数据库系统事务一致性验证问题研究

(工业软件产教联合主题交流会)

魏恒峰

hfwei@nju.edu.cn

2024 年 11 月 11 日

数据副本带来了**数据一致性**问题

(使用**形式化方法**) 解决**分布式系统**中的**数据一致性**问题


Microsoft
SQL Server

PostgreSQL

yugabyteDB

SQLite

ORACLE®

MEM
GRAPH

TiDB

mongoDB.

Dgraph

TDSQL

PolarDB-X
内核新版本

fauna

它们正确吗?

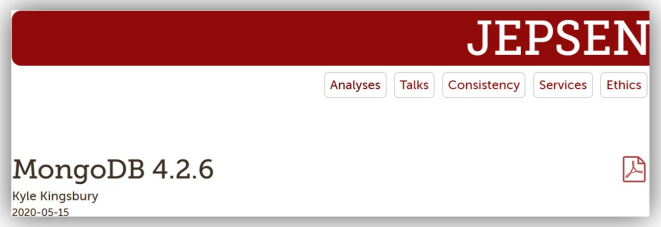
- **数据库系统**实现正确了吗?
- **客户端程序**编写正确了吗?

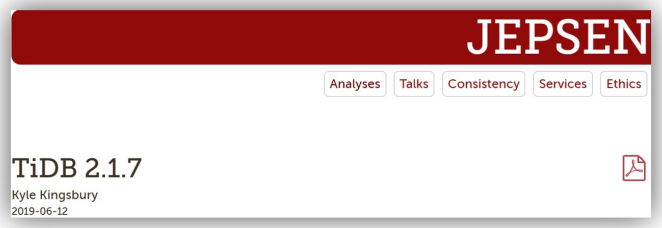
- **数据库系统**实现正确了吗?

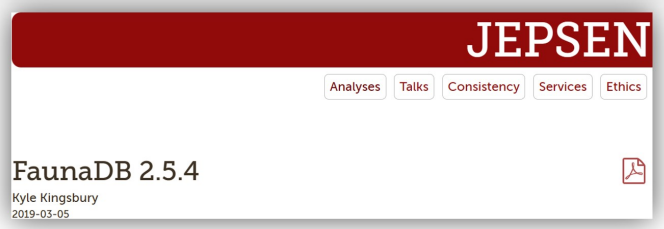
数据库系统通过事务与**隔离级别** (Isolation Levels) 保障数据的正确性

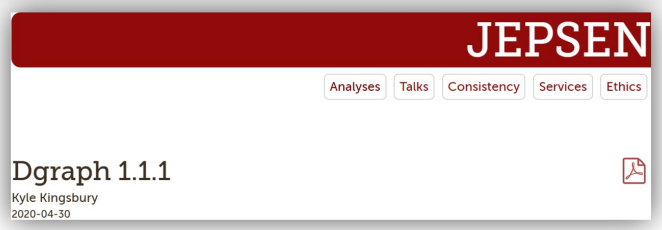
声称实现了**快照隔离** (Snapshot Isolation; SI)

但是, 仍存在违反**快照隔离**的**数据异常** (Data Anomalies) 情况

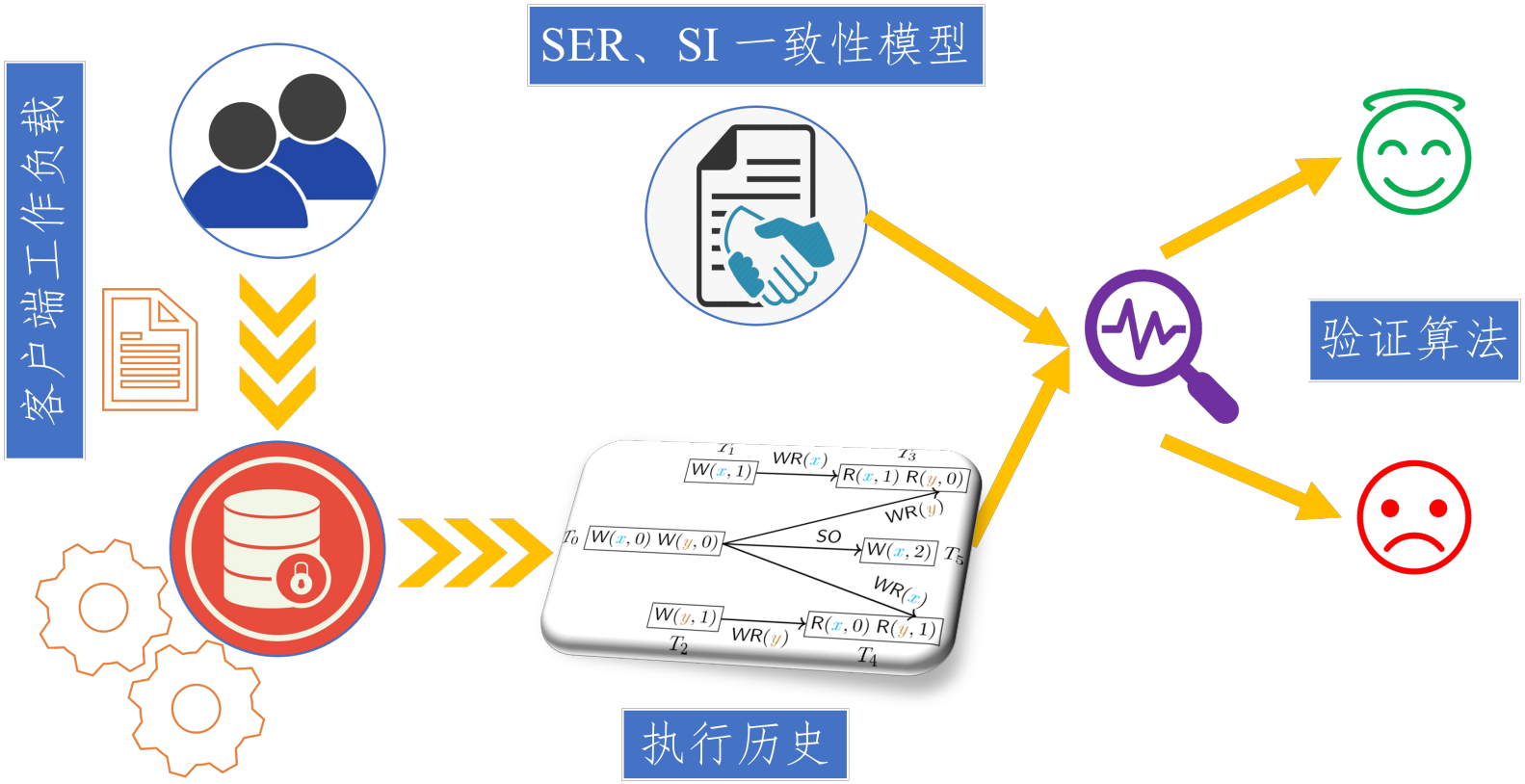


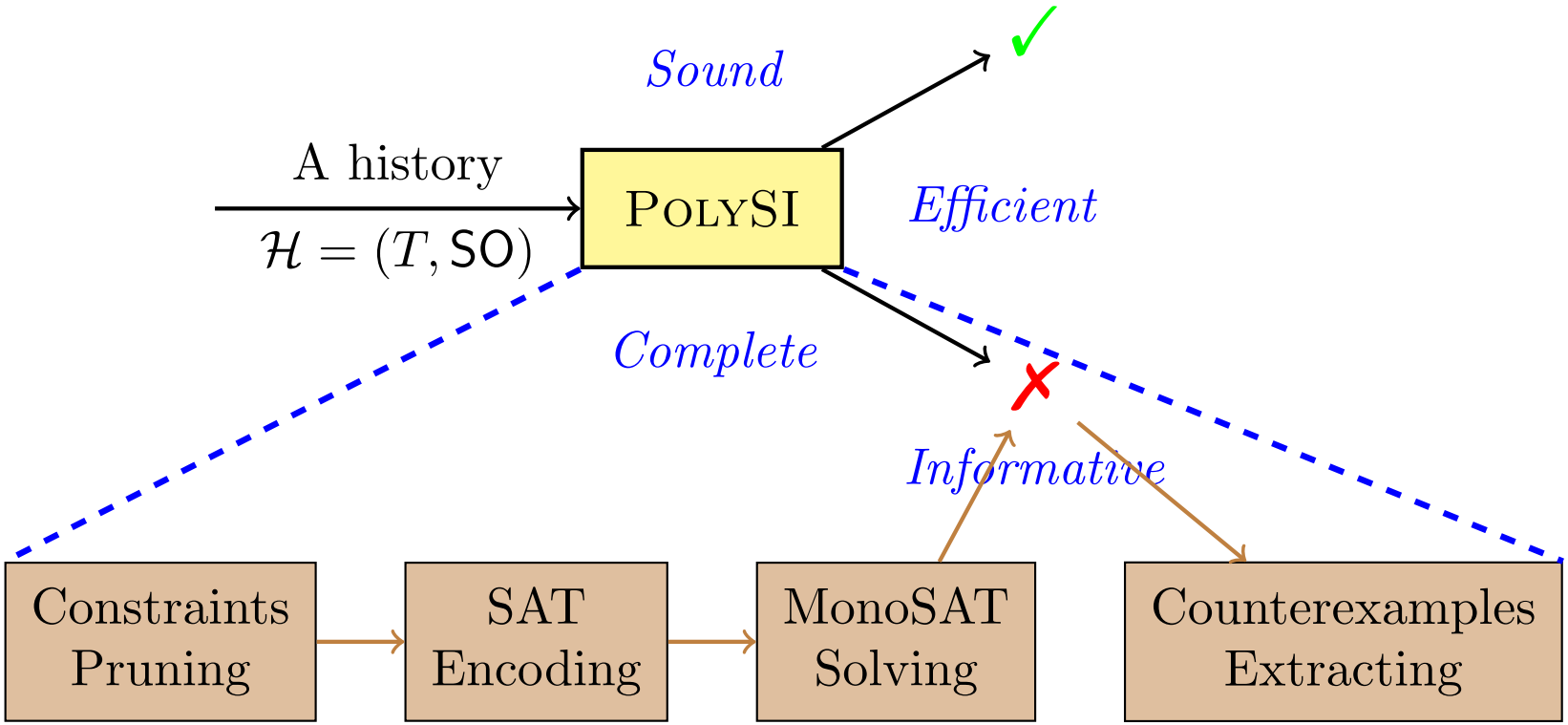




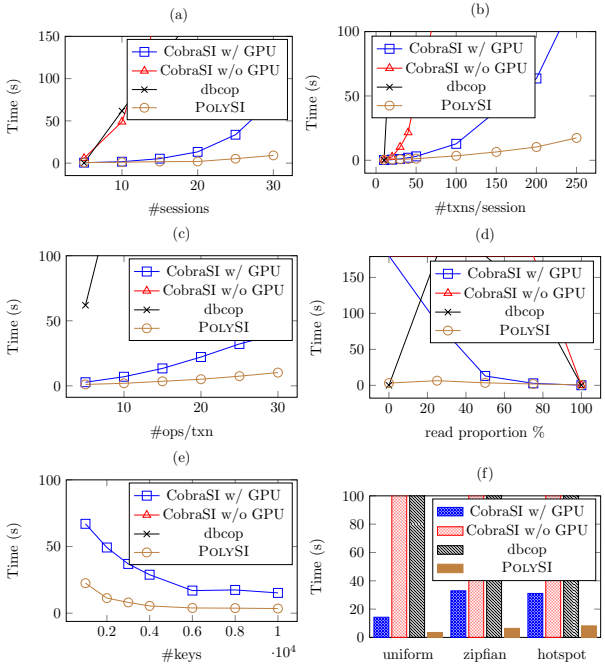


黑盒测试：数据库系统**执行历史**验证问题

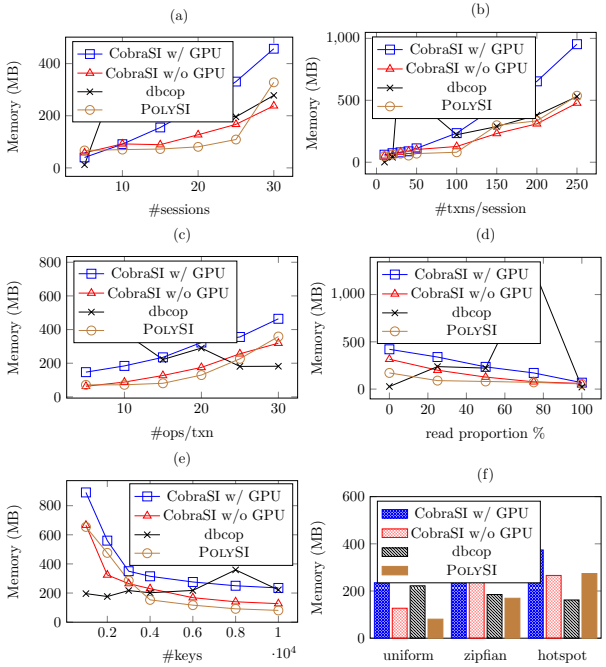




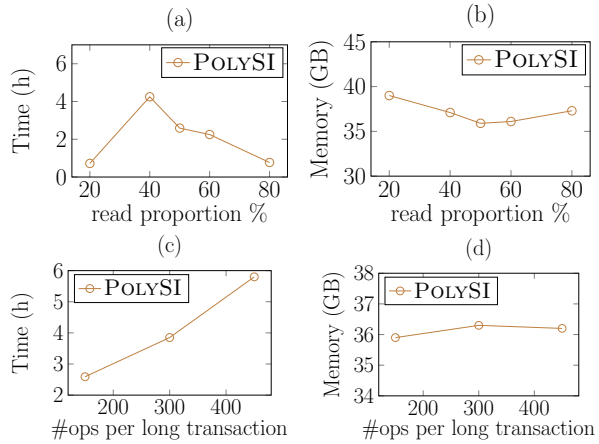
[PolySI@VLDB'2023]


(a)
(b)
(c)
(d)
(e)
(f)
CobraSI w/ GPU
CobraSI w/o GPU
dbcop
PolySI
Time (s)
#sessions
#txns/session
#ops/txn
read proportion %
#keys
uniform
zipfian
hotspot

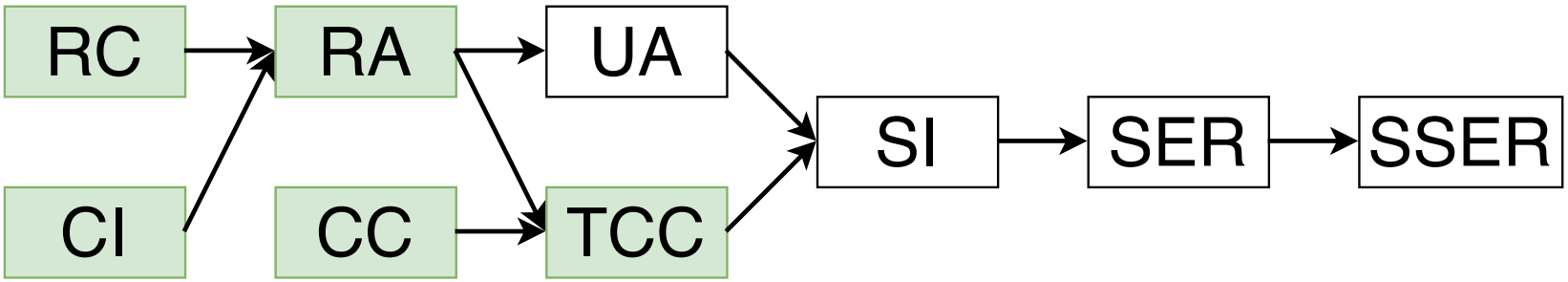
时间


(a)
(b)
(c)
(d)
(e)
(f)
CobraSI w/ GPU
CobraSI w/o GPU
dbcop
PolySI
Memory (MB)
#sessions
#txns/session
#ops/txn
read proportion %
#keys
uniform
zipfian
hotspot

内存


(a)
(b)
(c)
(d)
PolySI
Time (h)
Memory (GB)
read proportion %
#ops per long transaction

可扩展性



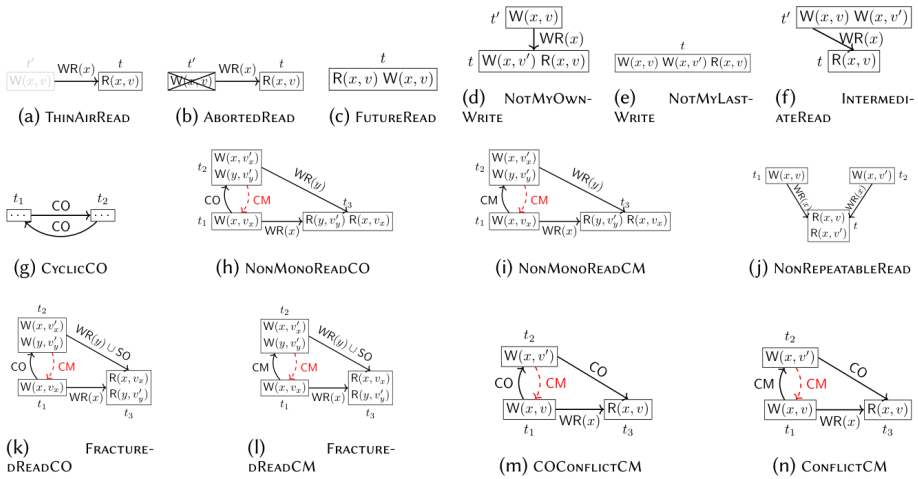
[Plume@OOPSLA'2024]

如何利用数据异常全面、准确地定义这些事务隔离级别

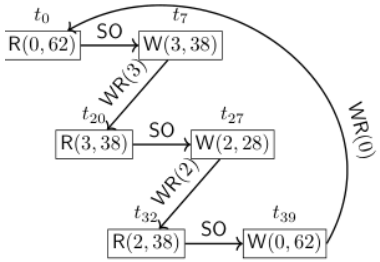
Isolation Level	Prohibited TAPs
Transactional Causal Consistency	All 14 TAPs
Read Atomicity	TAP-a to TAP-l (12 TAPs)
Read Committed	TAP-a to TAP-i (9 TAPs)
Cut Isolation	TAP-j

Table 1. The description and formalization of 14 TAPs (also visualized in Figure 3).

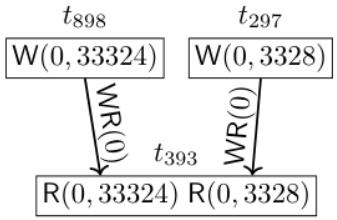
TAP	Description	TAP	Description
(a)	A transaction reads a value out of thin air. $\exists r \in R(T).\ \forall w \in W(T \cup T_{\otimes}).\ \neg(w \xrightarrow{\text{wr}} r)$.	(b)	A transaction reads a value written by an aborted transaction. $\exists r \in R(T).\ \exists w \in W(T_{\otimes}).\ w \xrightarrow{\text{wr}} r$.
(c)	A transaction reads from a future write within the same transaction. $\exists t \in T.\ \exists w \in W(t).\ \exists r \in R(t).$ $(w \xrightarrow{\text{wr}} r \wedge r \xrightarrow{\text{po}_t} w)$.	(d)	Transaction t reads key x from transaction $t' \neq t$, but t has written to x before this read. $\exists x \in \mathrm{K}.\ \exists t, t' \neq t \in W_x(t).\ \exists r \in R_x(t).\ \exists w \in W_x(t).$ $\exists w' \in W_x(t').\ (w' \xrightarrow{\text{wr}(x)} r \wedge w \xrightarrow{\text{po}_t} r)$.
(e)	Transaction t reads key x from a write w in itself, but w is not the last write on x in t before this read. $\exists x \in \mathrm{K}.\ \exists t \in T.\ \exists w, w' \neq w \in W_x(t).\ \exists r \in R_x(t).$ $(w \xrightarrow{\text{po}_t} w' \xrightarrow{\text{po}_t} r \wedge w \xrightarrow{\text{wr}(x)} r)$	(f)	Transaction t reads key x from a write w in transaction $t' \neq t$ which writes x more than once, but w is not the last write on x in t'. $\exists x \in \mathrm{K}.\ \exists t \in \mathrm{RT}_x.\ \exists t' \neq t \in \mathrm{WT}_x.\ \exists r \in R_x(t).$ $\exists w, w' \neq w \in W_x(t').\ (w \xrightarrow{\text{wr}(x)} r \wedge w \xrightarrow{\text{po}_{t'}} w')$.
(g)	The relation $\mathrm{SO} \cup \mathrm{WR}$ is cyclic. $(\mathrm{SO} \cup \mathrm{WR})^+ \cap I_T \neq \emptyset$.	(h)	Transaction t_3 reads y from t_2 and then reads $x \neq y$ from t_1. Transaction t_2 also writes to x but $t_1 \xrightarrow{\text{CO}} t_2$. $\exists x, y \neq x \in \mathrm{K}.\ \exists t_1, t_2 \neq t_1 \in \mathrm{WT}_x.\ \exists t_3 \in (\mathrm{RT}_x \cap \mathrm{RT}_y) \setminus \{t_1, t_2\}.$ $\exists w_x \in W_x(t_1).\ \exists w_y \in W_y(t_2).\ \exists r_x \in R_x(t_3).\ \exists r_y \in R_y(t_3).$ $(w_x \xrightarrow{\text{wr}(x)} r_x \wedge w_y \xrightarrow{\text{wr}(y)} r_y \wedge r_y \xrightarrow{\text{po}_{t_3}} r_x \wedge t_1 \xrightarrow{\text{CO}} t_2)$.
(i)	Transaction t_3 reads y from t_2 and then reads $x \neq y$ from t_1. Transaction t_2 also writes to x but $t_1 \xrightarrow{\text{CM}} t_2$. This is a general case of (h). $\exists x, y \neq x \in \mathrm{K}.\ \exists t_1, t_2 \neq t_1 \in \mathrm{WT}_x.\ \exists t_3 \in (\mathrm{RT}_x \cap \mathrm{RT}_y) \setminus \{t_1, t_2\}.$ $\exists w_x \in W_x(t_1).\ \exists w_y \in W_y(t_2).\ \exists r_x \in R_x(t_3).\ \exists r_y \in R_y(t_3).$ $(w_x \xrightarrow{\text{wr}(x)} r_x \wedge w_y \xrightarrow{\text{wr}(y)} r_y \wedge r_y \xrightarrow{\text{po}_{t_3}} r_x \wedge t_1 \xrightarrow{\text{CM}} t_2)$.	(j)	A transaction reads from a key from other transactions more than once, but with different values. $\exists x \in \mathrm{K}.\ \exists v, v' \neq v \in \mathrm{V}.\ \exists t \in \mathrm{RT}_x.\ \exists t_1 \neq t, t_2 \neq t \in \mathrm{WT}_x.$ $\exists r_1 \triangleq \mathrm{R}(x, v), r_2 \triangleq \mathrm{R}(x, v') \in R_x(t).$ $\exists w_1 \in W_x(t_1).\ \exists w_2 \in W_x(t_2).$ $(t_1 \neq t_2 \wedge w_1 \xrightarrow{\text{wr}(x)} r_1 \wedge w_2 \xrightarrow{\text{wr}(x)} r_2)$.
(k)	Transaction t_3 reads x from t_1 and $y \neq x$ from t_2. Transaction t_2 also writes to x but $t_1 \xrightarrow{\text{CO}} t_2$. $\exists x, y \neq x \in \mathrm{K}.\ \exists t_1, t_2 \neq t_1 \in \mathrm{WT}_x.\ \exists t_3 \in (\mathrm{RT}_x \cap \mathrm{RT}_y) \setminus \{t_1, t_2\}.$ $\exists w_x \in W_x(t_1).\ \exists w_y \in W_y(t_2).\ \exists r_x \in R_x(t_3).\ \exists r_y \in R_y(t_3).$ $(w_x \xrightarrow{\text{wr}(x)} r_x \wedge w_y \xrightarrow{\text{wr}(y)} r_y \wedge t_1 \xrightarrow{\text{CO}} t_2)$.	(l)	Transaction t_3 reads x from t_1 and $y \neq x$ from t_2. Transaction t_2 also writes to x but $t_1 \xrightarrow{\text{CM}} t_2$. This is a general case of (i) and (k). $\exists x, y \neq x \in \mathrm{K}.\ \exists t_1, t_2 \neq t_1 \in \mathrm{WT}_x.\ \exists t_3 \in (\mathrm{RT}_x \cap \mathrm{RT}_y) \setminus \{t_1, t_2\}.$ $\exists w_x \in W_x(t_1).\ \exists w_y \in W_y(t_2).\ \exists r_x \in R_x(t_3).\ \exists r_y \in R_y(t_3).$ $(w_x \xrightarrow{\text{wr}(x)} r_x \wedge w_y \xrightarrow{\text{wr}(y)} r_y \wedge t_1 \xrightarrow{\text{CM}} t_2)$.
(m)	Transaction t_3 reads x from t_1. There is a transaction t_2 that also writes to x such that $t_1 \xrightarrow{\text{CO}} t_2 \xrightarrow{\text{CO}} t_3$. $\exists x \in \mathrm{K}.\ \exists t_1, t_2 \neq t_1 \in \mathrm{WT}_x.\ \exists t_3 \in \mathrm{RT}_x \setminus \{t_1, t_2\}.$ $(t_1 \xrightarrow{\text{WR}(x)} t_3 \wedge t_1 \xrightarrow{\text{CO}} t_2 \xrightarrow{\text{CO}} t_3)$.	(n)	Transaction t_3 reads x from t_1. There is a transaction t_2 that also writes to x such that $t_1 \xrightarrow{\text{CM}} t_2 \xrightarrow{\text{CO}} t_3$. This is a general case of (l) and (m). $\exists x \in \mathrm{K}.\ \exists t_1, t_2 \neq t_1 \in \mathrm{WT}_x.\ \exists t_3 \in \mathrm{RT}_x \setminus \{t_1, t_2\}.$ $(t_1 \xrightarrow{\text{WR}(x)} t_3 \wedge t_1 \xrightarrow{\text{CM}} t_2 \xrightarrow{\text{CO}} t_3)$.



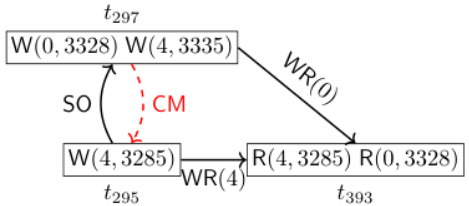
Checker	TAP-a	TAP-b	TAP-c	TAP-d	TAP-e	TAP-f	TAP-g	TAP-h	TAP-i	TAP-j	TAP-k	TAP-l	TAP-m	TAP-n
PolySI (SI)	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Viper (SI)	○	○	◐	◐	◐	○	◐	●	●	◐	●	●	●	●
Cobra (SER)	●	●	●	○	○	○	●	●	●	○	●	●	●	●
dbcop-TCC	●	●	○	●	●	●	●	●	●	●	●	●	●	●
dbcop-RA	●	●	○	●	●	●	●	●	●	●	●	●	–	–
dbcop-RC	●	●	○	●	●	●	○	○	○	–	–	–	–	–
Elle-TCC	●	●	◐	●	●	◐	○	○	◐	●	○	●	○	●
Elle-RA	●	●	◐	●	●	◐	○	○	◐	●	○	●	–	–
Elle-RC*	●	●	◐	○	○	●	○	–	–	–	–	–	–	–
Plume	●	●	●	●	●	●	●	●	●	●	●	●	●	●



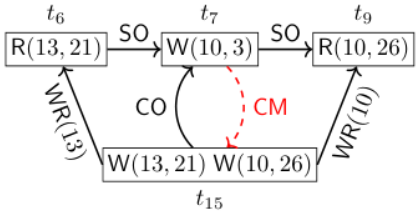
(a) TAP-g in AntidoteDB



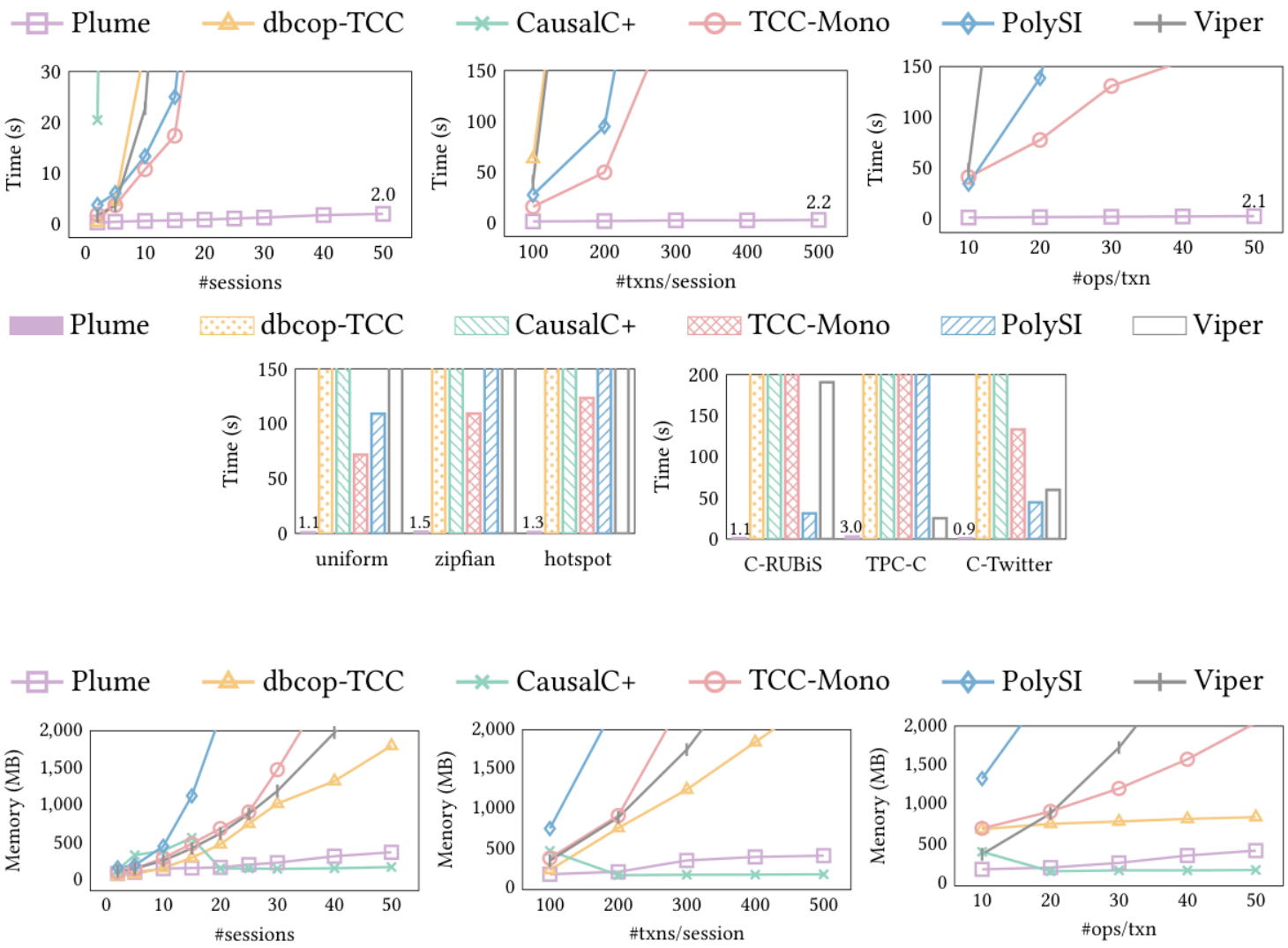
(b) TAP-j in MariaDB

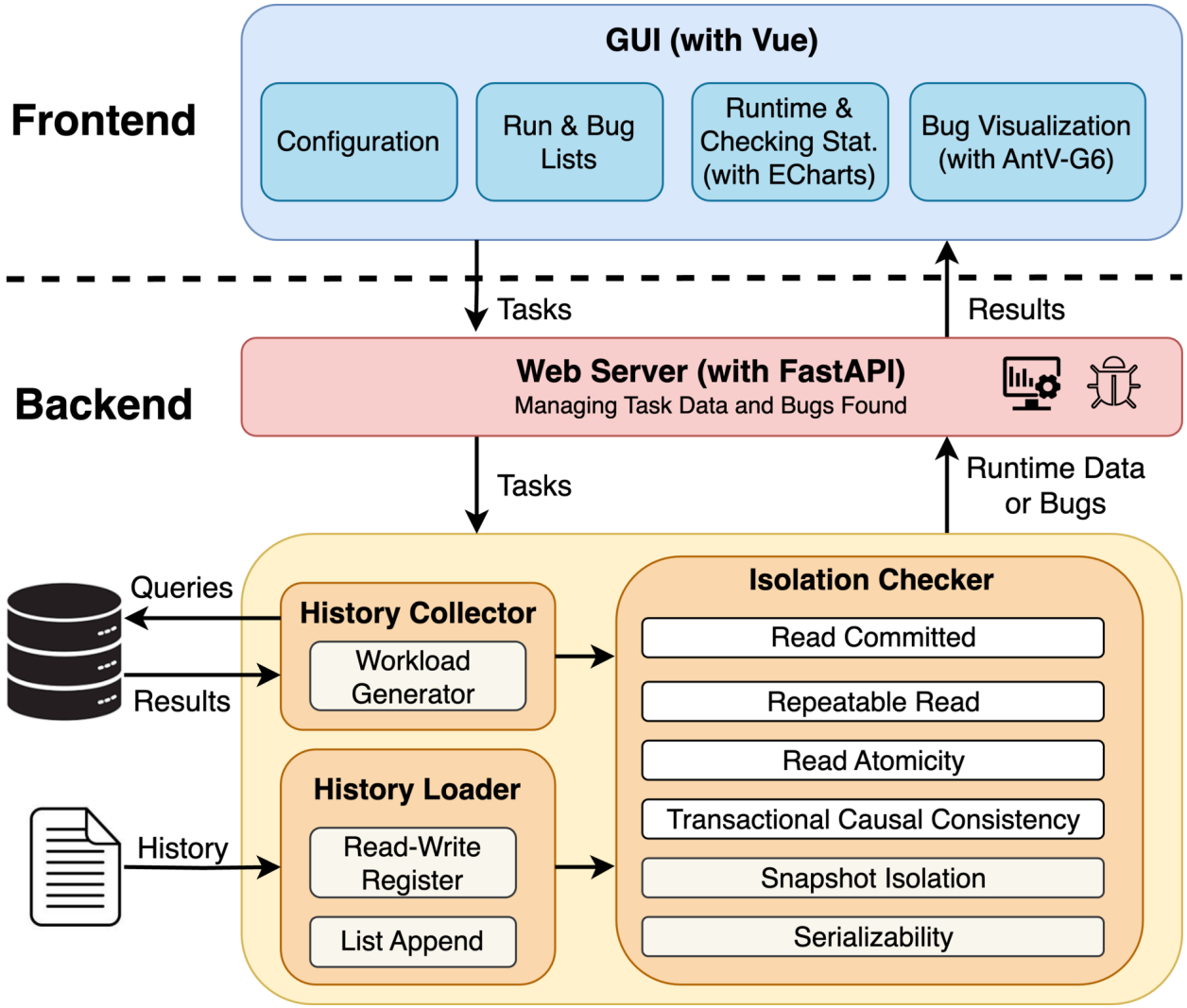


(c) TAP-k in MariaDB

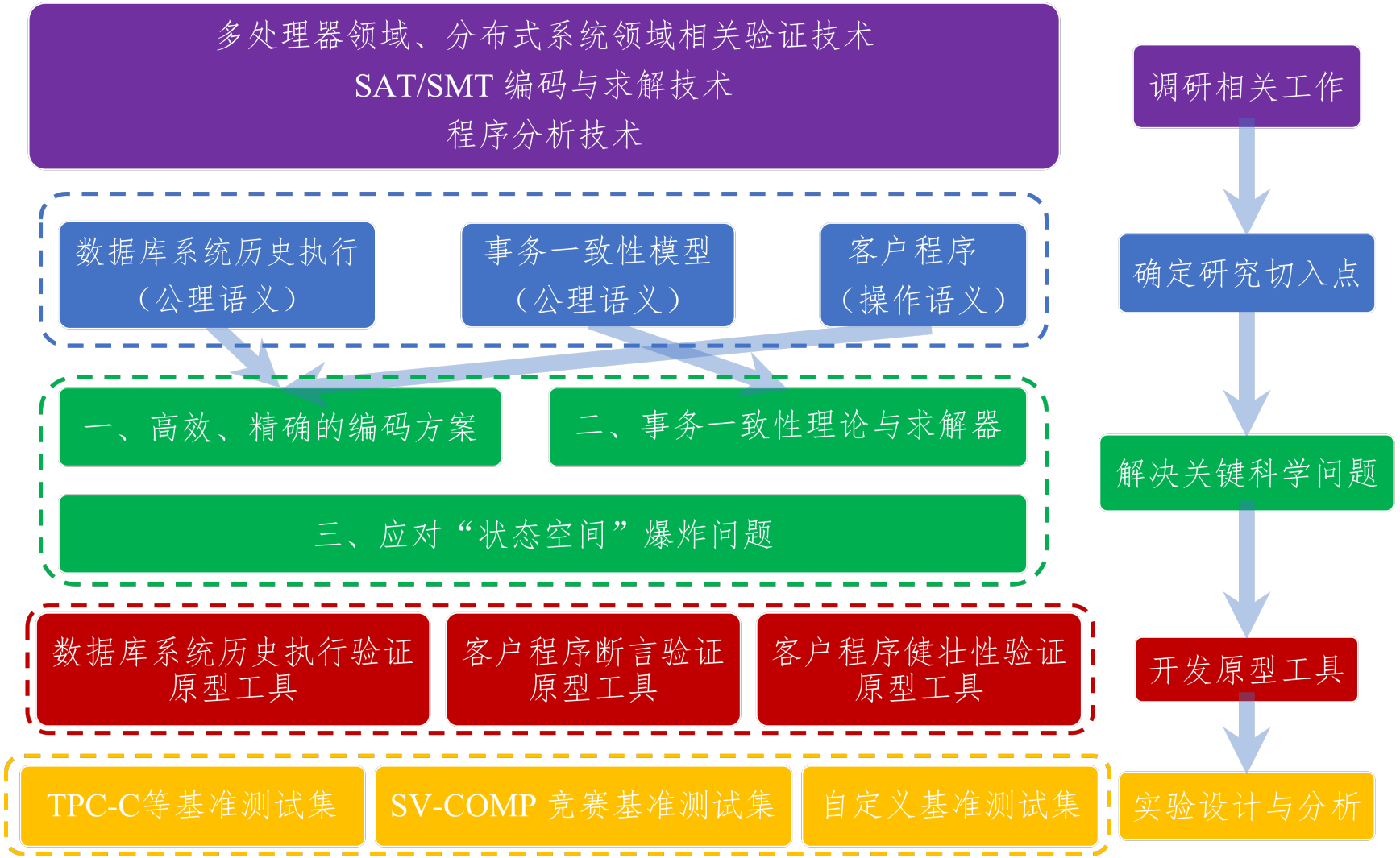


(d) TAP-m in YugabyteDB


Plume
dbcop-TCC
CausalC+
TCC-Mono
PolySI
Viper
Time (s)
#sessions
#txns/session
#ops/txn
2.0
2.2
2.1
uniform
zipfian
hotspot
C-RUBiS
TPC-C
C-Twitter
1.1
1.5
1.3
1.1
3.0
0.9
Memory (MB)
Menory (MB)



[IsoVista@VLDB'2024 (Demo)]


多处理器领域、分布式系统领域相关验证技术
SAT/SMT 编码与求解技术
程序分析技术
数据库系统历史执行（公理语义）
事务一致性模型（公理语义）
客户程序（操作语义）
一、高效、精确的编码方案
二、事务一致性理论与求解器
三、应对“状态空间”爆炸问题
数据库系统历史执行验证原型工具
客户程序断言验证原型工具
客户程序健壮性验证原型工具
TPC-C等基准测试集
SV-COMP 竞赛基准测试集
自定义基准测试集
调研相关工作
确定研究切入点
解决关键科学问题
开发原型工具
实验设计与分析

混合（Mixing）隔离级别

谓词 (Predicates)

Hengfeng Wei (hfwei@nju.edu.cn)