CFSCQ: Extending a verified file system with concurrency

Tej Chajed
advised by Frans Kaashoek and Nickolai Zeldovich

Goal: verify a concurrent file system

- Existing verified file systems are **sequential**
  - *e.g.*, FSCQ, Yggdrasil, BilbyFS
- All real file systems are **concurrent**
  - *e.g.*, ext4, btrfs

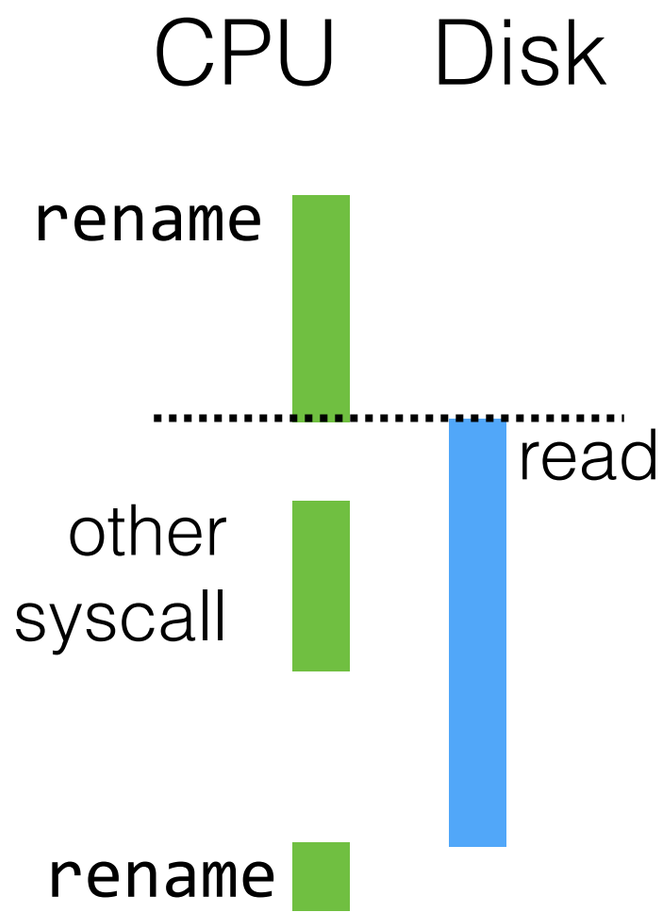
CFSCQ re-uses FSCQ (a verified sequential file system)

- FSCQ: 75,000 lines
- CFSCQ: +6,000 lines
- Concurrency verified separately from sequential behavior

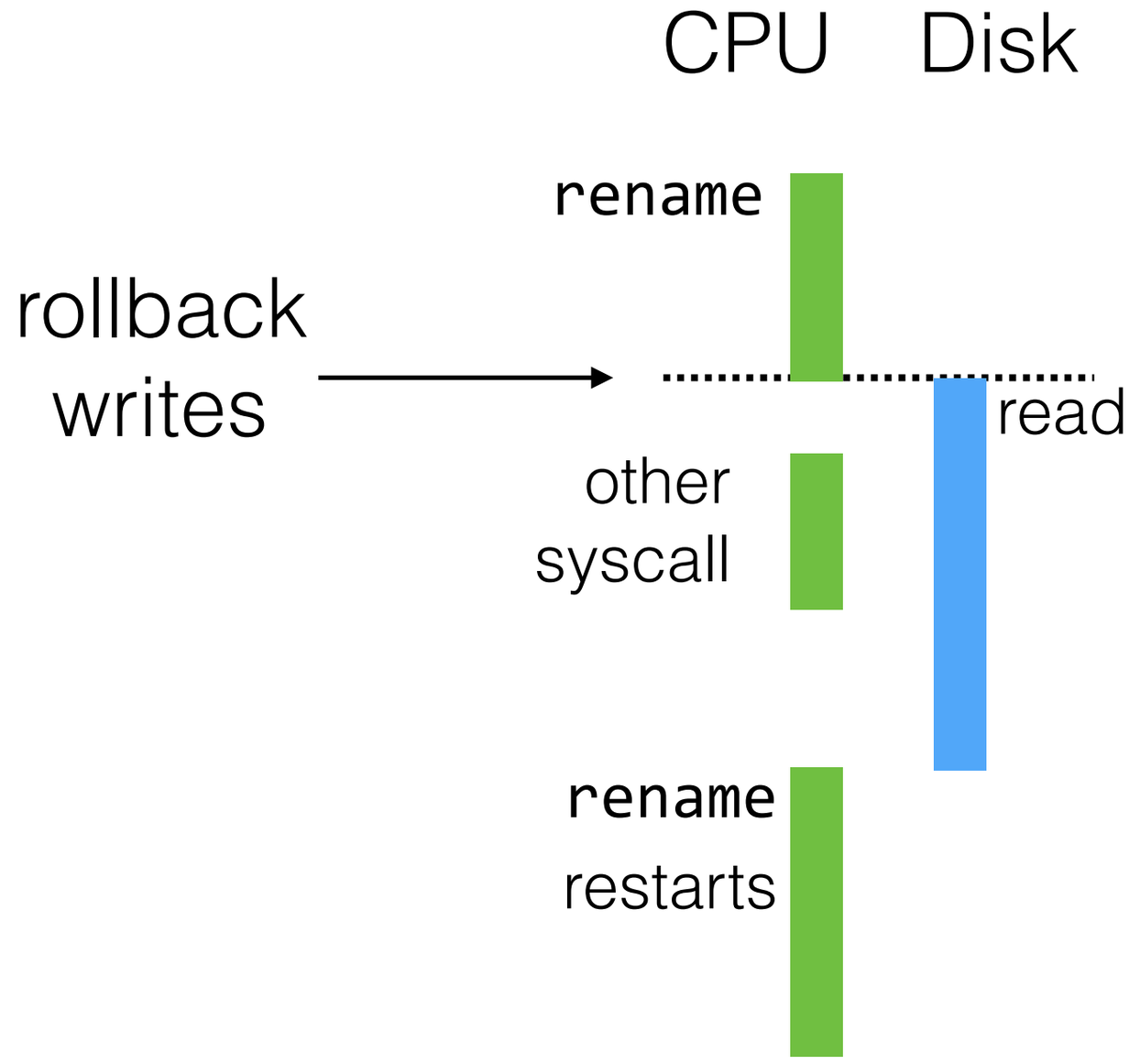
What can we achieve without modifying proofs?

- Make disk reads asynchronous
- Run read-only system calls on multiple cores
- Leverage FSCQ code, spec, and proof for bulk of concurrent implementation

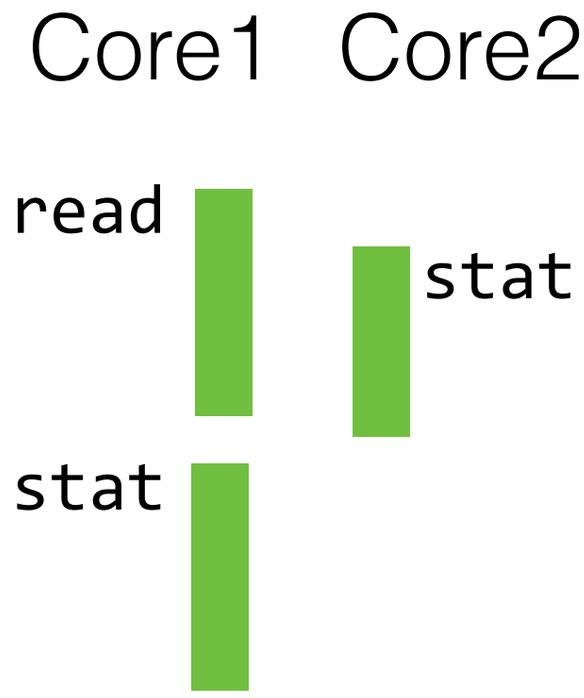
Asynchronous reads allow system calls to read from memory while disk is reading



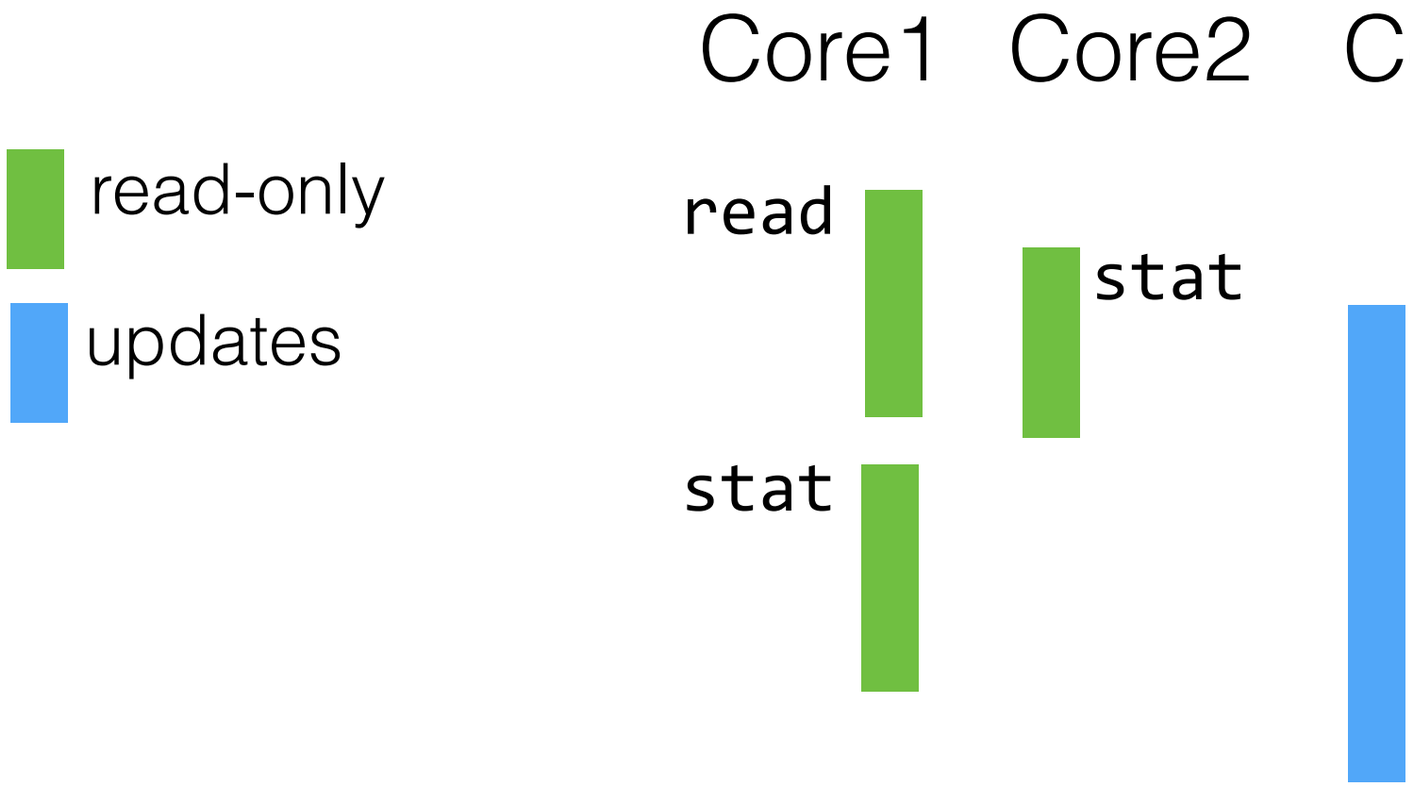
Asynchronous reads allow system calls to read from memory while disk is reading



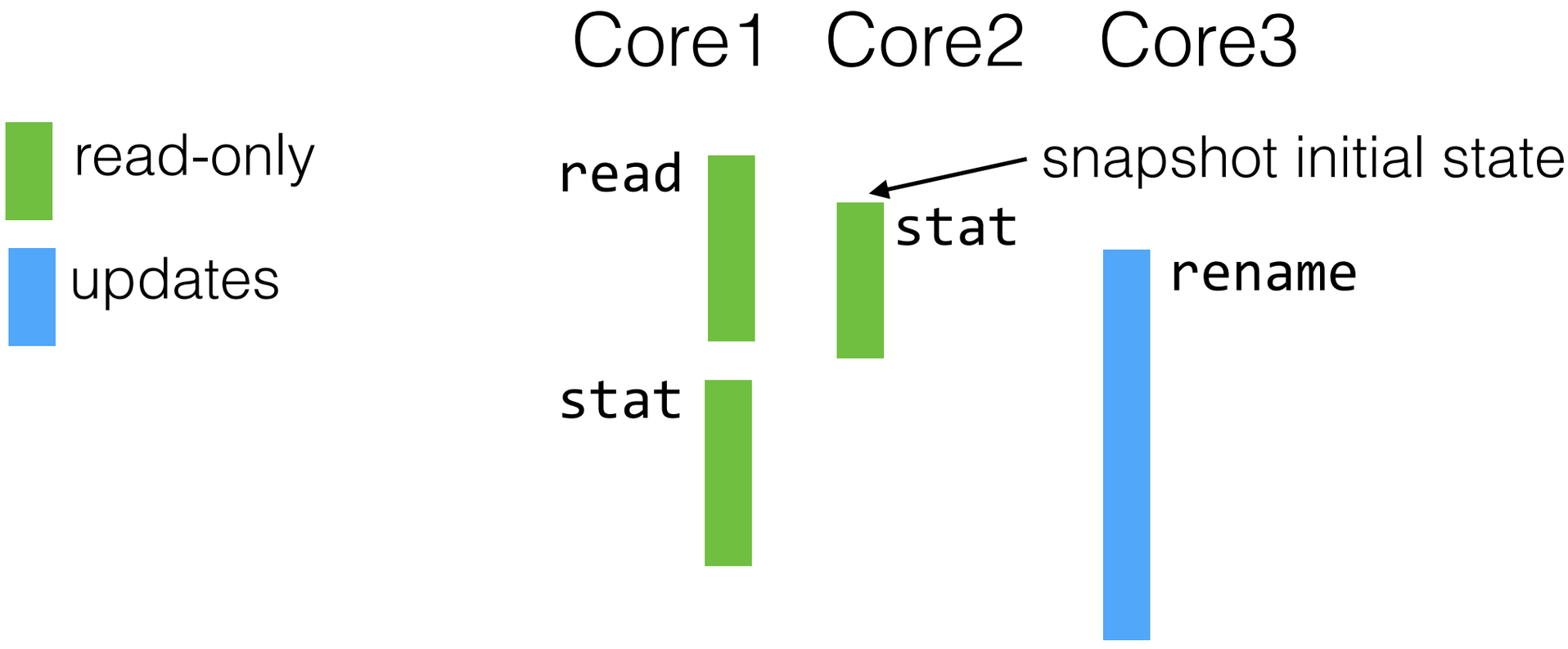
Read-only system calls run on separate cores



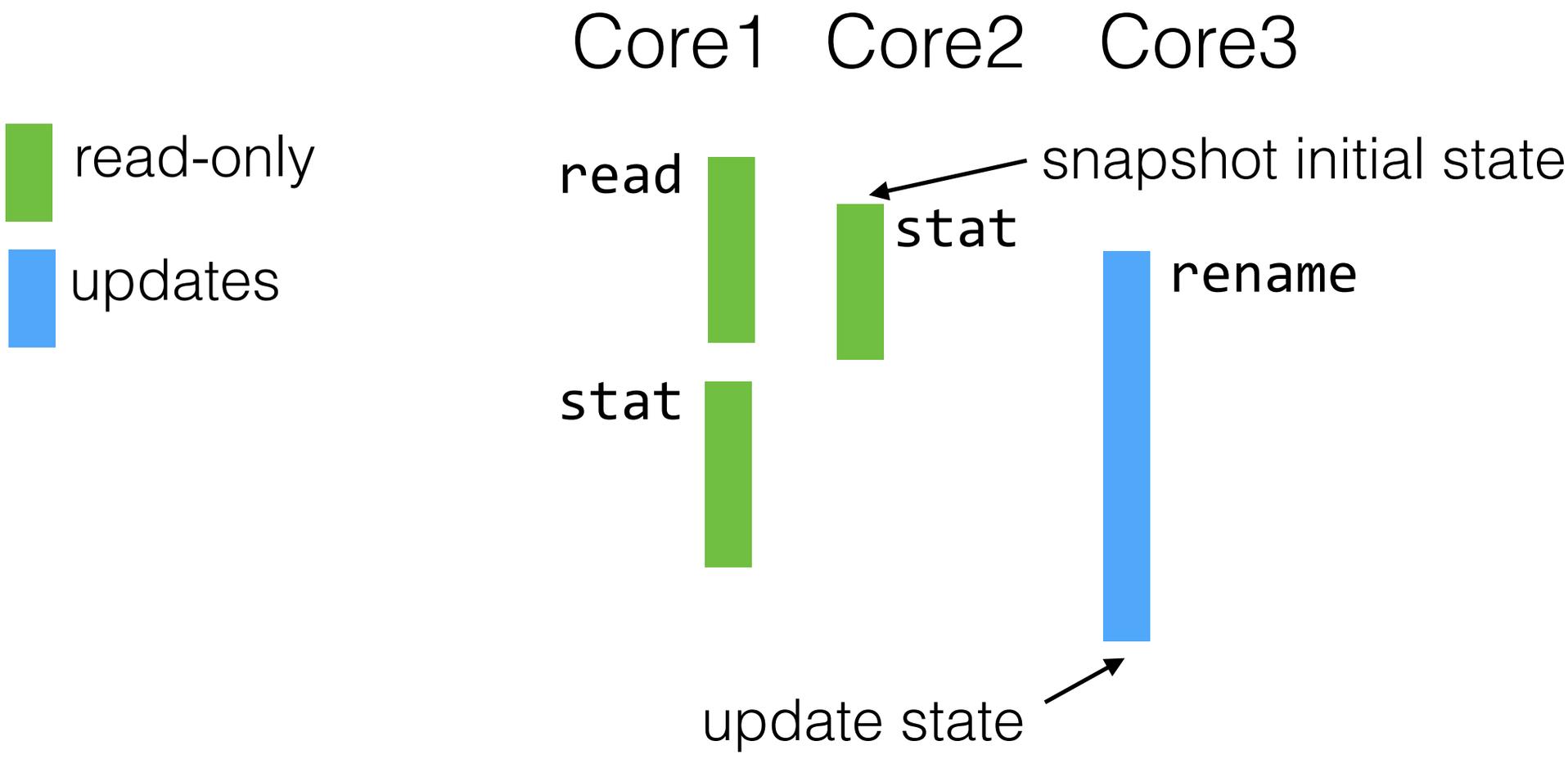
Read-only system calls run on separate cores



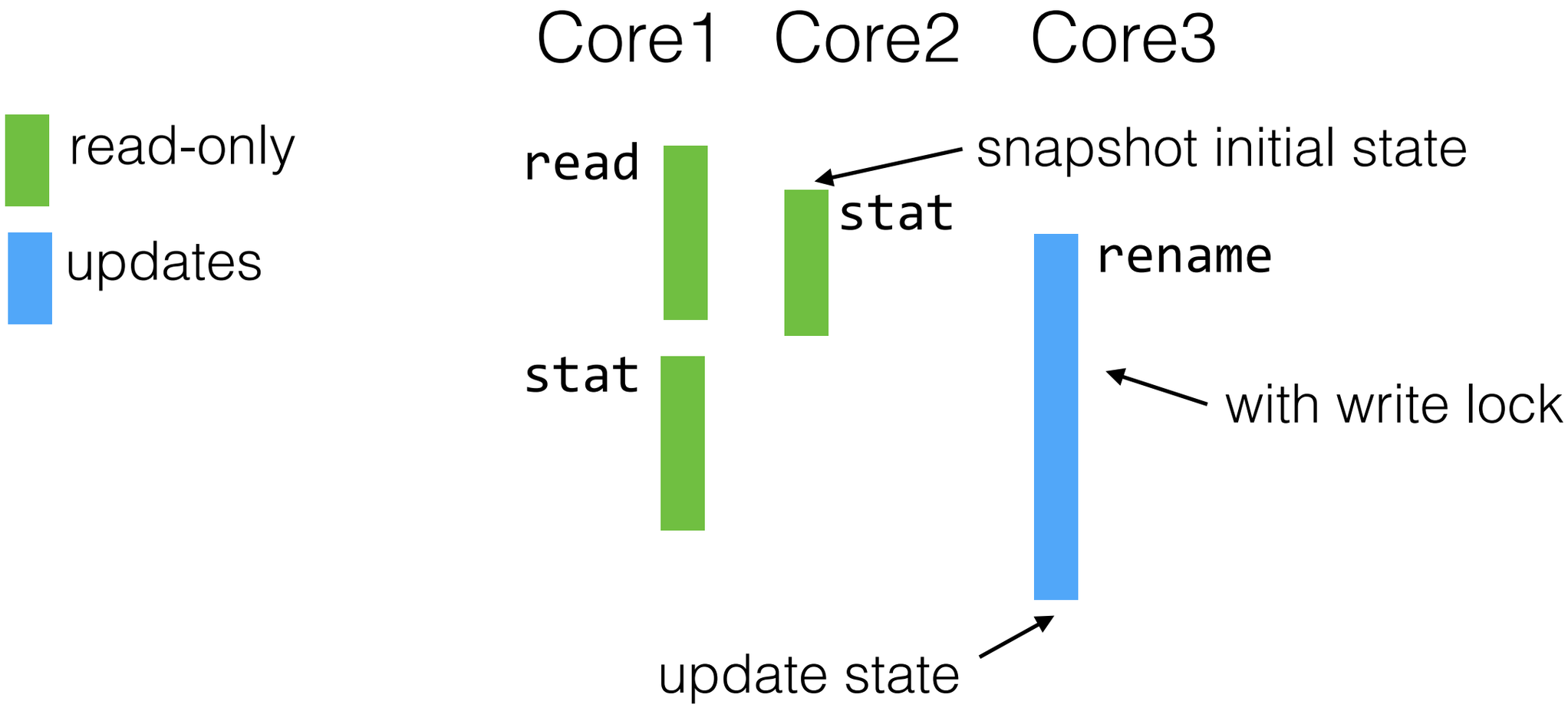
Read-only system calls run on separate cores



Read-only system calls run on separate cores



Read-only system calls run on separate cores



Progress

- ✔ Verified asynchronous disk reads and multicore concurrent reads
- ✔ Asynchronous disk reads improve throughput with slow I/O
- Working on performance and scalability of multicore reads