DMP
Deterministic Shared Memory Multiprocessing

This work originally appeared at ASPLOS 2009

Owen Anderson, Tom Bergan, Joe Devietti,
Nick Hunt, Brandon Lucia
Luis Ceze, Steve Gribble, Dan Grossman, Mark Oskin
A multithreaded medical billing system

```c
while (moreBills) {
    load t <- amtOwed
    t += 1e6
    store t -> amtOwed
}

amtOwed==2e6

while (moreBills) {
    load t <- amtOwed
    t += 1e6
    store t -> amtOwed
}

amtOwed==1e6
```
A multithreaded medical billing system

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 programs

- hard to debug
- hard to test
- hard to replicate
DMP from 10,000’

- We only care about communicating instructions
- Deterministic serialization → same communication
  - …but what about performance?
- Recover parallelism from non-communicating insns
Talk Outline

PERFORMANCE

DMP-Serial DMP-SharingTable DMP-TM DMP-TMForward

COMPLEXITY ↔

DMP: Deterministic Shared Memory Multiprocessing
DMP-Serial Example
DMP- Sharing Table

- Communicating insns cause cache line state transitions
- Break each quantum into communication-free parallel prefix and communicating serial suffix
DMP-TM: Recovering Parallelism with Speculation

- DMP-SharingTable conservatively assumes that all cache line state transitions are communication
  - ...but many transitions are not

- Use TM support to speculate that a quantum is not involved in communication
  - If communication happens, rollback + re-execute
  - Commit quanta in-order (need DT to commit)
DMP-TM Example
Simulation Results

Simulation results showing the runtime normalized to nondeterministic parallel execution for different thread counts (4, 8, 16) across different benchmark suites: splash2, benchmark suite, and parsec. The simulation results are categorized into four types: DMP-Serial, DMP-ShTab, DMP-TM, and DMP-TMForward.
Determinism @ UW

- CoreDet compiler+runtime
  - highly scalable, software-only determinism for arbitrary programs
- dthreads deterministic synchronization library
  - leverages race-freedom assumption for performance
- Deterministic Operating System
  - studies the full-system implications of determinism
  - best acronym ever!
Conclusions

- Determinism makes parallel programming easier
  - Execution is repeatable
  - Simplifies debugging, testing, replicating and deployment
- We are pursuing a variety of deterministic platforms
  - new computer architectures
  - new operating system designs
  - new compiler extensions
- Determinism is a worthwhile and achievable goal