Probabilistic Analysis of Causal Message Ordering

Li-Hsing Yen
Department of Computer Science and Information Engineering
Chung Hua University
Hsinchu, Taiwan 30067, R.O.C.
lhyen@chu.edu.tw

Abstract

Causal message ordering (CMO) demands that messages directed to the same destinations must be delivered in an order consistent with their potential causality. In this paper, we present a modular decomposition of CMO, and evaluate the probability of breaking CMO by assuming two probabilistic models on message delays: exponential distribution and uniform distribution. These models represent the contexts where message delays are unpredictable and, respectively, unbounded and bounded. Our analysis result helps understanding the necessity of CMO schemes, and suggests a probabilistic approach to CMO: deferred-sending. The effect of deferred-sending is analyzed.

1. Introduction

The nondeterministic nature of distributed systems, *i.e.*, asynchronous process execution speeds and unpredictable communication delays, is the major factor that complicates the design, verification, and analysis of distributed systems. *Causal message ordering*, henceforth referred to as CMO, is an ordering imposed on message deliveries to reduce system nondeterminism while retaining concurrency. In systems preserving CMO, messages directed to the same destination are delivered in an order consistent with their potential causality. The causality under consideration is determined by the *happens-before* relationship [13] but is restricted to message sending and receiving events. Specifically, if a message-sending event happens before the sending of another message, the former message is considered to have the potential for affecting the latter in a causal way, and therefore must be received before the latter, if they are destined for the same process, to retain their cause-effect relationship. CMO is considered important to reliable distributed systems [11, 7, 9]. It can also be used to simplify the design of distributed algorithms [1, 4]. Many implementations and extensions of CMO have been done in distributed shared memory system [3], multimedia systems [6, 2], and mobile computing systems [5, 16, 19].

In this paper, we present a modular decomposition of CMO, and analyze the probability of breaking CMO. The analysis is based on two probabilistic models on communication delays: exponential and uniform distributions. These two distributions respectively represent the contexts where communication delays are unbounded and bounded. The result suggests a simple approach to reducing the probability of CMO breakdown: deferred-sending.

Rest of this paper is organized as follows. Section 2 presents the definition of CMO and a modular decomposition of it. Section 3 analyzes the probability of breaking CMO in ordinary contexts and Section 4 derives the expected delay with deferred-delivery approaches. In Section 5, we present our probabilistic approach to CMO and evaluate its performance. Section 6 concludes this paper.

2. A modular decomposition of CMO

2.1. System model and definition

An asynchronous distributed system is a collection of processes that communicate asynchronously by means of message-passing. No shared memory or global clock is available. The message latency is arbitrary.

An event is an atomic operation that changes the state of a process. Three types of events are considered in distributed systems: *internal events*, *sendings of messages*, and *receipts of messages* [15]. The *happens-before* relation (denoted by "→") on the set of events is the smallest transitive relation satisfying the following conditions [13]: (1) if event a and event b occur in the same process and if a comes right before b then $a \rightarrow b$; (2) if a is the sending of message m and b is the receipt of m, then $a \rightarrow b$.

Let *sent*(m) and *recv*(m) be the events that correspond to the sending and receipt of message m, respectively. CMO is obeyed if, for any two messages m and m' that have the same destination, *sent*$(m) \to$ *sent*(m') implies *recv*$(m) \to$ *recv*(m').

A message is said to be *received* by a site when it arrives at that site. A message is said to be *delivered* by a site when it is formally accepted and disposed of by the associated application running at that site. A distributed computing system, in which CMO does not hold with respect to sending and receiving events, can employ a scheme to enforce CMO with respect to sending and delivering events. That is, let *deliv*(m) represent the event of delivering message m, a CMO scheme can ensure that *sent*$(m) \to$ *sent*(m') always implies *deliv*$(m) \to$ *deliv*(m').

2.2. *CMO*(k)

The general CMO can be decomposed according to the number of messages involved in the causal event chain between two message sending events. A *causal event chain* is a sequence of events $\{e_1, e_2, \ldots, e_r\}$, where $r \geq 2$, such that $e_1 \to e_2, e_2 \to e_3, \cdots, e_{r-1} \to e_r$. Let $\Phi(a, b)$ denote the set of all possible causal event chains starting at event a and ending with event b. A *causal message chain* contained in $\Phi(a, b)$ is a sequence of messages $\{m_1, m_2, \ldots, m_l\}$ such that the sequence of events $\{a$, *sent*(m_1), *recv*(m_1), *sent*(m_2), *recv*(m_2), $\ldots$, *sent*(m_l), *recv*(m_l), $b\}$ is in $\Phi(a, b)$. This message chain is said to have length l.

Definition 1 A message m' *transitively depends on* another message m *with degree* k if *sent*$(m) \to$ *sent*(m') and the maximal length of any causal message chain contained in $\Phi($*sent*$(m),$ *sent*$(m'))$ is k.

If *sent*$(m) \to$ *sent*(m') but there is no causal message chain contained in $\Phi($*sent*$(m),$ *sent*$(m'))$, we say that m' transitively depends on m with degree zero. This happens when m and m' are sent by the same process.

Definition 2 *Causal message ordering of degree k*, denoted by *CMO*(k), is preserved if *deliv*$(m) \to$ *deliv*(m') holds for any two messages m and m' such that m' transitively depends on m with degree k.

Note that *CMO*(0) is essentially the FIFO ordering. As an example of *CMO*(1), consider the scenario shown in Figure 1. Message m_3 transitively depends on m_1 with degree 1. Therefore, if m_3 arrives at P_2 before m_1, *CMO*(1) is not preserved.

3. Probabilistic analysis of CMO in ordinary contexts

CMO is not necessarily preserved in ordinary contexts. In this section, we evaluate the probability of CMO breakdown in contexts where no CMO scheme is employed. The obtained result can help understanding the necessity of CMO schemes.

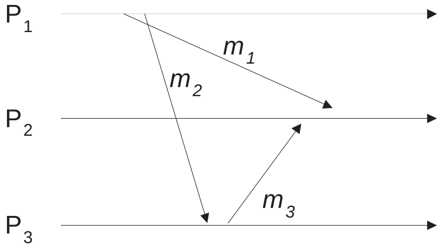

Figure 1. A scenario illustrating *CMO*(1)

Let us start with the case of *CMO*(1). Consider Figure 1 again. Let X_i $(i = 1, 2, 3)$ denote m_i's message delay. It can be seen that $X_1 > X_2 + X_3$ is a necessary condition for m_3 to arrive at P_2 before m_1. It follows that $Pr[X_1 > X_2 + X_3]$ is a upper bound of the probability that *CMO*(1) is not preserved. Extending this argument, we can see that $Pr[X_1 > X_2 + X_3 + \cdots + X_{k+2}]$, for some integer $k \geq 0$, represents the maximal probability that *CMO*(k) is violated. Let $Y = X_2 + X_3 + \cdots + X_{k+2}$ and $Z = X_1 - Y$. In the following, we will obtain the distribution of Z by assuming two typical random distributions on X_i: exponential distribution and uniform distribution. The former represents the case of unbound message delay while the later represents the case of bounded message delay.

Exponential distribution: X_i's are independent, identically distributed random variables with probability density function $f(x) = \alpha e^{-\alpha x}$ over the range $[0, \infty)$. It is known [10] that Y is a $(k+1)$-Erlang distributed random variable with probability density function

$$g(y) = \frac{\alpha^{k+1}}{k!} y^k e^{-\alpha y}$$

Therefore,

$$\begin{aligned} Pr[Z > z] &= Pr[X_1 > Y + z] \\ &= \int_{y=0}^{\infty} \int_{x_1=y+z}^{\infty} f(x_1) g(y) dx_1 dy \\ &= \int_{y=0}^{\infty} e^{-\alpha(y+z)} \cdot \frac{\alpha^{k+1}}{k!} y^k e^{-\alpha y} dy \\ &= \frac{1}{2^{k+1}} \cdot e^{-\alpha z} \end{aligned}$$

The upper bound of the probability that *CMO*(k) does not hold is therefore $Pr[Z > 0] = 1/2^{k+1}$. We also have the probability distribution function of Z, $F(z) = Pr[Z \leq z] = 1 - Pr[Z > z] = 1 - 1/2^{k+1} \cdot e^{-\alpha z}$, and the probability density function of Z, $f(z) = \frac{d}{dz} F(z) = \alpha / 2^{k+1} \cdot e^{-\alpha z}$. Figure 2 dipicts $f(z)$ with $\alpha = 1/8$ for $k = 1$ to 4.

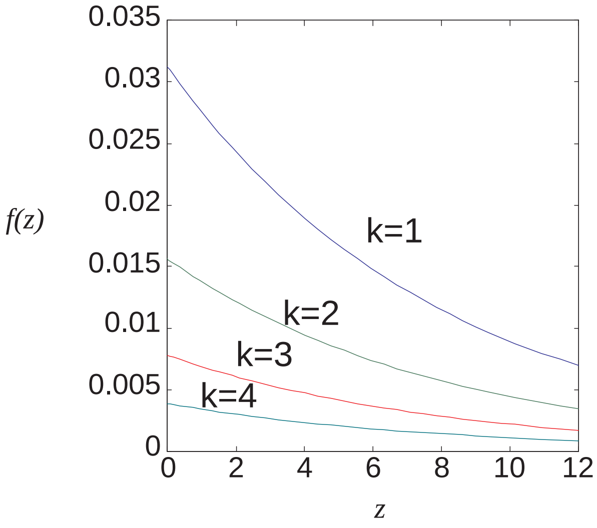

Figure 2. Probability density function for Z with $\alpha = 1/8$

Uniform distribution: X_i's are independent, identically distributed random variables with probability density function $f(x) = \frac{1}{b-a}$ over the range $[a, b]$. In the following, we derive $Pr[Z > 0]$ but not the probability density function of Z due to space limitation. Extending the result to derive the latter is straightforward. We know that $Pr[Z > 0] = Pr[X_1 > Y] = Pr[X_1 > Y|Y > b] + Pr[X_1 > Y|Y \leq b]$. When $Y > b$, it is impossible that $X_1 > Y$. Also, when $b \leq (k+1)a$, it is impossible that $Y \leq b$. It follows that

$$\begin{aligned} Pr[Z > 0] = Pr[X_1 > Y] &= Pr[X_1 > Y|Y \leq b] \\ &= \begin{cases} C & \text{if } b > (k+1)a \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where C is the value of the integral taken over the region determined by $(k+1)a \leq Y \leq b$. For $k = 1$, we have

$$\begin{aligned} C &= \int_{x_3=a}^{b-a} \int_{x_2=a}^{b-x_3} \int_{x_1=x_2+x_3}^{b} f(x_1)\, f(x_2)\, f(x_3)\, dx_1\, dx_2\, dx_3 \\ &= \frac{(b-2a)^3}{6(b-a)^3} \end{aligned}$$

Since $\frac{b-2a}{b-a} \leq 1$ for all positive settings of a and b such that $b > 2a$, the upper bound of $Pr[Z > 0]$ is $\frac{1}{6}$ (this happens when $b \gg a$).

Applying the same analysis technique, we can derive the probabilities of breaking *CMO*(k) for all $k \geq 2$. Table 1 lists the derived result. We can see that as k increases, the probability of breaking *CMO*(k) decreases.

4. Analyzing the deferred-delivery approach

Conventional CMO protocols take either piggybacking or deferred-delivery approaches. In the piggybacking approach [7], a message carries a history of all the messages that causally precede it. Thus when a message m is delivered to a process P, copies of all messages addressed to P that precede m also arrive with m or have arrived earlier. This scheme is straightforward and resilient to process failures. However, it requires a complex mechanism to prevent unbounded growth of the information added to messages. In deferred-delivery approaches [18, 8, 17], a received message m will be delivered to process P only if all messages that causally preceded m and were destined for P have already been delivered. Otherwise, message m is not delivered immediately but is buffered until the condition stated above is satisfied.

The probability density functions derived in the previous section can be used to compute the expected delay imposed on a received message with deferred-delivery approaches. In case of exponentially distributed communication delays, when a received message should be deferred to respect CMO, the expected delay is

$$\begin{aligned} E(Z) &= \int_{z=0}^{\infty} z f(z) dz \\ &= \int_{z=0}^{\infty} \frac{\alpha z}{2^{k+1}} e^{-\alpha z} dz \\ &= \frac{1}{\alpha 2^{k+1}} \end{aligned}$$

which indicates that if the delivery of a received message should be deferred, the expected delay time is proportional to the mean of communication delays (i.e., $1/\alpha$) times $1/2^{k+1}$. The case of uniformly distributed communication delays is analogous but more tedious, and is therefore omitted here.

5. The deferred-sending approach to CMO

Deferred-delivery CMO approaches assume finite communication delays, since otherwise they will fail to meet "liveness" requirement. Consider Figure 1. If m_1 is delayed infinitely, P_2 will defer the deliveries of m_3 and all subsequent messages from P_3, effectively breaking the communication channel from P_3 to P_2. Moreover, the delivery of any subsequently message from P_1 to P_2 will also be deferred infinitely.

The assumption of finite communication delays implies that deferred-delivery CMO approaches are only applicable to synchronous systems [14], where fixed upper bound on communication delays exists and, timers and time-out values can be used to detect the existence of lost or delayed

Table 1. Values of $Pr[Z > 0]$ and probability density functions for $k = 0$ to 3

	$Pr[Z > 0]$	Probability density function
CMO(0)	$\frac{1}{2}$	$\frac{-(z-b+a)}{(b-a)^2}, z \leq b - a$
CMO(1)	$\begin{cases} \frac{(b-2a)^3}{6(b-a)^3} & \text{if } b \geq 2a \\ 0 & \text{otherwise} \end{cases}$	$\frac{(z-b+2a)^2}{2(b-a)^3}, z \leq b - 2a$
CMO(2)	$\begin{cases} \frac{(b-3a)^4}{24(b-a)^4} & \text{if } b \geq 3a \\ 0 & \text{otherwise} \end{cases}$	$\frac{-(z-b+3a)^3}{6(b-a)^4}, z \leq b - 3a$
CMO(3)	$\begin{cases} \frac{(b-4a)^5}{120(b-a)^5} & \text{if } b \geq 4a \\ 0 & \text{otherwise} \end{cases}$	$\frac{(z-b+4a)^4}{24(b-a)^5}, z \leq b - 4a$

messages. In asynchronous systems where no such fixed upper bound exists, deferred-delivery approaches may fail.

We suggest a simple strategy that is applicable to both synchronous and asynchronous systems. It can be incorporated with any deferred-delivery CMO approaches, as well as be used as a stand-alone method under some condition. The basic idea is simple: defer sending a message to reduce the probability of breaking CMO. Several implementations are possible. We may hold sending a message

A) until d units of time elapses,

B) if any message has been sent in the previous d units of time,

C) if any message has been delivered in the previous d units of time, or

D) if any message has been sent or delivered in the previous d units of time.

Let us choose implementation C hereafter. Consider the scenario in Figure 1 again. The probability of breaking *CMO*(1) now becomes $Pr[X_1 > X_2 + X_3 + d]$. Intuitively, $Pr[X_1 > X_2 + X_3 + d] < Pr[X_1 > X_2 + X_3]$. In general, the probability of breaking *CMO*(k) will be less than $Pr[X_1 > X_2 + X_3 + \cdots + X_{k+2} + kd]$.

To evaluate the effect of our approach in asynchronous systems, we apply the same exponential distribution model. We have

$$\begin{aligned} Pr[X_1 > Y + kd] &= \int_{y=0}^{\infty}\int_{x_1=y+kd}^{\infty} f(x_1)g(y)dx_1dy \\ &= \int_{y=0}^{\infty} e^{-\alpha(y+kd)} \cdot \frac{\alpha^{k+1}}{k!}y^k e^{-\alpha y}dy \\ &= \frac{1}{2^{k+1}}e^{-k\alpha d} \end{aligned}$$

Note that $1/\alpha$ is the mean of X_i. The result indicates that if we hold a message to be sent for an amount of time equal to the mean of message delay, the probability of breaking *CMO*(1) will be less than $1/4e \simeq 0.1$. If the holding time is increased to four times the mean message delay, the probability drops to less than $1/4e^4 \simeq 0.005$.

The effect of deferred-sending in synchronous systems can be evaluated by applying the same uniform distribution model.

$$Pr[X_1 > Y + kd] \quad = \quad Pr[X_1 > Y + kd | Y + kd \leq b]$$

Since $Y + kd \geq (k+1)a + kd$, if $(k+1)a + kd > b$, we have $Pr[Y + kd \leq b] = 0$ and thus $Pr[X_1 > Y + kd] = 0$. It follows that if $d > (b - (k+1)a)/k$, *CMO*(k) is preserved. It is not difficult to extend the result to a more general rule:

Theorem 1 In context where the upper and lower bounds of communication delay are, respectively, a and b, *CMO*(k) will be preserved if $d > (b - (k+1)a)/k$, for all $k \geq 1$. (The communication delay is not necessarily a uniform distribution)

In case when $(k+1)a + kd \leq b$, $Pr[X_1 > Y + kd]$ can be computed in the same way as we compute C in the previous section. For example, when $k = 1$, we have $Pr[X_1 > X_2 + X_3 + d] = Pr[X_1 > X_2 + X_3 + d | X_2 + X_3 + d \leq b]$, which is equal to

$$\int_{x_3=a}^{b-d-a}\int_{x_2=a}^{b-d-x_3}\int_{x_1=x_2+x_3+d}^{b} f(x_1)f(x_2)f(x_3)dx_1dx_2dx_3$$

The value of $Pr[X_1 > X_2 + X_3 + d]$ is thus $(b - d - 2a)^3/6(b-a)^3$. The probabilities of breaking *CMO*(k) for all $k \geq 2$ can be derived by the same analysis technique. Table 2 shows some results. Compared with Table 1, it can be seen that except for *CMO*(0), the probabilities of breaking CMO are reduced.

6. Conclusions

For exponential and uniform distributions of communication delays, we have computed the probability of violating CMO without any control and with deferred-sending

Table 2. Probability result with our approach

	Exponential	Uniform
CMO(0)	$\frac{1}{2}$	$\frac{1}{2}$
CMO(1)	$\frac{1}{4}e^{-\alpha d}$	$\begin{cases} \frac{(b-d-2a)^3}{6(b-a)^3} & \text{if } b \geq 2a+d \\ 0 & \text{otherwise} \end{cases}$
CMO(2)	$\frac{1}{8}e^{-2\alpha d}$	$\begin{cases} \frac{(b-2d-3a)^4}{24(b-a)^4} & \text{if } b \geq 3a+2d \\ 0 & \text{otherwise} \end{cases}$
CMO(3)	$\frac{1}{16}e^{-3\alpha d}$	$\begin{cases} \frac{(b-3d-4a)^4}{120(b-a)^4} & \text{if } b \geq 4a+3d \\ 0 & \text{otherwise} \end{cases}$

control. In asynchronous systems where no fixed upper bound on message delays exists, conventional CMO approaches may fail to meet “liveness” requirement, while deferred-sending may break CMO. In synchronous systems where fixed upper bound on message delays exists, deferred-sending can totally preserve CMO (Theorem 1), if the bounds of communication delay are known.

It has been proven [12] that $O(n^2)$ message overhead is required for any deferred-delivery approaches to enforce *CMO*(k), where $k \geq 2$ and n is the total number of processes in a system. Such an overhead can be costly when n is large. Moreover, in many applications the number of processes participating in the computation may change from time to time. Conventional algorithms [18, 8, 17] seldom, if ever, deal with such dynamic participation.

Unlike conventional approaches, the deferred-sending method needs neither costly control information nor piggybacked messages in every message for CMO. This makes the method a bandwidth-efficient approach. Furthermore, since no information pertaining to the number of participating processes should be maintained, it is also resilient to dynamic process participation. On the other hand, the cost is that message delays are increased, and that CMO may not be preserved under some condition.

It seems that the optimal solution relies on integrating these two approaches. It is well known that *CMO*(0) (FIFO-order communications) can be easily achieved with the help of $O(1)$ message sequence numbers. For *CMO*(1), an $O(n)$ deferred-delivery CMO approach has been proposed [12]. If deferred-sending is merely used to cope with *CMO*(k) for $k \geq 2$, the cost incurred by deferred-sending can be reduced while $O(n)$ message overhead can be retained.

References

[1] A. Acharya and B. R. Badrinath. Recording distributed snapshots based on causal order of message delivery. *Inform. Process. Lett.*, 44:317–321, December 1992.

[2] F. Adelstein and M. Singhal. Real-time causal message ordering in multimedia systems. In *Proceedings of the 15th International Conference on Distributed Computing Systems*, pages 36–43, June 1995.

[3] M. Ahamad, P. Hutto, and R. John. Implementing and programming causal distributed memory. In *Proceedings of the 11th International Conference on Distributed Computing Systems*, pages 274–281, 1991.

[4] S. Alagar and S. Venkatesan. An optimal algorithm for distributed snapshots with causal message ordering. *Inform. Process. Lett.*, 50:311–316, 1994.

[5] S. Alagar and S. Venkatesan. Causal ordering in distributed mobile systems. *IEEE Trans. Comput.*, 46(3):353–361, March 1997.

[6] R. Baldoni, A. Mostefaoui, and M. Raynal. Causal delivery of messages with real-time data in unreliable networks. *Real-Time System*, 10(3):245–262, 1996.

[7] K. Birman and T. Joseph. Reliable communications in the presence of failures. *ACM Trans. Comput. Syst.*, 5(1):47–76, February 1987.

[8] K. Birman, A. Schiper, and P. Stephenson. Lightweight causal and atomic group multicast. *ACM Trans. Comput. Syst.*, 9(3):272–314, 1991.

[9] K. P. Birman and T. A. Joseph. Exploiting replication in distributed systems. In S. Mullender, editor, *Distributed Systems*. Addison-Wesley, New York, 1989.

[10] E. R. Dougherty. *Probability and Statistics for the Engineering, Computing, and Physical Sciences*, chapter 5, page 228. Prentice Hall, 1990.

[11] T. Joseph and K. Birman. Low cost management of replicated data in fault-tolerant distributed systems. *ACM Trans. Comput. Syst.*, 4(1):54–70, 1986.

[12] A. D. Kshemkalyani and M. Singhal. Necessary and sufficient conditions on information for causal message ordering and their optimal implementation. *Distrib. Comput.*, 11:91–111, 1998.

[13] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Comm. ACM*, 21(7):538–565, July 1978.

[14] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1997.

[15] F. Mattern. Virtual time and global states of distributed systems. In M. C. et al., editor, *Proceedings of the International Workshop on Parallel and Distributed Algorithms*, pages 215–226, North-Holland, 1989. Elsevier Science.

[16] R. Prakash, M. Raynal, and M. Singhal. An adaptive causal ordering algorithm suited to mobile computing environments. *J. Parallel Distrib. Comput.*, 41:190–204, 1997.

[17] M. Raynal, A. Schiper, and S. Toueg. The causal ordering abstraction and a simple way to implement it. *Inform. Process. Lett.*, 39:343–350, September 1991.

[18] A. Schiper, J. Eggli, and A. Sandoz. A new algorithm to implement causal ordering. In *Proceedings of the 3rd International Workshop on Distributed Algorithms*, 1989. Also published in *Lecture Notes in Computer Science*, 392.

[19] L.-H. Yen, T.-L. Huang, and S.-Y. Hwang. A protocol for causally ordered message delivery in mobile computing systems. *Mobile Networks and Applications*, 2(4):365–372, 1997.