Motivation

- Programmability
- Single global lock is easy but too conservative
    - little concurrency
- multiple locks increases concurrency but is a lot harder to get it to work
    - deadlocks and such
- size of critical sections also reduce concurrency
- TM, through the means of optimistic concurrency allows the simplicity of single global lock without losing concurrency

TM Implementation

- Main components: conflict detection and data versioning (buffering)
- Conflict detection: lazy, eager
- Versioning: lazy, eager
- Transactions all the time or only in parts of the program
- TM in HW and in SW:
  - differences and trade-offs
- Nesting issues:
  - open x closed nesting

Other Issues

- System
  - OS syscalls
  - I/O
- Overflow
- Other Uses
  - TLS = TM + ordering
  - aggressive optimizations like multiple-path execution