Deterministic Shared Memory Multiprocessing

Presenter: Wu, Weiyi
Yale University
Outline

- What is determinism?
- How to make execution deterministic?
- What’s the overhead of determinism?
What Is Determinism?

• The SAME input, the SAME output
• The same execution sequence (with Linearizability) (strong)
• The same resource-accessing ordering (weak)
(Weak) Determinism

Concurrent Program

Execution 1

Execution 2
Non-Determinism

Time Flow

Communications

Concurrent Program

Execution 1

Execution 2
Non-determinism in Practice

Figure 3. Amount of nondeterminism over the execution of *barnes* and *ocean-contig*. The x axis is the position in the execution where each sample of 100,000 instructions was taken. The y axis is $ND$ (Eq. 1) computed for each sample in the execution.
Is Record’n’Replay Deterministic?

• Yes!
  • When the log is replayed deterministically, the replayed execution is definitely deterministic

• No...
  • It cannot deploy to any machines and still run deterministically for any input
How to Execute Deterministically?

- Eliminating all instruction communications
- Deterministically arrange all communications
  - Read-after-Write
  - Write-after-Write, Write-after-Read
DMP-Serial

- No concurrency!
  Make it single-threaded!

- Programs are divided into quanta
- All processors act as one
  - Only execute when hold token
  - Deterministically pass token when quanta end
Example of DMP-Serial

Concurrent Program

Small Quanta

Bigger Quanta

Time Flow
Token Pass
Quantum
Recovering Parallelism

• DMP-Serial is way too strong
• Weak determinism is also acceptable
  • Instructions without communication can execute concurrently
  • Deterministically schedule communications
DMP-ShTab

• Break quantum into parallel prefix and serial suffix (dynamically), do prefix concurrently

• Deterministically control shared memory access
  • Only read from Shared memory position
  • Only write to Private memory position
Example of DMP-ShTab

Concurrent Program

DMP-Serial

DMP-ShTab

Time Flow
Token Pass
Quantum
Similarity to MESI coherence protocol

- **Invalid**
  - Reset
  - Read Miss, Shared
  - Read Hit
  - Probe Read Hit

- **Shared**
  - Read Miss, Shared
  - Read Hit
  - Probe Read Hit

- **Exclusive**
  - Reset
  - Read Miss, Exclusive
  - Read Hit
  - Write Hit
  - Probe Write Hit

- **Modified**
  - Write Miss
  - Probe Read Hit
  - Write Hit
  - Read Hit
  - Write Hit

Transitions include:
- Read Hit → Read Hit
- Read Miss, Exclusive → Read Miss, Exclusive
- Write Hit → Write Miss
- Probe Read Hit → Probe Read Hit
- Probe Write Hit → Probe Write Hit
Similarity to MESI coherence protocol
DMP-TM

- Atomicity, isolation and deterministic total order of quanta will guarantee determinism
- Make quantum transaction to explicitly execute concurrently
  - Deterministically commit
  - Abort and re-execute latter quantum when conflicting
DMP-TMFwd

- Read-after-Write will be no conflict if propagating modified values in time
- Forward writes across transactions
- Need to abort “infected” transactions when aborting
Example of DMP-TM

Figure 7. Recovering parallelism by executing quanta as memory transactions (a). Avoiding unnecessary squashes with un-committed data forwarding (b).
How to build quanta?

• Split code evenly - **QB-Count**

• The less waiting time, the better

• No spinning on lock - **QB-SyncFollow**
  (token is enough for synchronization)

• End quantum when finishing working on shared data others wait for - **QB-Sharing**
Example of QB-SyncFollowing

**Figure 8.** Example of a situation when better quantum breaking policies leads to better performance.
Example of QB-Sharing

Figure 4. Deterministic serialization of memory operations. Dots are memory operations and dashed arrows are happens-before synchronization.
Overhead & Scalability

Figure 9. Runtime overheads with 4, 8 and 16 threads. (P) indicates page-level conflict detection; line-level otherwise.
Quanta Sensitivity

**Figure 10.** Performance of 2,000 (2), 10,000 (X) and 100,000 (C) instruction quanta, relative to 1,000 instruction quanta.

**Figure 12.** Performance of QB-Sharing (s), QB-SyncFollow (sf) and QB-SyncSharing (ss) quantum builders, relative to QB-Count, with 1,000-instruction quanta.
Software-DMP

Figure 14. Runtime of Sw-DMP-ShTab relative to nondeterministic execution.
Conclusion

- (First?) implementation of DMP
- Demonstration of low-overhead determinism for concurrent programs
- Not pervasive
  - No deterministic interrupts
  - Can not save nondeterministic hardwares
Thanks~