DETERMINISTIC MULTIPROCESSORS

ONE INPUT => MULTIPLE OUTPUTS

NONDETERMINISM IS REALLY ANNOYING
DETERMINISM IMPROVES SOFTWARE DEVELOPMENT

Test
- no need to stress test
- tested inputs behave identically in production

Debug
- reverse debugging is possible
- production bugs can be reproduced in-house

Deploy
- more robust production code

testing results are reproducible

DETERMINISM IMPROVES SOFTWARE DEVELOPMENT

Test
- tested inputs behave identically in production

Deploy
- more robust production code

Debug
- reverse debugging is possible
- production bugs can be reproduced in-house

Deploy
- more robust production code

Test
- tested inputs behave identically in production
STARTING SIMPLE: SERIALIZATION

quantum round

T1

T2

T3

deterministic quantum size +
deterministic scheduling

determinism

quanta are necessary for forward progress

time →

RECOVERING PARALLELISM WITH DMP-TSO

parallel mode: buffer all stores (no communication, each thread is a single-threaded program)

commit mode: deterministically publish buffers

serial mode: for atomic ops

T1

wr A rd A

lock A

T2

lock B

rd A

T3

time →
stores are reordered from parallel mode into commit mode

causal cycle → no sequential consistency
ARCHITECTURE MODIFICATIONS

Store Buffers in Private $L2$
application/OS can choose nondeterminism
StoreToSB CommitSB
align context switches with quantum boundaries SaveSB RestoreSB
Precise Insn Counting
StartInsnCount StopInsnCount ReadInsnCount
Traps
SBFull QuantumReached

COHERENCE PROTOCOL MODIFICATIONS

no Exclusive state
merge conflicts deterministically
no false sharing!
THANKS!

- Recent work:
  - deterministic compilers, operating systems, distributed systems, programming languages
- Learn more:
  - sampa.cs.washington.edu
  - devietti@cs
  - luisceze@cs