CAPS: Content-bAsed Publish/Subscribe services for peer-to-peer systems[*]

Jordi Pujol Ahulló[‡]
jordi.pujol@urv.cat

Pedro García López[‡]
pedro.garcia@urv.cat

Antonio F. Gómez Skarmeta[§]
skarmeta@um.es

[‡]Universitat Rovira i Virgili
Av. Països Catalans, 26
43007 - Tarragona, Spain

[§]Universidad de Murcia
Campus Universitario de Espinardo
30100 - Murcia, Spain

ABSTRACT

In this paper we introduce a content-based publish/subscribe (pub/sub) system that leverages the peer-to-peer substrate. Thus, we avoid to build a specific overlay for the pub/sub system and use the rendezvous model to meet both events and subscribers. On the contrary to what could be expected, our system suits for high-dimensional pub/sub domains, requiring very low memory capacity and hops to run subscription and event notification processes. We outline its main properties and some simulation results.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed applications

Keywords

Data Distribution, Publish/Subscribe

1. INTRODUCTION

Peer-to-peer systems have received a lot of attention recently. This kind of systems are unmanaged and all participants cooperate in order to maintain the network structure and the services they provide. Particularly, structured peer-to-peer networks, also called distributed hash tables (DHTs) [4, 3], provide logarithmic communication cost between any pair of nodes, keeping a partial little view of the network for routing purposes. Nevertheless, pub/sub systems (e.g. Scribe [2] or Bayeoux [6]) built atop of these peer-to-peer networks incur on additional costs (like the pub/sub overlay maintenance, node churning, pub/sub links resolution) even when the sructured peer-to-peer network architecture is also maintained. In this paper we are interested in pub/sub systems that support *subscriptions* which define their particular interests, expressing conditions on the content of events (*content-based* model), rather than just on a category they belong to (*topic-based* model).

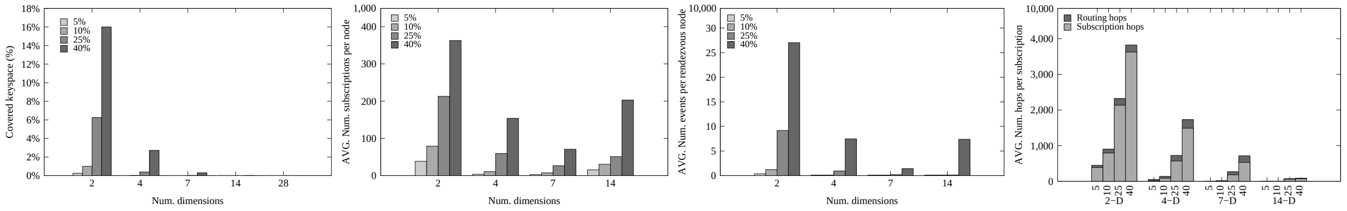
DHTs were introduced as scalable data structures for building large distributed applications. Peter Triantafillou *et al.* [5] introduced one of the first content-based approximations where Chord DHT is employed as reliable routing infrastructure, so that they do not build a specific pub/sub overlay. To do so, they employ the *rendezvous model.* The motivation behind that is because the multihop routing abstraction implemented by DHTs integrates naturally with the need for globally unique rendezvous nodes in these rendezvous-based routing approaches. Nevertheless, as long as this kind of approaches performs a DHT communication *for each event's attribute* mapping (as applied in [5]), such systems suffer lack of scalability on high-dimensional contexts. Later, Baldoni *et al.* [1] perform a similar approach but, in this case, they perform a particular mapping of events and subscriptions to keys from the DHT keyspace, instead of per-attribute mappings. Even though their approximation is quite interesting, their system requires some sort of *primitive multicast* provided by the DHT in order to obtain good performance. In summary, thus, the design of event dissemination in content-based pub/sub systems working onto DHTs has to take into account different factors:

• *Lighweight and portable proposal.* The goal behind the idea of building the pub/sub system onto DHTs is twofold: a) leverage the DHT routing infrastructure and, thus, avoiding to build a pub/sub overlay protocol over an existing DHT, and b) operate suitably onto (most of) current DHTs, without requiring ad-hoc functionalities to the DHT. Nevertheless, important properties like load balancing and low local state information maintenance must be retained.

• *Multiple sources.* Nodes cooperating in this distributed pub/sub system should have guarantees for publishing and subscribing at any time and concurrently.

• *Multi-attribute data.* Usually content-based systems support applications whose information is defined in terms of different parameters or attributes. Therefore, events contain a value per attribute, and subscriptions specify range of values of interest per attribute. This way, the system needs some mechanism to route multi-dimensional events and subscriptions throughout the network, while DHTs op-

[*]This work has been partially supported by project P2PGRID, TIN2007-68050-C03-03, and FPU National Program, AP-2006-04166, both funded by the Spanish Ministry of Education and Science.



(a) Estimated keyspace coverage by range selection mapping.

(b) Average number of stored subscriptions per node.

(c) Average number of events that reaches rendezvous nodes.

(d) Average number of hops performed per subscription.

Figure 1: First CAPS' results. Set up for (b)(c)(d): 10K-node networks, 1K subscriptions, 1K events.

erates with one-dimensional keyspaces.

As seen before, current systems (e.g. [5, 1]) do not deal with all the factors above described. For this reason, we introduce in this paper a novel system, called CAPS, that builds *content-based event dissemination infrastructures* onto DHTs in an efficient way. To do so, we employ the *rendezvous model* in order to meet both events and subscriptions. For this reason, the system defines a certain set of nodes from the DHT as *rendezvous nodes*, being responsible of matching events against subscriptions and start then the notification process. Additionally, these rendezvous nodes are selected deterministically, so that the node in DHT responsible for a given key then becomes the rendezvous node. Due to the DHT properties, the chosen node will be globally agreed upon by all nodes and, this way, every node can use the peer-to-peer routing substrate to send messages to this rendezvous node.

2. CAPS APPROACH

Any system disseminating information using CAPS will retrieve the following key benefits:

• The *rendezvous model* enables the system to avoid the construction of a specific overlay to disseminate events in a proper way. In fact, CAPS leverages the DHT routing properties to set rendezvous nodes every time, therefore achieving a **lightweight pub/sub system**. Unlike other pub/sub systems, CAPS does not need advertisements to meet both events with subscriptions, proportioning even a more lightweight solution. We also design *DHT-generic* subscription and notification algorithms that allow CAPS to work onto different DHTs, making the whole solution **portable**.

• CAPS employs an order preserving hash function (OPHF) to map conjunctive predicates from every *subscription* into *a set of keys* and every *event* into *a key*, deterministically, in order to deal naturally with **multi-dimensional domains**, and **multiple sources** cooperating within the system.

• Results demonstrate that CAPS has better performance on high-dimensional contexts, requiring low memory capacity and hops to perform event disseminations and subscriptions, with a *wide range of selectivity ratios* -which define the ranges of interest of subscriptions against events.

Fig. 1 depicts some of the preliminary results. We have simulated CAPS onto Chord, with a node identifier (ID) size of 28 bits. Note that bigger ID size means support to higger dimensionality and, consequently, better system performance as we can see below. Fig. 1(a) depicts the expected behavior of CAPS subscription mapping, which is closely related to the number of nodes where a given subscription will be installed. This behavior is followed by simulation results from Fig. 1(b) for subscription installation, and from Fig. 1(c) for event reception at rendezvous nodes. The routing cost, in particular for subscription installation (see Fig. 1(d)), is near optimal as long as the overhead appended (i.e. routing hops) to the number of nodes where subscription must be installed (i.e. subscription hops) is almost constant and insignificant. These first results demonstrate that our system is *efficient both in terms of communication cost and memory usage*, selecting very few nodes as rendezvous nodes. The results also show that CAPS provides its best performance on low selectivity ratios for any dimensionality and on *high-dimensional applications*. Additionally, with this paper, we aim to demonstrate that CAPS *is feasible, scales to big peer-to-peer networks and balances the load through the network*. One aspect does not treated is the effect in our system of nodes joining and leaving the network. This will be one point of our future work.

3. REFERENCES

[1] R. Baldoni, C. Marchetti, A. Virgillito, and R. Vitenberg. Content-based publish-subscribe over structured overlay networks. In *Proc. ICDCS'05*, pages 437–446, 2005.

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

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *Proc. SIGCOMM '01*, pages 161–172, New York, NY, USA, 2001. ACM Press.

[4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. SIGCOMM '01*, pages 149–160, 2001.

[5] P. Triantafillou and I. Aekaterinidis. Content-based publish-subscribe over structured p2p networks. In *Proc. DEBS'04*, 2004.

[6] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. D. Kubiatowicz. Bayeux: an architecture for scalable and fault-tolerant wide-area data dissemination. In *Proc. NOSSDAV'01*, pages 11–20. ACM, June 2001.