2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S) | 978-1-6654-0260-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/DSN-S54099.2022.00024

Modeling Composition of Cloud Services with Complex Dependencies for Availability Assessment

Xingjian Zhang
Dept. of Computer Science and Technology
Tsinghua University
Beijing, China
zhangxj18@mails.tsinghua.edu.cn

Long Wang
Lab for Reliability and Security of Networks and Systems
Institute for Network Sciences and Cyberspace
Tsinghua University
Beijing, China
longwang@tsinghua.edu.cn

***Abstract*—This paper presents a methodology to assess availability of cloud services through Bayesian network-based availability model. We propose a modeling technology to represent heterogeneous dependencies of cloud services in the methodology. We show that our modeling technology provides more expressive capability for complex dependencies than classic tools like RBD.**

Keywords—cloud service, availability assessment, Bayesian network, modeling, dependency

I. Introduction

Nowadays a large number of services are hosted in cloud systems. Cloud services providers (CSPs) provide high service availability which are documented in Service Level Agreements (SLAs) as the agreements on service qualities with service users [1]. Assessing cloud services' availability accurately is important for CSPs to evaluate cloud systems and improve services' availability.

Cloud services are composed of many software and hardware components such as software instances, containers, virtual machines (VMs), physical machines (PMs), switches, routers, etc. There are complex dependencies among the components of a cloud service with respect to the service availability. One kind of dependencies are those implied in the service's request processing (called *request processing dependencies*); e.g., for a service with two web servers and two application servers, a failure of both application servers imposes more severe impact on the service's availability than a failure of one web server and one application server. Another kind of dependencies are those between a component and its host (called *host dependencies*); e.g., when the PM hosting a web server fails the server also fails. Other kinds of dependencies exist in component sharing among services and the services' invocations of each other.

To assess the availability of cloud services the availability model of the services should accommodate all these dependencies. A classic instrument for availability modeling is reliability block diagram (RBD) [2]. However, RBD only deals with the composition of components in series, in parallel, and in a combination of series and parallel that depict the service as paths from a start point to a sink point (like an electric circuit). RBD is unable to accommodate aforementioned heterogenous dependencies. State-space models, such as Markov models and Petri Net models, capture the states of the entire services, and are subject to the state space explosion issue or render a huge number of simulations. Moreover, it is difficult to model specific dependencies involved in each individual component of a large cloud. Use of the same Markov model for all components of the same type with different parameters is insufficient for capturing the specific dependencies because they are represented as links between Markov nodes. We select Bayesian network as a suitable modeling tool for assessing availability of services in clouds. Currently Bayesian network is used mostly for knowledge inference and reliability assessment for mechanical systems or industrial control systems [3], however, the prior work does not, and is unable to, handle multiple kinds of dependencies such as those in the cloud domain. The use of Bayesian network for modeling request processing and other kinds of dependencies of IT services in large clouds and assessing their availability, is our innovation. Actually, our approach can be generally applied to those systems with complex heterogenous dependencies other than clouds (for systems without complex dependencies RBD may be sufficient).

The problem addressed in this paper is *to assess the availability of cloud services that involve heterogeneous dependencies among service components, in particular a mix of request processing dependencies, host dependencies and other kinds of dependencies put above*. We propose an availability-modeling technology based on Bayesian network and a methodology of constructing the model automatically for assessing the availability of cloud services.

II. Motivating Example

A simple example service we use to demonstrate our availability model is given in Fig. 1. The example service consists of two tiers, 2 web server instances as the first tier and 2 application server instances as the second tier. A request is first processed by a web server and then processed by an application server as depicted in Fig. 1(a). Fig. 1(b) shows the host dependencies of the four software components. Here we use asymmetric request processing only to show the generality of our model in handling arbitrary request processing dependencies (in real cloud systems such asymmetry exists for various reasons like performance and security concerns).

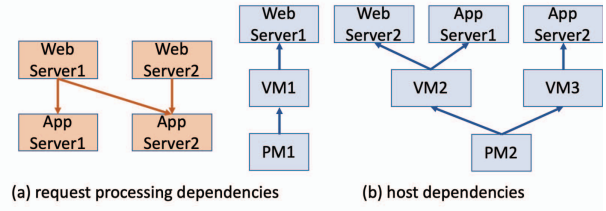

Fig. 1. The dependencies of the example service

We tried using RBD to model the example's dependencies. Fig. 2 shows two failed attempts: the left one is not a valid RBD and solving the right one does not bring about correct results. RBD models the service availability as paths from a start point to a sink point, and hence, it fails to handle the mix of the two kinds of dependencies. An alternative way is to solve the host dependencies obtain the availability values of software instances, and feed them into the RBD created from the request processing dependencies only. But RBD is still unable to deal with the cases when component-sharing and service-invocation dependencies are brought in.

We propose to design a technology that captures all kinds of dependencies of cloud services in one model. For this purpose, we employ Bayesian network which provides the capability to generally model dependencies (or relationships) among components. Fig. 3 illustrates the Bayesian network-based availability model for the example service. Compared with the service dependencies in Fig. 1, this availability model is constructed with the following operations:

978-1-6654-0260-6/22/$31.00 ©2022 IEEE
DOI 10.1109/DSN-S54099.2022.00024

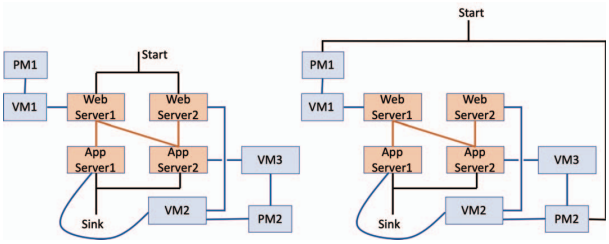

Fig. 2. Two failed attempts of modeling the example service using RBD

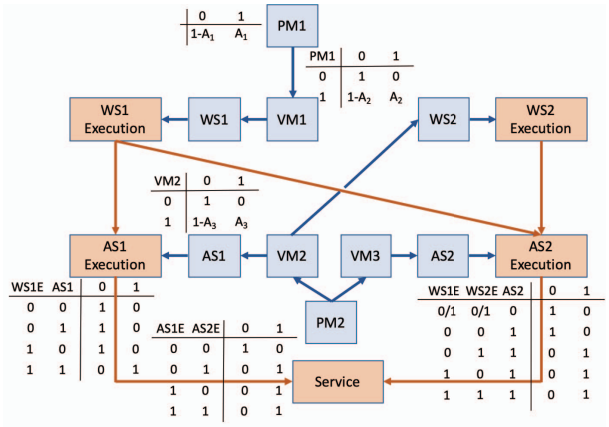

Fig. 3. The availability model of the example service in Bayesian network

(1) Build the model for the request processing dependencies. Because the four software instances have both their own availability values and their involvements in the request processing dependencies, we split each of them into two nodes to avoid the complications of crafting the conditional probability distribution (CPD) of Bayesian network nodes for accommodating both pieces of information. For example, the "AS1 Execution" node represents the request processing dependency involving AppServer1, "AS1" represents AppServer1's own availability value (A_3 in Fig. 3), and "AS1 Execution" has a dependency on "AS1". The connections among the four extra "Execution" nodes in Fig. 3 are the direct mapping of the request processing dependencies in Fig. 1(a).

(2) Add a "Service" node to join all request processing paths. Then the service availability is the availability of the "Service" node (the probability of "Service"=1) with no a priori condition.

(3) Apply the host dependencies of Fig. 1(b) onto the model.

(4) Construct the CPD for each node of the model. For those nodes that do not depend on other nodes ("PM1" and "PM2" in Fig. 3), the CPD is derived from its availability value directly. The CPD near "PM1" in Fig. 3 shows its probability of bearing the value 1 ("PM1"=1) is its availability value A_1, assuming its host is available. For those nodes that depend on some other nodes, we construct their CPDs according to the semantics of their involved dependencies. For example, when "VM2"=0, AppServer1 is also unavailable, so the probability of "AS1"=1 is 0; when at least one of the web servers executes successfully and AppServer2 is available (("WS1E"=1 or "WS2E"=1") and "AS2"=1), AppServer2 executes successfully (the probability of "AS2 Execution"=1 is 1). For charting neatness Fig. 3 illustrates 6 nodes' CPDs and the other nodes' CPDs are similarly created.

III. Methodology

Fig. 4 depicts the methodology of assessing cloud service availability by means of modeling their composition. Here we assume the data of software/hardware components' own availability values and different kinds of dependencies are given. Actually, the work presented here is part of a bigger project of assessing cloud service availability, which includes collection of such data, e.g. via technologies like REPTrace [4][5]. Data collection (the ovals in the figure) is beyond this paper's scope; the two rectangular blocks are this paper's contributions.

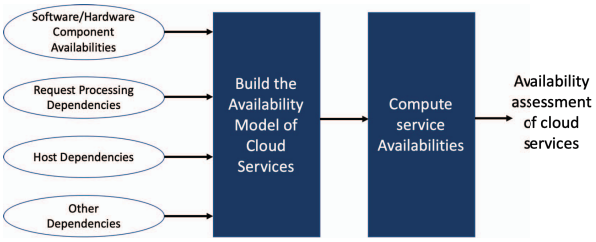

Fig. 4. Availability assessment of cloud services by means of modeling

The availability model of cloud services is built from these data. As described in Section II, we first split software components into "Execution" nodes and their availability nodes, and map the services' request processing dependencies into the "Execution" nodes. For each cloud service we add a "Service" node which joins all request processing paths of the service. Then we directly link the involved components' host dependencies to their availability nodes in the model. Followingly, the CPDs of the model nodes are constructed according to the semantics of involved dependencies. An algorithm is developed to automate the model building from given data. If there are other kinds of dependencies like component-sharing ones and service-invocation ones to be modeled, these dependencies are then added into the model by directly linking corresponding nodes and then constructing the CPDs according to the dependency semantics.

Finally, service availability values, i.e. the probabilities of the "Service" nodes bearing the value 1, are computed from the model. The exact inference and approximate inference solutions of Bayesian network in tools such as pgmpy [6] can do the task.

This work is conducted in a real-world cloud system. The availability assessment of cloud services will be used to further identify availability bottlenecks and improve service availability.

IV. Conclusion

This paper proposes a methodology to assess availability of cloud services involving heterogeneous dependencies, which exploits Bayesian network as the tool, more expressive than classic tools like RBD, to model service composition, and then computes service availability from the model. The methodology is being applied onto a real cloud for improving availability.

References

[1] M.R. Mesbahi, A.M. Rahmani, M. Hosseinzadeh. Reliability and high availability in cloud computing environments: a reference roadmap. *Human-centric Computing and Information Sciences*, vol. 8, 2018.

[2] E. Bauer, R. Adams. Reliability and availability of cloud computing. *John Wiley & Sons*, 2012.

[3] B. Cai, X. Kong, Y. Liu, et al. J. Lin, X. Yuan, H. Xu, R. Ji. Application of bayesian networks in reliability evaluation. *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, 2019.

[4] Y. Yang, L. Wang, J. Gu, Y. Li. Capturing Request Execution Path for Understanding Service Behavior and Detecting Anomalies without Code Instrumentation. *IEEE Transactions on Service Computing (TSC),* 2022.

[5] Y. Yang, L. Wang, J. Gu, Y. Li. Transparently Capturing Execution Path of Service/Job Request Processing. *International Conference on Service-Oriented Computing (ICSOC)*, 2018.

[6] A. Ankan, A. Panda. pgmpy: Probabilistic graphical models using python. *The 14th Python in Science Conference (SCIPY)*, Citeseer, 2015.