TRANSACTION FLOW DURING VBR

Problem 0.1. *"OK - so the thing to try to be very clear about is if after encountering a gap the recovery can ever switch back to non-VBR recovery. Also, it isn't clear what happens if the servers saw a few gaps, but power down and back up. It is possible that even when clients reconnect, they don't know anymore that there were gaps, yet it can affect sequence number recovery".*

1. Definitions

transno: - transaction number assigned to the operation. It is save on client side for replay. Used during recovery to process replays in order.

transaction sequence: - the sequence of transactions numbers that gives the order of replays. It is growing sequence with possible gaps.

version: - the last transaction the object was involved in. During each operation the old version of object is stored on client side for recovery. VBR checks that replay is applied on object with the same version as during original request handling

gap: - the missed replays producing the gap in transaction sequence. There may be allowed gaps which don't affect the recovery, e.g. open replays but missed client produces the gaps in transaction sequence that affect the followed replays.

ordinary recovery: - the recovery based on transactions only. All replays creates solid transaction sequence without gaps (except open replays). Any gap in transaction produces recovery failure.

version based recovery: - recovery process which uses versions of objects to control that object is changed in the same way as before failure. This recovery allows the gaps in transaction sequence but client will fail the recovery if version mismatch was seen.

mismatched replay: - replay failed due to version mismatch on server. The client recovery proceed to fill as match rep

2. Use Cases

2.1. Transition from VBR recovery back to ordinary one. The synopsis:

(1) the client missed the main recovery and VBR recovery was started on server
(2) the failure is occured during VBR and server start recovery again after reboot
(3) the missed client come back in the same time and is included in recovery process.
(4) The server finds all clients are participating in recovery and use ordinary recovery process instead of VBR

The question:

- Will this switching to the ordinary recovery be correct? Do we need to perform the VBR in such situation always?

The investigation:

(1) If all replays during VBR recovery were not committed then nothing was changed and recovery will proceed like there was no gap at all
(2) If some changes were committed already will that affect normal recovery flow?
   (a) the VBR was started with transaction greater then gap, so the missed client will be first one with replays and fill the gaps
   (b) all replays aplied during VBR were valid by versions
   (c) all replays failed during VBR due to versions should be repeated so there will be no gaps in transaction sequence and ordinary recovery may proceed
   (d) if the mismatched requests were not yet committed then they will be repeated and recovery is valid because gaps are closed.
   (e) else the client with mismatched requests will proceed with not yet committed replays but not all changes will be restored. **This is the issue, the ordinary recovery will lost some changes.**

The solution:

(1) Don't switch back to the ordinary recovery is VBR was started once. Store the flag that server is in VBR recovery persistently. Clear that flag when recovery complete. Therefore the VBR will not switch back to normal recovery after failure during recovery. This is simple to implement but not all clients will be reintegrated fully even if all are present.
(2) Don't drop mismatched requests from replay queue on last_committed value, therefore they may be repeated when VBR will switch to ordinary recovery. All successfully replayed requests were not depended on gap, all mismatched requests were not applied, so repeating them in transaction order will restore all data on server and all client will reintegrate. This case is better because allows to reintegrate fully for all clients if all is participating in recovery.