Towards Scalable Publish/Subscribe Systems

Shuping Ji[1], Chunyang Ye[2], Jun Wei[1] and Hans-Arno Jacobsen[3]

[1]Chinese Academy of Sciences, Beijing, China
[2]Hainan University, Hainan, China
[3]Middleware Systems Research Group

***Abstract*—Despite suffering from inefficiency and flexibility limitations, the filter-based routing (FBR) algorithm is widely used in content-based publish/subscribe (pub/sub) systems. To address its limitations, we propose a dynamic destination-based routing algorithm called D-DBR, which decomposes pub/sub into two independent parts: Content-based matching and destination-based multicasting. D-DBR exhibits low event matching cost and high efficiency, flexibility, and robustness for event routing in small-scale overlays. To improve its scalability to large-scale overlays, we further extend D-DBR to a new routing algorithm called MERC. MERC divides the overlay into interconnected clusters and applies content-based and destination-based mechanisms to route events inter- and intra-cluster, respectively. We implemented all algorithms in the PADRES pub/sub system. Experimental results show that our algorithms outperform the FBR algorithm.**

I. Introduction

Due to its asynchronous nature and inherent decoupling properties, the distributed content-based publish/subscribe paradigm (pub/sub, for short) has been widely used in the design of many distributed applications. The routing algorithm employed by a pub/sub system is crucial to managing performance, load distribution, and scalability. However, it is a challenging undertaking to design efficient and scalable routing algorithms for pub/sub. Currently the most widely used filter-based routing algorithm (FBR) [1] suffers from the following four limitations: (1) Difficulty in supporting general overlay topologies, (2) subscription duplication, (3) redundant and repeated event matching, and (4) lack of flexibility in supporting overlay reconfiguration.

We observe that the aforementioned limitations in FBR result from the coupling of event matching and event routing: each routing decision is based on the result of event matching. To overcome these limitations, we first propose a dynamic destination-based routing algorithm called D-DBR, which decouples the pub/sub system into two independent layers: Content-based matching and destination-based multicasting. The matching layer is responsible for subscription and event matching, whereas the multicasting layer is responsible for topology maintenance and message routing. When a message (advertisement, subscription, or event) is issued, it is first submitted to the matching layer to identify destination brokers. Then, the message is annotated with the addresses of those brokers and delivered to them via the multicasting layer. In D-DBR, subscriptions are not stored at any intermediate broker. An event only needs to be matched at its source and destination brokers. Changes to the overlay topology have no effect on the matching layer. Thus, supporting general overlays and dynamic overlay reconfiguration for fault-tolerance and performance optimizations becomes straight forward.

Although D-DBR is an effective solution, a factor limiting its scalability is that each broker needs to know all other brokers in the system, and thus, the topology maintenance cost can be expensive for large-scale networks (with hundreds or more brokers). To mitigate this issue and to achieve better scalability, we also propose a new routing scheme called MERC — Match at Edge and Route intra-Cluster. MERC divides the overlay into interconnected clusters, where it applies content-based and destination-based mechanisms for inter- and intra-cluster event routing, respectively. In MERC, each broker only needs to be aware of brokers in the clusters it belongs to. As a result, the destination list overhead is mitigated, the topology maintenance cost is reduced, and the impact of changes in one cluster can be isolated from brokers in other clusters.

We implemented both algorithms, D-DBR and MERC, in PADRES [3], [4], an open-source content-based pub/sub system. Our experimental results show that our algorithms outperform FBR in terms of improving the system throughput by up to 700% and reducing the communication latency by up to 55%, while the newly introduced overhead remains acceptable.

A more extensive technical report of this work is available in [5].

II. D-DBR Design

As shown in Fig. 1, in D-DBR, the pub/sub system is decoupled into two independent layers: Content-based matching and destination-based multicasting. The matching layer is responsible for event matching, whereas the multicasting layer is responsible for event routing. When a publisher issues an event at a broker, the event is matched against subscriptions managed by the broker's matching engine to obtain the addresses of brokers interested in the event (i.e., brokers hosting clients who are subscribed to the event.) The addresses are attached to the event. Then, the event is delivered to the interested brokers by the multicasting layer based on the event's destination addresses. Upon receiving an event, a destination broker matches the event against its local subscriptions and directly delivers it to the interested subscribers.

The matching engine of each broker maintains four routing tables: Local Subscription Routing Table (L-SRT), Remote Subscription Routing Table (R-SRT), Local Publication Routing Table (L-PRT), and Remote Publication Routing Table (R-PRT). Separating the routing information of local clients from that of other brokers reduces the message matching cost. The multicasting engine maintains two routing tables: The Topology Routing Table (TRT) and the Shortest Path Routing Table (SPRT). The multicasting layer provides a

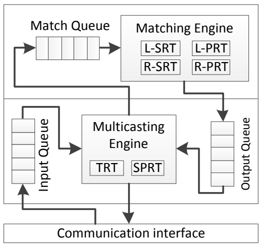

Fig. 1: Layers of D-DBR

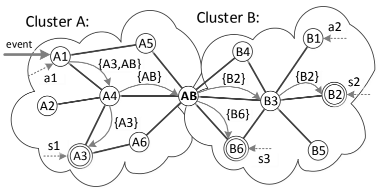

Fig. 2: Event routing in MERC

simple and efficient destination-based one-to-many message delivery service. At this layer, advertisements, subscriptions and events are routed in the same way, which simplifies the pub/sub system's design and implementation. Moreover, each message is delivered to its destinations along the shortest paths.

In D-DBR, content-based matching and destination-based multicasting are decoupled, changes to the overlay do not impact the routing tables of the matching engine. As a result, D-DBR can easily support dynamic overlay reconfiguration, enabling better fault-tolerance and performance optimizations.

III. MERC Design

For improved scalability in large-scale broker overlays, we propose another routing scheme called MERC—Match at Edge and Route intra-Cluster. MERC combines destination-based and content-based routing hierarchically. It has the advantages of D-DBR, i.e., low subscription duplication, low matching cost, etc. It also overcomes the scalability limitation of D-DBR: In MERC, each broker needs to know a limited number of brokers only, and the destination list is limited to brokers in the local cluster.

In MERC, the broker overlay is divided into interconnected clusters of brokers. Some brokers, called *edge brokers*, are located at the edge of clusters and belong to more than one cluster, whereas the other brokers, called *internal brokers*, belong to only one cluster. Each broker only knows the addresses of brokers in clusters it belongs to. Content-based and destination-based mechanisms are adopted for inter- and intra-cluster event routing, respectively.

In MERC, when an event is issued, it is first matched against subscriptions at the local broker to identify interested brokers in the local cluster. Then, the event is delivered to these brokers along the optimal paths, according to D-DBR. Once an event is received by an edge broker that also belongs to another cluster, the event is matched against subscriptions from that cluster at the edge broker to identify interested brokers in that cluster. It is then delivered to these brokers from the edge broker, again, according to D-DBR. This process is repeated until the event is delivered to all interested brokers in all clusters. Fig. 2 is an example of event routing in MERC. In this example, the topology is divided into two clusters and broker AB acts as the edge broker.

In MERC, routing tables of internal brokers at both the matching layer and the multicasting layer are the same as those in D-DBR: Each broker maintains the same six routing tables. But the edge brokers have different routing tables. Besides an L-SRT and an L-PRT, an edge broker maintains a group of the other four routing tables for each cluster it belongs to.

Whenever a message from a specific cluster is delivered to an edge broker, the *sourceID* of that message is first replaced by that edge broker's ID. Then, that message is processed based on its type: An advertisement is forwarded to all brokers in the other clusters, a subscription is forwarded to brokers in the other clusters with matching advertisements, and an event is forwarded to brokers in other clusters with matching subscriptions.

Today's Internet can be viewed as a collection of interconnected routing domains [2], which are groups of nodes that are under a common administration and share routing information. MERC follows this design: One cluster can be viewed as an administrative domain and different clusters can be connected in a hierarchical manner. So an appealing characteristic of MERC is that it provides a good reference to construct large-scale pub/sub systems that mimic the structure of the Internet.

IV. Evaluation

We implemented the D-DBR and MERC algorithm in PADRES [4], a representative, open-source, content-based pub/sub system based on the FBR algorithm. Both algorithms are evaluated through experiments run on the SciNet computing facility and experiments based on simulations. We use the FBR algorithm as a baseline. Experiments are run on an acyclic linear topology and on general topologies with cycles.

In the acyclic linear topology with 3 to 10 brokers, D-DBR and MERC achieve better performance than FBR, especially when there are a large number of subscriptions. In general topologies with 100 brokers, D-DBR exhibits the best performance and MERC lies between D-DBR and FBR: when the publishing rate (messages/minute) increases from 2,000 to 2,500, the event delivery latency of FBR increases from 1,087 ms to 2,843 ms. However, the event delivery latency of D-DBR is only 491 ms when the event publishing rate is 14,000. That is, compared with FBR, D-DBR improves the throughput by up to 700% and reduces the communication latency by up to 55%. Our experiments also show that the latency of MERC is stable and slightly higher than D-DBR when the event publishing rate increases to 11,000. This suggests that MERC can also achieve better performance than FBR.

Our experiments also show that the destination list overhead is small. For example, in the above experiment, the average destination list size at each broker ranges from 1 to 5 for D-DBR and 1 to 1.6 for MERC. On average, across the entire system, MERC exhibits a smaller destination list size than D-DBR(1.18 vs. 2.12).

This work was supported in part by the National Science Foundation of China under grant no. 61379047 and between 2008 and 20011 in part by an NSERC Discovery Grant.

References

[1] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *ACM TOCS*, 2001.

[2] D. D. Clark. Policy routing in internet protocols. *Policy*, 1989.

[3] E. Fidler, H.-A. Jacobsen, G. Li, and S. Mankovskii. The PADRES Distributed Publish/Subscribe System. *International Conference on Feature Interactions in Telecommunications and Software Systems (ICFI'05)*, pages 12–30, July 2005.

[4] PADRES. http://www.msrg.org/projects/padres/.

[5] S. Yi, C. Ye, J. Wei, and H.-A. Jacobsen. Scalable Content-based Publish/Subscribe. Technical report, Middleware Systems Research Group, University of Toronto, August 2011. http://msrg.org/papers/MERC.