04 – Multi-Version Concurrency Control

1. Microsoft Hekaton

   A. Timestamp management

      i. Each txn is assigned
         BeginTS (when it begins), CommitTS (when it commits)

      ii. Each record has two timestamp field
          BEGIN-TS : CommitTS of txn that created record
          END_TS : BeginTS of txn that created next version
          or CommitTS of txn that created record

   B. Version management

      i. Timestamp of uncommited version's first bit is set to 1 to check it is uncommited. First bit is excluded when compare it to check whether txn can see the version

      ii. Each txn can read uncommited version, but cannot update uncommited version.

   C. Transaction management

      i. Each txn is at specific "state" :
         ACTIVE, VALIDATING, COMMITED, TERMINATED

      ii. Txn meta-data
          Read set, Write set, Scan set, Commit dependencies

   D. Observations

      i. Validations are expensive for Analytical txn

      ii. O2N version chain is not good for OLAP, because of pointer chasing

      iii. Record-level conflict checks can occur false-positive aborts.

2. TUM HyPer
    A. Version management
        i. Main data table stores most recent version of records
        ii. Old versions are stored in delta storage, pointed by version vector
        iii. Delta storage is managed per txn, version chain is formed by just pointing next old version in storage.
        iv. Version synopses is used to skip pointer chasing when there are no other versions but only single main version.
    B. Validation
        i. First writer wins
        ii. Check the redo buffers of txns committed after validating txn beginned.
        iii. Precision locking is used to validating read set and scan set.

3. SAP HANA
    A. Version Management
        i. N2O storage, but store oldest version in main storage.
        ii. Every version has flag to indicate that there are newer versions

4. CMU Cicada
    A. Best-Effort Inlining : store meta-data in fixed location
    B. Validation
        i. Contention aware validation
        ii. Early consistency check
        iii. Incremental version search