1 TX Manager Recovery Issues

2 1. Xid treatment

2.0.1 gtrid formulation

The current gtrid is not adequate. IP address as a component is documented, but not implemented. Implementing would not be good enough since IP addresses are not guaranteed to be unique within an enterprise due to the possibility of duplicate subnet addresses employed by routers to facilitate firewalls.

The issues involved with proper gtrid generation are listed on the JOTM forum and in the Mike Spillee article on XA.

2.0.2 bqual discrimination

Currently all bqual values for each branch are identical. This is not correct. They must all be distinct within a transaction. Furthermore, given the existence of inbound transactions, we have to be able to identify Xid's handed to an XAResource which we did not originally create. The bqual value is the only field we are allowed to modify on such an Xid as it is passed into our local XAResources. Therefore, a JOTM instance specific token must be embedded within the bqual value.

Also see the JOTM forum and in the Mike Spillee article on XA.

2.0.3 Omit bqual from hash key

Currently, transaction object identifiers are stored into the hash using the entire Xid as a key. During recovery, each XAResource will be returning the Xid given to it by this version of JOTM. Since each Xid will have a different bqual field per the previous item, it will be necessary to store an entry for each resource association OR we should store the transaction object identifier using a key which can be ascertained from the Xid given to all XAResources participating in the given transaction. The Xid absent the bqual field meets the criteria.

2.0.4 Add bqual field to resource association list in the transaction object

The Xid field originally given to an XAResource at the time of enlistment must be passed to that XAResource during subsequent calls such as COMMIT or ROLLBACK. Since the bqual field of the Xid field given to each XAResource will be different, it will be necessary to remember or compute the proper bqual field in order to make the proper call.

We could just save the entire Xid given to an XAResource with its entry in the resource list in the transaction object. This would tend to waste space. We should note that each bqual value must be available after crash. [We could wait until it is supplied by the subordinate XAResource on the RECOVER call, but reporting would be more complete if it could be remembered or computed in case the RECOVER call cannot be done.]

Unlike the gtrid, there is no danger of accidental reuse of a bqual value. This is, in fact, allowed as long as it is not within the same transaction. Therefore, a simple sequence number is adequate to discriminate different bqual fields within a transaction. This sequence number could be the offset of the XAResource entry in the resource association list in the transaction. If the number of XAResources is stored on the journal, then the list could be reconstituted with null values during journal recovery. The proper dispensation of each list entry would be required before the transaction entry could be forgotten. Note that in this proposed implementation, the actual bqual value is implied and does not require a field value. However, a status entry or sentinal value would be required to keep track of which XAResources had been successfully recovered.

2.0.5 Plan to report external Xid's only on RECOVER requests

The recovery of inbound transactions can lead to the superordinate node sending a RECOVER command. Any transaction originated on this node cannot be directed by a superordinate node, therefore, there is no need to report the Xid's of transactions which we originated. If the Xid's of inbound transactions were marked in the

tables and marked on the journal, then only those need to be reported during a RECOVER command. This would potentially greatly reduce everyone's processing time.

3 2. Journal Record Content

3.0.1 COMMIT & ROLLBACK records

Fields:

Type	- Record type; COMMIT, ROLLBACK
Xid	- XA transaction identifier; initial Xid including bqual for external TXs
Branch count	- Number of XAResources to be recovered (possibly implied in “Info”)
Flag	- Indicates Internal / External Xid
Info	- String to supply information for heuristics; Optional?; User assisted config?
	- Possibly a bit map referencing a separately maintained list of XAResources
	- Possibly a list of small integers referencing a separately maintained list

3.0.2 DONE record [Forget]

Fields:

Type	- Record type; DONE
RecID	- Record ID of record being completed; e.g., block # / offset
	Using record ID without Xid can reduce journal I/O bandwidth

4 3. Journal API

OPEN - New or Reposition

CLOSE

READRECORD or " Xidinfo[] Recover() " [Probably an upper layer]

PUTRECORD - synchronous and asynchronous

5 4. Modes of Recovery

5.1 XAResource failure

5.1.1 J2EE platform may continue to operate

5.1.2 What triggers JOTM to call RECOVER? When and what thread?

5.2 JOTM / Server crash w/ all resources on board

5.2.1 Fairly normal case; sort out TXs which had committed and rollback all others

5.2.2 What happens when an XAResource will not recover?

5.3 Distributed resource failure

5.4 JOTM / Server crash with distributed resources

5.5 JOTM / Server crash w/Inbound TXs

5.6 Inbound TXs with crash of superordinate node

6 5. Locating XAResources

6.1 XAResources are not serializable

6.2 What representation of the XAResource can be placed on the journal to allow later call

6.3 Can representation be independent of J2EE platform configuration?

This would allow JOTM to perform recovery without running J2EE platform.

Also useful for free-standing version of JOTM.

Isolation from J2EE platform reconfiguration which could interfere with needed recovery.

7 6. Sequencing of Events

7.1 How to delay JOTM -> XAResource RECOVER call until XAResource is ready.

7.2 Should JOTM employ a thread pool to manage RECOVER?

7.2.1 May help to solve absent or delayed XAResource recovery

7.2.2 Improves recovery time if those threads perform XAResource recovery rather than just reporting

7.3 Avoid starting new work until XAResource is recovered.

7.3.1 Not strictly necessary; may reduce resources needed and time taken for recovery

7.4 Performing recovery when J2EE platform would normally elect to abort.

7.4.1 In some cases the recovery of the XAResource will be far more important that the repair of the application platform.

7.4.2 Either the application platform should elect to stay "up" for recovery completion or JOTM should have a mode which allows it to recovery without the application platform support.

7.5 Performing recovery on a failover platform

7.5.1 Possibly free standing without application platform support.

7.5.2 Different schemes of remote data replication should be considered; e.g. remote RAID 1 or EMC SRDF

7.5.3 Be as capable as best XAResource. Oracle is likely to provide the benchmark

8 7. Design Points or Goals

8.1 10,000 TX / sec for an application platform

Clearly higher that required by most customers, but we do not want to be the bottleneck on any platform. We have been told that higher rates would be achieved by running multiple platforms running multiple JVMs.

8.2 Expect, but don't force, use of journal

8.2.1 Warning messages if journal is not configured.

8.2.2 Require explicit configuration switch to turn off recovery capability

8.3 Recovery is idempotent

8.3.1 Inbound RECOVER commands can be performed at any time any number of times.

8.3.2 Subordinate XAResources may behave differently if called unexpectedly ; see Black List below

8.4 XAResources do not all implement recovery equally

8.4.1 Some databases are known not to implement recovery

8.4.2 Some databases are known to stub out recovery

8.4.3 Some XAResources may be found to behave outside of standards

8.4.4 Adopt manual "blacklist" and "whitelist" approach employed by Linux SCSI device driver community

- Accommodates databases with known poor behavior
- Accommodates databases with unknown behavior
- User access to list accommodates XAResources otherwise unknown to JOTM developers
- User access to list accommodates XAResources whose behavior changes with updates
- Limits the endless investigation required to interoperate with every possible XAResource