05 – Multi-Version Concurrency Control

1. In-Memory T-Tree
   - A. Based on AVL Tree
     Storing pointer instead of key
   - B. Has many pointers, 2 value in node
     - i. Pointer : left/right child, parent pointer, data pointers
     - ii. Value : minimum range, maximum range
   - C. Advantage / Disadvantage
     - i. Advantages
       1. Less memory : no keys are stored in the index
       2. Can evaluate predicate and access tuple at same time
     - ii. Disadvantages
       1. Rebalancing is difficult
       2. Thread-safe implementation is difficult
       3. Pointer chasing hurts cache locality

2. Latch-Free Bw-Tree

   A. Cannot build Latch-Free B+Tree because split/merge operations update multiple pointers
      if there are indirection layer, it can be possible (Bw-Tree)

   B. Delta Chains

      i. Each update make new delta record
         delta record points to base page

      ii. Replace mapped pointer in mapping table
          with new delta records pointer

      iii. If contention occurs,
           mapped delta wins, losed updates should retry or abort

      iv. Consolidate delta chain to copy of base page,
          and replace mapped pointer with this

   C. Mapping Table

      i. Map page ID and its physical pointer

      ii. All page can replace pointer to other pages with its page ID

   D. Garbage collection

      i. Reference counting

         1. Maintain counter to indicate # of threads that accessing it

         2. If counter become zero, it can be GCed.

      ii. Epoch based reclamation

         1. Maintain global epoch

         2. Maintain object-local epoch that indicates when object deleted

         3. If all threads left that epoch, the object can be reclaimed.

E. Modifications

   i. Split delta record – delta for B Tree split operation

      1. Delay to update mapping table with split operation

   ii. Separator delta record – delta for search shortcut