Advanced Science and Technology Letters
Vol.51 (CES-CUBE 2014), pp.15-19
http://dx.doi.org/10.14257/astl.2014.51.04

Causal Order Multicast Protocol Using Different Information from Brokers to Subscribers

Chayoung Kim[1] and Jinho Ahn[1],

[1] Dept. of Comp. Scie., Kyonggi Univ., Iuidong, Yeongtong, Suwon 443-760 Gyeonggi, Republic of Korea
{kimcha0, jhahn}@kgu.ac.kr

Abstract. Recently, in Publish/Subscribe (P/S) systems, there has been increasing emphasis in managing end-to-end message delivery performance and message order-based consistency, which are addressed in distributed collaborative applications for on-the-fly data. A causal message ordering is more useful for these distributed applications in which a large number of processes request collaboratively and interactively in services on P/S systems. Also, in P/S systems of wireless sensor networks (WSNs), data fusion, the process of correlating individual sensor readings originating from various nodes into high-level sensing results, depends on the time of occurrence of fused sensor readings, such as causal ordering. In this paper, we present two versions of causal ordering protocols. In the first protocol, only the information of the predecessors immediately before the message piggybacked on each multicast message is transmitted from brokers to subscribers through gossip-style disseminations based on local views for causal ordering. In the second protocol, every sensor broker disseminates the multicast message piggybacked with the latest time-stamped information that represents the gossip round in which the message is generated to subscribers by using global view of gossiping for causal ordering. The features of these two versions might be highly scalable and suitable for the area of the applications requiring only the minimum causal information of message delivery with flexible consistency.

Keywords: Publish/Subscribe, group communication, reliability, scalability, wireless sensor networks

1 Introduction

In most large-scale distributed applications, such as social web platforms, publish/subscribe (P/S) systems are suitable for communication between software components that are deployed over a large number of sites. P/S systems follow a many-to-many communication pattern, allowing a decoupling between senders and receivers to interact with publishers and subscribers [6]. Recently, social web platforms, such as Facebook and Twitter, have become real-time social communication ones, focusing on the dissemination, processing and caching of fresh data. So, it is important and reasonable that end users expect on-the-fly data, which is immediately available to all, interested other end users [5]. And, there has been increasing emphasis in managing end-to-end message delivery performance and

ISSN: 2287-1233 ASTL

message order-based consistency, which have been addressed in distributed collaborative applications. Especially, for on-the-fly data processing, some distributed applications and products have offered message order consistency guarantees, such as Isis2 system [3]. Isis2 supports virtually synchronous process groups and considers full-fledged atomic message ordering, but not causal message ordering. A causal order protocol ensures that if two messages are causally related and have the same destination, they are delivered to the application in their sending order [1]. A causal order is more useful than a strong atomic order for large-scale distributed applications in which a large number of processes request collaboratively and interactively in real-time social web platforms based on P/S systems [5]. And gossip protocols based on P/S systems are becoming one of the promising solutions for addressing P/S scalability problems and very useful for the applications with a mixture of diverse message order consistencies [3]. In this paper, we present two versions of causal order protocols using gossip protocols.

In large-scale distributed applications of P/S systems, if causal ordering protocol is performed by the all brokers on global membership views, it is likely to be high overloaded on every member and not scalable. In order to address this problem, promising gossip protocols should have all the required features by achieving a high degree of reliability and strong message delivery ordering guarantees, even if every broker has a local membership view [4]. So, we present two versions of causal order protocol, the one is based on a local view, which is for larger-scale distributed P/S systems and the other is based on a global view, which is for pre-planned wireless sensor networks(WSNs). One of the proposed protocol based on local views guarantees the causally ordered delivery by using the context graph [6], which manages the causal order information based on the semantics of sent and received messages. But, the other based on the global views uses the whole set of vectors [1], used for traditional reliable group communications [1]. In some WSNs, such as pre-planned and time-lines ones [2, 3], it is not considerable to manage the global views. In the proposed protocol based on local views, all ancestors before the multicast message are sent and received between the brokers. But, from brokers to subscribers, only the immediate predecessors are disseminated instead of all ancestors. Its features might result in its very low communication overhead between brokers and subscribers because the immediate predecessors are in the structure of one-dimensional vector. But, all ancestors are in the structure of two-dimensional context graph [6].

In the global views of the whole set of vectors, every broker can manage a vector per group that represents its knowledge for the number of multicast messages generated by other members, as same as each member in the protocol of Birman et. al. [1]. In some WSNs, which can manage global view without high loads, every broker can aggregate, send and receive the whole set of group vectors for causal ordering by gossip protocols and fire synchronization [6]. In the proposed protocol based on global views, between the sensor brokers, the whole set of vectors, which represents the knowledge for the number of multicast messages are sent and received. But, from brokers to subscribers, only the timestamp that represents the gossip round in which the immediate predecessor are generated are disseminated. Especially, in the protocol, the timestamp is represented in the way of colors, which stands for the gossip round. And broker A and B can generate a message per gossip round. That means that the proposed protocol needs one-dimensional vector, whose size is the number of brokers

because of one color per sensor broker. The protocol is appropriate for sensor networks in a pre-planned manner time-lines ones [6] because it is not a high burden to manage global membership views. Therefore, these two versions of causally ordered delivery protocols are highly scalable and suitable for the area of the applications requiring only the minimum causal information with flexible consistency.

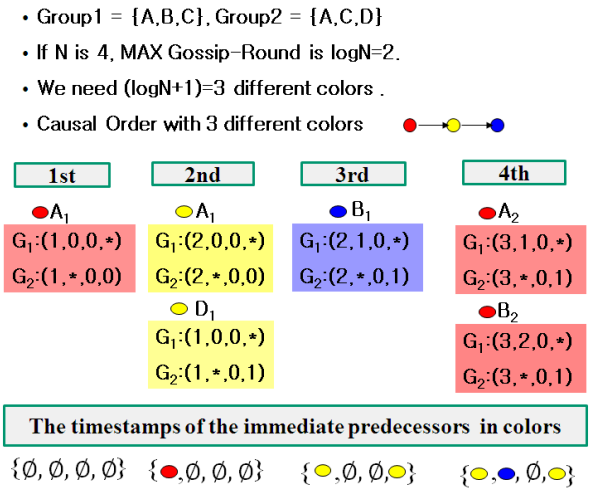

Fig. 1. Each gossip round represented in each color

2 The Proposed Protocol

In this section, we describe our proposed protocol for pre-planned WSNs through examples of figure 2 and 3, which show how in detail each broker generates a multicast message and aggregates causally ordered delivery information. As same as the protocol of Birman et. al. [1], in our proposed protocol, a vector timestamp (VT) per group, piggybacked on a multicast message counts the number of messages that causally precedes it. So, the notation g_α : VT[i] counts multicast messages sent in the group gα by a process i. In general, gossip protocols take O (logN) gossip rounds to reach all nodes [4], where N is the number of nodes. In each gossip round, every process has initiated a gossip message exactly once and gossips about it to $f \geq 1$ other processes, called as fan-out(f) or gossip targets, at random. So, each gossip round can be characterized as a unique notation represented using a color. The proposed protocol needs logN + α colors because the maximum number of gossip rounds in which all processes receive all messages eventually is logN and α may be application specific for buffering. As shown in figure 2, if two messages m(A_1) and m'(D_1) have been sent at the same gossip round, then they are independent of each other and represented in the same color, that is yellow.

This example of figure 2 shows how in detail each broker participating in G_1 = {A, B, C} and G_2 = {A, C, D} aggregates the information of causally ordered delivery and

sends it to subscribers. The example of figure 2 sets α to 1. So, it needs 3 colors because logN is 2 and α is 1. The stale messages might be removed periodically to respect the maximum number of gossip rounds as same as pbcast [4]. Also, our proposed protocol uses the epoch, which is incremented by 1 whenever all colors (in figure 2, red->yellow->blue) have been completed exactly once, distinguishing new message from previous message sent by the same process in the same color. As shown in figure 1, A_1 and A_2 in red can be distinguished by the epoch 1 and 2. In figure 2, in the first round, broker A generates the message and makes it with ID "A", the epoch "1" and the current gossip round color "red", as "red_{A1}". At the beginning, the vector of the immediate predecessors is all 0, that is {0, 0, 0, 0}. In the second round, broker A and D generate each message, and make it with ID "A" and "D", the epoch "1" and the current gossip round color "yellow", as "$yellow_{A1}$" and "$yellow_{D1}$", respectively. On receiving "red_{A1}", the broker A and D update the vector of the immediate predecessors, as {red, 0, 0, 0}. On being the fourth round, the epoch is incremented by 1 because red->yellow->blue have been completed exactly once. So, in the fourth round, broker A and B generate each message, and make it with ID "A" and "B", the epoch "2" and the current gossip round color "red", as "red_{A2}" and "red_{B2}", respectively. On receiving "$yellow_{B1}$", all brokers update the vector of the immediate predecessors, as {yellow, blue, 0, yellow}.

3 Conclusion

In this paper, we present two versions of broker-based causal order multicast protocols in P/S systems. In the protocol based on local views of gossiping, each broker sends and receives the multicast message including all ancestors of the context graph. But, from brokers to subscribers, each broker disseminates the multicast message including only the immediate predecessors instead of all ancestors. The immediate predecessors are in the structure of the one-dimensional vector, while all ancestors are the two-dimensional of the context graph. In the first step of sending a message, when every broker generates a multicast message, it puts its ID, the sequence number and the group lists of all groups that it participates on the message. In the second step, the broker attaches the message to all leaf nodes in its context graph. Then, the multicast message becomes the leaf and the parent messages of it become the immediate predecessors. In the last step, the broker sends the message including all ancestors to other brokers, but it including only the immediate predecessors, instead of all ancestors of the context graph to subscribers.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(grant number 2012R1A1A2044660).

References

1. Birman, K., Schiper, A., and Stephenson, P.: Lightweight Causal and Atomic Group Multicast. ACM Transactions on Computer Systems. Vol. 9, No. 3, 1991, pp. 272-314.
2. Birman, K., Hayden, M., Ozkasap, O., Xiao, Z., Budiu, M., and Minsky, Y.: Bimodal Multicast. ACM Transactions on Computer Systems. Vol. 17, No. 2, 1999, pp. 41-88.
3. Birman, K., Huang, Q., and Freedman, D.: Overcoming CAP with Consistent Soft-State Replication. IEEE Internet Computing. Vol. 12. 2012, pp.50-58.
4. Eugster, P., Guerraoui, R., Handurukande, S., Kouznetsov, P., and Kermarrec, A.-M.: Lightweight probabilistic broadcast. ACM Transactions on Computer Systems, Vol. 21, No. 4, 2003, pp. 341-374.
5. Eyers, D., Freudenreich, T., Margara, A., Frischbier, S., Pietzuch, P., and Eugster, P.: Living in the present: on-the-fly information processing in scalable web architectures. In CloudCP, 2012.
6. Felber, P., and Pedone, F.: Probabilistic Atomic Broadcast. in Proceedings of 21st IEEE Symposium on Reliable Distributed Systems (SRDS'02), Osaka, Japan, Oct. 2002, pp.170-179.

