## **论文介绍**

**标题：**Boosting End-to-End Database Isolation Checking via Mini-Transactions

**作者：** 魏恒峰，肖江，杨娜，刘思，殷梓敬，陈育兴

**关键词：**事务隔离级别，严格可串行化，可串行化，快照隔离，隔离性检测

**论文链接：**<https://arxiv.org/abs/2504.02344>

## **摘要：**

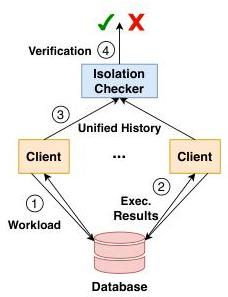
事务隔离性保证对数据库正确性至关重要。然而，近期研究发现，生产环境中的数据库中存在大量隔离性漏洞。常见的隔离性检测方法采用黑盒方式，通过大规模、并发、随机的事务负载对数据库施压，并验证生成的执行历史是否满足指定的隔离级别。对于严格可串行化、可串行化和快照隔离等强隔离级别，这种方法在历史生成和验证过程中常常带来显著的端到端开销。

本文通过提出**微事务（Mini-Transactions, MTs）**的全新设计，来应对上述低效问题。MTs 是紧凑且执行迅速的短事务，能够减少数据库阻塞和事务重试，从而在历史生成过程中显著降低开销。基于 MTs 的读-修改-写模式，我们进一步开发了高效的隔离级别验证算法，可以在线性或平方时间内完成验证。尽管设计简单，MTs 在语义上却十分丰富，能够有效捕捉文献中常见的隔离性异常。

我们在工具 **MTC** 中实现了上述验证算法及 MT 工作负载生成器。实验结果表明，MTC 在历史生成和验证性能上均优于现有最先进的工具。同时，MTC 能够在各类隔离级别下发现生产数据库中的漏洞，并在保持一般随机负载测试有效性的同时，大幅降低检测成本，成为一种高效的黑盒隔离性检测解决方案。

## **1、引言**

数据库系统是现代应用程序的核心支撑，确保其正确性对于数据完整性至关重要。其中，隔离性（Isolation）作为数据库系统最重要的正确性属性之一，指的是在并发事务执行过程中避免相互干扰的能力。尽管隔离性极为重要，但对其进行测试与验证仍然面临诸多挑战，尤其是在生产环境中，数据库内部机制往往被视为黑盒，进一步加大了检测难度。



***图 1：端到端隔离性检测流程：***

*（1）客户端在数据库上执行负载；（2）收集执行结果；（3）构建统一的执行历史；（4）隔离性检测器根据隔离性规范对历史进行验证。*

近期研究表明，生产环境中的数据库系统中存在大量隔离性漏洞，这凸显了高效且有效的隔离性测试方法的必要性。现有的黑盒隔离性检测工具（如 Cobra、PolySI、Elle 和 Porcupine）虽然在发现隔离性违规方面表现出色，但在历史生成与验证过程中普遍存在显著的性能开销问题。

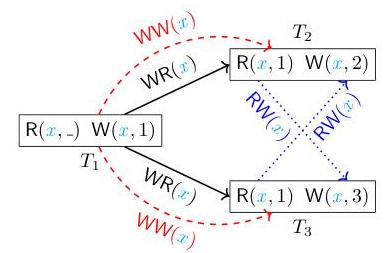
为了解决这些问题，本文提出了一种基于**微事务（Mini-Transactions, MTs）**的全新数据库隔离性检测方法。该方法由南京大学、苏黎世联邦理工学院（ETH Zurich）及腾讯公司联合提出，能够在提升检测效率的同时，保持甚至增强隔离性漏洞的检测能力。

## **2、理解数据库隔离性**

数据库隔离性保证了并发事务不会以导致数据不一致的方式互相干扰。不同的隔离级别提供了不同程度的并发行为保证：

1. **严格可串行化（SSER）**：最强的隔离级别，确保事务的执行顺序既是串行的，又符合实际时间上的顺序。
2. **可串行化（SER）**：保证事务以某种串行顺序执行，但不一定符合实时顺序。
3. **快照隔离（SI）**：比可串行化更弱，确保每个事务看到数据库的一致快照。

隔离性违例通常表现为微妙且难以检测的异常，但这些异常可能导致严重的应用层错误。

  
 *图 2：三个事务之间的写偏差异常示例，展示了隔离性不足如何导致数据不一致。*

## **3、微事务方法**

本文提出的核心创新是**微事务（Mini-Transactions，MTs）**的概念。MTs 是小型且聚焦的事务，具有以下特征：

1. **紧凑大小**：MTs 包含有限数量的读写操作（通常为 2-4 个）。
2. **读-修改-写模式**：MTs 遵循一种模式，首先读取一个值，进行计算，然后写入一个新值。
3. **唯一值条件**：每个写操作都会产生一个在历史中从未出现过的唯一值。

这些特性使得微事务能够支持更高效的验证算法，同时保持检测隔离性违规的能力。作者展示了尽管 MTs 设计简单，它们仍能捕捉当前框架中文献所定义的所有 14 种已知的隔离性异常。

# 伪代码：一个简单的微事务示例  
# 事务目标：读取值，执行计算，写入新值  
BEGIN TRANSACTION  
 value = READ(key)  
 newValue = function(value)   
 WRITE(key, newValue)   
COMMIT

**使用微事务（MTs）的几个优势**：

1. **较低的中止率**：与普通事务（最高可达 60% 的中止率）相比，微事务的中止率显著较低（低于 10%），从而使得历史生成过程更加高效。
2. **简化的复杂度**：微事务的紧凑特性简化了验证过程。
3. **高效的算法**：微事务的特定模式使得开发高效的验证算法成为可能。

## 4、微事务的验证算法

作者为检查不同隔离级别下的微事务历史，开发了专门的验证算法：

### （1）严格可串行化（SSER）验证

对于SSER验证，算法构建一个依赖图，其中：

* 节点表示事务
* 边表示事务之间的依赖关系（写读依赖（WR）、写写依赖（WW）、读写依赖（RW）或会话顺序）

与传统方法使用SAT/SMT求解器不同，基于微事务的方法可以直接检测图中的循环以识别SSER违规。该方法实现了线性时间复杂度 (O(|T| + |E|))，相比现有方法有显著的改进。

### （2）可串行化（SER）验证

SER验证算法类似地构建依赖图，但忽略了会话顺序边。该算法利用微事务的特性，能够高效地进行循环检测。

### （3）快照隔离（SI）验证

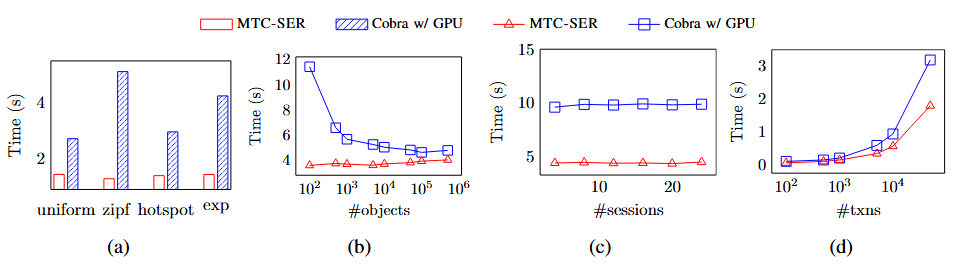
对于SI验证，算法检查特定的依赖模式，来识别SI违规，特别关注在SI下常见的写偏移异常（write-skew anomaly）。

这些算法的高效性源于微事务的唯一值条件，该条件简化了依赖分析，并减少了潜在违规的搜索空间。

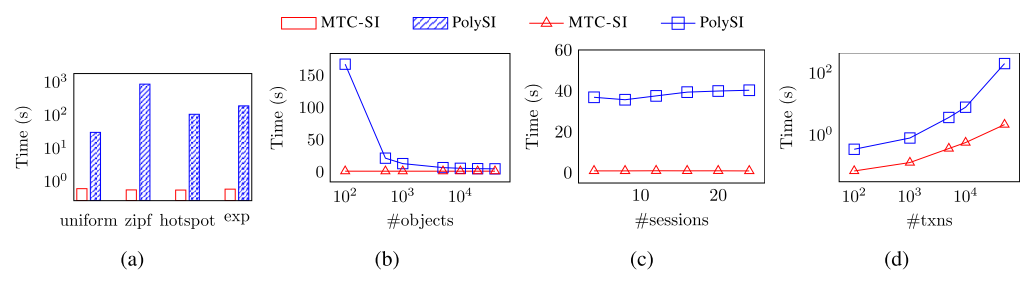
## **5、实验效果**

本研究在 PostgreSQL 和 MongoDB 数据库中，使用 Mini-Transaction（MT）负载和 General Transaction（GT）负载生成高并发事务历史，进行隔离级别检测实验。评价指标分为四类：  
 （1）验证性能（验证时间）、  
 （2）端到端检查开销（总时间、内存消耗）、  
 （3）事务提交率（Abort Rate）、  
 （4）实际 Bug 检测效果（Bug 数量、检测时间）。

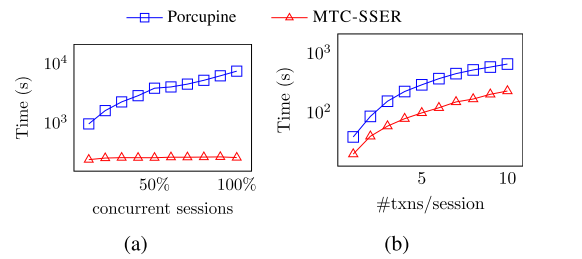
图 3 至图 5 分别展示了在验证 Serializability（SER）、Snapshot Isolation（SI）和 Strict Serializability（SSER）历史时，MTC 与现有主流方法（Cobra、PolySI、Porcupine）的性能对比。图 6 至图 8 展示了端到端检测开销、事务提交率。图 9 和图 10 进一步评估了 MTC 与 Elle 工具在 Bug 检测数量和检测效率上的差异。  
在 PostgreSQL 和 MongoDB 的真实和合成数据集上，MTC 在所有指标上均优于现有的基线方法，验证效率提升最高可达1600倍，端到端检测时间和内存开销大幅下降。MTC 生成的 MT 负载提交率更高，有效避免了 GT 负载中的高 Abort 问题。同时，在相同检测时间下，MTC 能发现更多 Isolation Bug，且 Bug counter example 更简洁、易于分析。



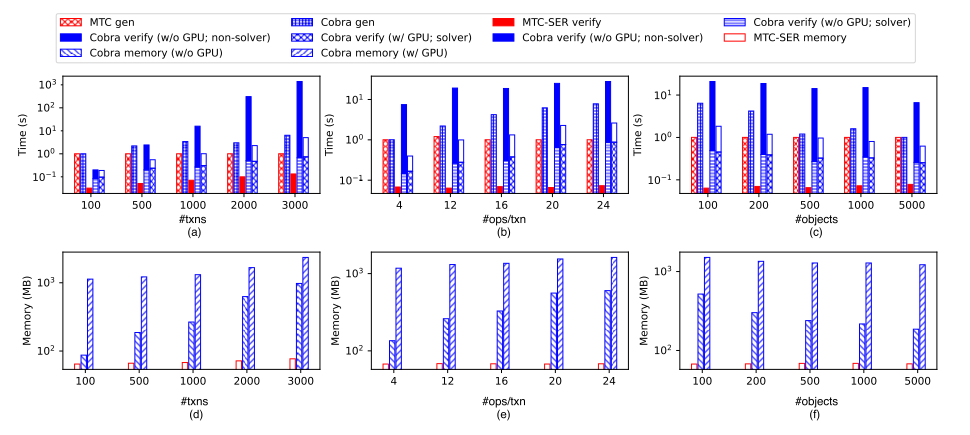
*图 3：MTC-SER与Cobra（有/无GPU加速）在不同负载条件下验证SER历史的性能比较。*



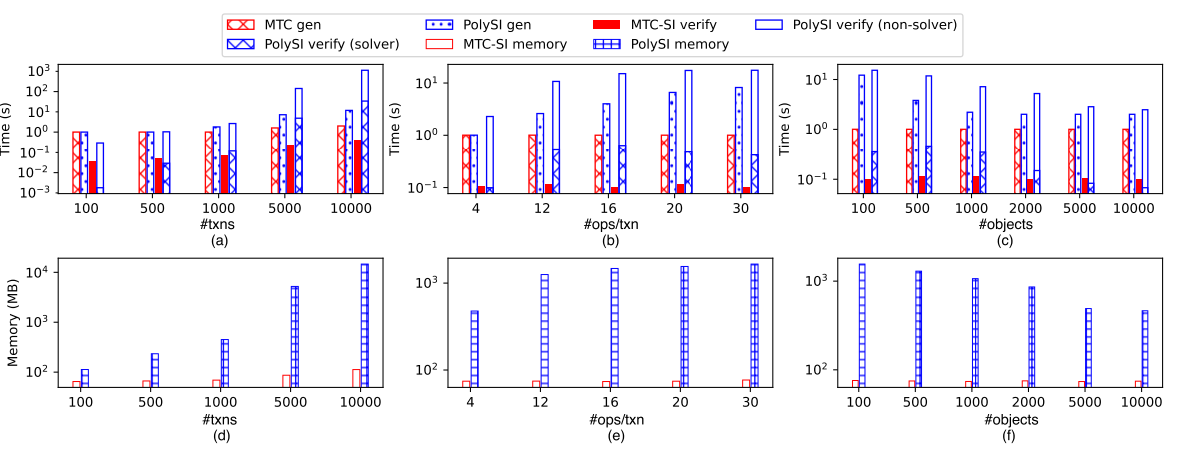
*图 4：MTC-SI与PolySI在不同负载条件下验证SI历史的性能比较。*



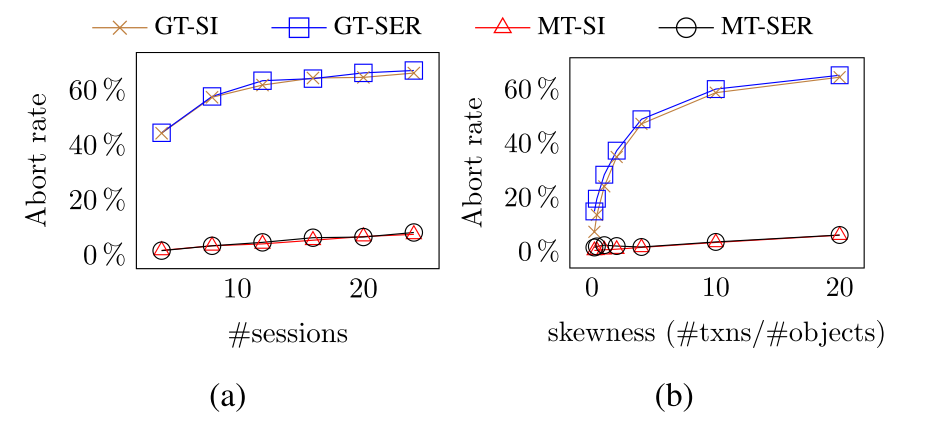
*图 5：MTC-SSER与Porcupine在不同并发度下验证SSER历史的性能比较。*



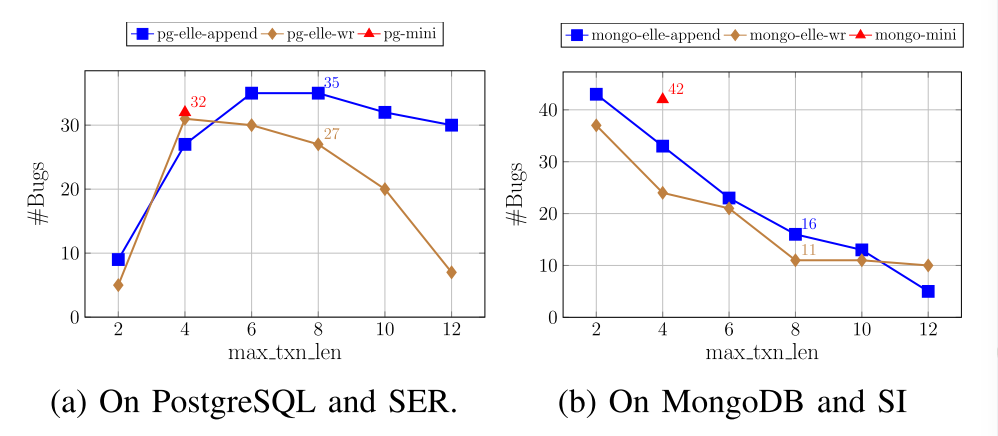
*图 6：比较MTC和Cobra的整体时间和内存消耗，包括****生成历史****和****验证历史****两个阶段（SER)。*



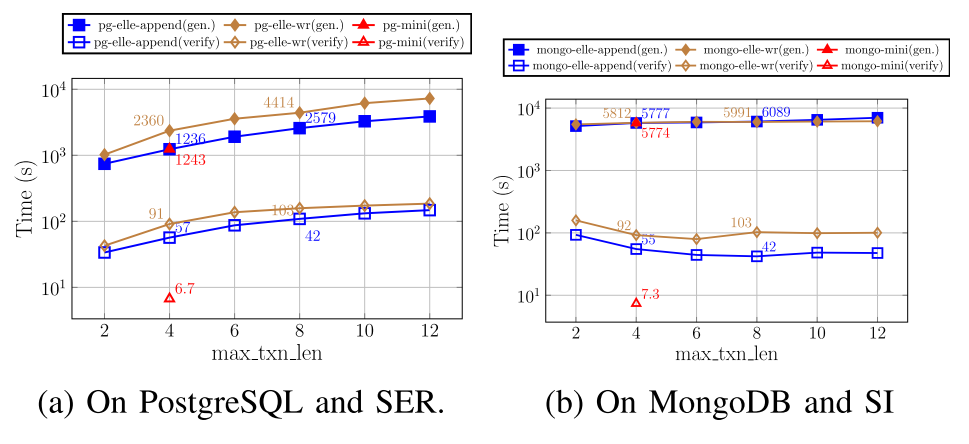
*图 7：比较MTC和PolySI的整体时间和内存消耗，包括****生成历史****和****验证历史****两个阶段（SI)。*



*图 8：比较GT（General Transaction）负载和MT（Mini-Transaction）负载的事务终止率。*



*图 9：在30分钟内比较MTC和Elle在PostgreSQL和MongoDB上检测Isolation Bug的数量。*



*图 10：比较MTC和Elle在不同事务长度设置下的****历史生成****和****历史验证****时间。*

## **5、结论**

本文在数据库隔离性测试领域做出了多项重要贡献：

1. **微事务（Mini-Transactions）**：提出了微事务的概念，作为一种成本较低的方法，用于生成测试负载并高效验证隔离性属性。
2. **高效验证算法**：针对严格可串行化、可串行化和快照隔离，开发了适用于微事务历史的专门验证算法，具有线性或二次时间复杂度。
3. **实用工具**：将上述算法实现于工具 **MTC** 中，实验结果表明其在性能上相较于现有工具具有显著提升。
4. **理论分析**：对验证问题的复杂度以及微事务的表达能力进行了深入的理论分析。

微事务方法为数据库测试提供了一种更加高效且实用的解决方案，有助于保障隔离性正确性。通过克服现有工具在性能方面的局限，MTC 能够实现对数据库系统更为全面的测试，进而有望提升数据管理的可靠性，减少生产环境中的隔离性相关错误。