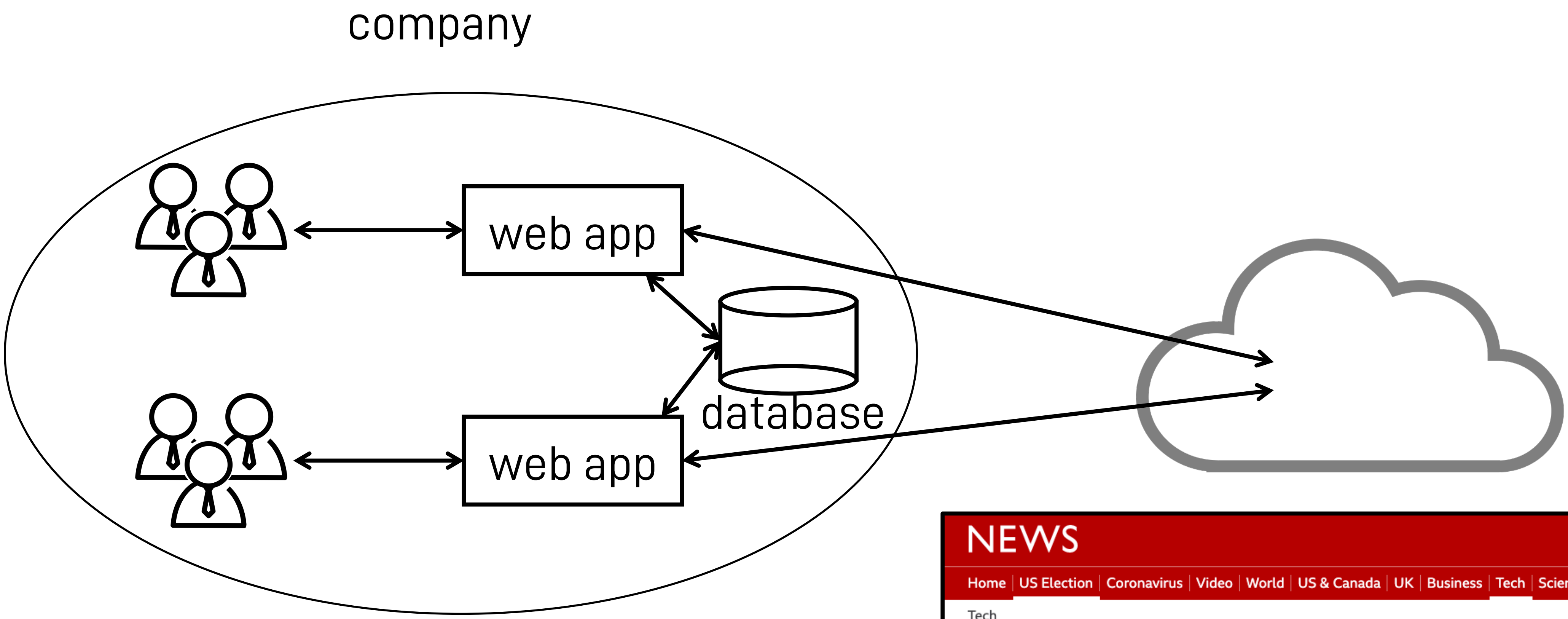
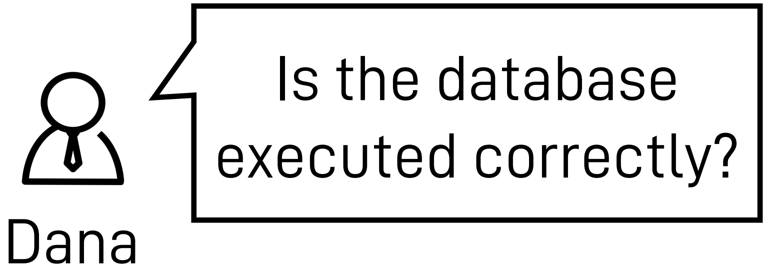
Cobra: Making Transactional Key-Value Stores Verifiably Serializable

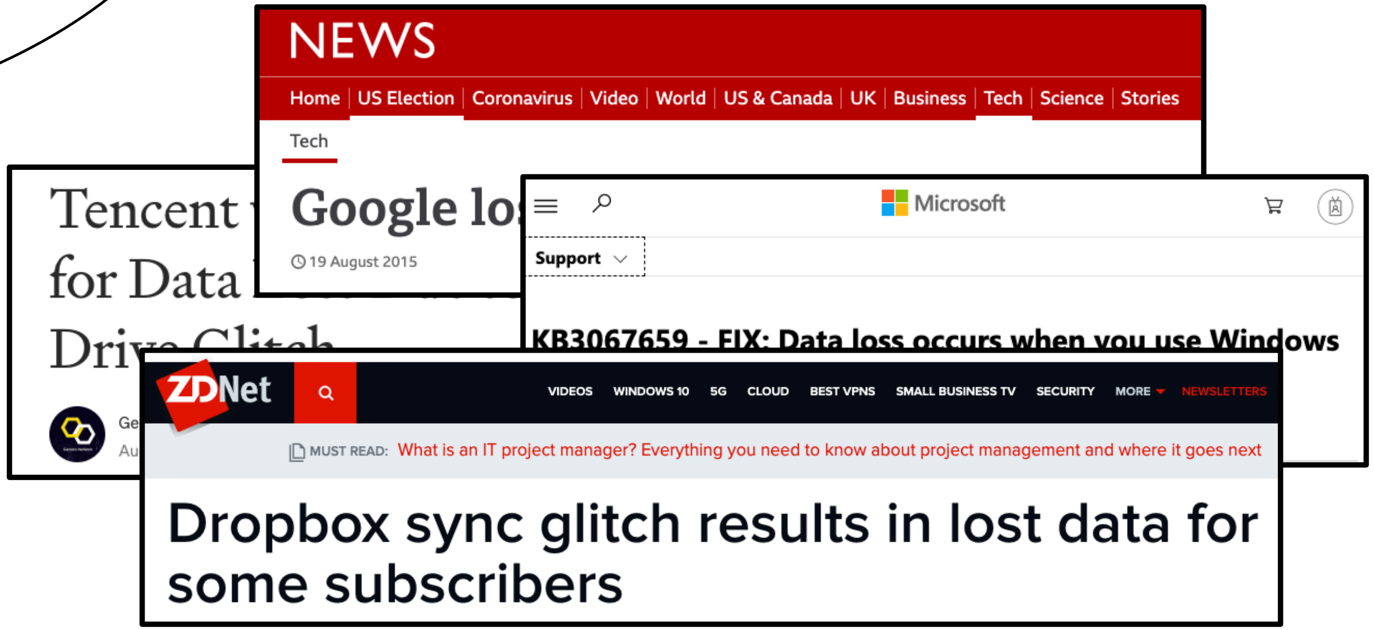
Cheng Tan, Changgeng Zhao, Shuai Mu*, Michael Walfish

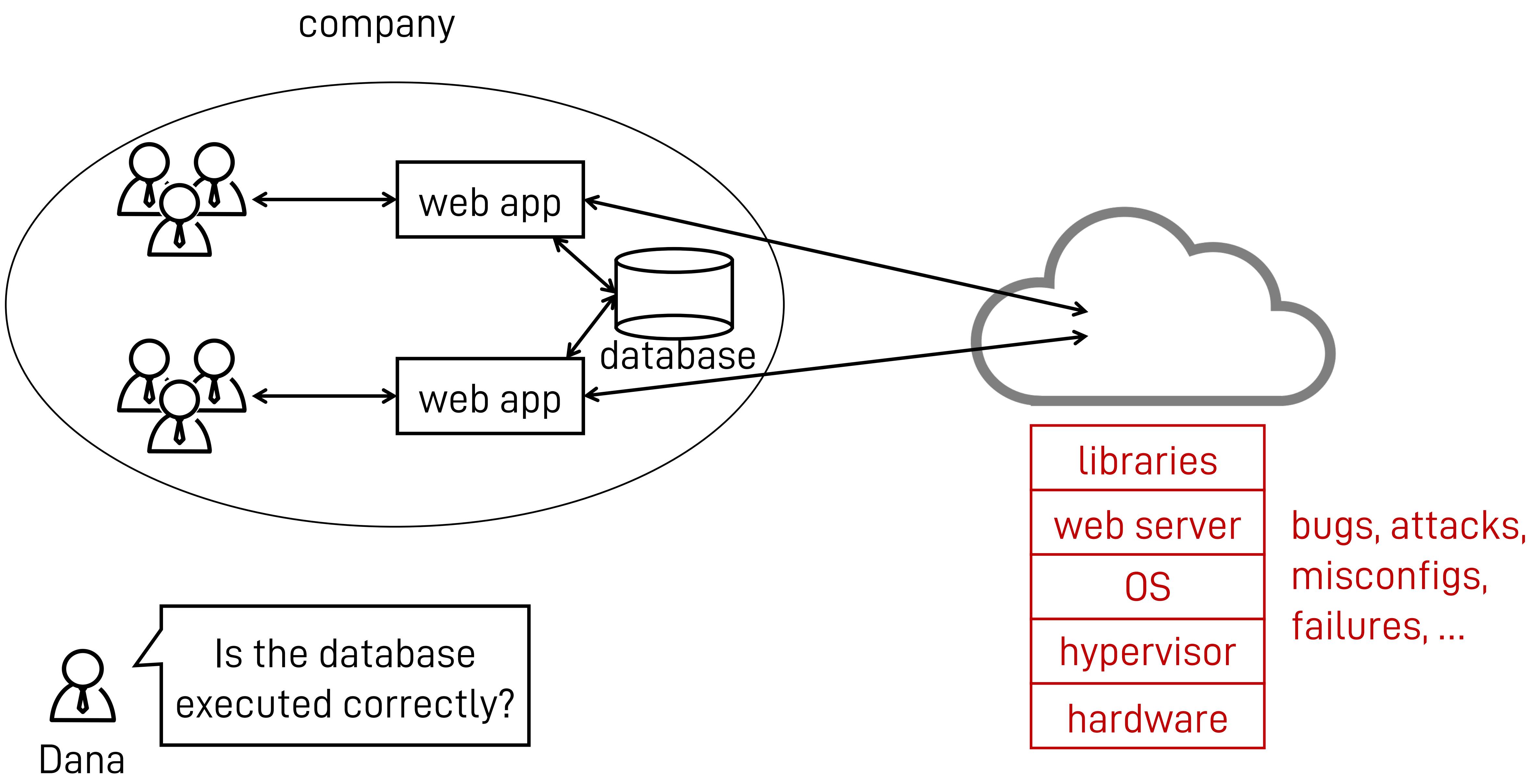
NYU Computer Science Department, Courant Institute

*Stony Brook University

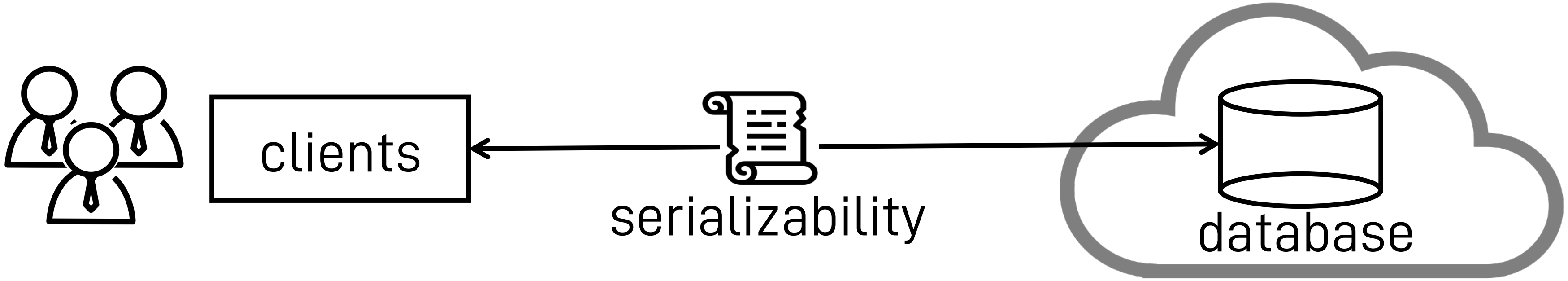

company
web app
web app
database


Is the database executed correctly?
Dana


NEWS
Home | US Election | Coronavirus | Video | World | US & Canada | UK | Business | Tech | Science | Stories
Tech
19 August 2015
Tencent
for Data
Microsoft
Support
KB3067659 - FIX: Data loss occurs when you use Windows
ZDNet
VIDEOS WINDOWS 10 5G CLOUD BEST VPNS SMALL BUSINESS TV SECURITY MORE NEWSLETTERS
MUST READ: What is an IT project manager? Everything you need to know about project management and where it goes next
Dropbox sync glitch results in lost data for some subscribers

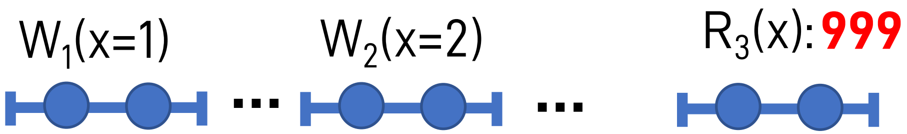

company
web app
web app
database
libraries
web server
OS
hypervisor
hardware
bugs, attacks,
misconfigs,
failures, ...
Is the database
executed correctly?
Dana

Serializability is a correctness contract

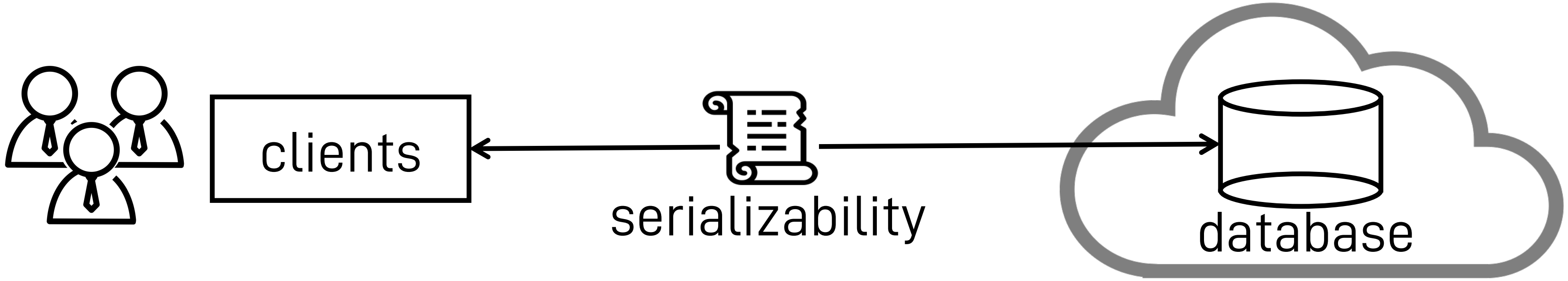


clients' transactions = sequential execution of these transactions

- Serializability implies basic data integrity, tolerating faults, ...



Serializability is a correctness contract



gold standard
isolation level
&
correctness contract

serializability
violations exist in
production systems

The underlying problem is ...

a) ... black-box checking of

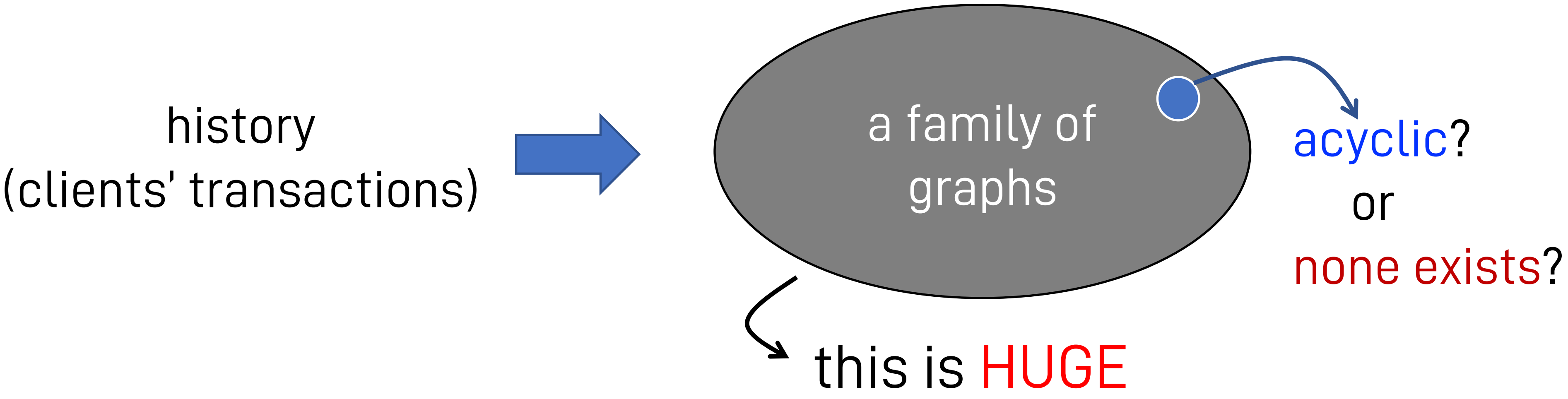
b) ... serializability

c) ... while scaling to real-world workloads.

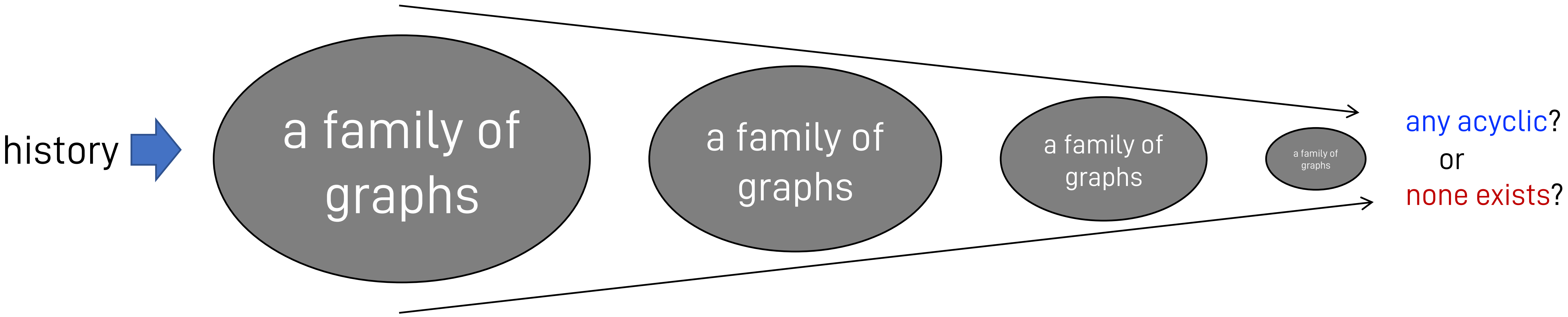
Note: we verify the executions, not the implementation.

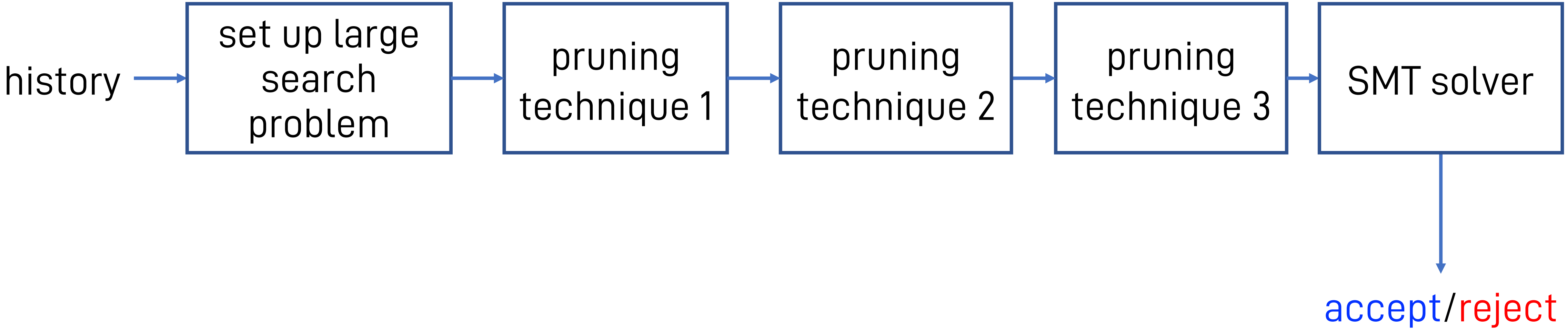
Cobra: verifying serializability of black-box databases

Cobra:

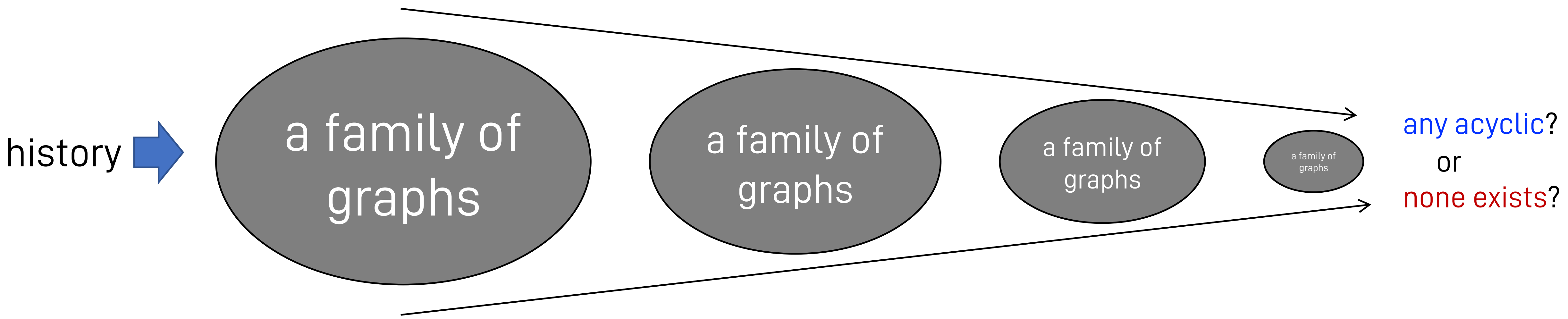


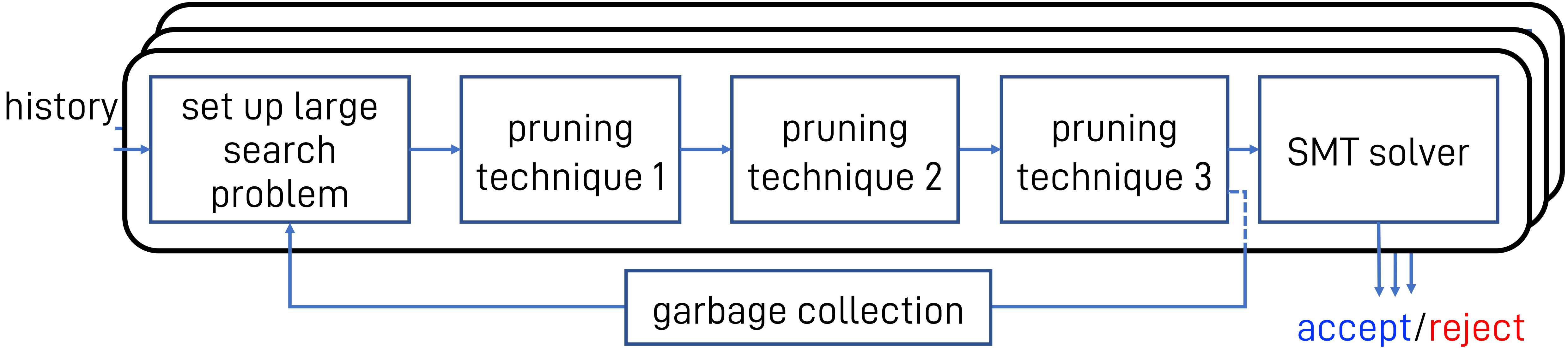
Cobra: verifying serializability of black-box databases





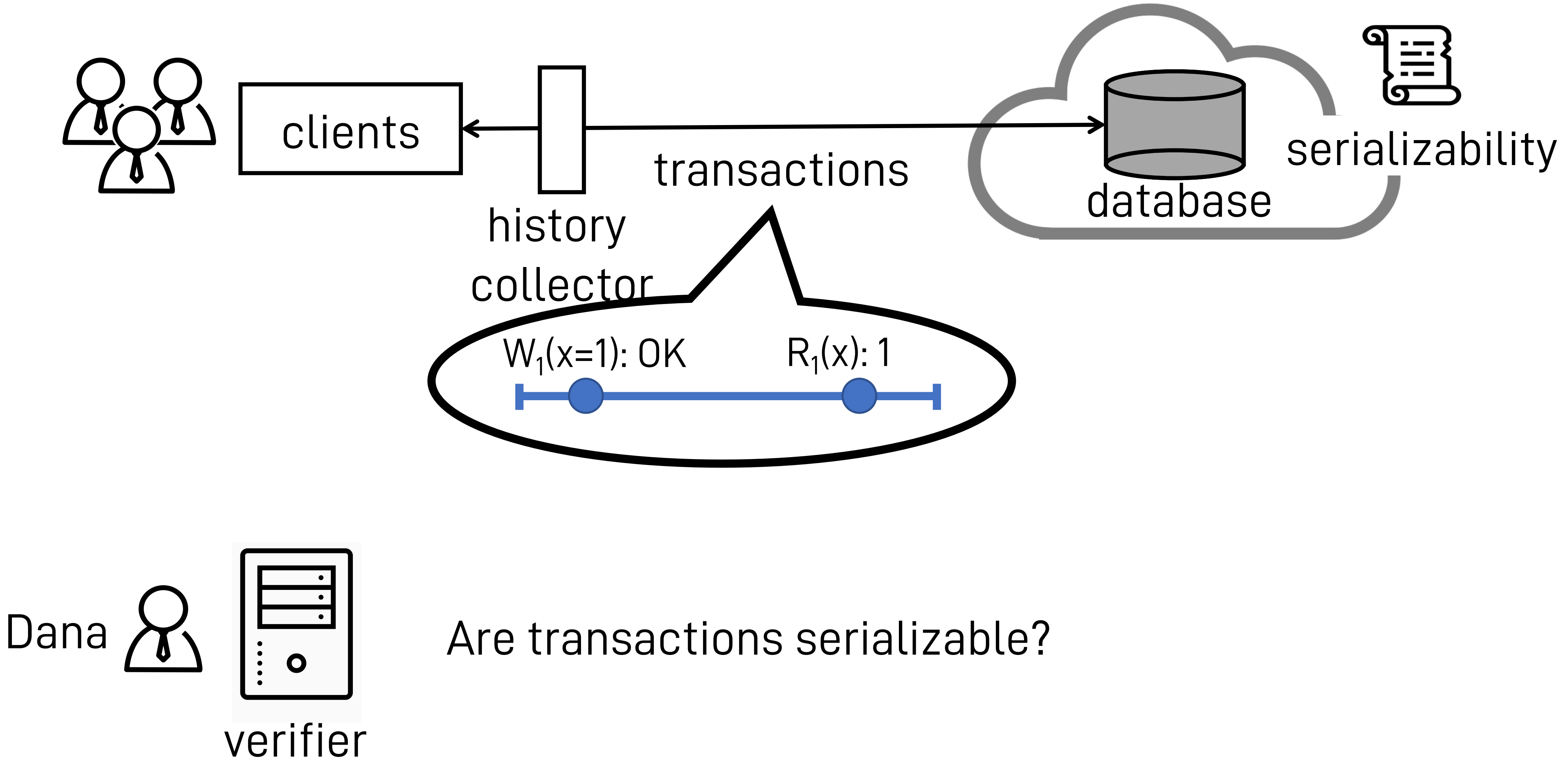
Cobra: verifying serializability of black-box databases





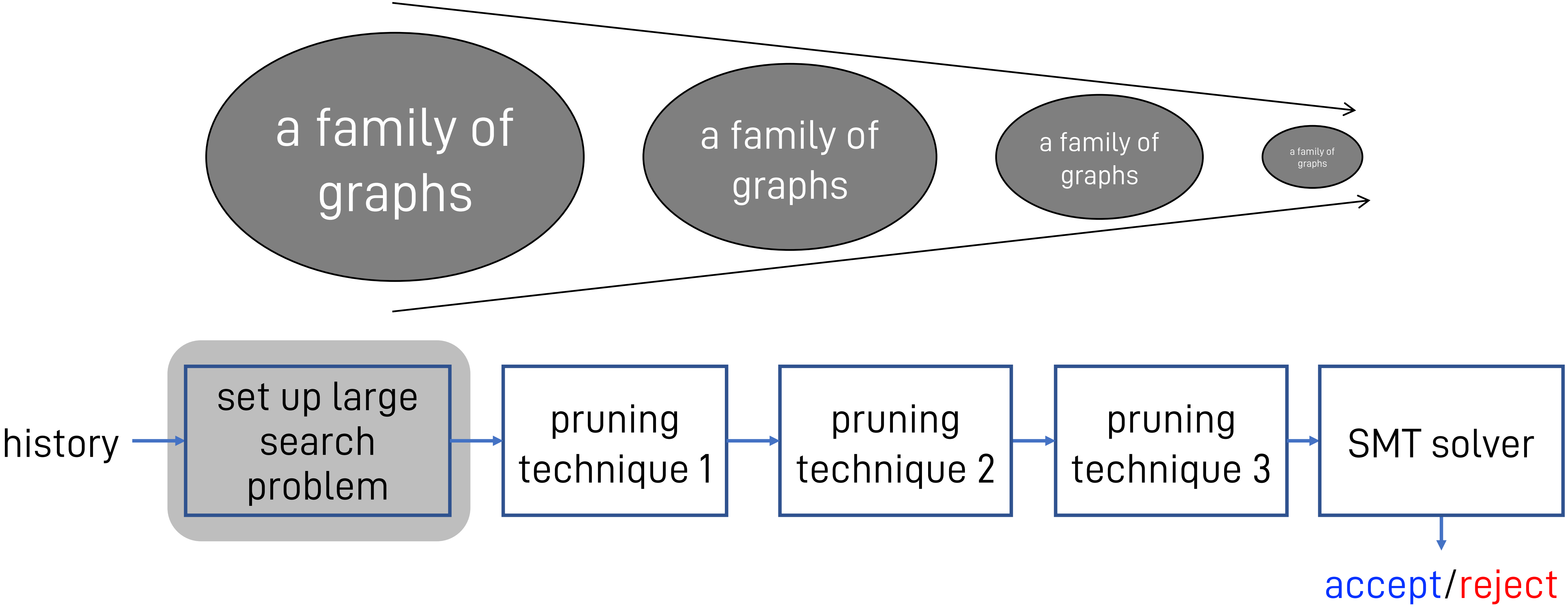
Rest of the talk

- The underlying problem
- Solution: Cobra


clients
history
collector
transactions
$W_1(x=1)$: OK
$R_1(x)$: 1
database
serializability
Dana
verifier
Are transactions serializable?

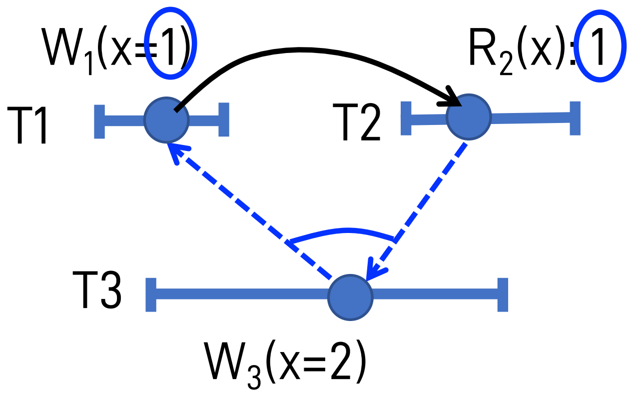
Starting point: brute-force search on a polygraph

- [Papadimitriou 79]: build a polygraph (a family of graphs) and search
- Step 1: building a polygraph from a history
- Step 2: searching for an acyclic graph


a family of graphs
a family of graphs
a family of graphs
a family of graphs
history
set up large search problem
pruning technique 1
pruning technique 2
pruning technique 3
SMT solver
accept/reject

Starting point: brute-force search on a polygraph

- [Papadimitriou 79]: build polygraph and search
- Step 1: building a polygraph from a history
- Step 2: searching for an acyclic graph



polygraph = (V, E, C)

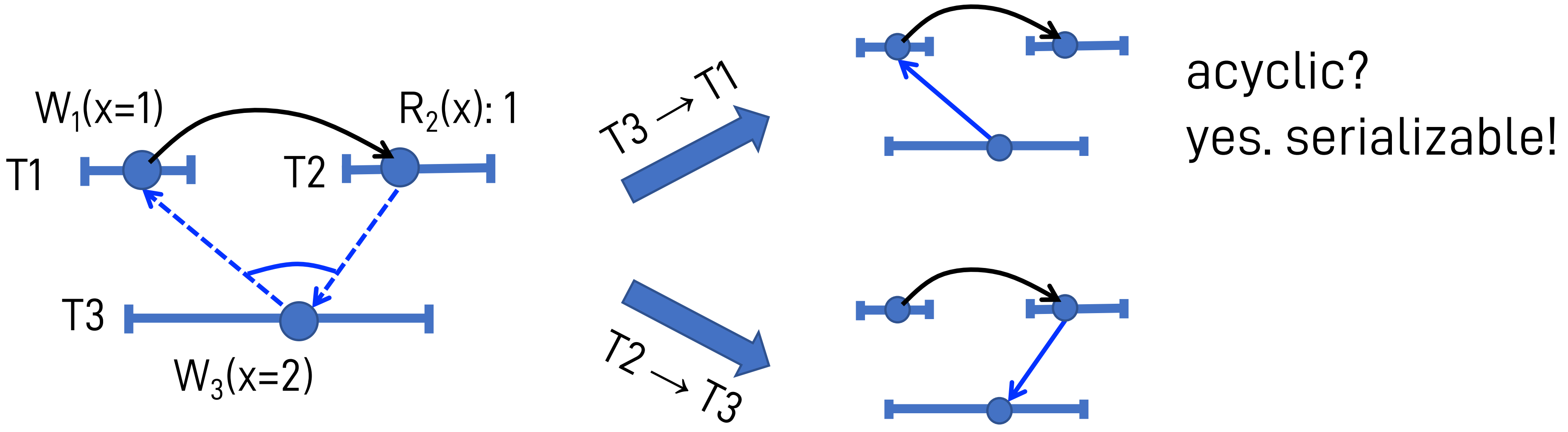
V = { T1 , T2 , T3 }

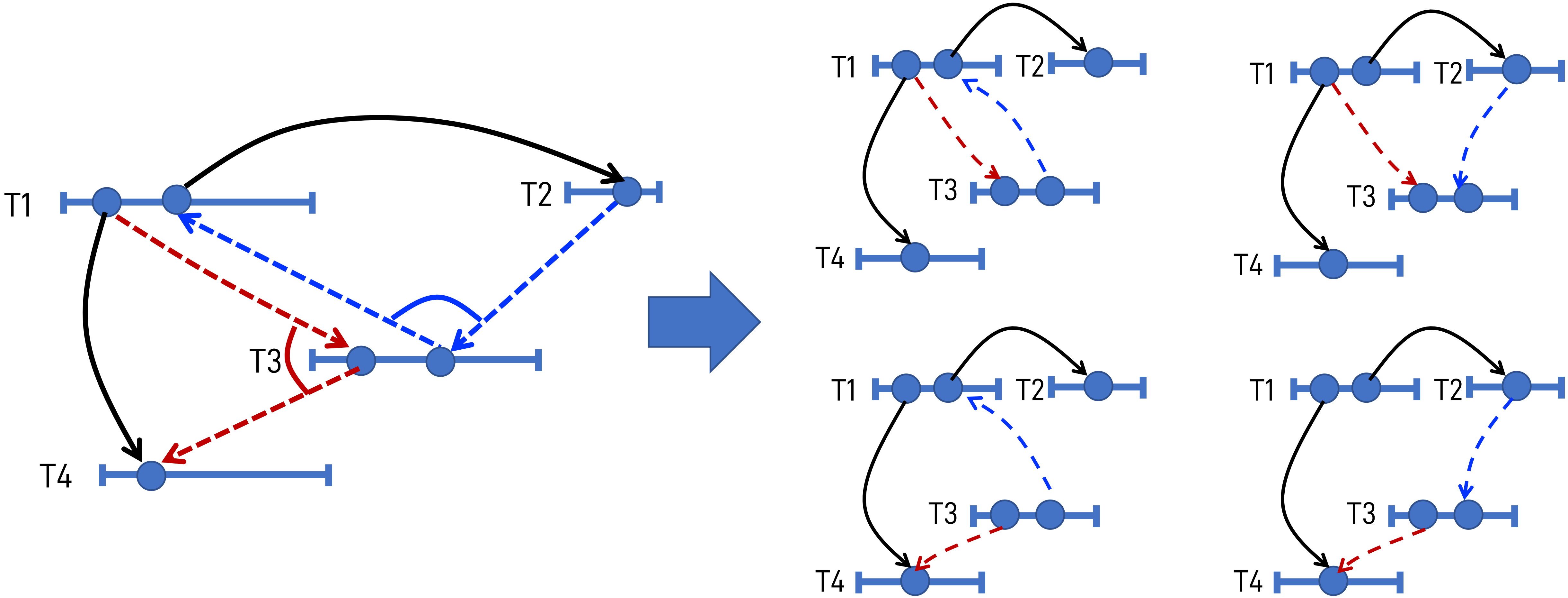
E = { T1 ⟶ T2 }

C = { <T3 ⟶ T1, T2 ⟶ T3> }

Starting point: brute-force search on a polygraph

- [Papadimitriou 79]: build polygraph and search
- Step 1: building a polygraph from a history
- Step 2: searching for an acyclic graph





polygraph (V, E, C)

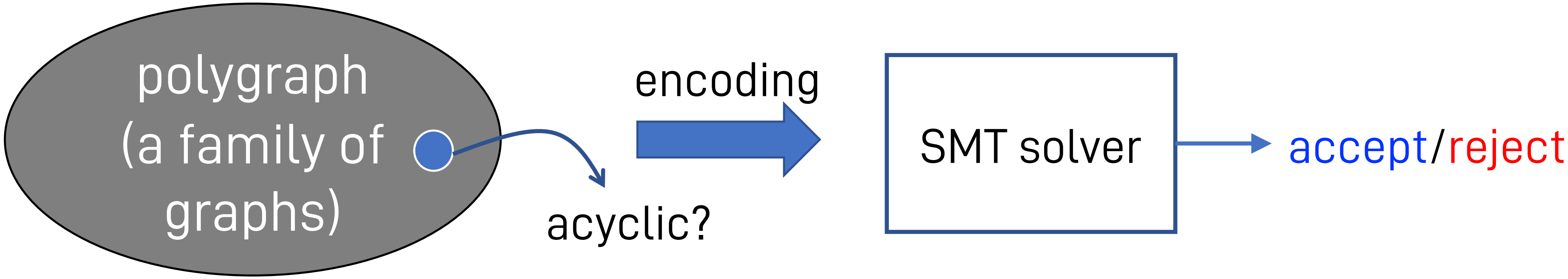
the search space is $2^{|C|}$

Outline

- The underlying problem
- Solution: Cobra

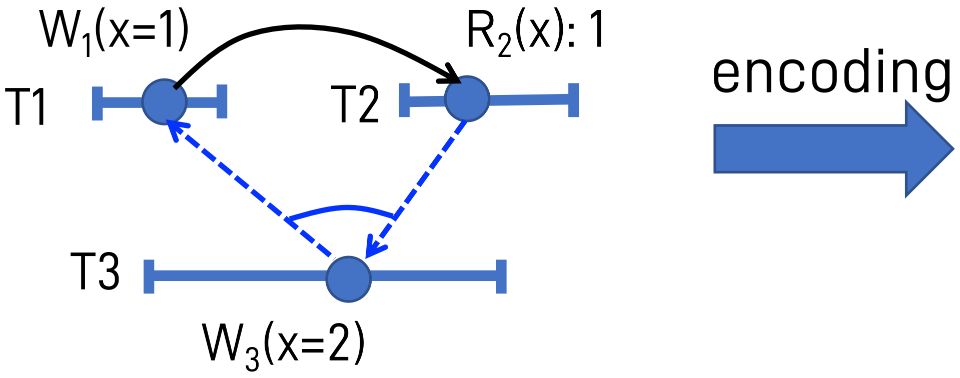
Checking serializability may be tractable in practice

- Intuitions:
  - advances of SAT/SMT solvers
  - heuristically solving many hard problems
- Baseline: encode the problem and use SMT solvers



Cobra aims at real-world workloads

- Intuition:
  - advances of SAT/SMT solvers
  - heuristically solving many hard problems
- Baseline: encode the problem and use SMT solvers



Edges:

$e_{(T1,T2)}$ = True

Constraints:

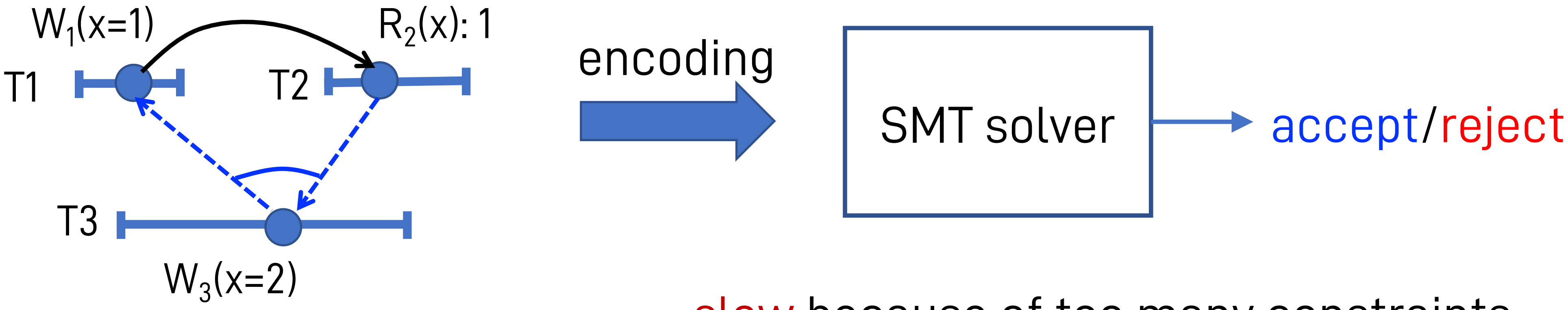
con = ($e_{(T3,T1)}$=True AND $e_{(T2,T3)}$=False) OR
($e_{(T3,T1)}$=False AND $e_{(T2,T3)}$=True)

Acyclicity:

graph with edges=True is acyclic

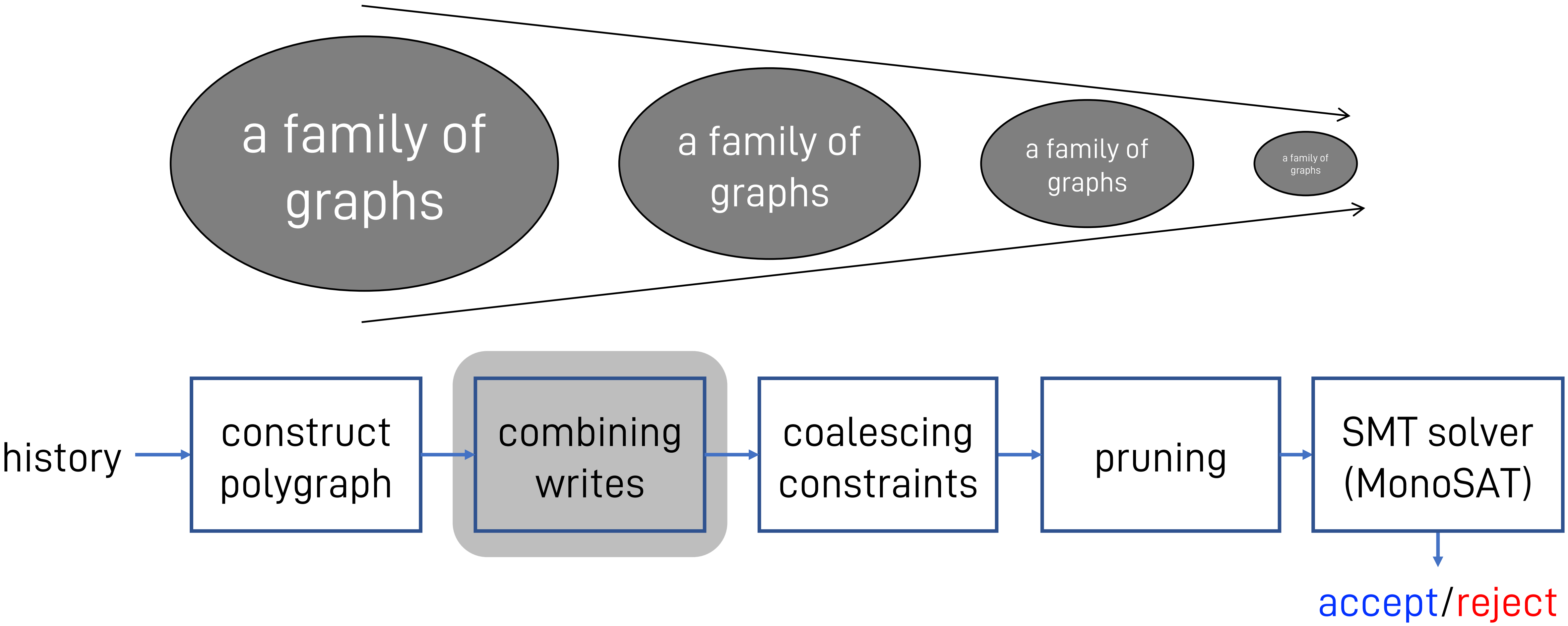
Cobra aims at real-world workloads

- Intuition:
  - advances of SAT/SMT solvers
  - heuristically solving many hard problems
- Baseline: encode the problem and use SMT solvers



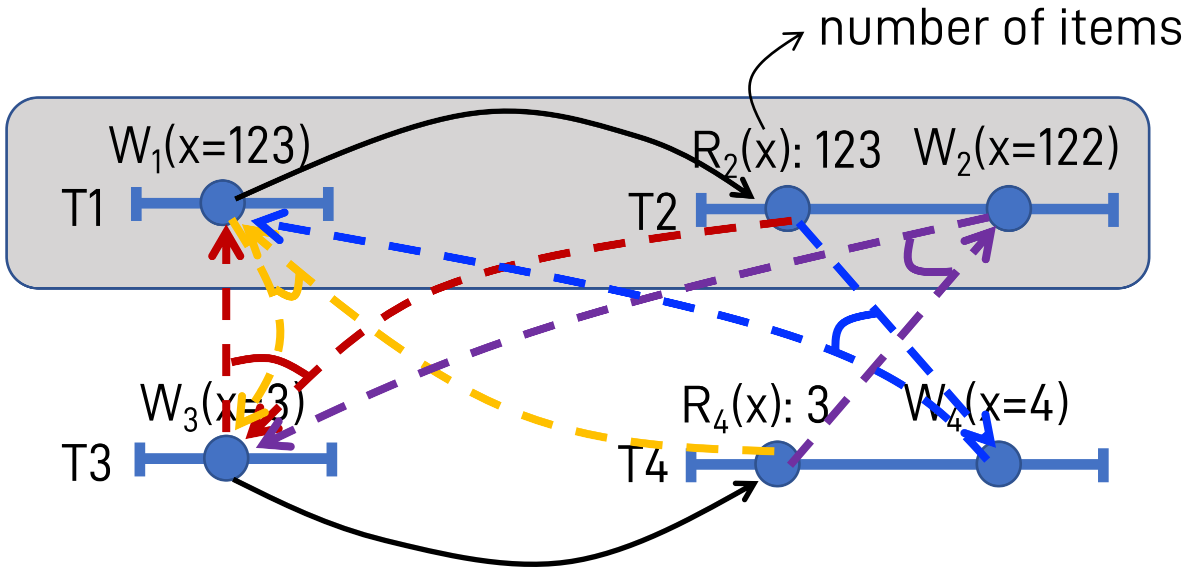
slow because of too many constraints

How to reduce constraints in a polygraph?



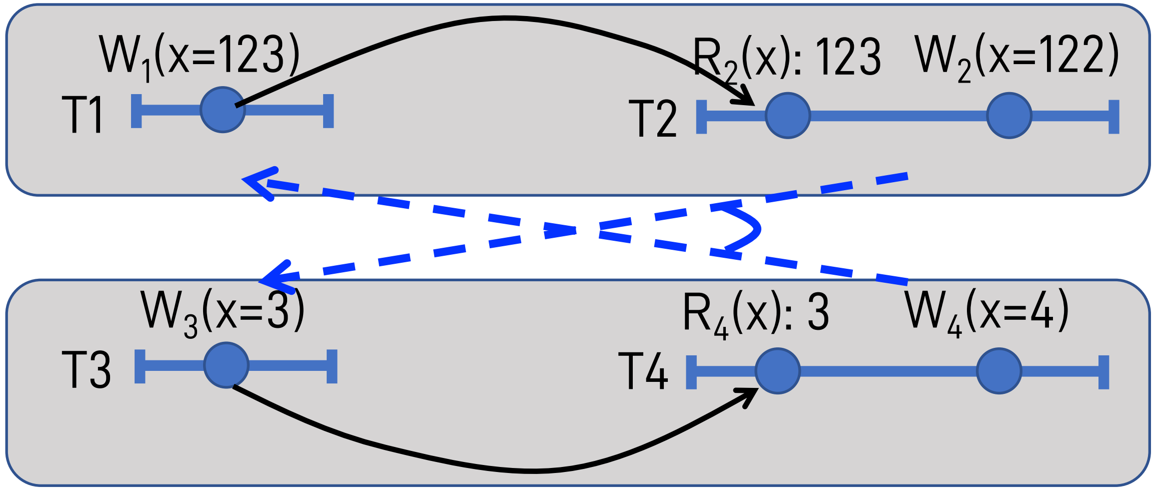
Combining writes exploits a common pattern

- Read-modify-write is a common pattern in practice.



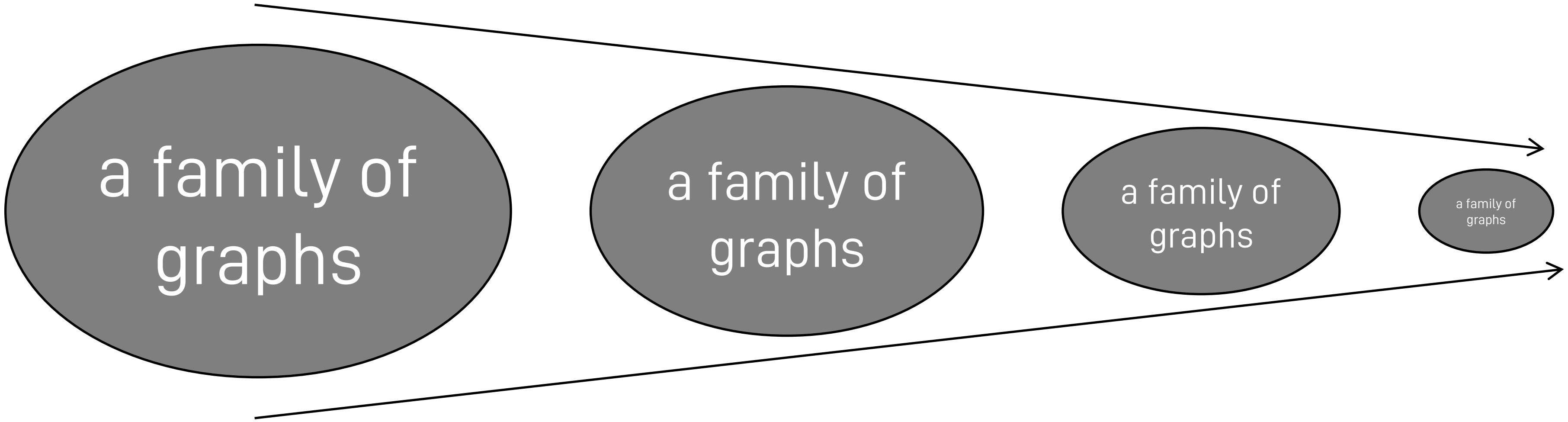
Combining writes: exploit common patterns

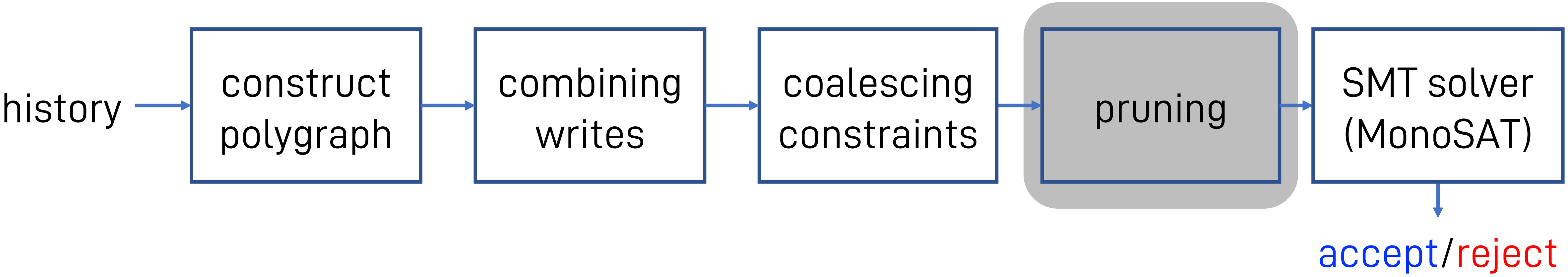
- Read-modify-write is a common pattern in practice.



Cobra produces just one constraint.

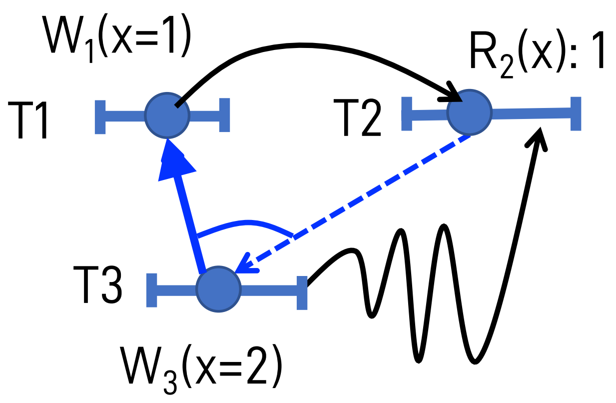
Cobra exploits characteristics of the problem





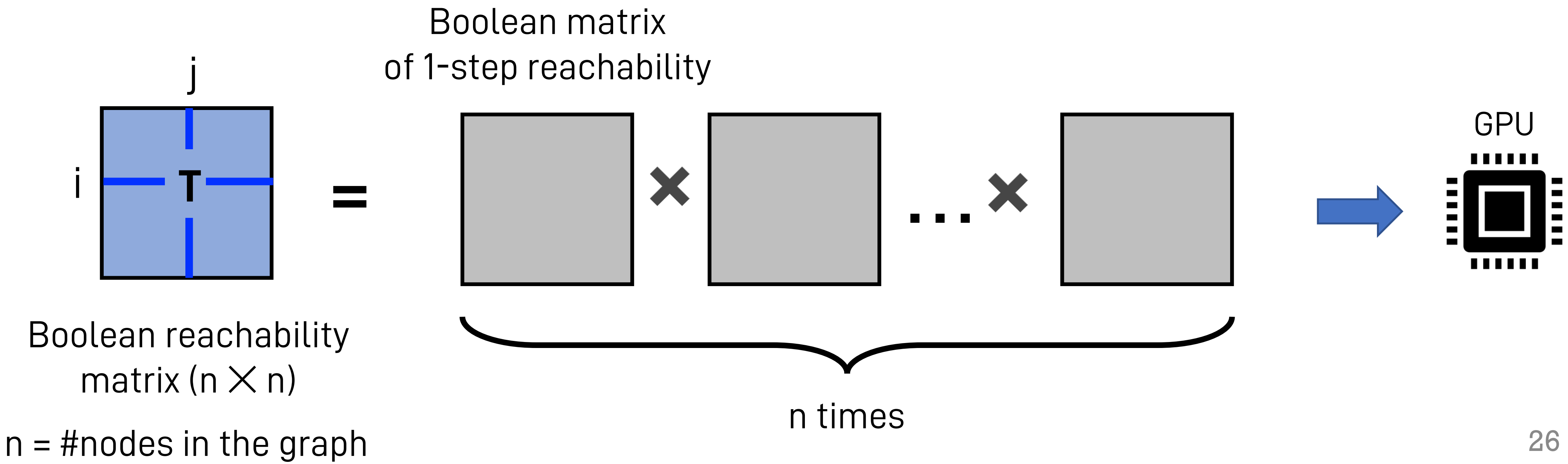
Pruning via graph paths (reachability)

- idea: reduce #constraints by reachability
  1) what can be learned from reachability?
  2) how to get reachability efficiently?

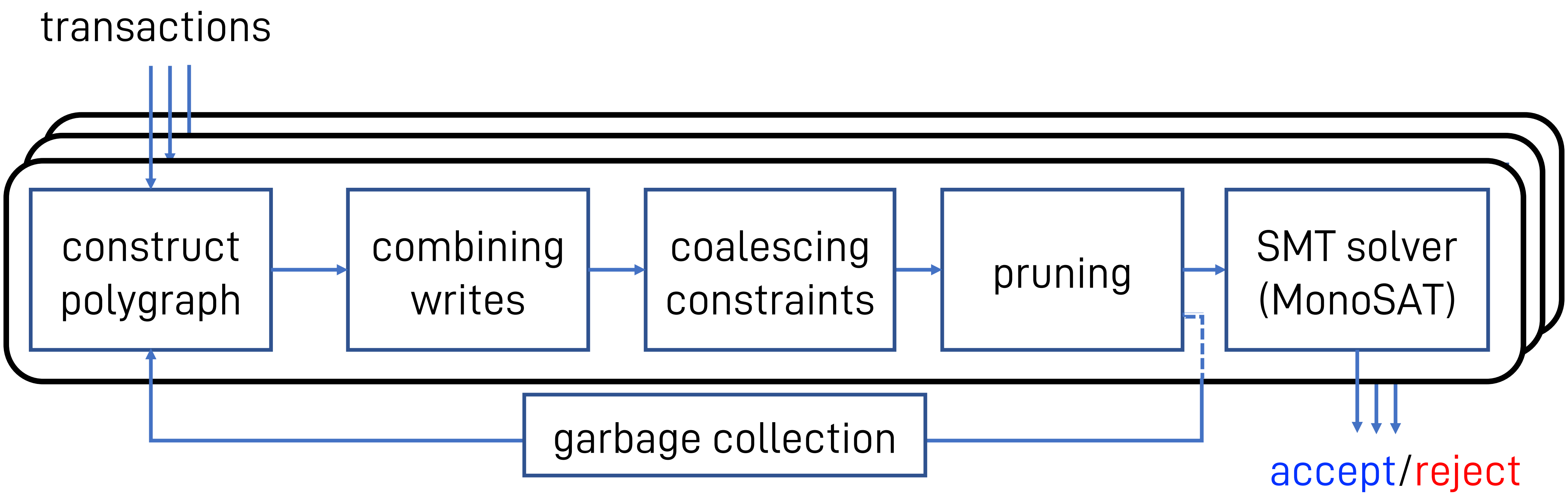


Pruning via graph paths (reachability)

- idea: reduce #constraints by reachability
  1) what can be learned from reachability?
  2) how to get reachability efficiently?



Cobra verifies in rounds to support growing histories



- Cobra needs to delete transactions after each round.

Experimental evaluation

- What are Cobra verifier's costs compared to the baseline (MonoSAT)?

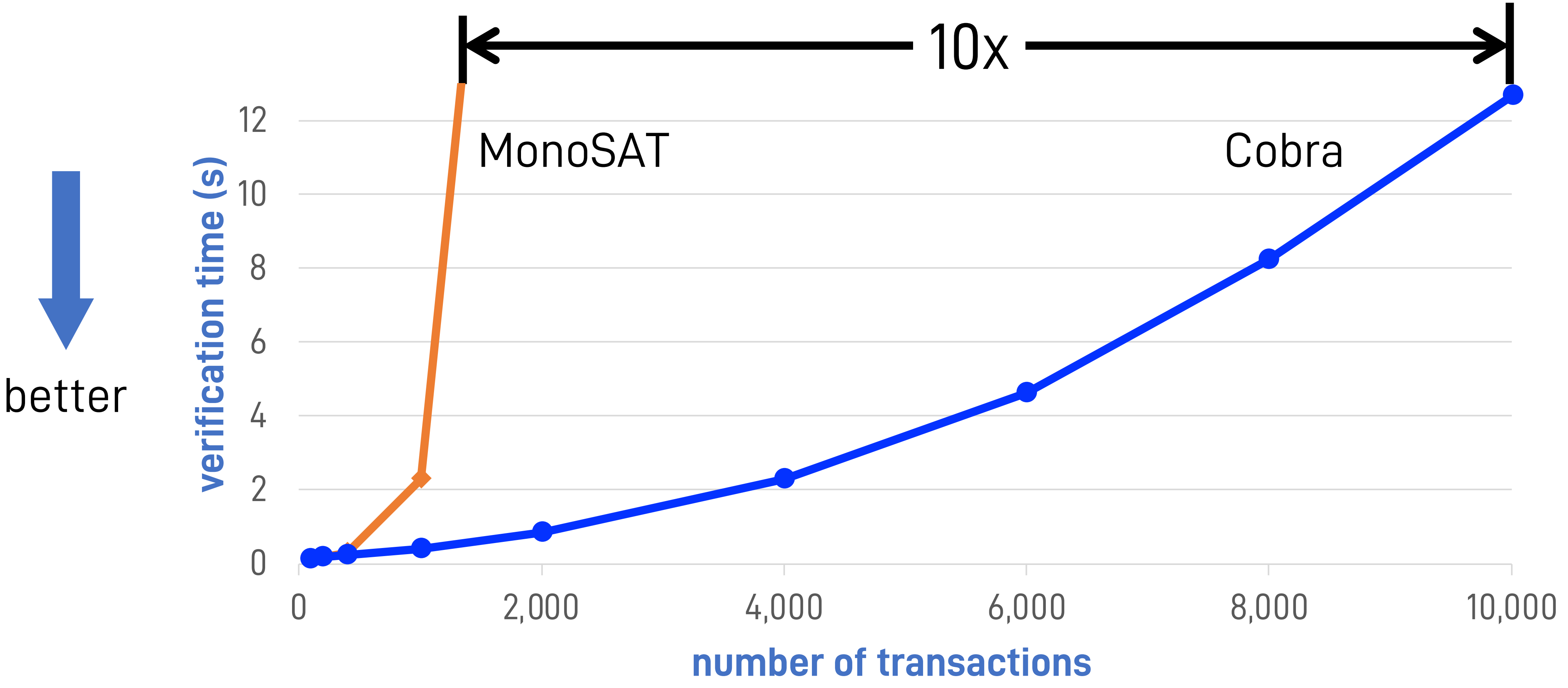
- How much time is spent on each phase of Cobra?

- What is the Cobra's verification throughput?
- How much runtime overhead does Cobra impose for clients?
- What are Cobra's storage and network costs?

Experiment setup

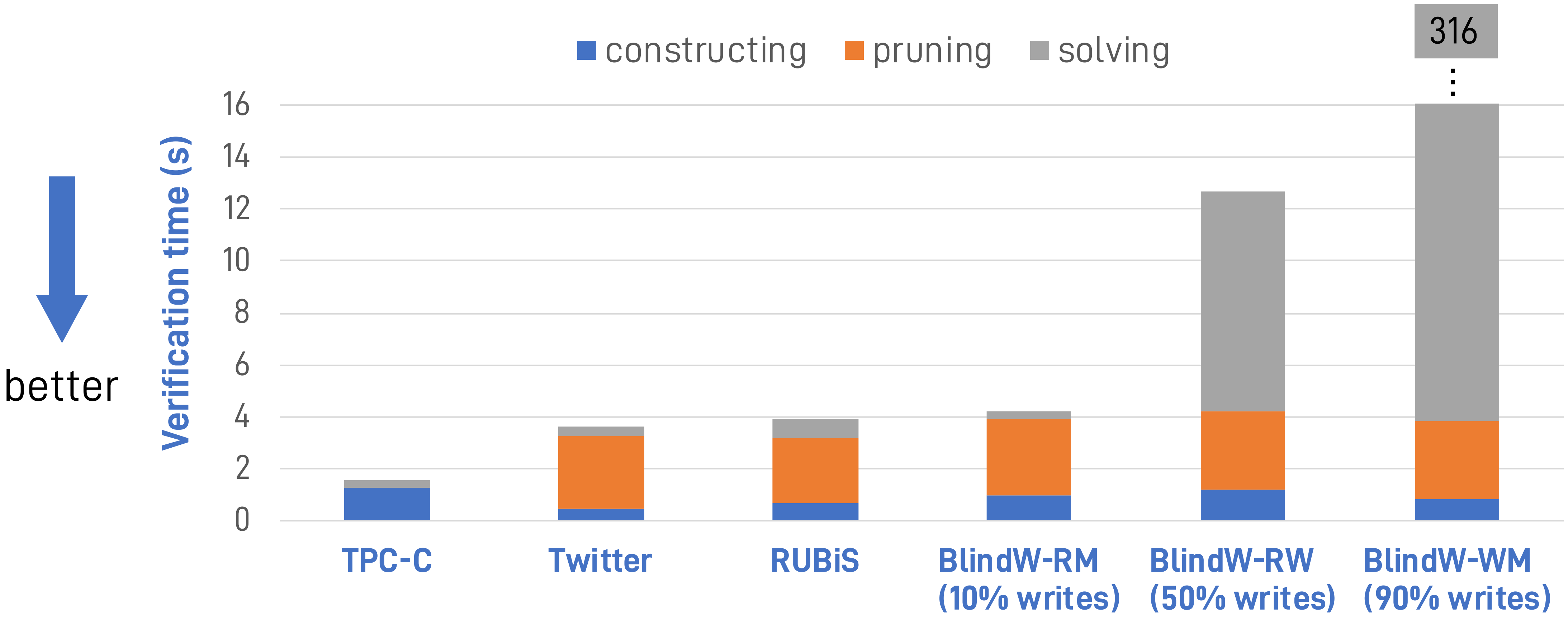
- Benchmarks:
  - TPC-C, Twitter, RUBiS
  - BlindW: RM (90% reads), RW (50% writes), WM (90% writes)
- Databases:
  - RocksDB, PostgreSQL, and Google Cloud Datastore
- Verifier:
  - p3.2xlarge EC2 instance: a V100 GPU, 8-core CPU, 64GB memory

Cobra can handle 10x larger workloads



- BlindW-RW: read-only and write-only transactions (50:50)
- 10k-key DB, 8 operations/txn, 24 concurrent clients

Decomposition of Cobra's verification runtime



- All workloads are with 10k transactions.

Recap

- Cobra verifies …
  - … serializability
  - … of black-box databases
  - … while scaling to real-world workloads.

Related work

a) black-box checking
b) serializability
c) scaling to real-world workloads

- Serializability checker for black-box databases
  - algorithms without SAT/SMT [BE19, SMWG11]
  - Gretchen, using a constraint solver (fzn-gecode)
- Elle, an isolation anomaly checker
- Checking/ensuring storage consistency
- Execution Integrity

Related work

a) black-box testing
b) serializability
~~c)~~ scaling to real-world workloads

- Serializability checker for black-box databases
- Elle, an isolation anomaly checker
  - mode 1: verify serializability by specific APIs and workloads (not black-box)
  - mode 2: testing serializability violations using heuristics (not verification)
- Checking/ensuring storage consistency
- Execution Integrity

Related work

~~a)~~ black-box checking
~~b)~~ serializability
c) scaling to real-world workloads

- Serializability checker for black-box databases
- Elle, an isolation anomaly checker
- Checking/ensuring storage consistency
  - Concerto [AEKKMPR17]
  - requiring extra information from the database [RGAKW12, ZK12]
  - relying on synchronized clocks [LVAHSTKL15]
  - requiring client-to-client communication [SCCKMS10]
- Execution Integrity

Related work

✕ {
a) black-box checking
b) serializability
c) scaling to real-world workloads

- Serializability checker for black-box databases
- Elle, an isolation anomaly checker
- Checking/ensuring storage consistency
  - Concerto [AEKKMPR17]
  - requiring extra information from the database [RGAKW12, ZK12]
  - relying on synchronized clocks [LVAHSTKL15]
  - requiring client-to-client communication [SCCKMS10]
- Execution Integrity

Related work

~~a)~~ black-box checking
b) serializability
c) scaling to real-world workloads

- Serializability checker for black-box databases
- Elle, an isolation anomaly checker
- Checking/ensuring storage consistency
- Execution Integrity
  - replication: PBFT [CL99], Ethereum
  - attestation: SGX-/TPM-based systems
  - probabilistic proofs: Pepper [SMBW12], Pinocchio [PGHR13], Pantry [BJRSBW13]
  - others: Ripley [VPL09], AVM [HARD10], Verena [KFPC16], Orochi [TYLW17]

Summary

- Cobra verifies serializability of a black-box database

  ... for real-world workloads.

- Users of cloud databases used to have to assume serializability;

  but now, with Cobra, they can be sure.

- Code is released at:

  https://github.com/DBCobra/CobraHome