Time characteristics assessment of performance of information processing system using modified GERT-network

M. I. Tsepkova[1], **A. A. Stupina**[1,2,3], **L. N. Korpacheva** [1], **A. V. Fedorova**[1], **Z. N. Shmeleva**[3]

[1] Siberian Federal University, 79 Svobodny pr., 660041 Krasnoyarsk, Russia
[2] Reshetnev Siberian State University of Science and Technology, 31 Krasnoyarsky Rabochy Av., 660037 Krasnoyarsk, Russia
[3] Krasnoyarsk State Agrarian University, 90 Mira Av., 660049 Krasnoyarsk, Russia

E-mail: marie1206@yandex.ru

Abstract. When analyzing the efficiency, performance and manageability of the information processing system, there arises the task to assess the time characteristics of the system or its nodes, as well as to estimate the task execution time in different startup modes under different conditions. Modified GERT-networks are used to solve this problem. The article describes the operation of the forward and reverse algorithms for calculating the modified GERT-network, as well as provides their analytical comparison.

1. Introduction

Currently, the issues of fault tolerance of information processing systems are becoming more acute. There are several reasons for this. First, the systems become more complex every year, include an increasing number of components, and therefore, the probability that one of the system components will fail increases. This can cause the entire system to fail if it is not designed to solve such a problem.

A large class of tasks does not require high performance processing of a single dataset. Much more important is the ability to use a reliable information processing system with acceptable performance and the ability to process multiple data sets at the same time.

2. Results and Discussion

Today, information processing systems are increasingly used to solve management, research and production problems. One of the types of such systems are heterogeneous computing systems of high reliability for high performance computing (high throughput processing systems). The most famous technologies: Legion, Condor, Apples PST, Netsolve, Punch, XTRemweb, etc. use simple distribution schemes when the central computer, responsible for the distribution, decides which tasks should be performed on which resource, using the cost functions specified by the system parameters. They do not consider the cost of using each resource, which means that the importance of running all applications at any time is the same, which in reality is far from the case. The importance should increase with the approach of the deadline for the application task [1].

Heterogeneity of the composition of computing nodes and unpredictable changes in the computing environment during the solution of the problem lead to the problem of rational use of computing power concentrated in the network.

Analysis of existing technologies and software for information processing shows that the process of developing applications using the network as a computing resource, is complex because it contains many stages, ranging from the development of a parallel algorithm and ending with the monitoring of resources and load distribution.

The problem of load distribution in parallel calculations is one of the most important. And it is the solution of this problem that mainly determines the effectiveness of the parallel solution of the problem, i.e. the gain in time that can be obtained in comparison with the sequential solution.

Some of the tools under consideration implement load balancing techniques, which reduce the complexity of application development when using these tools. But these methods do not provide rational use of network resources, because they do not exclude idle computers. For example, in ADM (ApplicationDataMovement), the accuracy of the distribution depends on some function that the developer must write. Metacomputing, Condor requires the resource script from the developer and produces the distribution of the load on the basis of this description. This does not take into account the actual computer load, which may change during the calculation. Metacomputing Piranha simply selects one of the free computers at random, not taking into account the actual capabilities of the computers and the required resources for the task. SunGridEngine and Netsolve perform load balancing on the basis of network monitoring data, where they collect information about the workload of computing nodes at the current time and, accordingly, send the task to the least loaded computer. Such an approach provides a more accurate load distribution that does not require any data or actions from the developer, but it creates additional overhead expenses for monitoring.

The above mentioned metacomputations do not provide optimal use of network computing resources, taking into account the minimization of the problem solution time and overhead expenses [2, 3].

Most research tasks are aimed at creating methods of information processing and management of the information processing system, seeking to bring the practical duration of work (performance, acceleration factor) to theoretically possible for this information processing system. To do this, the system of collecting and processing information about the work of the computer system and methods for assessing the maximum practical and theoretical performance of the information processing system for specific tasks are created and improved.

When analyzing the efficiency, performance and manageability of the information processing system, the task arises to assess the time characteristics of the system or its nodes, as well as to estimate the task execution time in different startup modes under different conditions. To optimize the loading (scheduling) of the information processing system, as a rule, methods of optimization theory are used. Petri nets and their modifications are used to model the system behavior and estimate the time characteristics of the system. In the study of the system, taking into account the possible unreliability of nodes, Markov chains or stochastic networks are usually used.

When analyzing the time characteristics of the nodes of a distributed heterogeneous processing system that uses users' personal computers as nodes during their "downtime", these methods become inapplicable. Such systems are not static in time. The "availability" and "fault tolerance" are important factors for their nodes [4].

Therefore, to solve the identified problem, modified GERT-networks are used that allow to use all six types of nodes; to evaluate the temporal characteristics of the stochastic GERT-network using an arbitrary number of additional real and stochastic parameters of the node of the stochastic GERT-network; to use arbitrary functions calculated at the time of node activation as conditional probabilities of the outgoing arc of the node.

Let the directed graph Gconsist of a non-empty set $E$of directed edges (arcs) and a non-empty set$V$of vertices (nodes). Let each edge be uniquely determined by its start and end nodes (the edge $(i, j)$is defined by the start node $i$and the end node j).

Each node in the network is activated with a certain probability. For stochastic GERT-network, the weight of arc $(i, j)$is the vector $[p_{ij}, F_{ij}]$, where p_{ij} – is the conditional probability of the arc execution (i, j)upon activation of the node i (the probability of the arc execution (the work) (i, j)), andF_{ij} – the conditional distribution function of execution time of arc (i, j), provided that (i, j) is performed.

Each node of network has input and output activation functions that also affect the parameters of the activated node.

Types of input functions.

AND-function – the node is activated if all arcs included in it are executed.

IOR-function – the node is activated if any arc entering it is executed.

EOR-function – the node is activated if any arc entering it is executed, provided that only one arc entering the node can be executed at a given time.

Types of output functions.

The deterministic function – all arcs coming from the node are executed if the node is activated.

The stochastic function – exactly one arc coming from the node is executed with a given probability if the node is activated.

By combining all input and output functions, we obtain six different node types.

Activating of the node means that the system has entered a certain state and determines many possible further actions. One or more actions start immediately after the node that starts them is activated. The node is activated if its input function is complete. After the output function of the activated node (the beginning of the corresponding arc) is executed, it becomes inactive.

GERT-network is the network with sources R and flow so Stype "workonarc", in which each node belongs to one of six node types, for each arc $(i, j)$the weight of the type is determined $[p_{ij}, F_{ij}]$ with the above value, the initial distribution of network sources is set.

Let's introduce the symbols:

$P(i) = \{j \in v | (i, j) \in E\}$;

$S(i) = \{j \in v | (i, j) \in E\}$;

$R(i) =$\{set of nodes reachable from the node$i$ (including$i$)\};

$\bar{R}(i) =$\{set of nodes from which the node i is reachable (including$i$)\}.

Let's designate $G(i)$, where $i \in R$ – subnetwork of network G, built on a set of vertices$R(i)$.

Let's designate Ψ the set of sequences of activation of the network.

The set of sequences of activation Ψ is called reachable, if for any $W_1, W_2 \in \Psi$, $W_1 \neq W_2$, pathsW_1иW_2donotcross, excluding the case, if they use the same deterministic start node and can use the same end node.

LetD_{ij}^{n} –be execution time in n-th time of arc(i, j).

MG-network is GERT-network$G(V, E)$if:

1. G has a single source and at least one flow.

2. Network G satisfies the constraints:

a. During each project execution, no more than one source from which a given flow is reachable is activated for each flow.

b. For each node of i MG-network, if the node i is activated, then the parameters of all the arcs coming out of it are computable.

To calculate the parameters of nodes with AND- and IOR-inputs, it is necessary not only to calculate the parameters of all incoming arcs, but also to know the probability of activation of the node that "generated" the process of simultaneous execution of events.

Let the node $i \in V$GERT-network$G(V, E)$have more than one predecessors$(|P(i)| > 1)$, than for any$j, k \in P(i)$we introduce function$Pr(i, j, k)$ – the set of nodes that are the closest common predecessors to the node$i$and paths ending in arcs$(j, i), (k, i)$:

$\Pr(i, j, k) = \{l | l \in \bar{R}(j) \cap \bar{R}(k), S(l) \cap \bar{R}(j) \cap \bar{R}(k) = \emptyset\}$.

Similarly, for the node $\mathrm{i} \in \mathrm{V}$GERT-network$\mathrm{G(V, E)}$, having more than one descendant $(|\mathrm{S(i)}| > 1)$, for any $\mathrm{j, k} \in \mathrm{S(i)}$weintroducefunction$\mathrm{Sc(i, j, k)}$ – the set of nodes that are the closest common descendants of the node$\mathrm{i}$and paths, starting with arcs$\mathrm{(i, j), (i, k)}$:

$\mathrm{Sc(i, j, k)} = \{\mathrm{l} \in \mathrm{V} | \mathrm{l} \in \mathrm{R(j)} \cap \mathrm{R(k)}, \mathrm{P(l)} \cap \mathrm{R(j)} \cap \mathrm{R(k)} = \emptyset\}$.

c. For any node k of an arbitrary cyclic structure Cthereis a path from$k$to the node out of$C$, so that $p_{ij} > 0$ for each arc$(i, j)$of this path.

d. For any node of i GERT-network G, having IOR- or AND-input, for any$j, k \in P(i)$, $Pr(i, j, k) = \{1\}$, with 1 – the only node and 1 has a deterministic output.

e. For any node of i GERT-network G, having a deterministic input, for any$j, k \in S(i)$, $Sc(i, j, k)$ =$\{1\}$, with 1 – the only node and 1 has AND- or IOR- input.

f. Implementation of a network is valid, if in the process of implementing each of the activated nodes is not activated more than 1 time; implementation of a network is valid, if in the process of execution of each of the activated network nodes is activated with probability, larger $P_{max} > 0$; if in the node$i$of cyclic structure$C$more than one arc is included and at least one arc does not belong to the cycle C, than the node$i$has EOR-input; if from the node$i$of cyclic structure$C$more than one arc exits and at least one arc does not belong to the cycle$C$, than the node$i$has stochastic output; if the node$i$with IOR- or AND-input belongs the cycle, than the node$j$with deterministic output, which is the stochastic beginning of the node$i$, belongs to this cycle. This ensures that for any network the number of implementations is finite, hence the network is computable using a computer.

3. Set of parameters for each node of the network (at least the probability of activation) is established.

4. For each arc the transformation function parameters of the nodes in the network are shown.

5. The source is activated at time 0.

Under the stochastic parameters of nodes, we always mean the function of distribution of real random variables.

The result of the network calculation will be many implementationsR_r, grouped by flowsr_k. For flow in each implementation, the probability of node activation (the probability of the implementation occurring), the runtime distribution function of the implementation, and any additional parameters will be calculated. Let$r$ – be the element of the set of implementations$R_r$with flowr_k.

The probability of activation of flow r_k: $P_{rk} = \sum_r p_r$.

The probability of activation of flow$r_k$totime$t$under the condition of activation of flow r_k (the distribution function of the run-time flowr_k): $F_{rk}(t) = p_{rk}(t)/P_{rk} = \sum_r p_r * F_r(t)\,/P_{rk}$.

Real or stochastic variables can be used as additional variables in the MG-network. For each real variable, its initial value in the network source is defined, and each arc contains a function to convert the value of the variable. For each stochastic variable, its initial distribution in the network source is determined, and each arc contains a conditional distribution function of some random variable and an indication of one of the operations:« $+$ », « $-$ », « $=$ » (assign) [5].

The result of the direct and inverse algorithms for the calculation of MG-network is a set of realizations that satisfy the constraint e.

General description of the inverse algorithm for MG-network calculation:

- bypass of the graph of MG-network is from the selected flow to the source;
- bypass is performed using a recursive algorithm;
- if the node has an EOR-input, a recursive call to the network bypass procedure is triggered for each incoming arc;
- if the node $i$has IOR- or AND-input, then the recursive call to the calculation procedure is started for the node j, being the deterministic source for the node$i$, than all the paths from j to i are calculated and, using found paths all implementations ending with the node i are built;
- calculation of real and stochastic parameters is made at the “exit” of the algorithm from the recursion.

The general description of the direct algorithm of calculation of the MG network:

- bypassing of the graph of MG-network is conducted from the source to the flows;
- bypass is performed using a recursive algorithm;
- if the node has a stochastic output, a recursive call to the network bypass procedure is triggered for each outgoing arc;
- if the node $i$has the deterministic output, all paths from $i$to$j$ are calculated and, using found paths, all implementations ending with the node j are built, where j – is the deterministic node flow$i$;
- the calculation of real and stochastic parameters is made at the “deepening” of the recursion algorithm.

The distribution functions are given by a set of nodes on a uniform grid. The first order formula is used to approximate the derivative of the distribution function, since it gives strictly non-negative values of the distribution density functions. For numerical integration of the convolution integral depending on the chosen calculation mode, either the quadrature trapezoid formula or the fast Fourier transformation are used [6, 7, 8].

3. Conclusion

According to the results of the analytical comparison, the direct algorithm does not work slower than the reverse, but requires the restriction e.

Thus, an arbitrary mathematical model of MG-network with six types of nodes can be calculated on a computer.

References

[1] Sazonov A.A., Sazonova M.V. 2015 Heterogeneous data processing systems via the method GERT Information technologies in science, management, social sphere and medicine 819-820.

[2] Wang H.-H., Zhu J.-J., Yao Y.-C. 2019 GERT network optimization with consideration of "time-resource" on large aircraft collaborative development *Kongzhi yu Juece/Control and Decision.* **34(2)** 309-316.

[3] Tsepkova M.I., Stupina A.A., Korpacheva L.N., Fedorova A.V. and Dzhioeva N.N. 2015 Analysis of work units of the distributed information processing system using gert-networks *Modern problems of science and education*. **2** (53).

[4] Miroshnik M.A. 2013 Synthesis of distributed computing environments on the basis of computer networks *Information processing systems.* **7 (114)** 86-89.

[5] Gergel A.V., Vinogradov R.V. 2002 Evaluation of the complexity of communication operations in cluster computing systems *High-performance parallel computing on cluster systems. Materials of the second International scientific-practical seminar / Editorprofessor. R.G. Strongina. Nizhny Novgorod: Nizhny Novgorod state University publishing house*, 73-77.

[6] Stupina A.A., Kozhevnikov S.V. and Pisman D.M. 2006 Analysis of time parameters of network models based on the modified GERT-network *Problems of mechanical engineering and automation.* **1** 55-61.

[7] Pisman D.M. 2006 Comparison of the performance of forward and reverse algorithms for calculating the modified GERT-network *Fundamental study*. **2** 45-47.

[8] Stupina, A.A., Shigina, A.A., Shigin, A.O. 2015 Mathematical formulation of technological processes optimization problem *IOP Conference Series: Materials Science and Engineering* **94(1)**.