

Reliable Storage Using Storage Class Memory (SCM)

Prashant Saxena, Michael Swift
prashant@cs.wisc.edu, swift@cs.wisc.edu



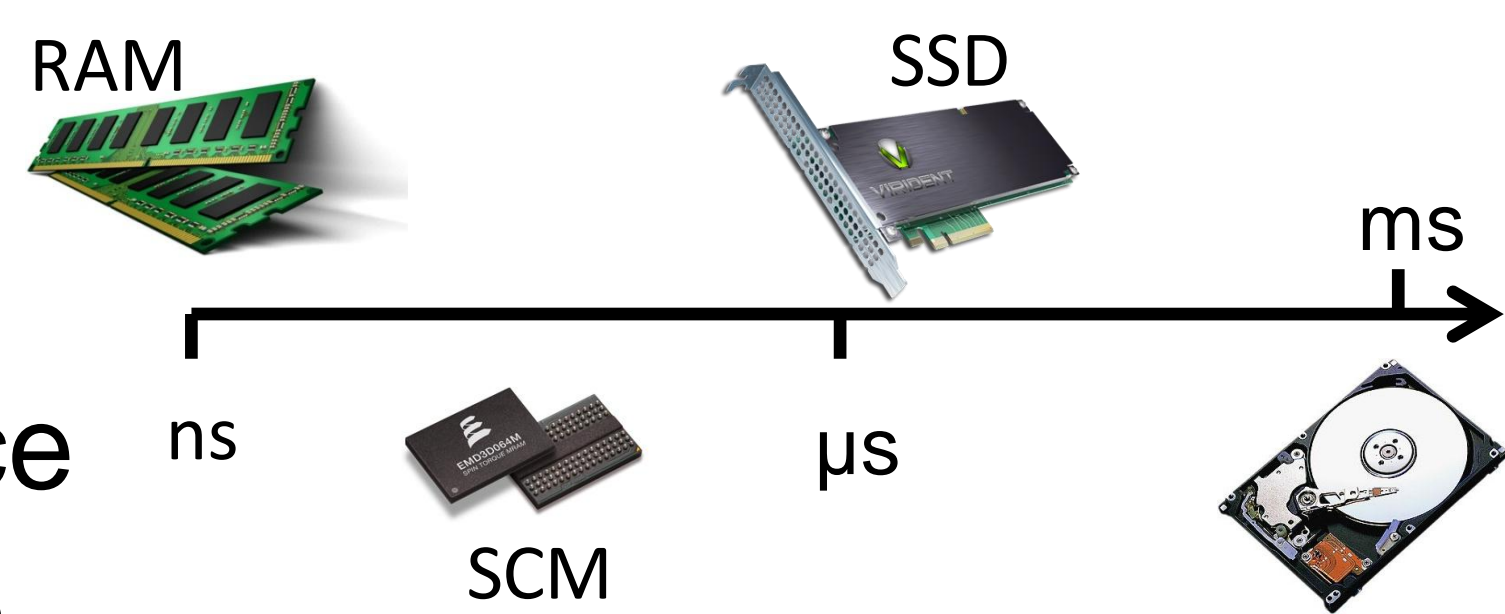
Storage Class Memory

Key Features

- Persistent
- Load/Store interface
- Short access times

I/O Interfaces

- POSIX style File Systems
- Key Value File Systems
- Library File Systems



Motivation

Pros

- High performance with low latency

Cons

- Limited endurance
- Single-node attachment as memory
- Standard reliability models like RAID reduce performance and limit programming models

Need to rethink reliability in SCM devices

Why reliability model for SCM is challenging?

RAID on SCM not effective

- Number of devices limited by on-board DIMM slots
- Doubling SCM capacity is expensive
- Replicating data to another SCM is expensive and wasteful
- Hard to compose with direct access as persistent memory

Using SSD as backing store

- Different interface: byte-addressable memory vs block-addressable device
- Latency of replication hurts performance (SSD at least twice as slow as SCM for write latencies)

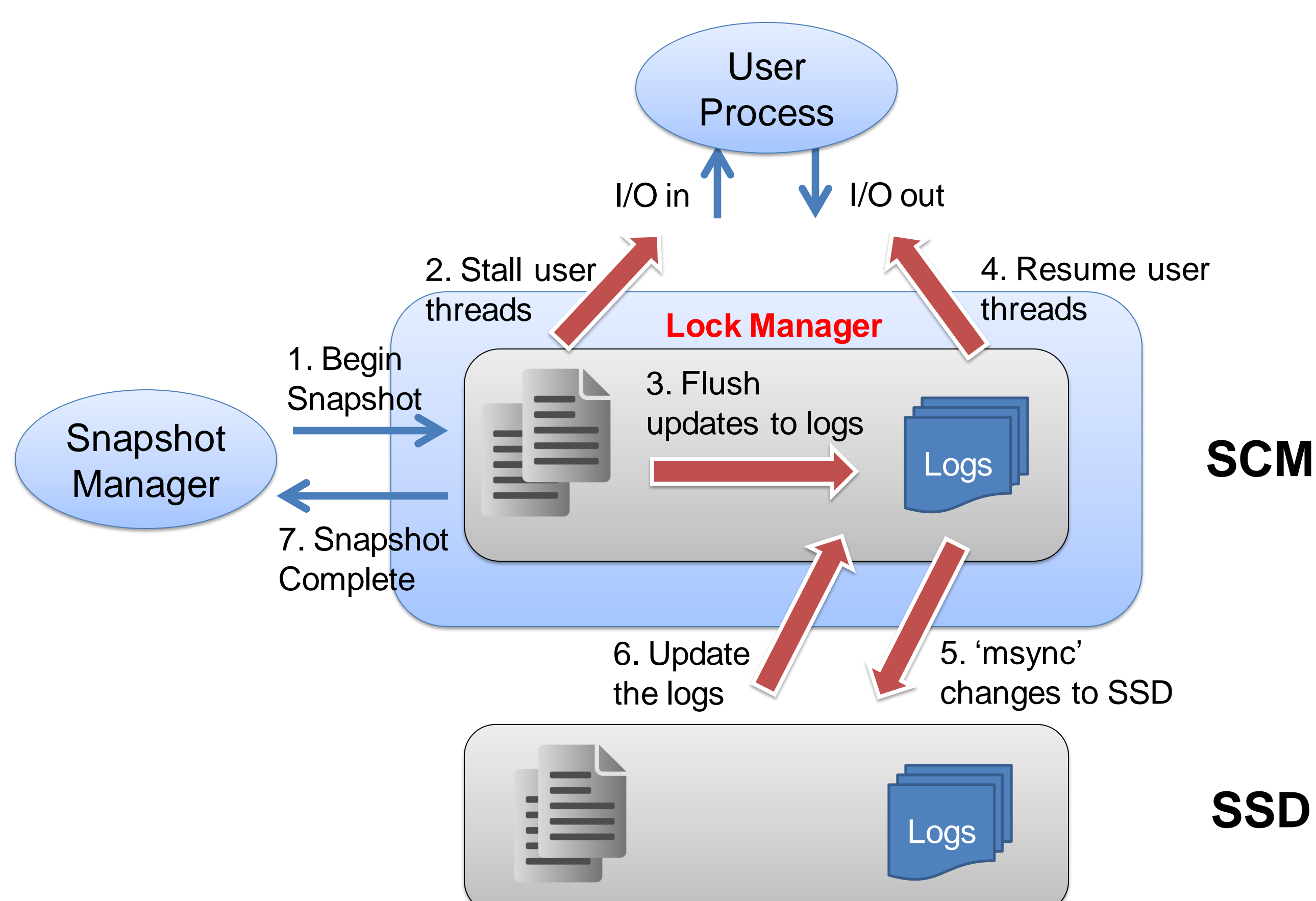
Our Approach

- Replicate data to a lower latency SSD
- Guarantee consistency but relax freshness

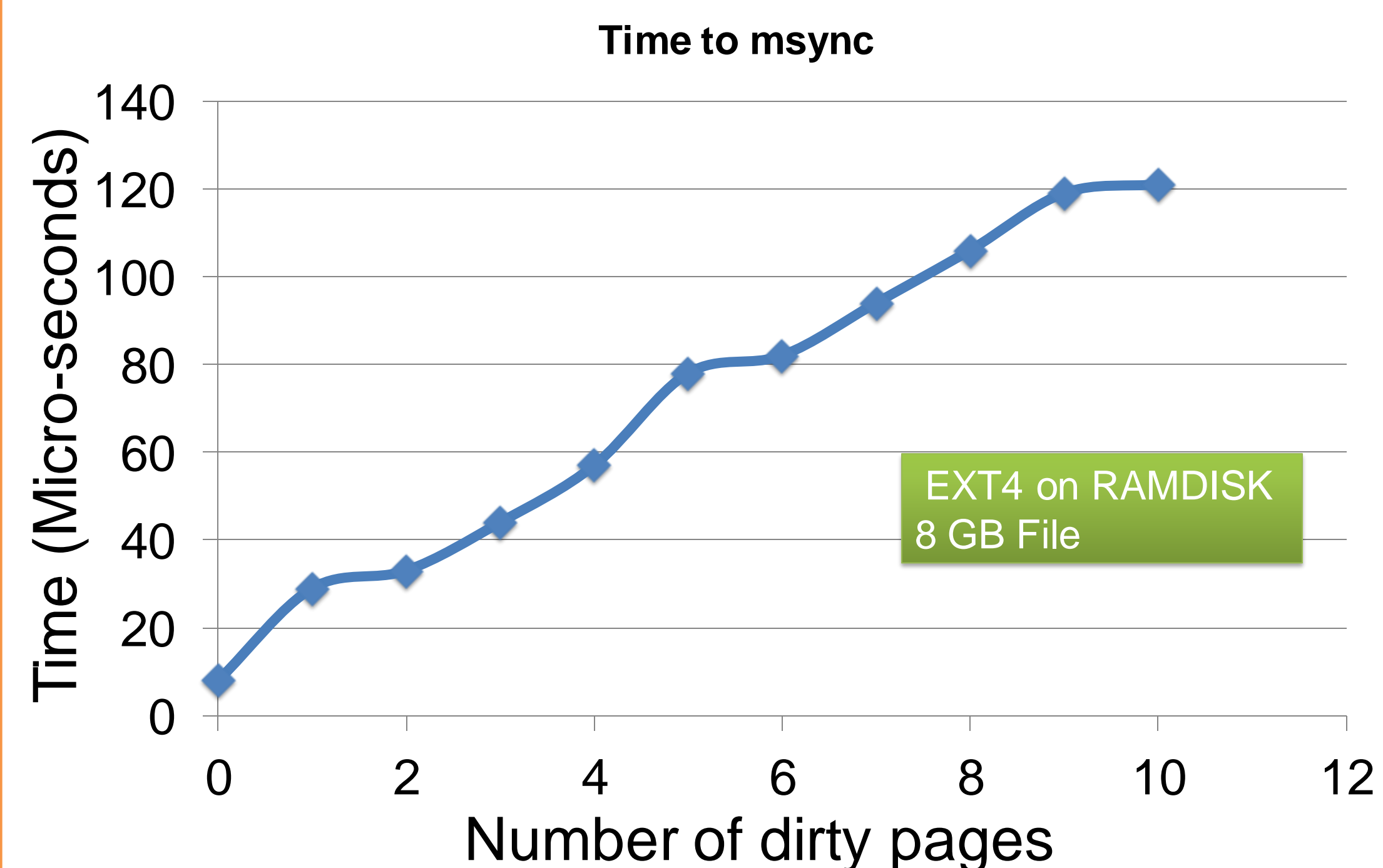
How

- Treat SCM contents as memory-mapped files
- Use 'msync'-like techniques to mirror data to a cheap SSD for recovery
- Use memory transactions to create and modify data on SCM
- Coordinate 'msync' with memory transactions to avoid long pauses

Our Solution



Why 'msync'?



Efficient at finding dirty pages in large files

Future Work

- Asynchronous 'msync' implementation
- Mnemosyne Integration
- Explore JDB2 layer of EXT4 for journaling
- Efficient locking protocol for minimum thread stalling