Future Generation Computer Systems 54 (2016) 159–167

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

DTA: Dynamic topology algorithms in content-based Publish/Subscribe

Xia Xie *, Hongguang Wang, Hai Jin, Feng Zhao, Xijiang Ke, Laurence Tianruo Yang

Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

ARTICLE INFO

Article history:
Received 28 September 2014
Received in revised form
19 December 2014
Accepted 19 January 2015
Available online 2 February 2015

Keywords:
Publish/Subscribe
Content-based
CYCLIC
Real time

ABSTRACT

Publish/Subscribe (Pub/Sub) has been used in many large-scale distributed systems owing to its scalability and loose coupling, and most of the existing Pub/Sub systems are built on the basis of the acyclic topology because the routing algorithm can be simplified. However, the flexibility is limited and the robustness is also a big problem. This paper proposes a dynamic topology algorithm to facilitate the adaptation to the dynamic change of the network conditions, which is composed of two parts: Topology Generation Algorithm (TGA) and Topology Transformation Algorithm (TTA). The topology can switch between cyclic topology and acyclic topology automatically, and the advantages of both the cyclic topology and the acyclic topology can thus be exploited. In addition, the efficiency and scalability can also be guaranteed when the topology changes. The topology of a Pub/Sub system will be transformed into a cyclic one to enhance robustness when the network condition gets worse, while into the acyclic pattern when the network condition is getting better. Experiments show that average number of edges in the topology reduces by 12% and subscription notification delivery time improves by 20%.

1. Introduction

With the rapid development of social networks and mobile Internet, numerous new systems have been built in these fields. In order to make these applications or systems more flexible and scalable, a decoupled, scalable, and dynamic communication scheme is required, such as the Pub/Sub paradigm that is widely used as a powerful messaging paradigm. Due to its loose coupling [1], simple interface, and inherent scalability, Pub/Sub has been attracting a number of companies and research institutes. In recent years, the effort of designing the Pub/Sub system has been made by researchers from the academy and industry [2–4]. For the researchers from academy, they focused on the enhancement of the system expression capacity (e.g., content-based routing algorithms and semantic [5,6]) and the compatibility with the new application environments (e.g. P2P network [7,8]). In industry, practical applicability is one of the main concerns, including the robustness [9], security [10], and efficiency [11]. A number of large-scale distributed applications (e.g., stock-market monitoring engines, RSS feeds, and workflow management) have been built upon the Pub/Sub paradigm [12].

In general, there are two types of Pub/Sub, namely topic-based and content-based Pub/Sub systems [13,14]. Topic-based Pub/Sub uses the notion of topics (or subjects), and extends the concept of the channel that connects two communication peers. Users can publish events and subscribe to individual topics that are identified by keywords. The communication channels in Content-based Pub/Sub are specified through a collection of attributes that could be arbitrary data types. Unlike the topic-based Pub/Sub variant that is a static scheme and offers the limited expressiveness, the content-based Pub/Sub introduces a subscription scheme according to the actual content of the considered events. At present, the content-based pub/sub has been used in many areas due to its expressiveness, filtering abilities, distributed event correlation, and complex event process capabilities. In this paper, a content-based Pub/Sub system is taken into consideration.

For a Pub/Sub paradigm, it is composed of brokers that are responsible for disseminating messages. The topology of a Pub/Sub can be classified into two categories: the cyclic topology with at least one cycle and the acyclic topology with no cycle. Most of the existing Pub/Sub systems have been so far built on the basis of the acyclic topology. This is because the routing algorithm can be simplified when an acyclic topology is used. In an acyclic topology based system, any two peers are connected via one unique path in a Pub/Sub overlay network. The flexibility of this kind of system, however, is still limited when the network conditions change

* Corresponding author.
E-mail address: shelicy@hust.edu.cn (X. Xie).

http://dx.doi.org/10.1016/j.future.2015.01.010
0167-739X/

dynamically. Moreover, the robustness is a big problem. Once some accidental situations (e.g., the broker failure and failure recovery) occur, the systems cannot deal with it very well.

Although these problems cannot occur in cyclic topology based systems, using cyclic topology means more brokers required to be configured and more paths between brokers in the overlay network. Moreover, in order to maintain the overlay topology, each node need continuously monitor the availability of its neighbors, which will lead to high cost [15]. Therefore, it is of importance for a Pub/Sub system to adapt to complicated network conditions. However, the previous work supports only one kind of topology, which cannot meet the requirement of the adaption.

To the end, this paper proposes a dynamic topology algorithm to facilitate the adaptation to the dynamic change of the network conditions. The topology can switch between cyclic topology and acyclic topology automatically, and the advantages of both the cyclic topology and the acyclic topology can thus be exploited. In addition, the efficiency and scalability can also be guaranteed when the topology changes. The topology of a Pub/Sub system will be transformed into a cyclic one to enhance robustness when the network condition gets worse, while into the acyclic pattern when the network condition is getting better. The dynamic topology algorithm is composed of two parts: Topology Generation Algorithm (TGA) and Topology Transformation Algorithm (TTA). These two algorithms are responsible for dealing with the topology transformation problem in Pub/Sub systems on the basis of relative operations, such as the monitoring and analysis. TGA is used to construct the initial topology based on the original network conditions, and TTA is used for detecting and transforming the topology in the Pub/Sub system. In particular, because the content-based Pub/Sub can process complex events and provide the expressiveness on distributed event correlation, these features are exploited by both TGA and TTA to simplify the two algorithms. Extensive experiment results show that the performance is significantly improved.

The paper is organized as follows. The system structure is described in Section 2 and our topology strategy is presented in Section 3. Section 4 shows our topology algorithms in detail and Section 5 shows our experiments results and the discussions. Finally, the paper is concluded, and the future wok is provided.

2. System structure

The content-based Pub/Sub system consists of four layers as shown in Fig. 1. The first layer is the user interface that is composed of two components: the client and the monitor. The second layer, the core level of a Pub/Sub system, is responsible for interacting with, controlling, and managing other parts of the system. It includes the topology controller, detector and analyzer. The following is the router layer that controls and adjusts all the incoming and outgoing information. The last layer is the broker layer that provides the infrastructure.

2.1. User interface

Clients connect with the specific brokers to transmit all the messages. Due to the flexible and efficient deployment, clients are also used to complete some tests by attaching them to the target brokers. In the system, according to the test results obtained by the analysis, the system design is possibly adjusted, and the algorithm may also be modified. In addition, Monitor module not only processes the display and presentation of the running work but also handles some of the configuration task.

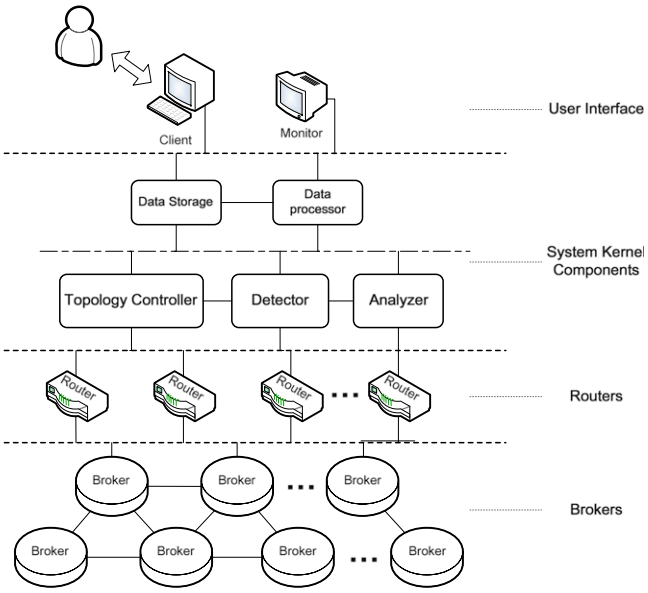

Fig. 1. System structure diagram.

2.2. System core components

As mentioned above, the second layer consists of several modules, including Topology Controller, Detector, and Analyzer. They deal with specific tasks, respectively. These modules cooperate with each other. After the overlay network is deployed, owing to the dynamic network conditions, the network structure needs to be detected, analyzed and adjusted continuously. Detector module is used for collecting the detection results, generating the detection data, and sending the detection signal. Analyzer module is used to process the detected results and then analyze them to get the statistics characteristics that will be used in the future. Topology Controller module is of importance because it is involved in many operations, such as the overlay initialization, the detector controlling, the analyzer controlling, topology transformation, and error processing. In the meanwhile, Data Storage module stores the test data to be processed by Data Processor.

2.3. Routers

The router is mainly responsible for information transmission. Routers are divided into two parts based on their locations around brokers. Some routers lie in brokers to route the messages that enter brokers while other routers are active outside the brokers for the purpose of keeping track of the broker states and routing the controlling messages. Besides delivering messages to the right destinations, the routers in the system can also adapt themselves to the network topology of the brokers. It means that the routers can update their routing table and immediately adjust their message matching strategy when the network topology changes. In this paper, a special routing strategy is proposed to make routers more flexibly adapt to the network topology change, which will be discussed in detail in Section 4.

2.4. Brokers

The broker is the fundamental element in the system, and is used to build the infrastructure. It contains some necessary subcomponents, such as the message queue, the communication interface, and the matching engine. Firstly, the network topology is mainly consisted of brokers. Participators, the publishers or subscribers, are able to connect to the topology by connecting with