Increasing the Expressiveness of Subscriptions and Advertisements in Distributed Content-Based Publish/Subscribe Systems

Sven Bittner*

Department of Computer Science
The University of Waikato, New Zealand
{s.bittner}@cs.waikato.ac.nz

Abstract. This paper gives a synopsis of the author's PhD project. In order to address a broad range of conference participants, we have specifically decided to give a general, rather non-technical outline of the tackled research problems. This decision is reflected in the overall structure of this paper.
After introducing the general research area and the context of the project, we outline the three research problems that are addressed within the associated dissertation. We then give a conceptional overview about the research that has already been carried out but also about the remaining steps that need to be undertaken. We conclude this paper by stepping back from the actual dissertation and relating its contributions to the broader research community.

1 Introduction

Within the last years, there has been an increasing academic interest in the publish/subscribe (pub/sub) communication paradigm. One reason for the growing popularity of this subject is the wide applicability of the general pub/sub approach, ranging from the low-level monitoring of distributed systems to high-level electronic commerce areas [11]. Another reason for its success is the asynchronous, loosely-coupled nature of pub/sub, allowing for the implementation of dynamic, event-driven, and thus independent application components.

The overall idea of pub/sub is straightforward and as follows: Pub/sub systems have two kinds of users, publishers and subscribers. *Publishers* provide information to the system, using *event messages*. Before this information can be published, publishers have to specify their future messages by the help of *advertisements*. *Subscribers*, conversely, are interested in information and specify their interests using *subscriptions*. The pub/sub system decouples the communication of these users and *notifies* subscribers whenever an incoming message *matches* their subscriptions. We have given an overview of these two user groups and their communication directives in Fig. 1.

The granularity of the concept of matching, i.e., a subscription specifies an interest in incoming event messages, divides pub/sub into topic-based and content-based systems. Early research focused on the former concept, and supposed event messages to

* The author has been partially funded by the New Zealand Government under the New Zealand International Doctoral Research Scholarships (NZIDRS) scheme.

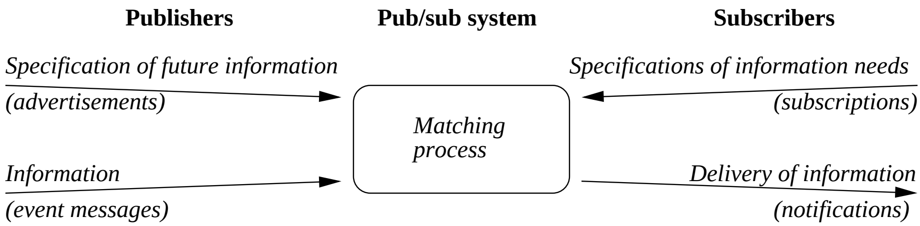

Fig. 1. Schematic overview of users and their communication directives in pub/sub

specify a topic and subscriptions to state an interest in topics. The subscriber thus receives notifications about all messages of the specified topic. A representative of this approach is newsgroups.

Current research allows for a more sophisticated concept of matching based on the actual content of messages (e.g., GRYPHON [1] and SIENA [10]). Within this paper, we assume event messages to state an *event type* and a set of *attribute-value pairs* according to this type. Thereby, the type and these pairs represent the content of this message. Subscriptions contain a *filter expression* for event messages, restricting the values of attributes. Hence, subscriptions specify the content a subscriber is interested in, leading to the term content-based pub/sub. We define advertisements in the same way as subscriptions. However, the semantics is different: Advertisements specify the content a publisher will send in the future.

To confirm the common claim of an increased popularity of the pub/sub area, we have analyzed the number of academic publications on this topic in various digital libraries (ACM, DBLP, Google Scholar, IEEE, and SpringerLink): Whereas the overall number of publications per year has doubled since approximately 2001, the number of publications on pub/sub has quadrupled since that year.

Pub/sub, however, is not only an academic subject. Various companies apply this general event-driven paradigm in the context of message queuing systems, including IBM, Microsoft, and Oracle. Other well-known (commercial) applications of pub/sub include RSS feeds and Google Alerts.

Having introduced this general context of our PhD project, we outline the identified research problems in the next section.

2 Research Problems

An investigation of content-based pub/sub systems reveals their focus on subscriptions and advertisements as conjunctive filter expressions. Nearly all of the proposed solutions strongly depend on this restriction and are not directly applicable to, e.g., subscriptions as arbitrary Boolean filter expressions. Although one could convert arbitrary expressions to disjunctive normal forms (DNF) and handle every conjunction as separate subscription/advertisement [14,15], we believe that this canonical conversion is not a reasonable approach in pub/sub. The reason for this hypothesis is the exponential size of DNFs in the worst case: The already large number of subscriptions and advertisements explodes (exponentially) due to conversions.

In our PhD project, we want to build a content-based pub/sub system that supports arbitrary Boolean subscriptions and advertisements, and show its advantages compared to conjunctive solutions. In order to allow for this support, our research needs to contribute to various parts of these systems and to extend different components, as presented in the following subsections.

2.1 Matching of Arbitrary Boolean Subscriptions

The first component in pub/sub systems that needs to be enhanced is the matching algorithm. Current approaches that target at an efficient and scalable matching only support conjunctions, e.g., [12,16]. Thus, we need to develop a novel matching solution for arbitrary Boolean subscriptions.

This novel approach should be a general-purpose solution that also efficiently supports the use of mere conjunctive subscriptions. An obvious idea is to extend an existing conjunctive algorithm to support arbitrary Boolean subscriptions.

2.2 Optimizing the Routing of Arbitrary Boolean Subscriptions

The second component to be extended is the event routing optimizations. These optimizations are required in distributed pub/sub systems to reduce the sizes of event routing tables. Similarly to matching algorithms, current optimizations are only applicable to conjunctions, e.g., covering [10,14] and merging [13,14]. The reason is the complexity of these approaches for arbitrary Boolean expressions, being (co-)NP-hard.

Our novel routing optimization should efficiently support both arbitrary Boolean and conjunctive subscriptions to be universally applicable. When using an optimization method that works orthogonally to current approaches, there should even exist an optimization potential in situations that cannot be optimized by today's solutions. Additionally, conjunctive systems could simultaneously apply different optimizations to improve the overall optimization effects.

2.3 Supporting Arbitrary Boolean Advertisements

The final area requiring enhancements is the symmetrical support of arbitrary Boolean advertisements. To practically use advertisements, pub/sub systems need to determine the *overlapping* relationships among subscriptions and advertisements. Informally, a subscription s and an advertisement a overlap if a describes at least one event message that matches s. Current computation methods for overlappings exploit the conjunctive nature of subscriptions and advertisements, and are thus not applicable.

Distributed pub/sub systems internally forward advertisements to build up subscription routing tables. To reduce the sizes of these tables, one applies subscription routing optimizations. Currently, event routing optimization approaches (covering and merging, cf. Sect. 2.2) are likewise applied for this purpose. However, these optimizations only support conjunctions.

As a final step, we thus need to develop an overlapping calculation approach for arbitrary Boolean subscriptions and advertisements, and an subscription routing optimization. Both of these solutions should also be applicable to conjunctions.

3 Undertaken Steps

Within our PhD project, we have addressed all three problem areas of pub/sub systems that have been identified in Sect. 2. In this section, we briefly outline these contributions.

3.1 Matching Algorithm

We have developed a matching algorithm [4] for arbitrary Boolean subscriptions. This approach is the first matching solutions that both supports this class of subscriptions and utilizes one-dimensional predicate indexes to realize an efficient and scalable matching process. It is an extension of the conjunctive counting algorithm [16]. An important property of our approach is the support of conjunctive subscriptions in nearly the same way as the original algorithm.

The algorithm works in three steps: Firstly, it determines all predicates that match the attribute-value pairs of an incoming message. Secondly, it computes a set of *candidate subscriptions*, i.e., subscriptions that potentially match the incoming message, by conducting a set of subscription indexes. Finally, the Boolean operators of these candidates are evaluated to determine whether they constitute a real match. We have illustrated these steps in Fig. 2.

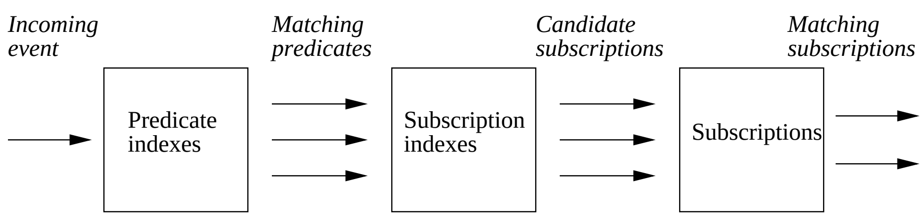

Fig. 2. Overview of the applied 3-step arbitrary Boolean matching algorithm

Memory Requirements. To analyze the memory requirements of various matching algorithms, we have developed a characterization scheme for subscriptions [3]. This scheme allows for the determination of the matching algorithm that requires less memory resources for a given set of subscriptions. Based on this scheme, we have then compared the memory usage of our Boolean and two conjunctive approaches [12,16]. We have found that various settings, even involving only one disjunction in subscriptions, should apply a Boolean matching algorithm due to its space efficiency.

Efficiency. As a next step, we have analyzed the efficiency properties of our Boolean and the conjunctive counting approach. We have evaluated both an artificial scenario but also a typical set of subscriptions for an online auction setting [7]. Our findings show that both matching approaches lead to comparable efficiency properties. Generally, the larger the canonical converted subscriptions, the less efficient the conjunctive and the more efficient our Boolean approach.

3.2 Event Routing Optimization

For the distributed part of pub/sub systems, we have developed a novel routing optimization, *subscription pruning* [9]. This optimization works for all kinds of Boolean subscriptions and can be tailored to optimize in respect to different target parameters, e.g., memory usage, efficiency, and network load [6].

The broad idea of subscription pruning is to remove (prune) parts of the tree structure that represents a Boolean subscription. To ensure correct event routing, every pruning operation needs to create a more general subscription. Pruning subscriptions in routing tables decreases the sizes of these tables by reducing the complexity of their routing entries. We have given an example event routing table before and after applying pruning in Fig. 3.

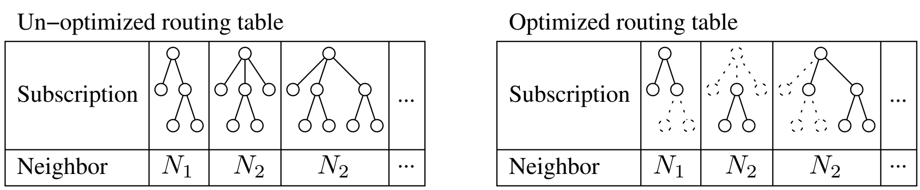

Fig. 3. Event routing table before (left) and after (right) applying subscription pruning

Analysis. We have analyzed and evaluated subscription pruning in an online book auction scenario [7] and found a promising optimization effect: At the same time, subscription pruning strongly decreases the sizes of event routing tables, increases the system throughput, and only slightly increases the internal[1] network load. The network-based optimization variant has led to the best overall results [6].

Comparison. Although subscription pruning is the first optimization that is practically applicable to Boolean subscriptions, we have additionally undertaken a comparison to conjunctive solutions: We have analyzed the conjunctive counting algorithm in combination with the subscription covering optimization, and our Boolean matching approach in conjunction with subscription pruning. Subscription pruning shows a more stable optimization behavior than subscription covering, i.e., it optimizes in all of the analyzed scenarios. Subscription covering, however, only optimizes if subscriptions fulfil its assumption, i.e., there are strong subset relationships among them.

Subscription pruning optimizes orthogonal to subscription covering, as it has been our design goal (cf. Sect. 2.2). Pruning can thus be additionally applied to subscription covering, which leads to an improved overall optimization, as we have shown in experiments.

[1] Subscribers are still notified correctly because the final notification delivery is based on the original, un-pruned subscription.

3.3 Advertisements

Our final step on the way to arbitrary Boolean pub/sub has been the symmetrical support of arbitrary Boolean advertisements. We have firstly developed an algorithm to calculate the overlapping relationships among arbitrary Boolean subscriptions and advertisements. Secondly, we have proposed the first optimization approach that is tailored to advertisements, advertisement pruning.

Overlapping Calculation. Our overlapping calculation approach uses the notion of *violating predicates* [5] (i.e., non-overlapping predicates) to determine the overlappings among subscriptions and advertisements. Conjunctive computation algorithms, conversely, can be based on overlapping predicates due to their common assumption that subscriptions contain exactly one predicate per attribute. Our semantics of Boolean subscriptions and advertisements, however, allows any number of predicates per attribute.

Similar to the matching approach (Sect. 3.1), our algorithm firstly determines violating predicates, secondly computes a set of *candidate overlappings*, and finally analyzes these candidates to resolve their real status of overlapping. The illustration in Fig. 4 exemplifies these steps from the viewpoint of subscriptions, i.e., for an incoming subscription the algorithm determines overlapping advertisements. Our approach works similarly for the other direction due to the symmetry of the overlapping relationship.

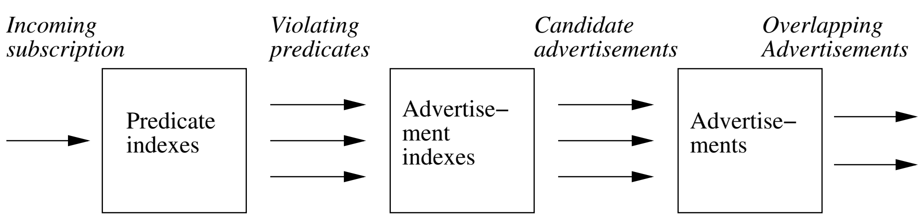

Fig. 4. Overview of the applied 3-step overlapping calculation algorithm

When comparing our approach to conjunctive solutions, we have found similar efficiency properties for the function problem (determination of all overlappings) in our online auction scenario. Our Boolean approach, however, shows a much better performance for the decision problem (determination whether at least one overlapping exists).

Advertisement-Based Optimization. *Advertisement pruning* [8] is the first designated subscription routing optimization. It takes a similar approach to subscription pruning (Sect. 2.2), i.e., it removes selected branches of the Boolean tree structure of advertisements. The overall goal of advertisement pruning is to reduce the memory usage for subscription routing tables (containing advertisements) without strongly increasing the existing overlapping relationships. The influence on the existing overlappings is estimated based on the violating predicates of advertisements and subscriptions, used in the overlapping calculation algorithm that has been presented in the previous paragraph.

4 Future Steps

In the previous section, we have described the research we have undertaken so far. These contributions widely align with the problems that have been identified in Sect. 2. However, some work still needs to be done.

4.1 Experimental Study

Currently, we are undertaking a large set of practical experiments to (1) further analyze the behavior of our approaches, and (2) to compare our solutions to existing methods in detail. One can, obviously, undertake an enormous amount of studies to investigate all details of the performance of our and existing approaches. One of the main challenges of our experiments is thus to identify those settings that provide the most valuable insights into the behavior of the analyzed algorithms.

4.2 Writing Up

Undertaking research is only the first step towards a PhD. The second (and at least equally important) step is to write up all findings and methods of research, to structure and organize thoughts and ideas, and finally to hand in a complete dissertation. We have just started this second step and, ultimately, work its conclusion.

5 Conclusions

The rather general presentation of our PhD project in this paper has aimed to give an overview about the broad topic we are working on in the associated dissertation. We have published details about our research findings in the three outlined subareas of pub/sub (cf. Sect. 3), as referenced in the respective sections. A more technical summary of our general PhD topic can be found in [2]. To conclude this paper, we now want to outline the general contributions of our research to the pub/sub area.

In our opinion, one of the most severe problems that pub/sub is facing is the lack of realistic, widely applied applications. A popular setting in the literature is the stock broker example, simplifying the pub/sub paradigm to the selection of stocks of certain companies. We believe that the choice of such scenarios has contributed to the focus of current pub/sub systems on conjunctive expressions, and that the analysis of more sophisticated settings will reveal the requirement of more complex subscriptions. We hope to have set a starting point with our choice of an online auction example scenario [9].

The chosen application scenario also influences the test settings that are applied in conducted studies. We believe that our analysis of typical event distributions [7] and the identification of various subscription/advertisement classes [8] is a step in the direction of more meaningful experiments and evaluations. This focus on actually existing problems, hopefully, will lead to an even wider adoption of pub/sub in everyday systems.

Although we strongly believe that academic research should not be controlled by economical constraints, and it should be possible to undertake research merely for the sake of it, a look into the real world might not always be damaging. This is particularly important to counteract the common prejudice that research creates new problems but does not solve existing ones.

References

1. G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman. An Efficient Multicast Protocol for Content-based Publish-Subscribe Systems. In *Proceedings of the 19th IEEE International Conference on Distributed Computing Systems (ICDCS '99)*, pages 262–272, Austin, USA, May 31–June 4 1999.
2. S. Bittner. Supporting Arbitrary Boolean Subscriptions in Distributed Publish/Subscribe Systems. In *Proceedings of the 3rd International Middleware Doctoral Symposium (MDS 2006)*, Melbourne, Australia, November 27–December 1 2006.
3. S. Bittner and A. Hinze. A Detailed Investigation of Memory Requirements for Publish/Subscribe Filtering Algorithms. In *Proceedings of the 13th International Conference on Cooperative Information Systems (CoopIS 2005)*, pages 148–165, Agia Napa, Cyprus, October 31–November 4 2005.
4. S. Bittner and A. Hinze. On the Benefits of Non-Canonical Filtering in Publish/Subscribe Systems. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW '05)*, pages 451–457, Columbus, USA, June 6–10 2005.
5. S. Bittner and A. Hinze. Arbitrary Boolean Advertisements: The Final Step in Supporting the Boolean Publish/Subscribe Model. Technical Report 06/2006, Computer Science Department, The University of Waikato, June 2006.
6. S. Bittner and A. Hinze. Dimension-Based Subscription Pruning for Publish/Subscribe Systems. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW '06)*, page 25, Lisbon, Portugal, July 4–7 2006.
7. S. Bittner and A. Hinze. Event Distributions in Online Book Auctions. Technical Report 03/2006, Computer Science Department, The University of Waikato, February 2006.
8. S. Bittner and A. Hinze. Optimizing Pub/Sub Systems by Advertisement Pruning. In *Proceedings of the 8th International Symposium on Distributed Objects and Applications (DOA 2006)*, pages 1503–1521, Montpellier, France, October 30–November 1 2006.
9. S. Bittner and A. Hinze. Pruning Subscriptions in Distributed Publish/Subscribe Systems. In *Proceedings of the Twenty-Ninth Australasian Computer Science Conference (ACSC 2006)*, pages 197–206, Hobart, Australia, January 16–19 2006.
10. 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.
11. M. Cilia and A. P. Buchmann. An Active Functionality Service For E-Business Applications. *ACM SIGMOD Record, Special Issue on Data Management Issues in Electronic Commerce*, 31(1):24–30, 2002.
12. F. Fabret, A. Jacobsen, F. Llirbat, J. Pereira, K. Ross, and D. Shasha. Filtering Algorithms and Implementation for Very Fast Publish/Subscribe Systems. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data (SIGMOD 2001)*, pages 115–126, Santa Barbara, USA, May 21–24 2001.
13. G. Li, S. Hou, and H.-A. Jacobsen. A Unified Approach to Routing, Covering and Merging in Publish/Subscribe Systems based on Modified Binary Decision Diagrams. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*, pages 447–457, Columbus, USA, June 6–10 2005.
14. G. Mühl and L. Fiege. Supporting Covering and Merging in Content-Based Publish/Subscribe Systems: Beyond Name/Value Pairs. *IEEE Distributed Systems Online (DSOnline)*, 2(7), 2001.
15. P. R. Pietzuch. *Hermes: A Scalable Event-Based Middleware*. PhD thesis, University of Cambrigde, Queens' College, February 2004.
16. T. W. Yan and H. García-Molina. Index Structures for Selective Dissemination of Information Under the Boolean Model. *ACM Transactions on Database Systems*, 19(2):332–364, 1994.