02 – In-Memory Databases

1. Disk Oriented DBMSs

   A. Primary storage : non-volatile storage, organized as fixed length pages

   B. Use buffer pool to cache pages in memory

   C. Concurrency Control :
   All txn could stall if it tries to access data that is not in memory.

   D. Logging and recovery :
   most use STEAL + NO-FORCE policy. So it must maintain log records to keep track of modification of DB

   E. Overhead
   Buffer pool operation, locking and latching is very big overhead
   and real work overhead is only 7% of all overhead

2. In-Memory DBMSs

   A. Primary storage : permanently in memory
   possible because DRAM prices are low and capacities are high
   no need to be organized as fixed length pages
   make variable length data blocks and map it with fixed data blocks.

   B. Memory specific indexes' throughut is worse than B+Tree because of cache.

   C. Since all data are in memory,
   sequential scan is not better than random access

   D. Logging and recovery

      i. Still use WAL

      ii. No need to track LSN, since there are not dirty pages

   E. Overhead

      i. Overhead for disk I/O is eliminated

      ii. Network, data copy and move, cache miss are still overhead of system.

3. Concurrency Control Bottlenecks
    A. Findings from paper
        i. Allocating unique timestamp can be a bottleneck
        ii. Copying workspace in OCC is bottleneck
        iii. Deadlock detection can be a bottleneck if contention occur a lot
        iv. Because of large abort rate, no-wait and wait die don't work well
    B. Bottlenecks
        i. Lock thrashing
            1. One txn waits longer to acquire locks, other txn waits longer
        ii. Timestamp allocation
        iii. Memory allocation
            1. Copying data on all operations are big overhead