



## Recovery from failures in Two Phase Commit Protocol (Distributed Transaction)

Last Updated : 29 Apr, 2021

**Prerequisite:** [Two-Phase Commit Protocol](#)

In the 2-phase commit protocol, the sites contributing to a distributed transaction and the coordinator that is managing the whole transaction globally may fail or crash, and this could lead to the whole transaction failure. Since unanimity is required in order to commit a distributed transaction successfully if any one of the sites fails, so the whole transaction will get aborted.

**Following kinds of failures could be encountered in the 2-phase commit protocol:**

**Failure of contributing site**– If the **coordinator(C)** detects that a site has crashed, so coordinator takes the following actions:

If the **site(Si)** has failed before responding with a **<ready T>** message to **Coordinator(C)**, the coordinator assumes that the site has responded with an **<abort T>** message.

If the site has failed after sending **<ready T>** message to **C**, so the coordinator will ignore the site failure and will execute the rest of the commit protocol in the usual manner.

When the failed **site (Si)** recovers from failure, so the site will examine its log record in order to know the destiny of the **transaction T**. whether it has failed or not-

- If the **log** contains a **<Commit T>** record, in this case, the site will execute **<redo T>**.
- If the **log** contains an **<abort T>** record, in this case, the site executes **<undo T>**.
- One most important case, if the **log** record contains the **<ready T>**, in this case, the site must contact the **Coordinator(C)**. But, if the **C** has failed itself then in this case the **Site(Si)** must consult the neighbouring sites and check their log status whether the **transaction(T)** has executed or aborted in its absence.
- If all the sites are unable to give an appropriate response then in this case the **Site(Si)** must wait for either coordinator recovery or appropriate response of the neighbouring sites.
- So, **Si** must enquire periodically about the **transaction T** fate by sending query messages to other sites.

If the log contains no record (abort, commit, ready) about **transaction T**, thus we know that **Si** has failed before responding to **<prepare T>** message from **Ci**. Hence, **Ci** must abort & execute **<undo T>**.

**Failure of Coordinator(Ci)**- If the coordinator fails in the midst of the execution of the **transaction T** in 2-phase commit protocol, then participating sites must decide the destiny of **transaction T**. In certain cases participating sites can't decide whether to commit or abort the **transaction T** and therefore these sites must wait for the recovery of the failed coordinator.

- If all sites contain a **<commit T>** record in their log, then **T** must be commit.
- If all the site contains a **<abort T>** record in their log, then **T** must be abort.
- If some sites don't contain **<ready T>** record in their log, then the failed **coordinator(C)** can't decide to **commit/abort**, so the site has not able to respond to the **<prepare T>** message of the coordinator hence it'll be better to **abort** the **transaction** rather than waiting.
- If none of the previous cases holds then all the active site has **<ready T>** record in their **log**, but no other record would be found since the coordinator failed it's impossible to determine whether a decision of **commit/abort** has been made unless the **coordinator(C)** recovers. So, all sites have to wait for the recovery of the **coordinator**. This situation is called the **Blocking Problem**.

**Network Partitioning-** is nothing but a kind of failure where the network connectivity is split between the partitions or nodes due to a failure. When the network partitioning occurs following two cases may occur:

- The coordinator and all the participating sites remain in one network partition, then the network failures have an impact on the commit protocol.
- If all participating sites and coordinator belong to two or more different partitions, so from the viewpoint of the site, the sites whose partition hasn't the coordinator has failed and starts recovery protocol while the other sites are simply executed whose partition containing the coordinator and follow the usual **2PC protocol**.

Reference: [Henry\\_korth](#)

Dreaming of M.Tech in IIT? Get AIR under 100 with our [GATE 2026 CSE & DA courses](#)! Get flexible **weekday/weekend** options, **live mentorship**, and **mock tests**. Access exclusive features like **All India Mock Tests**, and Doubt Solving—your GATE success starts now!

Comment

More info

Advertise with us

## Next Article

Two Phase Commit Protocol (Distributed Transaction Management)

## Similar Reads

### Two Phase Commit Protocol (Distributed Transaction Management)

Consider we are given with a set of grocery stores where the head of all store wants to query about the available sanitizers inventory at all stores in order to move inventory store to store to make balance over the...

5 min read

### Three Phase Commit Protocol

Three-Phase Commit (3PC) Protocol is an extension of the Two-Phase Commit (2PC) Protocol that avoids blocking problem under certain assumptions. In particular, it is assumed that no network partition occurs, and...

3 min read

### How do Database Systems Handle Transaction Failures?

Answer: Database systems handle transaction failures through two mechanisms, rollback and logging. Database systems handle transaction failures by leveraging the ACID (Atomicity, Consistency, Isolation, Durability)...

1 min read

### Two Phase Locking Protocol

The Two-Phase Locking (2PL) Protocol is an essential concept in database management systems used to maintain data consistency and ensure smooth operation when multiple transactions are happening...

9 min read

### Two Phase Locking (2-PL) Concurrency Control Protocol | Set 3

In database systems, managing concurrent transactions is essential to maintain data consistency and ensure smooth operation. Two-Phase Locking (2PL) is a widely used concurrency control protocol that enforces...