Abstract:

The emergence of large main memories and massively parallel processors has triggered the development of multi-core main-memory database management systems (DBMSs). Although the reduction of disk accesses helps the main-memory DBMSs shorten the single-thread execution time of on-line transactions, scaling these DBMSs on modern multi-core machines remains to be notoriously difficult. This is because processing massive amounts of concurrent transactions can confront several performance bottlenecks inherited from different DBMS components, and these bottlenecks altogether put a strict constraint on the scalability of the DBMSs.

In this thesis, we describe the design, implementation, and evaluation of multi-core main-memory DBMSs that achieve high performance for transaction processing. As concurrency control protocol is the central component for coordinating concurrent transactions, we first present the design of transaction healing, a robust scheme that exploits program semantics to scale the conventional optimistic concurrency control protocol towards dozens of cores even under highly contended on-line transaction processing (OLTP) workloads. To ensure durability property, the DBMSs have to frequently persist transaction logs into secondary storage during the system runtime. Witnessing this potential performance bottleneck, we then present PACMAN, a parallel transaction-level logging and recovery mechanism that leverages program analysis to enable speedy failure recovery without introducing any costly overhead to the transaction processing. Observing that most existing DBMSs adopt multi-version concurrency control (MVCC) protocols for increased degrees of concurrency, we further present a comprehensive performance study of DBMS's transaction management schemes to understand the major bottlenecks for processing varies types of workloads. We implemented these works on two multi-core main-memory DBMSs, namely Cavalia and Peloton, and the experiment results show that our mechanisms enable modern DBMSs to scale towards dozens of cores when processing various types of transactional workloads.