *atlas*, MMPP4 is shown to be the best fitted model. MMPP2 and MMPP3 have too long memories and cannot fit the interarrival time distribution closely, while IPP has no memory and fails to match the heavy tail of the data (Figure 23 and 24).

Although no general conclusions can be reached, some observations are found to be very interesting. As the VO size decreases from *lhcb* to *cms*, then to *alice* and *atlas*, the models with the best fit are MMPPs with an increasing number of states, from 2 to 3 then to 4, although deterministic components can complicate this. This observation suggests that MMPPs have very attractive properties for modeling job traffic in the VO level, being general and analytically simple. With the VO size decreasing in an exponential

[7] This can be seen from their transportation distances, for example, which are more sensitive to large data values than to small ones.

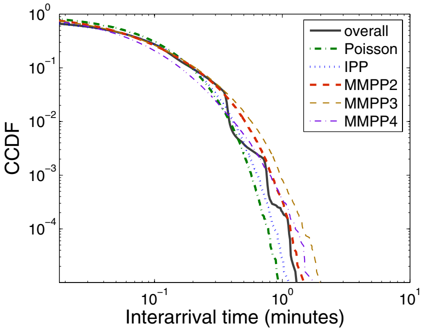

Fig. 13. Fitting the interarrival time distribution (CCDF) for the overall Grid job arrivals

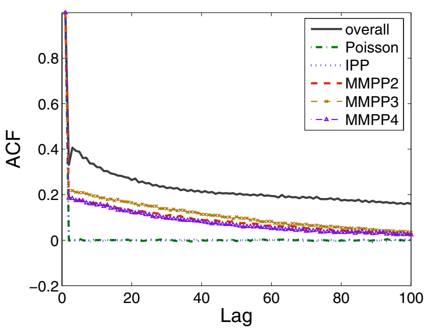

Fig. 14. Fitting the autocorrelation function (ACF) for the overall Grid job arrivals

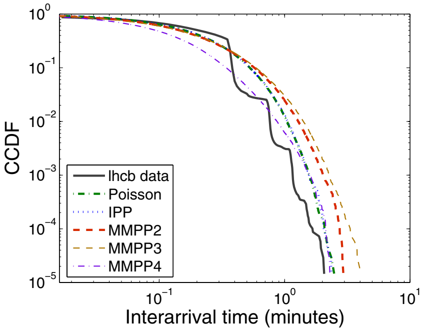

Fig. 15. Fitting the interarrival time distribution (CCDF) for job arrivals by VO *lhcb*

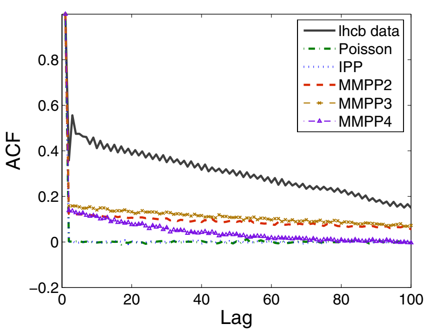

Fig. 16. Fitting the autocorrelation function (ACF) for job arrivals by VO *lhcb*

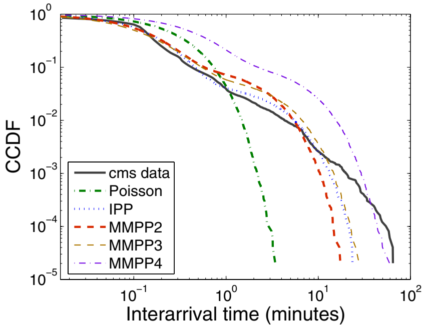

Fig. 17. Fitting the interarrival time distribution (CCDF) for job arrivals by VO *cms*

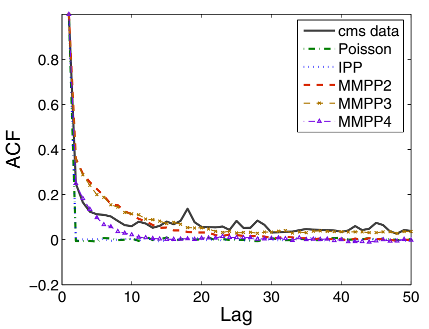

Fig. 18. Fitting the autocorrelation function (ACF) for job arrivals by VO *cms*

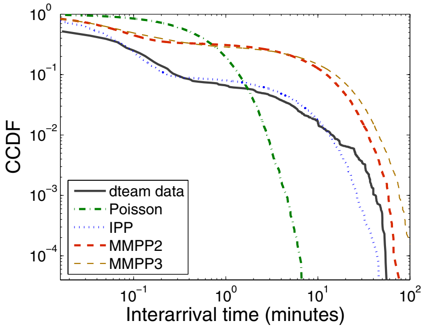

Fig. 19. Fitting the interarrival time distribution (CCDF) for job arrivals by VO *dteam*

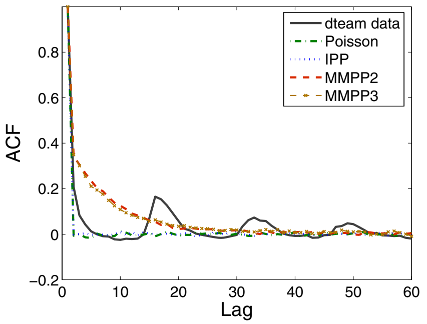

Fig. 20. Fitting the autocorrelation function (ACF) for job arrivals by VO *dteam*

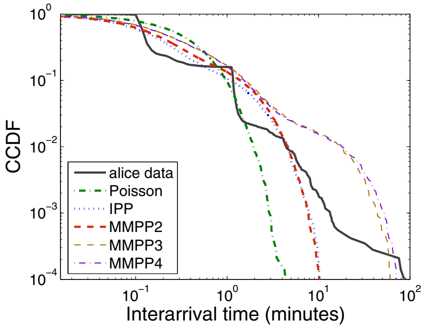

Fig. 21. Fitting the interarrival time distribution (CCDF) for job arrivals by VO *alice*

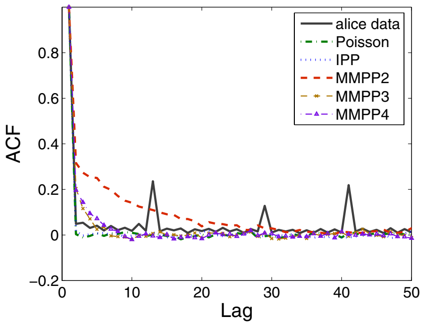

Fig. 22. Fitting the autocorrelation function (ACF) for job arrivals by VO *alice*

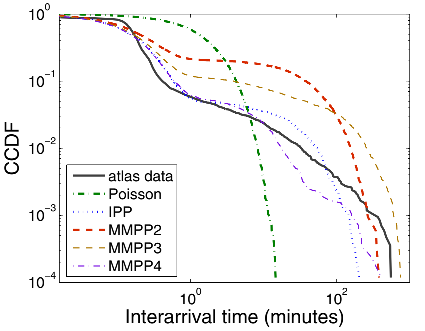

Fig. 23. Fitting the interarrival time distribution (CCDF) for job arrivals by VO *atlas*

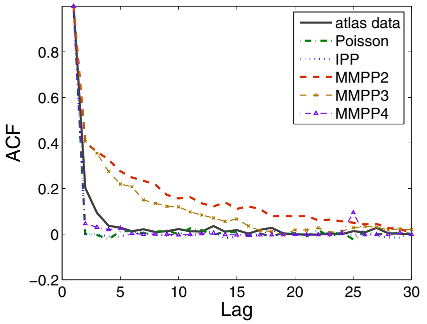

Fig. 24. Fitting the autocorrelation function (ACF) for job arrivals by VO *atlas*

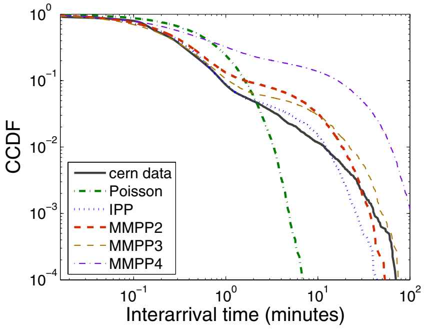

Fig. 25. Fitting the interarrival time distribution (CCDF) for job arrivals from *cern*

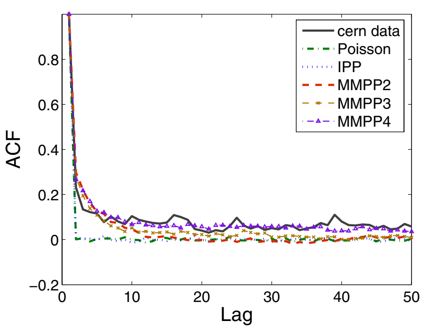

Fig. 26. Fitting the autocorrelation function (ACF) for job arrivals from *cern*

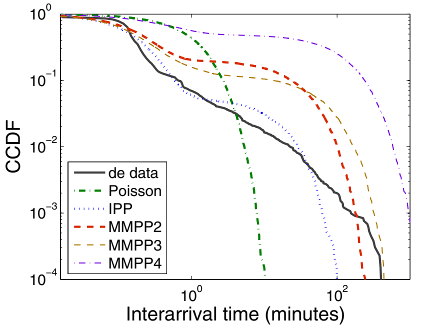

Fig. 27. Fitting the interarrival time distribution (CCDF) for job arrivals from *de*

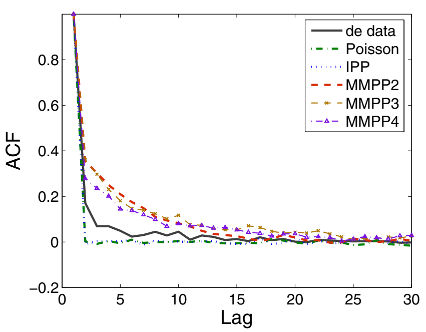

Fig. 28. Fitting the autocorrelation function (ACF) for job arrivals from *de*

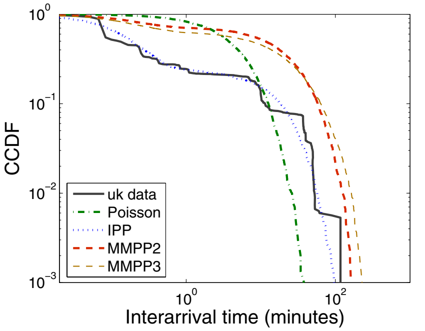

Fig. 29. Fitting the interarrival time distribution (CCDF) for job arrivals from *uk*

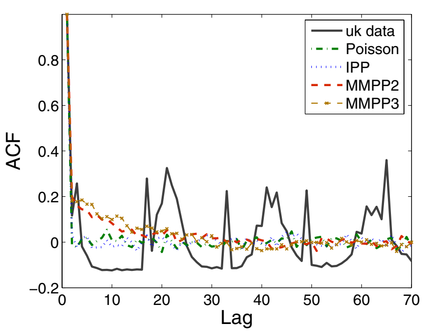

Fig. 30. Fitting the autocorrelation function (ACF) for job arrivals from *uk*

manner (see Figure 3), we can model the job arrivals of the corresponding VOs using MMPPs by increasing the number of states and/or further modeling of deterministic components. As a special case *dteam* will be discussed in detail in Section 4.4.

4.3 Region Level

Resource brokers in the current LCG testbed are distributed in regions, so it is important to model the job arrivals at the region level as well. Figure 25 to Figure 30 show the model fitting for *cern* (European Center for Nuclear Research), *de* (Germany), and *uk* (United Kingdom), respectively. Since a majority of jobs are originated in *cern* and routed by one of its eight resource broker instances, we use job arrivals by one randomly chosen resource broker in this study. From Figure 25 and 26 we can see that MMPP3 is the model with the best fit for *cern* data. MMPPs do not perform well for *de* and *uk*, introducing autocorrelations which are not observable in the real data. In these two cases IPP is shown to be the most suitable model, matching both the CCDFs and ACFs of the interarrival time processes.

4.4 Stochastic vs. Deterministic

In the modeling process, we find that for certain data such as *dteam* and *uk* the EM algorithm does not converge for estimating MMPP4 parameters (indicated by 'N/A' in the tables). This motivates us to plot the interarrival time sequences for all the data to see what kind of structures exist. The results are surprising: we find strong deterministic semi-periodic behavior for *lhcb*, *dteam* and *uk*. This is illustrated in Figure 11 and Figure 9. To further understand these patterns, we form a time series by counting number of jobs in intervals of 1 minute duration and plot the autocorrelation function (ACF) of this 'binned' counting process. Figure 12 shows these ACFs for the above mentioned data. The periodic behavior is clearly observed, with the period for *lhcb*, *dteam* and *uk* being 240 minutes, 180 minutes, and 120 minutes, respectively. For *dteam*, which stands for "deployment team", this pattern is explainable because jobs from this VO are mostly testing and monitoring jobs initiated by human or automatically by software. Jobs from *uk* during the period of study are mostly *dteam* jobs. It is interesting to see that the biggest VO *lhcb* also shows periodic behavior. If we take into account that close to 90% of *lhcb* jobs (around 60,000 jobs) are from one single user during the eleven days under study, we can assume that scripts are written to submit such production jobs, which are deterministic in nature.

We cannot say that the periodic behavior for large production VOs is a general feature and can be used in modeling. However, it is safe to assume that certain VOs are partly dedicated to testing and monitoring the Grid. In this case, for a realistic model to capture the behavior of such mixed deterministic (periodic) and stochastic components, we could follow the traditional route of time series analysis by either fitting and then subtracting the periodic components, or by introducing time-varying model parameters and change points [43].

5 Related Work

Traditionally, job arrivals have been analyzed and modeled on single parallel supercomputers. In [7] polynomials of degree from 8 to 13 are used to fit the daily arrival rates. In [25] a combined model is proposed where the interarrival times fit a hyper-Gamma distribution and the job arrival rates match the daily cycle. Time series models such as ARIMA are studied in [42], which try to capture the traffic trends and interdependencies. The impact of such models on the performance of parallel scheduling is also investigated.

The recent work by Medernach [27] is closely related to ours as he analyzes and models job arrivals on one cluster in LCG. The model developed is a ON-OFF Markov chain model, which essentially is a 2-phase hyperexponential renewal process (IPP). It is shown that for single users 2-phase hyperexponential distributions can fit the interarrival times well, although no analysis on dependencies of the series is available. As we model the job traffic at the VO and the Grid level, it can be regarded as a superposition of single user activities. It is well known that the superposition of individual renewal processes can be a correlated, nonrenewal stream [16,30], which justifies our choice of MMPPs as the candidate models. A further advantage of MMPPs is their stability in superposition: two or more superposed MMPPs are equivalent to some higher-order MMPP [14].

MMPPs have been very popular in modeling telecommunication traffic for more than twenty years. We refer to [17] for a comprehensive survey on stochastic modeling of traffic processes. Self-similarity based models have also been proposed in performance modeling of high-speed networks and we refer to [44] for a bibliographical guide.

6 Conclusions and Future Work

In this paper we present an initial analysis of job arrivals in a production data-intensive Grid, focusing on heavy-tail behavior and self-similarity of the interarrival time processes. Based on the analysis we investigate a set of m-state MMPPs to model the job traffic at different levels. Our conclusions can be summarized as follows:

1. There are no clearly observable daily patterns at the Grid level. Empirically, the number of jobs submitted by different VOs follows an exponential distribution.
2. The interarrival time process at the Grid level is distributed without a heavy tail and is strongly self-similar with $H \approx 0.84$. The best fitted model we find is MMPP2, but it still could not match the autocorrelation in the original trace.
3. The interarrival time processes of different VOs show strong, moderate, and weak self-similarity. The tail becomes longer as the number of jobs in the VO decreases. Experimental results suggest that with the VO size decreasing in an exponential manner, we can model the job arrivals of the corresponding VOs using MMPPs by increasing the number of its states.
4. At the region level, MMPPs are more suitable for processes with longer memories, while IPP can fit the interarrival time distributions very well, which is superior for those processes with very short memories.

5. The interarrival time processes for certain VOs show strong deterministic semi-periodic behavior. This explains the strong autocorrelations (long memory) of the data series. One source for such behavior is from large production VOs (e.g. *lhcb*), where scripts may be used for submitted production jobs. Others could be jobs for testing and monitoring purposes, which is essential for the operation and development of the Grid. Realistic modeling of job arrivals with mixed deterministic and stochastic components requires more future research.

We plan to release our Matlab programs developed for estimating and simulating MMPPs via [15]. Tools for calculating transportation distance are also available [29]. One interesting direction for further research is to correlate job arrivals with job run times to create a complete workload model for performance evaluation in a data-intensive Grid.

Acknowledgment

We are grateful to Gidon Moont, David Colling, and the e-Science group of HEP at Imperial College London who graciously provide us with the LCG Grid trace used in this study. M. Muskulus acknowledges support from NWO grant no. 635.100.006. We also want to thank all the reviewers for their many valuable suggestions that improve the quality of this paper.

References

1. S. Asmussen, O. Nerman and M. Olsson. Fitting phase-type distribution via the EM algorithm. *Scand. J. Statist.* 23:419–441, 1996.
2. A-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207–211, 2005.
3. S. Basu and E. Foufoula-Georgiou. Detection of nonlinearity and chaoticity in time series using the transportation distance function. *Physics Letters A*, 301:413–423, 2002.
4. J. Beran. Statistics for Long Memory Processes. Chapman and Hall, New York, 1994.
5. P. Brémaud. Markov Chains. Gibbs Fields, Monte Carlo Simulation, and Queues. Springer, New York, 2001.
6. S. J. Chapin, W. Cirne, D. G. Feitelson, J. P. Jones, S. T. Leutenegger, U. Schwiegelshohn, W. Smith, and D. Talby. Benchmarks and standards for the evaluation of parallel job schedulers. LNCS 1659:67–90. Springer-Verlag, 1999.
7. W. Cirne and F. Berman. A comprehensive model of the supercomputer workload. In *IEEE 4th Annual Workshop on Workload Characterization*, 2001.
8. A. C. Davison and D. V. Hinkley. Bootstrap Methods and Their Applications. Cambridge University Press, 1997.
9. A. B. Downey and D. G. Feitelson. The elusive goal of workload characterization. *Performance Evaluation Review*, 26(4): 14–29, 1999.
10. C. Dumitrescu, I. Raicu, and I. Foster. DI-GRUBER: A Distributed Approach to Grid Resource Brokering. In proceedings of *Supercomputing '05*, ACM, 2005.
11. Workload Management in EGEE and gLite. http://lxmi.mi.infn.it/egee-jra1-wm/.
12. EMpht program. http://home.imf.au.dk/asmus/.

13. D. G. Feitelson. Workload modeling for performance evaluation. LNCS 2459:114–141, 2002.
14. W. Fischer. and K. Meier-Hellstern. The Markov-modulated Poisson process (MMPP) cookbook. *Performance Evaluation*, 18(2):149–171, 1993.
15. Hui Li. Tools for Workload Modeling in the Grid. http://www.liacs.nl/home/hli/gwm/.
16. H. Heffes and D. M. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. on Sel. Areas in Comm.*, SAC-4(6):856–868, 1986.
17. D. L. Jagerman, B. Melamed, and W. Willinger. Stochastic modeling of traffic processes. *Frontiers in Queueing: Models, Methods and Problems*, CRC Press, 1996.
18. H. Kantz and T. Schreiber. Nonlinear Time Series Analysis. Cambridge University Press, 2003.
19. T. Karagiannis and M. Faloutsos. SELFIS: A Tool For Self-Similarity and Long-Range Dependence Analysis. In *1st Workshop on Fractals and Self-Similarity in Data Mining: Issues and Approaches*, Canada, 2002.
20. H. R. Künsch. The jackknife and bootstrap for general stationary observations. *The Annals of Statistics* 17, 1217–1241, 1989.
21. The Worldwide LHC Computing Grid project. http://lcg.web.cern.ch/LCG/.
22. W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. on Networking*, 2(1):1–15, 1994.
23. H. Li, D. Groep, and L. Wolters. Workload Characteristics of a Multi-cluster Supercomputer. LNCS 3277:176–193, Springer-Verlag, 2004.
24. lp_solve 5.5.0.7. http://lpsolve.sourceforge.net/5.5/.
25. U. Lublin and D. G. Feitelson. The workload on parallel supercomputers: modeling the characteristics of rigid jobs. *J. Para. and Dist. Comput.*, 63(11): 1105–1122, 2003.
26. A. Löbel. Solving large-scale real-world minimum-cost flow problems by a network simplex method. Technical Report SC 96-7, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), February 1996. Software available at http://www.zib.de/Optimization/Software/Mcf/.
27. E. Medernach. Workload analysis of a cluster in a Grid environment. In *proceedings of 11th workshop on Job Scheduling Strategies for Parallel processing*, 2005.
28. R. Moeckel and B. Murray. Measuring the distance between timeseries. *Physica D*, 102:187–194, 1997.
29. M. Muskulus et. al. Estimating differences between probability densities and time series. In preparation. Software available at http://www.math.leidenuniv.nl/˜muskulus/.
30. M. F. Neuts. Structured Stochastic Matrices of M/G/1-type and their Applications. Marcel Dekker, NY, 1989.
31. J. Nabrzyski, J. M. Schopf, and J. Weglarz (Editors). Grid Resource Management: State of the Art and Future Trends. ISBN: 1402075758, Springer, 2003.
32. D. N. Politis. The Impact of Bootstrap Methods on Time Series Analysis *Statistical Science* 18(2):219–230, 2003.
33. Parallel Workload Archive. http://www.cs.huji.ac.il /labs/parallel/workload/.
34. A. Riska. Aggregate Matrix-analytic Techniques and their Applications. PhD thesis, Department of Computer Science, College of William and Mary, 2002.
35. W. J. J. Roberts, Y. Ephraim, and E. Dieguez. On Ryden's EM algorithm for estimating MMPP's. *IEEE Sig. Proc. Let.*, to appear.
36. The LCG Real Time Monitor. http://gridportal.hep.ph .ic.ac.uk/rtm/.
37. T. Ryden. Parameter estimation for Markov modulated Poisson processes. *Communications in Statistics - Stochastic Models*, 10(4):795–829, 1994.
38. T. Ryden. An EM algorithm for estimation in Markov-modulated Poisson processes. *Comp. Stat. and Data Analysis*, 21:431–447, 1996.

39. A. Schrijver. Theory of Linear and Integer Programming. Wiley, Chichester, 1998.
40. S. L. Scott. Bayesian Methods for Hidden Markov Models: Recursive Computing in the 21st Century. *J. Am. Stat. Assoc.*, 97(457):337-351.
41. B. Song, C. Ernemann, and R. Yahyapour. Parallel Computer Workload Modeling with Markov Chains. LNCS 3277:47–62, Springer-Verlag, 2004.
42. M. S. Squillante, D. D. Yao, and L. Zhang. The impact of job arrival patterns on parallel scheduling. *ACM SIGMETRICS Performance Evaluation Review*, 26(4):52–59, 1999.
43. J. I. Takeuchi and K. Yamanishi. A Unified Framework for Detecting Outliers and Change Points from Time Series. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):482–492, 2006.
44. W. Willinger, M. S. Taqqu, and A. Erramilli. A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks. In *Stochastic Networks: Theory and Applications*: 339–366, Oxford University Press, 1996.