

# High Performance Hardware Transactional Memory does not Equal High Performance Transaction Systems

Justin Levandoski (Microsoft)

Darko Makreshanski (ETH Zurich)

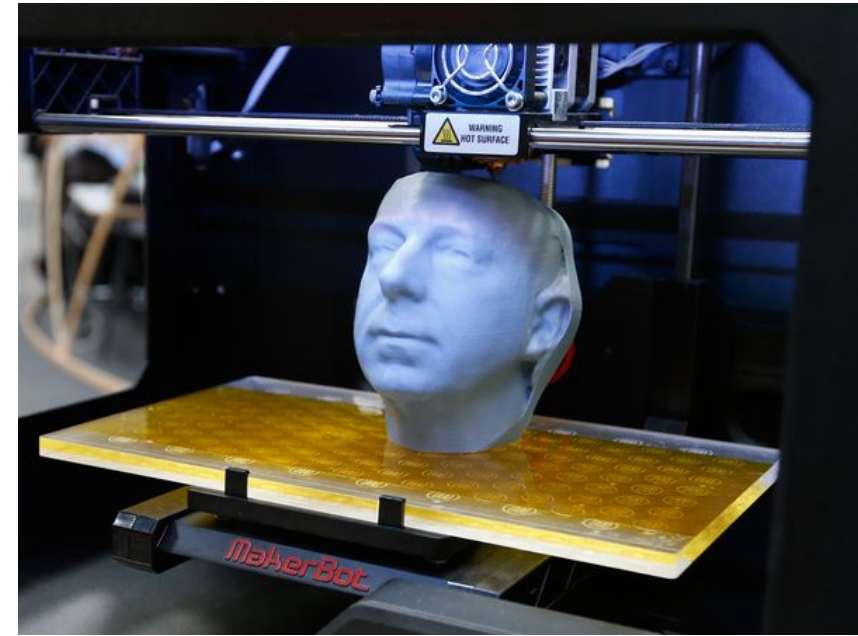
Ryan Stutsman (Utah)

# Lock-free Programming



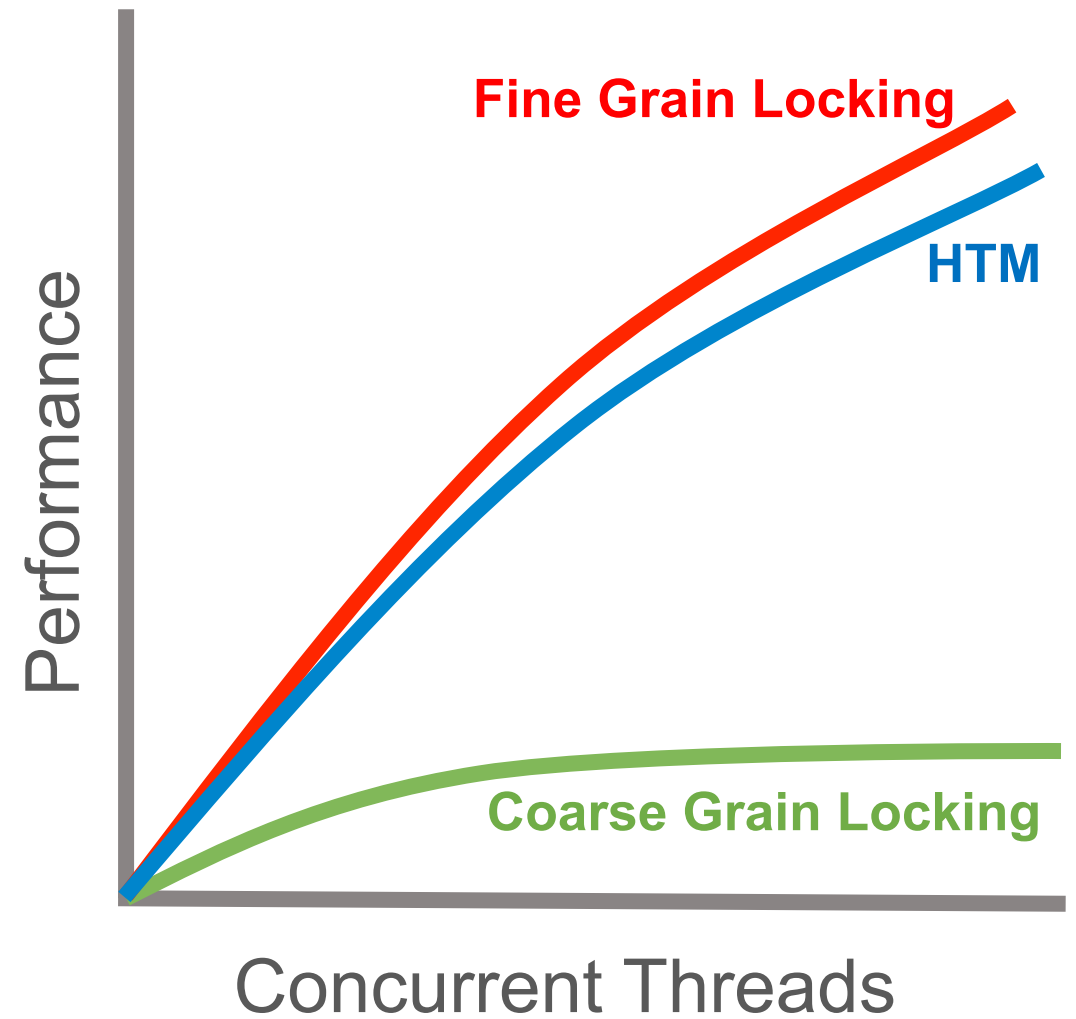
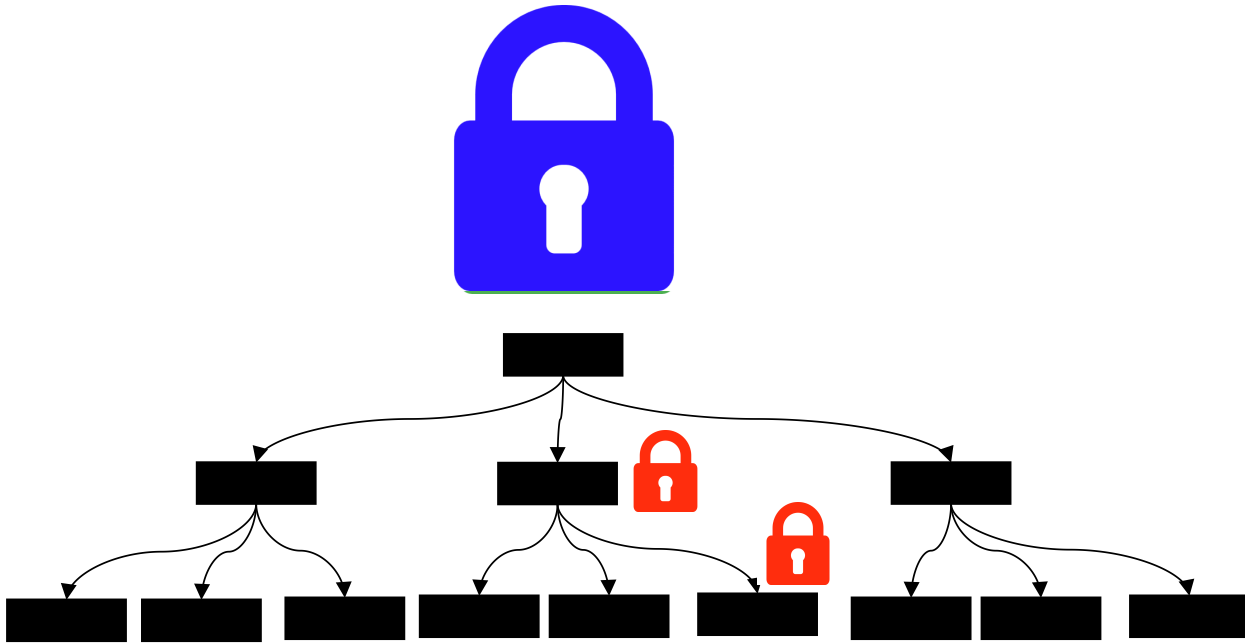
1. Carefully design/implement using CAS/FAI
2. Cross fingers
3. System crashes or corruption
4. Goto 1

# Hardware Transactional Memory

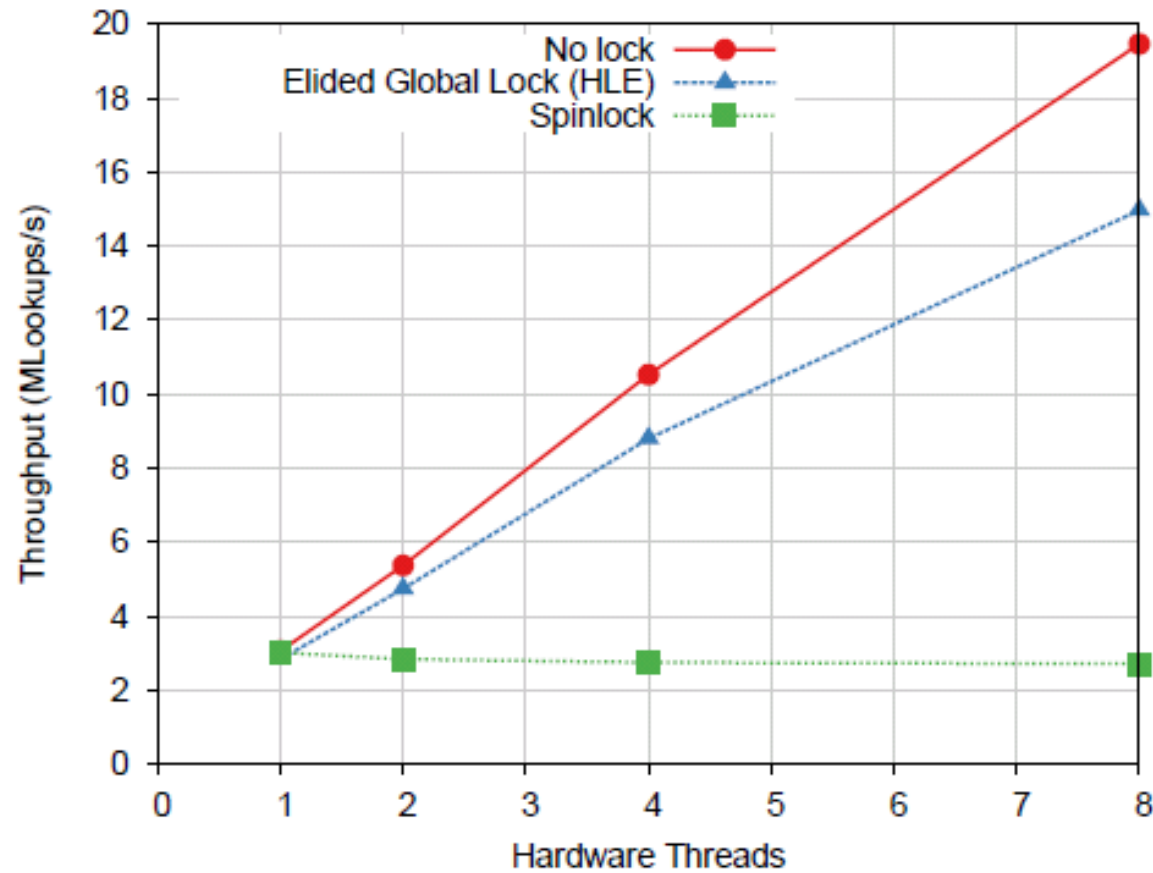


```
If (BeginTransaction())  
Then  
    < Critical Section >  
    CommitTransaction()  
Else  
    < Abort Fallback Codepath >  
EndIf
```

# Lock Elision

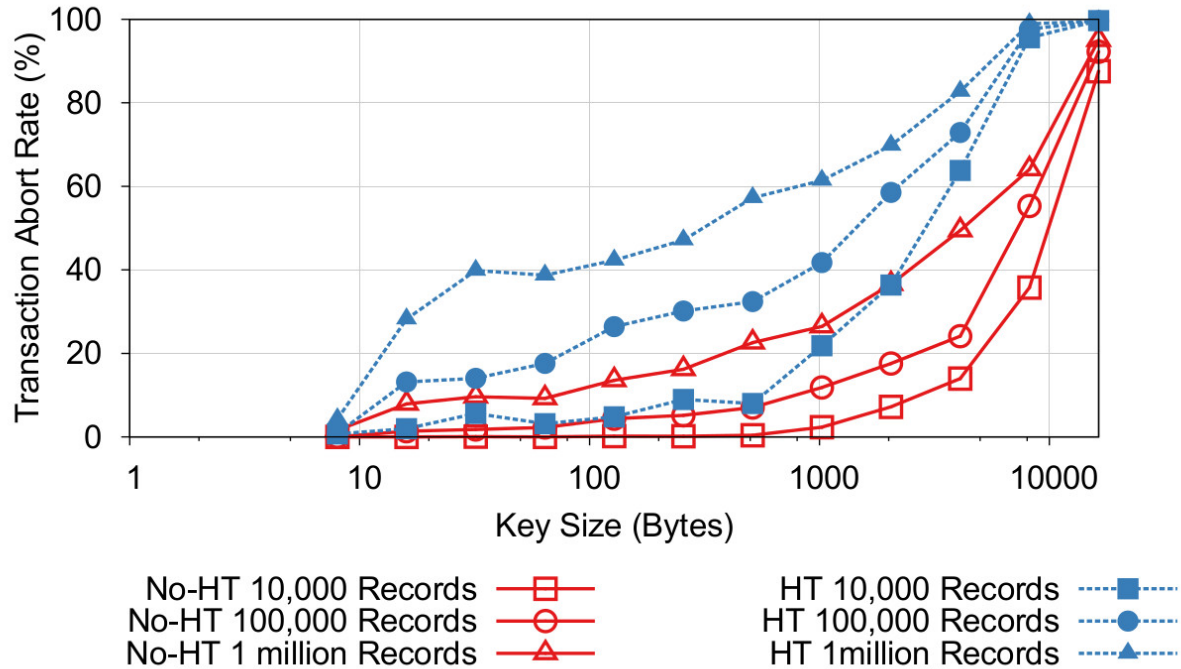


# Read Only Workload Performance (B+-Tree)



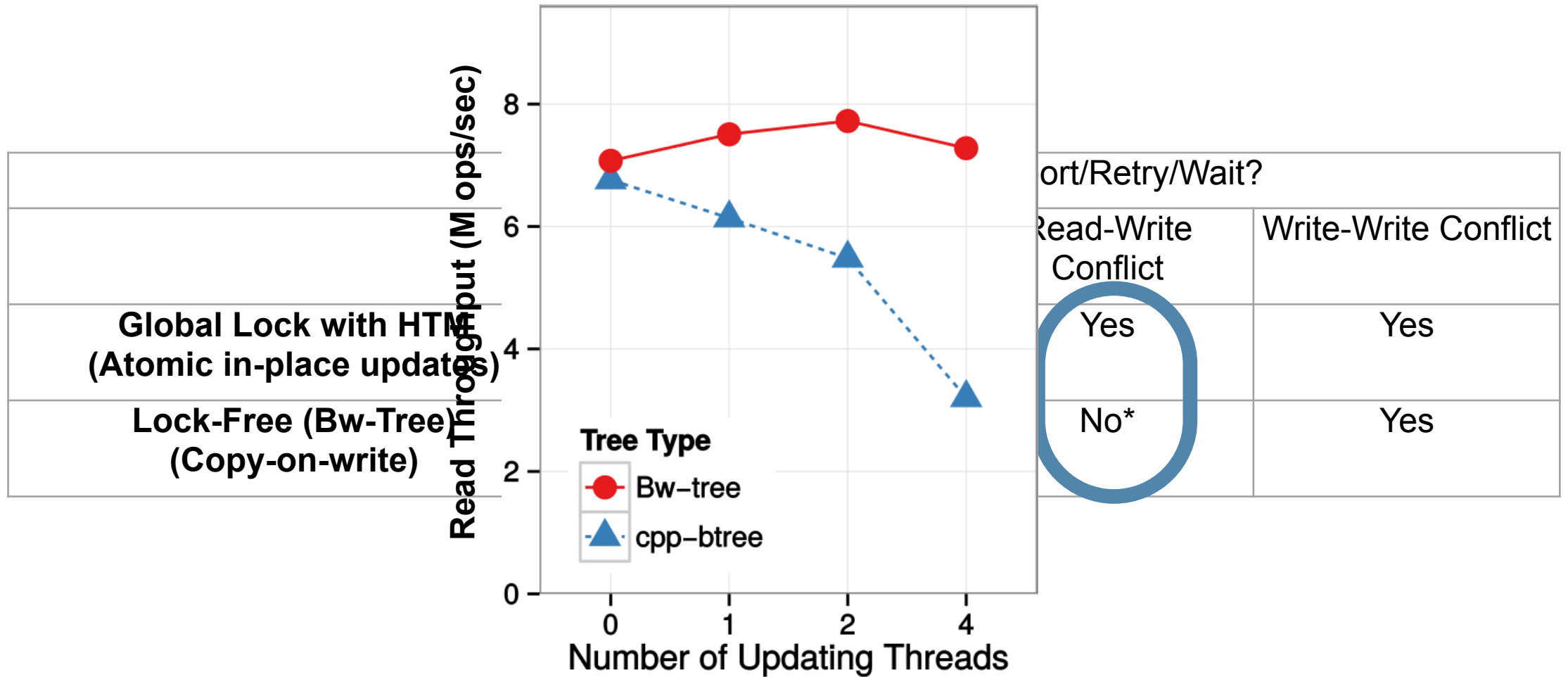
- 8B Keys, 8B Payloads
- 1M Records

# #1: HTM Not (yet) a General Purpose Solution



- Transaction limited by cache capacity
- Abort on cache line eviction
- Abort on interrupt, context switch, page fault, etc.
- Lock “crabbing” not possible

# #2: Fundamental Difference(s)



# #3: HTM Sweet Spot (Multi-Word Compare and Swap)

- Atomically update a handful of words
- Keeps transaction capacity low and length short (predictable)
- Beware: readers should execute in transactions (not free)