Transaction Processing is easy if you're God

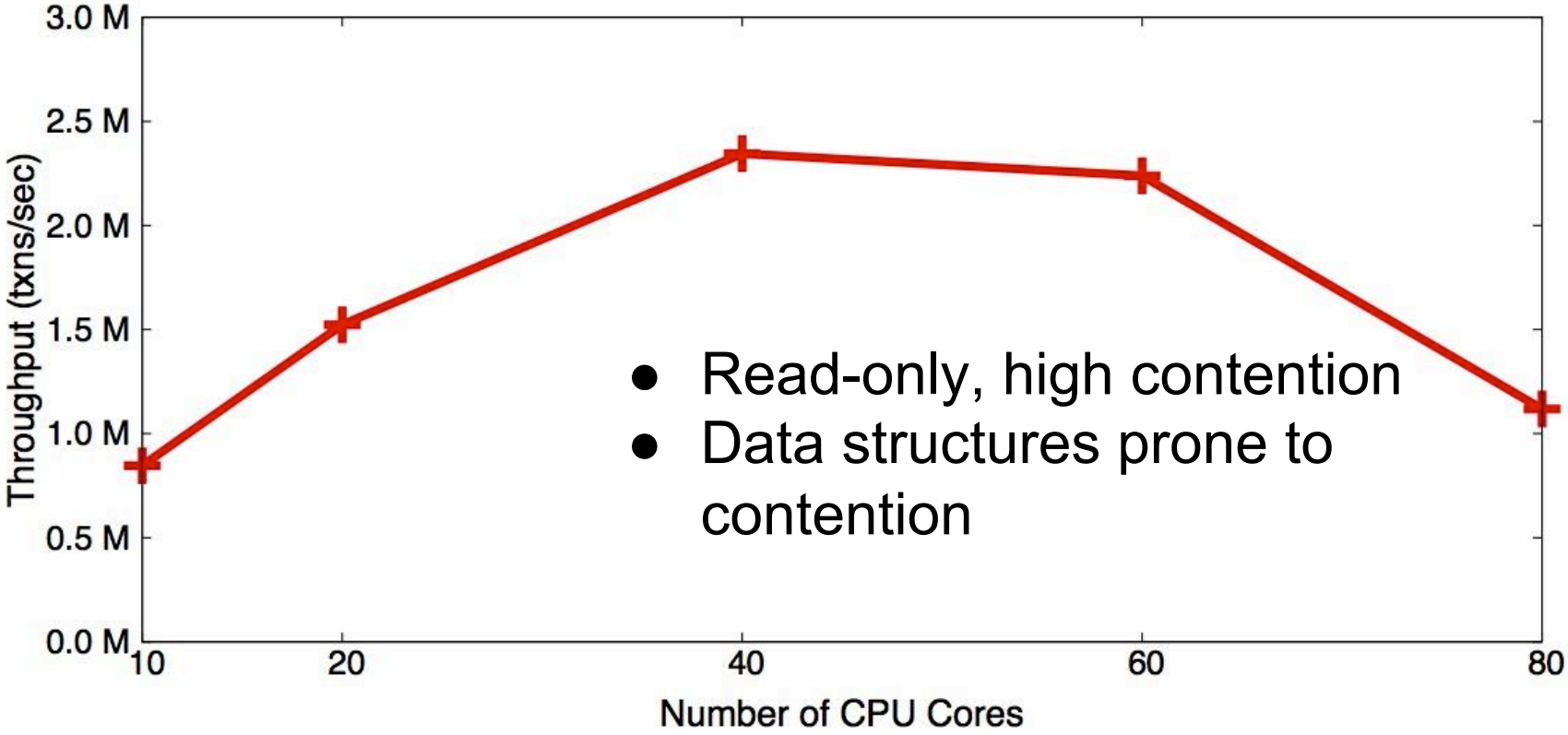
Jose Faleiro
Yale University

The Problem: Concurrency Control

- Don’t expect linear scalability under contention
  - Real conflicts among txns
  - Limited concurrency among conflicting txns

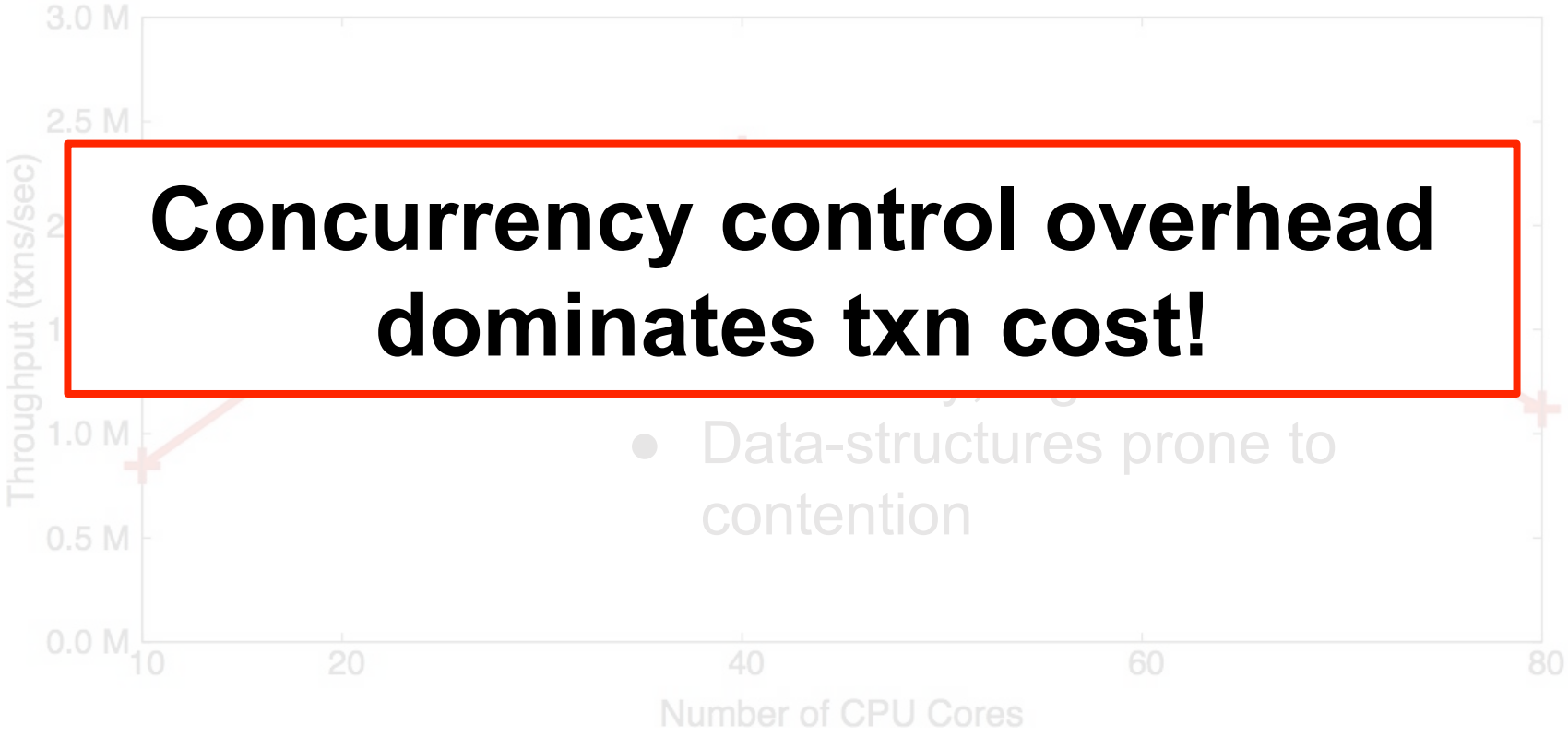
- But concurrency control exacerbates the problem
  - Data-structures prone to contention; e.g. latches, lock lists
  - Scheduling overhead

The Problem: Concurrency Control



- Read-only, high contention
- Data structures prone to contention

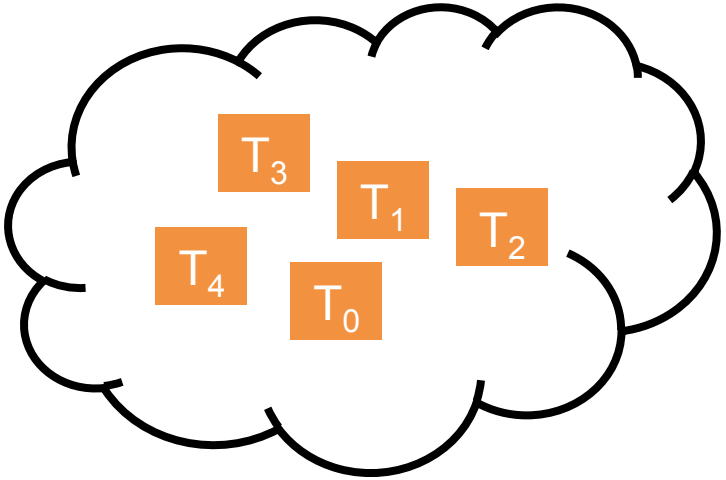
The Problem: Concurrency Control



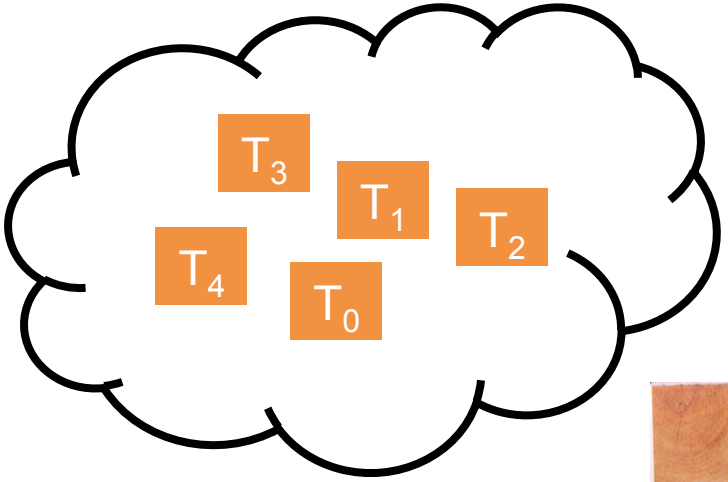
Concurrency control overhead dominates txn cost!

- Data-structures prone to contention

What would God do?

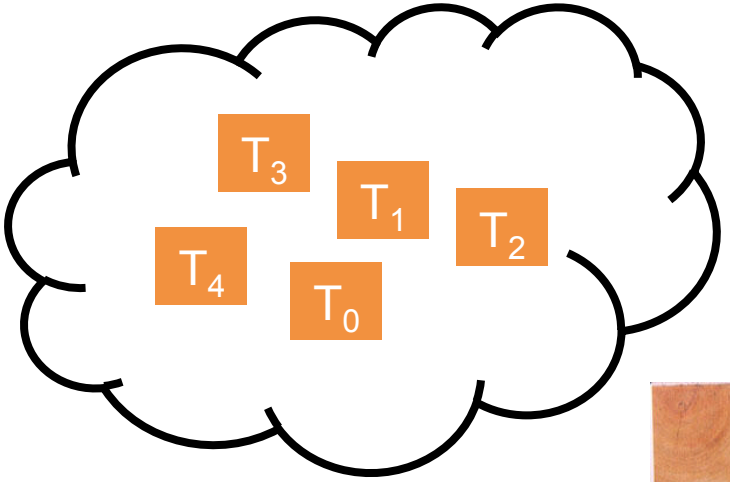


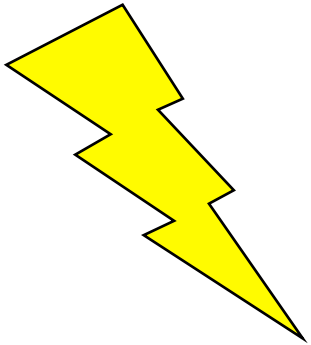
What would God do?

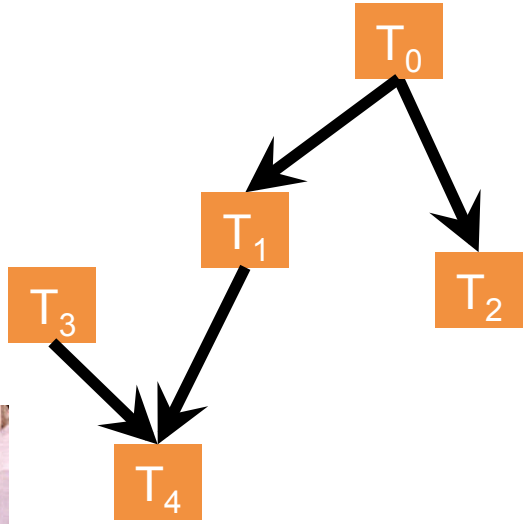


Let there be (a serializable) order!

What would God do?







Let there be
(a serializable)
order!

God's Concurrency Control Algorithm

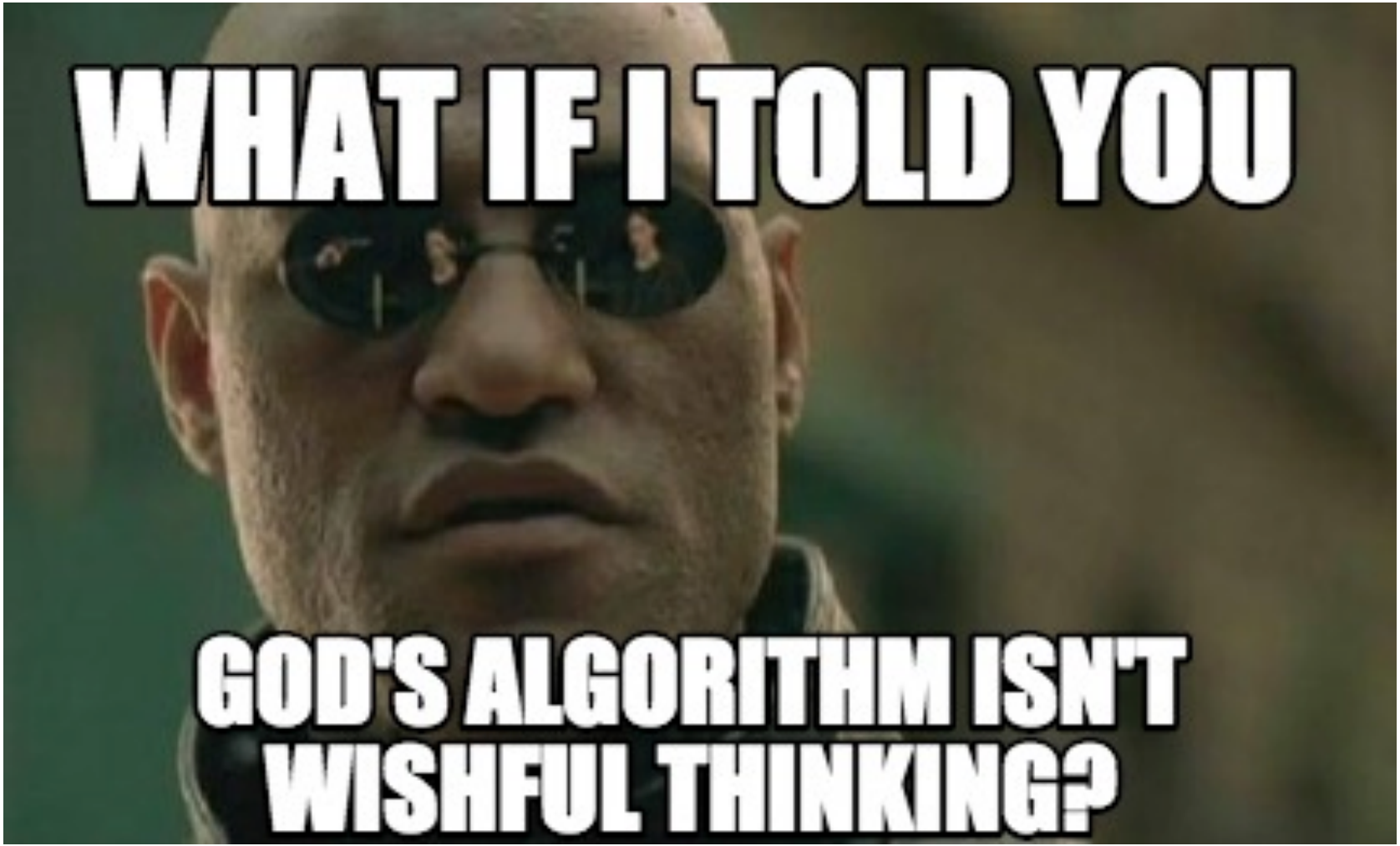
- Use omniscience to pull serializable order from thin air
- DB executes txns based on serializable order

God's Concurrency Control Algorithm

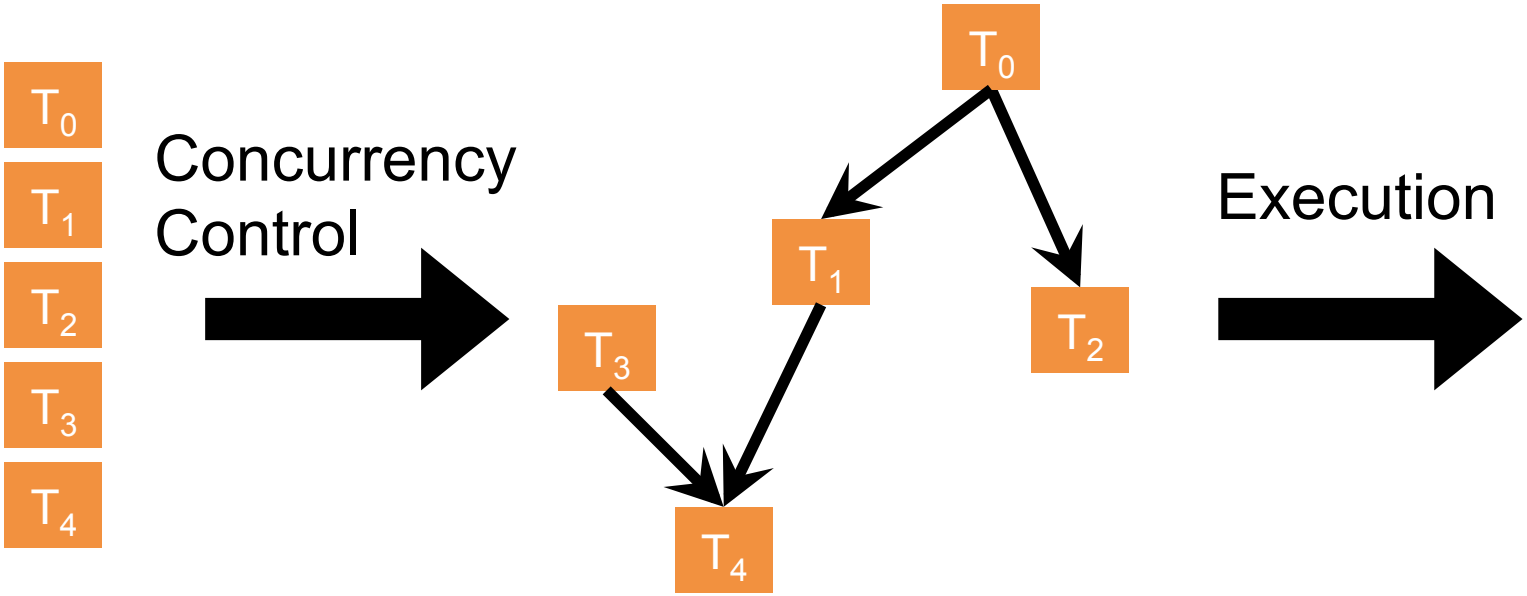
Serializable (*not* serial!) order given up front

vs

Obtain serializable order at execution time

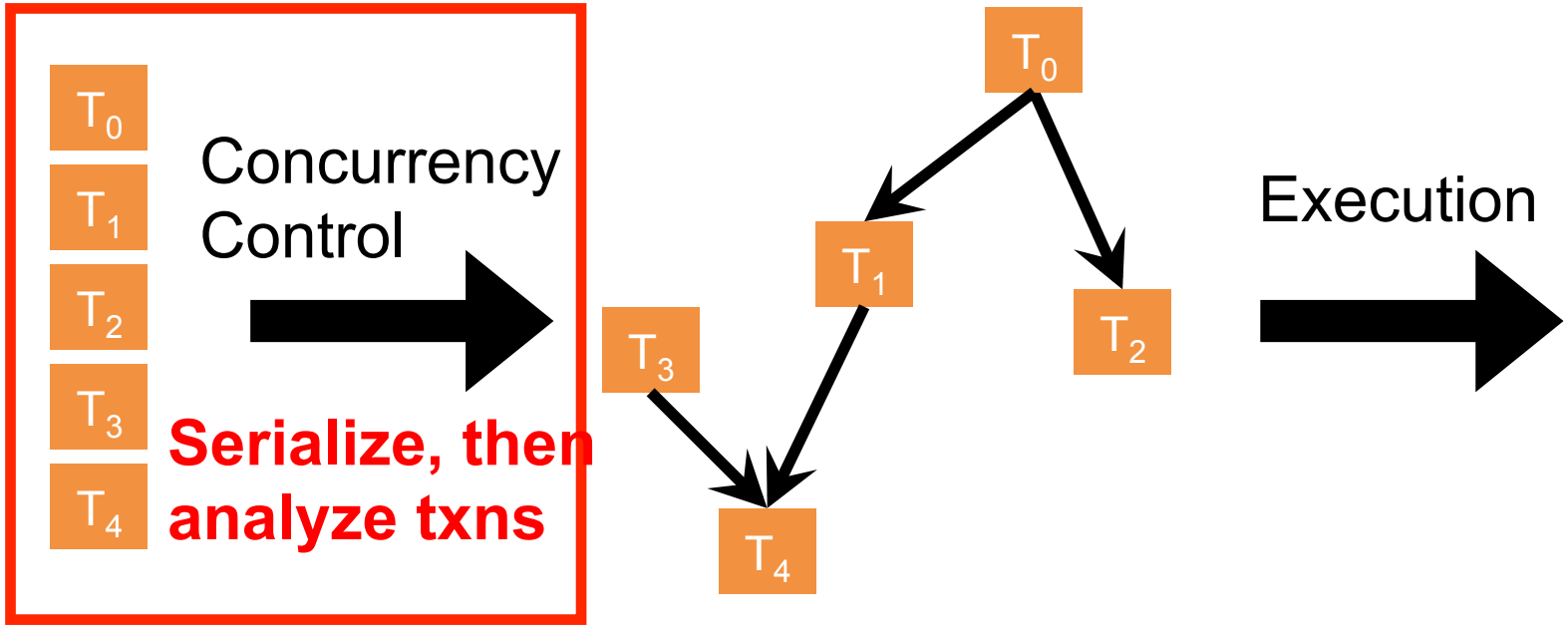

WHAT IF I TOLD YOU
GOD'S ALGORITHM ISN'T WISHFUL THINKING?

Example: Bohm



[Faleiro and Abadi, VLDB'15]

Example: Bohm



[Faleiro and Abadi, VLDB'15]

Divine Design

- Determine serializable order prior to execution
  - No concurrency control overhead during execution
- Minimal scheduling overhead
  - Efficiency of optimism/speculation
  - Contention resilience of pessimism
- Tradeoffs
  - Latency for throughput
  - Benefits applicable to workloads amenable to analysis