计划类别：省基础研究专项资金（自然科学基金）

指南代码：1401

申报代码：JB0103

项目受理号：SBK2024024200

江苏省科技计划项目申报书

（面上项目）

项目名称：数据库系统事务一致性验证问题研究

承担单位：南京大学

单位地址：江苏省南京市鼓楼区汉口路 22 号

项目负责人：魏恒峰 电话：13905194610

项目联系人：华新 电话：13512501980

主管部门：南京大学

申报日期： 2024 年 04 月 03 日

江苏省科学技术厅

二〇二四年

江苏省科技计划（资金）项目
项目负责人科研诚信承诺书

本人在省科技计划（资金）项目（**项目名称：数据库系统事务一致性验证问题研究 项目受理号：**SBK2024024200 ）申报、实施、验收等过程中，将严格遵守《江苏省科技计划项目信用管理办法》（苏科技规〔2022〕3 号）、江苏省科技计划项目管理办法和专项资金管理办法等相关规定和要求，并作出如下承诺：

1.如实填写项目申报材料、项目年度实施情况、总结报告、科技成果、验收材料、科技报告、科学数据等，对上述材料的真实性、完整性、有效性和合法性负直接责任。

2.恪守科研诚信，无抄袭或剽窃他人科研成果、捏造或篡改科研数据、侵犯他人知识产权、在职称简历和研究基础等方面提供虚假信息、违反科学伦理，以及其他科研不端及科研失信行为；没有通过贿赂或变相贿赂、故意重复申报等不正当手段申报项目；督促项目组成员恪守科研诚信并履行相关承诺，保证项目组成员身份及业绩真实有效。

3.按照项目合同约定组织、协调、推进项目实施，按期完成项目目标任务；依法依规使用项目经费，保证不发生套取、转移、挪用、贪污科研经费等行为。

4.在项目实施中，因科研活动实际需要，项目负责人可

以在项目总预算不变的情况下自主调整直接费用相关科目的经费支出，自主调整科研团队，在不降低研究目标的前提下自主调整研究方案和技术路线，报项目承担单位办理调剂手续、备案。对于项目合同约定的主要研究目标或关键考核指标发生变化的，以及其他严重影响项目实施的重大事项，及时报项目承担单位审核，由承担单位报主管部门和省科技厅。

5.加强项目组成员在项目实施过程中的科研诚信管理，若发现科研不端及科研失信行为，及时报告并积极配合相关部门调查处理。

若发生上述失信行为，本人将积极配合调查，并按照有关规定接受警告、通报批评、取消项目评审资格、撤销项目立项、阶段性或永久取消省科技计划项目和科技奖励申报资格等处理并记入不良科研信用记录，情节严重的按相关规定报送至省公共信用信息平台、列入社会信用记录、实施失信联合惩戒等，依法依规予以处理。

项目负责人（签字）：

_____年____月____日

江苏省科技计划（资金）项目
项目承担单位科研诚信承诺书

本单位在省科技计划（资金）项目（**项目名称：数据库系统事务一致性验证问题研究　项目受理号：**SBK2024024200 ）申报、实施、验收等过程中，将严格遵守《江苏省科技计划项目信用管理办法》（苏科技规〔2022〕3号）、江苏省科技计划项目管理办法和专项资金管理办法等相关规定和要求，并作出如下承诺：

1.严格审核把关项目申报材料、项目年度实施情况、总结报告、验收材料、科技报告、科学数据等，对上述材料的真实性、完整性、有效性和合法性负主体责任。

2.履行科研诚信管理责任，按照规定建立规范科研行为、调查处理科研不端及科研失信行为的相关制度，与本单位项目组成员签订科研诚信承诺书，督促其恪守科研诚信并履行相关承诺，保证本单位项目组成员身份、科技成果及科研业绩真实有效，无编报虚假预算、篡改单位财务数据、侵犯他人知识产权等科研不端及科研失信行为；没有通过贿赂或变相贿赂、故意重复申报等不正当手段申报项目，严肃查处发现的科研不端及科研失信行为。

3.严格执行项目管理规定，按照项目合同约定推进项目实施，落实相关项目保障条件，完善经费管理内控制度和监

督制约机制，加强对经费使用的监督和管理，保证经费专款专用，对项目经费实行单独核算，保证不发生套取、转移、挪用科研经费等行为。

4. 如发生项目负责人变更、承担单位变更、合同约定的主要研究目标或关键考核指标需要调整，以及其他严重影响项目实施等重大事项的，及时报主管部门和省科技厅。

若发生上述失信行为，本单位将积极配合调查，并按照有关规定接受警告、通报批评、取消项目评审资格、撤销项目立项、终止项目执行、追回已拨资金、阶段性或永久取消省科技计划项目和科技奖励申报资格等处理并记入不良信用记录，情节严重的按相关规定报送至省公共信用信息平台、列入社会信用记录、实施失信联合惩戒等，依法依规予以处理。

单位法人代表（签字）：　　（公　　章）

_____年____月____日

江苏省科技计划（资金）项目
项目主管部门科研诚信承诺书

本单位在省科技计划（资金）项目申报、实施、验收等过程中，将严格遵守《江苏省科技计划项目信用管理办法》（苏科技规〔2022〕3号）、江苏省科技计划项目管理办法和专项资金管理办法等相关规定和要求，并作出如下承诺：

1.本单位已切实履行审核责任，项目申报单位提交的申报资料完整齐全、真实有效，项目申报书附件清单中所列证明材料的完整性与项目信息表、项目申报书中内容一致，该单位无不良信用记录，项目负责人和申报单位符合申报资格要求；审核推荐项目过程中，无违规推荐、审核不严等行为。

2.切实履行主管部门管理职责，及时协调划拨省科技计划项目经费，监督项目实施和经费使用，督促项目承担单位及负责人按期实施和完成项目。

3.协助或接受委托做好项目检查、评估、验收和绩效评价等，协调项目的实施推进，及时向省科技厅报送项目实施情况和需解决的问题等。

4.加强对项目承担单位重大事项变更报告的审核，并及时报省科技厅。

5.做好项目执行情况和经费使用统计工作，积极配合省科技厅对项目承担单位及项目负责人进行信用评价。

若发生上述失信行为，本单位将积极配合调查，追究相

关人员责任，并按照有关规定承担相关责任。

单位负责人（签字）：　　（公　　章）

_____年____月____日

填报说明

填写申报书前，请先查阅《江苏省基础研究计划（自然科学基金）管理办法》及《关于印发<2024年度省基础研究专项资金（自然科学基金）项目指南>及组织申报项目的通知》。申报书各项内容，要实事求是，逐条认真填写。表达要明确、严谨，字迹要清晰。外来语要同时用原文和中文表达。第一次出现的缩写词，须注出全称。

审核推荐表

承担单位	（请出具具体审核推荐意见） 法人代表（签章） （公章） 年　　月　　日
合作单位	（请出具具体审核推荐意见） 法人代表（签章） （公章） 年　　月　　日
主管部门（市、县、国家或省高新区科技局、省有关厅局）	（请出具具体审核推荐意见） （公章） 年　　月　　日

备注：1、审核推荐表的签章、公章及日期须完整齐全。

2、自主推荐申报的部省属本科院校，既要在承担单位栏目签字盖章，也要在主管部门栏目签字盖章。

一、立项依据和研究内容（4000-8000 字）

1、项目的立项依据

（研究意义、国内外研究现状及发展动态分析，需结合科学研究发展趋势来论述科学意义；或结合国民经济和社会发展中迫切需要解决的关键科技问题来论述其应用前景。附主要参考文献目录）

数据库系统作为统一组织、存取、维护数据的基础软件，已广泛应用于金融、电信、政府、制造、交通等各大行业。从 2016 年起，作为信创产业链中的重要一环，数据库产业已逐渐成为我国国家战略的一部分。

事务是数据库系统中的核心概念，是支撑在线交易业务顺利进行的关键技术。事务具有 ACID 四大特性。其中，隔离性（Isolation）要求并发执行的事务之间互不干扰，是避免上层业务产生数据异常的关键。根据程度不同，隔离性包含若干种隔离级别[1][2]，也称事务一致性模型[3]。事务一致性模型是数据库系统与上层业务之间的一种契约。数据库系统负责实现某种事务一致性模型，上层业务则基于该模型开发客户程序。

然而，很多事务型数据库未能正确实现它们所声称的事务一致性[4]。因此，充分的测试是数据库系统开发与使用过程中的必要环节。数据库系统测试流程包括构造测试用例（即包含事务的客户程序）、运行数据库系统、收集运行时信息（比如事务日志）、抽取事务相关的执行历史、验证执行历史是否满足期望中的事务一致性。其中，最后一步决定着整个流程是否可行、是否高效，是本项目关注的研究问题之一，称为①**数据库系统执行历史验证问题**[5][6]。

基于（弱）事务一致性模型开发客户程序是一项极易出错的任务。②**客户程序断言验证问题**[7]要求判定客户程序在给定一致性模型下运行终止时是否一定满足程序断言；③**客户程序健壮性验证问题**[8]要求判定客户程序在某个弱一致性模型下运行产生的执行历史是否都满足更强的某个一致性模型。如果满足健壮性，则该客户程序可以运行在性能更优的、较弱的一致性模型下，同时不会引入额外的数据异常。

本项目将上述三个问题统称为“**事务一致性验证**”问题。图 展示了它们之间的递进关系：问题①考虑单个执行历史相对于某个事务一致性模型的正确性，问题②考虑客户程序产生的所有执行历史相对于某个事务一致性模型的正确性，而问题③则考虑客户程序在较弱的事务一致性模型下产生的所有执行历史相对于另一个较强的事务一致性模型的正确性。由此可见，问题的难度依次递增，并且较难的问题以较易的问题为基础。

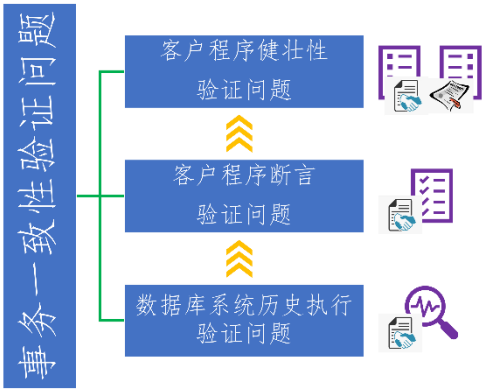

图 1 三个事务一致性验证问题之间的关系

从本质上讲，这三个验证问题都要从执行历史或者客户程序（可建模为有向图）中搜索违反一致性模型的结构（可建模为符合某种模式的环），属于同一类搜索问题。已有理论表明，这三个问题的复杂度都很高，某些情况下甚至是不可判定的[9][10]。因此，大多数解决方案都借助于高效的 SAT/SMT[11][12][13]求解器。然而，据我们所知，**它们通常把关注点放在前端编码阶段，而将后端 SAT/SMT 作为黑盒使用，未能充分发挥 SAT/SMT 框架中高效的搜索策略的优势。**

本项目通过设计特定的事务一致性理论[1]及求解器，并与 SAT/SMT 中经典的 CDCL（Conflict-Driven Clause Learning）搜索框架进行深度整合，进一步提升事务一致性验证问题的求解效率。

研究并解决上述科学问题将有助于：

（1） 加速“深度整合 SMT”技术趋势，促进事务一致性验证领域发展

[1] 此处的“理论”是 SMT（Satisfiability Modulo Theories）中 Theory 的含义。

“深度整合 SMT”技术方案是解决事务一致性验证问题的一种趋势。2019年，Biswas 与 Enea[6]提出“有可能设计一种类似 CDCL 的搜索算法，以提高数据库系统执行历史的验证效率”。2021 年，贺飞等人[14]在解决多处理器环境下客户程序断言验证问题时设计了首个内存一致性模型理论及其求解器，发挥了该技术方案的巨大潜力。2022 年，便出现了多份相关工作[15][16][17][18]。目前，这些工作还集中在多处理器领域，解决与内存一致性模型相关的验证问题。我们亟需在数据库系统领域探索该技术方案，更好地解决事务一致性验证问题。

（2） 加强数据库系统与客户程序正确性保障，推进数据库应用稳步增长

数据库产业的快速发展离不开数据库在各行各业的全面落地应用，而后者要以数据库系统以及客户程序的正确性为基础。探索当下具有创新性的技术方案，解决事务一致性验证问题，既有助于帮助数据库开发人员打造可靠的数据库产品，也有助于上层业务程序员开发符合业务逻辑的客户程序，避免因系统故障或程序错误带来的难以估量的经济损失。

国内外研究现状与发展动态分析

（1）数据库系统执行历史验证问题：经典工作采用 SAT/SMT 技术，未见深度整合 SMT 方案

2019 年，Biswas 与 Enea[6]证明了在执行历史的版本序（Version Order）未知的情况下，验证其是否满足 SER（**Ser**ializability；可串行化）[5]与 SI（**S**napshot **I**solation；快照隔离）事务一致性都是 NP-完全问题。近几年来，研究人员考虑将验证问题转化为约束求解问题，借助 SAT/SMT 进行求解。

Biswas 与 Enea[6]提出了一种增量式的 SER 与 SI 验证算法，称为 dbcop。然而，dbcop 仍属蛮力枚举算法，性能较低，仅能处理小规模的执行历史，实用性差。作者还提出了一种基于 SAT 的 SER 验证算法：将 SER 验证问题编码成命题逻辑公式，交由 MiniSAT 求解。同样地，该方案未经优化，性能较低。

Cobra[19]使用 polygraph[5]建模执行历史中不确定的版本序，将 SER 验证问题转化为 polygraph 上的判环问题，交由 MonoSAT[20]求解。Cobra 是目前已知最高效的黑盒 SER 验证工具。

VIPER 是[21]类似 Cobra 的 SI 验证工具。Viper 使用 BC-polygraph 建模执行历史中不确定的版本序以及事务开始（**B**egin）与提交（**C**ommit）的顺序，将 SI

验证问题转化成 BC-polygraph 上的判环问题，交由 MonoSAT 求解。

PolySI[22]是本项目申请人及其合作者开发的一种 SI 验证工具。我们基于 Cerone 与 Gotsman[23][24]提出的 SI 依赖图刻画定理，将 SI 验证问题转化为 polygraph 某种变体上的判环问题，交由 MonoSAT 求解。大量实验表明，PolySI 性能优异，内存消耗少，可扩展性强，可以在 40 小时内处理具有百万级事务数量、十亿级键数量的执行历史（其它同类工具均无法处理）。

然而，以上工具均将 SAT/SMT 求解器当作黑盒使用。仍需探索如何在数据库系统执行历史验证问题中应用“深度整合 SMT”的技术方案，大幅提升工具效率。

（2）客户程序断言验证问题：在多处理器领域“深度整合 SMT”方案成果颇丰，亟需探索数据库系统领域

在多处理器并发程序断言验证领域，研究人员通常采用 SMC（**S**tateless **M**odel **C**hecking）技术遍历并发程序状态空间，配合 DPOR（**D**ynamic **P**artial **O**rder **R**eduction）技术[25]避免重复遍历等价的程序路径，并借助 SAT/SMT 求解器判定是否每条路径都满足断言[26][27][28][29][30]。然而，此类工作仍将 SAT/SMT 当作黑盒使用。

2021 年，贺飞等人[14]提出了针对 SC（Sequential Consistency；内存顺序一致性）模型的 SMT 理论，设计了包括理论传播（Theory Propagation）、一致性检测、冲突子句生成等模块的理论求解器，并实现了高效验证工具 ZORD。作者等人在后续工作[15][16]中分别从决策变量选择策略与一致性检测及其冲突子句生成两个方面对此进行了改进，设计并实现了验证工具 DEAGLE[31][32]，在 SV-COMP2022 竞赛 *ConcurrencySafety* 组中获得了冠军。

Marmanis 与 Vafeiadis[17]考虑在 x86 持久性内存一致性模型 $\text{DPTSO}_{\text{syn}}$ 下，并发程序相对于持久性不变式的正确性验证问题。作者设计了针对 $\text{DPTSO}_{\text{syn}}$ 的 SMT 理论及其求解器，实现了首个满足完备性的验证工具，并在中等规模的基准程序上取得了良好效果。

Haas 等人[18]提出了 CAAT（**C**onsistency **a**s **a** **T**heory）理论，将 SMC 技术与针对内存一致性模型设计的 SMT 理论求解器相结合。该工作有两大创新：第一，不限于某个具体的内存一致性模型，而是提出了一种抽象的内存一致性模型，使

得该理论的应用范围更广；第二，选取了一个子类，其表达能力足以涵盖多种重要的内存一致性模型（包括 TSO、POWER、ARMv8、RISC-V、RC11 等），并为其设计了通用的、高效的理论求解算法。

以上工作考虑的均是多处理器领域中并发程序断言验证问题。仍需探索如何在数据库系统领域客户程序断言验证问题中应用“深度整合 SMT”的技术方案，大幅提升工具效率。

（3）客户程序健壮性验证问题：少量工作采用 SAT/SMT 技术，未见深度整合 SMT 方案

从技术方案的角度看，现有工作可以分为两类：一类仅给出客户程序健壮性的充分条件，另一类则给出健壮性的充要条件。不同的解决方案都试图在精确性与性能之间取得平衡。

2005 年，Fekete 等人[33]首次研究数据库系统中客户程序健壮性验证问题。作者基于依赖图给出了已知满足 SI 的执行历史违反 SER 的充分条件，提出了可用于建模客户程序的静态依赖图概念，推导出了客户程序在 SI 下相对于 SER 的健壮性的充分条件，并证明了 TPC-C 基准程序的健壮性。

直到 2016 年，Cerone 与 Gotsman[23][24]给出了基于依赖图刻画 SI 的充要条件，并提出了客户程序在 SI 下相对于 SER 的、更为精确的健壮性充分条件。[33]使用的是 SI 的操作语义，而[23]则使用了 SI 的公理语义[3]，推理更为简单自然，开启了后续一系列健壮性验证工作。Bernardi 与 Gotsman[8]以一种统一的方式系统地研究了客户程序在 CC（**C**ausal **C**onsistency）[3]、PC（**P**refix **C**onsistency）[3]、PSI（**P**arallel **S**napshot **I**solation）[34]下相对于 SER 的健壮性问题。[23][24]还给出了客户程序在 PSI 下相对于 SI 的健壮性充分条件。Ketsman 等人[35]以及 Vandevoort 等人[36]研究了客户程序在 RC（**R**ead **C**ommitted）与 RU（**R**ead **U**ncommitted）下相对于 SER 的健壮性充分条件以及验证复杂度。这一系列充分条件均要求客户程序的静态依赖图中不存在某种形式的环[37]。

Bouajjani 等人[38]指出，客户程序的静态依赖图是程序实际运行时可能发生的依赖的过近似（over-approximation）或欠近似（under-approximation），因此上述工作给出的程序健壮性结论是不精确的，比如可能产生大量的假警报，也就是说原本满足健壮性的客户程序被误判为不满足。Brutschy 等人[39][40]利用操作

之间的可交换性与可吸收性扩展了客户程序上的静态依赖图，有效地减少了假警报。通过将不同模型下的健壮性验证问题归约到 SC 下的可达性问题，Bouajjani 等人[41]给出了客户程序在 CC[42]、SI[9]下相对于 SER 的健壮性充要条件。Bouajjani 等人[43]还给出了客户程序在 CC 或 PC 下相对于 SI 的健壮性充要条件，以及在 CC 下相对于 PC 的健壮性充要条件。在事务类型受限于特定模板的情况下，Vandevoort 等人[44]给出了客户程序在 RC 下相对于 SER 的健壮性充要条件。

Nagar 与 Jagannathan[45]使用一阶谓词逻辑公式对客户程序以及事务一致性模型进行精确建模，利用 SAT/SMT 自动推导出违反该一致性的执行历史的结构，并在客户程序可能产生的所有执行历史中搜索该结构。León 等人[46][47][48]利用 SAT 编码猜测一条满足弱一致性模型的执行历史，并利用 SAT 求解器搜索一条等价的、但不满足强一致性模型的执行历史。

相关工作[40]、[45]、[47]（包括多处理器领域并发程序的健壮性验证相关工作[49][50]）都采用了对客户程序（的操作语义[51][52][53][54][55][56]）与事务一致性模型进行编码，并利用 SAT/SMT 进行求解的技术框架。然而，**它们都将 SAT/SMT 作为黑盒使用。仍需探索如何将编码、验证过程与 SAT/SMT 搜索框架进行深度整合，大幅提升验证效率，开发实用的自动化健壮性验证工具。**

本项目旨在采用“深度整合 SMT”的思想（设计专用的一致性模型理论，并与 SAT/SMT 的高效搜索框架进行深度整合），研究数据库系统中的事务一致性验证问题，包括数据库系统执行历史验证问题、客户程序断言验证问题以及客户程序健壮性验证问题，提升算法效率，开发实用的自动化验证工具。**“深度整合 SMT”的思想代表了一种技术趋势**，开展上述研究，有助于推进事务一致性验证领域的发展，保障数据库系统与客户程序的正确性，为数据库产业发展奠定扎实的基础。

主要参考文献

[1] Hal Berenson, Phil Bernstein, Jim Gray, Jim Melton, Elizabeth O'Neil, and Patrick O'Neil. 1995. A critique of ANSI SQL isolation levels. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data* (SIGMOD '95). 1–10.

[2] Atul Adya. Weak consistency: a generalized theory and optimistic

implementations for distributed transactions. Ph.D., MIT, Cambridge, MA, USA, Mar. 1999.

[3] Cerone, A., Bernardi, G., and Gotsman, A. A framework for transactional consistency models with atomic visibility. In *26th International Conference on Concurrency Theory* (CONCUR 2015), 58–71.

[4] Kyle Kingsbury and Peter Alvaro. 2020. Elle: inferring isolation anomalies from experimental observations. Proc. VLDB Endow. 14, 3 (Nov. 2020), 268–280.

[5] Christos H. Papadimitriou. 1979. The serializability of concurrent database updates. *J. ACM*. 26, 4 (Oct. 1979), 631–653.

[6] Ranadeep Biswas and Constantin Enea. 2019. On the complexity of checking transactional consistency. *Proc. ACM Program. Lang*. 3, OOPSLA, Article 165 (October 2019), 28 pages.

[7] Ahmed Bouajjani, Constantin Enea, and Enrique Román-Calvo. 2023. Dynamic Partial Order Reduction for Checking Correctness against Transaction Isolation Levels. Proc. ACM Program. Lang. 7, PLDI, Article 129 (June 2023), 26 pages.

[8] Bernardi, G., and Gotsman, A. Robustness against consistency models with atomic visibility. In *27th International Conference on Concurrency Theory* (CONCUR 2016), 7:1–7:15.

[9] Beillahi, S.M., Bouajjani, A., Enea, C. (2019). Checking robustness against snapshot isolation. In: Dillig, I., Tasiran, S. (eds) Computer Aided Verification. CAV 2019.

[10] Beillahi, S. M., Bouajjani, A., and Enea, C. Robustness against transactional causal consistency. In *30th International Conference on Concurrency Theory* (CONCUR 2019), 30:1–30:18.

[11] Clark W. Barrett, Roberto Sebastiani, Sanjit A. Seshia and Cesare Tinelli, "Satisfiability modulo theories" in Handbook of Satisfiability volume 185 of Frontiers in Artificial Intelligence and Applications, IOS Press, pp. 825-885, 2009.

[12] João P. Marques-Silva and Karem A. Sakallah. 1999. GRASP: a search algorithm for propositional satisfiability. IEEE Trans. Comput. 48, 5 (May 1999), 506–521.

[13] Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. 2006. Solving SAT and SAT Modulo Theories: From an abstract Davis--Putnam--Logemann--Loveland procedure to DPLL(T). J. ACM 53, 6 (November 2006), 937–977.

[14] Fei He, Zhihang Sun, and Hongyu Fan. 2021. Satisfiability modulo ordering consistency theory for multi-threaded program verification. In Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI 2021). 1264–1279.

[15] Hongyu Fan, Weiting Liu, and Fei He. 2022. Interference relation-guided SMT solving for multi-threaded program verification. In Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '22). Association for Computing Machinery, New York, USA, 163–176.

[16] Zhihang Sun, Hongyu Fan, and Fei He. 2022. Consistency-preserving propagation for SMT solving of concurrent program verification. Proc. ACM Program. Lang. 6, OOPSLA2, Article 158 (October 2022), 28 pages.

[17] Iason Marmanis and Viktor Vafeiadis. 2023. SMT-Based verification of persistency invariants of Px86 programs. In Verified Software. Theories, Tools and Experiments.: 14th International Conference, VSTTE 2022, Trento, Italy, October 17–18, 2022, Springer-Verlag, Berlin, Heidelberg, 92–110.

[18] Thomas Haas, Roland Meyer, and Hernán Ponce de León. 2022. CAAT: consistency as a theory. Proc. ACM Program. Lang. 6, OOPSLA2, Article 129 (October 2022), 31 pages.

[19] Cheng Tan, Changgeng Zhao, Shuai Mu, and Michael Walfish. 2020. COBRA: Making transactional key-value stores verifiably serializable. In OSDI’20. Article 4, 18 pages.

[20] Bayless, S., Bayless, N., Hoos, H., & Hu, A. (2015). SAT modulo monotonic theories. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1).

[21] Jian Zhang, Ye Ji, Shuai Mu, and Cheng Tan. 2023. Viper: A Fast Snapshot Isolation Checker. In Proceedings of the Eighteenth European Conference on Computer Systems (EuroSys '23). Association for Computing Machinery, New York, NY, USA, 654–671.

[22] Kaile Huang, Si Liu, Zhenge Chen, Hengfeng Wei, David Basin, Haixiang Li, and Anqun Pan. 2023. Efficient Black-Box Checking of Snapshot Isolation in Databases. Proc. VLDB Endow. 16, 6 (February 2023), 1264–1276.

[23] Andrea Cerone and Alexey Gotsman. 2016. Analysing snapshot isolation. In Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC '16). Association for Computing Machinery, New York, NY, USA, 55–64.

[24] Andrea Cerone and Alexey Gotsman. 2018. Analysing snapshot isolation. *J. ACM*, 65, 2, Article 11 (March 2018), 41 pages.

[25] Parosh Abdulla, Stavros Aronis, Bengt Jonsson, and Konstantinos Sagonas. 2014. Optimal dynamic partial order reduction. In Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '14). Association for Computing Machinery, New York, USA, 373–384.

[26] Parosh A. Abdulla, Stavros Aronis, Mohamed Faouzi Atig, Bengt Jonsson, Carl

Leonardsson and Konstantinos F. Sagonas. Stateless model checking for TSO and PSO. In TACAS volume 9035 of LNCS, Springer, pp. 353-367, 2015.

[27] Parosh Aziz Abdulla, Mohamed Faouzi Atig, Bengt Jonsson, and Tuan Phong Ngo. 2018. Optimal stateless model checking under the release-acquire semantics. Proc. ACM Program. Lang. 2, OOPSLA, Article 135 (November 2018), 29 pages.

[28] Parosh Aziz Abdulla, Mohamed Faouzi Atig, Bengt Jonsson, Magnus Lång, Tuan Phong Ngo, and Konstantinos Sagonas. 2019. Optimal stateless model checking for reads-from equivalence under sequential consistency. Proc. ACM Program. Lang. 3, OOPSLA, Article 150 (October 2019), 29 pages.

[29] Truc Lam Bui, Krishnendu Chatterjee, Tushar Gautam, Andreas Pavlogiannis, and Viktor Toman. 2021. The reads-from equivalence for the TSO and PSO memory models. Proc. ACM Program. Lang. 5, OOPSLA, Article 164 (October 2021), 30 pages.

[30] Michalis Kokologiannakis, Iason Marmanis, Vladimir Gladstein, and Viktor Vafeiadis. 2022. Truly stateless, optimal dynamic partial order reduction. Proc. ACM Program. Lang. 6, POPL, Article 49 (January 2022), 28 pages.

[31] Fei He, Zhihang Sun, and Hongyu Fan. 2022. Deagle: An SMT-based verifier for multi-threaded programs (Competition Contribution). In Tools and Algorithms for the Construction and Analysis of Systems: 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2–7, 2022, Proceedings, Part II. Springer-Verlag, Berlin, Heidelberg, 424–428.

[32] Dirk Beyer. 2022. Progress on software verification: SV-COMP 2022. In Tools and Algorithms for the Construction and Analysis of Systems: 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2–7, 2022, Proceedings, Part II. Springer-Verlag, Berlin, Heidelberg, 375–402.

[33] Alan Fekete, Dimitrios Liarokapis, Elizabeth O'Neil, Patrick O'Neil, and Dennis Shasha. 2005. Making snapshot isolation serializable. ACM Trans. Database Syst. 30, 2 (June 2005), 492–528.

[34] Yair Sovran, Russell Power, Marcos K. Aguilera, and Jinyang Li. 2011. Transactional storage for geo-replicated systems. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles* (SOSP '11). Association for Computing Machinery, New York, NY, USA, 385–400.

[35] Bas Ketsman, Christoph Koch, Frank Neven, and Brecht Vandevoort. 2020. Deciding robustness for lower SQL isolation levels. In *Proceedings of the 39th*

ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS'20). Association for Computing Machinery, New York, USA, 315–330.

[36] Brecht Vandevoort, Bas Ketsman, Christoph Koch, and Frank Neven. 2022. Robustness against read committed: a free transactional lunch. In Proceedings of the 41st ACM Symposium on Principles of Database Systems (PODS '22). 1–14.

[37] Cerone, A., Gotsman, A., and Yang, H. Algebraic laws for weak consistency. In *28th International Conference on Concurrency Theory* (CONCUR 2017), 26:1–26:18.

[38] Bouajjani, A., Derevenetc, E., Meyer, R.: Checking and enforcing robustness against TSO. In: Felleisen, M., Gardner, P. (eds.) Programming Languages and Systems - 22nd European Symposium on Programming (ESOP 2013), March 16-24, 2013, pp. 533-553.

[39] Lucas Brutschy, Dimitar Dimitrov, Peter Müller, and Martin Vechev. 2017. Serializability for eventual consistency: criterion, analysis, and applications. In Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL '17). Association for Computing Machinery, New York, NY, USA, 458–472.

[40] Lucas Brutschy, Dimitar Dimitrov, Peter Müller, and Martin Vechev. 2018. Static serializability analysis for causal consistency. In Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2018). Association for Computing Machinery, New York, USA, 90–104.

[41] Beillahi, S. M. (2021). Automated verification of programs running on top of distributed systems (Publication Number 2021UNIP7005). Ph.D. Dissertation. Université Paris Cité.

[42] Beillahi, S.M., Bouajjani, A., & Enea, C. (2019). Robustness against transactional causal consistency. International Conference on Concurrency Theory (CONCUR 2019). Article No. 30; pp. 30:1–30:18.

[43] Sidi Mohamed Beillahi, Ahmed Bouajjani, and Constantin Enea. 2021. Checking robustness between weak transactional consistency models. In Programming Languages and Systems: 30th European Symposium on Programming, ESOP 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 – April 1, 2021, Proceedings. Springer-Verlag, Berlin, Heidelberg, 87–117.

[44] Brecht Vandevoort, Bas Ketsman, Christoph Koch, and Frank Neven. 2021. Robustness against read committed for transaction templates. Proc. VLDB Endow. 14, 11 (July 2021), 2141–2153.

[45] Kartik Nagar and Suresh Jagannathan. 2018. Automated detection of

serializability violations under weak consistency. In 29th International Conference on Concurrency Theory, CONCUR 2018, September 4-7, 2018. 41:1-41:18.

[46] Florian Furbach (2021). Verification with memory models as input. Ph.D. Dissertation. Technische Universität Kaiserslautern.

[47] Hernán Ponce de León, Florian Furbach, Keijo Heljanko and Roland Meyer. Portability analysis for weak memory models. PORTHOS: one tool for all models. In SAS volume 10422 of Lecture Notes in Computer Science, Springer, pp. 299-320, 2017.

[48] H. Ponce-de-León, F. Furbach, K. Heljanko and R. Meyer. BMC with memory models as modules. Formal Methods in Computer Aided Design (FMCAD 2018), pp. 1-9.

[49] Jade Alglave, Daniel Kroening, Vincent Nimal, and Daniel Poetzl. Don't sit on the fence - a static analysis approach to automatic fence insertion. In CAV, volume 8559 of LNCS, pages 508–524. Springer, 2014.

[50] Jade Alglave and Luc Maranget. Stability in weak memory models. In CAV, volume 6806 of LNCS, pages 50–66. Springer, 2011.

[51] Natacha Crooks, Youer Pu, Lorenzo Alvisi, and Allen Clement. 2017. Seeing is believing: a client-centric specification of database isolation. In *Proceedings of the 2017 ACM Symposium on Principles of Distributed Computing* (PODC '17), Washington, DC, USA, July 25-27, 2017, 10 pages.

[52] Natacha Crooks. 2019. A client-centric approach to transactional datastores. Ph.D. Dissertation, The University of Texas at Austin.

[53] Kia Rahmani, Kartik Nagar, Benjamin Delaware, and Suresh Jagannathan. 2019. CLOTHO: directed test generation for weakly consistent database systems. Proc. ACM Program. Lang. 3, OOPSLA, Article 117 (October 2019), 28 pages.

[54] Shale Xiong, Andrea Cerone, Azalea Raad, Philippa Gardner: Data consistency in transactional storage systems: a centralised semantics. The *34th European Conference on Object-Oriented Programming* (ECOOP '20), 21:1-21:31.

[55] Shale Xiong. 2020. Parametric operational semantics for consistency models. Ph.D. Dissertation. Imperial College London.

[56] Ranadeep Biswas, Diptanshu Kakwani, Jyothi Vedurada, Constantin Enea, and Akash Lal. 2021. MonkeyDB: effectively testing correctness under weak isolation levels. Proc. ACM Program. Lang. 5, OOPSLA, Article 132 (October 2021), 27 pages.

2、项目的研究内容、研究目标和拟解决的关键科学问题

（此部分为重点阐述内容）

研究内容

本项目包含三项研究内容：

- **数据库系统执行历史验证问题**：给定数据库系统的一个执行历史，判定该执行历史是否满足数据库所声称的事务一致性模型。这是本项目研究内容中最基本的验证问题，我们将从 SAT/SMT 中 CDCL 搜索框架入手，重点探索如何设计一致性模型理论及其求解器。
- **客户程序断言验证问题**：给定一段客户程序、一组断言以及一个事务一致性模型，判定当该客户程序在该事务一致性模型下运行结束时，是否满足该组断言。客户程序可以包含复杂的 SQL 语句。我们将借鉴程序设计语言领域对程序语义（尤其是 SQL 语句的语义）的研究成果，探索如何为客户程序设计精确的编码方案。
- **客户程序健壮性验证问题**：给定一段客户程序以及两个有强弱关系的事务一致性模型，判定当该客户程序在较弱的事务一致性模型下运行时所产生的每个执行历史是否都满足较强的事务一致性模型。我们将借鉴多处理器领域并发程序验证中的动态偏序约简技术，探索如何应对程序状态空间爆炸带来的挑战。

研究目标

本项目面向事务型数据库系统与客户程序对正确性保障的迫切需求，针对当前事务一致性验证问题常用解决方法的不足，通过设计专用的事务一致性模型理论及其求解器，并与 SAT/SMT 的 CDCL 搜索框架进行深度整合，提升验证效率，开发实用的自动化验证工具。项目具体目标是提升三个事务一致性验证问题的求解效率，包括数据库系统执行历史验证问题、客户程序断言验证问题与客户程序健壮性验证问题。

拟解决的关键科学问题

一：如何为执行历史与客户程序设计高效且尽可能精确的编码方案？

数据库系统中的事务是由多个操作构成的序列，比多处理器系统中的读写变量的语义更复杂。数据库系统中的客户程序涉及事务操作，也比多处理器系统中

的并发程序的语义更复杂。对于客户程序来说，实际执行的事务数量是未知的，事务之间的 WR（Write-Read 读写关系）与 WW（Write-Write 写写关系）依赖关系也是未知的。此外，一个典型的、实用的客户程序可能包含 SQL 语句、条件语句、循环语句等语言结构，每个事务中的读操作、写操作是否一定会执行同样是未知的。保守的编码方式考虑所有潜在的读操作、写操作可能造成的所有 WR、WW、RW（Read-Write 写读关系）依赖关系，这种静态依赖图是非常不精确的，会产生大量的假警报。

为数据库系统执行历史与客户程序设计尽可能精确的编码方案是实施“深度整合 SMT”技术所面临的重大挑战。理想的编码方案要尽可能精确地刻画执行历史与客户程序的语义，在保证无漏报的前提下减少误报。需要解决的具体问题包括，如何精确地编码数据库系统执行历史中事务之间的动态依赖关系？如何尽可能精确地编码客户程序中事务之间的静态依赖关系，尤其是如何处理客户程序中的条件语句、循环语句、SQL 语句等语言结构？

二：如何为经典事务一致性模型设计专用的一致性理论及其求解器？

在 SAT/SMT 的求解框架中，SAT 求解器采用 CDCL 搜索算法为经过布尔抽象（Boolean Abstraction）的、独立于具体理论的公式搜索可满足性赋值。Theory 求解器根据赋值情况还原出一组理论约束并判定其是否可满足。如果理论约束是可满足的，则原公式也是可满足的。否则，Theory 求解器向 SAT 求解器返回一个冲突子句，说明冲突的原因。SAT 求解器与 Theory 求解器重复以上过程。

“深度整合 SMT”技术的核心是设计专用的一致性理论及其求解器，用作 SAT/SMT 求解器的后端，并在多个关键模块上替换 CDCL 搜索框架的默认实现，最大限度地发挥 SAT/SMT 求解器的优势，提升验证效率。需要解决的具体问题包括，如何为 SAT 求解器选择决策变量？如何为一致性理论求解器设计理论传播过程？如何为 CDCL 框架生成有效的冲突子句？

三：如何应对“状态空间爆炸”问题，压缩客户程序状态空间？

对于客户程序断言验证问题与健壮性验证问题，还需要应对“状态空间爆炸”带来的技术挑战。需要解决的具体问题包括，如何有效地压缩待搜索的程序状态空间？如何将客户程序编码、一致性理论求解器以及状态空间压缩策略进行深度整合，设计高效的验证算法？

3、拟采取的研究方案及可行性分析

（包括研究方法、技术路线、实验手段、关键技术等说明）

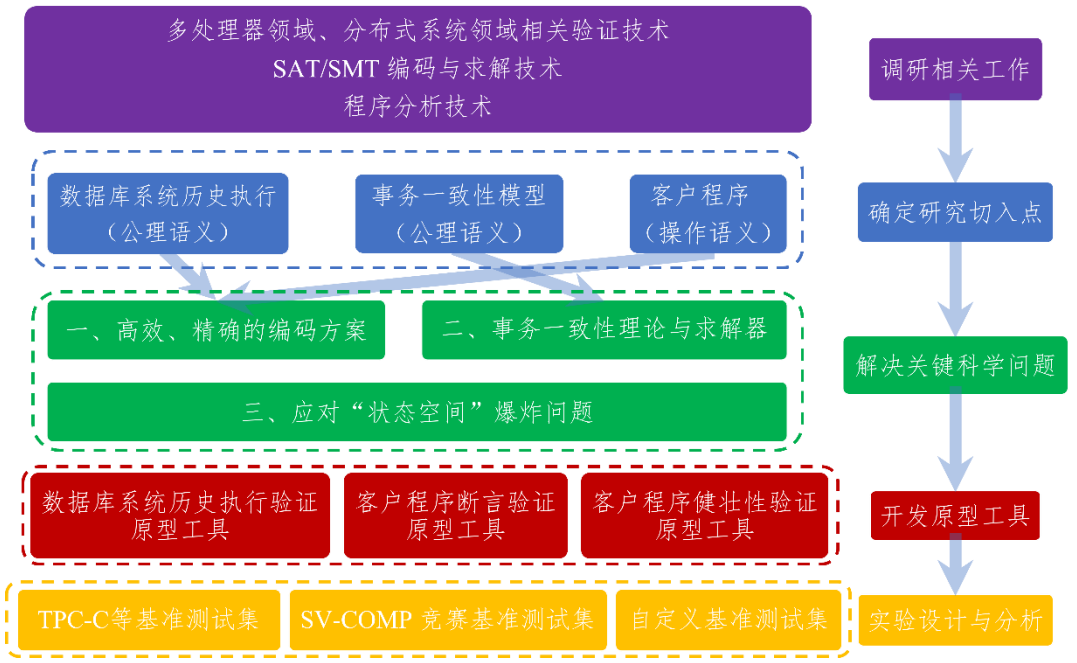

图 2 总体技术路线框图

技术路线

我们按照问题的难易程度依次研究问题①、②、③。图 给出了总体技术路线图。首先，充分调研相关工作，理解当前主流验证技术、SAT/SMT 编码与求解技术以及程序分析技术（例如数据流分析技术与控制流分析技术）。然后，以事务一致性模型、数据库系统执行历史以及客户程序的语义作为切入点，重点解决上文提出的关键科学问题。具体而言，基于数据库系统执行历史的公理语义以及客户程序的操作语义为它们设计高效的、精确的编码方案；基于事务一致性模型的公理语义构建一致性理论及其求解器；采用“语义等价类”思想缩减待搜索的状态空间，应对“状态空间爆炸”问题。接着，设计并实现相应的验证原型工具。最后，通过实验分析、检验理论与工具的实际效果，预计采用 TPC-C 等基准测试集、SV-COMP 竞赛所用的基准测试集以及从真实场景中提取的自定义基准测试集。

针对上文提炼的三个关键科学问题，我们提出如下具体研究方案。

关键问题一研究方案：基于操作语义并利用程序分析技术为客户程序的静态依赖图设计尽可能精确的编码方案

首先，一个数据库系统执行历史是由多个事务构成的偏序集，事务之间有三类依赖关系：WR 读写依赖、WW 写写依赖以及 RW 写读依赖。其中，RW 依赖可由 WR 与 WW 依赖推导得到。执行历史是数据库系统动态执行后得到的确定性结果，其中的每个读操作都包含了成功读到的值。因此，我们可以根据这些值推测事务之间的读写依赖关系，并编码为命题逻辑公式。另外，通过为同一变量上的每一对写事务引入一个布尔变量，我们也可以使用命题逻辑公式编码事务之间的 WW 依赖关系。因为只要存在一种合法的 WR、WW 依赖关系，即能说明该执行历史满足某事务一致性模型，所以这样的编码方案精确刻画了单个数据库系统执行历史的语义信息。

其次，为了精确地编码客户程序的静态依赖图，我们计划基于客户程序的操作语义为每种语言结构分别进行编码，这样得到的静态依赖图中刻画了每种依赖关系存在的条件。为了降低编码的复杂度，我们将利用程序分析技术排除不可能发生的依赖关系。比如，通过数据流分析与控制流分析，可以判断某些依赖关系不可能同时成立。

关键问题二研究方案：实例化 SAT/SMT 框架中的核心模块，构建事务一致性理论

事务一致性验证问题的本质是在有向图中搜索具有特定模式的环。为了提高搜索效率，我们计划设计事务一致性理论求解器。该求解器通过“决策变量选择”模块为 SAT 求解器确定下一步搜索的决策变量。SAT 求解器为决策变量赋值后（相当于向有向图中添加一条边），交由“一致性理论传播”模块确定相关变量的取值（相当于向有向图中添加一组相关边）。“一致性检测”模块判定是否已形成环。如果已检测到环，则由“冲突子句生成”模块生成冲突子句。

在**决策变量选择模块**中，我们为 SAT 求解器提供启发式规则，选择能尽快触发冲突的变量进行赋值。值得尝试的启发式规则包括（以 WW 类决策变量选择为例）：

- 选择与 SO、WR 重合的 WW 变量。该类 WW 变量的取值可以直接确定，无需 CDCL 尝试真/假两种情况。
- 选择关联更多 RW 变量的 WW 变量。选择该类 WW 变量会迫使 CDCL（在理论传播过程中）尽快确定更多 RW 变量的取值。
- 选择所在事务中操作数较多的 WW 变量等。选择该类 WW 变量更有可能

尽早发现与其它事务之间的冲突。

- 避免选择无关 WW 变量。如果一个变量的赋值不影响最终结果，则称该变量是无关变量。如果能在一致性理论求解器中识别出这些无关 WW 变量，则可以避免 SAT 求解器进行无用的搜索。需要注意的是，随着搜索过程不断推进，事务之间的依赖关系也逐渐明确。因此，识别无关 WW 变量是一个动态的、逐步求精的过程。

在客户程序断言验证问题与客户程序健壮性验证问题中，WR 依赖关系是未知。此时，需要同时考虑 WW 类与 WR 类决策变量。

在**一致性理论传播模块**中，RW 变量的取值可以由 WW 变量与 WR 变量通过理论传播来确定，无需 SAT 求解器进行试错。更关键的问题是，如何通过理论传播确定 WW 变量与 WR 变量的取值？对于客户程序断言验证问题与客户程序健壮性验证问题，随着求解过程不断深入，获取的程序语义也不断完善，当确定某个 WW 变量或 WR 变量的取值后，可以根据此时程序的数据流与控制流信息确定其它部分 WW 变量或 WR 变量的取值。

在**一致性检测模块**中，我们采用增量式算法进行环检测。关键的挑战在于，不同的事务一致性模型对应着不同模式的环，如何以增量的方式高效地检测它们？例如，违反 SI 的环中不能包含连续的 RW 边。也就是说，仅检测环是不够的，还要考虑边的类型与边的邻接情况。为此，我们计划利用关系代数中的关系复合操作对依赖图进行变换，使得在变换后的图中只需要查找是否有环。例如，根据 Cerone 与 Gotsman 给出的定理，一个执行历史满足 SI 当且仅当它的依赖图经过 $(SO \cup WR \cup WW);RW?)$ 变换后是无环的，其中‘?’表示关系的自反闭包，‘;’表示关系的复合。

在**冲突子句生成模块**中，我们的目标是生成更具一般性的冲突子句。也就是说，根据某个环生成的冲突子句，不仅要使得 SAT/SMT 避免在接下来的搜索中重复寻找相同的环，还要能避免搜索“等价”的环。具体方法将在“关键问题三研究方案”中论述。

关键问题三研究方案：采用“语义等价类”思想缩减程序状态空间

为了应对程序状态空间爆炸带来的挑战，我们借鉴多处理器并发程序断言验证中的动态偏序约简（DPOR）技术，首先针对各个事务一致性模型定义数据库系统执行历史之间的等价关系，然后设计搜索算法避免重复遍历等价的系统执行

历史，最后将该搜索算法与 SAT/SMT 搜索框架进行深度整合。

针对事务一致性，我们计划采用 SO-WR 等价关系，也就是说具有相同读写关系的执行历史是等价的。为了将 DPOR 搜索算法与 SAT/SMT 搜索框架进行整合，我们计划在事务一致性理论求解器的冲突子句生成模块中加入执行历史等价性测试功能，以便生成更具一般性的冲突子句。

实验手段

为了检验验证技术的高效性以及原型验证工具的实用性与可扩展性，我们计划选取各类数据库并采用多种基准测试集进行充分实验。

对于数据库系统执行历史验证问题，我们选取 MySQL、PostgreSQL、YugabyteDB、CockroachDB、MariaDB、MongoDB、Dgraph、TDSQL、TiDB 等国内外知名数据库进行测试。我们采用 TPC-C、RuBiS、C-Twitter 等经典基准测试集以及自定义基准测试集作为数据库负载。除了性能测试之外，我们还将考察验证工具的有效性：能否复现各类数据库的已知 bugs、能否找出各类数据库的未知 bugs。

对于客户程序断言验证问题以及客户程序健壮性问题，我们计划采用 TPC-C、RuBis、C-Twitter、YCSB+T、SmallBank、Banking、Auction、Courseware 等经典基准测试集。此外，我们还将从各类事务型数据库的真实使用场景中提取自定义基准测试集。

研究思路的可行性

本项目计划采用“深度整合 SMT”的思想，研究数据库系统中的事务一致性验证问题，进一步提升现有验证技术与工具的性能与可扩展性。根据前文对相关工作的论述，“深度整合 SMT”是解决一致性验证问题的一种技术趋势，而且已经在多处理器领域下的并发程序断言验证问题中表现出了显著的性能优势。该研究思路切合领域发展，具备良好的可行性基础。

研究方案的可行性

针对三个关键科学问题，我们都确定了基本的研究方案。这些研究方案综合运用了多处理器领域中的内存一致性验证技术、形式化方法领域中的 SAT/SMT 理论与动态偏序约简技术、程序设计语言领域中的程序操作语义以及数据库领域

中的事务一致性理论等多项理论与技术，有大量扎实的相关工作，具备实际的可操作性。

4、本项目的特色与创新之处

本项目计划采用“深度整合 SMT”的思想，研究数据库系统中的事务一致性验证问题，进一步提升现有验证技术与工具的性能与可扩展性，拟在如下三个方面形成特色与创新：

（1） 设计专门的事务一致性理论，通过与 SAT/SMT 搜索框架深度整合实现高效的理论求解器；以此为基础设计高效的数据库系统执行历史验证算法。

（2） 为面向事务型数据库的客户程序（包含 SQL 语句与复杂的语言结构）设计精确的、与事务一致性理论相适应的一阶谓词逻辑公式编码方案；以此为基础设计高效的客户程序断言验证算法。

（3） 为面向事务型数据库的客户程序设计一种与事务一致性理论相适应的、基于动态偏序约简技术的状态空间缩减策略；以此为基础设计高效的客户程序健壮性验证算法。

5、年度研究计划及预期研究成果

年度研究计划

（1） 2024 年 7 月--2025 年 6 月：研究数据库系统执行历史验证问题

以数据库系统单个执行历史的公理化语义为基础，以 SAT/SMT 求解策略为框架，设计面向数据库系统执行历史验证问题的事务一致性理论及其求解器。重点解决一致性理论深度整合难点，形成解决问题的整体架构、关键技术、核心算法等。主要考虑 SER、SI 两种经典的事务一致性模型。

（2） 2025 年 7 月—2026 年 6 月：研究客户程序断言验证问题

以事务一致性模型的公理语义与客户程序（包含 SQL 语句）的操作语义为基础，以 SAT/SMT 求解策略为框架，设计面向客户程序断言验证问题的事务一致性理论及其求解器。重点解决客户程序编码与一致性理论深度整合难点，形成

解决问题的整体架构、关键技术、核心算法等。主要考虑 SER、SI、CC 三种经典的事务一致性模型。

（3） 2026 年 7 月—2027 年 6 月：研究客户程序健壮性验证问题

以事务一致性模型的公理语义与客户程序（包含 SQL 语句）的操作语义为基础，以 SAT/SMT 求解策略为框架，设计面向客户程序健壮性验证问题的事务一致性理论及其求解器。重点解决客户编码、一致性理论深度整合与“状态空间爆炸”难点，形成解决问题的整体架构、关键技术、核心算法等。主要考虑 SER、SI、CC 三种经典的事务一致性模型。

此外，将持续开发、完善自动验证原型系统，开展基于大规模基准测试集的实例研究与应用验证。与产业界沟通合作，实施技术落地与成果转化。

预期研究成果

预期研究成果的表现形式包括高质量的研究论文、可执行的软件系统、技术发明专利等，具体如下：

- 在国内外高水平学术会议（如 VLDB、OOPSLA、FM 等）与学术期刊（如 FMSD、TPDS、软件学报等）上发表 3 篇高质量研究论文；
- 为事务一致性验证问题开发高效的自动验证原型工具
- 以上述技术与系统为核心内容提交一项发明专利申请书

二、研究基础和工作条件

1、研究基础

（与本项目相关的研究工作积累和已取得的研究工作成绩）

本项目团队近年来在分布式数据一致性与数据库事务一致性理论与技术领域开展了深入、系统的研究工作，研究成果发表在国内外重要期刊与会议上，包括 VLDB、SIGMOD、PODC、USENIX ATC、IEEE TC、IEEE TPDS、OPODIS、SRDS、软件学报、JCST 等，为本项目的研究工作奠定了良好的基础。下面介绍三份与本项目关系最为密切的研究工作。

（1） 基于 SAT/SMT 的数据库系统执行历史 SI 验证技术（VLDB'2023）

针对数据库系统执行历史验证问题，我们提出了一种针对 SI 的检测算法，并实现了原型验证工具 PolySI。检测算法将 SI 验证问题转化为在 Generalized Polygraphs 上找环的图论问题，并将其编码成命题逻辑公式，在经过关键的剪枝操作后，借助 MonoSAT 进行求解。我们证明了该算法的可靠性与完备性。大规模实验表明，PolySI 的性能大幅超越现有同类工具，并具有优异的可扩展性。相关工作已被 CCF A 类会议 International Conference on Very Large Databases（VLDB'2023）录用，具体信息如下：

Kaile Huang, Si Liu, Zhenge Chen, Hengfeng Wei, David Basin, Haixiang Li, Anqun Pan. *Efficient Black-box Checking of Snapshot Isolation in Databases*. Proc. VLDB Endow. Volumn 16, No. 6 (VLDB; April 2023), 1264–1276.

（2） 混合事务一致性协议与原型系统的设计与实现（USENIX ATC'2021）

我们实现了首个可扩展的、容错的且融合了因果一致性与强一致性的分布式事务型存储系统，称为 UniStore。UniStore 允许用户为每个事务指定一致性水平：强一致性事务或者因果一致性事务。我们严格证明了 UniStore 所用协议的正确性。我们在 Amazon EC2 上部署了 UniStore 系统，并使用 microbenchmark 与 RUBiS benchmark 对其进行评估。实验表明，UniStore 具有优异的性能与良好的接近线性的可扩展性。相关工作发表于 CCF A 类会议 USENIX Annual Technical Conference（USENIX ATC'2021），具体信息如下：

Manuel Bravo, Alexey Gotsman, Borja de Régil, Hengfeng Wei. *UniStore: A*

Fault-tolerant Marriage of Causal and Strong Consistency. The 2021 USENIX Annual Technical Conference (USENIX ATC), July, 2021.

（3） 复制数据类型规约框架的扩展与实证研究（SRDS'2020）

在分布式数据一致性领域，Burckhardt 等人基于可见（Visibility）关系与仲裁（Arbitration）关系提出了一种适用于最终一致性的复制数据类型规约框架，称为 (VIS, AR) 框架。然而，该框架有两点不足。第一，AR 要求所有操作构成全序，因此该框架无法刻画不收敛的一致性模型；第二，VIS 忽略了操作的返回值，因此该框架无法刻画那些需要观察到可见操作的返回值的一致性模型。通过将 AR 放松为偏序关系并引入用于限制返回值的 V 函数，我们将其扩展为 (VIS, AR, V) 框架，以涵盖更广泛的一致性模型，并揭示新的一致性模型。我们选取 MongoDB 数据库、GSP（Global Sequence Protocol）协议、RA- Linearizability 规约论证了新框架的实用性[2]。相关工作发表于 CCF B 类会议 IEEE Symposium on Reliable Distributed Systems（SRDS'2020），具体信息如下：

Xue Jiang, Hengfeng Wei, Yu Huang. *A Generic Specification Framework for Weakly Consistent Replicated Data Types*. In the Proceeding of the 39th International Symposium on Reliable Distributed Systems (SRDS), Sep. 2020.

2、工作条件

（包括已具备的实验条件，尚缺少的实验条件和拟解决的途径）

本项目依托南京大学计算机软件新技术国家重点实验室开展，已具备良好的研究与实验环境。

本项目涉及大量计算密集型任务，需要使用高性能服务器资源。目前，南京大学计算机软件新技术国家重点实验室拥有各类高性能服务器 600 余台套，各设备 24 小时不间断工作。此外，项目申请人所在学院也向各课题组提供高性能服务器资源租用服务。

另外，本项目需要在广域分布环境下部署多种类型的分布式数据库系统，计划租用腾讯云服务器。

3、个人简介

（包括申请人的教育经历和工作经历，在国内外学术组织、刊物及国际性学术会议任职情况，近期已发表与本项目有关的主要论著目录和获得学术奖励情况，正在承担、参加或完成的科研项目情况等）

[2] 其中，与 GSP 协议与 RA- Linearizability 规约相关的实证研究工作见于 TPDS'2023 期刊扩展版本论文，目前处于审稿阶段。

（1）申请人简介，包括受教育经历（从大学本科开始）和研究工作经历

魏恒峰，南京大学，软件学院，副研究员

教育经历：
(1) 2005-09 至 2009-06, 南京大学, 计算机科学与技术, 学士
(2) 2009-09 至 2016-12, 南京大学, 计算机软件与理论, 博士

科研与学术工作经历：
(1) 2017-01 至 2020-08, 南京大学, 计算机科学与技术系, 助理研究员
(2) 2020-09 至 今, 南京大学, 软件学院, 副研究员

（2）在国内外学术组织、刊物及国际性学术会议任职情况

无任职情况

曾担任 TCS（*Theoretical Computer Science*）、JPDC（*Journal of Parallel and Distributed Computing*）、JSEP（*Journal of Software: Evolution and Process*）等刊物评审

（3）近期已发表与本项目有关的代表性论著：按照以下顺序列出全部已经公开发表的论著目录：①近 5 年内发表的 3 篇代表性论著；②近 5 年内发表的其他论著；③5 年以外的代表性论著。上述论著目录均应按年份降序排列，要详细列出所有作者、论著题目、期刊名称或出版社名称、年、卷（期）、起止页码等。

① 近 5 年内发表的 3 篇代表性论著

Si Liu, Luca Multazzu, Hengfeng Wei, David Basin. *NOC-NOC: Towards Performance-optimal Distributed Transactions*. International Conference on Management of Data (SIGMOD; June 2024), Volume 2, Issue 1, Article No.: 9, pp 1-25.

Kaile Huang, Si Liu, Zhenge Chen, Hengfeng Wei, David Basin, Haixiang Li, Anqun Pan. *Efficient Black-box Checking of Snapshot Isolation in Databases.* Proc. VLDB Endow. Volume 16, No. 6 (VLDB), April, 2023, 1264-1276.

Manuel Bravo, Alexey Gotsman, Borja de Régil, Hengfeng Wei. *UniStore: A Fault-tolerant Marriage of Causal and Strong Consistency*. The 2021 USENIX Annual Technical Conference (USENIX ATC), July, 2021, 923-937.

② 近 5 年内发表的其他论著

Xue Jiang, Hengfeng Wei, Yu Huang. *Tunable Causal Consistency: Specification and Implementation*. The 28th IEEE International Conference on Parallel and Distributed Systems (ICPADS), Jan. 2023, 169-176.

Yi Huang, Hengfeng Wei. *Incremental Causal Consistency Checking for Read-Write Memory Histories*. The 13th Asia-Pacific Symposium on Internetware (Internetware), June 11-12, 2022, 181-191.

Hongrong Ouyang, Hengfeng Wei, Yu Huang, Haixiang Li, Anqun Pan. *Checking Causal Consistency of MongoDB*. Journal of Computer Science and Technology (JCST), 37(1):128-146, Jan. 2022, 128-146.

Kaile Huang, Hengfeng Wei, Yu Huang, Haixiang Li, Anqun Pan. *Brief Announcement: Byz-GentleRain: An Efficient Byzantine Fault-tolerant Causal Consistency Protocol*. The 23rd International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS), Nov, 2021, 495-499.

Yuqi Zhang, Hengfeng Wei, Yu Huang. *Remove-Win: a Design Framework for Conflict-free Replicated Data Types*. The 27th IEEE International Conference on Parallel and Distributed Systems (ICPADS), 2021, 607-614.

Xiaosong Gu, Hengfeng Wei, Lei Qiao, Yu Huang. *Raft with Out-of-Order Executions (in Chinese)*. Journal of Software (JOS), 32(6):1748-1778, 2021.

Hongrong Ouyang, Hengfeng Wei, Yu Huang. *Checking Causal Consistency of MongoDB*. The 12th Asia-Pacific Symposium on Internetware (Internetware), May 12-14. 2021, 209-216.

Lingzhi Ouyang, Yu Huang, Hengfeng Wei, Jian Lu. *Achieving Probabilistic Atomicity with Well-Bounded Staleness and Low Read Latency in Distributed Datastores*. IEEE Transactions on Parallel and Distributed Systems (TPDS), 32(4):815-829, Apr. 2021.

Hengfeng Wei, Ruize Tang, Yu Huang, Jian Lu. *Jupiter Made Abstract, and Then Refined*. Journal of Computer Science and Technology (JCST), 35(6):1343-1364, Dec.

2020.

Xue Jiang, Hengfeng Wei, Yu Huang. *A Generic Specification Framework for Weakly Consistent Replicated Data Types*. In the Proceeding of the 39th International Symposium on Reliable Distributed Systems (SRDS), Sep. 2020, 143-154.

Kaile Huang, Yu Huang, Hengfeng Wei. *Fine-grained Analysis on Fast Implementations of Distributed Multi-writer Atomic Registers*. In the Proceeding of the ACM Symposium on Principles of Distributed Computing (PODC), Aug. 2020, 200-209.

Xingchen Yi, Hengfeng Wei, Yu Huang, Lei Qiao, Jian Lu. *TPaxos in PaxosStore: Derivation, Specification and Refinement (in Chinese).* Journal of Software (JOS), 31(8):2336-2361, 2020.

Ye Ji, Hengfeng Wei, Yu Huang, Jian Lu. *Specifying and Verifying CRDT Protocols Using TLA+ (in Chinese)*. Journal of Software (JOS), 31(5):1332-1352, 2020.

③ 5 年以外的代表性论著

Hengfeng Wei, Yu Huang, and Jian Lu. *Specification and Implementation of Replicated List: the Jupiter Protocol Revisited*. The 22nd International Conference on Principles of Distributed Systems (OPODIS), 2018, 12:1-12:16.

Hengfeng Wei, Yu Huang, and Jian Lu. *Probabilistically-Atomic 2-Atomicity: Enabling Almost Strong Consistency in Distributed Storage Systems*. IEEE Trans. Comput. (TC), 66(3), Mar. 2017, pp. 502-514.

Hengfeng Wei, Yu Huang, and Jian Lu. *Parameterized and Runtime-Tunable Snapshot Isolation in Distributed Transactional Key-Value Stores*. The IEEE 36th Symposium on Reliable Distributed Systems (SRDS), Sept. 2017, pp. 21-33.

Hengfeng Wei, Marzio De Biasi, Yu Huang, Jiannong Cao, and Jian Lu. *Verifying Pipelined-RAM Consistency over ReadWrite Traces of Data Replicas*. IEEE Trans. Parallel Distrib. Syst. (TPDS), 27(5), May 2016, pp. 1511-1523.

（4）论著之外的代表性研究成果：包括学术奖励、会议特邀学术报告、授权发明专利等，其中学术奖励须列出全部受奖人员、获奖项目名称、奖励机构、奖励类别、奖励等级、颁奖年份，会议特邀学术报告须列出报告人、报告名称、会议名称、会议地址、会议时间，授权发明专利须列出全部发明人、专利名称、授权时间、国别、专利号。按照以下顺序列出论著之外的代表性研究成果：①近 5 年内发表的 5 项代表性成果；②近 5 年内发表的其他成果；③5 年以外的代表性成果。上述内容均按年份降序排列。

①近 5 年内发表的 5 项代表性成果

无

②近 5 年内发表的其他成果

无

③ 5 年以外的代表性成果

学术奖励：2017 年 CCF 优秀博士学位论文奖，中国计算机学会，2017 年

（5）正在承担、参加或完成的科研项目情况，要注明项目的名称和编号、经费额度、起止年月以及进展状态等

（1）国家自然科学基金委员会，青年科学基金项目，61702253，面向分布式系统的复制数据类型理论与技术研究，2018-01-01 至 2020-12-31，25 万元，结题，主持

（2）CCF 腾讯犀牛鸟基金项目，横向项目，CCF-Tencent RAGR20200124，分布式事务一致性模型与协议分析，2020-10 至 2021-12，15 万元，结题，主持

（3）腾讯犀牛鸟基金项目，横向项目，Tencent RAGR20200201，分布式事务一致性协议设计与验证技术研究，2021-12 至 2022-12，30 万元，结题，主持

（4）阿里巴巴（中国）有限公司（Alibaba Innovative Research），横向项目，128003031，基于 RDMA 和 NVM 加速的 ParallelRaft 协议设计与验证技术研究，2021-01 至 2022-01，49.042 万元，结题，参与

4、课题组其他主要成员自大学开始的学习经历、研究工作经历、主要论著、学术奖励以及承担科研项目情况简介

无。

三、经费申请使用说明

（本类项目经费管理实行包干制，不再编制项目预算，对项目直接经费、间接经费的使用进行简要说明。项目资助年限 3 年，省拨经费 15 万元）

直接经费使用说明：

主要包括云服务器租赁费用（用于部署分布式数据库系统）、设备更新费用、材料费（用于购置低值易耗品）、差旅费、会议费（用于参加国内外相关学术会议）、国际合作交流费、劳务费（用于支付参与项目的研究生的劳务费）、专家咨询费、图书资料费（用于购买中外文专业图书）。

间接经费使用说明：

主要包括项目管理费与绩效支出。

四、相关附件材料

1、项目申请人学位证书复印件或专业技术职称复印件；

2、身份证复印件；

3、近5年已发表与本项目有关的主要论著扫描件（不超过3篇）；

4、论著之外的代表性研究成果证明文件扫描件（不超过5项）；

5、其他相关附件材料。

说明：

1、各申报单位根据实际情况提供以上附件材料并上传系统，报主管部门审查；

2、论著、科技奖励、专利、会议报告等证明材料须作为附件上传。（如果篇幅过大，可以只提供封面、摘要、目录、版权页等扫描件）