DMP

Deterministic Shared Memory Multiprocessing

University of Washington

Joe Devietti, Brandon Lucia, Luis Ceze, Mark Oskin

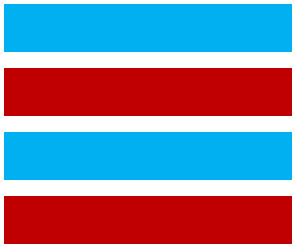
A multithreaded voting machine

thread 0

```
while (more_votes) {
  load t <- votes
  t++
  store t -> votes
}
```

thread 1

```
while (more_votes) {
  load t <- votes
  t++
  store t -> votes
}
```



votes == 2 votes == 2 votes == 1

A multithreaded voting machine

data race

we're not trying to make these bugs go away

we're trying to make them **come back!**

locking discipline violation

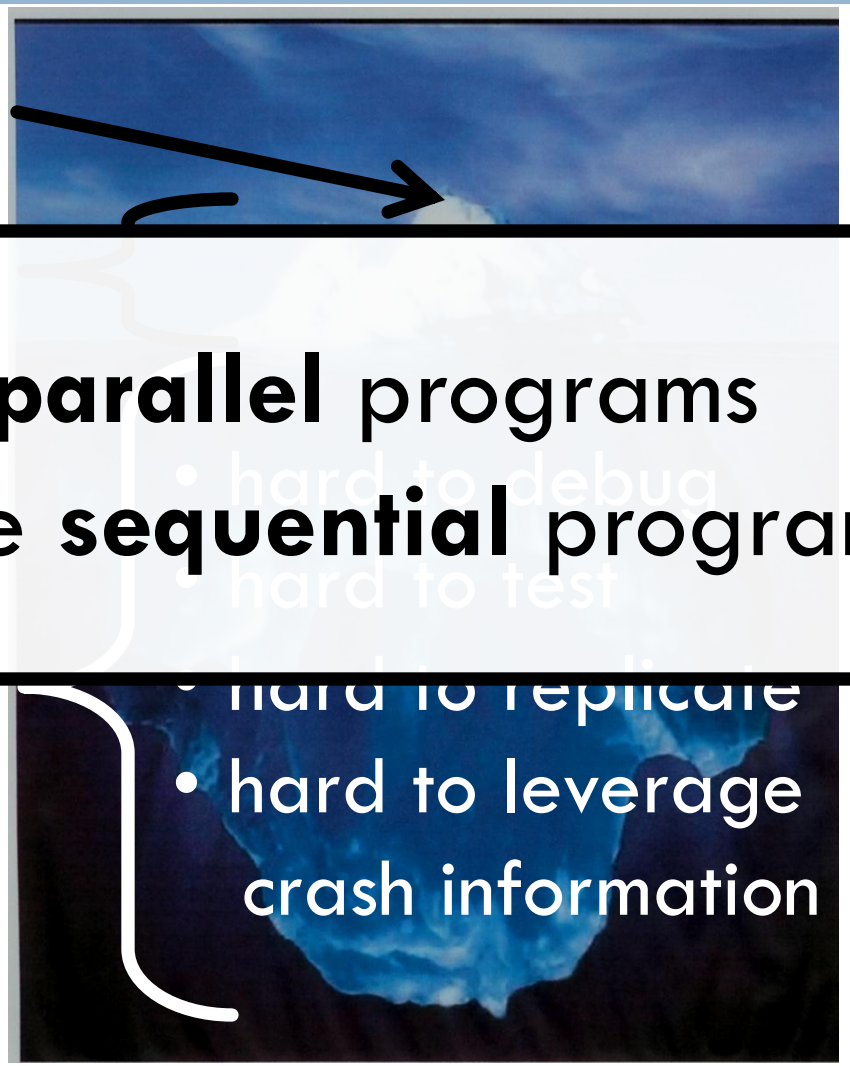
Why is parallel programming hard?

sequential bugs

concurrency bugs

nondeterministic

memory access interleavings



We want **parallel** programs
to behave like **sequential** program

Determinism Can Help

Development

- no more heisenbugs!
- time-travel debugging
- **test inputs, not interleavings**

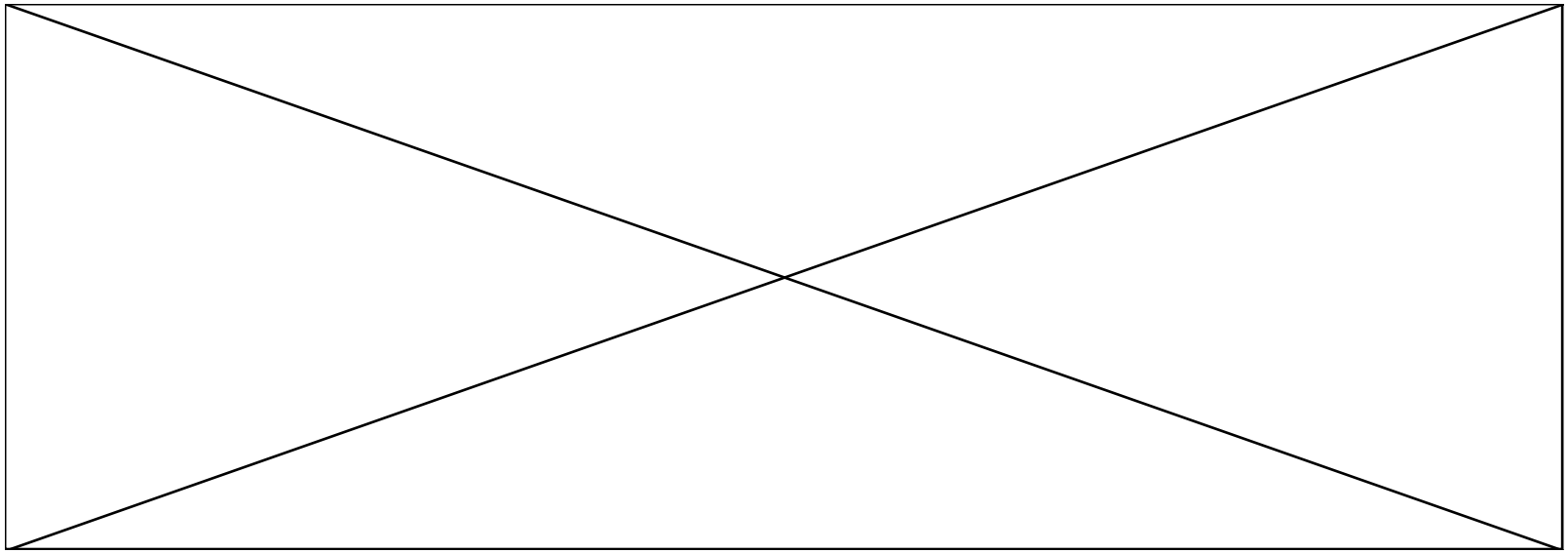
Deployment

- reproduce bugs from field
- easy to synchronize replicas
- **software behaves as tested**

Can we remove nondeterminism without removing performance?

DMP from 10,000'

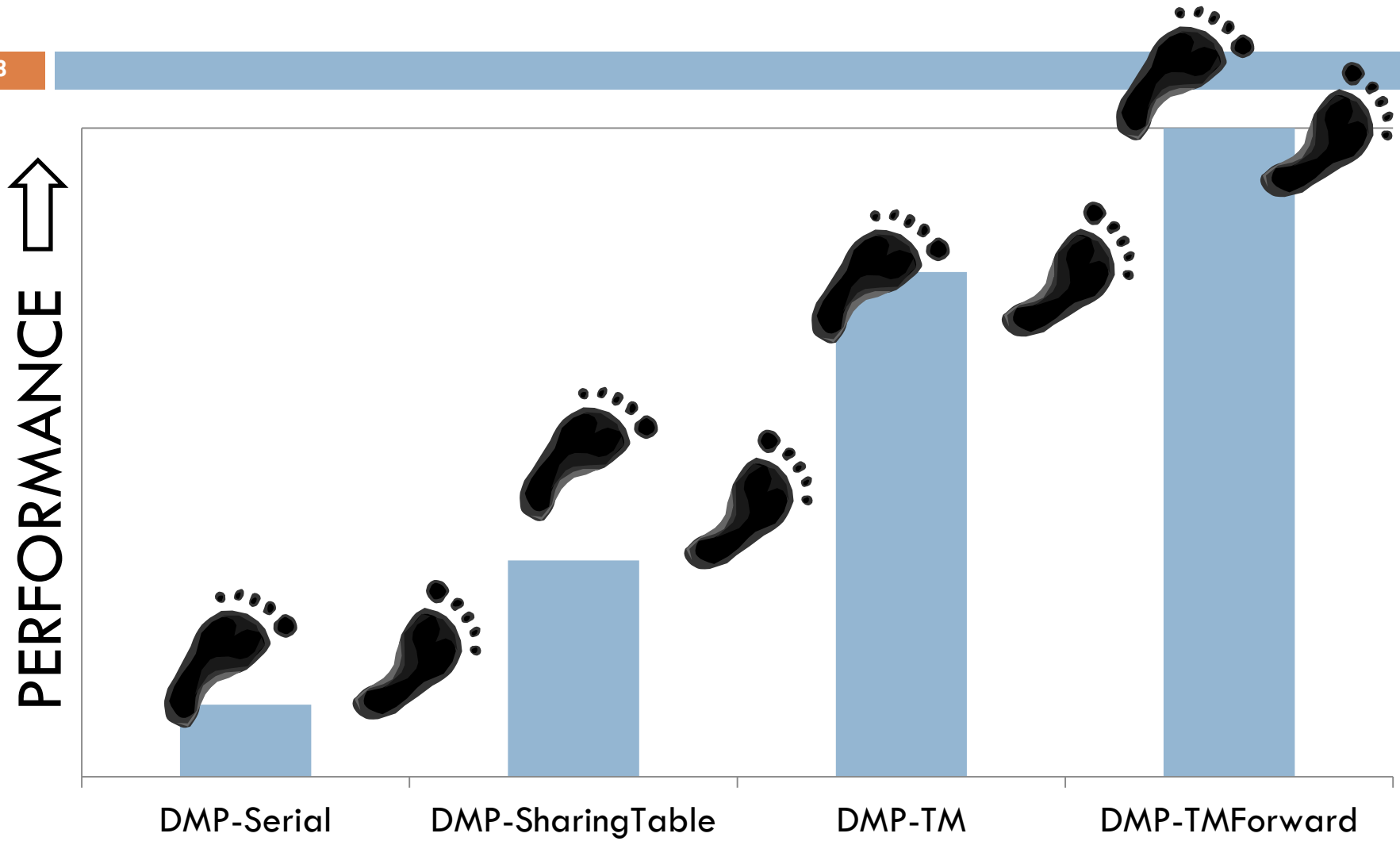
- We only care about communicating instructions
- Deterministic serialization → same communication
  - …but I promised you performance!
- Recover parallelism from non-communicating insns



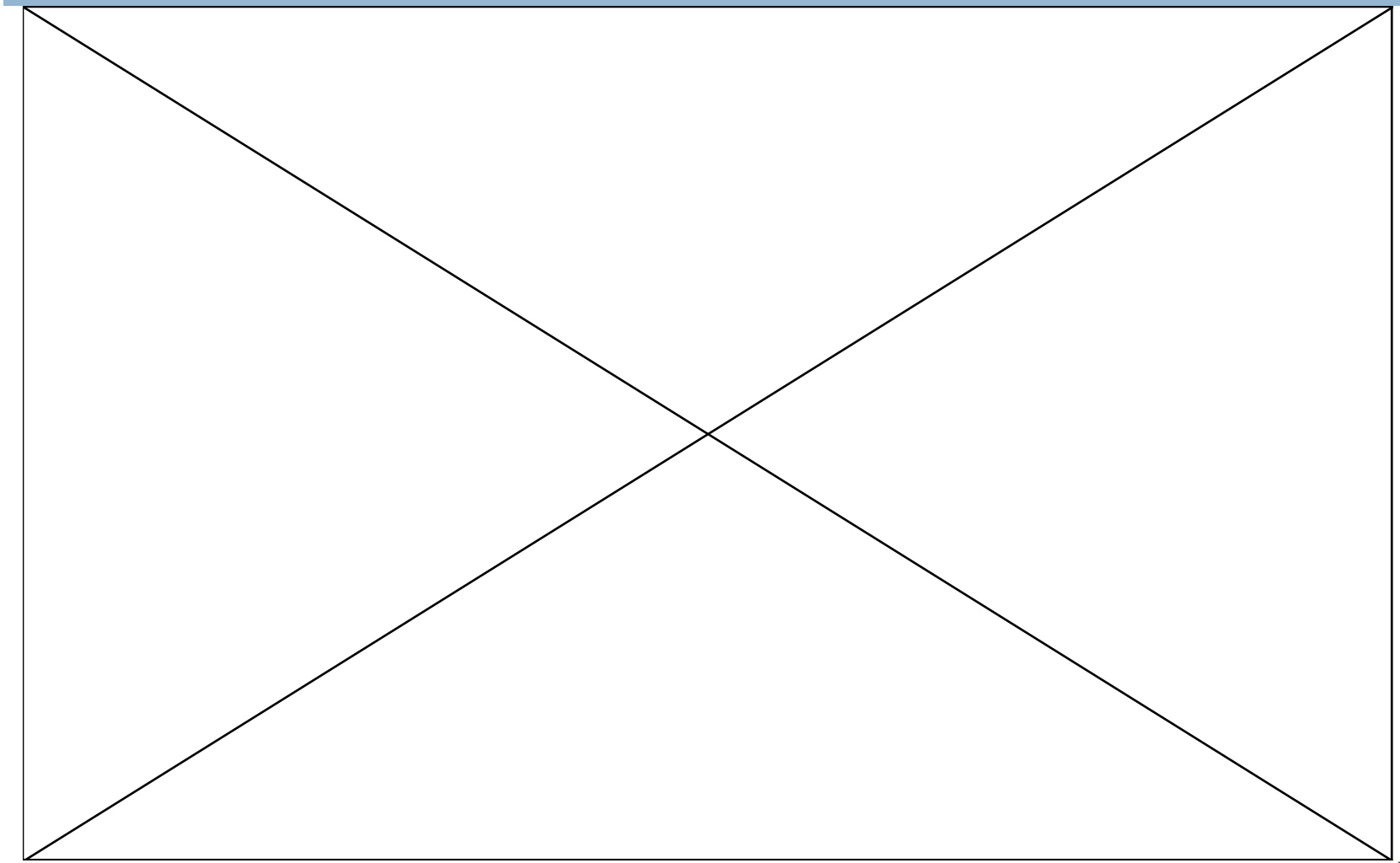
Related Work

Helps with…	Record + Replay
…debugging?	◒
…testing?	○
…replicas?	○
…deployment?	○
Needs hw?	usually
examples:	FDR, ReRun, Capo

Talk Outline

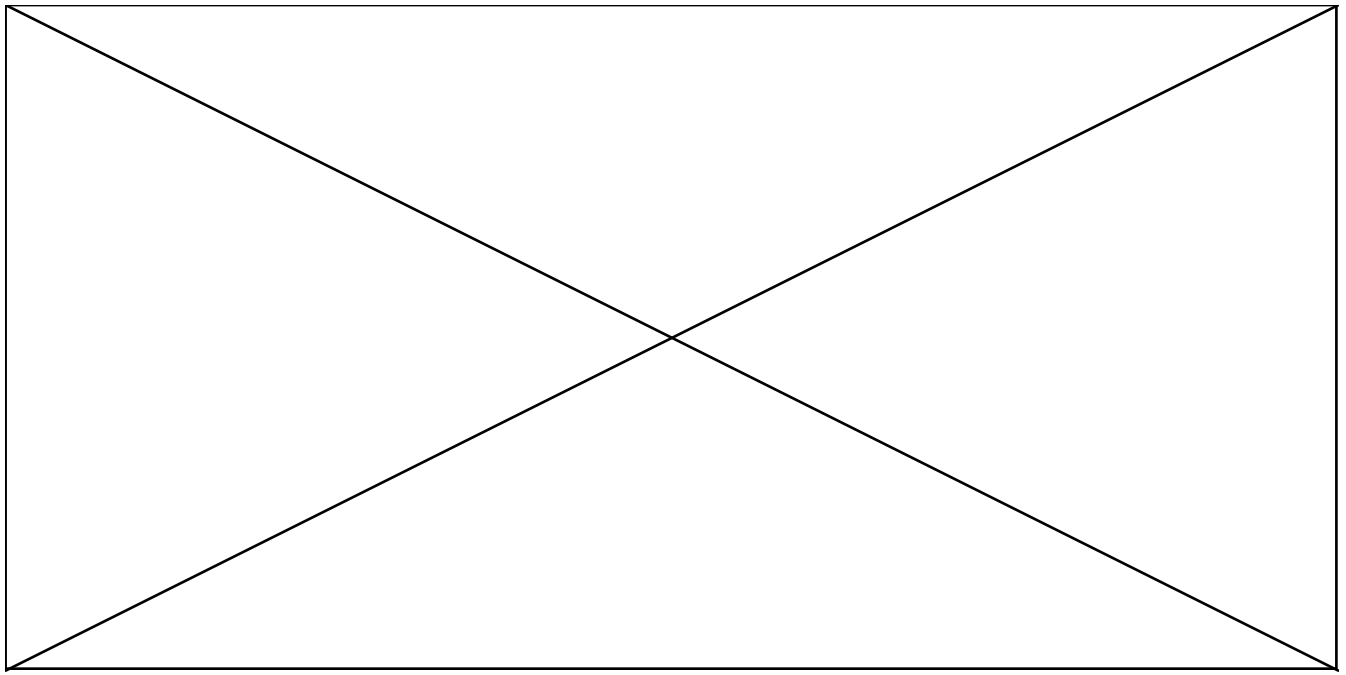


DMP-Serial Example



Can we do better?

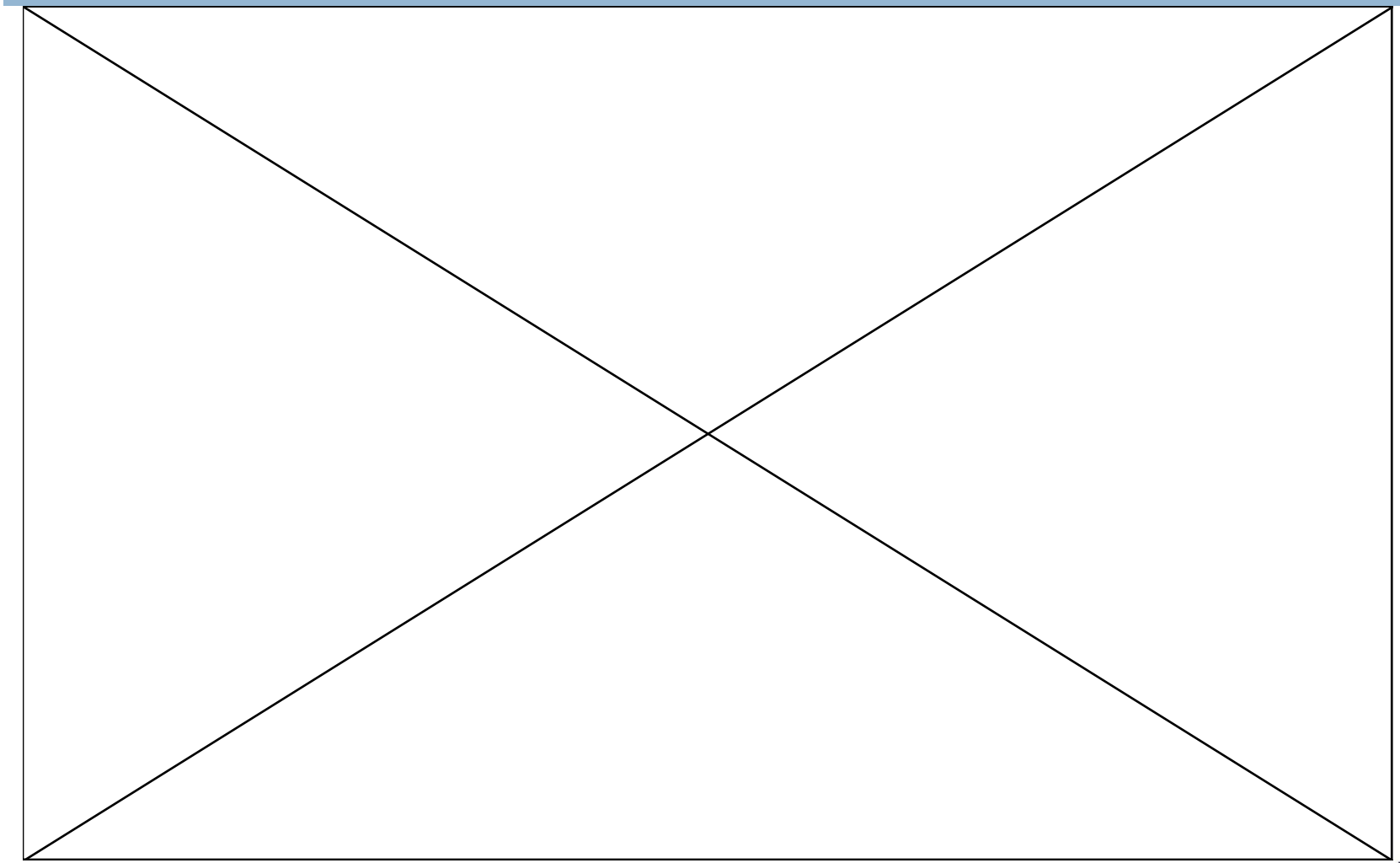
- Only need to serialize communicating instructions
- Break each quantum into communication-free **parallel prefix** and communicative **serial suffix**



DMP-SharingTable

- Need to know when communication happens, to transition from parallel to serial mode
  - Leverage existing cache coherence protocol
  - When a line moves between processors, communication is (potentially) happening!
  - The **Sharing Table** tracks information about ownership
- State of Sharing Table must evolve deterministically
  - Only allow updates during serial suffix

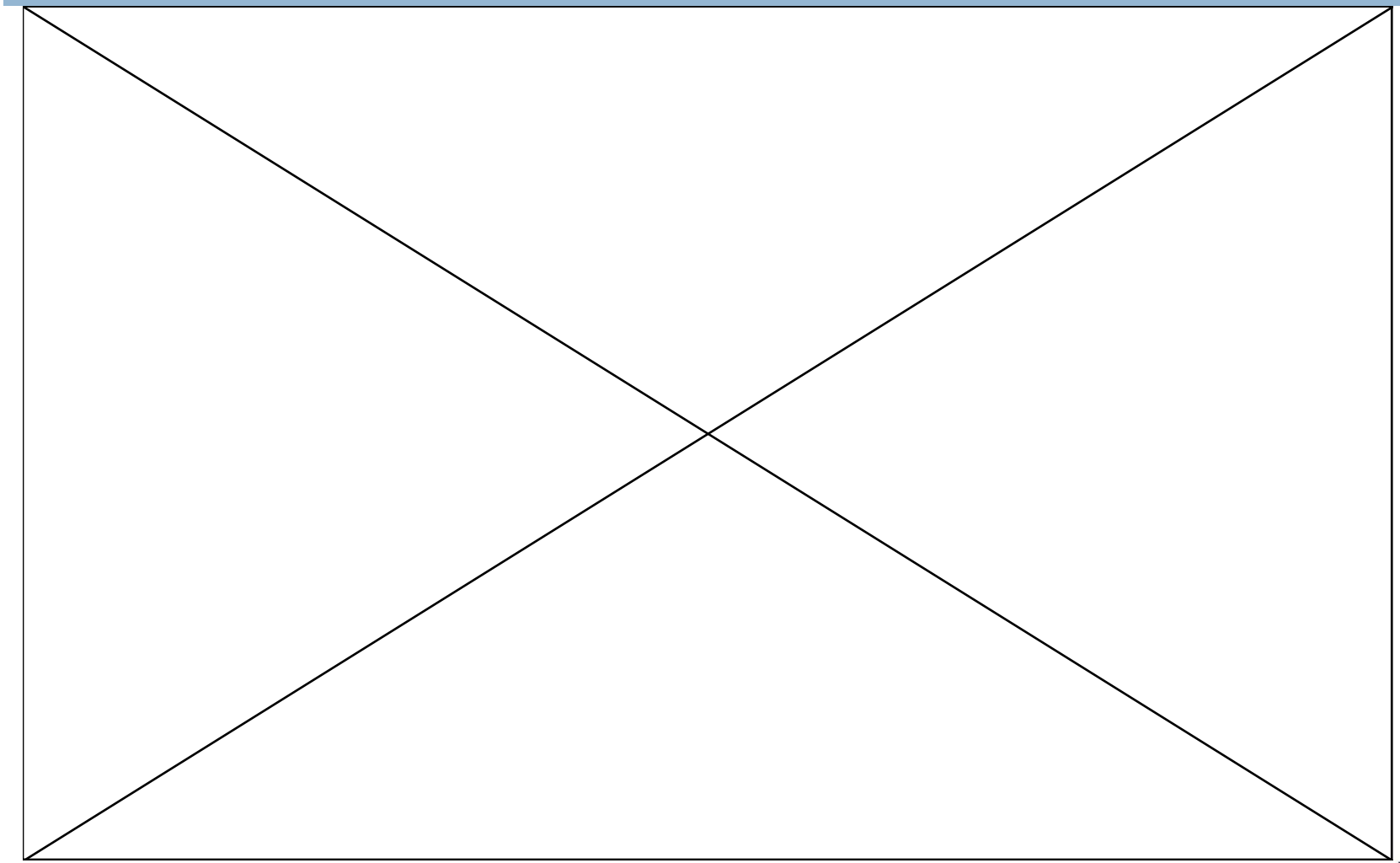
DMP-SharingTable Example



DMP-TM: Recovering Parallelism with Speculation

- DMP-SharingTable conservatively assumes that all changes in ownership are communication
  - …but most changes in ownership are **not** communication
- Use TM support to speculate that a quantum is not involved in communication
  - If communication happens, rollback + re-execute
- Each quantum is an implicit transaction
  - **Commit quanta in-order** (need DT to commit)

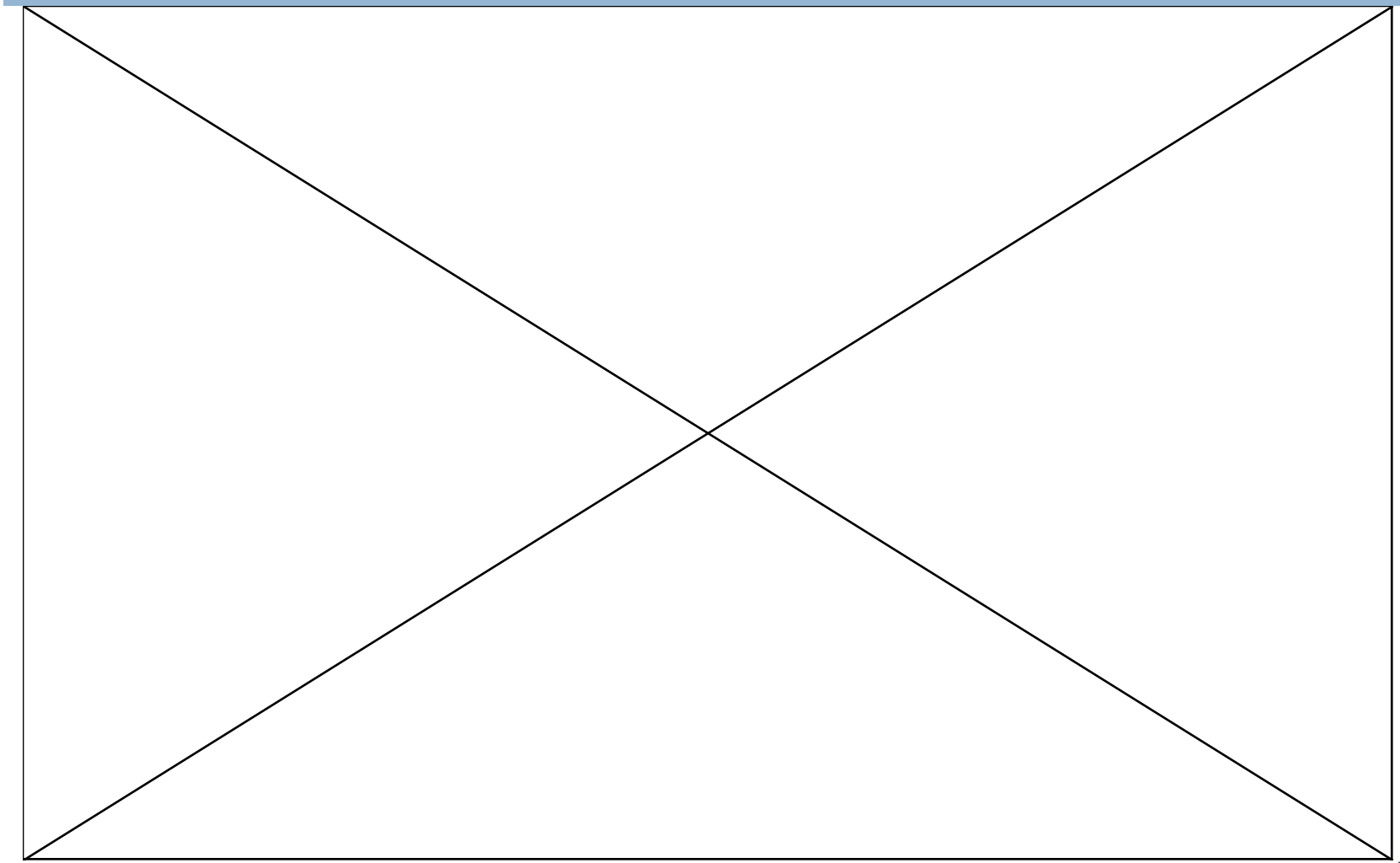
DMP-TM Example



DMP-TM-Forward: Speculative Value Forwarding

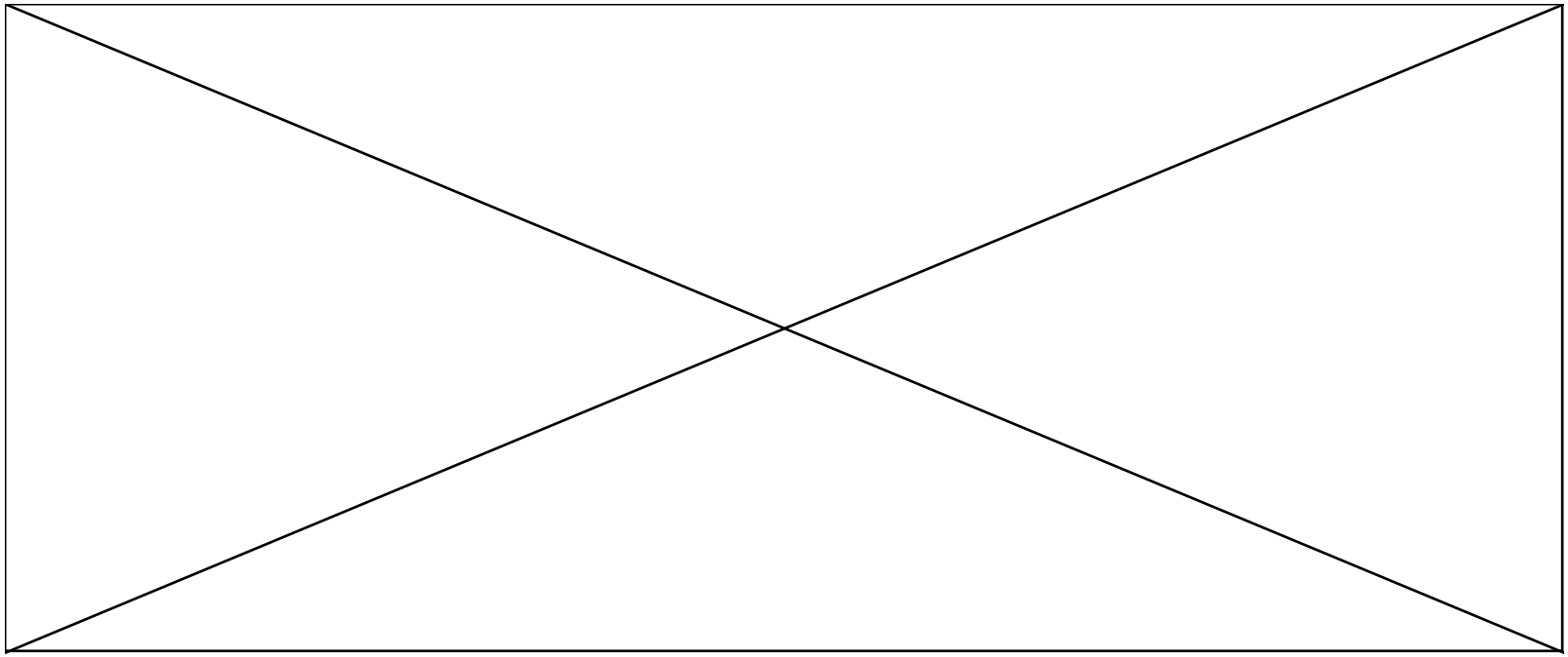
- DMP-TM eliminates WAW and WAR dependencies
  - but cannot speculate past true (RAW) dependences
- Idea: speculatively forward values to "future" quanta
  - coherence protocol + ordered transactions make it easy to decide *when* and *where* to forward
  - rollback if a quantum's speculatively read data is updated before the quantum commits

DMP-TM-Forward Example



Better Quantum Building

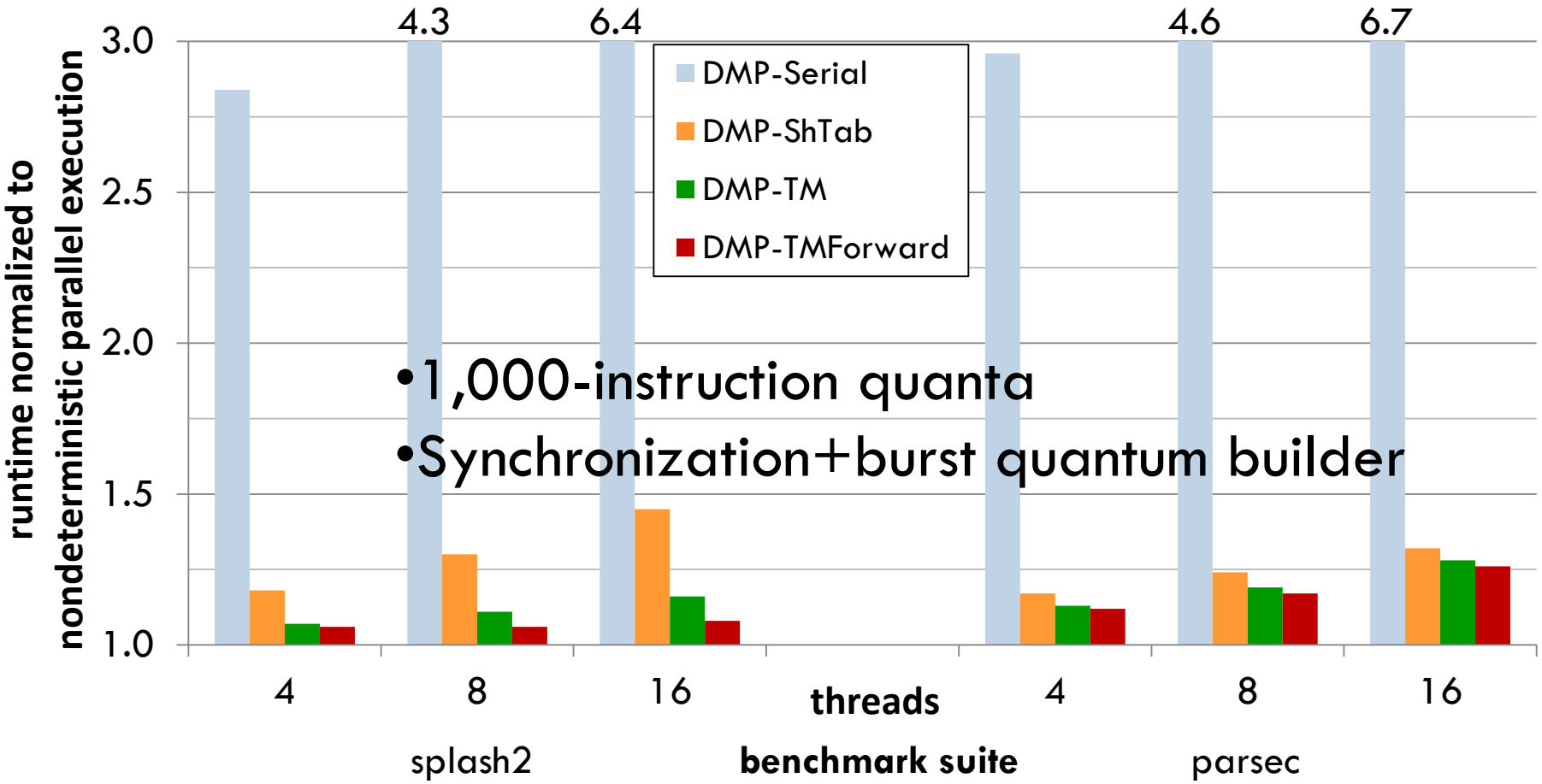
- Any deterministic policy will work
- We want quanta that are free of communication
  - no communication → no serialization, no rollbacks



Experimental Methodology

- PIN-based simulator
  - Models serialization, quantum building, address conflicts and transaction rollbacks
  - Assumes constant IPC with free commits
- SPLASH2 and PARSEC benchmark suites

Results


runtime normalized to nondeterministic parallel execution
3.0
2.5
2.0
1.5
1.0
4.3
6.4
4.6
6.7
DMP-Serial
DMP-ShTab
DMP-TM
DMP-TMForward
•1,000-instruction quanta
•Synchronization+burst quantum builder
4
8
16
threads
4
8
16
splash2
benchmark suite
parsec

Also in the paper…

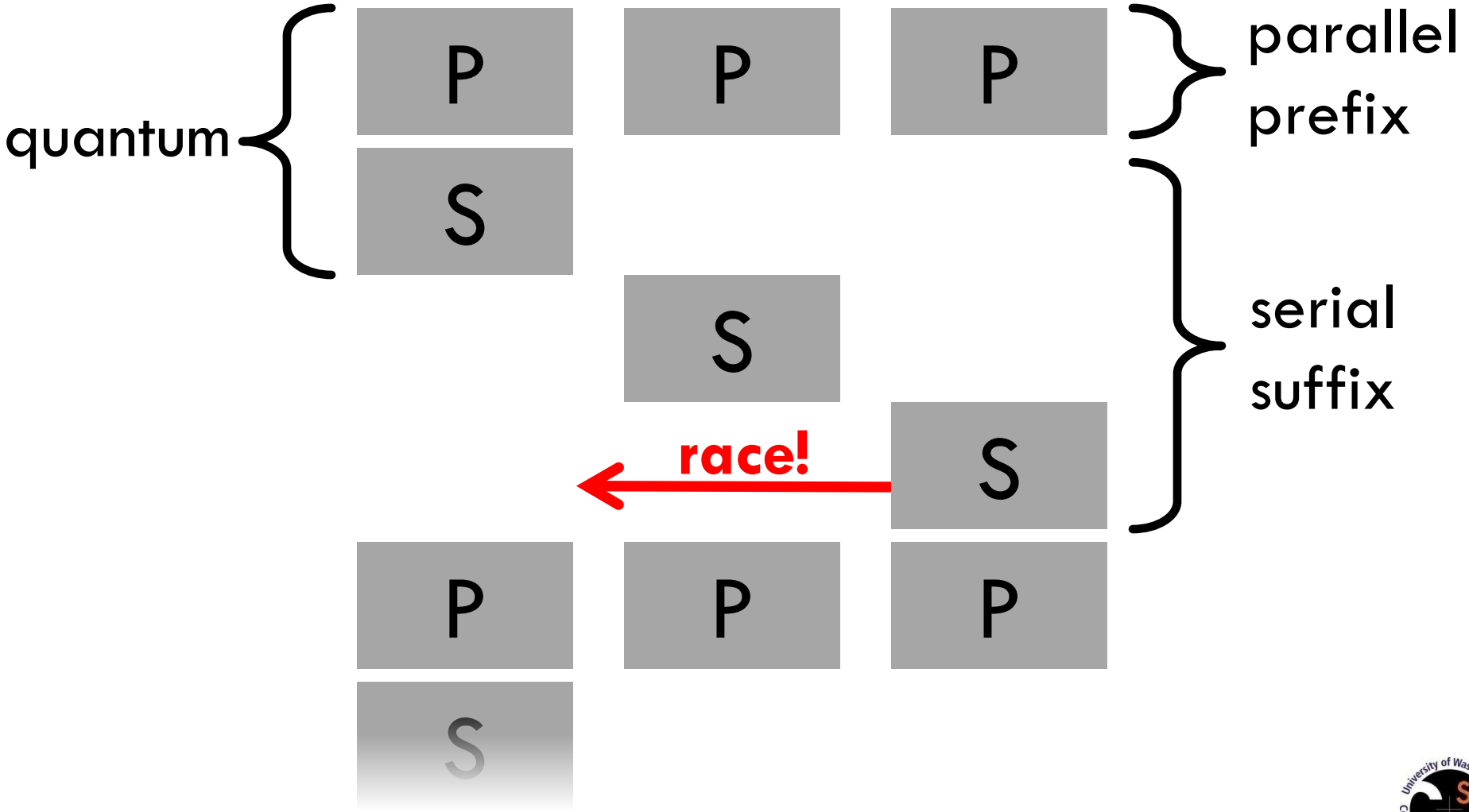
- Software-only Sharing Table implementation
- Support for debugging
  - Adding instrumentation without affecting communication
- Making execution deterministic across machines
- Dealing with nondeterminism from I/O and the OS

Conclusions

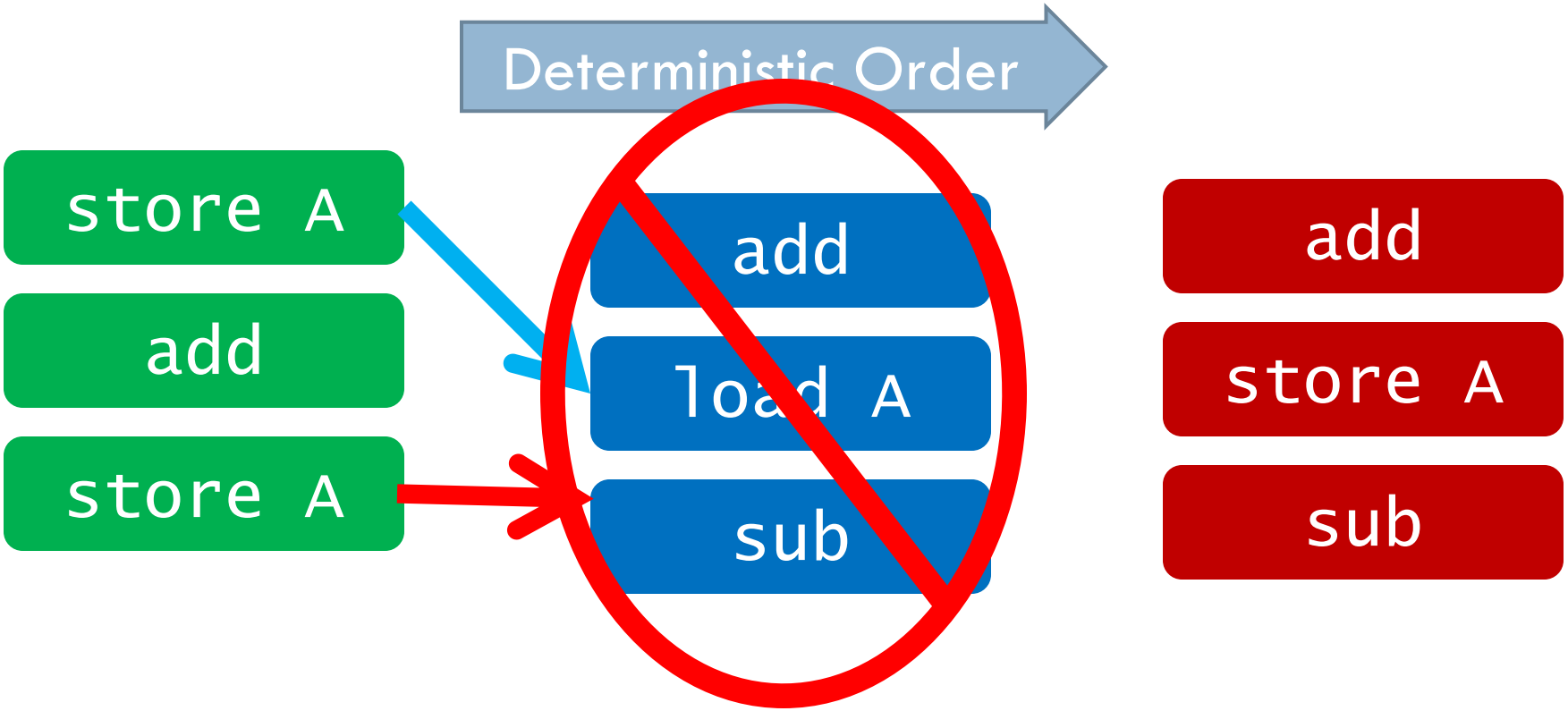
- Determinism makes parallel programming easier
  - Execution is repeatable
  - Simplifies debugging, testing, replicating and deployment
- DMP is a new multiprocessor architecture that provides determinism for arbitrary shared memory programs
  - Leverages existing architectural techniques
  - Performance very close to nondeterministic execution
- **Determinism is a worthwhile and achievable goal**

Questions?

Quantum Rounds

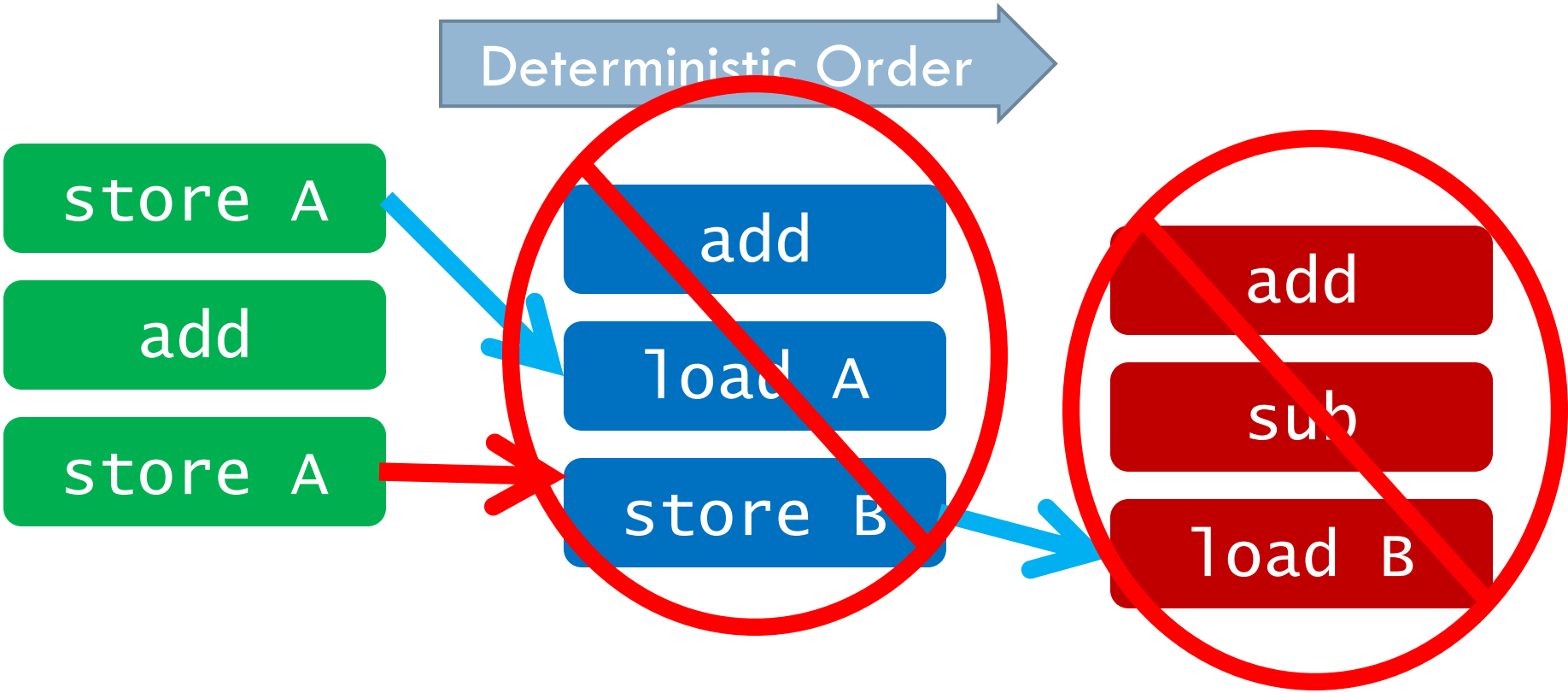


TM-Forward Rollback



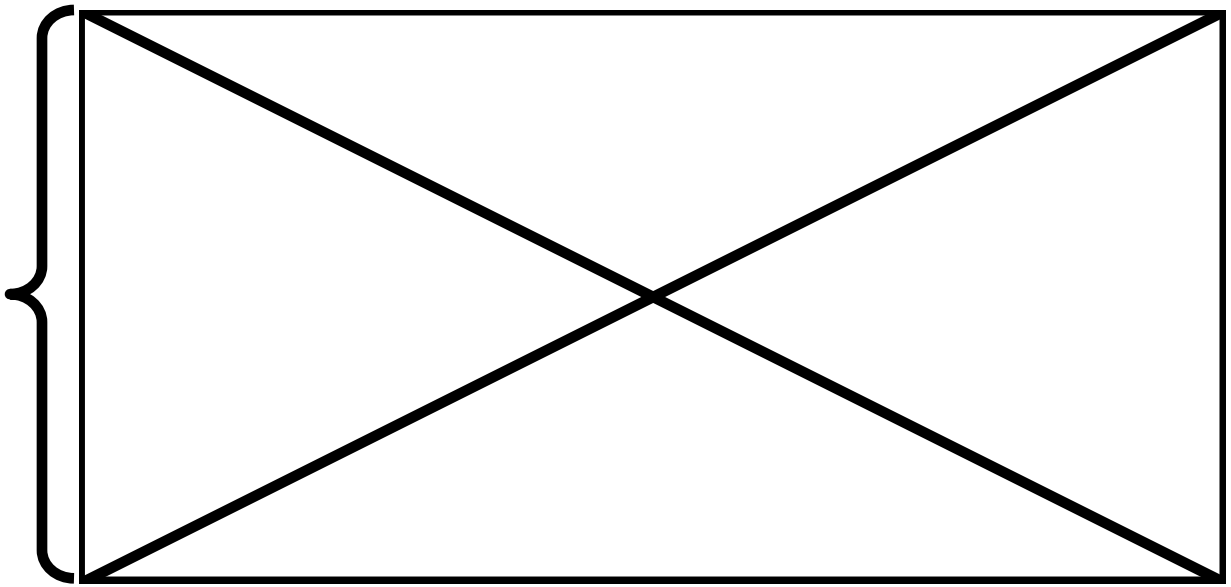
Need value from youngest older store

TM-Forward Recursive Rollback



Flash test slide

View this slide in slide show mode (shift-F5). You should see a blue box here (a Flash movie). It may take a few seconds to load.
It will check Flash movie permissions and interaction with Powerpoint.



Flash → powerpoint interaction works!
This presentation should work fine :-)

Learn more about this presentation at
http://linuxforlovers.wordpress.com/2009/03/15/dmp-asplos-presentation-readme/