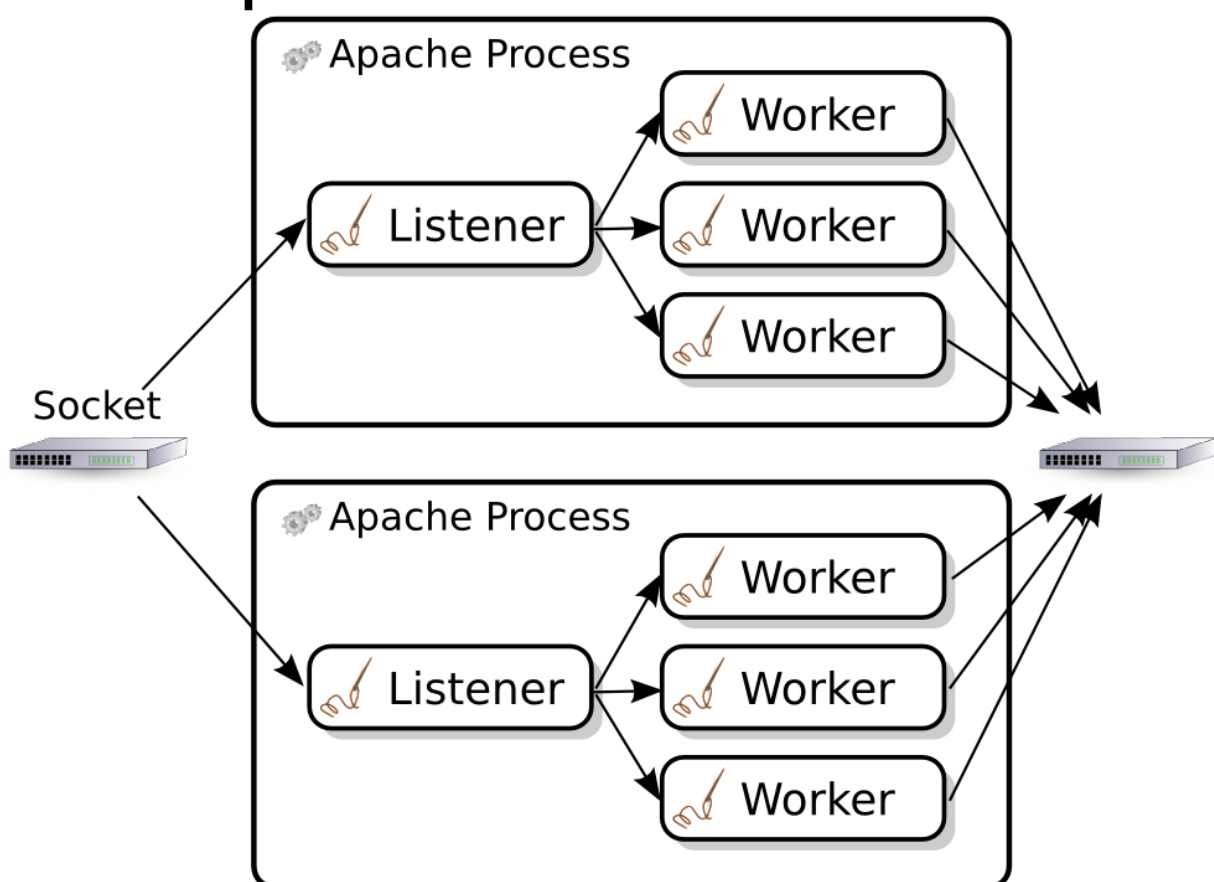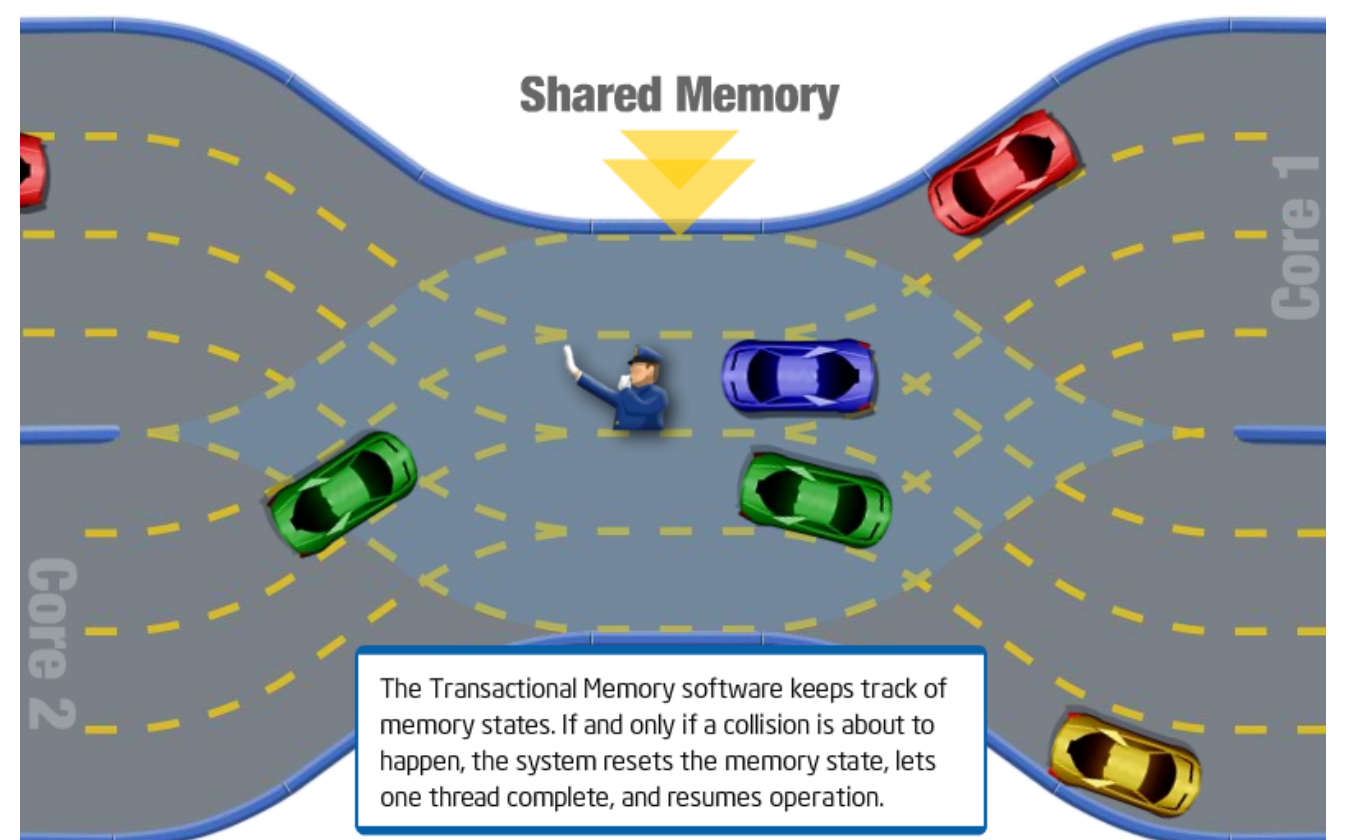# Transactified Apache

Haggai Eran, Ohad Lutzky
Advisors: Zvika Guz, Idit Keidar

Abstract Transactional Memory systems attempt to give multiprocessor programmers the ease of use of course-grained locks with the performance scalability of fine-grained locks. With the development of these systems a need for performance evaluation methods has emerged. This project offers a benchmark for transactional memory systems based on the popular Apache webserver. Results of running it with Intel's transactional memory manager have shown comparable performance to locking.

Apache Worker MPM

Transactional Memory

The Transactional Memory software keeps track of memory states. If and only if a collision is about to happen, the system resets the memory state, lets one thread complete, and resumes operation.

Transactional memory has emerged recently as an alternative synchronization method for parallel processing. A transaction is defined as an series of read and write operations and computations that should be executed atomically. The transactional memory system executes each transaction speculatively, without waiting for locks, and detects conflicts where two transactions might change global data in a way that distrubs atomicity, and abort one of the transactions, discarding all of its changes. If no such collision occurs, the transaction can commit, and its changes become visible to all other transactions.

Low Locality (s=1)   Medium Locality (s=2)   High Locality (s=3)