Semester Project

Multi version concurrency control study on Trireme

Problem: Main-memory OLTP engines are being increasingly deployed on multicore servers that provide abundant thread-level parallelism. However, recent research has shown that even the state-of-the-art OLTP engines are unable to exploit available parallelism for high contention workloads. While previous studies have shown the lack of scalability of all popular concurrency control protocols, they consider only one system architecture—a non-partitioned, shared everything one where transactions can be scheduled to run on any core and can access any data or metadata stored in shared memory.

In order to perform a thorough analysis of the impact of other architectural alternatives (Data-oriented transaction execution, Partitioned Serial Execution, and Delegation) on scalability under high contention scenarios, we are building Trireme, a main-memory OLTP engine testbed that implements four system architectures and several popular concurrency control protocols in a single code base [1].

Project: This project will extend Trireme by implementing a multiversion concurrency control protocol with the goal of identifying trade offs involved in implementing multiversioning in various OLTP engine architectures.

Plan:
1. Background study on multiversioning and preliminary analysis (2 weeks)
2. Introduction to Trireme + implementing baseline multiversioned storage (4 weeks)
3. Implement garbage collection (2 weeks)
4. Implement CC protocol (2 weeks)
5. Experimental analysis (1 week)
6. Write a report (6 pages using paper format) and present the work (1 week)

Supervisor: Prof. Anastasia Ailamaki, anastasia.ailamaki@epfl.ch
Responsible person: Dr. Raja Appuswamy, raja.appuswamy@epfl.ch
Dr. Angelos Anadiotis, angelos.anadiotis@epfl.ch

Duration: 3 months

References