



Distributed topic management in publish-process-subscribe systems on edge-servers for real-time notification service

Tanaka, Tomoya
Kamada, Tomio
Ohta, Chikara

(Citation)

IEICE Communications Express, 9(12):616-621

(Issue Date)

2020-12-01

(Resource Type)

journal article

(Version)

Version of Record

(Rights)

© 2020 The Institute of Electronics, Information and Communication Engineers

(URL)

<https://hdl.handle.net/20.500.14094/90008800>



Distributed topic management in publish-process-subscribe systems on edge-servers for real-time notification service

Tomoya Tanaka^{1, a)}, Tomio Kamada¹, and Chikara Ohta²

¹ Graduate School of System Informatics, Kobe University,
Rokkodai-cho, Nada-ku, Kobe, Hyogo 657–8501, Japan

² Graduate School of Science, Technology and Innovation,
Rokkodai-cho, Nada-ku, Kobe, Hyogo 657–8501, Japan

a) tomoya.tanaka@fine.cs.kobe-u.ac.jp

Abstract: The importance of real-time and data-driven notification has been growing for social services and Intelligent Transporting System (ITS). As an advanced version of Pub/Sub systems, publish-process-subscribe systems with MEC (Multi-access Edge Computing), where published messages are spooled and processed on edge servers, have been proposed. In this paper, we present a topic-based publish-process-subscribe system that allows a topic to be managed on multiple edge servers so that messages are processed near publishers and transferred to subscribers immediately. However, managing each topic on numerous edge servers can cause exhaustion of storage resources on edge servers. We introduce a simple topic allocation method on edge servers to discuss the problem. Experiments show the feasibility of our proposed system.

Keywords: real-time notification, data-driven notification, multi-access edge computing, publish-subscribe model

Classification: Network

References

- [1] A. Renznik, L.M.C. Murillo, Y. Fang, W. Featherstone, M. Filippou, F. Fontes, F. Giust, Q. Huang, A. Li, C. Turyagyenda, D. Wehner, and Z. Zheng, “Cloud RAN and MEC: a perfect paring,” ETSI White Paper, https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp23_MEC_and_CRAN_ed1_FINAL.pdf, accessed May 29, 2020.
- [2] S.A. Shaheen and R. Finson, “Intelligent transportation systems,” in *Reference Module in Earth Systems and Environmental Sciences*, Elsevier, 2013. DOI: [10.1016/B978-0-12-409548-9.01108-8](https://doi.org/10.1016/B978-0-12-409548-9.01108-8)
- [3] V. Setty, G. Kreitz, G. Urdaneta, R. Vitenberg, and M. van Steen, “Maximizing the number of satisfied subscribers in pub/sub systems under capacity constraints,” Proc. IEEE INFOCOM 2014, Toronto, ON, Canada, pp. 2580–2588, April/May 2014. DOI: [10.1109/INFOCOM.2014.6848205](https://doi.org/10.1109/INFOCOM.2014.6848205)
- [4] F. Zhang, B. Jin, W. Zhuo, Z. Wang, and L. Zhang, “A content-based publish/subscribe system for efficient event notification over vehicular ad hoc net-

- works,” Proc. 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing, Fukuoka, Japan, pp. 64–71, Sept. 2012. DOI: [10.1109/UIC-ATC.2012.21](https://doi.org/10.1109/UIC-ATC.2012.21)
- [5] B. Krishnamachari and K. Wright, “The publish-process-subscribe paradigm for the internet of things,” USC ANRG Technical Report, http://anrg.usc.edu/www/wp-content/uploads/2017/07/ANRG_TechReport_201704_PublishProcessSubscribeForIoT.pdf, accessed May 29, 2020.
- [6] S. Khare, H. Sun, K. Zhang, J. Gascon-Samson, A. Gokhale, X. Koutsoukos, and H. Abdelaziz, “Scalable edge computing for low latency data dissemination in topic-based publish/subscribe,” Proc. 2018 IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA, USA, pp. 214–227, Oct. 2018. DOI: [10.1109/SEC.2018.00023](https://doi.org/10.1109/SEC.2018.00023)
- [7] R. Kawaguchi and M. Bandai, “Edge based MQTT broker architecture for geographical IoT applications,” Proc. 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, pp. 232–235, March 2020. DOI: [10.1109/ICOIN48656.2020.9016528](https://doi.org/10.1109/ICOIN48656.2020.9016528)
- [8] J. Kreps, N. Narkhede, and J. Rao, “Kafka: A distributed messaging system for log processing,” Proc. 6th International Workshop on Networking Meets Databases (NetDB 2011), Athens, Greece, pp. 1–7, June 2011, <http://notes.stephenholiday.com/Kafka.pdf>, accessed May June 26, 2020.

1 Introduction

In recent years, the importance of real-time and data-driven notification has been growing dramatically for applications such as social services, IoT applications, and Intelligent Transporting System (ITS) [1]. For example, real-time decision-making services proposed in ITS are expected to react immediately to changes in traffic conditions, analyze the current conditions, and provide optimal vehicle behavior [2]. Multi-access Edge Computing (MEC) and Pub/Sub messaging models have been exploited to achieve instant reactions to changing conditions and immediate data-driven notifications to clients [3, 4]. In a more sophisticated version, the publish-process-subscribe model with MEC, where edge servers process messages published from clients followed by disseminating the generated notification to subscribers, have great potential to achieve real-time and data-driven notification [5].

We present a topic-based publish-process-subscribe system which allows a topic to be managed on multiple edge servers, though many researchers assume one server to manage a topic [6, 7]. In our system, publishers/subscribers can exchange messages immediately using nearby edge servers. Besides, our system duplicates messages for a topic on multiple servers so that notification can be generated locally using all messages published to the topic, which can include messages that first received by remote edge servers. Kafka also stores messages duplicated to enable fault-tolerance, but pay little attention to the geographical locality [8]. However, duplicating messages on numerous edge servers causes exhaustion of the storage capacity because the available storage capacity on edge servers is limited. In this paper, we discuss the feasibility of our publish-process-subscribe system, investigating the trade-offs between the network path length needed for notifications and the consumption of storage capacity.

2 System overview

In this section, we describe our publish-process-subscribe system, where a topic is managed on multiple edge servers. Figure 1(a) shows the overview of our system. Each publisher is assigned to an edge server that manages all topics the client publishes to. For example, Client B publishes messages to Topic P and Q, so the assigned server of Client B manages both of the topics. Besides, each subscriber is assigned to an edge server, from which they receive messages. We call the server assigned to each client the “home server”. Note that messages are spooled and processed only on publishers’ home servers, and subscribers’ home servers do not manage the subscribed topic such as Edge C in Fig 1.

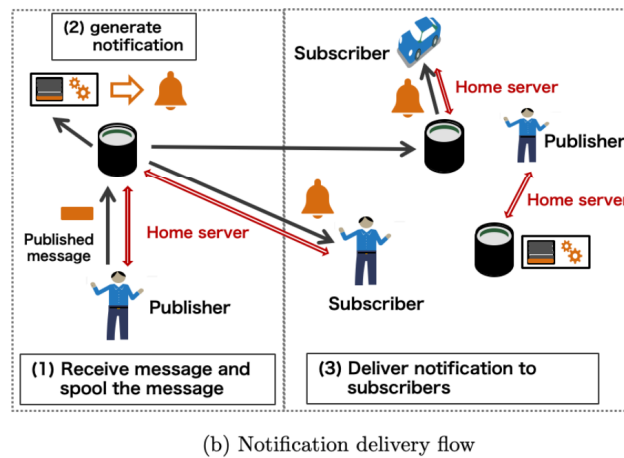
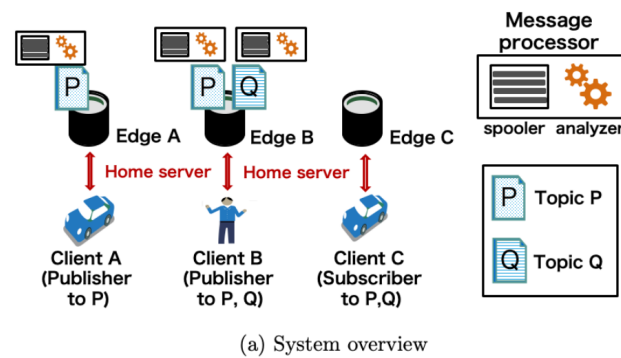


Fig. 1. Overview of our publish-process-subscribe system

Each edge server functions as a broker, which receives messages from publishers and delivers notifications to subscribers. A message processor, which functions as a message spooler and an analyzer of the spooled messages, is prepared for each managed topic. In Fig. 1(a), Edge B manages Topic P and Q, so Edge B prepares the message processor for each topic. Our system generates notifications and delivers them to subscribers in the steps shown in Fig. 1(b). When an edge server receives a message from a publisher, it spools messages in the message processor prepared for the topic. Next, the message processor analyzes the spooled messages and generates a notification. Finally, the generated notification is disseminated to subscribers’ home servers and delivered to subscribers.

Managing a topic on multiple edge servers can enable publishers/subscribers

to access nearby edge servers. However, it increases the storage consumption and may cause exhaustion of the storage capacity. We introduce a simple home server assignment method that divides the considered field into clusters and prepares only one server which manages a topic in a cluster. In each cluster, publishers to the same topic share spooled messages for the topic, so that the consumption of storage capacity is reduced.

The proximity between clients and their home server, as well as the usage of storage capacity on edge servers, are strongly influenced by the number of clusters. Let us consider the proximity and the storage capacity usage, changing the number of clusters under the condition that the number of servers is 16. There are 11 clients with a weak locality in the 20 km square field. Six of the clients publish to Topic P, while the others publish to Topic Q. Figure 2(a) shows a home server assignment when the number of clusters K is 1, which means that each topic is managed only one server in the field. On the other hand, Fig. 2(b) shows a home server assignment when $K = 16$, which means that numerous edge servers manage one topic. In Fig. 2(a), the distance between clients and their home servers can be long, whereas the usage of storage capacity is minimized. In Fig. 2(b), the storage consumption is higher than the case of Fig. 2(a), whereas the distance between clients and their home server is minimized. Our objective is to discuss the feasibility of topic allocations shown in Fig. 2(b) considering limited storage capacity on edge servers. In the experiment in Section 4, we increase the number of clusters and measures the increase of storage consumption when topics are managed by numerous edge servers. We first formulate storage consumption in the next section.

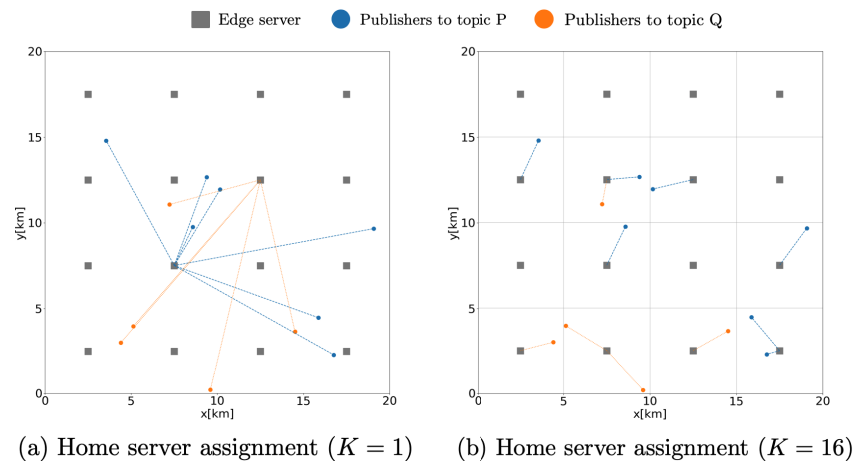


Fig. 2. Geographical view of home server assignment to clients

3 Problem formulation

We consider a field with L edge servers denoted $\mathcal{S} = \{s_1, s_2, \dots, s_L\}$. There is M active clients that include both publishers and subscribers, denoted $\mathcal{C} = \{c_1, c_2, \dots, c_M\}$, in the field. Each client $c_m \in \mathcal{C}$ has a home server $s_l \in \mathcal{S}$. Let \mathcal{C}_l denote the set of clients whose home server is s_l . The home server assignment, which determined the topic allocation on edge servers, is denoted by a binary

matrix $X = \{x_{m,l} | c_m \in C, s_l \in S\}$. If c_m 's home server is s_l (*i.e.* $c_m \in C_l$), $x_{m,l} = 1$, whereas $x_{m,l} = 0$ in the other cases.

We formulate storage consumption on edge servers as well as the proximity between publishers/subscribers and their home server. Let u_l denote the total size of spooled messages managed by the edge server s_l . As for the proximity between home servers and clients, let g_m denote the distance between a client c_m and its home server. The value of g_m and u_l is determined by the assignment X .

We increase the number of clusters in the assignment method described in Section 2 and measures the influence on the proximity g_m and the storage consumption u_l . The assignment method is elaborated by the following. We divide a field into K clusters based on edge servers' position using the K -means clustering algorithm. A client is assigned to a server belonging to the closest cluster. Dividing the field into numerous clusters reduces the value of g_m , but increase the value of u_l .

4 Experiments

We conduct simulations to investigate the increase of the edge server resource usage u_l as the field is divided into clusters finely, which reduces the value of g_m . The assignment X , and the value of u_l , g_m would be different depending on the number of clusters K .

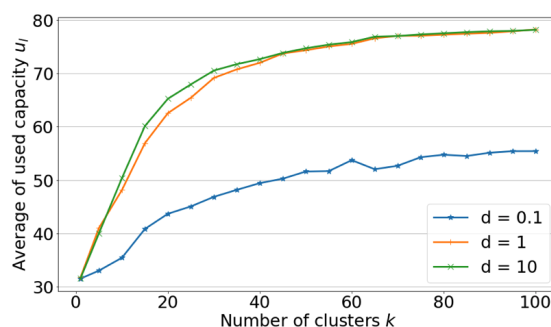
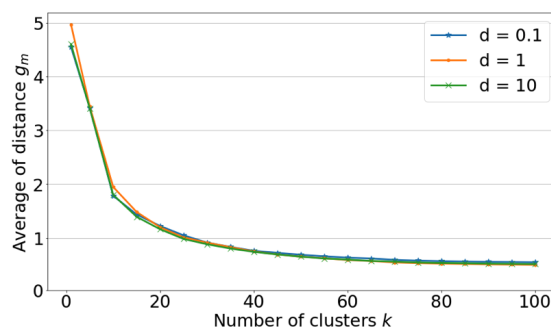
We adopt a metric Y_1 , and Y_2 , which denote the average of g_m and the average of u_l , respectively. We observe the value of Y_1 and Y_2 changing the number of clusters K . In the simulation, we set the number of edge servers L as 100. As preparation for the simulations, we distribute 1865 clients who can take the publisher's role following a Gaussian distribution with a standard deviation d centered on a randomly chosen point in 10 km square field. We conduct simulations in the case of $d = 0.1$ km, $d = 1$ km and $d = 10$ km.

We assign a home server to each client after dividing the field into K clusters. We give 1 MB spooled messages in advance in all message processors. Therefore, u_l is calculated as the number of topics managed by edge server s_l , multiplied by 1 MB.

Figure 3(a) shows the distance between publishers and their home server against the number of clusters, while Fig. 3(b) shows the average storage capacity usage. We can observe that dividing the field into small pieces (*i.e.* K is close to $L = 100$) reduces the average distance between clients and their home server smaller than 1 km in the 10 km square field. As the value of K increases, the average storage capacity usage increases. But, the difference of used storage capacity between $K = 1$ to $K = 100$ is smaller than a certain value, even when the locality of clients are weak (*i.e.* $d = 10$). In the case of these simulations, the difference is less than 50 MB. If the extra preparation for the storage capacity is practicable, we can manage topics on numerous edge servers and achieve proximity between clients and their home servers.

5 Conclusion

In this paper, we presented a publish-process-subscribe system that allows a topic to be managed on multiple edge servers so that information can be exchanged

(a) Average distance between home servers and clients (Y_1)(b) Average usage of storage capacity (Y_2)**Fig. 3.** Trade-off measurements

immediately between nearby publishers/subscribers. We modeled the presented system. Numerical experiments demonstrated the feasibility of our system, where each topic are managed on numerous edge servers.

Future work involves constructing cooperative management method of the spooled messages between edge servers, so that edge servers which is not heavily used can spool messages instead of servers which is heavily used. Moreover, we will consider two-hierarchical publish-process-subscribe system with edge servers and cloud servers.