Hybrid Publish-Subscribe: A Compromise Approach for Large-Scale

Roman Szarowski

Czech Technical University in Prague,
Karlovo náměstí 13, Prague 2, 121 35, Czech Republic
xszarows@fel.cvut.cz,
http://cs.felk.cvut.cz/~xszarows/

Abstract. The purpose of this article is to propose a scalable, topologically and traffic-wise self-adapting approach to a publish/subscribe paradigm for supporting event-based applications on a wide-area network. Large event systems requiring an efficient group communication as well as anonymity of particular peers tend to be fully decoupled. The publish/subscribe paradigm provides this requirement which enlarges a range of dimensions involving such system. There are two main approaches, namely topic-based and content-based publish/subscribe. The advantages and disadvantages of the approaches are often complementary one to another. This article presents a method of improving efficiency of large-scale systems while preserving scalability by involving cooperation of topic-based and content-based techniques.

1 The Publish/Subscribe Paradigm

The *publish/subscribe paradigm* is a loose communication scheme for modelling the autonomous interaction among participants in *event systems* and introduces a powerful mechanism for *distributed and mobile architectures*. On the contrary to the classic interaction models like request/reply, publish/subscribe provides three levels of decoupling: time, space and data flow. The publish/subscribe interaction scheme is represented by three types of participants: a *publisher*, a *subscriber* and a *broker*. Whereas the first two are notification providers or consumers (clients), the last one is a mediator (server) assuring data storage and delivery. A typical publish/subscribe system consists of a number of brokers linked together into a general graph. Publishers register their interest to notify about events (by sending *advertisements*) to their own broker as well as subscribers register their interest (by sending *subscriptions*) to be notified about events. It is upon particular broker to effectively disseminate appropriate notifications across the whole network of brokers on the basis of received subscriptions/advertisements. The basic system model for publish/subscribe interaction is illustrated in Figure 1. For detail description we refer to [1].

Subscribers usually register their interest in particular events, rather than in all events. The different approaches to handle the events with respect to subscriptions/advertisements have led to define two characteristic event schemes,

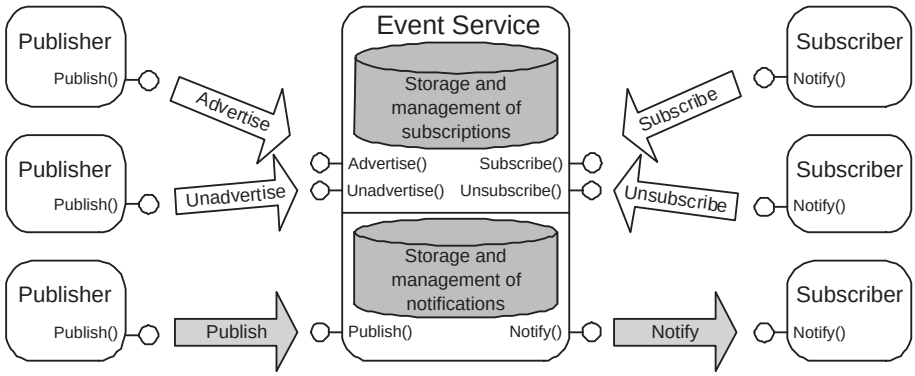

Fig. 1. A simple object-based publish/subscribe system

namely *topic-based* and *content-based publish/subscribe*.

The topic-based interaction model is based on the notion of *topics*. Every publisher or subscriber interested in publishing or receiving events related to particular topic basically joins a group. Hence, subscribing to a topic T means to become a member of a group G_T and consequently members of G_T communicate along the *subscription channel* CH_T. Commercial publish/subscribe systems are implemented broadly on the basis of this approach, e.g. [2].

The content-based publish/subscribe model gives more flexibility to the application by removing limitations on topics. Subscribers can register their individual interests by specifying the properties of the event notifications they are interested in. Typical representatives of this scheme are Gryphon [3, 4] and Siena [5]. Thanks to expressiveness of this approach the dissemination of redundant messages can be avoided. It also brings a better scalability with respect to number of brokers, because notifications are disseminated incrementally. Every single broker cares just about its direct neighbors and ignore the others.

2 The Hybrid Publish/Subscribe

If a topic-based system grows up to a large network of interconnected brokers, a particular subscription channel spreads along this network. As a consequence, a subscription group can: 1. *contain too many brokers* to handle its consistent state in reasonable time interval because group state changes become too frequent; 2. become a very *sparse set of interested brokers* in such a manner that an average network distance between two brokers within the group become intolerably high according to a number of physical network hosts on the path between them.

In content-based publish/subscribe, exchange of subscriptions and advertisement are very effectively distributed, so the problem with consistence of the system disappears. However, there is other difficulty to consider. Every event transfer is based on a *store/forward* or *store/filter/forward technique* (according to the content-based scheme in [5]). Although filter mechanisms give a routing

freedom to brokers, every event has to *pass through all the brokers* (hosts) on a path between the source and the most distant interested broker.

The hybrid publish/subscribe approach comes from the requirements on a good scalability and using sophisticated group communication techniques in cases where it is possible and suitable. The basic principle is the following.

When the system starts, every active broker works as a common peer of a content-based system with ability to manage subscriptions/advertisements, to filter and to route events. For simplicity, we adopt basics of Siena's routing strategies [5]. Since Siena's broadcasting of subscriptions/advertisements is not intended to be used in general P2P networks, we would have to use some mechanisms to avoid cycles, e.g. timestamping of requests.

After some time, our system can reach a state where an event E will be *just forwarded* among several brokers without any filtering operation. If it continues for a longer period of time, it will induce a good opportunity to make a routing optimization. These brokers transferring particular events among themselves create a group. On the basis of this heuristics, the brokers within this group can use more sophisticated group communication techniques to transfer data than a simple store/forward, e.g. IP multicast. As a consequence, we create a channel similar to channels in the topic-based publish/subscribe. We can call it a *local channel*. A number of members in the group can increase if some other members join the group or decrease if some of them leave the group. Furthermore, there may exist several local channels separately, transferring the same events in the system. Two groups may join together as well. Figure 2 illustrates this concept.

There are three types of brokers in Figure 2. Brokers represented by squares are interested in receiving event E_1. Brokers represented by triangles are interested in event E_2. Circles are the brokers without any relation neither to E_1 nor to E_2. There are also two types of lines interconnecting a pair of brokers, each of them symbolizes a different network throughput.

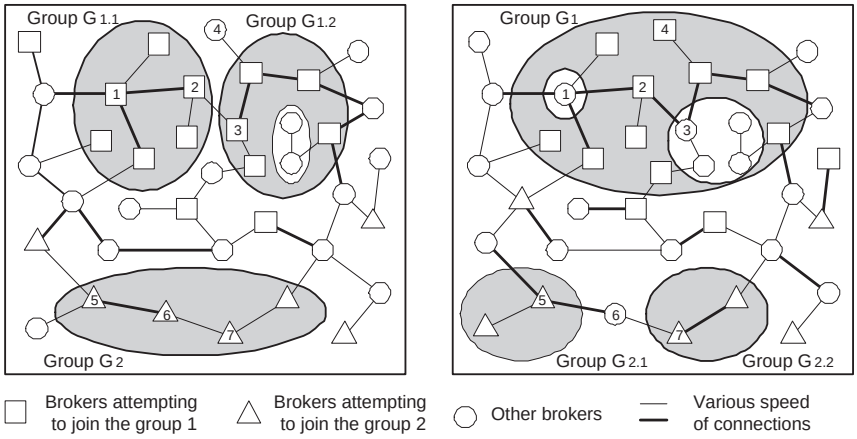

Fig. 2. Hybrid publish/subscribe scheme

On the left part of the figure we observe that several squared brokers created two groups: $G_{1.1}$ and $G_{1.2}$. Though both of them are interested in the same event, a low throughput between brokers 2 and 3 do not allow them to be a part of a single group. On the other hand, triangled brokers create a single group G_2. Note that there are many other brokers, both squared and triangled, which do not participate in groups at all.

After unspecified time interval, the state of our system can vary insofar, that the groups sharing particular events will need to be rearranged. So the considered groups G_1 and G_2 can look like these on the right part of the Figure 2. As we can see, group $G_{1.1}$ and $G_{1.2}$ merged into the single group G_1 on the contrary to the group G_2, which split into two groups $G_{2.1}$ and $G_{2.2}$. The reason why the first happened can be e.g. increased throughput of the network between brokers 2 and 3. Splitting of the group G_2 may occur e.g. due to high traffic between the brokers $5, 6$ and 7. Note that no change in throughput between brokers $5, 6, 7$ happened (represented by the same thickness of edges).

Some brokers joined one of the groups, e.g. broker 4 joined the group G_1, but some of them left their group, e.g. $1, 3$ and 6. An important fact to comment on is that joining or leaving a group G is not conditioned by a direct connection between members of the group G. It comes from the fact that members within a group may interchange notifications using a low-level network protocols and do not depend on an application-level interaction that is required e.g. in Siena [5].

3 Conclusions and Future Work

The goal of this paper was to propose a compromise publish/subscribe scheme approach having advantages of both widely used schemes. The main principle of this approach is to maintain local groups of brokers on the basis of a heuristics, in order to increase a data transfer performance among the members within these groups. The cost of the approach is a higher implementation complexity according to the basic schemes. The future work related to the hybrid publish/subscribe will be concentrated on empirical methods to prove correctness of the concept.

References

1. P. Eugster, P. Felber, R. Guerraoui, A. Kermarrec. *The Many Faces of Publish/Subscribe.* Technical Report, Micosoft Research Cambridge, UK, 2001.
2. Sun Microsystems: *Java Message Service^{TM}.* JMS (version 1.0.2b), June, 2001. http://java.sun.com/products/jms/
3. M. Aguilera, R. Strom, et al. Matching Events in a Content-based Subscription System. In *Proceedings of the ACM Symposium on Principles of Distributed Computing, pp. 53-61*, 1999. http://www.research.ibm.com/gryphon
4. G. Banavar, et al. An Efficient Multicast Protocol for Content-Based Publish-Subscribe Systems. In *Proceedings of the 19th Int. Conference on Distributed Computing Systems*, MIT, Boston, 1999. http://www.research.ibm.com/gryphon
5. A. Carzaniga. *Architectures for an Event Notification Service Scalable to Wide-area Networks.* PhD Thesis, Technical University in Milano, Italy, 1998.