Journal of Engineering and Applied Sciences 12 (24): 7548-7554, 2017
ISSN: 1816-949X

Methods of Modeling Incoming Jobs Stream used on the Computing Cluster UniLu-Gaia

Sergey Vladimirovich Gaevoy, Wesam Mohammed Abdo Ahmed,
Sergey Alekseevich Fomenkov and Sergey Grigorievich Kolesnikov
Volgograd State Technical University, Lenin Av. 28, 400005 Volgograd, Moscow,
Russian Federation

Abstract: At present, the need to analyze the maintenance of tasks by computing clusters is urgent. In the study the question of transition from a log of operation of a computing cluster (the list of all comers of jobs) to its approximation is considered. Stochastic parameters are approximated: runtime of jobs intervals between their arrivals. The result is checked by simulation modeling.

Key words: Method of the greatest (maximum) credibility, method of the moments, distribution function, parallel loadings, not scalable tasks, time of performance of tasks, imitating modeling, stochastic approximation

INTRODUCTION

The problem of optimal execution of parallel and high-performance computing is now quite actual (Sinisterra *et al*., 2012). In particular, there is the issue of optimal load balancing and selection of optimal performance (Gaevoy *et al*., 2014a) of Computing Clusters (CC). One of the possible ways to solve this problem is to simulate the work of the CC including the simulation of it (GridMe: Grid Modeling. https://code.google.com/p/gridme). The latter requires the construction of a mathematical model of CC and the incoming workload (Jann *et al*., 1997). In this study, we construct a stochastic model of the workload.

The real CC is built from computers that serve incoming tasks (Den optimalen Rechnerverbund gibt es nicht einmal auf dem Papier (in German). In: Computerwoche. http://www.computerwoche.de/a/den-optimalen-rechnerverbund-gibt-es-nicht-einmal-auf-dem-papier,1087149). In this study, an already elaborated model (Gaevoy *et al*., 2014b; Gaevoy and Al-Hadsha, 2013) of such a CC is used as a serving unit with not prioritized, unlimited queue. That ensures that all incoming jobs are served.

Clusters are usually constructed from computers of equal performance (Gaevoy *et al*., 2004, 2014ab; Gaevoy and Al-Hadsha, 2013). Tasks for execution are coming into the cluster system. Each task can be executed in parallel on several machines (service channels).

We introduce the following definitions (Avetisyan *et al*., 2004). The number of computers on which a task is executed is called its width. The length of the task will be called the time of its execution. The square of the job is the product of the length and the width. Obviously, the square is the complexity of the task. It also represents the total used machine time. Note that different researchers of existing publications on the topic of this study use different terminology. We will assume that the width of the job is determined at the time of its creation which is a fairly frequent assumption (Jann *et al*., 1997; Anonymous, 1996)

By Lublin and Feitelson (2003) logs of real computer system's workloads are provided. A parallel workload contains arrival times jobs, their widths and lengths. This is the loading of a computer system. The goal is to approximate stochastic values: the intervals between job arrivals, lengths (or squares) and job widths for the subsequent simulation.

It is necessary to build a model that makes it possible to switch from the logs to random variable distributions, to find methods for generating stochastic task parameters. This transition will reduce the amount of information needed to store the workload, show patterns in it, predict possible load options, obtain material for service modeling, etc.

The quality of the result is estimated by simulating generated workloads. The model from (Gaevoy *et al*., 2014a, b; Gaevoy and Al-Hadsha, 2013) is used as the model and the deterministic load from (Lublin and Feitelson, 2003) is modeled as the standard.

Corresponding Author: Sergey Vladimirovich Gaevoy, Volgograd State Technical University, Lenin Av. 28, 400005 Volgograd, Moscow, Russian Federation

MATERIALS AND METHODS

Workload approximation; Used approximation methods: Approximation methods are Moment Method (MM) and Maximum Likelihood Method (MLM). MM assumes the calculation of distribution parameters by the moments. Thus, the number of moment estimations should be equal to the number of parameters of a distribution.

We denote moments as: E(X) the expectation of the variable X, VAR(X) its variance, stDev(X) standard deviation, cov(X) = stDev(X)/E(X) the coefficient of variation. If we are dealing with a moment estimate, we will make a horizontal line.

The necessary estimates can be obtained from the formulae (Reducing the approximation time of cluster workload by using method of moments on hyperexponential distribution (in Russian):

$$\overline{E(X)} = \frac{1}{N}\sum_{i=1}^{N} X_i \quad (1)$$

$$\overline{VAR(X)} = \frac{1}{N-1}\sum_{i=1}^{N}\left(X_i \overline{E(X)}\right)^2 = \frac{N}{N-1}\left(\overline{E(X^2)} - \overline{E(X^2)}\right) \quad (2)$$

Where:
N = The number of observations
X_i = Specific observation No. i

Also, we will denote pdf (x) and cdf (x) the probability density function and the cumulative distribution function. Use MM only for distributions that have no more than four parameters as in practice the obtaining of the moments above fourth order is difficult because of accidents.

Maximum Likelihood Method (MLM) does not have this drawback but requires large computational resources. The likelihood function has the form:

$$L = \prod_{j=1}^{N} pdf\left(X_j\right) \quad (3)$$

Where:
N = The number of observations
X_j = Specific observation No. j

In accordance with the method, it is necessary to produce the maximization of this function. Analytical solution in the general case can be difficult. The function can take values very close to zero and this can lead to serious problems due cto rounding v in the computer, so, the maximization of the function should be replaced by the maximization of its logarithm InL (or minimization of InL). As an optimization method we will use the method of Hooke-Jeeves or (where possible) an analitic solution.

Used distributions: After verifying the conclusions ,) we are using now (Reducing the approximation time of cluster workload by using Method of moments on hyperexponential distribution (in Russian) (Anonymous, 1998).

M; exponential distribution:

$$pdf(x) = \lambda e^{-\lambda x} \quad (4)$$

$$cdf(x) = 1 - e^{-\lambda x} \quad (5)$$

$$E(X) = stDev(X) = \frac{1}{\lambda} \quad (6)$$

For this distribution, the evaluation of MLM and MM coincide and give. The solution for MLM is possible analytically:

$$\lambda = \frac{1}{\overline{E(X)}} \quad (7)$$

Γ; Gamma distribution:

$$pdf(x) = \lambda \frac{(\lambda x)^{v-1}}{\Gamma(v)} e^{-\lambda x} \quad (8)$$

$$cdf(x) = \frac{\gamma(v, \lambda x)}{\Gamma(v)} = P(v, \lambda x) \quad (9)$$

$$E(X) = \frac{v}{\lambda} \quad (10)$$

$$VAR(X) = \frac{v}{\lambda^2} \quad (11)$$

Where:
$\Gamma(x)$ = The gamma function Euler's
$\gamma(x, y)$ = The lower incomplete gamma function
$P(x, y)$ = The lower gamma function

Using MM (Γμ) gives:

$$\lambda = \frac{\overline{E(X)}}{\overline{VAR(X)}} \quad (12)$$

$$v = \lambda \overline{E(X)} \quad (13)$$

Using MLM (Γ_λ) is more difficult. The partial derivatives of the logarithmic likelihood function are equal to zero, thus, we have:

$$\lambda = \frac{v}{\overline{E(X)}} \quad (14)$$

$$\psi(v) = \overline{E(\ln X)} - \ln\left(\overline{E(X)}\right) + \ln? \quad (15)$$

Where:

$\Psi(v)$ = The digamma-function

$\overline{E(\ln X)}$ = The average value of the logarithm of the random variable

It should be noted that both estimates give the same mathematical expectation which coincides with the estimate (E (X) = E (X)) but the remaining moments will differ in general case.

All distributions before will be assumed simple in contrast to the hyper-distributions.

H(n); Hyperexponential distribution:

$$\text{pdf}(x) = \sum_{i=1}^{n} \alpha_i \lambda_i e^{-\lambda_i x} \quad (16)$$

$$\text{cdf}(x) = 1 - \sum_{i=1}^{n} a_i e^{-?_i x} \quad (17)$$

$$1 \geq \alpha_i \geq 0 \quad (18)$$

$$\sum_{i=1}^{n} a_i = 1 \quad (19)$$

$$\text{cov}(X) \geq 1 \quad (20)$$

Where n number of branches, distribution branches (given before the approximation as part of the distribution type).

From the condition $\sum_{i=1}^{n} a_i = 1$ it follows that one α_i is determined by the other, therefore, the number of parameters of this distribution is 2n-1. If a distribution with two and three branches is used, then you need to define up to five parameters. Because of the versatility of MLM, denote its approximation as the distribution itself H(n).

In (Logs of real parallel workloads from production systems, The Rachel and Selim Benin School of Computer Science and Engineering. http://www.cs.huji.ac.il/labs/parallel/workload/logs.html) it is shown that, it is possible to use MM on hypererlang distribution but one had to reduce the number of parameters. For hyperexponential distribution with two branches MM still applies, since, the number of parameters is three. After the preparation and the solution of a system of three equations (Downey, 1997) we get:

$$v^2 = \frac{\overline{VAR(X)}}{\overline{E^2(X)}} \quad (21)$$

$$\beta = \sqrt{\frac{v^2 - 1}{2}} \quad (22)$$

$$\bar{\gamma} = \frac{\overline{E(X^3)}}{6E^3(X)\beta^3} - \frac{1+3\beta^2}{\beta^3} \quad (23)$$

$$\lambda_1 = \left(E(X)\left(1 - \sqrt{\frac{1-a}{a}}\beta\right)\right)^{-1} \quad (24)$$

$$\lambda_2 = \left(E(X)\left(1 + \sqrt{\frac{a}{1-a}}\beta\right)\right)^{-1} \quad (25)$$

$$\alpha_1 = \max\left(\frac{1}{2}\left(1 + \frac{\bar{\gamma}}{\sqrt{\bar{\gamma}^2 + 4}}\right); \frac{\beta^2}{1+\beta^2}\right) \quad (26)$$

$$\alpha_2 = 1 - \alpha_1 \quad (27)$$

For the moments:

$$E(X) = \overline{E(X)} \quad (28)$$

$$VAR(X) = \overline{VAR(X)} \quad (29)$$

$$\begin{aligned} E(X^3) &= \max\left(\overline{E(X^3)}; 6E^3(X)\left(1+\beta^2\right)^2\right) \text{if } \beta > 0 \\ E(X^3) &= 6E^3(X) \text{if } \beta = 0 \end{aligned} \quad (30)$$

The third moment may differ from the estimate but the expectation and the variance are always equal to their estimates. We denote this simplified hyperexponential distribution approximation using MM as Hμ.

HΓ(n); Hypergamma distribution:

$$\text{pdf}(x) = \sum_{i=1}^{n} a_i \lambda_i \frac{(\lambda_i x)^{v_i - 1}}{G(v_i)} e^{-\lambda_i X} \quad (31)$$

$$\text{cdf}(x) = \sum_{i=1}^{n} a_i P(v_i, \lambda_i x) \quad (32)$$

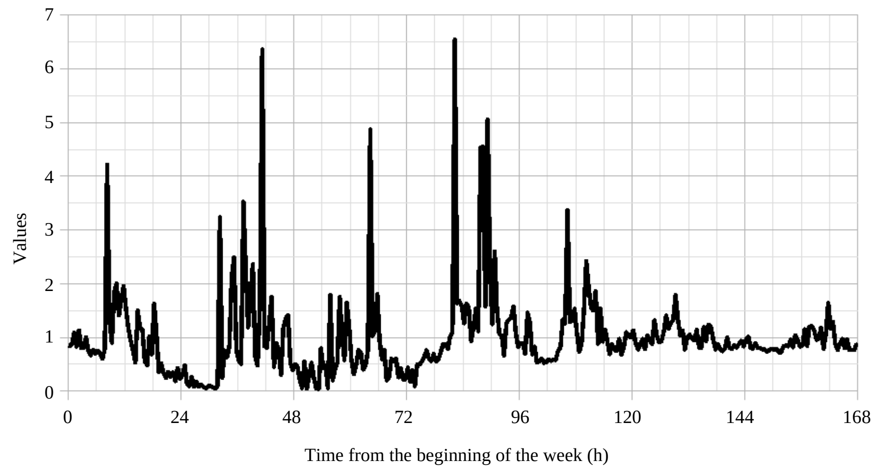

Fig. 1: Changes of the hazard of the job income during the week

$$\sum_{i=1}^{n} a_i = 1 \quad (33)$$

$$1 \geq a_i \geq 0 \quad (34)$$

$$\text{cov}(X) \in (0;\infty) \quad (35)$$

Where n number distribution branches (given before the approximation as part of the distribution type). The number of parameters of this distribution is given by the formula 3n-1. In this research, we will use the distribution with two branches n = 2 which has 3n-1 = 3×2-1 = 5 parameters.

Therefore, even for a distribution with two branches, MM is not applicable. To consider a hyper-distribution with one branch does not make sense as it will typically be exponential or gamma-distribution. We will use MLM. Because of the versatility of MLM, denote its approximation as the distribution itself HΓ(n).

Workload models: In this study, we present a significant modification and refinement of the five basic models of loads (7, 9). For model description our proposed modification of Kendall's notation of will be used.

Each workload model consists of two parts: a model of arrival times of jobs and a service model. Let's start with the simplest option: A, B, B^ where A approximation by some distribution of time interval between arrivals of jobs, B approximation of the square (when you specify "^" -length).

The width is a discrete random variable and represented by a finite number of values. Therefore, it can be approximated simply as an array of probabilities. Different widths can have very different distribution characteristics of the arrival/length/square. So, it makes sense to allocate separate parameters for intervals of width (Logs of real parallel workloads from production systems, the Rachel and Selim Benin School of Computer Science and Engineering. http://www.cs.huji.ac.il/labs/parallel/workload/logs.htm). In the simplest case, we allocate one prameter's set for each width. We will denote this by an icon "$" before the designation of the distribution: $B. Due to the fact that there is only one width in each set the length will be proportional to the squre and no separate approximation for the lenght is needed.

The second division is to select separate groups of each width that are powers of two. This makes sense, since, according to Fomenkov *et al*. (2014) in the logs jobs whose width is a power of two are dominating, even when there are no technical prerequisites. Works such as (Downey, 1997) consider thera are also other dominant (but weaker than power of two) widths, for example, multiples of ten. In other research (HPC @ Uni.lu. https://hpc.uni.lu/systems/gaia/) the researchers are trying to get away from this trend.

The widths between the powers of two are also separate groups: one group for each interval. So, we get the groups: 1-64, etc. let's denote this separation of groups with "&": &B and &B^. A similar division can be done for the input of time tasks: we can select multiple input streams. Using the same principles of partitioning and labeling we get $A, &A.

By Fomenkov *et al*. (2014) is proposed to analyze the input stream as non-stationary. In this study, we will consider the change of the hazard of incoming jobs during the week. In Fig. 1 there are apparent fluctuations in the hazard during the 7 days. In the beginning of the week we took midnight from Sunday to Monday. We assume that the arrival hazard remains constant for half of an hour (as by Fomenkov *et al*. (2014). We are using the normalized hazard:

$$\overline{\lambda(t)} = \lambda(t)/\bar{\lambda} \quad (36)$$

Where:

$\overline{\lambda(t)}$ = The normalized hazard of the arrival
$\lambda(t)$ = The hazard
$\bar{\lambda}$ = The average hazard of the arrival

To generate the interval between arrivals of the nonstationary stream we try to scale the timeline. We assume the time interval between the real points t_0 and t_1 to be the value of the integral:

$$L(t, t_0) = \int_{t_0}^{t} \overline{\lambda(t)}dt \quad (37)$$

Let's call it "normalized time" between the arrivals of tasks. This stream will be stationary and can be approximated by a usual method and then we can return to the original timeline. We denote this model the sign "~" before the designation of the input stream for example, ~A. The designations ~$A and $~A are not the same. In the first case, we introduce a single hazard for all input streams and in the second each stream gets its own hazard.

Thus, we obtain the following : A, ~A, $A, &A, ~$A, ~&A, $~A, &~A input streams and service options: B, $B &B, B^, &B^. The combination of these two models gives the parallel workload model. We will denote it by a combination with a slash, e.g., ~A/&B. So, we have 40 combinations. Seven possible approximation (M, Γ_μ, Γ_λ, H_μ, H(2), H(3), HΓ(2)) give us 1960 models. We can't show all these model simulation but we'll show te most important cases.

RESULTS AND DISCUSSION

Approximation of the log of cluster UniLu-Gaia: The 13 provides the logs of the real computer systems, giving the arrival times of tasks, the length and the width. We will use the log UniLu-Gaia-2014-2.swf which belongs to the cluster UniLu Gaia (The University of Luxemburg Gaia Cluster log (18)) with 2004 service channels (Varrette, 2017).

An example of the approximation of the Empirical Distribution Function (EDF) of the time between arrivals of jobs is depicted in Fig. 2-4. The hyperdistributions when using MLM do more accurately describe the random variable but MM does reduce the quality.

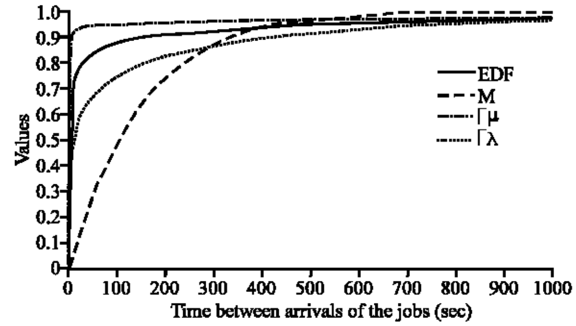

Fig. 2: Approximation of the time between arrivals of the jobs by the simple distributions

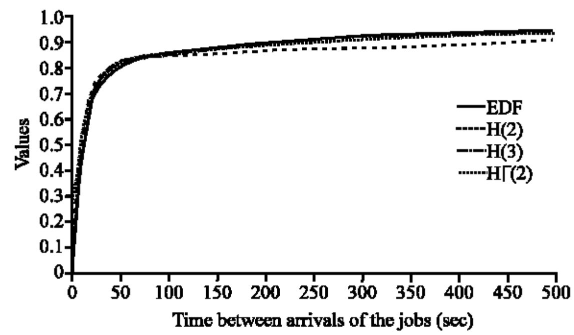

Fig. 3: Approximation of the time between arrivals of the jobs by the hyperdistributions with MLM

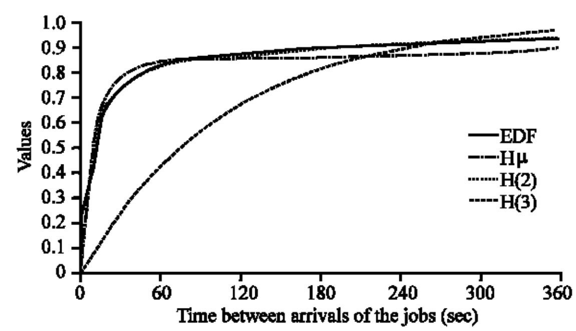

Fig. 4: Approximation of the time between arrivals of jobs by hyperexponential distribution

Table 1: Execution speed of various types of approximation

	Execution time			
Analysis	H_μ	H(2)	H(3)	HΓ(2)
Yes	5 sec	2 min 16 sec	10 min 31 sec	17 min 12 sec
No	2 min 34 sec	4 min 45 sec	12 min 59 sec	20 min 29 sec

We should compare the time and quality of the results. Each approximation in our program (6, 8) usually accompanied by an analysis. So, it takes more time to calculate (Table 1). It is obvious that the analysis takes about 2 min 30 sec.

Table 2: Best approximations

Variables	&~H(2)/Γλ^	~&HΓ(2)/H(2)^	$~H(2)/&HΓ(2)^	&~HΓ(2)/H(2)^	~&H(3)/&Γλ^	Original
The mean execution time (sec)	14257	14274	14313	14274	14325	14329
The average number of running jobs	96.114	96.71	96.556	95.641	96.813	93.067
The average number of busy channels	959.93	963.36	926.74	958.65	917.9	872.26
The average waiting time (sec)	75.776	68.331	66.145	65.643	84.703	72.41
The average waiting time (sec)	2193.1	2250.5	2293.6	2264	2224.2	2259.7
The percentage of the queued jobs	0.031249	0.02806	0.026674	0.025697	0.034756	0.032044
The average length of the queue	0.51156	0.46544	0.44823	0.43867	0.57612	0.4703
The average width of the queue	14.185	14.686	14.941	15.439	14.8	15.31
The average sojourn time in the system (sec)	14332	14343	14379	14340	14410	14402
The average length of the system	96.625	97.176	97.004	96.08	97.39	93.537
The average width of the system	974.12	978.05	941.68	974.09	932.7	887.57
Deviation	0.070089	0.072419	0.063817	0.082707	0.083977	0

Table 3: Best approximations with H_μ

Variables	&~Hμ/$Hμ	~&Hμ/Hμ^	&~Hμ/Hμ^	&~Hμ/&Hμ^	&~Hμ/&Hμ	Original
The mean execution time (sec)	14352	14196	14196	14375	14324	14329
The average number of running jobs	96.511	96.2	96.105	97.161	96.816	93.067
The average number of busy channels	905.98	966.42	959.97	920.9	910.3	872.26
The average waiting time (sec)	38.28	35.247	33.043	28.605	25.24	72.41
The average waiting time' (sec)	1873.8	1391.7	1323.5	2327.7	1789.4	2259.7
The percentage of the queued jobs	0.016405	0.019913	0.020755	0.007809	0.009683	0.032044
The average length of the queue	0.26044	0.2413	0.2262	0.19438	0.17137	0.4703
The average width of the queue	9.395	12.048	11.916	8.023	7.073	15.31
The average sojourn time in the system (sec)	14390	14231	14229	14404	14349	14402
The average length of the system	96.771	96.441	96.331	97.355	96.987	93.537
The average width of the system	915.37	978.46	971.88	928.92	917.37	887.57
Deviation	0.27683	0.27981	0.29096	0.37155	0.38778	0

Modeling approximations: To assess the quality of the approximation carried out we use a stochastic simulation of using the proposed models. For the simulation we have improved a tool developed in (6-8) allowing to reduce the simulation time and thus, to examine a much larger number of models. In our case 1961 including the standard.

To estimate the simulation results we take the result of a deterministic simulation of the original workload. To choose best option the criterion of deviation is used:

$$\text{Dev} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\overline{P}_i - P_i}{P_i}\right)^2} \quad (38)$$

Where:

m = The number of parameters
P_i = The reference value of the parameter
$\overline{P}_i$ = The value obtained from the stochastic model

Note that we have received two average waiting times in the queue due to the fact that not every job does pass through the queue. Thus the average waiting time can be estimated two ways. We can do this for all jobs, considering the zero waiting time for not queued job (without an apostrophe). We can determine this parameter only for queued (with an apostrophe). This value cannot be less than the previous one but in practice is often much greater. In other words jobs rarely go into the queue, but those which do wait for a very long time.

The error of simulation was set to 5%. By central limit theorem (15) this requires more than 40 simulation experiments for each case. This is already more than 80000 tests. The models with the smallest value of the deviation parameter are presented in Table 2. The differences between the good models are within the error of the simulation, so, it makes no sense to talk about which one is better. About 20 other models give us very results very close to that.

If we compare the results of MM and MLM for hyperexponential distribution (Table 3), we can see that the quality of MLMs is high enough compared to the time consuming (Table 1). In Table 3, we did not consider mixed versions, i.e., one method was used for the intervals between incomes and the another for length/square.

CONCLUSION

Thus, an improved method of approximating a stream of jobs in computer systems I proposed. The list of the distribution to use was made up. Due to an optimization of the calculations the number of models was increased. This time we got about 25 good models, although, in the previous researches this number was only a few models.

RECOMMENDATIONS

Obtained models allow us to recreate a random workload of a computing system and to use it instead of a log in the further studies in order to determine the quality of service, optimize a computing system's parameters, find ways to balance the workload.

Also we have received a little rough but very fast approximation which gives the satisfactory results. Its deviation is only four times greater than the best resulta but we have significantly reduced the calculation time.

ACKNOWLEDGEMENT

Research is performed with financial support of the Russian Federal Property Fund (projects 16-07-00534-a and 15-07-09142-a).

REFERENCES

Anonymous, 1996. Logs of real parallel workloads from production systems. Selim and Rachel Benin School of Computer Science and Engineering, Jerusalem, Israel. http://www.cs.huji.ac.il/labs/parallel/workload/logs.html.

Anonymous, 1998. [The optimal computer network does not even exist on paper]. IDG Business Media GmbH, Munich, Germany. (In German) https://www.computerwoche.de/a/den-optimalen-rechnerverbund-gibt-es-nicht-einmal-auf-dem-papier,1087149.

Avetisyan, A.I., S.S. Gaisaryan, D.A. Grushin, N.N. Kuzyurin and A.V. Shokurov, 2004. [Heuristics of job distribution for grid resource brocker (in Russian)]. Proc. Inst. Syst. Program. Russ. Acad. Sci., 5: 269-280.

Downey, A.B., 1997. A parallel workload model and its implications for processor allocation. Proceedings of the International Symposium of High Performance Distributed Computing, August 5-8, 1997, Portland, OR, USA., pp: 112-123.

Fomenkov, S.A., V.A. Kamaev and Y.A. Orlova, 2014. [Mathematical Modeling of System Objects]. Volgograd State University, Volgograd, Russia, Pages: 335 (In Russian).

Gaevoy, S.V. and F.A.H. Al-Hadsha, 2013. Simulation of computing cluster discovering LANL CM5. Electron. Periodical Sci. Mag., 3: 304-313.

Gaevoy, S.V., F.A.H. Al-Hadsha and S.A. Fomenkov, 2004. Approximation of job execution time discovering computing cluster LPC EGEE. Actual Prob. Manage. Comput. Hardware Inf. Eng. Syst., 12: 135-141.

Gaevoy, S.V., F.A.H. Al-Hadsha and S.A. Fomenkov, 2014. [Deterministic simulation model of clusters of a grid-system for comparison of heuristics for task distribution (in Russian)]. Caspian J. Control High Technol., 2: 148-157.

Gaevoy, S.V., F.A.H. Al-Hadsha and V.S. Luk'yanov, 2014. [Deterministic simulation model of clusters of a Grid-system executing jobs (in Russian)]. Her. Comput. Inf. Technol., 6: 39-43.

Jann, J., P. Pattnaik, H. Franke, F. Wang and J. Skovira *et al.*, 1997. Modeling of Workload in MPPs. In: Job Scheduling Strategies for Parallel Processing, Feitelson, D.G. and L. Rudolph (Eds.). Springer, Berlin, Germany, ISBN:978-3-540-63574-1, pp: 95.

Lublin, U. and D.G. Feitelson, 2003. The workload on parallel supercomputers: Modeling the characteristics of rigid jobs. J. Parallel Distrib. Comput., 63: 1105-1122.

Sinisterra, M.M., T.M.D. Henao and E.G.R. Lopez, 2012. [Cluster load balancing and high availability for web services and mail (In Spanish)]. Tech. Inf., 79: 93-102.

Varrette, S., 2017. The Gaia Cluster. University of Luxembourg, Luxembourg, Europe.