GSPN-Based Reliability-Aware Performance Evaluation of IoT Services

Songyuan Li[†], Jiwei Huang*[‡]
[†]*School of Computer Science*
Beijing University of Posts and Telecommunications, Beijing 100876, China
[‡]*State Key Laboratory of Networking and Switching Technology*
Beijing University of Posts and Telecommunications, Beijing 100876, China
Email: sylee1416@gmail.com, huangjw@bupt.edu.cn

***Abstract*—With the growing popularity of Internet of Things (IoT) services being applied in several aspects of real-life applications, performance has become an important requirement. Meanwhile, the techniques for reliability enhancement such as virtual machine migration and recovery also have significant impact on end-to-end performance. This paper proposes a predictive approach of reliability-aware performance evaluation for recoverable IoT services using the modeling techniques of generalized stochastic Petri net (GSPN). Mathematical models formulating the dynamics of both server clusters and IoT systems are presented, and quantitative analyses of performance metrics are provided. Empirical experiments based on real-world data obtained from IoT services and cloud systems are conducted, and parameter settings as well as experimental results are discussed in detail.**

***Keywords*-Internet of Things (IoT); IoT service; performance evaluation; reliability; generalized stochastic Petri net.**

I. Introduction

The Internet of Things (IoT) is an emerging technique which is expected to offer advanced connectivity of devices, systems and services that goes beyond machine-to-machine (M2M) communications and covers a variety of protocols, domains, and applications [1]. It can be seen as smart environments composed of pervasively distributed things (e.g., devices, sensors, actuators, smartphones, and appliances) offering heterogeneous capabilities abstracted as services in a well-defined architecture for higher flexibility facing business dynamics [2]. With the rapid development of microchips, sensor devices, networks and software in recent years, IoT is becoming more and more popular in several aspects, such as industrial manufacturing, smart home, transportation, etc.

With the growing popularity of IoT services being applied in real-life applications, the performance of IoT services has become an important requirement. Since the requirements of real-time adaptive sensing as well as data analytics arise significantly these days, a novel computing paradigm namely edge computing has been proposed for IoT service implementation, which pushes the frontier of services away from centralized nodes to the logical extremes of a network, enabling immediate data analytics and knowledge generation to occur at the source of the data [3]. By leveraging the techniques of both cloud computing and mobile computing, such paradigm has become very popular for providing services in various applications [4].

In order to guarantee the end-to-end quality of service (QoS) of the IoT services, the very foundation is to provide a precise evaluation of the performance of IoT services/systems, which is the basic criterion of Service-Level Agreement (SLA) and the reference for QoS optimization. To this end, there have been several research works dedicating to performance evaluation of IoT services, from both measurement [5] and modeling aspects [6]. Their basic ideas are to measure the performance metrics directly in real-life systems, or to build mathematical models based on the system design and predict the performance with quantitative analyses, respectively. Since the model-based evaluation can be conducted at the design phase before system implementation and thus much cheaper than measurement-based approaches, it appears significant advantages for performance evaluation especially in large-scale services computing systems.

Meanwhile, in order to improve the reliability and fault-tolerance, services are usually deployed in virtual machines (VMs) or virtualized cloud servers, which are equipped with VM migration and recovery techniques allowing the fast restart of a virtual machine without severe interrupt when errors or failures occur. However, introducing VM migration brings several challenges to performance evaluation, for the error probing, fault repairing and system restarting take time, affecting the overall end-to-end QoS of the services. Although there has been some existing research as well as our existing work that studied the performance evaluation considering the reliability issue [7]–[9], few of them presented any methodology of performance modeling for IoT systems especially in edge computing paradigm.

To fill this gap, this paper proposes a theoretical approach of reliability-aware performance evaluation of IoT services. Generalized stochastic Petri net (GSPN) is applied to formulate the dynamics of the IoT systems, including request arrivals, task scheduling, queueing, failures, repairs and recoveries. Both atomic services and comprehensive systems are modeled, and corresponding quantitative analyses are presented. Empirical experiments based on real-world data are conducted, and system parameters as well as experimental results are analyzed in detail. This paper is expected to provide a predictive methodology of performance evaluation of IoT services, and offer a useful reference for design and optimization of IoT service systems.

*Jiwei Huang is the corresponding author. Email: huangjw@bupt.edu.cn.

2474-2473/17 $31.00 © 2017 IEEE
DOI 10.1109/SCC.2017.70

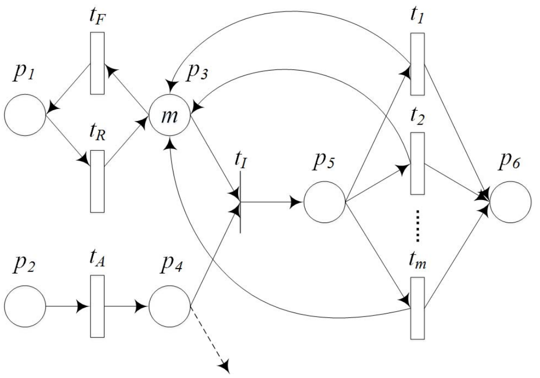

Figure 1. Cluster model.

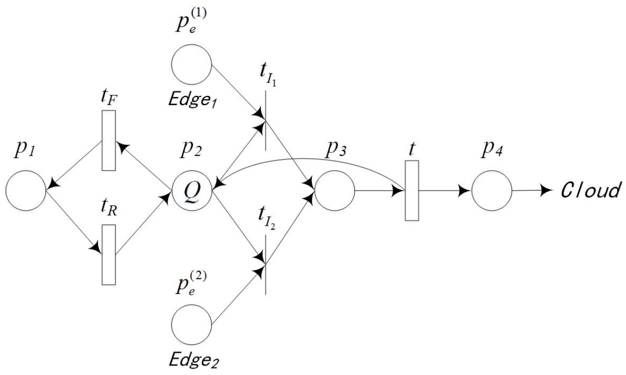

Figure 2. Communication model.

II. GSPN Model for Performance Evaluation

A. Cluster Model

We take advantage of generalized stochastic Petri net (GSPN) to formulate the dynamics of an IoT system. GSPN generalizes traditional Petri Net model with a stochastic timed automata, and appears advanced ability of performance evaluation for complex processes and systems. The basic formal definition of a GSPN is shown as follows.

Definition 1 (GSPN): A generalized stochastic Petri net (GSPN) Ψ for formulating the dynamics of an IoT system is defined by a 7-tuple $\Psi = (P, T, F, W, \Pi, M_0, \lambda)$ where,

- P is the finite set of places.
- $T = T_t \cup T_i$ is the finite set of transitions, where T_t is the set of timed transitions while T_i is the set of immediate transitions.
- $F \subseteq (S \times T) \cup (T \times S)$ is the finite set of arcs from places to transitions and from transitions to places.
- $W : F \to \mathbb{N}^+$ is the weight function of arcs.
- $\Pi : T \to [0, 1]$ is the priority function of the transitions, expressed by the transition probabilities.
- M_0 is the initial state (marking) of the GSPN.
- $\lambda : T_t \to \mathbb{R}^+$ is the set of firing rate of the timed transitions.

Dynamics of IoT services and systems can be formulated by a GPSN, where the transitions represent basic activities while tokens assigned to the places demonstrate the task requests residing in the system. The arcs link places to transitions or transitions to places, which bridges the gap between services and their associated task requests. Meanwhile, the weight of arcs indicates the maximum volume of items which can be served at each dispatch time.

For service activities of IoT systems in reality, task requests are usually processed in a server cluster. In order to characterize the dynamics of a server cluster, a GSPN based model is built up, shown by Fig. 1.

The loop $p_3 \to t_F \to p_1 \to t_R \to p_3$ means the procedure of machine failure and recovery. Without loss of generality, we assume that there are at most m machines being performed, and their service procedures are formulated by $t_1, t_2, \ldots, t_m$. The average failure rate for each machine is λ_F, while the repair rate is λ_R. The timed transition t_A indicates task arrivals at the cluster, with the arrival rate of λ_A. The tokens residing at the place p_3 denote the idle machines without failure ready to accept requests and perform, while tokens in p_5 indicate that the servers are busy processing the accepted requests. The outward transition at p_4 indicates the requests being delivered to the cloud layer, whose transition probability is denoted by $1 - \pi_I$. For the servers at the cloud layer, π_I is set to be 1.

With Little's Law, the average response time of tasks processed by the cluster can be calculated by

$$RS = \frac{q_4 \cdot \pi_I + q_5}{\lambda_A \cdot \pi_I}, \tag{1}$$

where q_4 and q_5 are the queue length at p_4 and p_5 (i.e. the average amount of tokens), respectively.

B. Communication Model

With edge computing paradigm, a task is initialized at the edge site, and can be processed by either an edge server or a cloud server. If a request is scheduled to the cloud servers, it has to be transmitted from the edge server to the cloud site via Wide Area Network (WAN). Such communication procedures are formulated by a GSPN shown by Fig. 2.

Considering that network bandwidth is limited, we assume that there are at most Q instances buffered at the gateway. Since network failures may cause the instability affecting the communication performance, we model the failure and repair processes by the timed transitions t_F and t_R, respectively. The failure rate is defined by λ_F, and the repair rate is expressed by λ_R. In Fig. 2, transition t characterizes the data transmission between the edge layer and the cloud layer, while the place p_4 indicates task arrivals at the cloud servers. Since the servers in a cluster are commonly connected via high speed Local Area Network (LAN), their transmission delay is omitted in our analysis.

Therefore, the average time consumption during the period of the communication can be obtained by

$$TR = \frac{q_e^{(i)} \cdot \left(1 - \pi_I^{(i)}\right) + q_3}{\lambda_A^{(i)} \cdot \left(1 - \pi_I^{(i)}\right)}, \tag{2}$$

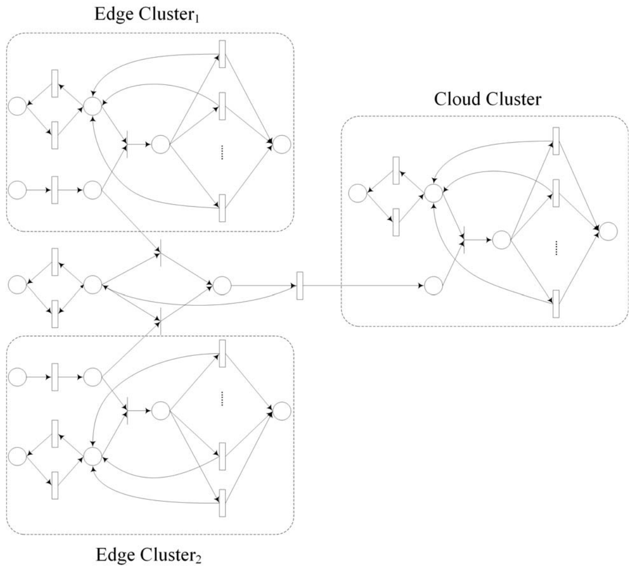

Figure 3. System model.

where i refers to the index of the edge cluster, and q denotes the average amount of tokens at the relevant place.

C. System Model

With the cluster model and communication model established above as basic building blocks, we put forward a comprehensive GSPN model formulating an IoT system with edge computing paradigm.

As demonstrated by Fig. 3, the system consists of two layers, including the edge layer and the cloud layer. Task requests enter into the system from the edge layer. Some of them are served and terminated locally at the edge cluster while others are processed at the cloud cluster. The proportion of tasks being served at the cloud is determined by the probability of bypass transmission.

Suppose there is an edge cluster with the probability of bypass transmission $(1-\pi)$, and the average time consumption TR for bypass transmission can be calculated by (2). Moreover, the average response time $RS^{(e)}$ of tasks served at the edge layer and the servicing time $ST^{(c)}$ of tasks at the cloud layer can be both calculated by (1). Thus, the average response time $RS^{(c)}$ of tasks served at the cloud layer can be formulated as follows.

$$RS^{(c)} = ST^{(c)} + TR. \tag{3}$$

Hence, the total average response time for the IoT system is

$$RS^{(sys)} = \pi \cdot RS^{(e)} + (1-\pi) \cdot RS^{(c)}. \tag{4}$$

III. Evaluation

A. Experimental Setup

We conduct real data based experiments simulating an IoT system with our models and analyze the experimental results. The empirical results are obtained by applying a GPSN simulation software called PIPE [10], [11].

A dataset named "T-Drive" [12], [13] released by Microsoft Research is adopted in our simulations. Such dataset includes the GPS trajectories of 10,357 taxis within the city of Beijing during a period of one week in 2008, among which the timestamps indicating the data submission time are applied as the task arrivals at the IoT system.

The failure rate and repair times are obtained from an open source dataset provided by Los Alamos National Laboratory, which recorded 23,739 failure situations of 23 High Performance Computing (HPC) systems [14]. We make use of the log information, and analyze different types of failures in our GSPN model. The data on software and hardware failures is applied to our cluster models, while the failure rates and repair rates of network failures are applied as the parameters of the communication model.

We simulate an IoT system with multiple edge clusters and one cloud cluster, each of which consists of 10 parallel servers sharing one buffer queue of user requests. The unknown parameters that cannot be obtained from the original datasets are manually set and tuned with various values whose impact to the results will be shown in the next subsection, and the timed transitions in our simulations are assumed to be exponentially distributed.

B. Results and Discussions

We firstly tune the mean time to failure (MTTF) and mean time to repair (MTTR) of machines in the cluster with the same task arrivals to see how they affect the performance of the system. As shown in Fig. 4, the utilization of cluster trends down along with the growth of MTTF while the reduction of MTTR leads to the decreasing of utilization. The decline of utilization means higher redundancy of servers, implying the promotion of system reliability. Also, with the same task arrivals, the decrease of utilization leads to the improvement on performance. Therefore, reducing the failure rate (i.e., increasing MTTF) and fastening repair processes (i.e., decreasing MTTR) can help to improve not only reliability but also performance of the systems.

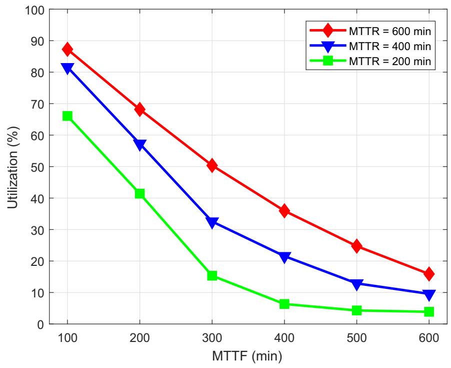

Figure 4. Empirical Results for Cluster Utilization.

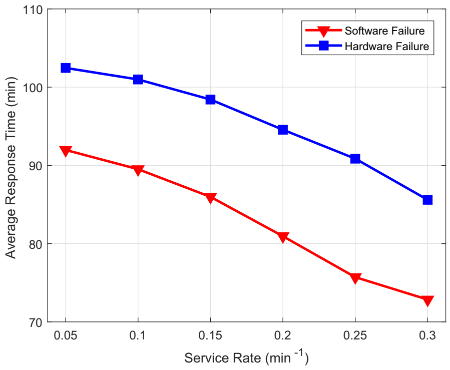

Figure 5. Empirical Results for Average Response Time.

We use the average response time as the performance indicator. Fig. 5 demonstrates the average response time with the variations of service rate. It can be concluded that the average response time decreases with the increase of service rate. Meanwhile, we apply the failure and repair rates of software and hardware failures respectively to evaluate the performance of system. It can be obtained that software failures can be timely repaired while hardware failures commonly requires more periods of time to repair. Thus, it can be seen from the empirical results that the system with software failures has less response time than the one with hardware failures. From this, we can learn that hardware failures are more difficult to be handled than software failures, and sometimes they can be regarded as catastrophic to system reliability as well as performance.

IV. Conclusion

In this paper, we propose a modeling approach of performance evaluation for IoT services in edge computing paradigm. GSPN models are put forward to formulate the dynamics of the services and systems, where reliability issues are fully considered. Based on the models, quantitative analyses are carried out to obtain the performance metrics. Finally, simulation experiments based on real-world data obtained from IoT and cloud systems are conducted, and the experimental results which show the relationship between the performance and system parameters are discussed.

This paper is a primitive work expected to offer an efficient approach for predicting the performance of IoT services before their implementation. The models can be further specified according to different types of failures in real-life IoT systems, and the statistical distributions of the failures, repairs, arrivals and service processes can be mathematically described more precisely. Moreover, the experiments in reality can further validate our approach and provide a better understanding of the model description and parameter settings.

Acknowledgment

This work is supported by Beijing Natural Science Foundation (No. 4162042), National Natural Science Foundation of China (No. 61502043), and BeiJing Talents Fund (No. 2015000020124G082).

References

[1] J. Holler, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press, 2014.

[2] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing*. Springer Berlin Heidelberg, 2007.

[3] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, “Edge-centric computing: Vision and challenges,” *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 37–42, Sep. 2015.

[4] P. Corcoran and S. K. Datta, “Mobile-edge computing and the Internet of Things for consumers: Extending cloud computing and services to the edge of the network,” *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 73–74, Oct. 2016.

[5] M. Stusek, J. Hosek, D. Kovac, P. Masek, P. Cika, J. Masek, and F. Kröpfl, “Performance analysis of the OSGi-based iot frameworks on restricted devices as enablers for connected-home,” in *the 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT 2015)*, Oct. 2015, pp. 178–183.

[6] R. Zhang, M. Wang, X. Shen, and L. L. Xie, “Probabilistic analysis on QoS provisioning for Internet of Things in LTE-A heterogeneous networks with partial spectrum usage,” *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 354–365, Jun. 2016.

[7] J. Zhao, Y. Xiang, T. Lan, H. H. Huang, and S. Subramaniam, “Elastic reliability optimization through peer-to-peer checkpointing in cloud computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 491–502, May 2017.

[8] Y. Xia, M. Zhou, X. Luo, S. Pang, and Q. Zhu, “Stochastic modeling and performance analysis of migration-enabled and error-prone clouds,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 495–504, Apr. 2015.

[9] J. Huang, C. Lin, and J. Wan, “Modeling, analysis and optimization of dependability-aware energy efficiency in services computing systems,” in *the 2013 IEEE International Conference on Services Computing (SCC 2013)*, Jun. 2013, pp. 683–690.

[10] N. J. Dingle, W. J. Knottenbelt, and T. Suto, “Pipe2: A tool for the performance evaluation of generalised stochastic petri nets,” *SIGMETRICS Performance Evaluation Review*, vol. 36, no. 4, pp. 34–39, Mar. 2009.

[11] P. Bonet, C. Llado, R. Puijaner, and W. Knottenbelt, “Pipe v2.5: A petri net tool for performance modelling,” in *the 23rd Latin American Conference on Informatics (CLEI 2007)*, Oct. 2007.

[12] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, “T-drive: Driving directions based on taxi trajectories,” in *the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS 2010)*, Nov. 2010, pp. 99–108.

[13] J. Yuan, Y. Zheng, X. Xie, and G. Sun, “Driving with knowledge from the physical world,” in *the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*, Aug. 2011, pp. 316–324.

[14] Los Alamos National Laboratory. (2011) all systems failure/interrupt data 1996-2005. [Online]. Available: http://institute.lanl.gov/data/fdata/