An Optimal Algorithm for Generalized Causal Message Ordering

Ajay D. Kshemkalyani
IBM Corporation
P. O. Box 12195
Research Triangle Park, NC 27709

Mukesh Singhal
Dept. of Computer & Info. Science
The Ohio State University
Columbus, OH 43210

Asynchronous execution of processes and unpredictable communication delays create nondeterminism in distributed systems that complicates the design, verification, and analysis of distributed programs. To simplify the design and development of distributed applications, the idea of "causal message ordering" (CO) was introduced by Birman and Joseph [1]. If for any two messages M and M' sent to the same destination d, $Send(M) \longrightarrow Send(M')$, then CO ensures that $Delivery_d(M) \longrightarrow Delivery_d(M')$ at d. Existing algorithms that provide CO have high message overhead and use more storage than is required by an optimal algorithm, or make simplifying assumptions by assuming certain topology/communication patterns and inbuilt synchronization.

We present an optimal CO algorithm [2] under the following framework:

- processes are allowed to multicast to arbitrary and dynamically changing multicast groups,
- the system has asynchronous communication with reliable non-FIFO message delivery using a nonblocking protocol – at both the application level and lower layers – without any inbuilt synchronization, and
- there is no a priori knowledge about the network topology or communication pattern among processes.

The algorithm achieves optimality by storing and transmitting the bare minimum causal dependency information and using an encoding scheme to represent, store, and transmit this information. The algorithm is shown to be optimal both in space complexity of message overheads and in space complexity of message logs.

We first identify necessary and sufficient conditions, in the form of *Propagation Constraints*, on information about messages sent in the past that must be stored and propagated to enforce CO optimally [2]. An algorithm is then devised to meet these Propagation Constraints [2]. The algorithm stores and propagates information about a message sent in the causal past *as long as* and *only as long as* (I) it is not known that the message is delivered and (II) it is not guaranteed that the message will be delivered in CO. In addition to the Propagation Constraints, the algorithm follows a *Delivery Condition* by which a message M' that carries information "d is a destination of M" about a message M sent to d in the causal past, is not delivered to d if M has not yet been delivered to d.

Condition (I) and the Delivery Constraint contribute to optimality as follows: To ensure that M is delivered to d in CO, the information "d is a destination of M" is stored/propagated *on* and *only on* all causal paths starting from $Send(M)$, but nowhere in the causal future of $Delivery_d(M)$.

PODC'96, Philadelphia PA, USA
© 1996 ACM 0-89791-800-2/96/05..$3.50

Condition (II) and the Delivery Constraint contribute to optimality by the following transitive reasoning: Let messages M, M' and M'' be sent to d, where $Send(M) \longrightarrow Send(M') \longrightarrow Send(M'')$ and M' is the first message sent to d on all causal chains between the events $Send(M)$ and $Send(M')$. M will be delivered optimally in CO to d with respect to (w.r.t.) M'' if (i) M is guaranteed to be delivered optimally in CO to d w.r.t. M', and (ii) M' is guaranteed to be delivered optimally in CO to d w.r.t. M''. Condition (i) holds if the information "d is a destination of M" is stored/propagated *on* and *only on* all causal paths from $Send(M)$, but nowhere in the causal future of $Send(M')$ other than on message M'. This follows from the Delivery Condition. Condition (ii) can be shown to hold by applying a transitive argument comprising of conditions (II)(i) and (I). Thus, to achieve optimality, the information "d is a destination of M" must not be stored/propagated in the causal future of $Send(M')$ other than on message M' (condition (II)) or in the causal future of $Delivery_d(M)$ (condition (I)).

Information about messages (I) not known to be delivered and (II) not guaranteed to be delivered in CO, is explicitly tracked by the algorithm using (source, destination, scalar timestamp) information. The information is deleted as soon as either (I) or (II) becomes false. Information about messsages already delivered and messages guaranteed to be delivered in CO is implicitly tracked without storing or propagating it, and is derived from the explicit information. Such implicit information is used for determining when (I) or (II) becomes false for the explicit information being stored or carried. The explicit information and the encoded implicit information is the bare minimum causal dependency information required to be stored and transmitted to enforce CO optimally as per the Propagation Constraints. The algorithm stores/propagates only this information.

The full paper [2] gives the Propagation Constraints, the optimal CO algorithm, and a rigorous correctness proof. The proof shows that certain invariants satisfied by the algorithm satisfy the proposed necessary and sufficient conditions for optimality and correctness.

References

[1] K. Birman, T. Joseph, Reliable Communication in Presence of Failures, *ACM Trans. on Computer Systems*, 5(1), 47-76, Feb. 1987.

[2] A. D. Kshemkalyani, M. Singhal, Necessary and Sufficient Conditions on Information for Causal Message Ordering and Their Optimal Implementation, Technical Report 29.2040, IBM Research Triangle Park, July 1995. (Also Tech. Rep. CISRC-7/95-TR33, July 1995, Dept. of Computer Science, Ohio State Univ., ftp.cis.ohio-state.edu/pub/tech-report/1995/TR33.ps.gz.)