Gravity: An Interest-Aware Publish/Subscribe System Based on Structured Overlays

Sarunas Girdzijauskas *
EPFL, Switzerland

Gregory Chockler
IBM Research, Israel

Roie Melamed
IBM Research, Israel

Yoav Tock
IBM Research, Israel

1. INTRODUCTION

Publish/subscribe (pub/sub) is a popular communication middleware that allows users to subscribe to topics of interest, and then be notified of messages being posted on any of the topics in their subscriptions. Traditionally, most uses of pub/sub have been limited to building applications integrating multiple (possibly diverse) data sources, such as e.g., event processing engines, live event broadcast, RSS feed readers, etc. Recently, there has been a growing interest in applying pub/sub to support communication in an emerging class of applications involving fine-grained information sharing on massive scales [7, 8], such as e.g., on-line gaming, Internet chat rooms, Second Life, etc.

In order to adequately address the scaling needs of these applications, a pub/sub solution must be able to effectively deal with large populations of dynamic users, large numbers of topics, and arbitrary subscription patterns. In particular, many traditional solutions concentrate the message processing load in a few fixed system components (such as centralized servers, or fixed hierarchies thereof), and therefore, do not scale well as the system grows in size. As an alternative, several decentralized pub/sub implementations have been proposed (e.g., [2, 3, 4, 12]). In these implementations, the nodes are typically organized into an *overlay network*, whose links are then used to propagate published data.

Some of the proposed overlay-based implementations are quite effective in dealing with scaling issues (such as the number of nodes and geographical spread) thanks to the sub-linear degree and low diameter properties of their underlying communication graphs. However, due to possible lack of connectivity among the nodes sharing the same interests, the message dissemination overhead could potentially be quite high. One way to minimize this overhead is to maintain a separate communication graph for each topic with a non-empty set of subscribers. That however, results in a topology in which the nodes could no longer be guaranteed to maintain a fixed number of neighbors thus loosing one of the primary scalability advantages of the overlay-based solutions.

In this paper, we therefore, take a different approach. In particular, we introduce *Gravity*, an overlay-based pub/sub system that achieves communication efficiency while preserving fixed node degree by biasing the link creation process so that the nodes sharing similar interests are more likely to be closely connected (i.e., *clustered*). As in the prior work [4], Gravity is based on a ring-based structured overlay, and leverages the Distributed Hash Table (DHT) capabilities to construct distribution trees spanning the subscribers of each particular topic. However, unlike [4], in Gravity, the node placement on the ring is not uniform but rather determined by their subscriptions. More specifically, the Gravity's placement strategy allows the nodes to dynamically adjust their positions on the ring by moving closer to ("gravitating towards") the nodes with similar interests. This mechanism ensures that for well-correlated inputs, the nodes sharing similar interests would eventually end up clustered in a few contiguous regions on the ring.

We then leverage the resulting interest-based clusters to construct efficient dissemination trees for each topic, such that the nodes sharing similar interests end up being reachable through the same branches of a dissemination tree. As a result, we reduce the publication cost by amortizing the overhead of disseminating messages within the topic tree over the total number of nodes interested in that topic. Moreover, since the nodes within each region would likely share similar interests, we are able to amortize the overhead over multiple topics at once without increasing the degree budget.

The reduction in the message dissemination overhead achieved by Gravity depends on the degree of correlation exhibited by the individual node interests. To verify this claim, we used several synthetic workloads generated using models similar to those of [11] (see Section 3). We show that under these workloads, Gravity is indeed able to achieve significant reduction in the message dissemination cost, which for some realistic settings can be as high as 10-fold compared to the pub/sub systems based on DHTs with uniform node placement (such as [4]). Furthermore, we demonstrate that this cost reduction is *adaptive* to both the extent to which the individual node subscriptions correlate, and the amount of information about the other node subscriptions available to each node. In particular, we show that Gravity can exploit even slight subscription correlations among the peers, and in the worst case scenario, when the subscriptions are completely uncorrelated and/or unknown, the message dissemination cost would not be worse as that of the uniform DHT based systems.

However, while it could be reasonably expected that the users involved in large-scale on-line collaboration and data sharing activities would indeed share interests in a large number of logically-related data items (such as e.g., the cells within a jointly edited spreadsheet, the details of a shared scene in an on-line game, or the

*Supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

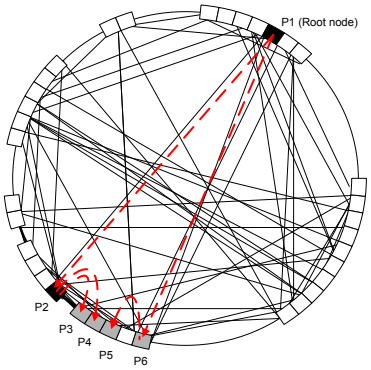

Figure 1: Effect of clustering on spanning trees in Gravity (black peers represent unsubscribed nodes, grey - subscribed)

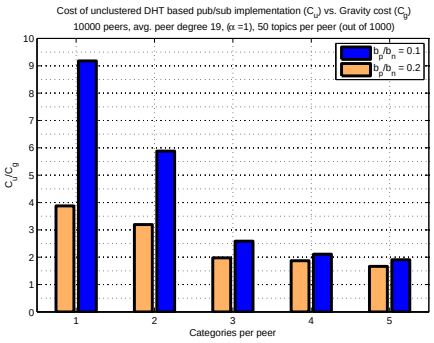

Figure 2: The impact of the interest correlation on the Gravity's performance

popular stock portfolios in an on-line trading service), it still remains to be seen to which extent this behavior is indeed typical of the real-world applications. The existing models mainly capture the popularity of individual topics (which in several prior studies, has been found to follow a power law), and are therefore, too coarse-grained to discern the correlation patterns arising among the actual subscriptions. Deriving more realistic subscription models using realistic data sets is the subject of our ongoing work.

2. IMPLEMENTATION OVERVIEW

In order to facilitate the construction of efficient distribution trees, the nodes in Gravity are dynamically organized into a small-world overlay topology. That is, in addition to the two ring links, each node is also maintaining a small (at most logarithmic) number of additional *long-range* pointers (or *fingers*). These fingers are created so as to ensure that in the resulting overlay, any two pair of nodes is connected by a routing path of a logarithmic length. This is accomplished using a Distributed Hash Table (DHT) construction protocol, similar to [1, 5], in which fingers are created adaptively, based on the actual distribution of the nodes on the ring[1].

The dissemination tree for each topic is then constructed by following the greedy routing paths in the overlay from a fixed root node (determined by uniformly hashing the topic's name into a node identifier) to each of the topic's subscribers. This ensures that the distribution trees preserve the small-world properties of the overlay. That is, the subscribers to each particular topic, sharing the same ring neighborhood will all end up being closely connected in the topic's distribution tree. In particular, in the ideal case, when all of the topic's subscribers are located close to each other on the ring (see Figure 1), the overhead of reaching all of them from the topic's root would roughly amount to the cost of reaching any subscriber within the range which is bounded (on expectation) by $O(\log n)$.

3. PERFORMANCE EVALUATION

We implemented Gravity in a simulated setting, and conducted extensive experimental study to validate our performance and scalability claims. The workloads for our experiments were synthesized using the subscription models that have been demonstrated in the prior work [6, 11] to be a truthful representation of correlated subscription patterns occurring in many real-world scenarios. The overall collection of topics was first partitioned into a fixed number b_n of fixed size categories, and then, every peer had to choose b_p categories u.a.r. out of b_n. Within each individual category the topics were assigned based on a power-law distribution. The degree of correlation between the peer interests can be adjusted by changing the b_p and b_n parameters while keeping the b_p/b_n ratio intact.

Figure 2 illustrates some of the Gravity's performance evaluation results. Specifically, we depict the impact of the interest correlation on the relative improvement in the message dissemination cost. As this figure shows, Gravity outperforms the implementations based on uniform DHTs even under very low subscription correlation.

[1]Note that since out placement strategy might result in non-uniform identifier distributions, we cannot use standard DHT protocols, such as e.g., [9, 10], in which the fingers are always created to point towards fixed locations on the ring.

4. REFERENCES

[1] A. Bharambe, M. Agrawal, and S. Seshan. Mercury: Supporting scalable multi-attribute range queries. In *ACM SIGCOMM, Portland, USA*, 2004.

[2] S. Bhola, R. Strom, S. Bagchi, Y. Zhao, and J. Auerbach. Exactly-once delivery in a content-based publish-subscribe system. In *DSN*, 2002.

[3] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, 2001.

[4] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron. SCRIBE: a large-scale and decentralized application-level multicast infrastructure. *IEEE J. Selected Areas in Comm. (JSAC)*, 20(8):1489–1499, 2002.

[5] S. Girdzijauskas, A. Datta, and K. Aberer. Oscar: Small-world overlay for realistic key distributions. In *DBISP2P 2006, Seoul, Korea*, 2006.

[6] H. Liu, V. Ramasubramanian, and E. G. Sirer. Client behavior and feed characteristics of rss, a publish-subscribe system for web micronews. In *Internet Measurement Conference (IMC), Berkeley*, October 2005.

[7] K. Ostrowski, K. Birman, and D. Dolev. Quicksilver scalable multicast. In *submission*.

[8] K. Ostrowski, K. Birman, and D. Dolev. Live distributed objects: Enabling the active web. In *IEEE Internet Computing, 11(6), p. 72*, 2007.

[9] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329–350, 2001.

[10] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *SIGCOMM*, pages 149–160, 2001.

[11] Y. Tock, N. Naaman, A. Harpaz, and G. Gershinsky. Hierarchical clustering of message flows in a multicast data dissemination system. In *17th IASTED Int'l Conf. Parallel and Distributed Computing and Systems*, pages 320–327, 2005.

[12] S. Voulgaris, E. Riviere, A.-M. Kermarrec, and M. van Steen. Sub-2-sub: Self-organizing content-based publish subscribe for dynamic large scale collaborative networks. In *IPTPS*, 2006.