Vol.2022-IOT-56 No.36
2022/3/8

Resilience Evaluation by SLA of Line Connectivity Using Discrete Structure Processing System

Hiroki Kashiwazaki[1,a)] Masahiro Mizuta[2] Dai Sato[3]

Abstract: In a distributed system consisting of multiple computers and networks connecting them, failures can occur in each component with a certain probability. In the course of research and development to improve the operational quality of distributed systems, the author focused on the relationship between the probability of failure and the operational quality of distributed systems. And the author also proposed an index to evaluate the resilience of services based on the multiple failure probability of circuits and the failure probability of services running on distributed systems. However, this proposal only considered resilience under limited circumstances with limited fault representation and limited-service continuity requirements. In addition, since the computational complexity increases exponentially with the size and complexity of the target network, only small-scale systems can be used to complete the evaluation in real-time. In this paper, we formulate this problem as a discrete-structure problem and use existing discrete-structure processing systems to evaluate resilience under various combinations of failure representations and different service continuity conditions. At the same time, we aim to expand the concept of distributed systems and promote its application to natural disaster prevention and mitigation.

1. Introduction

Since the advent of computers, the computing power of computers has been enhanced and the services provided by computers have been continuously diversified. As a result, we now have a form in which multiple computers provide multiple services. A system consisting of various computers and a network that supports the transmission of information among them is called a distributed system. Both computers and networks can fail with a certain probability. Many mechanisms have been proposed to continue the services running on the distributed system even if the components of the distributed system fail, such as redundant configurations of equipment and route detour methods for networks.

The conditions that a service requires of the platform that supports it to continue the service (service continuity requirements) depend on the content of the service and vary. For example, some distributed database implementations require a certain number of nodes to be connected to guarantee Byzantine failure tolerance. Some virtualization infrastructure implementations require that the latency between virtualization infrastructures be less than a certain amount of time to move virtual machines to geographically distant virtualization infrastructures without stopping. With the current increase in microservices initiatives, where software is composed of multiple small independent services, the service sustainability requirements of software become more complex. As a result, it can be challenging to determine whether the functionality provided by the platform can satisfy the sustainability requirements of its services.

Site Reliability Engineering (SRE) has been gaining attention since 2016 as an initiative to improve service reliability[1]. SRE is an approach to reduce the time required for recovery from failures by describing the procedures from failure to recovery as reproducible procedures and reducing the percentage of complicated procedures to automate. The advantage of this approach over services that do not explicitly state quantitative quality assurance is that the quality of service continuity provided to users can be quantified and indicated as service level agreements and service level goals.

The representative of this research proposal attempted to create a single indicator of service resilience by simplifying the problem set. Resilience is defined as the tenacity with which a service can be continued even if one or more of its constituent elements fails. When a distributed system, which is the platform supporting a service, is observed at a certain point in time, this distributed system can be represented as a graph G consisting of a set V of computers, a set E of networks connecting the computers, and a set P of failure probabilities for each network. When the failure is limited to a communication breakdown in the network, the sum Q of the probabilities that a failure will occur in one or more networks is calculated, and the sum S of the probabilities that a failure will occur in one or more networks but

[1] National Institute of Informatics
[2] Hokkaido University
[3] Tohoku Medical and Pharmaceutical University
[a)] reo_kashiwazaki@nii.ac.jp

the service running on it will be able to continue is calculated. The relationship of S to Q is an evaluation value that indicates the resilience of this service. If we remove or add some elements from set E, this evaluation value will change, and we can compare how vulnerable or robust the service will be to failures.

This attempt has the following problems at present.

(1) It uses only a simple connected graph judgment as to the service continuity requirement and does not formulate various service continuity requirements.

(2) Since all combinations of failures are to be verified exhaustively, the calculation for evaluation increases in equal proportions as the number of elements constituting the graph increases. The amount of computation increases even more when various service continuity requirements are verified.

(3) The components that make up a distributed system are simplified to only computers and networks and have not been applied to real-world distributed systems where the components are connected by human communication.

The SRE mentioned above efforts have proposed some methods to improve the fault tolerance of various infrastructures and the services that run on them. In a paper written by Bruneau et al. in 2003, in the broad sense of incident response, not only in response to failures but also in disaster recovery and mitigation, it is argued that there is a "steady state of constant quality" before the "occurrence of a failure (disaster)," and that the quality degrades as the failure occurs and recovers over time to a "steady state quality"[2]. It is often quoted as a schematic diagram that the quality degrades as failures occur and then recovers over time to "slowly return" to the quality of the steady state. However, in reality, distributed systems are constantly being requested to perform different calculations by the multiple services running on them, and different traffic demands are constantly being generated. The idea of returning to steady-state quality after a disaster is to "undo the damage". Still, if it can be presumed that quality can be improved by means other than "undoing", then there is room to choose means to improve quality over the previous one.

Even if one element of the distributed system fails, the quality may not degrade if specific service continuity requirements are met. However, if another components fails, the service continuity requirements may not be met, and in this case, quality will certainly deteriorate. By introducing a quantitative measure of resilience, it can be determined that recovering from a failure in the first element when it occurs is quantitatively more beneficial than letting it go. However, if it is possible to augment other components at equal cost and equal time, and if doing so increases the resilience rating, then recovering the failure and augmenting it may be a candidate.

In this way, if we can calculate resilience at each point in time, as well as the next n moves ahead, we can form a more diverse relationship with failures than just "fix the failure" when it occurs. As a method to perform such evaluation calculations, an approach that introduces a probabilistic and discrete mathematical perspective by focusing on the resilience provided by the platform and service continuity requirements uses discrete structure processing systems represented by BDDs and ZDDs[3], [4], is considered. In addition, although distributed systems in a narrow sense are composed of computers and networks, a situation in which humans connect these multiple distributed networks will occur in the event of a real natural disaster. SRE is expected to be applied not only to distributed systems consisting of computers and networks, but also to disaster prevention and mitigation in the event of a natural disaster where humans are forced to connect distributed systems to each other.

2. Related Works

As an approach to evaluate the robustness of networks, methods to solve the satisfiability assessment (SAT) problem and the satisfiability modulo theory (SMT) problem have been proposed in SIGCOMM and NSDI[5]. In addition, methods for reliability analysis of link failures in power and communication networks using ZDDs have also been devised[6]. The purpose of this proposal is to extend these existing methods to evaluate the resilience of various services by mathematical modeling of the service continuity requirements.

In the course of our research on "Research on Wide-Area Distributed Edge Computing Environment with Incentives Based on Operational Quality" (Grant-in-Aid for Young Scientists 19K20256), the principal author of this research proposal has found that when selecting a network redundancy configuration to ensure connectivity to a certain computer with higher probability, the network redundancy can be used as an incentive. We wondered if it would be possible to evaluate the effect of this investment in network redundancy quantitatively. Therefore, we focused on the Service Level Agreement (SLA) and Service Level Objective (SLO, e.g., probability of service availability) of the network provided by the circuit operator and investigated the probability of occurrence of failures that do not "meet" the requirements of the service running on the distributed system by comprehensively examining all possible failures. In this way, the probability of a failure that does not meet the requirements of the service running on the distributed system is investigated.

Google first used the technical term SRE in 2003, and SRE efforts began to attract broader attention in 2016 when O'Reilly published "Site Reliability Engineering". In addition, USENIX has held SREcon[*1], an international conference on SRE, since 2014, and it is now held once a year in North America, Europe, and Asia.

[*1] SREcon — USENIX `https://www.usenix.org/conferences/byname/925`

3. Methodologies

This research aims to establish a quantitative evaluation method for resilience based on a mathematical solution and apply this method to practical disaster prevention and mitigation by considering the relationship between availability provided by distributed systems and service continuity requirements as a constraint satisfaction problem with a discrete structure. This evaluation method is characterized by the fact that it does not evaluate resilience at a single point in time but has a time-series change to the next state, including worsening and improving the situation.

In this study, we set the following three sub-objectives to solve the problems presented in the previous section.

(1) Formulation of various service continuity requirements and solution of the discrete structure problem

(2) Solving and accelerating constraint satisfaction problems using discrete-structure processing systems.

(3) Application to disaster prevention and mitigation in the real world and feedback from the field.

Specific approaches to each of these sub-objectives are described below.

3.1 Formulation of various service sustainability requirements and solution of discrete structure problems

- **Development of Mathematical Models for Service Continuity Requirements**
  We can describe the service availability requirements by characterizing a distributed system as an effective weighted graph. The properties, including bandwidth and communication delay corresponding to the graph's nodes, are used to build a mathematical model of service availability. We aim to clarify the relationship between failures in individual parts and availability concerning the whole system.

- **Logical rules for availability and probability evaluation based on them**
  In the mathematical model described above, the conditions required for a distributed system (availability conditions) are described as logical rules for availability. The graphs of the situations that satisfy the logical rules are clarified. Describe the effect of the failure of a part on the availability of the distributed system. Furthermore, by setting the probability of simultaneous occurrence of multiple partial failures, we can set up a problem to obtain a more realistic service continuity probability of the distributed system.

- **Enumeration of graphs satisfying the availability condition**
  Enumerating the graphs that satisfy the availability conditions makes it possible to examine specific failures. To list such graphs, we specify how to determine whether or not they satisfy the logical rules for availability. Furthermore, we will formulate the failures of the parts. With these preparations, the problem set for graph enumeration becomes clear.

3.2 proposal of diverse and fast resilience evaluation methods and their applications

- **Design and Implementation of Diverse and Fast Resilience Assessment Methods**
  We design and implement a method that generates and evaluates all possible combinations of failures in a brute force fashion to determine whether the various service sustainability requirements of services running on distributed systems are satisfied. We use TdZdd[*2], an implementation of BDD/ZDD, a discrete structure processor, and Graphillion[*3] as references to achieve a fast implementation and clarify how long it takes to complete the evaluation for graphs of arbitrary size.

- **Verification of resilience evaluation in a real environment for widely distributed microservices**
  We will conduct a demonstration experiment to evaluate the resilience of the microservices that compose the wide-area distributed services that run on geographically distributed computers. In this way, we will clarify what kind of services can be evaluated by our proposed method. We will also clarify the requirements for deploying the infrastructure for distributed tracing, one of the monitoring methods for microservices, in a wide-area distributed environment.

- **Proposal and Establishment of a Preemptive Disaster Mitigation Method**
  When a failure occurs, it may be possible to improve resilience while satisfying the constraints by reinforcing different parts of the system instead of repairing the failed part. This can be clarified by assuming a failure in advance and performing a brute force evaluation of reinforcement points and resilience under that condition. This preemptive disaster mitigation method is proposed, and simulation experiments verify its effectiveness.

3.3 Application to disaster prevention and mitigation in the real world and feedback from the field

(1) **Formal description of disaster response systems**
Responding to disasters involves people in various positions, including government, NPOs, disaster volunteers, and residents. For efficient activities, a great deal of information needs to be shared, including the damage

[*2] https://hs-nazuna.github.io/tdzdd-manual/intro.html
[*3] https://github.com/takemaru/graphillion/wiki

ⓒ 2022 Information Processing Society of Japan

situation, support needs, and each activity's current status and plans. On the other hand, in disaster response activities, roles and tasks that did not exist before the disaster arose. It is common for supporters who were not present before the disaster to rush to the scene after the disaster and participate in the activities. For such temporary and cross-sectoral disaster response systems to function, it is essential to prepare for and implement a dynamic system in which disaster relief workers who did not exist before the disaster rush to the disaster site to participate in activities. We will establish a method for describing disaster response activities as a service that operates on a distributed system by identifying and organizing the participants and their roles in the support activities, formulating each functional unit as a computer, and formulating mutual communication and information sharing as a network.

(2) **Resilience assessment of disaster response activities**
We will describe the disaster relief activities conducted for past disasters as a service running on a distributed system. By expressing the evaluation of each disaster relief activity as the resilience of the service, we establish a method for setting parameters to describe each node's functions and network. Since the structure of a disaster response system is highly dynamic, there are no clear criteria for evaluating its strategy and effectiveness. Still, by using this method, we can clarify the system's vulnerability and provide a basis for studying the overall optimization of the disaster response system.

(3) **Study of strategies to improve the resilience of disaster response activities**
Based on the above discussion, we will assess the characteristics of the current disaster response system and consider strategies to improve its resilience. In other words, we will apply various patterns of adding and reinforcing nodes and networks within realistic limitations and simulate their effects to identify effective ways to improve resilience. Formulating a disaster response system as a service that functions on a distributed system, and providing a method to predict and evaluate its functioning, will significantly contribute to the development of disaster response systems.

In this study, we introduce a discrete-structure problem-solving approach to quantitative resilience evaluation and clarify the range of service continuity requirements that can be met. This will enable us to quantitatively calculate the cost-effectiveness of recovery, augmentation, and decommissioning of elemental failures in distributed systems. In addition, by clarifying problems that are difficult to formulate as discrete-structure problems (or, even if they can be formulated, it is difficult to make a fast judgment of the requirements using a processing system implementation), we can clarify areas where quantitative evaluation of resilience is possible and areas where it is difficult. In addition, by applying the proposed method to practical disaster prevention and mitigation and obtaining feedback, we will clarify the contributions and problems of SRE not only for distributed systems, but also for distributed systems in a broad sense that includes humans as a component.

Since 2011, the author has been designing, constructing, and operating a wide-area distributed platform "Distcloud" in which 13 sites in Japan participate. Using this platform, it is possible to conduct demonstration experiments in real environments. In addition, the platform has been adopted by the Joint Usage and Research Center for Interdisciplinary Large-scale Information Network (JHPCN) for research projects, and we are ready to expand the verification environment on the distributed system quickly[*4].

3.4 Probabilistic Quantitative Evaluation

The total number of failure patterns depends on the topology that constitutes the wide-area distributed system. It takes a lot of time to perform benchmarks on all the failure patterns and perform quantitative evaluations. Unless all failure patterns should be evaluated quantitatively, it is hard to obtain the result of a quantitative evaluation. Meanwhile, various designs are implemented for wide-area distributed services in order to improve fault tolerance, and these designs require some constraints for their proper operation. So we have proposed a pruning method to reduce the total number of failure scenarios [7], [8].

For example, in a file system, there is a redundant design in which when a chunk is written to a node, a duplicate copy of this chunk is written to other n nodes to increase fault tolerance. In this design, there must be n or more other nodes connectable from a certain node. If the possibility of connection is lost due to the occurrence of failures and the number of other nodes that can be connected from a certain node falls below n, this writing process will fail. There are other measures against split-brain syndrome. In this case, when the total number of nodes is n, when write requests of chunks occur in a certain node, the requests will succeed only when the total number of nodes included in the cluster including the node is larger than $\frac{n}{2}$. Similarly, the requests will fail in a cluster where the total number of nodes is less than $\frac{n}{2}$.

There are no systems that can run under all the situation on the earth. The targeted system has its constraints for its expected environment. The behavior of the system under an arbitrary failure pattern can be classified into the following three by using the constraint.

(1) Requests from all nodes are defined.

(2) Requests from some (or all) nodes are not defined (therefore may return errors).

By matching the constraint conditions under which the

[*4] Joint Usage and Research Center for Interdisciplinary Large-scale Information Infrastructures https://jhpcn-kyoten.itc.u-tokyo.ac.jp/en/

wide-area distributed service operates and the given failure pattern, it is possible to know in advance which class the benchmark request belongs to before performing the benchmark. In the case of 1, the result obtained by the benchmark request may show a quantitative evaluation value of the wide-area distributed service in the failure pattern. In cases 2, the method of handling evaluation values for undefined results must be defined. That is, there can be a method of setting the evaluation value at the time of undefined operation to 0, or a method of excluding the evaluation value from the quantitative evaluation because the evaluation value is undefined because it is an undefined operation. By this exclusion, the time required for benchmarking for quantitative evaluation can be shortened.

Meanwhile, it is possible to quantitatively calculate the fault tolerance under the constraints of the topology and the design of the target system according to the number of failure patterns that can be expected as defined operations and not defined operations. When the identifier of each site is i, the nodes in the topology can be represented as n_i. N, the set of all nodes, can be expressed as follows.

$$N = \{n_1, n_2, ..., n_\nu\} \tag{1}$$

ν means a total number of nodes. In the same way, the identifier of each interconnection is j (the number of i and j are not related), the edges in the topology can be represented as e_j. E, the set of all nodes, can be expressed as follows.

$$E = \{e_1, e_2, ..., e_\epsilon\} \tag{2}$$

ϵ means a total number of edges. A network failure f_k can be expressed as a subset of E (k is the identifier of each failure). The failures include simultaneous multiple failures. The set of all failures F can be expressed as the summation of single failures (expressed as a set F_1), double failures (F_2), and all ϵ-fold failures (F_ϵ). A number of F_n can be calculated as combinations of ϵ things, taken n at a time. Thus, $num(F)$ that is a total number of F can be expressed as follows .

$$\begin{aligned} num(F) &= \sum_{i=1}^{\epsilon} num(F_i) \\ &= {}_\epsilon C_1 + {}_\epsilon C_2 +, ..., {}_\epsilon C_\epsilon \end{aligned}$$

A set of all nodes N and a set of all edges E are defined. Then a probability of failure on the edge e_i is defined as p_{e_i}. A set of a probability on all edges P_E can be expressed as follows.

$$P_E = \{p_{e_1}, p_{e_2}, ..., p_{e_\epsilon}\} \tag{3}$$

f is also defined as a subset of E and it can express a failure pattern. F is a set of all failure patterns. An arbitrary f_k can be expressed as follows.

$$f_k = \{e_i, e_j, ..., e_\zeta\} \tag{4}$$

Then, $p(f_k)$, the probability of the failure pattern f_k can be calculated as a product of the probability of f_k multiplied by a product of the "non-failure" probability $(1 - p_{e_i})$ of remaining of f_k. So $p(f_k)$ can be expressed as follows.

$$\begin{aligned} p(f_k) &= \prod P_{f_k} \prod (1 - P_{E - f_k}) \\ &= \prod_{e \in f_k} p_e \prod_{e \in E - f_k} (1 - p_e) \end{aligned}$$

The set F can be separated to a set D that the system can run under a defined condition and U that the system can not run under the condition. P_D and P_U are defined as a summation of the probabilities of each failure pattern in D and U. P_D and P_U can be expressed as follows.

$$P_D = \sum_{f_k \in D} p(f_k) \tag{5}$$

$$P_U = \sum_{f_k \in U} p(f_k) \tag{6}$$

According these equations shown above, the value of resilience on the targeted system R can be expressed as follows.

$$R = log \frac{P_U}{P_D + P_U} \tag{7}$$

4. Evaluations

4.1 Effectiveness of qualitative pruning

Table 1 shows a classification of the number of node groups that can reach each other by an arbitrary edge (cluster) and the number of multiple failure of failure patterns in the five node, full-mesh topology .

All failure patterns in the five-node, full-mesh topology shown in Fig. 5 are classified by the multiplicity of failures and the number of nodes (clusters) that can reach each other by any edge. Is shown in Table 7. In this topology, the maximum multiplicity of failures is 10. All failure patterns with a multiplicity of failures of 3 or less have a cluster number of 1 and all nodes can reach each other. A failure pattern with a multiplicity of failures of 4 or more and a cluster count greater than 1 appears. When the multiplicity of failures is 7 or more, there is no failure pattern with 1 cluster.

As of Cloudian Hyperstore , the result of an object creation request is undefined unless it is possible to connect to three or more locations. Therefore, a failure pattern with two or more clusters is undefined, and quantitative evaluation is performed with a failure pattern with one cluster. Since the total number of failure patterns with 1 cluster is 727 and the total number of failure patterns is 1023, 29% of benchmarks can be omitted compared to benchmarking all failure patterns.

Calculate the expected value of the predefined motion probability, weighted by the failure probability described in Section 3.4. Here, it is assumed that the probability that a failure occurs at any edge is uniformly p. At this time, the total W_d weighted by P_G for the number of failure patterns that can be expected to be defined is $W_d = 10p + 45p^2 + ..., +225p^5 + 125p^6$. On the other hand, the total W_u weighted by P_G for the number of failure patterns resulting in undefined behavior is $W_u = 5p^4 + 30p^5 + ..., +10p^9 + p^1 0$. The expected value R of the defined motion probability weighted by the failure probability is, for example, $R = 5.07 \times 10^{-7}\%$ when $p = 0.01$, $R = 5.07 \times 10^{-10}\%$ when $p = 0.001$ prospectively.

multiplicity of failures	number of failure patterns	number of clusters				
		1	2	3	4	5
1	10	10	0	0	0	0
2	45	45	0	0	0	0
3	120	120	0	0	0	0
4	210	205	5	0	0	0
5	252	222	30	0	0	0
6	210	125	85	0	0	0
7	120	0	110	10	0	0
8	45	0	0	45	0	0
9	10	0	0	0	10	0
10	1	0	0	0	0	1

Table 1 Classification of failure patterns with number of clusters

5. Conclusion

In this paper, the authors really wanted to implement a rudimentary evaluation method using TdZdd and Graphillion, evaluate the method, and introduce the effect of parallelization. However, due to my inherent deadline driven laziness, I have not been able to make much progress in writing the program, and I hope to be able to demonstrate something before the March meeting.

References

[1] Beyer, B., Jones, C., Petoff, J. and Murphy, N. R.: *Site Reliability Engineering: How Google Runs Production Systems*, O'Reilly Media, Inc., 1st edition (2016).

[2] Bruneau, M., Chang, S. E., Eguchi, R. T., Lee, G. C., O'Rourke, T. D., Reinhorn, A. M., Shinozuka, M., Tierney, K., Wallace, W. A. and von Winterfeldt, D.: A Framework to Quantitatively Assess and Enhance the Seismic Resilience of Communities, *Earthquake Spectra*, Vol. 19, No. 4, pp. 733–752 (online), DOI: 10.1193/1.1623497 (2003).

[3] Bryant: Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Transactions on Computers*, Vol. C-35, No. 8, pp. 677–691 (online), DOI: 10.1109/TC.1986.1676819 (1986).

[4] Minato, S.-i.: Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, *Proceedings of the 30th International Design Automation Conference*, DAC '93, New York, NY, USA, Association for Computing Machinery, p. 272 – 277 (online), DOI: 10.1145/157485.164890 (1993).

[5] Beckett, R., Gupta, A., Mahajan, R. and Walker, D.: A General Approach to Network Configuration Verification, *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, New York, NY, USA, Association for Computing Machinery, p. 155 – 168 (online), DOI: 10.1145/3098822.3098834 (2017).

[6] Inoue, T.: Reliability Analysis for Disjoint Paths, *IEEE Transactions on Reliability*, Vol. 68, No. 3, pp. 985–998 (online), DOI: 10.1109/TR.2018.2877775 (2019).

[7] Kashiwazaki, H., Takakura, H. and Shimojo, S.: Resilience Evaluations of a Wide-area Distributed System with a SDN-FIT system, *2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pp. 1–8 (2019).

[8] Kashiwazaki, H., Takakura, H. and Shimojo, S.: An Evaluation of Stochastic Quantitative Resilience Index Based on SLAs of Communication Lines, *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1449–1454 (online), DOI: 10.1109/COMPSAC51774.2021.00215 (2021).

ⓒ 2022 Information Processing Society of Japan