Profiling Computation Jobs in Grid Systems

Michael Oikonomakos, Kostas Christodoulopoulos, Emmanouel (Manos) Varvarigos
Computer Engineering and Informatics Department
University of Patras, 26500 Rion, Patras, Greece
manos@ceid.upatras.gr

Abstract

The existence of good probabilistic models for the job arrival process and job characteristics is important for the improved understanding of grid systems and the prediction of their performance. In this study, we present a thorough analysis of the job inter-arrival times, the waiting times at the queues, the execution times, and the data sizes exchanged at the kallisto.hellasgrid.gr cluster, which is part of the EGEE Grid infrastructure. By computing the Hurst parameter of the inter-arrival times we find that the job arrival process exhibits self-similarity/long-range dependence. We also propose simple and intuitive models for the job arrival process and the job execution times. The models proposed were validated and were found to be in very good agreement with our empirical measurements.

1. Introduction

Grid computing is an emerging computing paradigm that exploits networked computers to create a virtual computer architecture for the distributed execution of computational tasks. Grids use job scheduling and resource management to establish a global architecture for sharing computing and storage resources across geographically separated sites. The job arrival times, execution times, and data sizes in Grids are unknown and are better modeled probabilistically. The existence of good probabilistic models for the job arrival process and the job characteristics is important for the improved understanding of grid systems. Such models would facilitate the design and dimensioning of grid systems, the prediction of their performance, the evaluation of new scheduling strategies, and the design of a QoS framework for Grid users.

A great deal of work has appeared in the literature on job characterization and modeling [1] for single parallel supercomputers [2], [3], but the corresponding work in the area of Grid computing is quite limited [4], [5]. Medernach [4] analyzed the workload of a LCG/EGEE cluster, proposing a 2-dimensional Markov chain for modeling user behavior in a Grid environment. The user shifts between login and logout states and submits jobs when in the login state. The results indicate that this model can satisfactorily approximate the submission behavior of a single user.

Taking a different approach, Li et al [5] used the LCG Real Time Monitor [6] to collect data from Resource Brokers (RBs) participating in the EGEE project [7], and propose models for the job arrival process at three different levels: Grids, Virtual Organizations and regions. By comparing a set of m-state Markov modulated Poisson processes (MMPP) with Poisson and hyper exponential processes, they conclude that MMPP models with a sufficient number of states are capable of simulating the job traffic at the three examined levels. However, the proposed models are not intuitive enough, and they do not provide an easily adaptable or extensible way for profiling arrival processes in a general Grid environment.

D. Nurmi et al [8] proposed the enhancement of the workflow scheduler through methods which make accurate predictions of both the execution time of the task on specific hardware, and the time tasks will spend waiting in batch queue. Experiments in 5 HPC showed that incorporating these enhancements improves workflow execution time in settings where batch queues impose significant delays on workflow tasks.

Our work differs from the aforementioned works in the scope-level of the observation and modeling. Particularly, we analyze the inter-arrival times, the workload and the data exchanges (grid-ftp) at our local LCG/EGEE cluster, named *kallisto.hellasgrid.gr,* and propose models for the job arrival process and execution times at a grid node. Our models are simple and depend on a small number of modeling parameters, so as to remain comprehensive and intuitive.

Our results indicate that it is difficult to observe patterns with respect to the weekly and daily cycle of the arrival process. The job arrival process has similar characteristics at different time periods. By computing the Hurst parameter of the inter-arrival times we found

Seventh IEEE International Symposium on Cluster Computing and the Grid(CCGrid'07)
0-7695-2833-3/07 $20.00 © 2007 IEEE

that the job arrival process exhibits self-similarity/long-range dependence. We investigated four models for the job arrival process: a non-homogeneous Poisson process model, a hyper exponential model, a Markov modulated Poisson process model and a Pareto-Exponential model. We found that, despite its simplicity, the Pareto-Exponential model appears to adequately describe the job arrival process and is more accurate than the other models examined. We also observed that a hyper-exponential process with 3 states is sufficient to model the stepwise patterns observed in the distribution of the jobs' Worker Node (WN) execution time. By looking at the job waiting times we found that a high percentage of jobs are served almost immediately, while there are also jobs that remain for a long period in the corresponding queues. In addition to the Computing Element (CE), we also looked at the Storage Element (SE), and observed that the cumulative distribution function of the bytes transferred via the grid-ftp exhibits stepwise characteristics.

The rest of the paper is organized as follows. The *kallisto.hellasgrid.gr* Grid node is presented in Section 2. Section 3 presents the basic metrics for the CE and SE that we used for the statistical analysis presented in Section 4. In Section 5 we propose and validate models for the job arrival process and the job execution times. Conclusions are presented in Section 6.

2. Local grid infrastructure

The *kallisto.hellasgrid.gr* node is part of the EGEE (Enabling Grids for E-sciencE) infrastructure and has been a production site since February 1, 2006. The node's hardware consists of 2 HP racks with 64 servers with Intel Xeon CPUs at 3.4GHz. There are 4 HP servers, each with two 80GB SCSI hard disks running RAID1, 2GB RAM and 2 processors that comprise the core elements of the EGEE site (CE, SE, Monitoring Box and Quattor server). The remaining 60 machines are the Working Nodes (WN), each of which has 80GB SATA hard disk, 1GB RAM and one processor. The racks also include a SAN that controls the 14 SCSI disks (300GB each) of the main storage and an optical switch to connect the servers to the storage. The total capacity of the Storage Element is 4.2TB. All servers are running Scientific Linux v.3 (SL3) and the deployed middleware is the LCG v.2.7 software developed by EGEE. In the near future we are planning to migrate to gLite middleware [9].

Grids are organized in Virtual Organizations (VOs), which are dynamic collections of individuals and institutions sharing resources in a flexible, secure and coordinated manner. Particularly, the *kallisto* node serves the following VOs: Dteam (Development Team), See (South Eastern Europe), Lhcb (Large Hadron Collider Beauty), Esr (Earth Science Research), Atlas (A Toroidal LHC Apparatus), Cms (Compact Muon Solenoid), Biomed (Biomedical), Magic (MAGIC telescope), Compchem (Computational Chemistry) and Hgdemo (Hellas Grid demo). These VOs determine the queues in the MAUI configuration of the CE. MAUI [10] is a local scheduling engine that is used together with the PBS batch system [11]. The configuration of our node, reserves one slot for Dteam so that site functional tests can run without waiting. Previous LCG versions used queues that were based on the estimation of the job execution times, and thus our site configuration and the presented results differ from those in [4] in this respect.

Generally, a user cannot submit a job directly to a cluster; instead, the user has to login to a local User Interface (UI) and submit a job-description written in a specific format (JDL – job description language) [12]. This is forwarded to the corresponding Resource Broker (RB) where the matching process is performed. RB runs the services of the Workload Management System (WMS) that intercommunicates with the Information System (IS) that provides information about the Grid resources available and their status. The RB takes into account the job description, the related VO and the available global resource utilization information and decides where to forward the job. Users give a rough estimation of its maximum execution time, when submitting a job, but this value is usually overestimated and is often considerably larger than the actual job duration. This estimate was used at previous versions of the LCG middleware, for assigning jobs to specific priority queues of a node.

The workload of the LCG/EGEE is solely composed of work-pile tasks termed *bags*. A *bag* is a collection of serial independent jobs that perform *no* communication and are not required to execute simultaneously or to be assigned to the same cluster/site. Jobs communicate, by writing output files to grid Storage Elements or to the user's machine enabling other jobs to read and work on the generated data (forming "pipelines" of jobs). Each job requests a single processor and thus the degree of parallelism is one (trivial parallel tasks). A higher level scheduler fragments each *bag* into individual jobs and places them on (possibly) different sites. Therefore, observing the jobs executed or queued at a site we get a set of independent processes and thus we cannot see if there are additional jobs belonging to the same *bag* running on the same or other remote machines.

3. Measurements

Using the log files of the CE (located under the directory /var/spool/pbs/server_priv/accounting/) we acquired information that was locally maintained in the

kallisto node. The time period of the observation was three months (from February 1, 2006 until April 30, 2006). The total number of jobs submitted during this period was 25737.

We parsed the log files and obtained the desired information in a form suitable for processing using statistical analysis tools. This was achieved by enhancing the Perl scripts [13]. The file we obtained after parsing the log files consist of a table with the following entries:

- A consecutive number (id) for each job.
- The job's exact submission date and time.
- The job's relative submission time.
- The time interval each job waited in its queue.
- The Worker Node execution time (I/O+CPU time).
- The job CPU time.
- The amount of memory each job utilized.
- The estimate of the job CPU time, assigned by the user who submitted the job. (This has some default values – in almost all cases was 259200sec = 3 days).
- The estimate of the amount of memory required by a job, assigned by the user who submitted the job. (This number has some default values – in almost all cases was 512 MB).
- The Job's status (whether the job finished successfully, was canceled, or failed to complete).
- The id of the user who submitted the job.
- The id of each queue.

Apart from the workload analysis we also examined the data transfers between our cluster and the remaining EGEE infrastructure. More specifically, we used the log files of the SE (located under the directory /var/log) to acquire relevant information. The period of the observation was the same with the CE's and the number of grid-ftp connections was 10587.

4. Statistical Analysis

In order to obtain good models for the job submission process and the job characteristics, we performed a thorough statistical analysis of the measurements presented in the previous section. Apart from examining the weekly and daily cycles of the workload we studied the job inter-arrival times, the job (WN) execution times, the waiting times of the jobs and the data transfers involved.

4.1 Submission date and time

Among the first things we looked at is whether the cluster is in use for all days of the week and for 24 hours per day, or its utilization decreases during specific days (e.g., weekends, holidays) or specific daily periods (e.g., at nights). Fig. 1 shows the number of submitted jobs during different days in a week, while Fig. 2 shows the number of jobs during different submission periods within a day. The graphs show that it is difficult to identify any patterns with respect to the date and time of the submission process. Jobs are submitted to the cluster during all days of the week and, contrary to our expectations, the cluster exhibits a gradual increase of its usage at the late hours of the day. These observations can be explained by the fact that users are active across different time zones, and they often schedule their jobs for later times, resulting in a rather even distribution of jobs across all weekly/daily cycles. In interpreting these results we also have to take into account the geographical position of Greece relative to that of the other EGEE users.

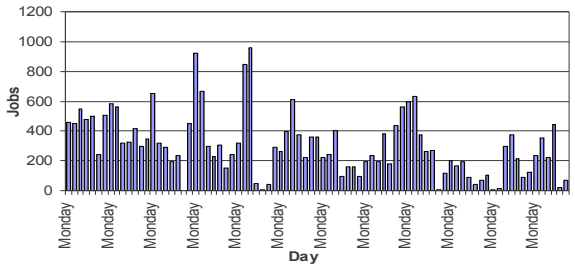

Figure 1: Number of jobs per day

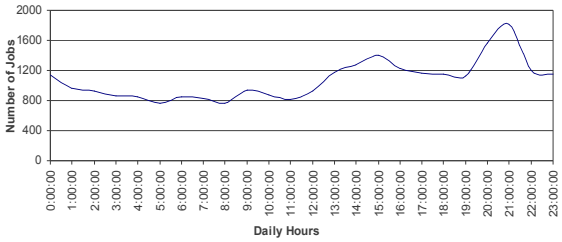

Figure 2: Daily distribution of jobs

4.2 Job execution times

The node's resources are not utilized to the same degree by all VOs. The five most active VOs are listed in Table 1, while the other VOs had a relatively small number of jobs (~ 3% maximum). The Atlas VO contributed approximately 50% of the jobs submitted to our cluster during the duration of our observations.

Table 1: Number and percentage of jobs per VO

VO	Atlas	Biomed	Dteam	Lhcb	Magic
Num. of Jobs	12548	3126	1315	4395	1929
Percentage	49%	12%	5%	17%	7%

Tables 2 and 3 show the mean and standard deviation of the CPU execution time and the Worker Node execution time which is the total running time (CPU + I/O), for all jobs and for each VO separately. Comparing these tables we observe that the standard deviations for the whole set of jobs and for each VO

separately were almost equal. The difference between the averages of Tables 2 and 3 correspond to the duration of the I/O operations and, since it is relatively small, we can deduce that the jobs sent to our cluster were CPU and not I/O intensive.

Table 2: Mean and std of the CPU execution time (sec)

VO	Total	Atlas	Biomed	Dteam	Lhcb	Magic
Mean	15321	16139	24656	13	8511	2736
Standard deviation	29801	30146	25964	25	21236	546

Table 3: Mean and std of the WN execution time (sec)

VO	Total	Atlas	Biomed	Dteam	Lhcb	Magic
Mean	15400	16150	24682	17	8532	2749
Standard deviation	29850	30163	25978	27	21258	567

4.3. Job inter-arrival times

In this section we present results on the job arrival process at our local node. Fig. 3 illustrates the cumulative distribution function (cdf) of the inter-arrival times for the jobs belonging to all the VOs and for the jobs belonging to the VO Atlas, which is the one that contributed the majority of jobs to our node. It is worth noting that site functional tests from the Dteam VO are performed every 3 hours (10800 sec) [7], posing an upper limit on the inter-arrival times.

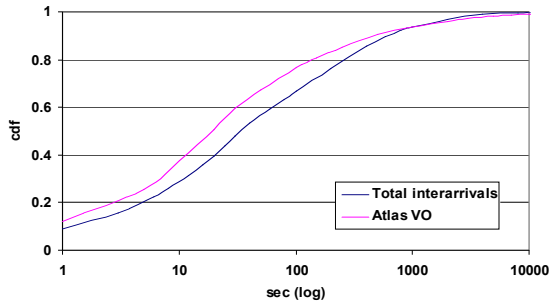

Figure 3: Empirical cdf's of the inter-arrival times for the jobs belonging to all the VOs and for the VO Atlas

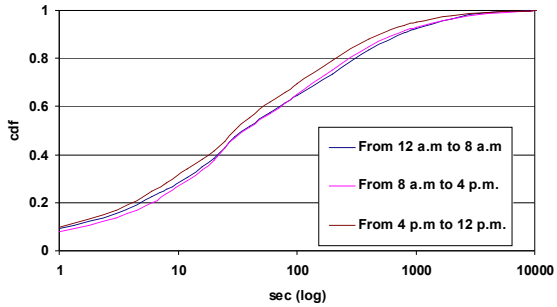

Figure 4: Empirical cdf's of the inter-arrivals per periods of day

To study the way job arrivals are distributed with respect to the time of day, we divided the 24 hours of a day into three 8-hour periods, and present the corresponding graphs in Fig. 4. We observe that the cdfs have the same shape for the different time periods, while jobs that arrive between 4p.m. and 12p.m have a higher frequency when compared to the other two investigated periods (these results are in agreement with the results presented in Fig. 2).

4.4 Self similarity

Self similarity deals with burstiness, and is a measure of the degree to which a process includes periods of increased activity and periods of little or no activity. Self similarity implies correlation across different time scales, in the sense that what happens at the present time is correlated to what happened in the recent and also in the more distant past.

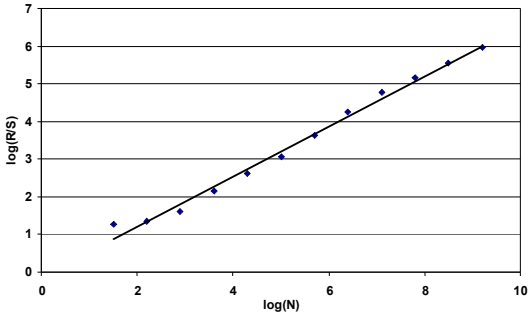

Figure 5: Hurst parameter estimation using the R/S method

One way for checking if a process is self similar is the Rescaled Range Method (or R/S) originally used by Hurst. It produces a log-log plot of the R/S statistic versus the number of points of the aggregated series. This plot should be a straight line with the slope being an estimation of the Hurst exponent. We computed the Hurst parameter (H) of the inter-arrival times using a variety of methods (Aggregate Variance, R/S, Periodogram, Absolute Moments, Variance of Residuals, Abry-Veitch Estimator, Whittle Estimator) [14]). For the above methods we also obtained the correlation coefficient, which gives us a reliability factor for the H estimate (values higher than 0.9 should be sufficient). The higher correlation coefficient (99.31%) was computed using the R/S method, indicating that this was in our case the most reliable method for estimating the Hurst parameter. Using that method, the Hurst parameter of the job arrival process at our local cluster was found to be H=0,684 (Fig. 5). The Poisson process, which is not self-similar as indicated by its memoryless property, has H=0.5. When $0.5 \leq H \leq 1$, as is true in our case, the process has positively correlated consecutive steps. Thus, we conclude that the job arrival process in our local cluster exhibits self-similarity / long-range dependence.

4.5. Job waiting times

We present results regarding the waiting times of the jobs, defined as the time between the acceptance of the job by the Local Resource Management System (LRMS) and the time it starts execution on a WN. When a job arrives at the LRMS, it enters a queue until a CPU becomes available to serve it. Our system in particular uses the MAUI-PBS LRMS whose configuration employs a separate queue for each VO and reserves one time slot for the Dteam.

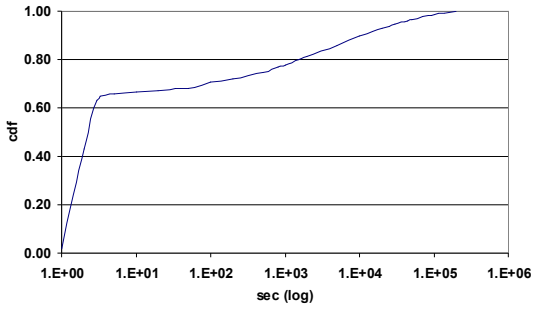

Figure 6: Empirical cdf of the job waiting times

The results of Fig. 6 indicate that a job stays in a queue for less than 2 seconds with large probability (~0.7). There are also, however, a few jobs that stay in their queue for a long time period due to congestion, general or specific problems of our system. The mean and the standard deviation of the waiting time for all the VOs together and separately for each VO are shown in Table 4. We can observe that Dteam experiences the lower average delay, while Biomed the highest. This is because of the local queues priority policies and the fact that Dteam's jobs require the smallest CPU times (Table 2), while Biomed's jobs are CPU-intensive and thus exhibit the highest delays.

Table 4: Mean and std of the waiting time (sec)

VO	Total	Atlas	Biomed	Dteam	Lhcb	Magic
Mean	5503	3412	9731	236	2450	867
Standard deviation	19851	13809	19774	19851	11223	4625

4.6. Job Worker Node execution time

The job WN execution time is the actual execution time of a job including the I/O time. When users submit their jobs they also provide an estimate of the job run time, but this is usually a very loose overestimate of the job run time. In Fig. 7, we give the cdf of the actual job WN times obtained from our experiments. It can be seen that the job WN execution times exhibit stepwise characteristics:

- With small probability (~0.15) a job completes its execution within a few seconds (less than 60 sec). Usually such jobs are site functional tests, ldap queries, etc.
- With medium probability (~0.25) a job completes its execution within several minutes after entering the WN (less than 30 minutes) – small jobs.
- With large probability (~0.6) a job completes its execution several hours after entering the WN. These jobs usually correspond to large experiments.

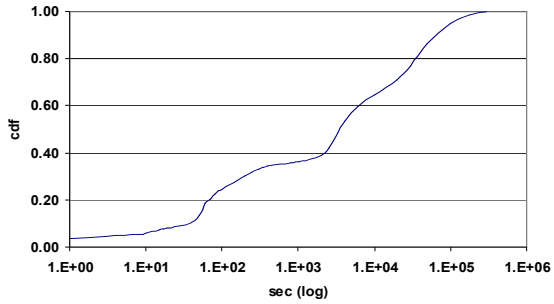

Figure 7: Empirical cdf of the job WN execution times

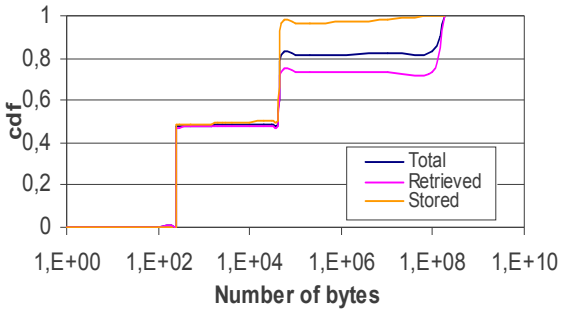

Figure 8: Empirical cdf's of the grid-ftp transfer sizes

4.7. Storage measurements

We have also analyzed the grid-ftp traffic at our local node. Fig. 8 shows the cdfs of the retrieved, stored and total number of bytes exchanged between our Storage Element (SE) and the remaining EGEE infrastructure. During the observation period, the total number of grid-ftp connections to our SE was 10587 (3753 store and 6834 retrieve requests). We observe that the cdf graphs have a step-wise constant form. This is because the majority of the data exchanges are related to the Dteam site functional tests (SFTs). More specifically there are two sets of SFTs: (i) of size 240B that are sent periodically every 1 or 3 hours (~5000 connections), and (ii) of size 41KB that are sent at irregular time intervals (~3450 connections). The Atlas VO exchanged a large number of 103 MB chunks of data (~1800 connections), while the other VOs had rather low activity with respect to data transfers.

5. Modeling

It is possible to use directly log traces of the job arrivals as an input to a static simulation, but it is usually more convenient to define and use analytic models for the job arrival process. Analytic models are more flexible, since they allow the generation of traces using different values of the parameters involved, helping better understand the way these parameters affect system performance.

In this section we are interested in modeling the job arrival process and the execution time of the jobs in our cluster. We decided to model the job arrival process and the execution times for the traffic generated by all the VOs together, and not separately for every VO, in order to look for general properties in the workload that the local cluster has to tackle.

5.1 Modeling the job arrival process

We considered and evaluated four different models for the job arrival process:

(a) **Non-Homogeneous Poisson Process (NHPP) model**

Taking into account the variations of the job arrival rate with respect to the days of a week (Fig. 1) and the hours of day (Fig. 2) we initially investigated if the job arrival process can be modeled as a non-homogeneous Poisson process (NHPP). A NHPP is a Poisson process whose arrival rate λ at time t is a function of time $\lambda(t)$. More specifically, the number of arrivals $N(t)$ in the interval [0,t) follows the distribution:

$$\Pr(N(t)=n)=e^{-m(t)}\frac{(m(t))^n}{n!}, n\geq 0 \ and \ m(t)=\int_0^t \lambda(s)ds$$

Using the results of Fig. 2, we defined a stepwise function for $\lambda(t)$, obtained by averaging over all days in our observation period the number of job arrivals observed during each 1 hour interval of a day.

(b) **Hyper Exponential model**

An m-Phase-Type distribution (PT) represents random variables that are the transition times until absorption of a continuous-time Markov chain with m transient states and one absorbing state. Generally, any inter-arrival process can be approximated by a phase-type distribution if a sufficient number of states is used.

From this general class we chose to consider only the hyper exponential subclass, which is the one most often used in the literature. The probability density function of an m-phase hyper exponential random variable X is given by:

$$f_X(x)=\sum_{i=1}^{m} p_i\, f_{Yi}(y)=p_1 f_{Y1}(y)+p_2 f_{Y2}(y)+...+p_m f_{Ym}(y)$$

where Y_i is an exponentially distributed random variable with rate parameter λ_i, and p_i is the probability that X will take on the form of Y_i (thus, $\sum_{i=1}^{m} p_i = 1$).

More specifically we considered two cases: (i) a 2-phase (H2) and (ii) a 3-phase hyper exponential distribution (H3). To find suitable parameters (3 parameters in case (i) and 5 parameters in case (ii)) we used the EMpht program [15], which employs an Expectation Maximization (EM) algorithm [16], to obtain the following parameters: Case (i) p_1=0.37, λ_1= $1.37\cdot 10^{-3}$ sec^{-1}, λ_2= $4.65\cdot 10^{-2}$ sec^{-1}, and λ_3= $1.46\cdot 10^{-5}$ sec^{-1}, and Case (ii) p_1=0.444, p_2= 0.457, λ_1= $5.38\cdot 10^{-2}$ sec^{-1}, λ_2= $9.07\cdot 10^{-2}$ sec^{-1}, λ_3= $5.12\cdot 1^{0-3}$ sec^{-1}

(c) **Markov Modulated Poisson Process (MMPP) model**

An m-state MMPP is a doubly stochastic Poisson process [17]. Assuming an m-state continuous-time Markov chain (CTMC), arrivals occur according to a Poisson process of rate λ_i when the chain is in state i. An MMPP can be fully described by

$$Q=\begin{bmatrix} -\sigma_1 & \sigma_{12} & ... & \sigma_{1m} \\ \sigma_{21} & -\sigma_2 & ... & \sigma_{2m} \\ ... & ... & ... & ... \\ \sigma_{m1} & \sigma_{m2} & ... & -\sigma_m \end{bmatrix},\ \sigma_i=\sum_{j=1,j\neq i}^{m}\sigma_{ij} \text{ and } \Lambda=[\lambda_1,\lambda_2,...\lambda_m]$$

where Q is the generator of the CTMC, and the entries of Λ correspond to the Poisson arrival rates at each state.

We investigated two MMPP models: (i) a 3-state MMPP (3MMPP) and (ii) a 4-state MMPP (4MMPP). To find suitable parameters (4 parameters in case (i) and 9 in case (ii)) we used the program found in [18], that employs an EM, to obtain the following MMPP parameters that best fit our measurements (sec^{-1}): Case (i): $\sigma_{12}=6\cdot 10^{-3}$, $\lambda_1=98\cdot 10^{-3}$, $\sigma_{21}=0.45\cdot 10^{-3}$, $\lambda_2=4.1\cdot 10^{-3}$, and Case (ii): $\sigma_{12}=3.2\cdot 10^{-3}$, $\sigma_{13}=4.3\cdot 10^{-3}$, $\lambda_1=139\cdot 10^{-3}$, $\sigma_{21}=0.1\cdot 10^{-3}$, $\sigma_{23}=0.2\cdot 10^{-3}$, $\lambda_2=0.9\cdot 10^{-3}$, $\sigma_{32}=0.45\cdot 10^{-3}$, $\sigma_{32}=0.55\cdot 10^{-3}$ and $\lambda_3=11.9\cdot 10^{-3}$.

(d) **Pareto-Exponential model**

We also investigated a third model for the job arrival process, to be referred to as the Pareto-Exponential model. Under this model, the VOs submit jobs that have exponential inter-arrival times (with rate λ jobs per sec) during busy periods, each of which has an exponential duration (with mean $1/\mu$ sec). The times between the beginnings of the VO busy periods are distributed following a truncated Pareto distribution with Pareto shape parameter a, minimum value parameter X_{min} and maximum value parameter X_{max}. The proposed model is depicted in Fig. 9.

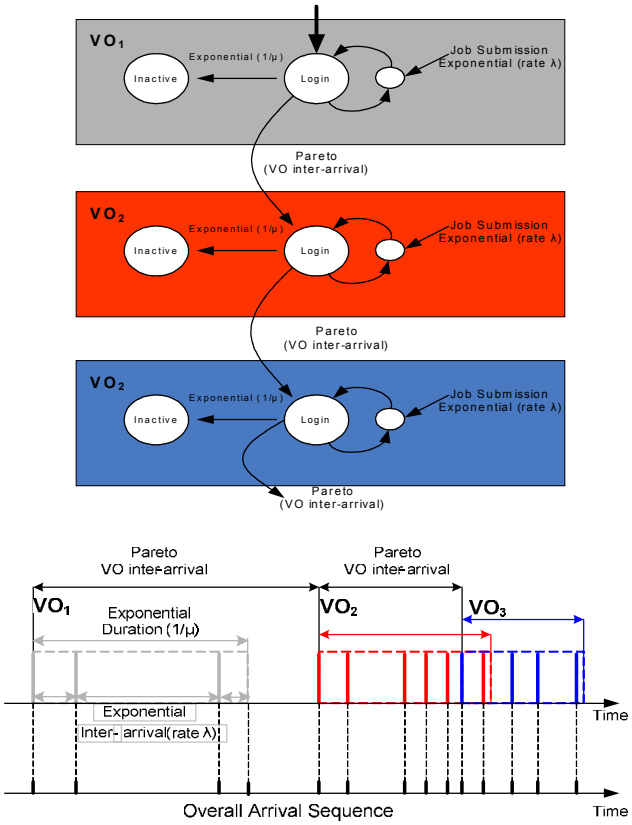

Figure 9: Proposed Pareto-Exponential model for the job arrival process

We have chosen to use a truncated Pareto distribution with X_{max}=10800 sec since we know that the job inter-arrival times are upper-bounded by 3 hours (the times of the Dteam periodic submissions of site functional tests). For the other parameters we conducted a number of trials and concluded in the following values for our case: mean λ=18 arrivals per sec for busy periods, mean duration $1/\mu$=22.5 sec of the busy periods, a=0.48 and X_{min}=32 sec.

5.1.1 Validation of the job arrival process model.

In order to evaluate and compare the proposed models we have simulated them in C++ and generated trace files. Fig. 10 shows the cdf of the inter-arrival times as presented in Section 4.3 and the cdfs we obtained from the traces of the four proposed models. Fig. 11 shows the Probability-Probability (P-P) graphs of the better performing H3, 3MMPP and Pareto-Exponential models versus the actual measurements. Given two CDFs, a P-P plot is constructed by pairing percentiles that correspond to the same value. A "good" fit corresponds to a P-P plot that is nearly linear.

From the above graphs we can conclude that the proposed Pareto-Exponential model generates traces that are very close according to the P-P plot to those

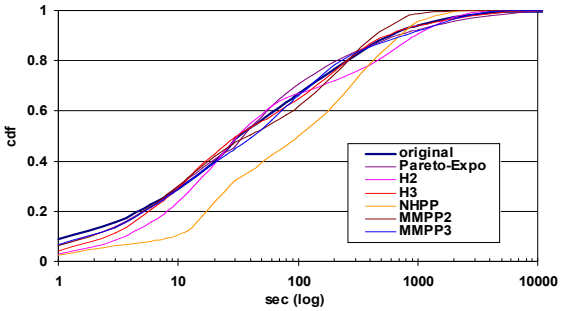

Figure 10: Cdf's of the inter-arrival times of the original observations and the proposed models

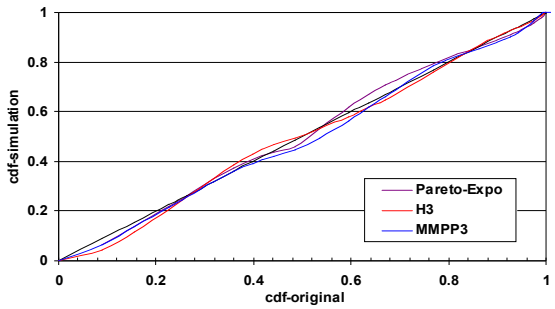

Figure 11: P-P functions of the proposed models

observed in our cluster. H3 and 3MMPP models simulate also satisfactorily the job arrival process. However, the Pareto-Exponential model is simpler, more concise and more intuitive than the other proposed models, since it is based on a smaller number of parameters, and seems to correspond to actual VO behavior.

As expected, by increasing the number of phases in the hyper-exponential process the accuracy of that model also improves. This is, however, only due to fact that by adding complexity (more states) to the hyper-exponential model, we can approximate any process. Similarly, by increasing the states in the MMPP process we obtain better accuracy. However, this is a "mechanical" not an intuitive way to model the inter-arrival process.

We have also computed the Hurst parameter for the four models. Only the Pareto-exponential and the MMPP models experience long-range dependence (H=0.58 for the Pareto-Exponential, H=0.62 for 2MMPP and H=0.64 for 3MMPP with confidence levels higher than 99%), while the models (a) and (b) have a Hurst parameter of 0.5. Given that the MMPP model requires a large number of parameters, the Pareto-exponential model seems to be more appropriate for modeling the job arrival process at a grid node, since it also fits very well the real traffic in our observations and exhibits long-range dependence as indicated by the calculated Hurst parameter.

5.2 Modeling the job WN execution times

The Worker Node execution times, presented in section 4.6 (Fig. 7), exhibit peaks at certain values. Execution times differ in their nature from the inter-arrival times since they do not depend on the human factor, and thus it is difficult to find a physical explanation for their behavior. Therefore, our criteria for modeling WN execution times are more relaxed. We investigated how a hyper exponential process can fit the observed behavior. More specifically, we considered two cases: (i) a 3-phase (H3) and (ii) a 4-phase (H4) hyper exponential distribution. We chose to use these values for the number of phases driven by the observation that Fig. 7 is of a stepwise form with 3 noticeable steps. We used again the EMpht utility to obtain the corresponding parameters: Case (i) p_1=0.3290, p_2=0.2805, λ_1= $1.0731 \cdot 10^{-2}$ sec^{-1}, λ_2= $2.65 \cdot 10^{-4}$ sec^{-1}, and λ_3= $2.1 \cdot 10^{-5}$ sec^{-1}, and Case (ii) p_1=0.3270, p_2=0.2805, p_3=0.14, λ_1= $1.0531 \cdot 10^{-2}$ sec^{-1}, λ_2= $2.65 \cdot 10^{-4}$ sec^{-1}, λ_3= $2.4 \cdot 10^{-5}$ sec^{-1}, and λ_3= $1.8 \cdot 10^{-5}$ sec^{-1}

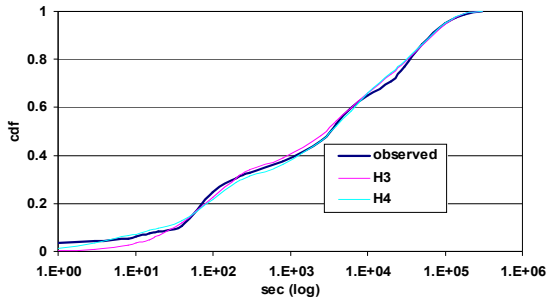

Figure 12: Empirical cdf and cdf's of the proposed models for the WN execution times

5.2.1 Validation of job execution time model.

Similar to section 5.1.2, we simulated the proposed models in C++, produced traces and compared them with those obtained in actual measurements. Fig. 12 shows the empirical cdf of the job WN execution time as presented in section 4.6 and the cdfs we obtained from the traces of the two hyper exponential processes. Since the modeling accuracies obtained by the 3- and 4-phase processes are almost similar, we can conclude that a 3 phase hyper exponential process is sufficient for modeling the CPU execution times.

6. Conclusions

A comprehensive and thorough traffic analysis of a local Grid node was presented. Our results show that there are no noticeable daily or weekly patterns in the job arrival sequence. The job arrival process exhibits long-range dependence as indicated by the Hurst parameter calculated. We proposed several models for the job arrival process one of which is simple and matches well the actual measurements. The model incorporates exponential job inter-arrival times during busy periods of exponential duration (corresponding to a single VO's job submissions). The times between VO busy periods are distributed according to a truncated Pareto distribution. Finally, a 3-state hyper-exponential process was proposed and found to be sufficient for modeling the stepwise patterns of the job execution times.

7. Acknowledgments

The *kallisto.hellasgrid.gr* grid infrastructure is partially funded by the EGEE project [7]. This work has been supported by European Commission through IP Phosphorus and NOE e-Photon/One+ IST projects.

8. References

[1] D. Feitelson, "*Workload modeling for computer systems performance evaluation*", www.cs.huji.ac.il/~feit/wlmod
[2] W. Cirne and F. Berman, "*A comprehensive model of the supercomputer workload*", Proc. 4th IEEE annual workshop on workload characterization, 2001.
[3] B. Song, C. Ernemann and R. Yahyapour, "*Parallel Computer Workload Modeling with Markov Chains",* Proc. 10th JSSPP workshop, 2004.
[4] E. Medernach, "*Workload analysis of a cluster in a Grid environment*", Proc. 11th JSSPP workshop, 2005.
[5] H. Li, M. Muskulus and L. Wolters, "*Modeling Job Arrivals in a Data-Intensive Grid*", Proc. 12th JSSPP, 2006.
[6] Real Time Monitor: http://gridportal.hep.ph.ic.ac.uk/rtm
[7] The EGEE project homepage: http://public.eu-egee.org/
[8] D. Nurmi, A Mandal, J Brevik, C Koelbel, R Wolski and K Kennedy "*Grid scheduling and protocols-Evaluation of a workflow scheduler using integrated performance modelling and batch queue wait time prediction*", Proc. ACM/IEEE conference on Supercomputing 2006
[9]gLite: http://glite.web.cern.ch/glite/
[10] Maui Scheduler: http://supercluster.org/maui
[11] Open PBS: http://www.openpbs.org/
[12] *Job description language: How To*. Publicly available at http://www.infn.it/workload-grid/docs/DataGrid-01-TEN-0102-0_2-Document.pdf.
[13] http://www.cs.huji.ac.il/labs/parallel/workload/swf.html
[14] T. Karagiannis, M. Faloutsos and M. Molle, "*A User-Friendly Self-Similarity Analysis Tool"*, ACM SIGCOMM Computer Communication Review, 2003.
[15] The EMpht program: publicly available at http://home.imf.au.dk/asmus/pspapers.html.
[16] S. Asmussen, O. Nerman and M. Olsson, "*Fitting phase-type distribution via the EM algorithm*", Scnd. J. Statist. 23:419-441, 1996.
[17] W. Fischer. and K. Meier-Hellstern. "*The Markov-modulated Poisson process (MMPP) cookbook*". Performance Evaluation, 18(2):149–171, 1993.
[18] http://www.liacs.nl/~hli/gwm/index.htm