Problem: How do we make OLTP workloads go 10x faster?
Solution: Single thread execution; Partitioning; Avoid locks; Use speculation

Traditional Concurrency | Our Approach: H-Store | % CPU Cycles (Shore)
--- | --- | ---
Idle Resources: | | 8.1%
• Wait for disk | • Main memory | B-Tree Keys
• Wait for user | • Stored procedures | Logging
Physical Concurrency: | | 21.0%
• Multiple CPUs, disks | • Multiple partitions | Locking
| | 18.7% | Latching
| | 10.2% | Buffer Pool
| | 29.6% | Application
| | 12.3% |

Single Partition Transactions
No locks, no undo logging: no overhead

Multi-Partition Transactions
Two-phase commit; network stall (bad)

Low Overhead Concurrency Control: Do useful work during network stall

Speculation: Speculate next transactions during stall, after txn is prepared
• Best for simple multi-partition transactions: one round of work on partitions
Locking: Don't acquire locks if only executing single partition transactions
• Best for workloads with complex transactions; inter-partition communication

Experimental Results
Microbenchmark: Two partitions; Change fraction of multi-partition transactions
TPC-C like: Two partitions varying the number of warehouses