

Commit Protocols in Distributed Database System: A Comparison

Manoj Kumar

Assistant Professor

BBKDAV College for Women, Amritsar-India -143001

Abstract

Distributed database systems use commit protocols to ensure atomicity in case of distributed transactions and a wide variety of commit protocols have been proposed so far. Two-phase commit (2PC) is a widely accepted standard protocol for distributed transactions. But 2PC suffer from blocking problem due to which three-phase commit (3PC) protocol was proposed which is an extension of 2PC. This paper present a review and comparison of commit protocols used for distributed database systems.

Keywords: Distributed database systems, commit protocols, atomicity, two-phase commit (2PC), distributed transactions, blocking, and three-phase commit (3PC)

I. INTRODUCTION

The distributed database systems are becoming popular day by day due to emerging trends in information processing and are become a hot area of research for upcoming researchers. The distributed systems execute application on multiple autonomous computers connected through communication networks. Application on distributed systems are coordinated and controlled using distributed protocols. The transaction concept is very important for design of fault tolerant distributed applications [2][3]. When a transaction happens in distributed database it must satisfy the ACID properties but in distributed environment a transaction may decide to commit at some sites and abort at other sites resulting in violation of atomicity property [survey of commit] to overcome this problem commit protocols are used.

Distributed database systems use distributed commit protocols to ensure the atomicity across the sites, when a transaction execute at multiple sites it must either be committed at all the sites or aborted at all the sites instead a transaction committed at one and aborted at another.[4].One phase(1PC) protocol was the first one proposed by Jim Gray[Gray 78] but due its number of shortcomings, two-phase commit (2PC) was proposed which is a widely accepted standard protocol for distributed transactions. 2PC has a blocking problem to overcome this three-phase commit (3PC) was proposed which is an extension of 2PC. 3PC has extra phase called pre-commit which make this protocol a non-blocking but this causes transfer of extra messages [5]. The rest of the paper organized as follows: Section II describes distributed system architecture, section III introduces two-phase commit and three-phase commit protocols and section IV presents comparison among 2PC and 3PC.

II. DISTRIBUTED DATABASE SYSTEM ARCHITECTURE

In distributed database system a transaction may access data from several locations (sites), each site has a local transaction manager (TM) responsible for maintaining a log for recovery purpose and a transaction coordinator (TC) used for execution of transaction originates at the site, distributing the transaction at appropriate sites and coordinating the execution of transaction at multiple sites. Fig. 1 is showing typical distributed database system architecture

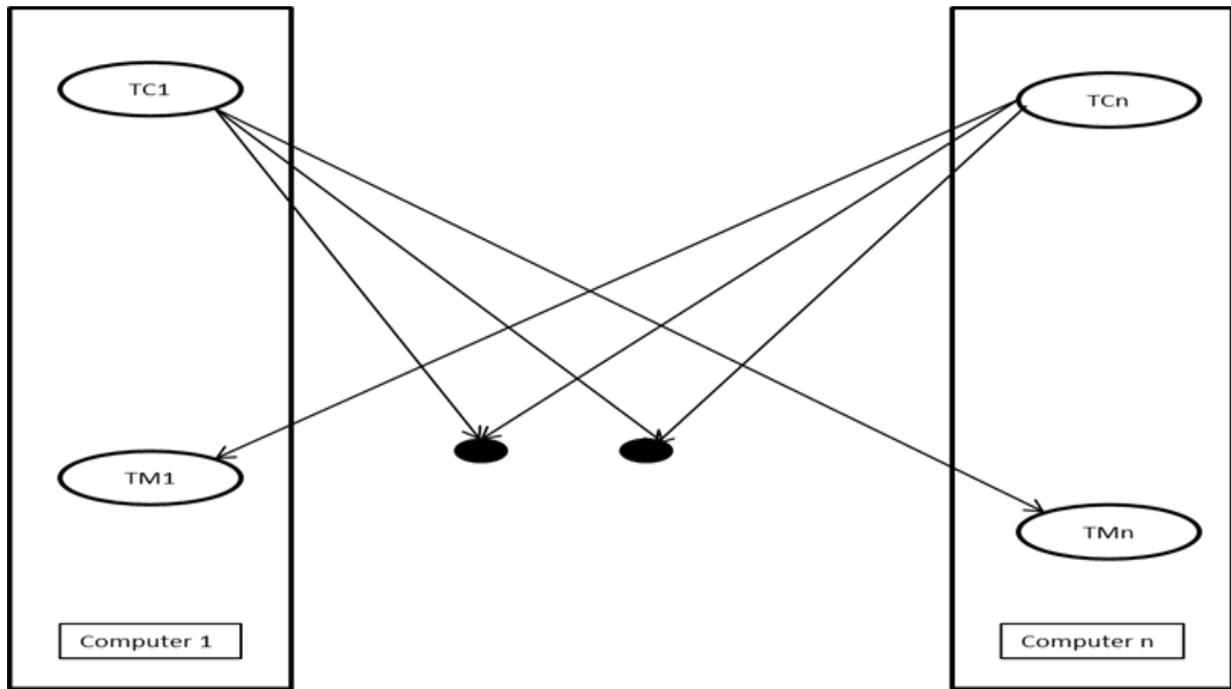


Fig. 1: Distributed database system architecture

III. COMMIT PROTOCOLS

A. Two-Phase Commit (2PC) Protocol

The correctness of distributed commit protocol lies in its atomicity. A distributed transaction running on distributed system is set of independent transactions executing at different sites [1]. Transaction is initiated by a coordinator, participating sites execute the transaction on behalf of coordinator and commit/abort as directed by the coordinator. The two-phase commit protocol is a more general version of the commit protocol and most common regarding atomic communication in distributed system. The protocol is based on the principal that all the participating sites must agree to commit before the transaction is executed if any of the site vote against the commit then the transaction is aborted for all the sites [6]. 2PC has two phases voting and commit phase.

1) Voting phase

- 1) Coordinator sends a Prepare message along with the transaction to all participants and asks each one of them to cast their vote for commit or abort.
- 2) If participant can commit the transaction Vote-commit is send to the coordinator and if participant cannot commit Vote-abort is send to the coordinator.

2) Commit phase

- 3) Decision for commit or abort is taken by the coordinator in this phase. If Vote-commit is received from all the participants then Global-commit is send to all the participants and if at least one Vote-abort is received then coordinator send Global-abort to all those voted for commit.
- 4) Coordinator ask for acknowledgement (Ack) from participants. If a participant receives Global-commit, it commit the transaction and Ack is send to the coordinator. In case participant receives Global-abort it abort the transaction. Fig.2 shows two-phase commit protocol

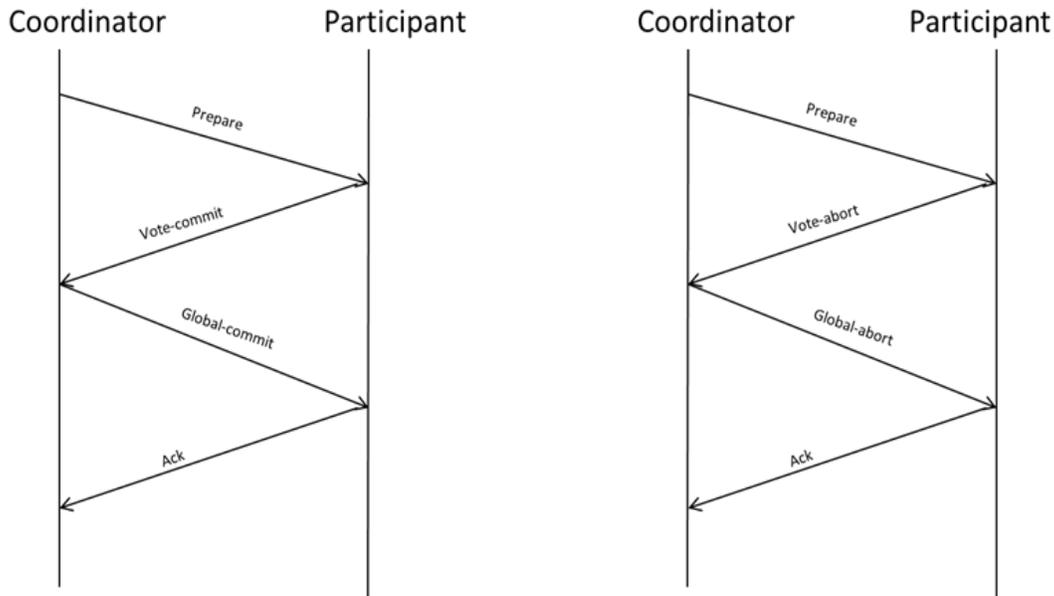


Fig. 2: Two-phase commit process

Two-phase commit state transition of coordinator and participant is shown in Fig 3. Coordinator send prepare message to all the participants and enter into wait state to indicate that it waiting for response of the participants, if coordinator receives vote-commit from all the participants it goes to commit state and send global-commit to all participants, if at least one vote-abort is received coordinator goes to abort state and send global-abort to all participants. A participant receives prepare message and if it can commit the transaction, it goes to ready state otherwise unilateral abort is done.

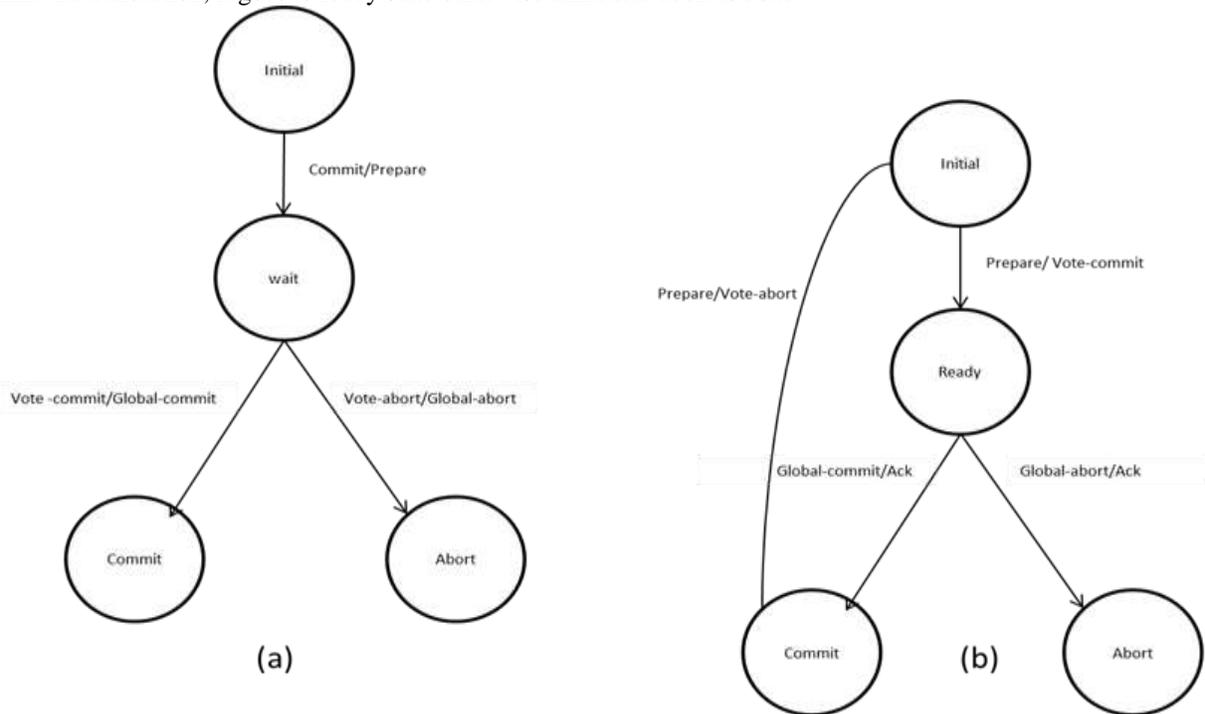


Fig. 3: State transition of (a) Coordinator (b) participant in 2PC [8]

Two-phase commit protocol ensures atomicity and can handle network failures .But it suffers due to blocking of participant site in case of coordinator failure, increased latency due to forced writes of logs and more communication overhead as compared to simple optimistic protocol.

B. Three-Phase Commit Protocol (3PC)

Three-phase commit protocol is an extension of 2PC which avoid blocking by introducing additional round of message exchange and delaying the prepared state until processes receive pre-commit message. Unlike 2PC, 3PC does not immediately commit if all

participants send vote-commit instead coordinator sends out prepare-to-commit message, on receiving this message participants enter into Pre-commit state and send an acknowledgement .After receiving acknowledgement from all participants coordinator sends commit and participants commit the transaction.3PC has an extra phase called pre-commit along with voting and commit phase.

1) *Voting phase*

- 1) Coordinator sends a Prepare message along with the transaction to all participants and asks each one of them to cast their vote for commit or abort
- 2) If participant can commit the transaction Vote-commit is send to the coordinator and if participant can not commit Vote-abort is send to the coordinator.

2) *Pre-commit phase*

- 3) Coordinator waits for replies from all participants, if all participants have send Vote-commit then coordinator sends Prepare-to-commit to all. If any of participant has send Vote-abort, coordinator sends Global-abort to all those who have sent Vote-commit.
- 4) Participants wait for reply from coordinator, if Prepare-to-commit is received participants enter into Pre-commit state and send Ready-to-commit to coordinator.

3) *Commit phase*

- 5) Coordinator waits for replies from participants, if all participants send Ready-to-commit coordinator sends Global-commit to all participants. If coordinator does not receives Ready-to-commit message from a participant within pre-defined time , it concludes that participant has voted for commit earlier but it unavailable due to some reason so coordinator sends Global-commit to all the remaining participants.[8]
- 6) On receiving Global-commit participants commit the transaction and send acknowledgment to coordinator.

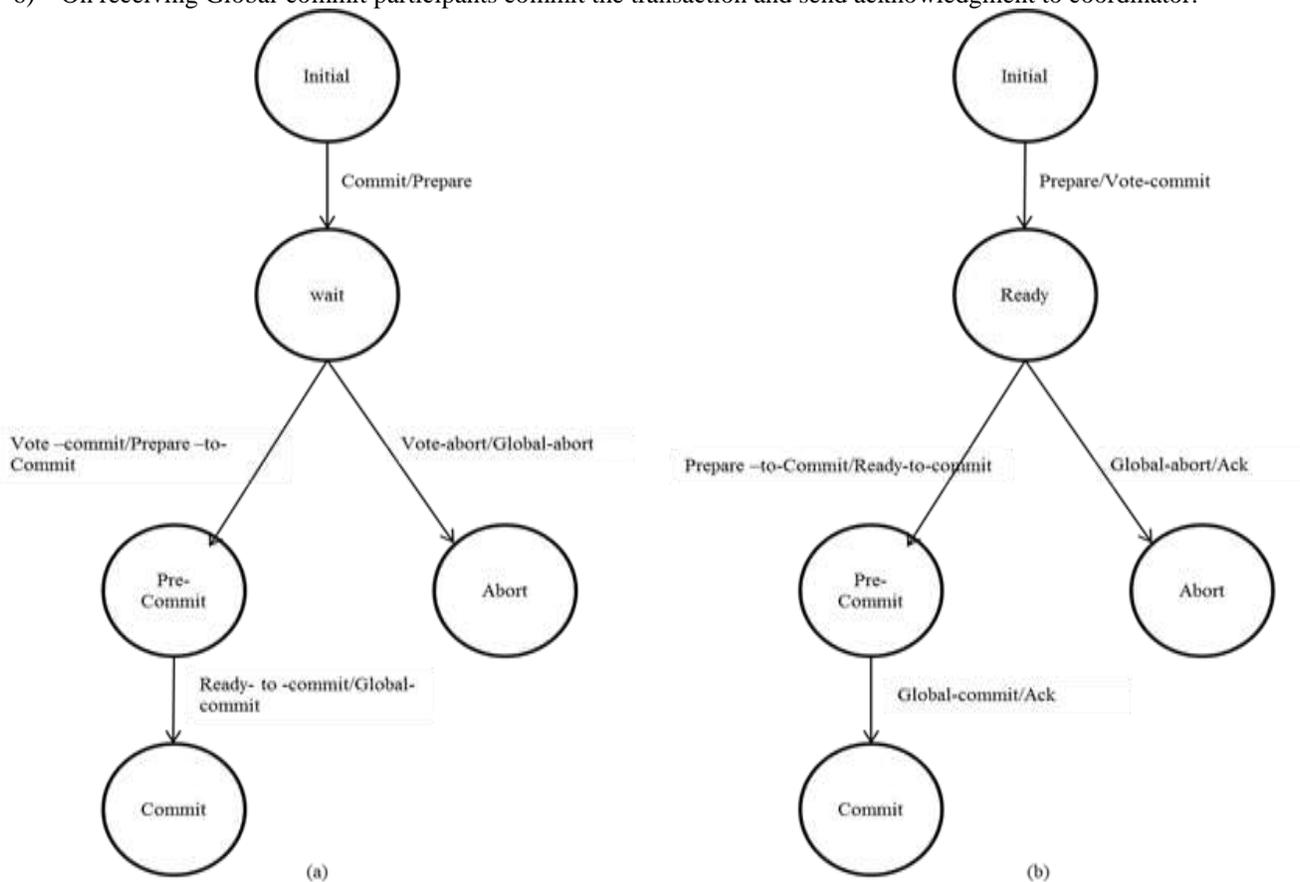


Fig. 4: State transition of (a) Coordinator (b) participant in 3PC [8]

IV. COMPARISON BETWEEN 2PC AND 3PC

Two-phase commit and three-phase commit protocols can be compared on following criteria.

1) *Blocking:*

Two-phase commit protocol causes blocking of participants site when coordinator site fails. But three-phase commit avoid blocking by adding an extra phase called Pre-commit.

2) *Message Exchange:*

2PC has $4(n-1)$ messages exchange comprises of $n-1$ messages in Vote-request, $n-1$ in local decision for commit or abort, $n-1$ in Global-commit/Global-abort and $n-1$ messages exchange for ack. 3PC commit protocol causes $5(n-1)$ messages to exchange as compare to $4(n-1)$ in 2PC. The extra $n-1$ message exchanges in 3PC are due to extra phase in 3PC.

3) *Communication Overhead:*

3PC has more communication overhead due to an extra phase as compared to 2PC protocol.

4) *Log Writes:*

2PC and 3PC both have $2n$ log writes

5) *Complexity:*

3PC protocol is more complex and costly to implement as compare to 2PC protocol

6) *Performance:*

3PC protocol has more message exchanges due which less performance as compared to 2PC. But performance of 3PC is more in case failure of coordinator site .

V. CONCLUSION

Designing a efficient commit protocol is very important for distributed database systems as Commit protocols are required to ensure the atomicity of a distributed transaction. 2PC is commonly used commit protocol in distributed database system transactions but 2PC suffers due to blocking of participants sites in case of failure of coordinator site. 3PC commit protocol was proposed to eliminate the blocking but has its own problems like more communication overhead due to an extra phase. This paper has compared 2PC and 3PC based on some important parameters like message exchange, log writes, complexity and performance. In future these commit protocols can be compared with other non-blocking protocols.

REFERENCES

- [1] S. Ceri, M.A.W. Houtsma, A.M. Keller, P. Samarati "A Classification of Update Methods for Replicated Databases" via Internet, May 5, 1994
- [2] M. Abdallah and P. Pucheral "A Low-Cost Non-Blocking Atomic Commitment Protocol for Asynchronous Systems" 1999.
- [3] Harit Sharma "Token-based Decentralized Non-blocking Two-phase Commit Protocol" IJIRST, vol 1. Issue 1, 2014
- [4] Muzammil Hasan " Transaction Commit Protocol in Mobile Environment: A Survey " IJETAE Vol.5 Issue 1,2015
- [5] Tabassum k, Taranum F, Damodaram A "A Simulation of Performance of Commit Protocols in Distributed Environment", in PDCTA, CCIS 203, 665–681, Springer, 2011
- [6] D. Skeen "Non-blocking commit protocols", ACM SIGMOD, June 1981
- [7] Tanenbaum A., Van Steen M. Distributed systems Principles and Paradigms, second edition, 2007
- [8] Ozsü, Tamer M., and Valduriez, Patrick, Principles of Distributed Database Systems Prenticehall, 1991
- [9] Toufik Taibi, Abdelouahab Abid, Wei Jiann Lim, Yeong Fei Chiam, and Chong Ting Ng, "Design and Implementation of a Two-Phase Commit Protocol Simulator", The International Arab Journal of Information Technology, Vol. 3, Issue.1, 2006.