Advanced Engineering Forum *Online: 2012-09-26*
ISSN: 2234-991X, Vols. 6-7, pp 595-600
doi:10.4028/www.scientific.net/AEF.6-7.595
© 2012 The Author(s). Published by Trans Tech Publications Ltd, Switzerland.

Filter-Based Information Selection Mechanism in Publish/Subscribe Middleware

Jigang Xiao[1,a], Yunqing Chen[2,3,b], Zhuowei Shen[2,3,c]

[1] Systems Engineering Research Institute, China State Shipbuilding Corporation, Beijing, China

[2]Key Lab of Computer Network and Information Integration, MOE, Nanjing, China

[3] School of Computer Science and Engineering, Southeast University, Nanjing, China

[a]xiaojigang5878@sina.com,[b]yunlongzi@126.com,[c]zwshen@seu.edu.cn

Keywords: Information Filtering, Publish/Subscribe middleware, Data Distribution Service, Filter

Abstract. Publish/Subscribe middleware is getting more and more attentions for its feature of loose coupling.In some scenarios,Subscribersof a topic need not all the information belonging to that topic, but only those of interest. To fulfill this requirement,some kind of information selection mechanism is needed. In this paper, a filter-based information selection mechanism is proposed, which is compliant with OMG DDS specification. Then it is implemented in a publish/subscribe middleware prototype system.With the mechanism, a publish/subscribe middleware can use the compiler generated by Flex&Bison to compile the filtering rules of the users ,which are conform to SQL-like syntax,and generate the filtering syntax trees.At the subscriber side, when receiving a sample, the middleware substitute the values of the sample into the corresponding positions in the filtering syntax tree. By traversing the tree, the middleware makes the filtering decision. Experiment results reveal that the proposed mechanism is effective.

Introduction

Publish/Subscribe system is a kind of distributed system, whose participants interact with each other through the Publish/Subscribemechanism. Characteristics of loose coupling between the participants of the system make Publish/Subscribe system more and more popular in many application areas[1]. In a Publish/Subscribe system, the subscribersfocus on access the messages, the Publishers focus on publish their messages. The match between Publishers and Subscribers is completed by the system.The approach allows that publishers and subscribers decouple in time, space, synchronization,and greatly increases the flexibility and scalability of the applications[2, 3].OMG's DDS specification（Data Distribution Service for Real-time Systems）has become the industry standard of Information Distribution forPublish /Subscribe system. Most of the industry-leading publish /Subscribe systems, such as RTI DDS, are compliant with DDS specification[4, 5, 6, 7].

In many application scenarios of topic-based Publish /Subscribe system,subscribers are often only interested in the part of the information under the topic.For example,stock information integrated management system publishes information for the stock name and its price. Subscribers tend to care about only a few stocks. If the system pushesall the information of thousands of stocks to user, users have to filter thatirrelevant information, which is high time- and resource-consuming.So the middleware needs to provide users with the information filtering mechanism.In OMG DDS specification, a mechanism namedContentFilteredTopicis provided. The mainstreamPublish/Subscribe products have implemented this mechanism [4,5].In this paper, based on an OMG DDS compliant Publish/Subscribe prototype middleware we called it I^2MS (Information Integration Management Software), we developed a information selection mechanism, which conforms to ContentFilteredTopic interfaces in OMG DDS specification.

Filtering rules syntax

Filtering rules, also known as subscription expressions, are used to describe the requirements for relevant data of information, indicating what the user interests.In OMG DDS specification, a SQL-like syntax is used to define subscription expressions. According to the syntax, a filtering rule is a simple comparison operator expression or a complex logic operator expression.

This article is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0)

◆ Simple comparison operator expression

Value of a field in the information of the topic is equal to (not equal, greater than, and less than) a givenvalueor fallsintoaninterval set, for example, *key=10*，*key>20*，*key between 10 and 20*.

◆ Complex logic operator expression

Logic operation of simple comparison operator expression, such as:*key>10 AND key<20*.

With this syntax, a filtering rule has the enough capacity to express complicate filtering conditions.

In addition to the syntax rules defined in OMG DDS specification, we introduce the operatorstrlikeinto the filtering rules syntax, which supports fuzzy filtering of the string type.The operator based on the concept of edit distance(numbers of modified characters when modifying a string to another one)judges whether two strings are similar.

Information Selection mechanism based on filter

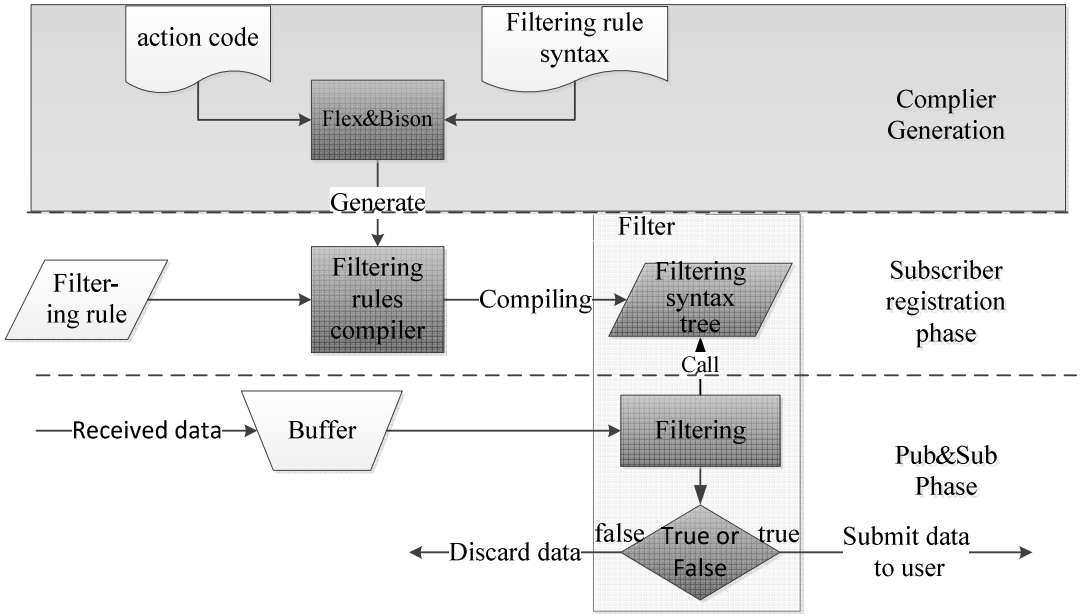

Fig.1 Framework of Information selection mechanism

Framework of Information Selection Mechanism. Fig.1 shows the processing flow of the information selection, in which the filtering rules compiler and the filter are key components.

Filtering rules compiler was generated by the general compiler tools Flex & Bison.The inputs of the tools are theaforementioned filtering rules syntax andcorresponding action code. The tools generate the source code of filtering rules compiler, which is then built into the library of Publish/Subscribe middleware and called online.The action code specifies the instructionsthe compiler executeswhen the input conform to certain grammatical rules.Since the compiler builds the filtering syntax tree according to the input filtering rule, here the action code means the operation of the filtering syntax tree, such as adding a node in the tree [8].

When a subscriber wants to subscribe a topic, it needs to register itself in the system firstly. At this time, it providesa subscription expression (filtering rule) which represents its interest.The compiler compiles the expression and builds the corresponding filtering syntax tree.

At the Publish/Subscribe phase, when the subscriber receives the data published by the publisher, the filter checks the dataaccording to the value of data and the filtering syntax tree, to determine whether to deliver it to the user or just to discard it.

Generation of Filtering Syntax Tree. We select the tree data structure to representthesubscription expression because the treeis intuitive to express the logical relationship between each part of the subscription expression, and convenient to be used in the filtering process. For example, a subscriber of stock information has the subscription expression as follows: “StockCode=AAPLOR StockCode=MSFT”, in which AAPL stands for Apple’s stock code and MSFT stands for Microsoft’s stock code. In subscriber registration phase, the filtering rules compiler takes the expression as the input and parses it to build the filtering syntax tree as shown in Fig. 2.

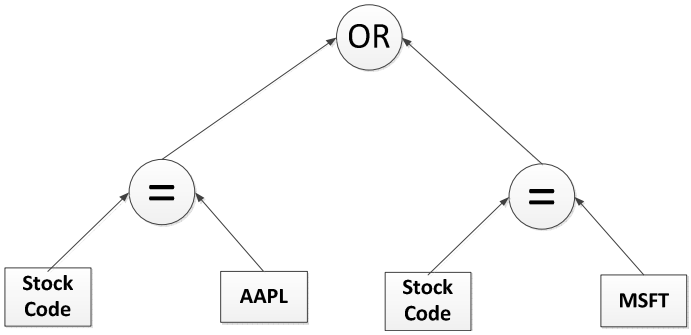

Fig. 2 Filtering syntax tree of the expression "StockCode=AAPL OR StockCode=MSFT"

Filtering in Publish/Subscribe Phase. Filteringmeans the filter extracts the related components of the data according to the filtering syntax treeand determines whether to accept the data or not based onthe traversing result of the filtering syntax tree.

There are three kinds of nodes in the syntax tree relating to the data.

- Fixed value node

The node is generatedfrom explicit numerical in the user'sfiltering rules, data types and values of information are stored in a node structure.

- Parameters node

The node is generated from the "*%n*" form parameters in the user'sfiltering rule. *n* represents the value's index positionin the parameter storage queue.It improves filtering flexibility. Modifying parameters can affect the filtering results, without reconstruction filter expressions.

- Variable node

While filtering, this kind of nodes needs to be substituted by the actual value of the correspondingvariable in the data.

In the process of filtering, the value of data is substituted into thecorresponding node of the filtering syntax tree, then the filter traverses the whole tree from the leaf node to root, the result in the root node indicates whether thedata will be filtered or not.

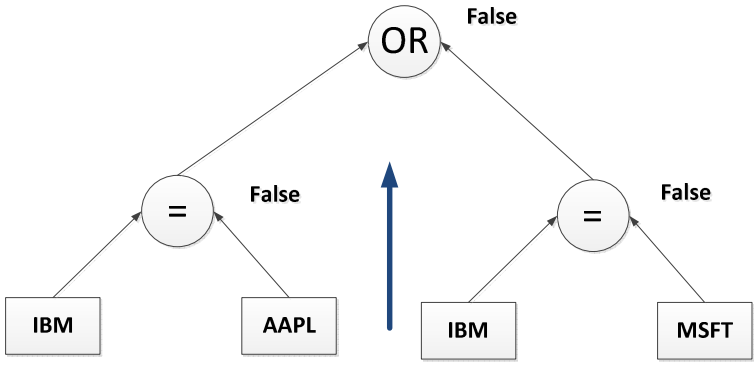

Fig.3 Filtering process of the corresponding filtering syntax tree in Fig. 2

As shown in Fig. 3, in the received data, the value of the Stockcode is IBM，so the corresponding nodes are replaced with IBM. The result of traverse is False which means the data will be discarded.

Evaluation

The filter-based information selection mechanism has been implemented in I^2MS, a Publish/Subscribe prototype middleware developed by authors' team. Functional tests show that the implementation is conform to OMG DDS specification, and can satisfy the information selection requirement. Besides functional tests, we evaluated the proposed mechanism by performance

testing. The performance metrics are subscription delay and throughput. We designed 6 test cases. In these cases, we used two data structures as topic information, one is simple and the other is complex. The complexities of the subscription expressions are different. The detailed description of test cases is shown in Table 1.

Table 1 Design of test cases

Data structure / Filtering rules		Simple data structure Non-nested datastructure, orone-dimensional array	Complexdata structure Nested data structure, or multidimensional array
No filter		**Test case1** Simple data structure without filtering	**Test case 4** Complex data structure without filtering
Simple filtering rule	Simple comparison operator expression, such as:key=10	**Test case 2** Simple data structure simple filtering rule	**Test case 5** Complex data structure simple filtering rule
Complex filtering rule	Complex logic operator expression, such as：key>=10 AND key<= 20	**Test case 3** Simpledatastructure complex filtering rule	**Test case 6** Complex data structure complex filtering rule

As shown in Fig.4, delay with no filtering rule is the shortest, and the cases with complex filtering rule have the longest delays. We can also observe thatwhen using the same filteringrule, the delay of the case with complex data types is longer than that of the case with simple data types.The test results reveal that both the data type complexity and filtering rule complexity have an impact on subscriptiondelay. Moreover, in comparison, the complexityof filtering rulehas a greater impact. Considering that the complex filtering rule is much more complicated than the simple filtering rule (the nodes of the filtering syntax tree corresponding to the complex filtering rule is 5 times than that corresponding to the simple filtering rule), butthe growth of delay is only about 15%. In another words, the filter we implementedshows a good performance when the complexity of filtering rulesincreases significantly.

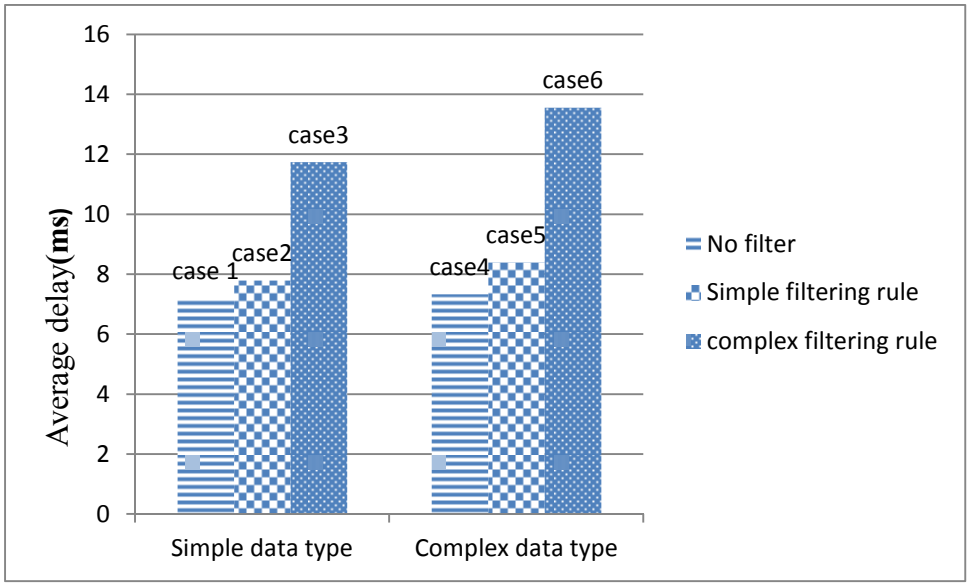

Fig.4 Average delay under various scenarios

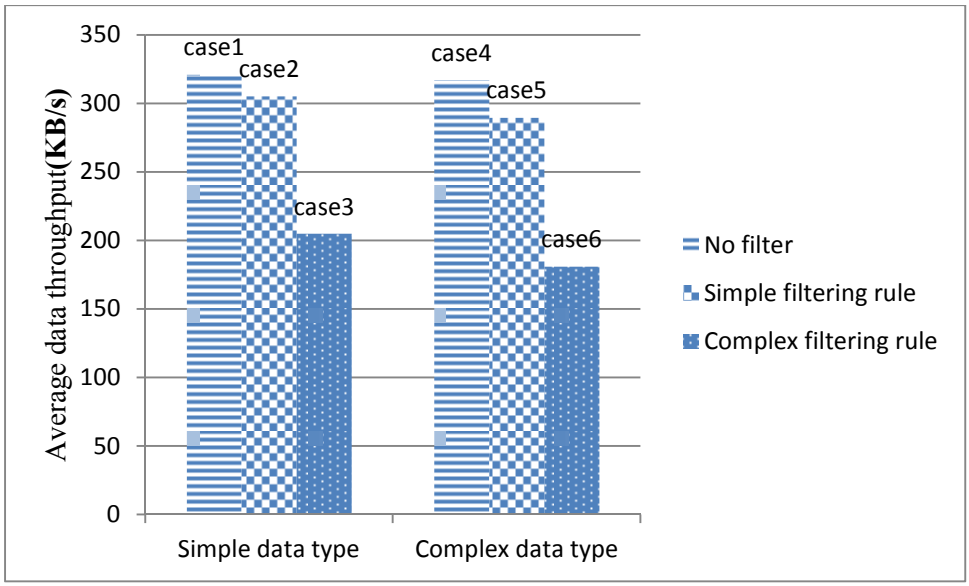

Fig. 5 Average throughputunder various scenarios

Throughput under various scenarios is shown in

Fig. 5. The results are similar to that of Delay test. As shown in Fig. 5: Both the data type complexity and the filtering rule complexity have an impact on subscription throughput. Comparatively speaking, the complexity of filtering rule is the main factor to affect the throughput.

Conclusions

In this paper, wemade use ofthe general compiler tools Flex & Bison, proposed a filter solution to realize the information selection function in Publish/Subscribe middleware, which is based onfiltering syntax tree. Then we implemented the solution in I^2MS, aPublish/Subscribe middleware prototype system.The tests on the prototypeshow that our solution is conform to the OMG DDS specification and its performance is as expected.

Acknowledgments

This paper is supported by the Natural Science Foundation of China (Grant No. 60903163) and Aviation Science Foundation (Grant No. 20101969010).

References

[1] Patrick TH. Eugster, Pascal A. Felber,RachidGuerraoui, Anne-Marie Kermarrec, The Many Faces of Publish/Subscribe, ACM Computing Surveys, 35(2):114-131, 2003.

[2] Object Management Group, Data Distribution Service for Real-time Systems Specification,Version1.1, 2005.

[3] Object Management Group, The Real-time Publish/Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification, Version 2.1, 2009.

[4] Gerardo P C. OMG Data-Distribution Service (DDS): Architectural Update. 2004 IEEE Military Communications Conference, 2: 961-967, 2004.

[5] Schneider S, Farabaugh B, Using the DDS Standard for High Reliability Applications. Real-Time InnovationsInc., 2004.

[6] Real-Time Innovations. The Real-Time Publish/Subscribe Middleware User's Manual, Version 4.5C, 2010

[7] Object ComputingInc.,OpenDDS Developer’s Guide, Version 2.3,2007.

[8] John Levine. Flex&Bison, Southeast University Press, Nanjing, 2011.