Topic-based distributed publish-process-subscribe system with metrics on geographic distance and permissible delay

Matsumoto, Ryota
Kamada, Tomio
Finnerty, Patrick
Ohta, Chikara

(Citation)
IEICE Communications Express,11(12):748-753

(Issue Date)
2022-12-01

(Resource Type)
journal article

(Version)
Version of Record

IEICE Communications Express, Vol.11, No.12, 748–753
Special Cluster in Conjunction with IEICE General Conference 2022

Topic-based distributed publish-process-subscribe system with metrics on geographic distance and permissible delay

Ryota Matsumoto[1, a)], **Tomio Kamada**[2], **Patrick Finnerty**[1], **and Chikara Ohta**[1]

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

[2] *Department of Intelligence and Informatics, Konan University, Nishi Okamoto, Higashinada-ku, Kobe, Hyogo 658–0073, Japan*

a) *rmatsumoto@fine.cs.kobe-u.ac.jp*

Abstract: As IoT devices spread, real-time data analysis and notification are increasingly important. To achieve low latency notifications in edge environments, we have proposed a distributed publish-process-subscribe system that allows multiple edge servers to handle a single topic. In our system, topic allocation to edge servers is formalized as a notification delay optimization problem. In this article, we introduce new metrics that consider the geographic distance of respective pub/sub pairs, or the permissible delay defined by each topic. Using these metrics succeeded in reducing the notification delay to neighboring subscribers and prioritizing topic allocation to meet the delay limit in simulation.

Keywords: Pub/Sub, edge computing, real-time notification, data-driven notification

Classification: Network

References

[1] P.T. Eugster, P.A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," ACM Comput. Surv., vol. 35, no. 2, pp. 114–131, 2003. DOI: 10.1145/857076.857078

[2] B. Krishnamachari and K. Wright, "The publish-process-subscribe paradigm for the internet of things," Tech. Rep., USC ANRG Technical Report, ANRG 2017-04, 2017.

[3] T. Tanaka, T. Kamada, and C. Ohta, "Topic allocation method on edge servers for latency-sensitive notification service," International Journal of Network Management, vol. 31, no. 6, e2173, 2021. DOI: 10.1002/nem.2173

[4] J. Kreps, N. Narkhede, and J. Rao, "Kafka: a distributed messaging system for log processing," Proceedings of 6th International Workshop on Networking Meets Databases (NetDB), pp. 1–7, 2011.

DOI: 10.1587/comex.2022COL0009
Received May 31, 2022
Accepted June 28, 2022
Publicized July 8, 2022
Copyedited December 1, 2022

1 Introduction

As IoT devices become more widespread, there are increasing chances for real-time notification applications that analyze data from devices and deliver useful notifications to consumers. Especially in the intelligent transport system field, a candidate is collision detection for drivers or automatic driving vehicles.

Publish-subscribe (Pub/Sub) systems are attracting attention as systems with real-time properties, loose coupling, and scalability [1]. The publish-process-subscribe system is an extension that collects and analyzes the data published on each topic and delivers notifications to the subscribers [2]. Our group has proposed a distributed publish-process-subscribe system for edge computing environments that allows multiple edge servers to handle a single topic [3]. Apache Kafka [4] also assigns multiple brokers to a single topic for fault tolerance and scalability but does not consider geographic location. Our system can allocate multiple brokers to a topic to enable quick notification to nearby subscribers. We have formalized topic allocation to edge servers as a notification delay optimization problem.

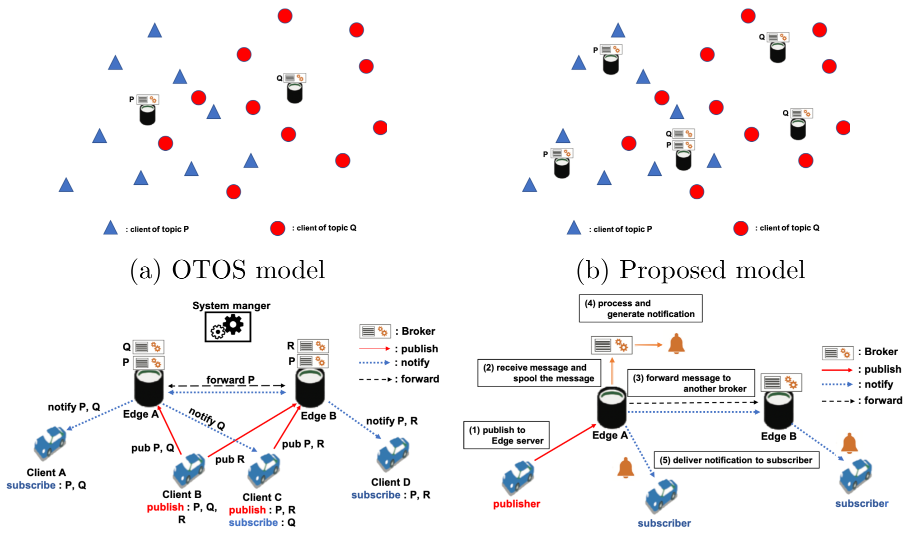
In this paper, we introduce two new metrics for topic allocation. One uses weight based on the geographic distance of respective pub/sub pairs and the other handles the permissible delay defined by each topic. Respectively, these metrics are intended to preferentially decrease notification delays to neighboring subscribers and to meet the permissible delay. In addition to improving the metrics, we have relaxed the restrictions on publishers so that they can submit messages to a separate server for each topic. This avoids concentrating the load on specific edge servers and removes redundant communication paths through the servers. In our evaluation, we use the mathematical optimization solver to find the optimal allocation that minimizes the proposed metrics. As a result, the metrics succeeded in reducing the notification delay to adjacent subscribers or prioritizing topic assignments so that permissible delays are met.

2 System overview

The One Topic One Server (OTOS) model, where a single topic is handled by a single server, works well when the publishers and subscribers of a topic are in adjacent locations (Fig. 1(a)). However, if a topic does not have enough locality, publishers or subscribers that reside far from the servers suffer from communication delays even if pub/sub pairs are located nearby. To improve the situation, our model allows multiple servers to handle the same topic to allow fast notification to neighboring subscribers (Fig. 1(b)).

Figure 1(c) gives an overview of our system and Fig. 1(d) illustrates the notification flow. (1) A publisher publishes messages for a topic to an edge server specified by the system manager. (2) The messages are handled on the edge server by the broker which consists of a spooler and a processor. (3) If there are other brokers for this topic, the message is also forwarded and spooled to these brokers. (4) Thus, the messages are spooled by all brokers about the topic, but only the broker that directly receives the message processes it using the already spooled message for the topic and generates notifications. (5) Finally, the notifications for subscribers are delivered via an edge server, called *home server* for the subscriber.

DOI: 10.1587/comex.2022COL0009
Received May 31, 2022
Accepted June 28, 2022
Publicized July 8, 2022
Copyedited December 1, 2022



(a) OTOS model (b) Proposed model

(c) System model (d) Notification flow

Fig. 1. System overview

The above model has relaxed restrictions on publishers compared to the previous model [3]. In the previous model, each client is assigned to one edge server and publishes/subscribes messages about all topics via that server. However, because of this constraint, brokers tend to concentrate on specific edge servers, which increases the possibility of exceeding storage capacity. The new model allows publishers to submit messages to separate servers for each topic. The purpose is to reduce unnecessary concentration and redundant communication paths by selecting servers according to the distribution of publishers/subscribers for each topic.

The system needs to determine the topic allocation to edge servers and the assignment of edge servers to which clients publish messages for each topic. If the system allocates too many topics to an edge server and the total message spool exceeds the server's storage capacity, the broker function for some topics is delegated to the cloud server, increasing communication delays.

3 Formulation

The trade-off between latency reduction and storage consumption makes it difficult to decide where to allocate brokers. In the previous study [3], we introduced a formal model of our publish-process-subscribe system and formulated the topic allocation problem as an optimization problem of the total delay.

This paper introduces two metrics. The first uses weights based on the geographic distance between the publishers and the subscribers. This metric is intended to reduce delays in notifications to nearby subscribers. For example, when developing an application that generates notifications to avoid vehicle collisions, quick notification to nearby vehicles will be essential. The other metric deals with the permissible delay for each topic. This can suppress resource consumption by topics that already satisfy their permissible delay and prioritize broker allocation towards topics that still need to reduce their latency.

DOI: 10.1587/comex.2022COL0009
Received May 31, 2022
Accepted June 28, 2022
Publicized July 8, 2022
Copyedited December 1, 2022

3.1 System model

This section briefly shows the major variables used in our model and points out the difference from the previous study [3]. The set of clients acting as publishers or subscribers is denoted as $C = \{c_1, c_2, \ldots, c_M\}$. $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$ is the set of topics and A_n is the size of the spooler for $t_n \in \mathcal{T}$. $\mathcal{E} = \{e_1, e_2, \ldots, e_L\}$ is the set of edge servers and v_l is the storage capacity of $e_l \in \mathcal{E}$. Let $d_{m,l}$ be the communication delay between $c_m \in C$ and e_l, and $d^{\mathrm{S}}_{l,l'}$ be the delay between e_l and $e_{l'} \in \mathcal{E}$. When brokers for some topics are delegated to the cloud for a capacity reason, the delay d^{C} to the cloud will be needed, which is a relatively large value relative to $d_{m,l}$ and $d^{\mathrm{S}}_{l,l'}$. Let $g^{\mathrm{C}}_{m,m'}$, $g_{m,l}$, and $g^{\mathrm{S}}_{l,l'}$ be the geographical distance between the client pair, the client-edge pair, and the edge server pair, respectively. β_n represents the permissible delay for t_n.

3.2 Problem formulation

The topic allocation problem is formulated as an optimization problem under the given Pub/Sub relationships and locations of edge servers and clients. The following are the variables that represent the assignment of clients and topics to edge servers (Fig. 2(b) (2)–(7)). In the new model, $x^n_{m,l}$ and $y_{m,l}$ represent the assignments of clients to edge servers. $x^n_{m,l}$ represents whether c_m publishes messages about topic t_n via server e_l. $y_{m,l}$ represents whether c_m subscribes messages from server e_l. In addition, z^n_l represents whether a broker for topic t_n is actually allocated to server e_l.

$D^n_{m,m'}$ represents the delay from publisher c_m to subscriber $c_{m'}$ about topic t_n as shown in Fig. 2(b) (11). The third and fourth terms on the right-hand expression are penalty terms when the total spool size exceeds the storage capacity of the edge server and the broker cannot be allocated to the edge server e_l.

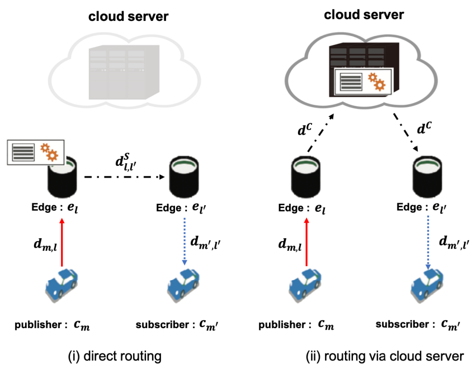
The objective function of the optimization problem is the metric D_1, D_2, or D_3. D_1 corresponds to the metrics used in the previous study. It represents the total sum of the delays between all publisher/subscriber pairs. In D_2, the delay $D^n_{m,m'}$ is multiplied by a weight $\alpha_{m,m'}$, which is based on the geographic distance $g^{\mathrm{C}}_{m,m'}$ between c_m and $c_{m'}$. D_3 is defined as Fig. 2(b) (10) using the permissible delay β_n for topic t_n.

4 Evaluation

We use the Gurobi Optimizer, a mathematical optimization solver, and experiment with the effectiveness of new metrics using an artificial small data set. The data set contains 30 clients and 7 servers randomly placed in a 10 km square field, as shown in Fig. 3(a). There are 6 topics with different permissible delays. We assume that the communication delay is proportional to the length of its communication path. We set $d_{m,l} = \gamma g_{m,l}$ and $d^{\mathrm{S}}_{l,l'} = \gamma g^{\mathrm{S}}_{l,l'}$ for a γ (ms/km). γ is set to 0.1 in this simulation.

First, we compare the first metric D_2 with the previous one D_1. Figure 3(b) shows the results for a topic. Each point represents a publisher/subscriber pair, and the x and y coordinates represent their geographic distance and communication delay. The delay decreases and increases in the left/right area, respectively. This means that we succeeded in obtaining an assignment that preferentially reduces delay to nearby subscribers using the new metric.

DOI: 10.1587/comex.2022COL0009
Received May 31, 2022
Accepted June 28, 2022
Publicized July 8, 2022
Copyedited December 1, 2022



(a) Differences in routes with and without brokers

$$\text{minimize} : D \tag{1}$$

$$\text{s.t} : x^n_{m,l} = \{0,1\}, \forall c_m \in \mathcal{C}, \forall e_l \in \mathcal{E}, \forall t_n \in \mathcal{P}_m \tag{2}$$

$$\sum_{l=1}^{L} x^n_{m,l} = 1, \forall c_m \in \mathcal{C}, \forall t_n \in \mathcal{P}_m \tag{3}$$

$$y_{m,l} = \{0,1\}, \forall c_m \in \mathcal{C}, \forall e_l \in \mathcal{E} \tag{4}$$

$$\sum_{l=1}^{L} y_{m,l} = 1, \forall c_m \in \mathcal{C} \tag{5}$$

$$z^n_l = \{0,1\}, \forall e_l \in \mathcal{E}, \forall t_n \in \mathcal{T} \tag{6}$$

$$\sum_{n=1}^{N} A_n z^n_l \leq v_l, \forall e_l \in \mathcal{E} \tag{7}$$

where

$$D_1 = \sum_{c_m \in \mathcal{C}} \sum_{t_n \in \mathcal{P}_m} \sum_{c_{m'} \in \mathcal{S}_n} D^n_{m,m'} \tag{8}$$

$$D_2 = \sum_{c_m \in \mathcal{C}} \sum_{t_n \in \mathcal{P}_m} \sum_{c_{m'} \in \mathcal{S}_n} \alpha_{m,m'} D^n_{m,m'} \tag{9}$$

$$D_3 = \sum_{c_m \in \mathcal{C}} \sum_{t_n \in \mathcal{P}_m} \sum_{c_{m'} \in \mathcal{S}_n} \max(D^n_{m,m'} - \beta_n, 0) \tag{10}$$

$$D^n_{m,m'} = \sum_{l=1}^{L} d_{m,l} x^n_{m,l} + f_n + \sum_{l=1}^{L} \sum_{l'=1}^{L} d^{\mathrm{S}}_{l,l'} z^n_l x^n_{m,l} y_{m',l'} + \sum_{l=1}^{L} 2d^{\mathrm{C}}(1 - z^n_l) x_{m,l} + \sum_{l'=1}^{L} d_{m',l'} y_{m',l'} \tag{11}$$

$$\alpha_{m,m'} = \frac{1}{g^{\mathrm{C}}_{m,m'} + 0.1} \tag{12}$$

(b) Optimization problem

Fig. 2.

We then compare the second metric D_3, which takes the permissible delay into account, with the previous one D_1. Figures 3(c) and 3(d) show the result for two topics, `t1` and `t2`. The permissible delay is shorter for `t1` than for `t2`.

In Fig. 3(c), many publish/subscribe pairs that exceed the permissible delay have succeeded in reducing the delay by using D_3 compared to D_1. In Fig. 3(d), by contrast, many publish/subscribe pairs increase their communication delay by using D_3, but keeping within the permissible range. D_3 seems to have successfully prioritized the broker allocation for `t1` over that of `t2`.

DOI: 10.1587/comex.2022COL0009
Received May 31, 2022
Accepted June 28, 2022
Publicized July 8, 2022
Copyedited December 1, 2022

ComEX

IEICE Communications Express, Vol.11, No.12, 748–753

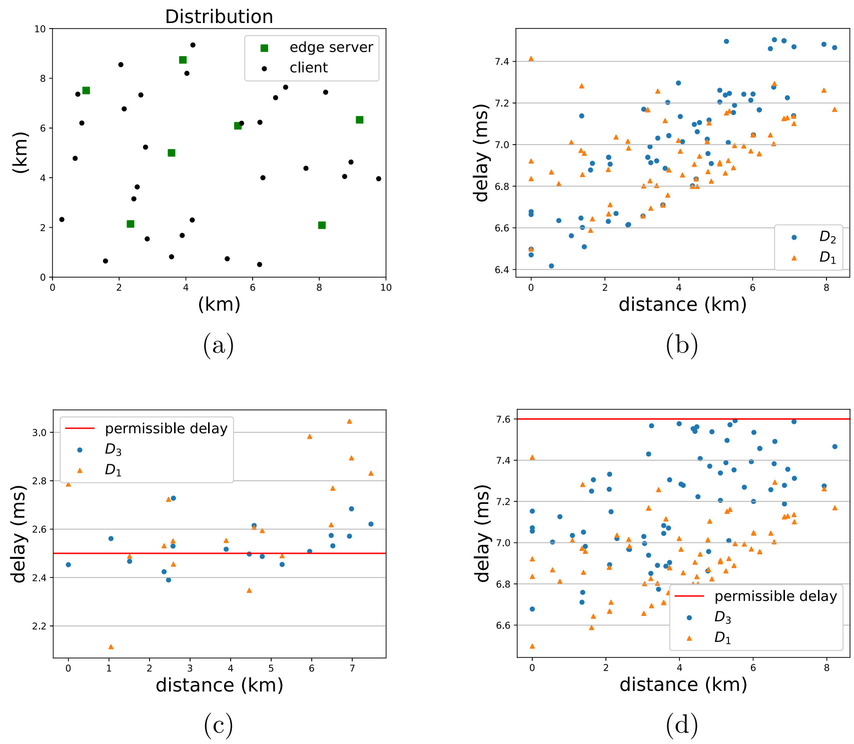


Fig. 3. Simulation result

5 Conclusion

In this paper, we improved the model of our distributed publish-process-subscribe system and proposed new two metrics. The one using weights based on the geographical distance between each publisher/subscriber pair succeeded to reduce notification delays to neighboring subscribers. The other specifies the permissible delay for each topic. It succeeded in prioritizing the broker allocation for topics that require latency reduction. We intend to extend our model to handle mobile clients in future work, allowing the system to be used in intelligent transport systems.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP18H03232, JP20K11841, JP22H03585.

DOI: 10.1587/comex.2022COL0009
Received May 31, 2022
Accepted June 28, 2022
Publicized July 8, 2022
Copyedited December 1, 2022