Simplifying End-to-End Verification of High-Performance Distributed Systems

Carnegie Mellon University

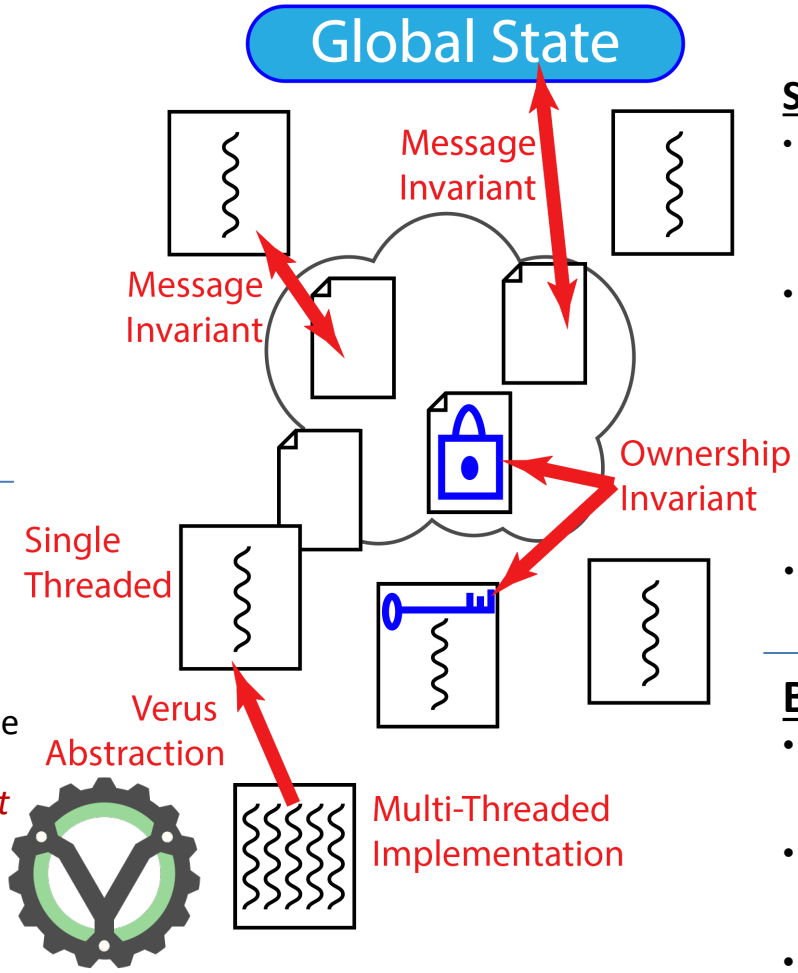
University of Michigan

Challenge:

- Distributed systems are difficult to reason about
  1. Unreliable networks
  2. Reasoning about distributed state
  3. Complexities of real-world implementations

Solution:

1. Use *message invariants* to abstract away the network
2. Use *ownership invariants* to reason about distributed state
3. Use **Verus** to *verify high-performance, concurrent Rust* implementations



Scientific Impact:

- Identified an *invariant taxonomy* that creates automation opportunities
  - **Kondo** tool and methodology simplifies verification of diverse protocols [*OSDI'24*]
- Developed **Verus**, an automated tool for *verifying (concurrent) systems software*
  - Handles safe and unsafe Rust
  - Reduces developer burden
  - Produces verification results orders of magnitude faster [*SOSP'24*]
  - Automated debugging for failed proofs [*CAV'24*, Distinguished Paper]
- Two of three *Best Paper awards* at OSDI'24 used Verus

Broader Impact:

- Simpler protocol verification means more reliable digital infrastructure
- Verus makes systems verification available to a broad community of developers
- **WIP**: Verified distributed system textbook
- **WIP**: Verus day-long tutorial

Bryan Parno (CMU), Manos Kapritsos (U. Michigan)
Project # 2318953