

УДК 519.17 + 519.24  
DOI: 10.17223/19988605/47/8

Д.А. Мигов, Д.В. Винс

## ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ И ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ОЦЕНКИ НАДЕЖНОСТИ СЕТИ МЕТОДОМ МОНТЕ-КАРЛО

*Работа поддержана РФФИ, гранты № 16-07-00434, 18-07-00460, 17-47-540977.*

Рассматривается NP-трудная задача расчета надежности сети, элементы которой подвержены случайным отказам. Для оценки надежности сети используется метод Монте-Карло, улучшенный при помощи проверки связности реализации графа одновременно с генерацией этой реализации. Имитационное моделирование позволило оптимально настроить параметры параллельной версии алгоритма для повышения масштабируемости, сделав возможным его применение для приближенного расчета надежности сетей на суперЭВМ большой производительности (пета- и эксафлопсные).

**Ключевые слова:** надежность сети; случайный граф; связность; обход в ширину; параллельный алгоритм; методы Монте-Карло; имитационное моделирование.

Универсальным методом для оценки различных показателей надежности сетей является метод Монте-Карло [1–4], в том числе и в параллельной реализации [5–7]. Несмотря на принадлежность задачи точного расчета надежности сети к классу NP-трудных [8], для сетей небольшой размерности (до сотни ребер) возможно, как правило, осуществлять точный расчет надежности. Для этого используются метод факторизации (ветвления, Мура–Шеннона) и различные методы ускорения, в первую очередь методы редукции, декомпозиции, стратегия выбора элемента для факторизации, точные формулы для графов малой размерности и др. Однако подобные техники разработаны далеко не для всех показателей сетевой надежности. Больше всего таких приемов существует для классического показателя надежности – вероятности связности случайного графа. Некоторые методы ускорения расчета известны для надежности с ограничением на диаметр и для математического ожидания числа связных пар узлов сети. Отсутствие подобных методов сильно затрудняет расчет исследуемого показателя, делая его сложность близкой к экспоненциальной по числу ненадежных элементов (каналов связи и / или узлов).

Для анализа надежности сетей средней и большой размерности (сотни, тысячи элементов) точные методы неприменимы, вследствие чего используются различные приближенные методы. Большинство методов основывается на структурных и других особенностях сетей, что делает эти методы эффективными для определенных типов сетей. Наиболее известны методы на основе рассмотрения циклов, разрезов, остовных деревьев. Метод Монте-Карло является среди приближенных методов наиболее универсальным, способным за приемлемое время дать ответ. Другим универсальным методом является метод, основанный на кумулятивном уточнении границ надежности [9–10].

При расчете надежности сети методом Монте-Карло необходимо случайным образом сгенерировать определенное число реализаций сети и усреднить по полученной выборке интересующий нас показатель (связность, число связных пар и т.д.), что и будет оценкой показателя надежности. Исходя из дисперсии выборки можно сделать заключение о погрешности полученного решения. Для некоторых показателей возможно оценить дисперсию в зависимости от размера выборки еще перед началом расчета и, соответственно, определить необходимое количество итераций алгоритма для достижения требуемой погрешности решения. Для других показателей необходимо динамически вычислять дисперсию текущей выборки и принимать решение либо о продолжении расчета, либо о его окончании. Для сокращения дисперсии при расчете надежности сети методом Монте-Карло применяются различные подходы; отметим расслоенную выборку Ван Слайка [1] и рекурсивный метод RVR [2].

В данной работе в качестве показателя надежности рассматривается вероятность связности случайного графа с ненадежными ребрами. Для этого показателя мы предлагаем при генерации реализации графа делать заключение о связности реализации без дополнительного обхода графа. Также для вероятности связности возможно оценить дисперсию в зависимости от размера выборки еще перед началом расчета и определить необходимое количество итераций алгоритма для достижения требуемой погрешности решения. Такой подход также значительно упрощает параллельную реализацию, избавляя от необходимости периодически всем процессам приостанавливать расчет и ждать, пока процесс-сборщик проведет оценку дисперсии. Для исследования эффективности данного параллельного алгоритма при его исполнении на высокопроизводительном суперкомпьютере (в том числе и на потенциальном эксафлопсном суперкомпьютере) применяется имитационное моделирование на основе мультиагентного подхода. При имитационном моделировании с использованием платформы мультиагентного моделирования AGNES предполагалось, что архитектура эксафлопсного суперкомпьютера не отличается от архитектуры кластера НКС-30Т ССКЦ СО РАН.

В результате выявлено, что параллельный алгоритм идеально масштабируется до 512 000 вычислительных ядер, далее наблюдается уменьшение эффекта ускорения от количества ядер. Решением в данном случае могут быть увеличение количества агентов-«сборщиков» и их иерархическая организация. Отметим, что предложенная модель позволяет осуществлять моделирование расчета методом Монте-Карло и для других показателей сетевой надежности, для которых возможно оценить дисперсию перед началом расчета.

## 1. Случайный граф как модель сети с ненадежными элементами

Для анализа надежности сетей различного назначения используют, как правило, аппарат случайных графов. Рассмотрим произвольный неориентированный граф  $G = (V, E)$ , где  $V$  – это множество вершин, а  $E$  – множество ребер графа  $G$ . Пусть для каждого ребра  $e$  задана вероятность его присутствия в графе  $p_e$ , при этом предполагается, что вершины абсолютно надежны.

*Элементарным событием* будем называть частную реализацию графа, определяемую исправностью или отказом каждого ребра. Элементарное событие удобно представлять как список длины  $|E|$ , в котором  $i$ -й элемент равен 1, если  $i$ -е ребро присутствует в реализации, и равен 0 в противном случае.

*Вероятность элементарного события* равна произведению вероятностей присутствия исправных ребер, умноженному на произведение вероятностей отсутствия отказавших ребер.

Произвольное событие (событие есть объединение некоторых элементарных событий) будем называть *успешным*, если в каждом элементарном событии, входящем в это событие, все вершины могут быть связаны исправными ребрами.

*Вероятность связности графа  $G$ ,  $R(G)$* , есть вероятность того, что эти вершины связаны исправными ребрами, т.е. вероятность события, состоящего из всех успешных событий, и только из них.

## 2. Оценка надежности сети методом Монте-Карло

Опишем общую схему применения методов Монте-Карло к рассматриваемым задачам. Как уже упоминалось, для оценки надежности сети методом Монте-Карло нам нужно случайным образом сгенерировать определенное число реализаций сети, усреднить по полученной выборке интересующий нас показатель и получить таким образом оценку показателя надежности.

Для оценки вероятности связности графа показателем будет индикаторная функция связности  $\chi(E)$ , заданная как случайная величина на пространстве элементарных событий. Эта функция равна 1, если событие успешно, и 0 в противном случае. Факт успешности  $E$  определяется возможностью соответствующей сети успешно функционировать, исходя из практических соображений. В данном случае это связность сети.

Математическое ожидание случайной величины и будет точным значением надежности. Среднее значение по выборке будет приближенным значением надежности, сходящимся к точному с ростом размера выборки  $L$  по закону больших чисел:

$$R = M(\chi) \approx \frac{1}{L} \sum_{i=1}^L \chi_i = R_L. \quad (1)$$

Согласно ЦПТ, с ростом  $L$  вычисляемое приближение  $R$  становится все ближе к точному значению. Таким образом, по правилу трех сигм получаем следующую оценку:

$$P\left\{\left|R - \frac{1}{L} \sum_{i=1}^L \chi_i\right| \leq 3\sigma\right\} = 0,9973. \quad (2)$$

Дисперсия случайной величины равна

$$D(\chi) = M(\chi^2) - M^2(\chi) = R - R^2.$$

Соответственно, дисперсия оценки надежности (среднего значения по выборке) составит

$$D(R_L) = \frac{L \cdot D(\chi)}{L^2} = \frac{R(1-R)}{L},$$

а утроенное среднеквадратичное отклонение, которое мы используем для оценки погрешности  $\varepsilon$  (2), оценивается сверху как:

$$\varepsilon = 3\sigma = 3\sqrt{\frac{R(1-R)}{L}} \leq 3\sqrt{\frac{1}{4L}} = \frac{3}{2\sqrt{L}}. \quad (3)$$

Неравенство в центре в выражении (3) имеет место, так как  $0 \leq R \leq 1$ .

Таким образом, для количества итераций статистического моделирования

$$L = \frac{9}{4\varepsilon^2} \quad (4)$$

неравенство (3) выполняется, и получаемая оценка надежности (1) верна с погрешностью  $\varepsilon$  с вероятностью 0,9973 (2).

При параллельной реализации удобно разбить число  $L$  на максимально близкие друг к другу  $L_i$ , количество которых совпадает с количеством доступных вычислительных ядер минус 1. Далее на каждом ядре запускается свой процесс порождения реализаций графа. Предполагается, что количество вычислительных ядер достаточно велико, и для связи между ними используется MPI. Поэтому одно ядро остается как «сборщик» и не участвует в порождении реализаций графа. Остальные ядра по завершении работы пересылают усредненные значения на процесс-«сборщик».

### 3. Проверка связности графа при оценке вероятности связности сети методом Монте-Карло

Оценивание вероятности связности сети методом статистического моделирования требует многократного порождения графов, являющихся элементарными событиями, и проверки их связности. Каждая из этих операций имеет линейную по числу ребер  $M$  трудоемкость  $O(M)$ . Так, проверку связности графа удобно делать обходом графа в ширину. Несмотря на невысокую трудоемкость этих операций, их многократное использование для обеспечения нужной точности решения (сотни тысяч и миллионов раз) и большие размерности реальных сетей (сотни и тысячи элементов) делают актуальной разработку максимально быстрых методов их осуществления.

Мы предлагаем осуществлять генерацию частной реализации графа одновременно с проверкой его связности. В этом случае генерация ребер проходит при помощи обхода графа в ширину. Заводится список узлов, инициализируется как состоящий из некоторого начального узла. Определяем, какие из ребер, инцидентных узлам из списка, присутствуют в данной реализации. Если таких ребер нет – заканчиваем процесс. Если есть – формируем список смежных узлов через присутствующие инцидентные им ребра и продолжаем его обновление по указанной схеме. Если по завершении про-

цедуры количество посещенных узлов будет меньше, чем общее количество узлов, то полученная реализация несвязна. Учет появления изолированных вершин дополнительно сокращает трудоемкость.

В таблице приведены результаты для оценки надежности европейской научно-образовательной сети GÉANT образца 2009 г. (390 узла и 503 ребра; рис. 1). Предполагалось, что надежности всех ребер совпадают. Был рассмотрено четыре различных значения надежности, как показано в таблице. Для обоих методов и для каждого из рассматриваемых значений надежности ребра было проведено  $L = 10^7$  итераций метода. Расчет проводился на ПЭВМ Intel Core Duo 2.4 GHz.

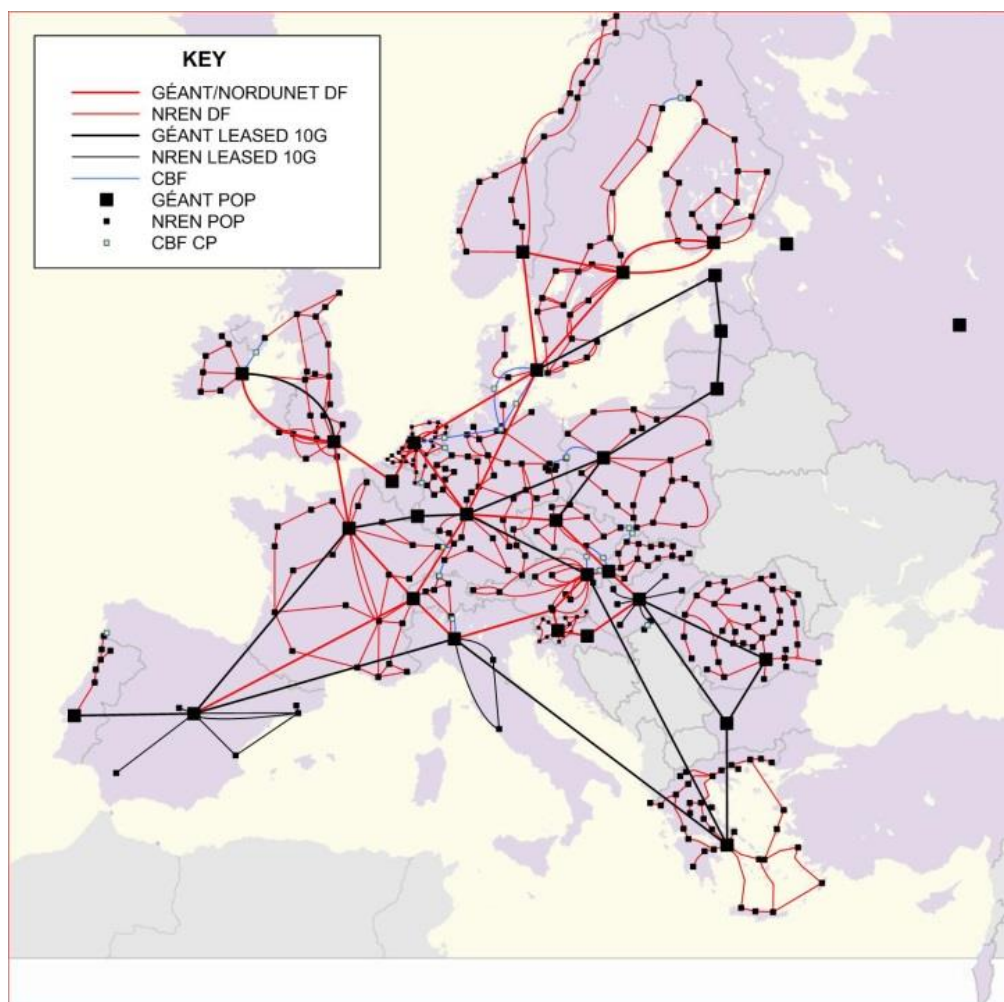


Рис. 1. Европейская научно-образовательная сеть GÉANT (2009)

Fig. 1. European research and education network GÉANT (2009)

Время оценки надежности сети GÉANT

Надежность ребра	$P = 0,75$	$P = 0,9$	$P = 0,99$	$P = 0,999$
Время расчета (с). Новый метод	45	72	314	402
Время расчета (с). Стандартный метод	67	92	421	623

Как видно, ускорение при использовании предложенного метода получается разным для различных значений надежности ребра. Максимально эффективен метод для высоконадежных сетей, а именно такие, как правило, существуют и проектируются в настоящее время. В этом случае метод дает ускорение более чем в полтора раза.

#### 4. Моделирование оценки надежности сети методом Монте-Карло на суперЭВМ

Для исследования эффективности исходного алгоритма при его исполнении на потенциальном экзафлопсном суперкомпьютере применяется имитационное моделирование на основе мультиагентного подхода. При имитационном моделировании с использованием платформы мультиагентного моделирования AGNES [11] предполагалось, что архитектура экзафлопсного суперкомпьютера не отличается от архитектуры кластера НКС-30Т.

Как известно, теоретическое ускорение при распараллеливании для методов статистического моделирования практически «идеальное», т.е. время расчета уменьшается пропорционально увеличению вычислительных ядер. Этот критерий и используется для исследования эффективности данных вычислительных алгоритмов.

Для исследования такой системы будем исходить из следующих предположений:

1.  $M$  – общее число ядер, задействованных в статистическом моделировании.
2. Уровень допустимой статистической погрешности, достижение которого означает завершение расчета:

$$\varepsilon = \frac{9}{4\sqrt{L}}.$$

3. Достижение уровня допустимой статистической погрешности достигается при совершении вычислительными ядрами определенного числа итераций вычислений  $L$ :

$$L = \sum_{i=1}^M L_i.$$

где  $L_i$  – количество итераций цикла вычислений на  $i$ -м ядре-вычислителе.

4. Время моделирования вычислительным ядром итерации  $T_{count}$  одинаково для однотипных вычислительных узлов.

5. Время выполнения коммуникаций будем считать по следующей формуле:

$$T^{Comm} = T_c + \Lambda.$$

где  $\Lambda$  – латентность при передаче данных,  $T_c$  – среднее время передачи полученных результатов.

6. Общее время выполнения алгоритма статистического моделирования по методу Монте-Карло –  $T_L(M)$ .

7. Величина относительно ускорения от распараллеливания определяется следующим образом:

$$S_L(M) = \frac{T_L(M_{\min})}{T_L(M)}.$$

Для имитации исполнительного алгоритма расчета по методу Монте-Карло создано два класса функциональных агентов: «вычислитель» и «сборщик». Агент-«вычислитель» имитирует проведение необходимого количества итераций одним вычислительным узлом, после чего отправляет результаты агенту-«сборщику». Агент-«сборщик» дожидается прихода сообщений от всех вычислителей и останавливает моделирование [12].

Исходные данные для моделирования были получены на кластере НКС-30Т, а именно:  $T_{count} = 1,53 \cdot 10^{-5}$ ,  $T_c = 2 \cdot 10^{-5}$ . Для этого были осуществлены запуски расчетов на НКС-30Т на количестве ядер до 512 с использованием новой техники генерации графов, представленной в предыдущем разделе. В качестве тестируемой структуры бралась структура GÉANT с надежностью ребра 0,999. Требуемая допустимая статистическая погрешность  $\varepsilon = 10^{-8}$ , соответственно, общее число итераций  $L = 10^{15}$ . Было исследовано поведение алгоритма при его исполнении на от 8 до 1 536 000 вычислительных ядер, полученные параметры  $T_L(M)$  и  $S_L(M)$  сравнивались с «идеальным» теоретическим ускорением. Результаты исследования приведены на рис. 1, 2.

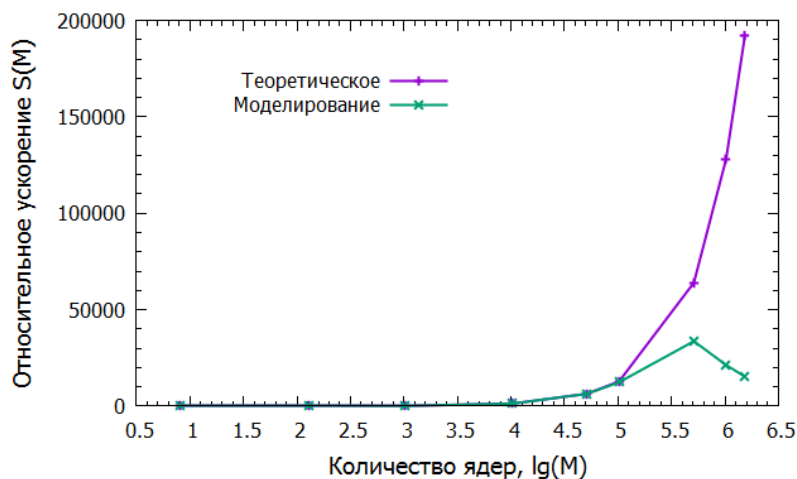


Рис. 2. Зависимость относительного ускорения  $S_L$  от общего числа моделируемых ядер  $M$  (горизонтальная ось – в логарифмическом масштабе)

Fig. 2. Dependence of the relative acceleration  $S_L$  on the total number of simulated nuclei  $M$  (the horizontal axis is on a logarithmic scale)

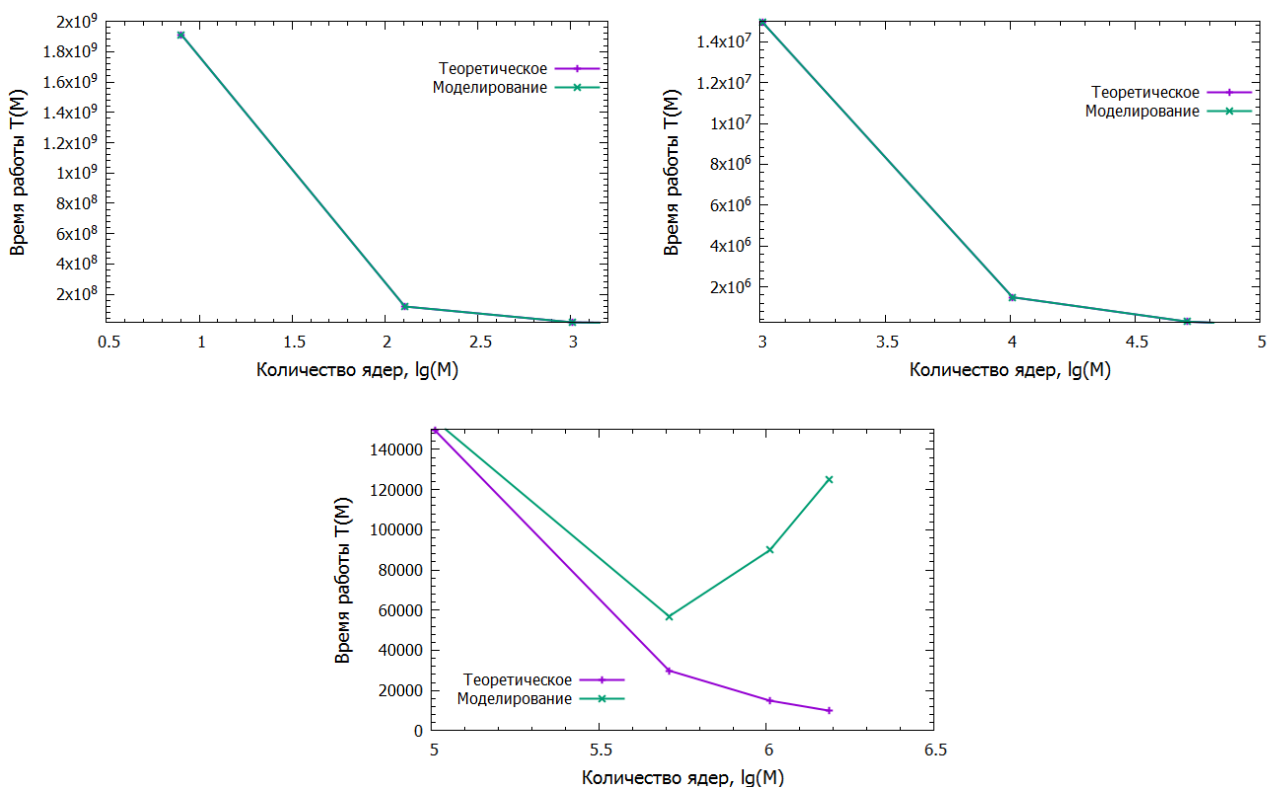


Рис. 3. Зависимость времени исполнения алгоритма  $T_L$  от общего числа моделируемых ядер  $M$  (горизонтальная ось – в логарифмическом масштабе)

Fig. 3. Dependence of the execution time of the  $T_L$  algorithm on the total number of simulated kernels  $M$  (the horizontal axis is on a logarithmic scale)

Как можно увидеть на рис. 2, 3, исходный алгоритм масштабируется до 512 000 вычислительных ядер практически идеально, однако на большем числе наблюдается уменьшение эффекта ускорения от количества вычислительных ядер. Это можно объяснить огромным количеством сообщений, которые должен принять и обработать агент-«сборщик» в конце вычислений. Решением в данном случае могут быть увеличение количества агентов-«сборщиков» и их иерархическая организация.

## Заключение

Был исследован такой показатель надежности, сетей как вероятность связности случайного графа с ненадежными ребрами. Предложена новая техника генерации графов с заключением о связности реализации без дополнительного обхода графа. Данная техника позволяет существенно ускорить (на 30–50 %) приближенный расчет методом Монте-Карло как в последовательной, так и в параллельной реализациях.

Для исследования эффективности параллельного алгоритма оценки надежности сети методом Монте-Карло при его исполнении на высокопроизводительном суперкомпьютере (в том числе и на потенциальном эксафлопсном суперкомпьютере) мы использовали имитационное моделирование на основе мультиагентного подхода. В результате выявлено, что параллельный алгоритм идеально масштабируется до 512 000 вычислительных ядер, далее наблюдается уменьшение эффекта ускорения от количества ядер. Решением в данном случае может быть увеличение количества процессов-«сборщиков», по одному на 512 000 вычислительных процессов. Предложенная модель позволяет осуществлять моделирование расчета методом Монте-Карло и для других показателей сетевой надежности, для которых возможно оценить дисперсию перед началом расчета.

## ЛИТЕРАТУРА

1. Slyke R.V, Frank H. Network reliability analysis. Part I // Networks. 1972. V. 1. P. 279–290.
2. Cancela H., Urquhart M.E. Adapting RVR simulation techniques for residual connectedness network reliability models // IEEE Trans. Computers. 2002. V. 51, No. 4. P. 439–443.
3. Botev Z.I., Kroese D.P. Efficient Monte Carlo simulation via the generalized splitting method // Statistics and Computing. 2012. V. 22, No. 1. P. 1–16.
4. Cancela H., Murray L., Rubino G. Highly reliable stochastic flow network reliability estimation // Proc. XLII IEEE Latin American Computing Conference. CLEI'16. IEEE Press, 2016.
5. Khadiri M.E., Marie R., Rubino G. Parallel estimation of 2-terminal network reliability by a crude Monte Carlo technique // Proc. of the 6th Int. Symposium on Computer and Information Sciences. 1991. Elsevier, 1991. P. 559–570.
6. 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. European Safety and Reliability Conference: Advances in Safety, Reliability and Risk Management. PSAM 2011/ESREL. Helsinki, 2012. P. 1–6.
7. Мигов Д.А. Параллельные методы точного и приближенного расчета надежности сетей // Молодежь. Наука. Технологии (МНТК-2017) : труды Междунар. конф. Новосибирск : Изд-во НГТУ, 2017. Ч. 3. С. 67–69.
8. Colbourn Ch.J. The combinatorics of network reliability. New York : Oxford University Press, 1987. 160 p.
9. Won J.-M., Karray F. Cumulative update of all-terminal reliability for faster feasibility decision // IEEE Trans. on Reliability. 2010. V. 59, No. 3. P. 551–562.
10. Rodionov A.S., Migov D.A., Rodionova O.K. Improvements in the efficiency of cumulative updating of all-terminal network reliability // IEEE Trans. on Reliability. 2012. Vol. 61, No 2. P. 460–465.
11. Podkorytov D., Rodionov A., Choo H. Agent-based simulation system agnes for networks modeling: review and researching // Proc. 6th Int. Conf. on Ubiquitous Information Management and Communication. ACM ICUIMC 2012. ACM New York, USA. Article No. 115.
12. Glinsky B., Rodionov A., Marchenko M., Podkorytov D., Weins D. Scaling the distributed stochastic simulation to exaflop supercomputers // Proc. 14th IEEE Int. Conf. on High Performance Computing and Communications. HPCC'2012. IEEE Press, 2012. P. 1131–1136.

Поступила в редакцию 17 июля 2018 г.

Migov D.A., Weins D.V. (2018) PARALLEL IMPLEMENTATION AND SIMULATION OF NETWORK RELIABILITY CALCULATION BY MONTE CARLO METHOD. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 47. pp. 66–74

DOI: 10.17223/19988605/47/8

The paper considers the NP-hard calculation problem of the reliability of a network, which elements are subject to accidental failures. A universal method for evaluating various indicators of network reliability is the Monte Carlo method, which can be performed in parallel implementation. In spite of the fact that the exact reliability calculation problem is NP-hard, it is usually possible

to perform an accurate calculation of reliability for networks of small dimension (up to hundreds of edges). To this end, the factorization method and various acceleration methods are used, primarily reduction methods, decomposition, the selection strategy for the factorization element, exact formulas for low-dimensional graphs, and others. However, such techniques are not developed for all indicators of network reliability. Most of all such techniques exist for the classical reliability index - the probability of the connectedness of a random graph. Some methods of speeding up the calculation are known for the reliability with diameter constraint, and the mathematical expectation of the number of connected pairs of nodes of the network. The absence of such methods greatly complicates the calculation of the investigated indicator, making its complexity close to the exponential in the number of unreliable elements (communication channels and/or nodes).

When calculating the reliability of a network using the Monte Carlo method, it is necessary to randomly generate a certain number of implementations of the network and to average values obtained (connectivity function, the number of connected pairs, etc.), which will be an estimate of the reliability index. Based on the sample variance, one can draw a conclusion about the error of the solution obtained. For some indicators it is possible to estimate the variance depending on the sample size even before the calculation begins and, accordingly, to determine the necessary number of algorithm iterations to achieve the required error of the solution. In this paper, the probability of connectedness of a random graph with unreliable edges is considered as an index of reliability. For this indicator, we propose to generate an implementation of the graph simultaneously with its connectivity check. In addition, for the probabilistic connectivity, it is possible to estimate the variance depending on the sample size even before the calculation begins and determine the necessary number of iterations of the algorithm to achieve the required error in the solution.

This approach also greatly simplifies the parallel implementation, eliminating the need to periodically stop all processes and wait for evaluation of the variance. In the case of parallel implementation, it is convenient to split the iteration number between all processes. Further, on each core, the process of generation of the implementations starts. It is assumed that the number of computing cores is large enough, and MPI is used for communication between them. Therefore, one core remains as a master and does not participate in the generation of graph implementations. The rest of the cores send the averaged values to the collector process after completion.

As a result, it was shown that the parallel algorithm ideally scales up to 512,000 computing cores, then there is a decrease in the acceleration effect from the number of cores. The solution in this case may be to increase the number of collector agents and their hierarchical organization. It should be noted that the proposed model allows simulating Monte Carlo calculation for other indicators of network reliability, if it is possible to estimate the variance before the calculation begins.

Keywords: network reliability; random graph; connectivity; Monte Carlo methods; simulation.

*MIGOV Denis Aleksandrovich* (Candidate of Physics and Mathematics, senior research fellow, Laboratory of System modeling and optimization of The Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russian Federation).

E-mail: mdinka@rav.sccc.ru

*WEINS Dmitry Vladimirovich* (Candidate of Technical Sciences, research fellow, Laboratory of supercomputer modeling of The Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russian Federation).

E-mail: vins@sscc.ru

## REFERENCES

1. Slyke, R.V & Frank, H. (1972) Network reliability analysis: Part I. *Networks*. 1. pp. 279–290.
2. Cancela, H. & Urquhart, M.E. (2002) Adapting RVR simulation techniques for residual connectedness network reliability models. *IEEE Trans. Computers*. 51(4). pp. 439–443.
3. Botev, Z.I. & Kroese, D.P. (2012) Efficient monte carlo simulation via the generalized splitting method. *Statistics and Computing*. 22(1). pp. 1–16. DOI: 10.1007/s11009-009-9133-7
4. Cancela, H., Murray, L. & Rubino, G. (2016) Highly reliable stochastic flow network reliability estimation. *Proc. XLII IEEE Latin American Computing Conference. CLEI'16*. IEEE Press.
5. Khadiri, M.E., Marie, R. & Rubino, G. (1991) Parallel estimation of 2-terminal network reliability by a crude Monte Carlo technique. *Proc. 6th International Symposium on Computer and Information Sciences 1991*. Elsevier. pp. 559–570.
6. Martinez, S.P., Calvino, B.O. & Rocco, S.C.M. (2012) All-terminal reliability evaluation through a Monte Carlo simulation based on an MPI implementation. *Advances in Safety, Reliability and Risk Management*. Proc. European safety and reliability conference. PSAM 2011/ESREL 2012. Helsinki. pp. 1–6.
7. Migov, D.A. (2017) [Parallel methods for network reliability exact calculation and evaluation]. *Molodezh'. Nauka. Tekhnologii (MNTK-2017)* [Youth. Science. Technologies]. Proc. of the International Conference. Novosibirsk. pp. 67–69. (In Russian).
8. Colbourn, Ch.J. (1987) *The combinatorics of network reliability*. New York: Oxford University Press.
9. Won, J.-M. & Karray, F. (2012) Cumulative update of all-terminal reliability for faster feasibility decision. *IEEE Transactions on Reliability*. 59(3). pp. 551–562.
10. Rodionov, A.S., Migov, D.A. & Rodionova, O.K. (2012) Improvements in the efficiency of cumulative updating of all-terminal network reliability. *IEEE Transactions on Reliability*. 61(2). pp. 460–465.



11. Podkorytov, D., Rodionov, A., & Choo, H. (2012) Agent-based simulation system AGNES for networks modeling: review and researching. *Proc. 6th Int. Conf. on Ubiquitous Information Management and Communication*. ACM ICUIMC 2012. ACM New York. USA. Article No 115.
12. Glinisky, B., Rodionov, A., Marchenko, M., Podkorytov, D. & Weins, D. (2012) Scaling the distributed stochastic simulation to exaflop supercomputers. *Proc. 14th IEEE Int. Conf. on High Performance Computing and Communications*. HPCC'2012. IEEE Press. pp. 1131–1136.