

Measure of Adequacy for the Supercomputer Job Management System Model

Anton Baranov, Dmitriy Lyakhovets, Gennady Savin, Boris Shabanov, Pavel Telegin
Joint Supercomputer Center of the Russian Academy of Sciences
Leninskiy pr., 32a, Moscow, Russia
Email: {abaranov, ptelegin, shabanov}@jscc.ru, anetto@inbox.ru

Abstract—In this paper we investigate the problem of modelling modern supercomputer job management systems (JMS). When modelling the JMS, one of the main issues is the adequacy of the model used in experimental studies. The paper attempts to determine the measure of the JMS model adequacy by comparing the characteristics of two job streams, one of which was acquired from a real supercomputer and the other is obtained from the JMS model. We show that the normalized Euclidean distance between vectors of jobs residence times obtained from the job streams of the real system and the JMS model can serve as a measure of the adequacy of the JMS model. The paper also defines the reference value of the measure of adequacy corresponding to the JMS model with virtual nodes.

I. INTRODUCTION

SUPERCOMPUTER centers are usually shared facilities for the users. The users share the supercomputer computational field, which consists of computational nodes (CN) integrated by a high-performance communication network. Typically, to perform calculations on a supercomputer, the user must create a so-called passport of computational job. The passport consists of a parallel program, input data and system requirements (number of cores or nodes, amount of RAM) and execution time limit.

Special software [1] like SLURM [2], PBS [3] or the Russian native job management system SUPPZ [4] manage jobs in supercomputers. The kernel of any job management system (JMS) is the scheduler. The scheduler generates a schedule for the jobs launches according to job passports. Information in the job passport includes required execution time of job, amount and types of resources. The JMS scheduler provides quite an accurate time prediction of the launch time for each queued job. Changes to this forecast are usually made when the schedule is renewed due to a new job submission, job removal from the queue, or premature completion of a running job.

A set of indicators is used to measure the quality of scheduler. These indicators include average load, average waiting time for a job in a queue, etc. [5]. These indicators are influenced by both the configuration parameters of the scheduler and the characteristics of the input job stream. At the same time, this influence is not always evident and cannot be esti-

mated or predicted, since modern JMS are rather complicated systems with many adjustable parameters. This is why JMS modelling is relevant for studying the way that the input stream characteristics and JMS configuration influence job scheduling quality indicators.

Functioning of the real system and its simulation will be somewhat different, this results two interrelated problems. First, it is necessary to find out how to measure the model reproduction accuracy of the simulated system, i.e. to determine the measure of accuracy (adequacy) of the model. This will make it possible to compare different models by their adequacy. Secondly, it is necessary to establish the maximum valid value (limit) of the measure of accuracy. The overrun of this limit means that the model is not adequate and cannot be used to analyse the real system behaviour. The main goal of the paper is search and selection of models adequacy measure, as well as definition of the adequacy limits for the JMS models.

II. THE PROBLEM OF A SUPERCOMPUTER JOB MANAGEMENT SYSTEM MODELLING

A number of external and internal events occur during the operation of JMS. External events include job submission, premature job termination, deletion of job from a queue or job interruption by a user or an administrator, JMS start and stop, change in number of available computing nodes. The internal events include the next job launch at the appointed time according to the schedule. JMS logs the time stamp and type of each event.

We consider JMS simulation as the process of submitting the external events stream to the model input and logging the internal events stream. The model's resulting stream of the internal events should be similar to the same stream in the real system. These two streams are identical when the model is fully adequate.

Existing methods of JMS modelling can be categorized as follows:

1. Development of a JMS analytical model.
2. Experiment with a real supercomputer.
3. Study of the JMS with virtual nodes (VN) [6].
4. Development of a JMS simulation model.

This work was supported by RFBR grants no 18-29-03236 and 18-07-01325 and state assignment topic No. 0065-2019-0016

The analytical model allows investigating the impact of JMS changes on its interval indicators, but does not provide a way for predicting the launch time of individual jobs, which is necessary for forecasting. Due to the complexity of construction and orientation on interval indicators, the analytical model is not be considered in this paper.

III. NATURAL EXPERIMENT AS THE WAY TO SIMULATE A JMS

By term “natural experiment”, we mean the reproduction of an input external event stream in a real supercomputer. Therefore, the JMS model in a natural experiment will be fully adequate. Nevertheless, a natural experiment cannot provide reproduction of simulation results with 100% accuracy. In fact, processing time of a job consists of three generally random variables:

- job launch time: the time spent by the JMS for the allocation of computational nodes and their configuration in accordance with the job requirements;
- job execution time on the selected nodes;
- job completion time: the time spent by the JMS to release the selected nodes, including control the completion of all job processes, deletion of temporary files and shared resources created by the job, reconfiguration of the nodes, etc.

Job launch and completion time will be referred to as overheads. The billing subsystems for the most of the JMS include overheads into job execution time. At the same time, the proportion of overheads is a random value and can depend on many factors, such as network delays, changes in the state of calculations in the operating system kernel, etc.

The main disadvantage of a natural experiment is difficulty of its reproduction, since expensive supercomputer resources in such an experiment will duplicate the calculations already performed. Practically, a natural experiment is performed by changing JMS studied parameters. In accordance with the change of the JMS quality indicators, the decision is made whether to save the changes or to return to the previous version of the JMS settings.

IV. SIMULATION OF JMS WITH VIRTUAL NODES

Virtual nodes (VN) can be used to model the JMS. This is a software subsystem, which, instead of launching jobs on computational nodes of a computational field, makes a note that virtual nodes are engaged for the duration of the assignment. Real calculations are not performed in this case.

There are two ways to simulate a JMS with VN: in real time mode and in model time mode. In real-time VN is presented to the JMS as a computational field, the JMS actually operates in a natural experiment mode without launching jobs on a supercomputer. This allows us to speak about the accuracy of such modelling as comparable with the accuracy of a natural experiment. The disadvantage of this method is a long simulation time corresponding to the real time of the JMS operation.

The basis of JMS with virtual nodes in the model time is the idea of «advancing» system time in those moments when external or internal events do not occur. For example, if at some point of the experiment no new jobs are received, at the current moment, one job is being processed and it will be completed in an hour, then it is possible to move the system time one hour forward. Simulation in this case is significantly accelerated. To implement this method, it is necessary to develop a special software tool for advancing the system time with additional verification of the experimental results accuracy.

V. JMS SIMULATION MODELLING

To build a JMS simulation model, specialized languages can be used, like AnyLogic, ExtendSIM, Simulink [7], GPSS World [8]. Modelling languages fully provide the modelling process – the model time advancing and the interaction of objects in the system, allowing the researcher to focus on the description of the essential properties and characteristics of the simulation model.

Beside specialized modelling languages, there are so-called JMS simulators: GridSim [9], CloudSim [10], WorkflowSim [11]. Simulators supply with a set of implemented job scheduling algorithms and provide the formation of interval indicators based on the processing of the input event stream. It is also necessary to mention JMS emulators, e.g. MicroGrid [12]. A distinctive feature of the emulator is the possibility of sharing the real system components and the emulated JMS parts in the experiment.

Existing simulation tools allow us to build a predictive JMS model and conduct experiments with it on any model input event stream. However, it is necessary to validate the experiments results for simulation models in order to determine the model adequacy. To do this, it is necessary to set a measure of adequacy, express this measure by some quantitative characteristic and determine the allowable limits of this characteristic values, within which the model will be considered adequate.

VI. JMS EVENT STREAM MODEL

Let all events in the JMS occur at discrete points in time t_i . Consider the stream of independent submitted jobs $J_1, J_2, \dots, J_k, \dots, J_N$. Each job J_i in the queue has the following characteristics:

- the moment of the job submit r_i ;
- the required resources p_i ;
- ordered processing time e_i ;
- real processing time w_i , $0 \leq w_i \leq e_i$, which consists of job launch time a_i , execution time b_i , completion time c_i .

Note that the actual execution time is not available for the job management system and cannot be used to build a schedule. As shown above, a_i , b_i and c_i are random variables and can vary from launch to launch of the same job.

The scheduler determines the job launch moment s_i . Derived characteristics of the job are wait time for a job in the queue $q_i = s_i - r_i$; job residence time (full time spent in the system from submit to job completion) $f_i = q_i + w_i$; the moment of the job completion $g_i = s_i + w_i$.

An events stream with some characteristics is fed to the JMS model input. The result of the JMS model is an output model stream of events with a different characteristic set. Denote the characteristics of this stream in capital letters.

There are three well-established approaches to the formation of the input event stream [13]. The first approach is to use the real JMS event log. The approach allows reproducing the input event stream of a real supercomputer, taking into account all its features. The second approach is based on the SWF (Standard Workload Format) [14]. Event logs of some supercomputers, including university ones, published in SWF. The essential drawback is the incompleteness of the event flow: SWF represents only events related with jobs in the queue, and there is no information about changes in the nodes number, job deletions from the queue or interruptions in the job execution by the user. The third approach is to generate an input stream of events [15]. Each job parameter (submit time, ordered and real execution time, required computing resources) is a random variable with a certain distribution law. The law and distribution parameters are selected, as a rule, based on the analysis of the studied supercomputers event logs. This approach allows creating several different instances of input streams with the same distributions.

VII. JMS MODEL ADEQUACY

The variant of determining the adequacy measure proposed by the authors is based on the proposed in [16] the model's reliability evaluation method — event validity, when comparing event streams of simulated and real systems. In the paper [16], no numerical indicators allowing comparing two event streams are provided.

Let us define the proximity measure of two event streams as follows. We formulate criteria for the unreliability of the predictive model. A model is defined as unreliable if the events number in the simulation did not coincide with the number of events in the real system. If any of the events were not reproduced in the simulation, or new events have arisen, then the model is unreliable. We also consider the model unreliable if the job submit time in the model and the real system do not match, if the job execution time or ordered computing resources do not coincide. Thus, the model is unreliable if $n \neq N$, $r_i \neq R_i$, $p_i \neq P_i$, or $e_i \neq E_i$.

The number, the order and the time of occurrence of all events are coincided in the experiment and in the real system for a completely reliable model. In practice, the construction of a fully reliable forecasting JMS model is practically impossible even for a natural experiment, as shown above.

Let us consider two model streams of events represented by jobs $j = (j_1, j_2, \dots, j_n)$ and $J = (J_1, J_2, \dots, J_N)$. The job

characteristics $j_i = r_i$ (submit time), p_i (resources required), e_i (required processing time), w_i (real processing time), s_i (job launch time). Similar characteristics has the job $J_i = R_i, P_i, E_i, W_i, S_i$. The difference measure will be not determined if in the streams do not consider either the number of jobs $n \neq N$, or the submit times of any job $r_i \neq R_i$, or the ordered resources and processing times for any job $p_i \neq P_i$, $e_i \neq E_i$. In this regard, the characteristics can be rewritten as follows: $J_i = r_i, p_i, e_i, W_i, S_i$.

Let us construct two vectors of dimension $n = N$. For the stream j we define the vector of job residence times in the system $v = (v_1, v_2, \dots, v_n)$, $i \in (1, \dots, n)$, where each component corresponds to the job number in the order in which it enters the system. The value of the component $v_i = (s_i - r_i + w_i)$ is defined as the residence time of the job in the system, that is, the sum of the wait time and the processing time. For the stream J we similarly define the vector $V = (V_1, V_2, \dots, V_n)$, $V_i = (S_i - R_i + W_i)$, $i \in (1, \dots, n)$.

Thus, we obtained two vectors, v and V , the difference between the components of which actually determines the difference between the two JMS models. A natural measure of the proximity of two n -dimensional vectors is the Euclidean distance between them:

$$E = \sqrt{\sum_{i=1}^n (V_i - v_i)^2} \quad (1)$$

As shown by our experiments, the Euclidean distance increases with the number of processed jobs in the compared experiments. This dependence makes the Euclidean distance inapplicable as a measure of adequacy. We will normalize measure (1) and obtain the measure of the difference P of the streams j and J :

$$P = \sqrt{\frac{\sum_{i=1}^n (V_i - v_i)^2}{n}} \quad (2)$$

The measure of the difference P (2) does not depends with the number of jobs processed. This fact makes it possible to use the measure P as a measure of the model adequacy for experiments of any duration.

VIII. THE REFERENCE VALUE OF THE MEASURE THE JMS MODEL ADEQUACY

The following method is proposed for determining the adequacy measure. The stream j is determined based on the statistics analysis of a real supercomputer work over a sufficiently long period, and so is the vector v on stream j basis. The events s_i related to the moments of launching jobs (internal scheduler events) are excluded from the stream j . The selected substream of external events is fed to the JMS model input, and as a simulation result, the stream J and the corresponding vector V are generated. The measure P of the difference between the streams is calculated. The smaller the value of P , the more adequate the JMS model.

When $P = 0$, the JMS model will be completely reliable. The question arises about the maximum permissible value of

the measure P_{\max} , such that a model with an adequacy measure $P \leq P_{\max}$ will be considered adequate.

As was shown above, the repetition of a natural experiment does not give a precise reproduction of the result. At the same time, since the real JMS is adequate to itself, some measure P_{ideal} of the difference between the streams j and J , obtained during two repetitions of the same natural experiment, by definition will be less than the acceptable adequacy limit: $P_{\text{ideal}} \leq P_{\max}$.

Let us call P_{ideal} the reference value of the adequacy measure. Any model that has an adequacy measure less than or equal to the reference value does not differ in its behaviour from the real system.

Since, for the reasons listed above, carrying out two identical natural experiments in practice is very difficult, it is proposed to determine the adequacy measure reference value by comparing the results of JMS simulation with virtual nodes. We formed a model stream of 1000 jobs based on the statistics of the supercomputer MVS-10P OP installed in the JSCC RAS. This stream was used to model a Russian job management system SUPPZ with virtual nodes. The results are presented in Table 1. The column «number of jobs» corresponds to the number k of the first jobs of the stream j (the real SUPPZ) and the stream J (the SUPPZ with virtual nodes). The column «the number of different jobs» indicates the number of jobs for which the wait times were different in the streams j and J . From table 1 we can conclude that the reference value of the JMS model adequacy measure, calculated by the formula (2), is equal to 12.

III. CONCLUSION

This paper attempts to determine the JMS model adequacy measure by comparing the characteristics of two job streams, one of which is derived from a real supercomputer and the other is derived from the JMS model. Each job in these streams is associated with a set of events – entering the queue, launching, completion. The authors reduced all the events of the job stream into a single vector, in which each component corresponds to a specific job and contains the time that job has spent in the system. The following pairs of vectors are explored in the article: the first vector was acquired from the job streams in the real system and the second one was the generated by JMS model.

TABLE I.
MEASURES OF JOB STREAMS DIFFERENCE FOR THE SUPPZ WITH
VIRTUAL NODES

Number of jobs (size of compared vectors)	Measure of stream difference	The number of different jobs
50	0	0
100	12.0	4
250	11.4	13
500	12.0	20
750	11.6	28
1000	11.2	35

It is shown that the normalized Euclidean distance between the vectors in the pair can be used as a JMS model adequacy measure. Besides that, the paper defines the adequacy measure reference value corresponding to the JMS model with virtual nodes.

REFERENCES

- [1] A. Reuther, et al., “Scalable system scheduling for HPC and big data,” in *Journal of Parallel and Distributed Computing*, vol. 111, 2018, pp. 76–92. <https://dx.doi.org/10.1016/j.jpdc.2017.06.009>
- [2] A.B. Yoo, M.A. Jette, M. Grondona, “SLURM: Simple Linux Utility for Resource Management,” in *Lecture Notes in Computer Science*, vol 2862, 2003, pp. 44–60. https://dx.doi.org/10.1007/10968987_3
- [3] R.L. Henderson, “Job scheduling under the Portable Batch System,” in *Lecture Notes in Computer Science*, vol 949, 1995, pp. 279-294. https://dx.doi.org/10.1007/3-540-60153-8_34
- [4] SUPPZ. (In Russian) URL: <http://suppz.jssc.ru/> (accessed: 23.04.2019).
- [5] A.V. Baranov, D.S. Lyakhovets, “Comparison of the Quality of Job Scheduling in Workload Management Systems SLURM and SUPPZ,” in *Scientific Services & Internet: All Facets of Parallelism: Proceedings of the International Supercomputing Conference*, 2013, pp. 410–414 (in Russian).
- [6] N.A. Simakov et al., “A Slurm Simulator: Implementation and Parametric Analysis,” in *Lecture Notes in Computer Science*, vol 10724, 2017, pp. 197-217. https://dx.doi.org/10.1007/978-3-319-72971-8_10
- [7] I.M. Yakimov, M.V. Trusfus, V.V. Mokshin, and A.P. Kirpichnikov, “AnyLogic, ExtendSim and Simulink Overview Comparison of Structural and Simulation Modelling Systems,” in *Proc. 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC)*, Vladivostok, 2018, pp. 1-5. <https://dx.doi.org/10.1109/RPC.2018.8482152>
- [8] S.W. Cox, “GPSS World: A brief preview,” in *1991 Winter Simulation Conference Proceedings*, Phoenix, AZ, USA, 1991, pp. 59-61. <https://dx.doi.org/10.1109/WSC.1991.185591>
- [9] S.R. Chelladurai, “Gridsim: a flexible simulator for grid integration study,” 2017. <https://dx.doi.org/10.24124/2017/1375>
- [10] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, and R. Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” in *Softw.: Pract. Exper.*, 2011, pp. 23-50. <https://dx.doi.org/10.1002/spe.995>
- [11] W. Chen, and E. Deelman, “WorkflowSim: A toolkit for simulating scientific workflows in distributed environments,” in *IEEE 8th International Conference on E-Science, Chicago, IL*, 2012, pp. 1-8. <https://dx.doi.org/10.1109/eScience.2012.6404430>
- [12] H. Xia, H. Dail, H. Casanova, and A.A. Chien, “The MicroGrid: using online simulation to predict application performance in diverse grid network environments,” in *Proc. of the 2d Int. Workshop on Challenges of Large Applications in Distributed Environments*, 2004, pp. 52-61. <https://doi.org/10.1109/clade.2004.1309092>
- [13] W. Cirne, and F. Berman, “A model for moldable supercomputer jobs,” in *Proc. 15th International Parallel and Distributed Processing Symposium. IPDPS 2001*, San Francisco, CA, USA, 2001, p. 8. <https://dx.doi.org/10.1109/IPDPS.2001.925004>
- [14] Standard Workload Format. URL: <http://www.cs.huji.ac.il/labs/parallel/workload/swf.html> (accessed 24.04.2019)
- [15] U. Lublin, D.G. Feitelson, “The workload on parallel supercomputers: modeling the characteristics of rigid jobs,” in *Journal of Parallel and Distributed Computing*, vol. 63, issue 11, 2003, pp 1105-1122. [https://dx.doi.org/10.1016/S0743-7315\(03\)00108-4](https://dx.doi.org/10.1016/S0743-7315(03)00108-4)
- [16] B.M. Glinsky, A.S. Rodionov, M.A. Marchenko, D.I. Podkorytov, and D.V. Weins, “Agent-Oriented Approach to Simulate Exaflop Supercomputer with Application to Distributed Stochastic Simulation,” in *Bulletin of the South Ural State University, Series «Mathematical Modelling, Programming & Computer Software»*. 2012, no 18(277), pp. 93-106 (in Russian).