Bull. Nov. Comp. Center, Comp. Science, 44 (2020), 17–22

Using GPU for the network reliability evaluation by Monte Carlo method*

D. Migov, T. Snytnikova

Abstract. The problem of the network reliability calculation is studied. As the network reliability we mean the classical reliability measure, i.e. the network probabilistic connectivity. It is assumed that the network has unreliable communication links and perfectly reliable nodes. The problem of computing this characteristic is known to be NP-hard. For solving this problem, we use GPU (graphics processing unit). The parallel methods for exact calculation and evaluation of the network reliability, based on the associative processor are proposed. The numerical experiments along with comparison with experiments of the network reliability calculation on CPU are presented.

Introduction

The reliability analysis is an important part of the network design. In the present paper, we consider the networks where the links are subject to random failures under the assumption that failures are statistically independent. Random graphs are commonly used for modeling such networks. As a rule, the network reliability is defined as some connectivity measure. The most common reliability measure of such networks is the probability that all the terminal nodes in a network can keep connected together, given the reliability of each network node and edge. The problem of calculation of the network probabilistic connectivity (PC) is known to be NP-hard [1].

Nevertheless, it is possible to conduct the exact calculation and evaluation of reliability for networks with dimension of a practical interest by taking into consideration some special features of real network structures and based on modern high-speed computers. However, despite the improvements achieved of the efficiency of the computational methods for reliability analysis, they are still ineffective and so their parallel realizations are needed for executing on modern supercomputers. By now, we have in this area only the parallel approach for estimating the network reliability by Monte Carlo technique [2, 3], and the parallel implementation of the well-known factorization method, which was proposed in one of our previous publication [4].

In this paper, we propose parallel implementations of the enumeration-based method for calculating PC. Also, a parallel PC estimation method based on statistical modeling is presented. Both algorithms are associative

*Supported by Russian Foundation for Basic Research under Grant 18-07-00460.

and were developed for the use on the GPU. The GPU is used to emulate an associative processor.

1. The basic definitions and notations

We represent a network with unreliable edges and perfectly reliable nodes by an undirected graph $G = (V, E)$. The number of edges we denote as M, the number of nodes we denote as N. For each edge e, the presence probability $0 \leq p_e \leq 1$ is given. Further on we refer to this probability as edge reliability. In the general case, we have a set of terminals K also. It is supposed that the network operates well when any pair of terminals can establish a connection via only the operational edges. Two cases are most important: $|K| = 2$ and $K = V$, which corresponds to 2-terminal reliability and all-terminal reliability. The last measure is often called the probabilistic connectivity. Further on, we consider this measure.

An elementary event is a special realization of a graph defined by the existence or absence of each edge. The probability of an elementary event equals the product of probabilities of the existence of operational edges times the product of probabilities of the absence of faulty edges:

$$P(E) = \prod_{i \in E_V} p_i \prod_{i \notin E_V} (1 - p_i), \tag{1}$$

where E_V is the set of numbers of existing edges.

An arbitrary event (a union of some simple events) we consider to successful if all graphs that correspond to these simple events are connected, that is, that all nodes can be connected by existing edges.

The reliability of the graph G, $R(G)$, is the probability of the graph G being connected, that, is the probability of the event that is the union of all successful events and of them only.

2. Methods for the network reliability calculation

The definition of the network reliability gives a method for computation of this parameter. However, such a direct approach leads us to an exhaustive search for all graph realizations, whose quality is equal to 2^M. Thus, this method may be used only for small-scale networks. That is why the other methods are used for the calculation of different reliability measures of a medium and a large scale. The most common method among them is the factorization method [5, 6], which can be applied to any network reliability measure.

The factorization method divides the probability space into two sets, based on the success or failure of one graph particular unreliable element: a node or an edge. We consider networks with unreliable edges only, so we may choose any unreliable edge. The chosen edge is called pivot. So, we obtain two subgraphs, in one of them the pivot edge is absolutely reliable and in the second one the pivot edge is absolutely unreliable, that is, absent. The probability of the first event is equal to the reliability of the pivot edge; the probability of the second event is equal to the failure probability of the pivot edge. Thereafter obtained subgraphs are subject to the same procedure. The total probability law gives the expression for the network reliability:

$$R(G) = p_e R(G_e^*) + (1 - p_e) R(G \setminus e), \tag{2}$$

where p_e is the reliability of the edge e, G_e^* is the graph obtained by contracting the edge e from G, $G \setminus e$ is the graph obtained by deleting e from G. Recursions continue until either a disconnected graph is obtained, or until a graph for which the probabilistic connectivity can be calculated directly is obtained, i.e. it can be a graph of a special type or a small-dimension graph.

One of the way to avoid the exhaustive search is to randomly generate a given number of random subgraphs (taking into account the probabilities of the presence of edges), instead of enumerating all of them. On this basis, it is possible to evaluate the reliability by Monte Carlo method, which is equal to the ratio of the number of connected implementations to the number of all implementations. In this case, based on the variance of a sample, it is possible to make a conclusion about the error of the solution obtained [7].

3. The network reliability calculation using GPU

When calculating the probability of the network connectivity with unreliable edges according to the definition of this indicator, it is necessary to process all subgraphs (implementations) of the network graph. The number of such subgraphs grows exponentially with the number of edges of the original graph. For each subgraph, the fact of its connectivity is established, then the probabilities of obtaining connected realizations are summed up to obtain the network reliability value.

So, to exactly calculate the reliability of the 4×4 grid network (16 vertices, 24 edges), it is necessary to check the connectivity of more than 2.5 million subgraphs. It is possible to do this using GPU. In our realization the subgraphs are processed in batches of 8,196, which takes about 1.3 seconds using an Nvidia GeForce 920m graphics card. But for the 5x5 grid network, calculation cannot be performed in a reasonable time.

Therefore, instead of a complete enumeration of all subgraphs, we can generate a given number L of random subgraphs (taking into account the probabilities of the presence of edges) and, on its basis, estimate the reliability using Monte Carlo method [7, 8], which is equal to the ratio of the number of connected realizations to the number of all realizations.

To check a random subgraph for the connectivity, we represent the graph as a list of edges. Since the subgraph connectivity check is running independently for each subgraph, then each block of GPU is used to generate and to check a random subgraph, whereas a thread corresponds to some edge. The library cuRAND to generate subgraph is used.

First, subgraphs are generated and checked on each of the blocks. The result of the first stage is an array, where the ith element saves the count of connected subgraphs on the ith block. The technique of checking connectivity is similar to the associative parallel algorithm from [9]. Thereafter, all the elements are summed.

Calculations were performed using a graphics card GeForce 920m. We assume that reliabilities of all the edges are equal to each other. Let us denote this value by P. The dependence of the running time for the parameters P and L is shown in Table 1. The GEANT network structure in 2009 (the figure) was used for calculations.

Comparison of the time for estimating the reliability of the GEANT network by different methods for $L = 10^7$ is given in Table 2. Corresponding Monte-Carlo method on CPU and numerical experiments with the use of Intel Core Duo 2.4 GHz were represented in [7].

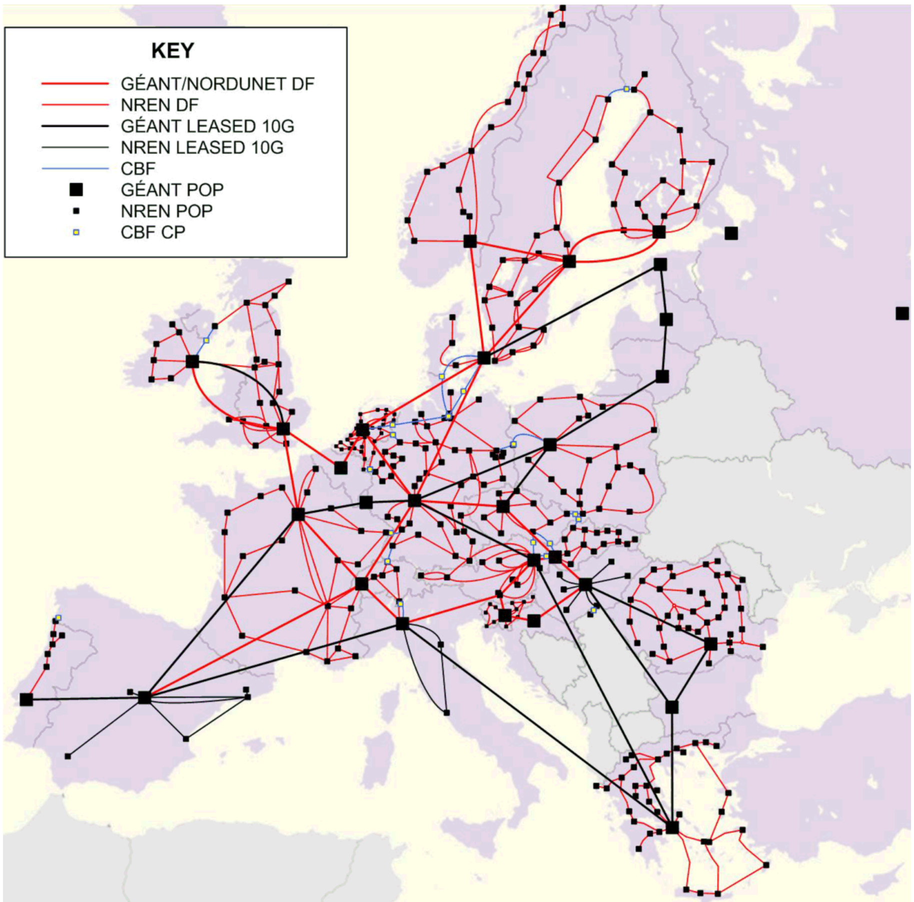
The analysis of execution has shown the following. On the one hand, the processing time for a connected subgraph is slightly longer than for a disconnected one. On the other hand, the proportion of connected subgraphs among the generated subgraphs significantly decreases with decreasing P. Obviously, the runtime linearly depends on L, taking into account the additive constant. The main advantage of using GPU in the presented way in compassion with CPU using for the network reliability evaluation is a weak dependence between the runtime and the edge reliability.

Table 1. Running time of GPU method (s)

L	$P = 0.75$	$P = 0.9$	$P = 0.99$	$P = 0.999$
10^5	22.85	23.00	23.16	23.18
10^6	27.18	29.10	30.65	30.77
10^7	69.60	88.50	103.65	104.81
10^8	493.72	680.85	833.82	843.92

Table 2. Time comparison on the GEANT network in 2009

Processing by	$P = 0.75$	$P = 0.9$	$P = 0.99$	$P = 0.999$
CPU method	45	72	314	402
GPU method	70	89	104	105



GEANT network structure in 2009

Conclusion

In this paper, we introduce the parallel implementation of Monte Carlo method for the evaluation of the network reliability and parallel implementation of exhaustive search for exact calculation of network reliability. Both methods were developed for the use on the GPU, which is used to emulate an associative processor. The main advantage of such an approach, as conducted numerical experiments show, is the dependence between the runtime and the edge reliability, which is not so strong as for Monte Carlo network reliability evaluation with the use of CPU.

References

[1] Colbourn Ch. J. The Combinatorics of Network Reliability.—New York: Oxford University Press, 1987.

[2] Khadiri M.E., Marie R., Rubino G. Parallel estimation of 2-terminal network reliability by a crude Monte Carlo technique // Proc. of the 6th international symposium on computer and information sciences. — Elsevier, 1991. — P. 559–570.

[3] Martinez S.P., Calvino B.O., Rocco S.C.M. All-terminal reliability evaluation through a Monte Carlo simulation based on an MPI implementation // Proc. of the European safety and reliability conference: advances in safety, reliability and risk management (PSAM 2011/ESREL 2012). — Helsinki, 2012. — P. 1–6.

[4] Nesterov S., Migov D. Parallel method for reliability calculation of diameter constrained networks // Bull. Novosibirsk Comp. Center. Ser. Numerical Analysis. — Novosibirsk, 2015. — Iss. 17. — P. 35–42.

[5] Rebaiaia M.-L, Ait-Kadi D., Merlano A. A practical algorithm for network reliability evaluation based on the factoring theorem – a case study of a generic radiocommunication system // J. Quality. — 2009. — Vol. 16, Iss. 5. — P. 323–336.

[6] Nath M., Ren Yi., Eubank S. An approach to structural analysis using Moore–Shannon network reliability // Proc. of the Int. Conf. on Complex Networks and their Applications VII (COMPLEX NETWORKS 2018). — Springer, 2019. — P. 537–549. — (Studies in Computational Intelligence; 812).

[7] Migov D.A., Weins D.V. Parallel implementation and simulation of network reliability calculation by Monte Carlo method // Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitelnaja tehnika i informatika [Tomsk State University Journal of Control and Computer Science]. — 2018. — Vol. 47. — P. 66–74 (In Russian). DOI: 10.17223/19988605/47/8.

[8] Cancela H., Robledo F., Franco R., et al. Monte Carlo estimation of diameter-constrained network reliability conditioned by pathsets and cutsets // Computer Communications. — 2013. — Vol. 36. — P. 611–620.

[9] Nepomniaschaya A.Sh., Snytnikova T.V. Associative parallel algorithm for dynamic update of shortest paths tree after inserting an arc // Applied Discrete Mathematics. — 2019. — Vol. 46. — P. 58–71. DOI: 10.17223/20710410/46/5.