A novel adaptive content-based subscription management system

Emiliano Casalicchio, Federico Morabito, Giovanni Cortese, Fabrizio Davide

2005
A novel adaptive content-based subscription management system

Emiliano Casalicchio,*
Federico Morabito, Giovanni Cortese, Fabrizio Davide†

October 20, 2005

Abstract

Content-based Publish/Subscribe (CBPS) interaction paradigm is suitable for a variety of large scale dynamic applications: news delivery, stock quoting, air and metropolitan traffic control, on-line games, dissemination of multimedia contents, dissemination of auction bids, services and resources discovery, remote control of critical infrastructures and management of large scale systems. In contrast to their flexibility and expressiveness scalable CBPS systems are difficult to implement and the proposed solutions are not again mature. In this paper we propose a novel adaptive content-based subscription management system, relying on a Distributed Hash Table routing infrastructure. We define a model for the event space guaranteeing the expressiveness for any application domain. Also we provide mechanism to dynamically identify groups of users with similar preferences (multicast group). Two different approaches for multicast group identification are proposed and discussed: a dummy regular grid partitioning of the event space and a more complex clustering of users preferences.

1 Introduction

The publish/subscribe interaction paradigm provides subscribers with the ability to express their interest in an event or a pattern of events, in order to be notified subsequently of any event, generated by a publisher, that matches their registered interest. The different ways of specifying the events of interest have led to several subscription schemes, the channel-based, the topic-based (or

*University of Roma ”Tor Vergata”, Computer Science Department, casalicchio@ing.uniroma2.it
†Telecom Italia Learning Services R&D Department, federico.morabito@telecomitalia.it, fabrizio.davide@telecomitalia.it, g.cortese@computer.org
subject-based), and the content-based. Topic-based systems, an evolution and refinement of channel-based systems [3], was widely studied in literature and are nowadays mature. These systems evolve from a centralized implementation [8, 2, 12] to a distributed implementation. The latter can rely on different delivery infrastructures: an Event Brokering System is used in [4] and [18]; peer-to-peer overlay networks using efficient routing infrastructure, such as Distributed Hash Table are used in [6, 1]. In topic based publish/Subscribe, users express their interest by joining a group (or topic). Then all messages related to that topic are broadcasted to all users participating to the specific group. Content-based systems are relatively new and give users the ability to express their interest by specifying predicates. Publications are matched to submissions on the basis of their content. The content-based publish/subscribe paradigm is more powerful than topic based and is able to support rich subscription languages, by introducing a subscription scheme based on the actual content of the considered events. In other terms, events are not classified according to some pre-defined external criterion (e.g., topic name), but according to the properties of the events themselves.

In contrast to their advantages, scalable CBPS systems are difficult to implement. Existing solutions are not again mature and use different approaches: Gryphon [11] uses a message brokering system, SIENA [5] uses both hierarchical server and peer-to-peer implementation, Hermes [14, 13] and the solution proposed by Terpstra et al. [17] use peer-to-peer infrastructures and DHT-based routing. The main issues in design content-based publish/subscribe systems are the following:

- to maximize the expressiveness [5].
- To guarantee the scalability with respect to the number of subscriber, the published events and size of the event space.
- To balance the load among peers in the system.
- To adapt the network topology and the groups of common interest subscriptions to the dynamics of the system thus to maintain the desired level of performances.

In this paper, we propose an approach that explicitly addresses the system adaptability problem and that presents a trade-off solution between the first two problems.

We propose two methods to automatically identify multicast groups: the first based on a dummy regular grid partitioning and the second based on a clustering algorithm.

The reorganization is based on run time measurement of system efficiency.
2 Problem description

As mentioned in Section 1, our solution aims to realize a system with an high degree of expressiveness, guaranteeing the scalability with respect to the number of subscribers, publishers and with respect the size of the event space and adapting the network topology and the groups of common interest subscriptions to the dynamics of the system. The expressiveness is addressed proposing a flexible application domain data model that can be applied to any specific application domain.

Limiting the number of multicast groups is one of the main characteristic of the proposed data model, exploiting the intrinsic scalability provided by the application layer multicast (DHT-based) to solve the problem.

The adaptivity of our system is guaranteed by a mechanism that automatically evaluates the efficiency of the actual configuration and starts the multicast groups reconfiguration process.

2.1 Problem Notation

Here we introduce specific notation we will use in the rest of the paper.

- Let $N$ denote the number of attributes of the application domain schema.
- Let $\Omega \subseteq \mathbb{R}^N$ define the event space (an $N$-dimensional cartesian space).
- Each event being published within the system can be uniquely described with a single value $\omega$ such that $\omega \in \Omega$.
- Let $V$ be the set of nodes of the network.
- Let $V_S \subseteq V$ be the set of nodes hosting subscribers. $|V_S| = N_S$
- Let us assume that each subscriber $v_i \in V_S; i = 1 \ldots N_S$, has a set of $r_i$ subscription preference expressed by $I_i = \{b_{ij}\}_{j=1}^{r_i}$. Each $b_{ij} \in \Omega$ is an aligned rectangle in space $\Omega$.
- Let us define $I = \bigcup_{v_i \in V_S, i=1 \ldots N_S} I_i$ to be the set including all subscription rectangles.

3 Application domain data model

For each application domain, the publishers and subscribers use a data model representation based on a set of application dependent attributes. We indicate this set of attributes as application schema (or shortly schema)\(^1\)

\(^1\)We remark that our notion of schema is different from the concept introduced by Tam et al. in [16]
Attributes are characterized by their type, name, and constraints on possible values, specifying the general format of data and their possible values (within each application domain). Each application domain has its own schema, thus multiple domain schemas can be handled simultaneously by the same application (and also different applications may run on the same network). For example, a complex system management tool may handle simultaneously three domain: a distributed system management domain, a power grid monitoring domain and resource discovery domain. The proposed application domain data model allows subscribers to specify the subscription preferences, indicating the attributes and the related range values; and it allows publishers to disseminate events to the interested users. The proposed data model, inspired by [15], permits to represent the application domain through an N-dimensional cartesian space \( \Omega \) in which each event can be uniquely described with a single multidimensional element \( \omega \) such that \( \omega \in \Omega \). The axes of the event space represent the attributes and they are labeled with the name field contained in the schema, while the ranges of the axes are specified into the values field. In our solution, both the attribute name field and attribute values field are relevant to organize the subscription in multicast groups to properly disseminate the generated events. Using the proposed abstraction, each subscription preference of the form \{ [name attribute 1, range value 1], [name attribute 2, range value 2], [name N attribute, range value N] \} is represented as an hyper-rectangle into the event space.

Synthesizing, the main purposes of using the proposed application schema are to provide an high level of expressiveness in any application domain and to map the application domain into a N-dimensional cartesian space, manageable by the clustering and the partitioning algorithms.

4 System Architecture

Our solution is based on a two layers architecture. On the top of the transport layer we build a DHT-based multicast application layer based on Scribe[6] and Pastry[7]. Thus, the DHT multicast layer embeds the application layer multicast protocol and the DHT-based overlay network, providing to the upper layer, the overlay network management primitives (e.g. join(), subscribe(), unsubscribe(), lookup(), route()) and the multicast primitives (e.g. join(), create(), leave(), multicast()).

The content-based pub/sub functionalities are implemented by the following three processes:

Multicast Group Identification (MGI) process that associates a subscription \( h_{i,j} \), to one or more multicast trees;

Multicast Group Matching (MGM) process that identifies the multicast groups to which deliver an event \( \omega \).
Multicast Group Creation (MGC) process that re-organizes the multicast groups when the system state change compromising the efficiency.

In the following, we describe two possible approaches to create the multicast groups, characterizing the different implementation of the above processes in the two approaches.

4.1 Regular partitioning solution

This solution is based on the regular grid partitioning of the event space $\Omega$. The event space $\Omega$ is partitioned in cells $c(i, j, ..., k)$ of granularity $(\alpha_1, \alpha_2, ..., \alpha_N)$, where $\alpha_i$ represents the step of the cell along the axes $i$, and $i, j, k \in [1, N]$. Each cell is represented by a centroid $cen(i, j, ..., k) = (\alpha_1 \cdot i, \alpha_2 \cdot j, ..., \alpha_k \cdot k)$ equal to the center of the cell. As already mentioned, the regular partitioning solution associates a multicast group $MG(i, j, ..., k)$ to each cell $c(i, j, ..., k)$. The nodes in a multicast group $MG(i, j, ..., k)$ are organized in a multicast tree $MTree(i, j, ..., k)$, with root $cen(i, j, ..., k)$.

The multicast tree $MTree(i, j, ..., k)$ is responsible for all the published events and the subscribers with preferences intersecting the cell $c(i, j, ..., k)$. In the regular partitioning solution multicast groups are identified evaluating the intersections among the hyper-rectangle $b_{h,l}$ expressing the $l^{th}$ preference of subscriber $v_h$ and the cells of the events space. All the intercepted cells determine the multicast groups to which can be associated the subscriber $v_i$. Different heuristics can be used to trade-off the efficiency and the infrastructure management cost when the number of intercepted cell is too high. For example, could be selected the cells that have an intercepting volume greater than a predefined threshold or the multicast groups with the greatest intercepting volume. Every time an event $\omega$ is published, the Multicast Group Matching process looks for the cell containing the event $\omega$ and then the event $\omega$ will be delivered through the associated multicast tree to the interested subscribers. To speed up the search process, the partitions of $\Omega$ are organized in a R-Tree [9] data structure. R-Tree is a dynamic indexing structure for spatial search allowing logarithmic searching with the numbers of clusters. The Multicast Group Creation process is responsible to evaluate the inefficiency of the actual configuration of the multicast groups. This process, periodically, reorganizes the multicast groups considering the dynamics of the subscriptions and the pattern of published events.

4.1.1 Multicast Group Identification process

When a new node joins the overlay network, it receives from the bootstrap node the access to the network, (by the nodeId and the routing tables), the application schema, that contains the list of all the attributes and their domains (thus giving the possibility to the node to express its own preferences and to publish the events) and the information about the grid granularity, to permit locally the node to compute the partitioning of the event space $\Omega$. The MGI process
evaluates the interception among the subscription $b_{i,j}$ and the $\Omega$’s partitions and assigns the node to one or more multicast groups. We can describe the process by the following steps:

1. Determine the set $MG(b_{i,j})$ of the cells intercepting the subscription $b_{i,j}$.

   $$MG(b_{i,j}) = \{c(h,k,l) | b_{i,j} \cap c(h,k,l) \neq \emptyset \forall h,k,l \in [1,N]\}$$

2. Determine the set of centroids $CEN(b_{i,j})$ of the cells in the set $MG(b_{i,j})$.

   $$CEN(b_{i,j}) = \{cen(h,k,l) | c(h,k,l) \in MG(b_{i,j}) \forall h,k,l \in [1,N]\}$$

3. Determine the set of multicast trees $MT(b_{i,j})$.

   $$MT(b_{i,j}) = \{MTree((h,k,\cdot,\cdot,l)) | cen(h,k,\cdot,\cdot,l) \in CEN(b_{i,j}) \forall h,k,l \in [1,N]\}$$

4.1.2 Multicast Group Matching process

The information received by a node when joins the network (e.g., the application schema and the grid granularity) permit the MGM process to be locally solved by the publisher node, with the following steps:

1. Determine the centroid $cen(h,l,m)$ of the cell $c(h,l,m)$ containing the event $\omega$, that is:

   $$c(i,j,\cdot,\cdot,k) = \{c(h,l,m) | \{\omega\} \cap c(h,l,m) \neq \emptyset \forall h,l,m \in [1,N]\}$$

2. Deliver the event $\omega$ to the appropriate multicast tree $MTree(h,l,\cdot,\cdot,m)$

4.1.3 Multicast Group Creation process

The Multicast Group Creation process is responsible to evaluate the inefficiency of the actual configuration with respect to the pattern of subscriptions and the number of created multicast groups and to re-organize the multicast groups and the associated multicast trees. As final result of the re-organization, the subscribers will be associated to more efficient multicast groups.

We use the number of false positive to evaluate the inefficiency of the grid granularity. This performance metric gives an approximated measure of the duplicate and useless messages with respect to the actual pattern of subscriptions and the multicast groups composition. The false positive number for the
subscription $b_{i,j}$ with respect to the grid granularity $(\alpha_1, \alpha_2, ..., \alpha_N)$ is defined as

$$FP(b_{i,j})|_{(\alpha_1, \alpha_2, ..., \alpha_N)} = \sum_{(h,k,l), b_{i,j} \cap c(h,k,l) \neq 0} \frac{\alpha_1 \cdot \alpha_2 \cdot ... \cdot \alpha_k - b_{i,j} \cap c(h,k,l)}{\alpha_1 \cdot \alpha_2 \cdot ... \cdot \alpha_N}$$

where $\alpha_1 \cdot \alpha_2 \cdot ... \cdot \alpha_N$ is the volume of the cell.

$FP(b_{i,j})|_{(\alpha_1, \alpha_2, ..., \alpha_N)}$ is computed locally on each node. The number of false positive overall the entire network, i.e. the system inefficiency is defined as

$$FP(I)|_{(\alpha_1, \alpha_2, ..., \alpha_N)} = \sum_{i \in V, j \in [1,r_i]} FP(b_{i,j})|_{(\alpha_1, \alpha_2, ..., \alpha_N)}.$$  

Using the overall system inefficiency, the process determines the new granularity, re-computes the inefficiency, evaluates the gain and cost of the multicast groups and grid reconfiguration and starts the reorganization of the multicast groups.

### 4.2 Clustering based solution

In this solution, we group dynamically the different pattern of subscriptions into clusters, using the Minimum Spanning Tree [10] algorithm for clustering and the Euclidean distance between centroids of hyper-rectangles as distance function to be minimized. The set $I \subset \Omega$ of all subscriptions is partitioned into clusters of different volume. Each cluster $c_i$ is a rectangle in the $\Omega$ space of centroid $cen(i) = (x_1, x_2, ..., x_N)$, where $x_j, j = 1..N$ are the coordinates of the centroid. Note that in the clustering-based solution, the coordinates of the centroid are not fixed as in the regular partitioning solution, but they dynamically change depending on the the pattern of subscriptions. As well as in the partitioning solution, also the clustering based solution associates a multicast group $MG(i)$ to each cluster $c_i$ and then the nodes in the multicast group $MG(i)$ are organized in the multicast tree $MTree(i)$ where the centroid $cen(i)$ is the root. Also in the clustering based solution, the clusters are organized on a R-Tree data structure.

#### 4.2.1 Multicast Group Identification process

The Multicast Group Identification problem is solved by evaluating the intersections among the hyper-rectangle $b_{i,j}$, expressing the preference $j$ of subscriber $v_i$, and the dynamic clusters $c_i$ in the events space. The intercepted clusters determine the multicast groups to which the subscriber $v_i$ will be associated. The MGI is very similar to that in the case of grid-partitioning, with the differences that the information about the clusters is contained and represented in the R-Tree data structure (and not in the grid granularity) and that the clusters $c_i$ are
evaluated minimizing the distance function among hyper-rectangle expressing
the subscriptions and applying the Minimum Spanning Tree algorithm.

4.2.2 Multicast Group Matching process
Also in the clustering based solution, the node can resolve locally the MGM
process, having all the information needed to compute the interception between
the generated event $\omega$ and the clusters $c_k$.

More formally, we can formulate the MGM process by the following steps:

1. Determine the cluster $c_k$ containing the event $\omega$, that is:
   
   $$ c_k = \{ c_l | \omega \cap c_l \neq \emptyset \forall l \}.$$  

2. Determine the centroid $cen(k)$ of the cluster $c_k$.

3. If the multicast tree $MTree(k)$ exists, deliver $\omega$ to $MTree(k)$, else drop
   the event $\omega$.

4.2.3 Multicast Group Creation process
In this solution, the inefficiency of the clustering on the set $I$ depends on the
new subscriptions not just contained in the existing clusters. For example, a
subscription not completely contained in a cluster would be a source of false
negatives, while a subscription that intercepts more than one cluster would
be a source of duplicated messages (false positives). We use the number of false
negatives as performance metric considering not acceptable that some event
could be lost and that some subscribers will not receive events they are looking
for. The false negatives number for the subscription $b_{i,j}$ with respect to the
actual set of $K$ clusters $\{c_1, c_2, \ldots, c_K\}$ is defined as

$$ FN(b_{i,j})|_{\{c_1, \ldots, c_K\}} = 1 - \frac{vol(\Omega - \{c_1, \ldots, c_K\}) - vol(b_{i,j} - b_{i,j} \cap \{c_1, \ldots, c_K\})}{vol(\Omega - \{c_1, \ldots, c_K\})}, $$

where $vol(\cdot)$ represents the volume of the hyper-rectangle. $FN(b_{i,j})|_{\{c_1, \ldots, c_K\}}$
can be locally computed on each node, having the information about the actual
clusters. The number of false negatives over the entire network, i.e. the system inefficiency is defined as

$$ FN(I)|_{\{c_1, \ldots, c_K\}} = \sum_{i \in V_{S}, j \in [1,r]} FN(b_{i,j})|_{\{c_1, \ldots, c_K\}}, $$

and is computed by some specialized nodes responsible to collect the local inefficiency. The algorithm determines the new clusters, re-computes the inefficiency, evaluates the gain and cost of the grid reconfiguration and multicast groups adjustment.
5 Concluding remarks

The proposed approach to adaptive content-based subscription management is actually a work in progress.

We propose the guidelines to solve the main emerging problems. The novelty of our approach, respect the existing solutions, is in the adaptability of the systems to the offered load, and the capability to specify a desired level of efficiency and cost constraints.

Another advantage is the scalability, obtained limiting the number of multicast groups and using application level multicast. Obviously this solution reduces the expressiveness, because the data model represents the event space through a finite set of attributes.

As mentioned before, the load balancing among the peers and the realization of a solution that does not need specialized servers, still remain open problems. Indeed ongoing works are aimed to realize a fully distributed solution for the computation of the system inefficiency and for the multicast group reorganization.

References


