Local On-Line Maintenance of Scalable Pub/Sub Infrastructure

Alexander Shraer* Gregory Chockler† Idit Keidar* Roie Melamed† Yoav Tock†
Roman Vitenberg‡

1 Introduction

A publish-subscribe infrastructure [3] supports the dissemination of varied information to a large population of users with individual preferences. In this paper we focus on topic-based publish-subscribe systems, where users can subscribe to topics of interest, and receive all updates related to these topics. One notable application of such systems is real-time dissemination of trading data to stock brokers. Another example is a modern data center offering a large variety of application services that are accessed through the Internet. In such environments, the individual applications are typically replicated for performance and availability thereby creating overlapping multicast domains each of which is mapped into a separate pub/sub topic.

Modern publish-subscribe applications have stringent scalability requirement. Information is often published by multiple sources, and is disseminated to potentially tens or even hundreds of thousands of users, who may be geographically dispersed. Moreover, the information is categorized according to a rich collection of topics – possibly in the order of tens of thousands – and users typically subscribe to many topics. In order to deal with the huge number of users as well as with the system's geographical scale, nodes are organized in a logical structure, e.g., a hierarchical [8] or other overlay organization [5, 2], where each node has a small number of neighbors.

The main challenge in adapting existing overlay network technology for publish-subscribe applications is coping with the immense number of topics. A solution whereby a separate overlay network is used per topic might not scale for a setting with tens of thousands of topics, as each node is required to maintain a number of connections proportional to the number of topics it is interested in. To deal with this problem we introduced the SpiderCast [2] protocol that aims to create a single overlay network where the nodes with overlapping subscriptions are clustered together using additional "interest-aware" links (i.e., links based on the partial subscription views available to each node). As we show in [2], local coverage-optimizing heuristics can be effective to reduce the average node degree while ensuring that nodes with overlapping interests are well-connected.

Although SpiderCast significantly reduces the number of links per user, it does not solve all the difficulties stemming from the vast number of topics. For example, the difficulty of ensuring reliable message delivery in each group of users subscribed to a topic. The memory and processing requirements of managing reliability for each topic separately would be prohibitive. It is therefore common to employ aggregation of topics into channels [8, 9], which is akin to the use of lightweight groups in early group communication systems [4]. Of course, the use of a single group for all topics is not feasible, as clients have limited resources (e.g., bandwidth), and cannot filter out thousands of irrelevant topics. Using aggregation can balance the two needs, and achieve scalability in the number of managed topics without excessive filtering [8, 9]. This paper focuses on such aggregation, which is complementary to the techniques used in SpiderCast. The output of the aggregation algorithm can then be used to either construct a separate overlay spanning the nodes subscribed to each cluster, or serve as the subscription input to the SpiderCast protocol.

In order to be efficient, aggregation must take the overlap of interests into account, so as to minimize the amount of filtering at clients (or at brokers or proxies, if these are employed). Moreover, topic aggregation should dynamically evolve in response to the change in subscribers' interests, which is not addressed by existing solutions, none of which is adaptive. For example, in QSM [8] scalability is achieved by organizing subscribers into "regions", in which their interests are shared by other subscribers. However, the assignment of topics to regions do not change at run time. Another example is the work of Tock et al. [9], which introduced a centralized offline algorithm for computing the assignment of topics to clusters such that the overall filtering induced in the system by subscribing to these clusters is minimized.

*Department of Electrical Engineering, Technion, Haifa, Israel. {shralex@tx, idish@ee}.technion.ac.il

†IBM Haifa Research Labs, Haifa, Israel. {chockler, roiem, tock}@il.ibm.com

‡Department of Informatics, University of Oslo, Norway. roman.vitenberg@gmail.com

This position paper advocates the idea of adapting topic clustering (and corresponding overlays) dynamically according to changing needs, while preserving some level of optimality. That is, a change in subscription to a handful of topics should not necessitate computing anew all the topic aggregations, and should be confined to a small number of overlays/clusters, i.e., incremental solutions are of essence. Moreover, to achieve scalability, the optimization problem should be solved in a distributed manner and should not rely on global knowledge of subscriptions.

2 Local Algorithms

We believe that a scalable solution can be achieved using local computations, whereby the algorithm reaches its decision and quiesces locally at each node, without communicating with the entire system. Locality has recently emerged as a promising approach in order to provide scalable solutions of other problems [10, 6, 7, 1]. For example, in our context, one cannot allow every subscription change to disrupt 50,000 nodes. But are local computations feasible? We believe that most of the time they are. Intuitively, local computation of the assignment of topics into clusters should be possible if the statistics regarding the overlap of interests are similar in different areas of the system, and hence local decisions made independently in different areas may well be the same, eliminating the need for global communication. Likewise, changes that do not significantly impact the overlap statistics should have little impact on the assignment, and accordingly, induce little communication.

Local solutions where shown to be feasible in related problems such as majority voting [10], where there was little communication if all participants had the same votes or if votes changed insignificantly. We plan to investigate the feasibility of local solutions for our problem using typical pub/sub workloads.

3 Efficient Optimistic Reconfiguration

Subscriptions and membership continuously evolve, thus the solution should be *anytime*, i.e., continuously output a result which improves over time, and stabilize if the system is stable long enough.

Dynamically changing the assignment of topics to overlays requires coordination among all the involved entities. Scalable mechanisms like overlay-based or gossip-based dissemination of aggregation decisions should be preferred over unscalable consistency mechanisms, such as consensus. One difficulty that arises is that such a solution allows users to be temporarily in disagreement regarding which overlay serves a given topic. A second difficulty that stems from the first are possible fluctuations in the assignment of topics to clusters. Moreover, the system should provide continued service while the topology is reconfigured. For this purpose, a make-before-break policy can be employed, i.e., keeping both old and new topologies alive for a while, and adding an out-of-band mechanism like system-wide gossip for detecting message losses and inconsistencies, and requesting explicit recoveries.

4 Conclusions

We advocate the use of distributed topic clustering to achieve scalability in pub/sub systems. Such systems should be "subscription-aware", i.e., reconfigure topic aggregation and possibly underlying overlay topology dynamically, in response to changes in subscriptions. Anytime local algorithms and gossip-based recovery should be used for such reconfigurations, and continued service should be provided while a reconfiguration is in progress.

References

[1] Y. Birk, I. Keidar, L. Liss, A. Schuster, and R. Wolff. Veracity radius: capturing the locality of distributed computations. In *PODC*, pages 102–111, 2006.

[2] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg. Spidercast: A scalable interest aware overlay for topic-based pub/sub communication. Technical report, IBM Haifa Research Labs, 2006.

[3] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.

[4] B. B. Glade, K. P. Birman, R. C. Cooper, and R. van Renesse. Light-weight process groups. In *OpenForum 92 Technical Conference*, Nov. 1992.

[5] R. W. Hall, A. Mathur, F. Jahanian, A. Prakash, and C. Rassmussen. Corona: a communication service for scalable, reliable group collaboration systems. In *CSCW*, pages 140–149, 1996.

[6] D. Krivitski, R. Wolff, and A. Schuster. A local facility location algorithm for sensor networks. *DCOSS*, 2005.

[7] F. Kuhn, T. Moscibroda, and R. Wattenhofer. On the locality of bounded growth. In *PODC*, pages 60–68, 2005.

[8] K. Ostrowski and K. Birman. Extensible web services architecture for notification in large-scale systems. In *ICWS*, 2006.

[9] Y. Tock, N. Naaman, A. Harpaz, and G. Gershinsky. Hierarchical clustering of message flows in a multicast data dissemination system. In *IASTED PDCS*, pages 320–326, 2005.

[10] R. Wolff and A. Schuster. Association rule mining in peer-to-peer systems. *ICDM*, 2003.