Causal Order Multicast Protocol based on P/S Considering Delivery Time

Chayoung Kim¹ and Jinho Ahn¹,

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

Abstract. Recent researches on designing scalable architectures for social web platforms focus on the on-the-fly fresh data based on a content-based distributed P/S(publish/subscribe) infrastructure consisting of stable brokers. P/S based on gossip protocols is easily to scale in and out but does not deal with end-to-end message delay and message order-based consistency. This paper proposes deadline-constrained causal order protocol based on scalable P/S architecture to guarantee causally ordered message delivery with little communication overhead. In the proposed protocol, every broker manages a 2-dimensional vector, representing its knowledge the last message sent by each broker at a certain time t. On the other hand, every broker disseminates a 1-dimensional vector that represents the maximum number of gossip rounds (deadline) to subscribers.

Keywords: publish/subscribe system, multicast, real-time, message delivery order

1 Introduction

As the role of social web platforms shifts from being portals for largely historic data towards providing platforms for real-time fresh data analytics, they focus on shifting from the management of past data to the on-the-fly processing of fresh data [5]. The social web platforms should provide an adequate communication infrastructure to easily scale in and out and use a content-based distributed publish/subscribe infrastructure that act as a scalable communication backbone, such as ASIA [5, 6]. ASIA [6] provides resilience against failures and overloaded brokers, such as multi-path routing of messages [4]. But, path redundancy has still a challenging issue, such as network diversity [4]. So, in our proposed protocol, P/S architecture is based on gossip protocols, which are seem more appealing in many P/S systems because they are more scalable than traditional reliable broadcast. Gossip protocols are one of the temporal redundancy, which allows a publish/subscribe middleware to be defined timeliness: given a certain time deadline, Δ all non-faulty subscribers are notified of a published event before Δ is expired [4]. On the other hands, P/S based on gossip protocols does not deal with end-to-end message delay and message order-based consistency, which are addressed in real-time collaborative
applications [1]. If P/S architecture for social web platforms could deal with message timelines and ordering consistencies, real-time collaborative applications, such as [1] might focus on synchronizing such video playback on multiple devices with different playback engines and network bandwidth.

Therefore, in this paper, we present a causal order multicast protocol based on P/S by gossip protocols to guarantee causal order delivery of messages, respecting deadline-constraints from brokers(providers) to subscribers for real-time collaborative applications. Our proposed protocol could extend deadline-constrained causal order one [7] in the context of scalable distributed P/S.

To prevent causal order violation, either message might be forced to wait for messages in their past or late messages might have to be discarded [7]. For real-time collaborative applications [1], the first approach is not suitable since when a message has bypassed its deadline, all messages that causally depend on it might be forced to bypass their deadlines. In our proposed protocol, every broker manages a 2-dimensional vector like in the protocol [7], representing its knowledge, the last message sent by a broker x to broker y has been sent at time t. On the other hand, every broker disseminates the multicast message including a 1-dimensional vector, the time-stamped information that represents the maximum gossip round, which means its deadline (lifetime), to subscribers only using gossips. In between brokers, because collaborative multicast messages are based on IP-Multicast and gossip protocols to all interested brokers, their messages have their unique deadline(lifetime) for each collaborative applications. But, from brokers to subscribers, because all messages are based on P/S using gossip protocols and dependent on each periodic gossip round and the maximum number of gossip rounds is deadline(lifetime). That is, the proposed protocol needs a vector, which size is the number of brokers because one color of the lifetime represents the deadline of the last messages of each broker. Therefore, our proposed protocol on P/S using gossip protocols results in very low communication overhead from brokers to subscribers in the context of respecting deadline-constrains because of the same deadline of all messages.

2 The proposed protocol

In this section, we describe our proposed protocol through an example of figure 1, which shows how messages and information are disseminated from brokers to subscribers. The proposed protocol respects deadline-constraints causal order using MCP_i (MCP stands for Message Causal Past) between brokers, like in the protocols of Rodrigues et. Al [7]. MCP_i describes the causal past of the message each broker i sends and receives for causal order and MCP_i[x,y]=t describes broker i knows the last message sent by broker x to broker y has been sent at time t.

Let us an example of figure 1. Each broker has to manage the whole set of these vectors, 2-dimensional vectors (MCP_m) between brokers. On the other hand, every broker disseminates the multicast message with 1-dimnesional vector(VLC_m) from brokers to subscribers. The meaning of the vector, VLC_i={0,0,0,0} is that each index is each broker, A=1, B=2, C=3, and D=4 and the deadline of the last sent message sent by B is red. So, this message should be delivered before the immediate oncoming
red. In figure 1, in the first round, broker B generates the first multicast message and the current gossip round color is "red", denoted "redB1". For all \( j \in \text{Dest}_m \), VLC\([j]\) is stored in VLC\(_m\) and the multicast message \( m \) with VLC\(_m\) is disseminated for all destination subscribers for deadline-constraints causal order. So, if subscriber \( j \) is one of the destinations, then VLC\([j]\) is updated as the sending group round, send\(_\text{color}_m\) = current\(_\text{color}\). So, we use \( m \rightarrow m' \) \( \iff \) VLC\(_m\) < VLC\(_{m'}\) according to this approach. In the second round, broker A generates the multicast message and the current gossip round color is "yellow", denoted "yellowA1". In the third round, B generates the multicast message and the current gossip round color is "blue", denoted "blueB2". And after the end of the first maximum gossip round, the order the order in which they appear: red, yellow, and blue is restarting. So, in the fourth round, broker B generates the multicast message and the current gossip round color is "red", denoted "redB3". In this proposed protocol, when a broker disseminates multicast messages to its subscribers there are two cases, 1) the messages might be delivered or 2) the messages might have to be discarded.

For discarding a message, when same\(_\text{color}_m\) \( \rightarrow \) same\(_\text{color}_{m'}\) \( \iff \) VLC\(_{\text{same-color}_m}\) < VLC\(_{\text{same-color}_{m'}}\), same\(_\text{color}_m\) have arrived and are not yet delivered at subscriber \( S_i \) not receiving same\(_\text{color}_m\). This means that same\(_\text{color}_m\) was generated in the gossip round, \(|\log N + 1|\) rounds ahead of the gossip round of \( m'\), and same\(_\text{color}_m\) has bypassed deadline\(_{\text{same-color}_m}\) and if its delivery violate causal order, then it should be discarded. This is the case of message redB1 arrived at subscriber \( S_3 \). On receiving message redB1, subscriber \( S_3 \) delivers yellow\(_{A1}\), blue\(_{B2}\) and red\(_{B3}\) to the application layer in order not to violate deadline-constraints causal order. So, if redB1 has bypassed deadline (lifetime)\( = |\log N + 1| = 3\), redB1 should be discarded because its delivery violate causal order. If \((|\text{Deadline}_m| < \text{current}_\text{color}) \land \text{VCL}_m[j] < \text{VCL}_i[j]\) or Delivery-Violate-Causal-Order \( \equiv (\text{VCL}_m[j] \leq \text{VCL}_i[j])\) is true, then \( m \) is discarded. Message redB3 was generated in the gossip round, \(|\log N + 1|\) rounds ahead of the current gossip round and it has bypassed deadline\( = |\log N + 1| = 3\). And also, VLC\(_i[x]\) is updated after deliveries of yellow\(_{A1}\), blue\(_{B2}\) and red\(_{B3}\) in order not to violate deadline-constraints causal order. So, redB3 should be discarded.

**Fig. 1.** An example of messages with the maximum gossip round(deadline) in 1-dimensional vector from brokers to subscribers
3 Conclusions

This paper presented a causal order multicast protocol based on P/S using gossiping for real-time collaborative applications in social web platforms, respecting deadline-constraints from brokers to subscribers. In between brokers, because collaborative multicast messages are based on IP-Multicast and Bimodal Multicast [2] or LPBCast[3], their messages have their unique deadline. On the other hands, from brokers to subscribers, the maximum number of gossip rounds is deadline and all messages from brokers to subscribers have the same deadline because all messages are using gossip protocols. From brokers to subscribers, each message with deadline represented in colors of the lifetime of each message is disseminated using gossip protocols from brokers to subscribers. So, the proposed protocol needs a vector, whose size is the number of brokers because one color of the lifetime represents the deadline of the last message generated by each broker. So, our proposed protocol is not expensive because of low overhead from brokers to subscribers.

Acknowledgments. This work was supported by the Gyonggi Regional Research Center (GRRC) and Contents Convergence Software (CCS) research center in Korea(Project No.: GRRC Kyonggi 2013-B04).

References