92072427：一种基于关联事务融合和GPU三维加速的全同态数据库中间件

姓名：关荣鑫、齐冀、崔鹤鸣、姚信、陈仁海、王森、张弓
（香港大学）　　（华为）

www.huawei.com

项目编号：9451784

项目名称：基于加速器加速数据库架构、理论和算法研究合作项目

HUAWEI

专利申请合规Checklist

检查项类型	检查提示项	确认结果 [不涉及，是，否]
是否直接引用外来非公开信息	（1）Idea/交底书/专利申请文档不涉及在外部交流、技术合作、产品合作、采购等场景中从他人获取到的外来非公开信息，包括但不限于他人资料、仿真数据、测试结果、模型图、原理图、器件结构图、软件算法和其他非公开技术方案等；尤其注意不能引入带有他人标志或保密标识/水印的图片或资料；	否
是否以外非信息为基础进行改进	（2）Idea/交底书/专利申请文档不存在以外非信息的技术方案或技术构思为基础或主要设计参考而提出的改进方案，尤其注意不能在背景技术中引用外非	否
是否涉及外网引入	（3）Idea/交底书/专利申请文件不涉及从外部网站下载的有保密义务和/或使用限制的信息，如在线签署保密协议或者最终用户协议中限制分发或商业使用等	否
	（4）Idea/交底书/专利申请文件不涉及利用个人注册账号等途径从外部网站下载的非公开信息	否
职务成果归属	（5）Idea/交底书/专利申请文档所涉及的发明方案100%属于发明人在华为的职务成果，不涉及前雇主/前研究机构/学校的职务成果	是
其他	（6）Idea/交底书/专利申请文档等使用的技术材料中，不涉及来自于其他部门/其他产品线或关联公司，不包括来源不明的或无法确定是否属于外非信息的图片及资料	不涉及
	（7）Idea/交底书/专利申请文档不涉及其他公司的私有协议、私有接口、私有标准等。	不涉及
	（8）Idea/交底书/专利申请文档不涉及通过对未上市产品或软件通过拆解、反编译、反汇编等技术手段获取到的信息	不涉及
是否涉及出口管制	（9）如果Idea含有海外发明人，请确认技术是否达到出口管制规格。如是，是否已获得相应的出口许可?	否
发明人署名确认	（10）确保所有发明人都做出了实质性贡献。（包括：a. idea的初始创造者，关键技术路线设计者； b.《技术交底书》主笔人，技术实施例完整设计者；c.对idea的至少一个创新点的新颖性或创造性提供了有帮助的技术特征者。）	是

发明背景 - 云端数据库广泛应用于各行各业

- 在数据规模爆炸增长的时代，越来越多政府、企业，与个人使用云端数据库存储海量私有数据，享受云端数据库例如弹性扩展等诸多优点。

案例一：政务数据管理

政府存储海量政务数据（例如财政信息和公民身份信息）于云端。政府查询与更新财政信息和公民身份信息。

图1：陕西省财政厅采用华为云 ServiceStage 重构数字财政，实现政府财政信息的统一化存储与管理。

案例二：金融数据分析

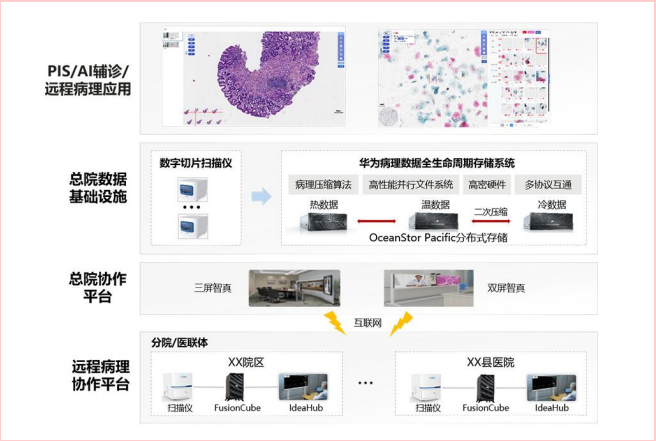
银行存储海量金融数据于云端。银行查询云端数据库进行融合数据分析和异构数据整合管理。

图2：工商银行采用华为 GaussDB (DWS) + Fusioninsight MRS 实现湖仓一体的融合数据分析解决方案。

案例三：医疗数据诊断

医院存储海量医疗数据于云端。医院查询病理数据以进行高效的阅片诊断。

图3：上海瑞金医院采用华为 OceanStor Pacific 分布式存储，实现数字化病理高性能调阅和数据缩减



发明背景 - 云端密态数据库极具商业价值

- 两大商业动机推动云端密态数据库发展
  - **数据机密性是数据库核心需求**：用户私有数据具有高度机密性。当部署于云端数据库时，一旦数据明文泄漏，恶意攻击者便可攫取机密数据进行牟利。因此，必须保证云端数据库的数据机密性才能释放其商业价值。
  - **法律法规强制保护数据机密性**：政府立法保护机密数据，例如中华人民共和国数据安全法。因此，必须保证云端数据库的数据机密性才能推动其合法商业落地。
- 工业界亟须云端数据库保障数据机密性，投入海量资源研发“密态数据库”。
  - 案例1：华为 GaussDB 迫切需要密态数据库解决方案，在云存储，金融，电子商务等领域实现数据的可算不可见（摘录自 GaussDB Full Encryption [VLDB’21]）。
  - 案例2：微软 Azure SQL AE 依靠 TEE 实现了基于 AES256 算法的数据透明加解密，但具有极大局限性，例如不支持密文层级直接运算，依赖 TEE 硬件环境（摘录自 Azure SQL Database Always Encrypted [SIGMOD’20]）。
  - 案例3：MySQL Enterprise Encryption 提供基于非对称加密算法的列级数据加密能力，但面临和 Azure SQL 数据库类似困境。
- 学术界发表众多关于密态数据库的顶级国际会议论文
  - CryptDB [SOSP’11], Senate [Security’21], Operon [VLDB’22], HE3DB (CCS’23) ...
  - 然而，现有研究存在三大痛点：
    - ◆性能低下：低吞吐、高时延，为了完成特定的密态数据库 CRUD 任务须要承受极高的密码学代价。
    - ◆安全易损：执行密文事务需进行密文解密、明文运算与再加密的过程，容易遭受数据泄露。
    - ◆架构单一：绑定单一特定数据库架构（例如 SQL 数据库），无法推广到其他多种数据库架构。

发明背景 - 全同态加密

- 全同态加密（Fully Homomorphic Encryption，FHE）是一类强安全的新型非对称加密算法。
- 全同态加密**可抵抗量子攻击**。它基于容错学习问题(LWE/RLWE)或格密码学最短向量问题(CVP)，对比当前主流的非量子安全的非对称加密算法（例如 RSA）更具安全性。
- 可以直接对全同态加密的密文进行**加法**和**乘法**运算，而无需经历传统加密算法的不安全且开销高的运算流程（即先解密，明文运算，再加密）。

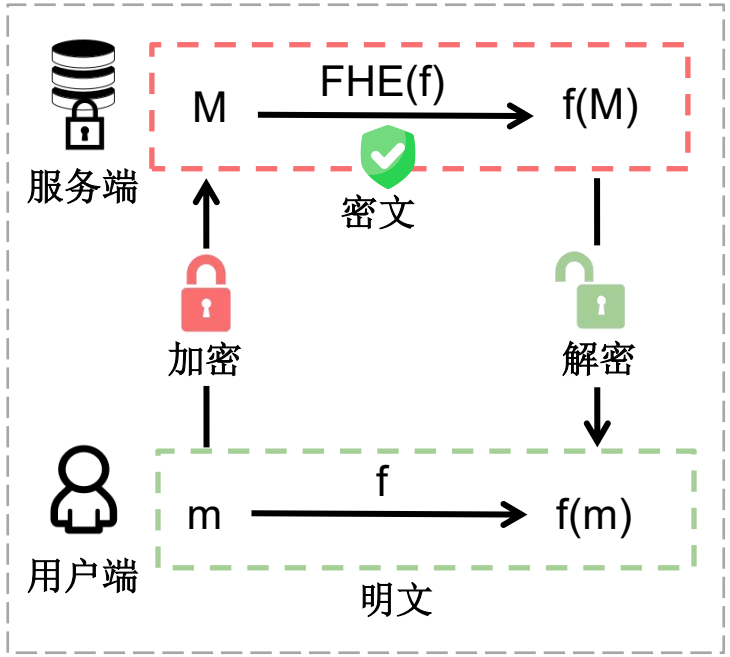


图4：全同态加密算法的运算流程

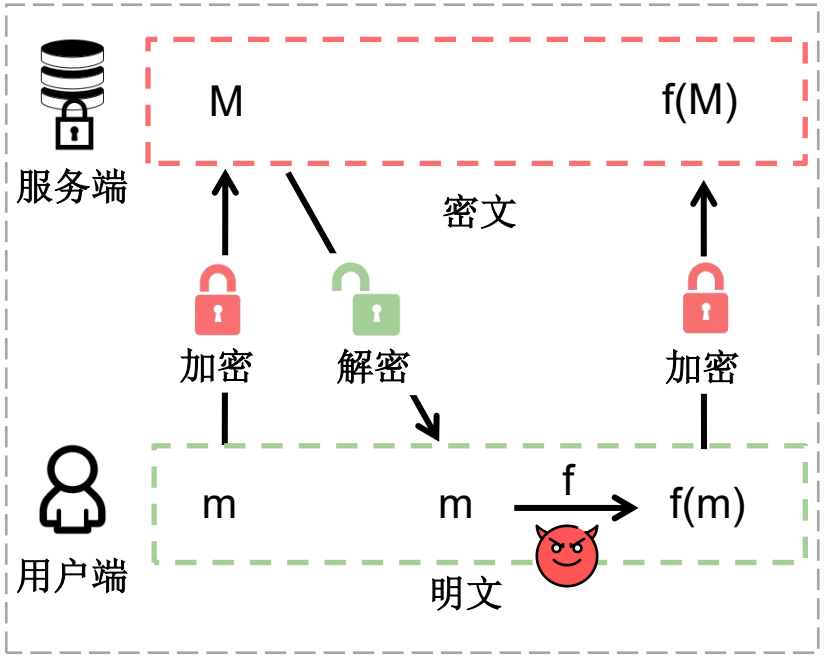


图5：传统加密算法的运算流程

发明背景 - 全同态加密

- 全同态加密发展迅猛，目前已迭代到第四代：
  - 第一代：基于格密码学，证明全同态加密的理论可行性，但性能不可用。代表作：[Gentry 2009]，[DGHV 2010]。
  - 第二代：基于LWE/RLWE，引入leveling schemes，支持整数算术运算，性能可用。代表作：[BFV 2012]，[BGV 2011]。
  - 第三代：基于第二代引入多种性能优化策略，例如避免relinearization和优化bootstrapping。代表作：[GSW 2013]，[FHEW 2014]。
  - 第四代：引入高效的密文取整操作，降低噪声增速，支持浮点数算术运算或布尔表达式运算。代表作：[CKKS 2017]，[TFHE 2016]。
- 本发明考虑纯软件的（不使用第三方可信硬件的）基于全同态加密的密态数据库场景，支持基于整数或浮点数算术运算的全同态加密算法。

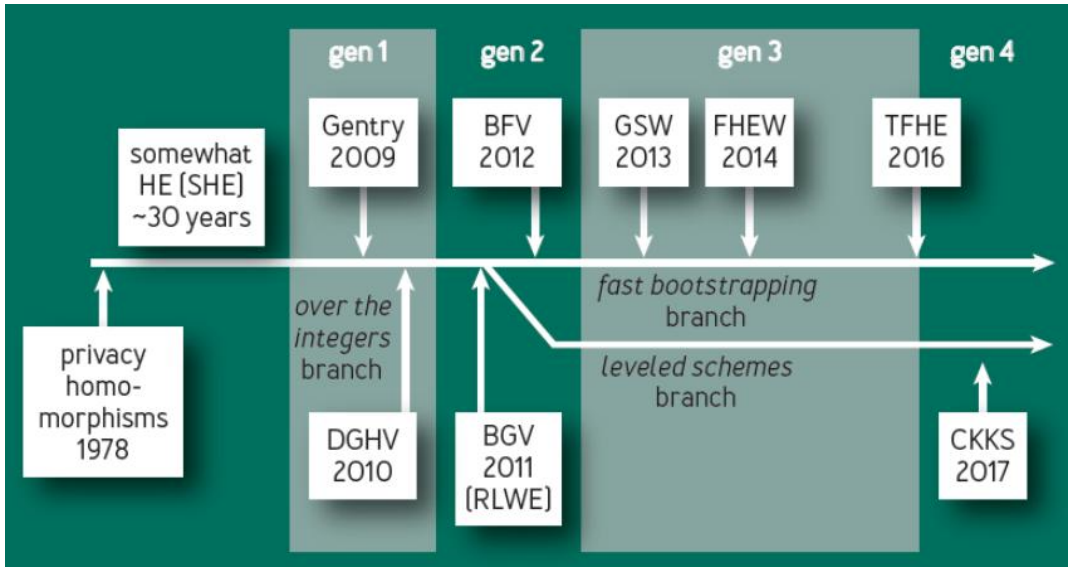


图6：全同态加密算法发展路线

发明背景 - 多元加密

- **技术问题：**全同态密文比较基于迭代算法（如图9所示），时间复杂度为O(MN)，其中M为参与比较密文差值的精度，N为参与比较的密文数量。这使得涉及全同态密文比较的数据库访问都会引发**全表线性扫描**，使各种索引结构（例如B+树）失效，严重损害数据库**可伸缩性**。
- **解决方案：**多元加密
  - **核心思想：**避免索引字段的全同态密文比较，同时兼顾保护所有字段的数据机密性。
  - 根据字段是否为索引字段，采取不同的(加密)存储策略：
    - 对于非索引字段，使用**全同态加密**（支持密文级别算术运算）
    - 对于索引字段，使用**确定性加密**（支持高效的等值查询）和**保序加密**（支持高效的范围查询）
- **局限性：**多元加密只能解决全同态数据库的可伸缩性问题，**不能降低全同态运算的高昂开销**。

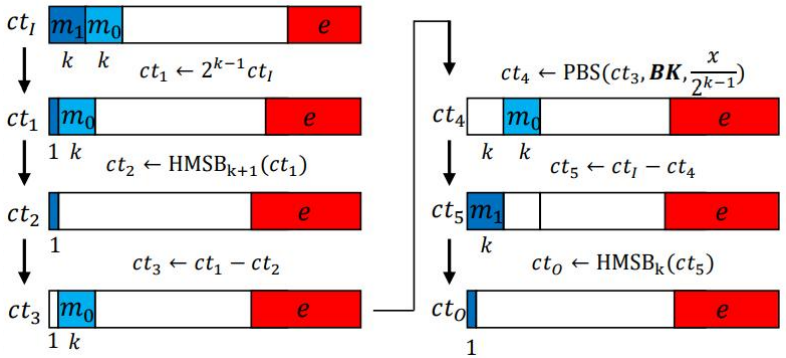


图7：全同态密文比较迭代算法

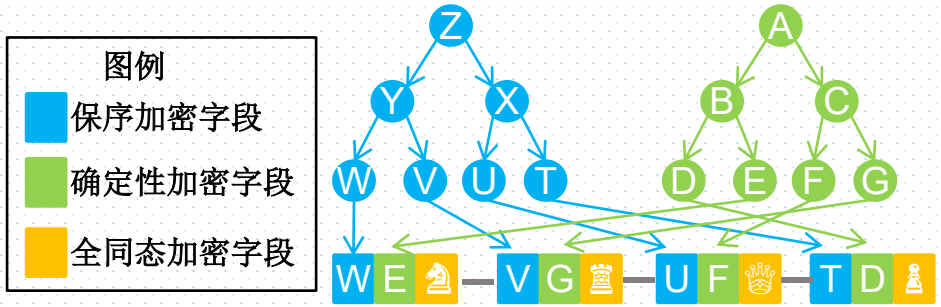


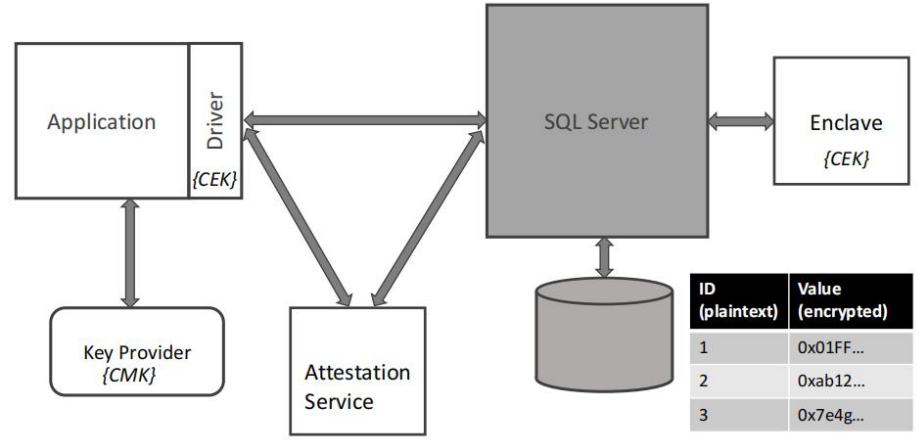
图8：采用多元加密的数据库B+树索引

现有工作 - 密态数据库

基于传统密码学的密态数据库

- 使用传统密码学算法加密关键数据
- 优点：
  - 增删查改性能优秀
- 缺点：
  - **不支持密文直接运算**，需要借助客户端或 TEE 完成数据加解密和明文运算

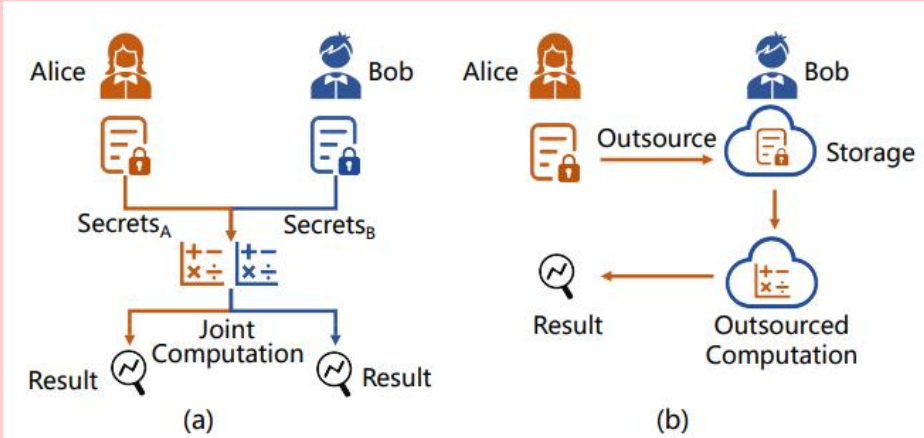
图9：Azure SQL AE 支持加密列级数据，并在 TEE 中提供明文比较和字符串匹配等复杂功能。



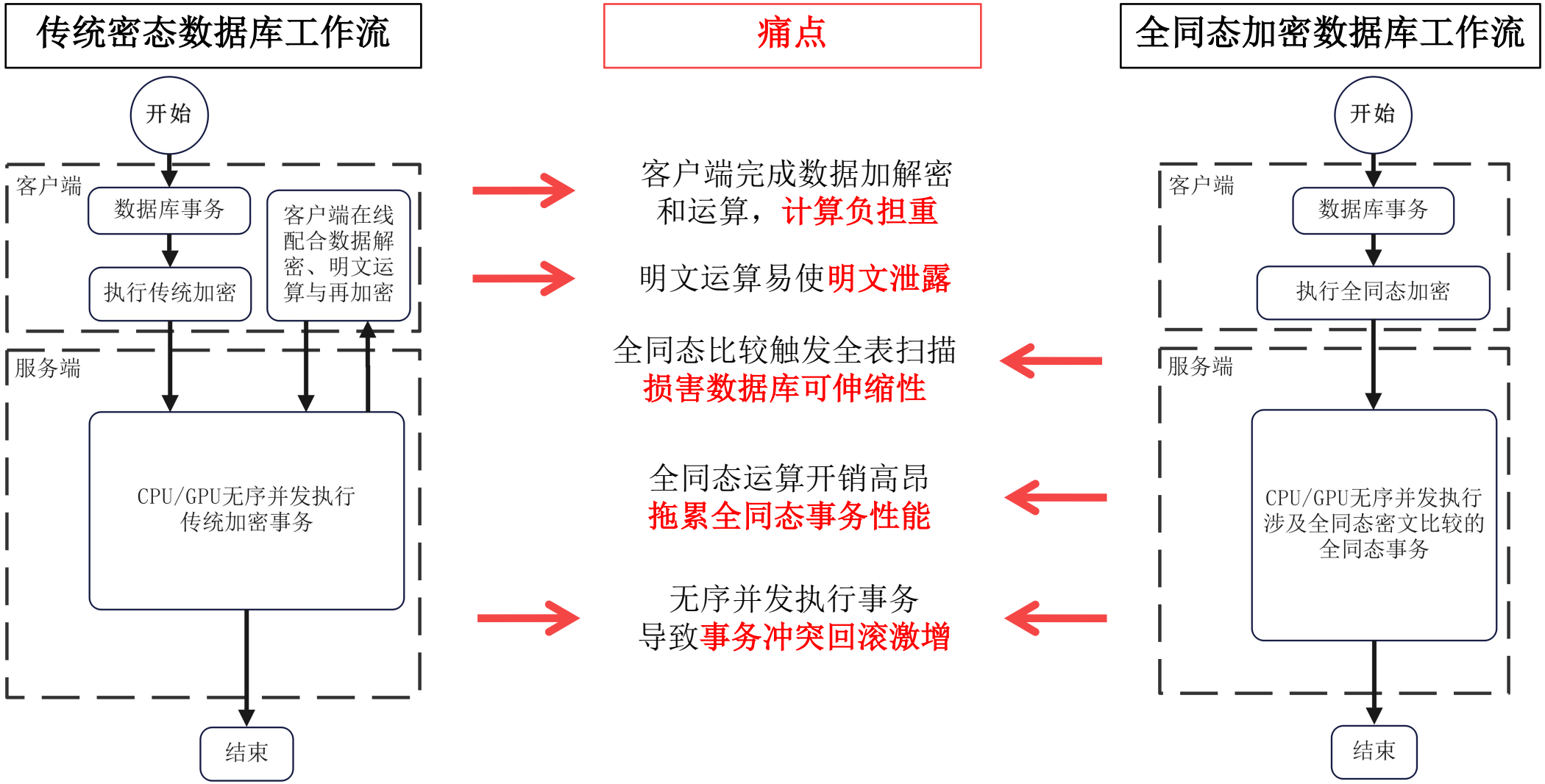
基于同态加密的密态数据库

- 使用同态密码学算法加密全表数据
- 优点：
  - 支持密文直接运算
- 缺点：
  - **性能极差**，数据库查询和修改操作均需进行全表扫描，且同态运算开销极大

图10：HE3DB 使用 TFHE 完成查询的谓词比较操作，使用 CKKS 完成聚合运算。



现有传统/全同态密态数据库工作流程与痛点



现有传统/全同态密态数据库痛点与解决方案

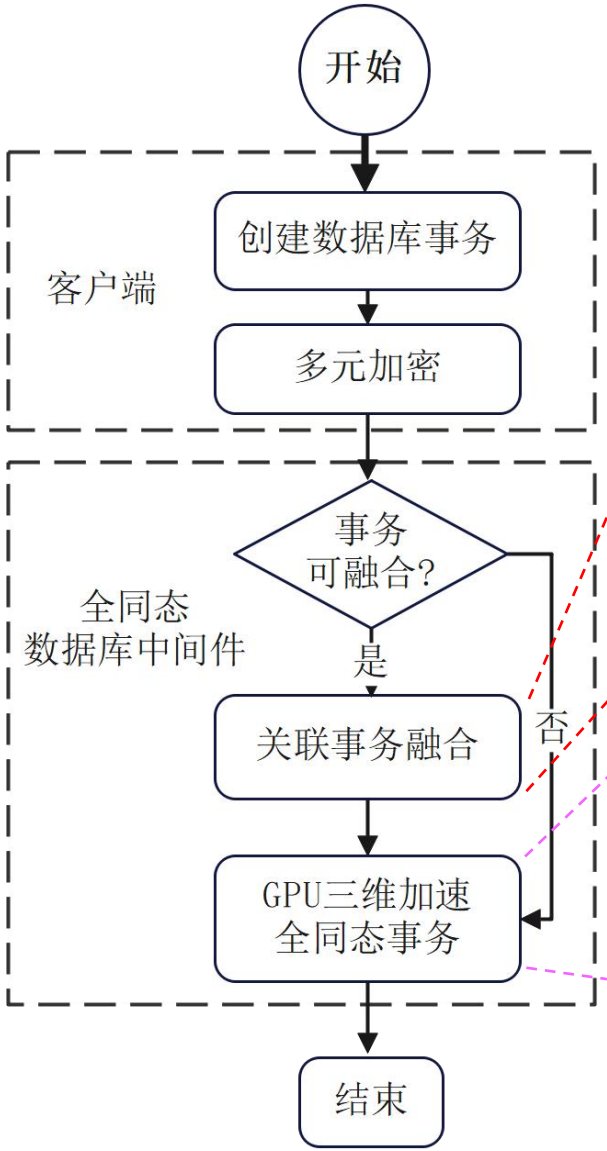
痛点	解决方案
客户端完成数据加解密和运算，**计算负担重**	**数据库中间件架构** 将数据加解密和运算转移到单独的中间件上 数据库专注数据存储与检索
明文运算易使**明文泄露**	**多元加密** 非索引数据采用全同态加密 索引数据采用确定性加密和保序加密
全同态比较触发全表扫描 **损害数据库可伸缩性**	
全同态运算开销高昂 **拖累全同态事务性能**	**关联事务融合（本专利保护点1）**
无序并发执行事务 导致**事务冲突回滚激增**	**GPU三维加速全同态事务（本专利保护点2）**

发明目标 - 安全、高效的全同态数据库中间件

- 本发明是首个采用关联事务融合和GPU三维加速的全同态数据库中间件。
  - **安全性**：所有数据端到端加密，全同态加密数据直接在密文层级运算，保护所有数据不被泄露与篡改
  - **高效性**：通过关联事务融合（保护点1）和GPU三维加速全同态事务（保护点2）取得最高19.3倍性能提升

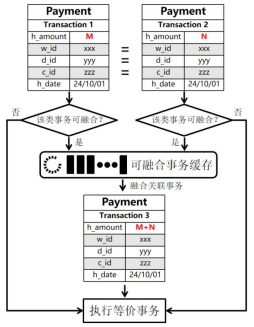
系统	数据机密性	密文可算性	高效性
MySQL	✗	✗	✓
Azure SQL AE [SIGMOD' 20]	✓	✗	✓
GaussDB [VLDB' 21]	✓	✗	✓
CryptDB [SOSP'11]	✓	✓	✗
HE3DB [CCS'23]	✓	✓	✗
本发明GECO	✓	✓	✓

本发明工作流程与实现目标总结



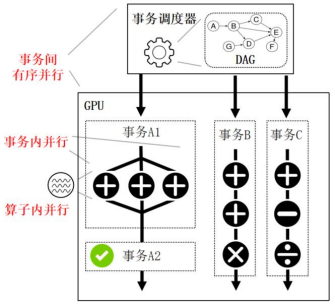
- **保护点1：关联事务融合**

实现目标：基于静态声明和动态融合相结合的方式，将多个关联事务融合为单一等价事务，通过减少全同态事务实际执行数量，从而减少极耗时的全同态运算次数，保证高效性。



- **保护点2：GPU三维加速全同态事务**

实现目标：从三个维度（事务间有序并行、事务内并行、算子内并行），使用GPU加速全同态事务执行，在提高并发度的同时消除事务并发冲突引发的回滚重试，保证高效性。



保护点1 - 关联事务融合

- **形式化定义：**在任意相同起始数据库状态下，对于多个同类事务（$T_1, T_2, ..., T_n$），存在一个同类事务$T_=$，使得 commit $T_=$ 后的数据库状态等价于以任意顺序 commit $T_1, T_2, ..., T_n$ 的数据库状态，则称 $T_1, T_2, ..., T_n$ 为关联事务，$T_=$ 为等价事务。

- **我们的观察：**在各类数据库应用中，关联事务广泛存在且高频执行。
  - 例如 TPC-C 的 Payment 事务和 SmallBank 的 WriteCheck 事务
  - 通过识别多个关联事务并融合传入参数，生成并执行单一等价事务，**显著降低全同态运算次数**并**减少事务并发冲突**。
  - 现有数据库未对关联事务进行识别与融合操作。

保护点1 - 关联事务融合

- **关联事务融合的形式化描述**
  - **关联事务静态声明**：
    1. 在事务编码阶段，开发者首先声明该类事务的可融合性。
    2. 对于可融合类型事务，开发者需声明其关联事务判定谓词与参数融合方法，以供在线融合过程调用。
  - **关联事务在线融合**：在事务动态执行阶段，根据事务可融合性对事务分流处理：
    1. 对于不可融合事务，直接执行。
    2. 对于各类可融合事务：
       I. 将预设时间窗口内（例如100ms）接收的事务请求放入缓冲区。
       II. 当时间窗口闭合或者缓冲区空间已满时：
           a. 基于该类事务的关联事务判定谓词，将缓冲区内所有事务请求划分为若干组关联事务。
           b. 基于该类事务的参数融合方法，为各组关联事务生成、执行并commit等价事务。
- **本保护点与现有方法的差异**
  - 识别并融合关联事务，**关联事务实际commit数量小于事务请求成功执行数量**
  - 关联事务commit后，**多个事务请求对应同一个实际commit事务ID**

保护点1 - 关联事务融合

- 示例：TPC-C Payment 关联事务融合

```
@mergeable(
  merge_predicate = {
    "w_id": equals,
    "d_id": equals,
    "c_id": equals
  }
  merge_method = {
    "h_amount": h_add
  }
)
def payment(h_amount, w_id, d_id, c_id):
  # 开始事务
  tx = DB.start_tx()
  # 获取顾客数据
  customer = get_customer(w_id, d_id, c_id)
  # 更新顾客数据
  customer.balance = h_sub(customer.balance, h_amount)
  customer.ytd_payment = h_add(customer.ytd_payment, h_amount)
  tx.execute(UPDATE_CUSTOMER_SQL, customer)
  # 更新其他数据
  ...
  # 提交事务
  tx.commit()
```

关联事务判定谓词

参数融合方法

图11：带融合元信息的TPC-C Payment事务部分实现代码

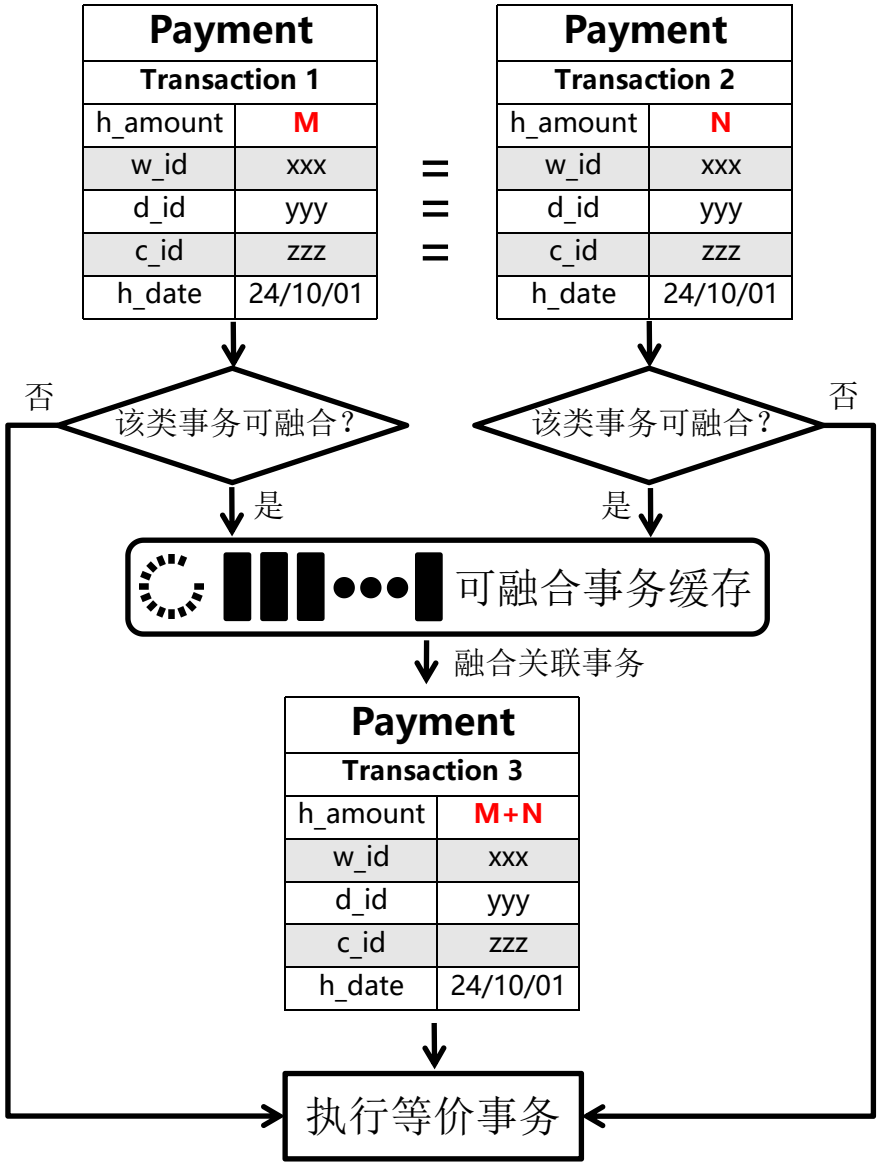


图12：关联事务在线融合

保护点1 – 关联事务融合

- 有益效果
  - 图14展示使用TPC-C测试SQL数据库的端到端吞吐，其中item表含10^6行记录。随机生成所有事务请求且设置50%为关联事务。GECO-M是保护点1的端到端性能结果。
  - 实验表明，保护点1使得本发明带来**最高2.3倍**吞吐提升和**最高17.4%**平均时延降低。
  - 更全面的数据参考有益效果 – 本发明实现更高吞吐量和更低平均时延

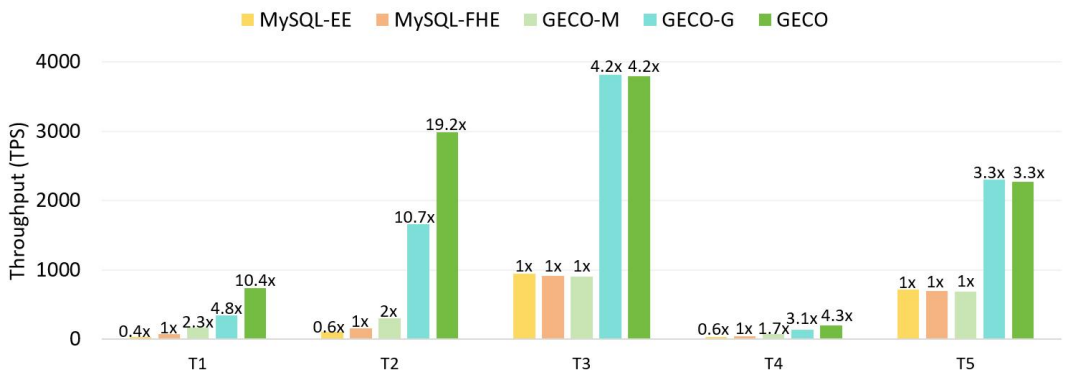


图13：基于TPC-C数据集的SQL密态数据库吞吐量对比

TPC-C	MySQL-EE		MySQL-FHE		GECO-M		GECO-G		GECO	
	tput	latency	tput	latency	tput	latency	tput	latency	tput	latency
T1: New Order	28	2130	71	960	165	886	342	933	737	855
T2: Payment	99	398	155	248	302	239	1663	255	2988	240
T3: Order Status	940	73	912	95	903	86	3817	99	3792	88
T4: Delivery	29	1887	45	1840	76	1574	138	1690	195	1361
T5: Stock Level	720	179	694	213	683	176	2304	219	2271	180

图14：基于TPC-C数据集的SQL密态数据库端到端性能

保护点1 – 关联事务融合

- **传统方案特征**
  - 实际事务commit数量始终等于客户事务请求成功执行数量
  - 同类事务具有一致的吞吐量和平均时延
- **本方案外显特征**
  - **外显特征1：第三方系统实际事务commit数量小于客户关联事务请求成功执行数量，且多个关联事务请求对应同一个commit事务ID**
  - **外显特征2：第三方系统关联事务比非关联事务具有更高吞吐和更低平均时延**
- **侵权检测方法**

1. 对于同类可融合事务，分别构造**一组关联事务请求 T_R** 和**一组非关联事务请求 T_N**，发送给被检测系统执行
2. 使用数据库监控工具拦截被检测系统的执行日志（例如 GaussDB 的 *gs_collector* 和 *gs_dump*，MySQL 的 *mysqlbinlog* 和 *MySQL Enterprise Monitor*），获取 T_R 和 T_N 对应的实际commit的事务ID、事务传入参数和事务数量。检查 ① T_R 对应的实际commit事务数量小于 T_R 的事务请求成功执行数量，且数据库状态等价于串行执行 T_R 中的所有事务请求的数据库状态。
3. 重复执行上述步骤 1，在客户端记录事务请求的事务ID、端到端吞吐量和平均时延。检查 ②多个事务请求对应同一个commit事务ID；③不论在任何运行环境下，T_R 的端到端吞吐量都显著高于 T_N，且 T_R 的端到端时延都显著低于 T_N
4. **如果被检测系统满足上述条件①②③，则该系统极有可能侵权**

保护点2 - GPU三维加速全同态事务

- GPU 加速全同态事务的现有方法
  - 方法一：GPU 并行加速全同态算子内部运算，但**同一事务不同算子仍然串行执行**
  - 方法二：GPU **无序并行**执行多个全同态事务，但由于并发度提高导致事务冲突回滚增加，**平均时延不降反增**
- 我们的观察
  - 在各类数据库应用中，普遍存在**事务内部语句无读/写集依赖关系**，这类语句**不必串行执行**。例如 TPC-C 的 Delivery 事务分别对十个 district 执行无读/写集依赖的查询与更新语句。
  - 在各类数据库应用中，普遍存在**多个事务具有读/写集依赖关系**，并发执行这类事务将导致**事务回滚与重试**。例如 SmallBank 的 Amalgamate事务，如果并发执行两个事务请求，参数分别为 (Alice, Bob) 和 (Bob, John)，则后执行的事务将回滚重试。

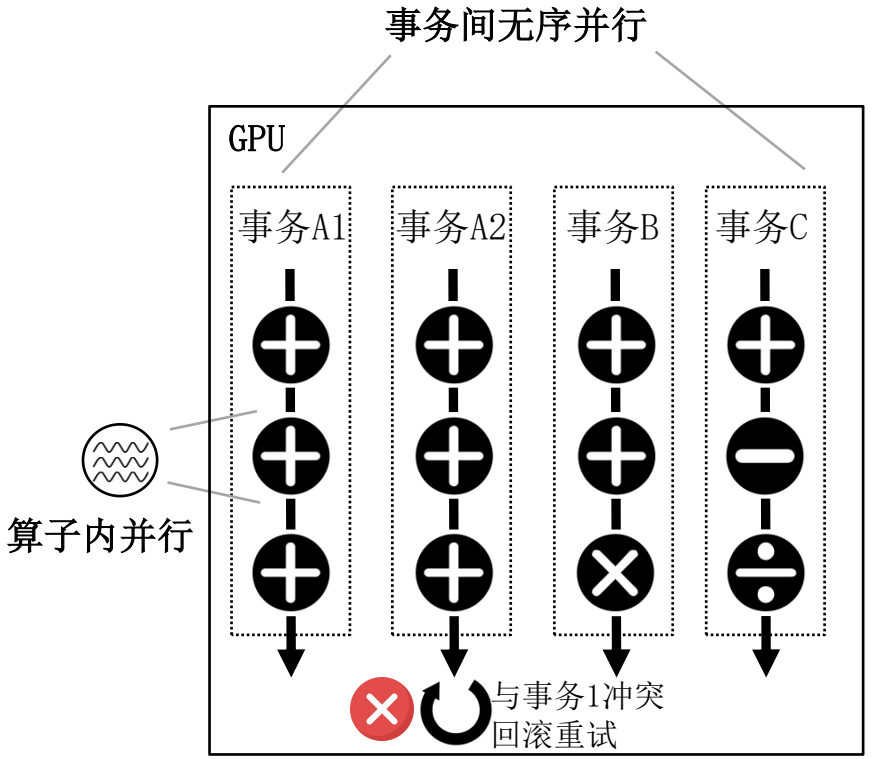


图15：GPU加速全同态事务现有方法示意图

保护点2 - GPU三维加速全同态事务

- **本专利解决方案：**使用GPU从三个维度加速全同态事务执行
  1. **算子内并行**：GPU多线程并行执行单个全同态算子内部运算
  2. **事务内并行**：GPU多线程并行执行单个事务内部多个无读/写集依赖关系的语句
  3. **事务间有序并行**：
     1. 事务调度器基于事务读/写集分析所有事务请求的读/写集依赖关系，依照事务请求的读/写集依赖关系和接收时序将事务请求编排为有向无环图（DAG）
     2. 事务调度器将多个无前序依赖的事务请求调度到多个GPU计算流中并行执行。
     3. 事务commit后，事务调度器将已commit事务移出DAG，重复上述步骤。

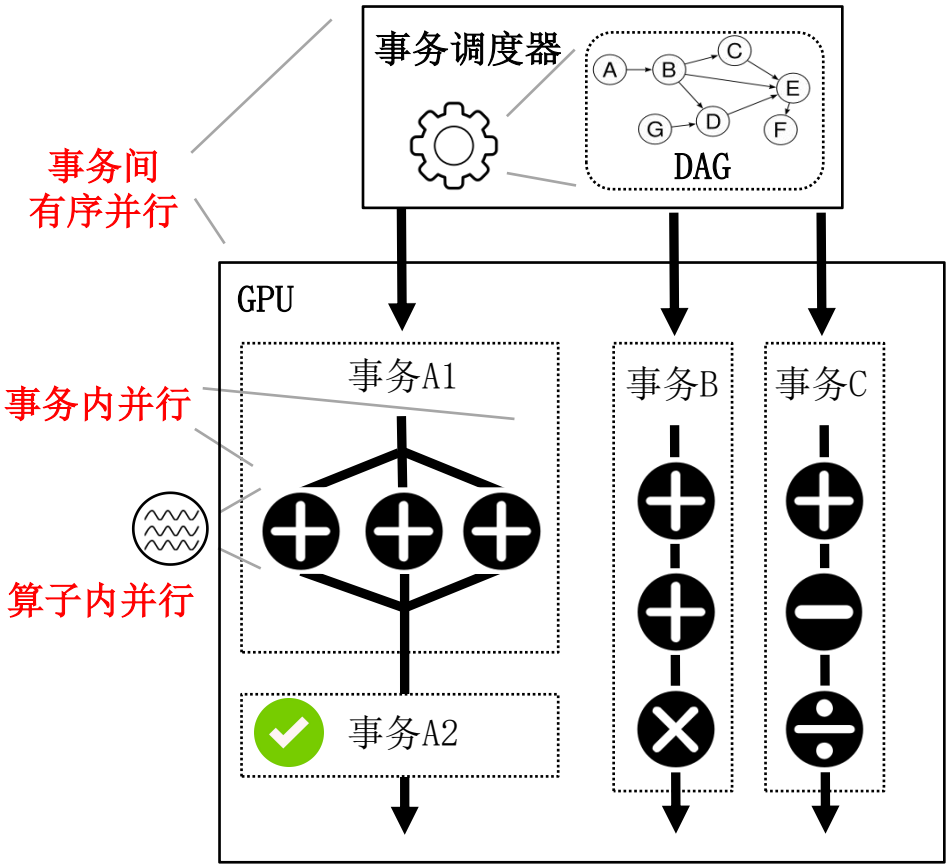


图16：GPU三维加速全同态事务协议

- **本保护点与现有方法的差异**
  - 引入**事务内并行**，提高并行度，降低平均时延
  - 引入事务的读/写集依赖分析和DAG调度，实现**事务间有序并行**，提高吞吐量和资源有效利用率

保护点2 - GPU三维加速全同态事务

- 有益效果
  - 图18展示使用TPC-C测试SQL数据库的端到端吞吐，其中item表含10^6行记录。随机生成所有事务请求且设置50%事务存在读/写集依赖关系。GECO-G是保护点2的端到端性能结果
  - 实验表明，GPU三维加速全同态事务（保护点2）可带来约**最高10.7倍**吞吐提升和**最高8.2%**平均时延降低
  - 更全面的数据参考有益效果 – 本发明实现更高吞吐量和更低平均时延

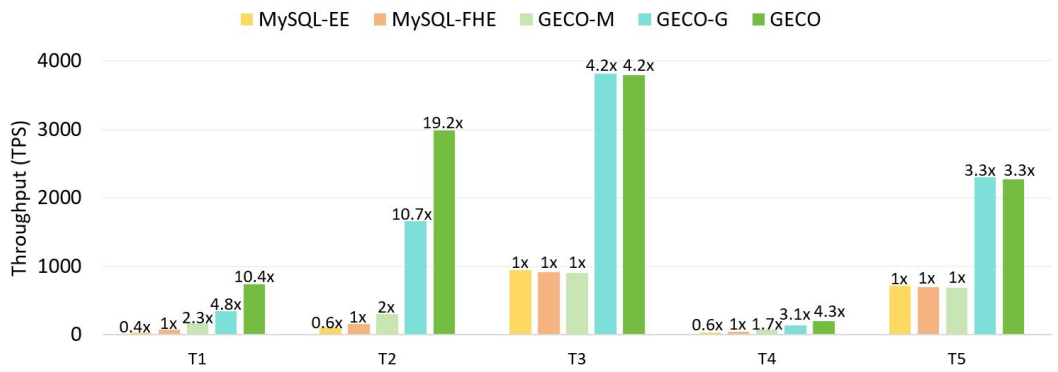


图17：基于TPC-C数据集的SQL密态数据库吞吐量对比

	MySQL-EE		MySQL-FHE		GECO-M		GECO-G		GECO	
TPC-C	tput	latency	tput	latency	tput	latency	tput	latency	tput	latency
T1: New Order	28	2130	71	960	165	886	342	933	737	855
T2: Payment	99	398	155	248	302	239	1663	255	2988	240
T3: Order Status	940	73	912	95	903	86	3817	99	3792	88
T4: Delivery	29	1887	45	1840	76	1574	138	1690	195	1361
T5: Stock Level	720	179	694	213	683	176	2304	219	2271	180

图18：基于TPC-C数据集的SQL密态数据库端到端性能

保护点2 - GPU三维加速全同态事务

- **传统方案特征**
  - 无论事务之间是否存在读/写集依赖关系，总使用多个SM并行执行多个事务
  - 执行存在读/写集依赖关系的多个事务时，发生多次回滚与重试
- **本方案外显特征**
  - **外显特征1：第三方系统使用GPU加速，仅使用单个SM(流多处理器)执行具有读/写集依赖关系的多个事务时，但使用多个SM执行无读/写集依赖关系的多个事务**
  - **外显特征2：第三方系统对于具有读/写集依赖关系的多个事务采用串行执行且不发生回滚**
- **侵权检测方法**
  1. 对于同类可融合事务，分别构造一组**具有读/写集依赖的事务请求T_D**和一组**无读/写集依赖事务请求T_I**，发送给被检测系统执行。
  2. 使用性能监测工具（例如nvidia-smi或perf）检查 ① T_D执行**只使用一个SM**，以及T_I执行**使用多个SM**。
  3. 使用数据库监控工具拦截被检测系统的执行日志，检查 ② T_D采用**串行执行**，T_I采用**并行执行**。
  4. 客户端监控事务请求执行情况，检查 ③ T_D的**吞吐量显著低于T_I**，**平均延迟显著高于T_I**，④ **T_D和T_I不发生冲突回滚**
  5. **如果被检测系统满足上述条件①②③④，则该系统极有可能侵权**

本发明技术方案实施例（对传统非密态数据库进行加密与加速）

- 实施例：使用本专利加速传统非密态云端数据库(例如MySQL和PostgreSQL)运行关键数据加密的银行转账业务
  - 银行账户数据库可以简化为三元组(账户ID，账户余额，最近操作时间)，其中账户余额是关键数据，使用全同态加密进行保护；其余数据例如账户ID与最近操作时间是非关键数据，使用明文存储或确定性加密。
  - 客户执行本专利编译完成的客户端程序，生成或获取用户密钥对，根据银行账户数据库的多元加密策略对银行转账事务传入参数进行加密，随后将全同态银行转账事务请求发送至中间件程序。
  - 中间件程序在云端部署，负责对银行转账关联事务的融合与使用GPU加速事务执行。
  - 中间件程序通过特化数据库驱动与各类云端数据库(例如MySQL和PostgreSQL)进行CRUD等操作。

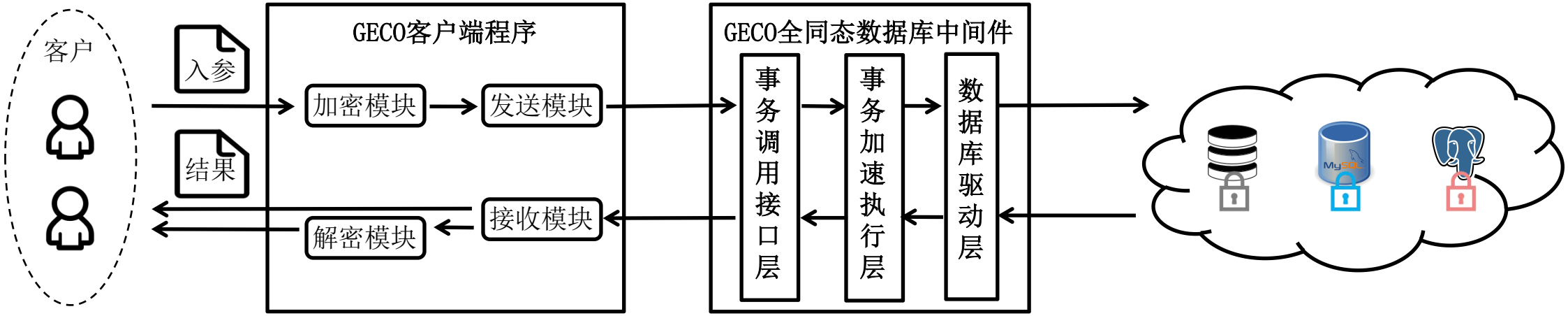


图19：本发明技术方案实施案例

本发明防侵权检测方法（总结）

- **防侵权核心方法**
  - 多次构造事务，采样事务执行端到端吞吐和时延指标，获取外显特征（保护点1，2）
  - 监控第三方系统硬件资源占用信息（保护点1，2）
- **本发明具有以下外显特征：**
  - 外显特征1：第三方系统实际事务commit数量**小于**客户关联事务请求成功执行数量，且**多个关联事务请求对应同一个commit事务ID** → 保护点1
  - 外显特征2：第三方系统关联事务比非关联事务具有**更高吞吐和更低平均时延** → 保护点1
  - 外显特征3：第三方系统使用GPU加速，使用**单个SM**(流多处理器)执行具有读/写集依赖关系的多个事务时，使用**多个SM**执行无读/写集依赖关系的多个事务 → 保护点2
  - 外显特征4：第三方系统对于具有读/写集依赖关系的多个事务采用**串行执行且不发生回滚** → 保护点2
- **若第三方系统声称满足本发明所有的发明目标，且同时满足上述所有外显特征，则该第三方系统极有可能侵权**

本发明防侵权检测方法（总结）

- **保护点1：关联事务融合**
  - **外显特征1：**第三方系统实际事务执行数量小于客户事务请求数量，且多个关联事务请求对应同一个commit事务ID
  - **外显特征2：**第三方系统可融合事务比不可融合事务具有更高吞吐和更低平均时延
  - **侵权检测方法：**
    1. 构造**关联事务请求T_R**和**非关联事务请求T_N**，发送给被检测系统执行
    2. （**白盒检测**）使用数据库监控工具**拦截被检测系统的执行日志**，获取T_R和T_N对应的实际commit的事务ID、事务传入参数和事务数量。检查T_R对应的实际commit事务数量是否小于T_R的事务请求数量，且数据库状态等价于串行执行T_R中的所有事务请求的数据库状态
    3. （**黑盒检测**）多次采样，检查多个关联事务请求对应同一个commit事务ID，检查T_R的端到端吞吐量是否显著高于T_N，且T_R的端到端时延是否显著低于T_N

本发明防侵权检测方法（总结）

- **保护点2：GPU三维加速全同态事务**
  - **外显特征3：**第三方系统使用GPU加速，使用**单个SM**(流多处理器)执行具有读/写集依赖关系的多个事务时，使用**多个SM**执行无读/写集依赖关系的多个事务
  - **外显特征4：**第三方系统对于具有读/写集依赖关系的多个事务采用**串行执行且不发生回滚**
  - **侵权检测方法：**
    1. 构造一组**具有读/写集依赖的事务请求T_D**和一组**无读/写集依赖事务请求T_I**，发送给被检测系统执行。
    2. （**白盒检测**）使用**性能监测工具**（例如nvidia-smi或perf）检查T_D执行是否**只使用一个SM**，以及T_I执行**使用多个SM**。
    3. （**白盒检测**）使用**数据库监控工具**拦截被检测系统的执行日志，检查T_D是否采用**串行执行**，T_I是否采用**并行执行**。
    4. （**黑盒检测**）客户端**监控处理事务请求的端到端吞吐量和时延**，检查T_D的**吞吐量是否显著低于T_I**，**平均延迟是否显著高于T_I**

有益效果

- **原型系统**：GECO是本发明的原型系统，GECO-M只实现关联事务融合（保护点1），GECO-G只实现GPU三维加速全同态事务（保护点2）。
- **基线系统**：本发明对比了 MySQL-EE 和 MySQL-FHE
  - MySQL-EE (MySQL Enterprise Encryption) 是一个广泛商用的SQL传统密态数据库，采用传统加密算法（例如 RSA）。
  - MySQL-FHE是一个基于MySQL和全同态加密的基线SQL数据库，未实现关联事务融合和GPU三维加速全同态事务。
- **测试基准**：本发明使用 TPC-C 测试 SQL 数据库场景下各系统的性能（MySQL-EE，MySQL-FHE，GECO-M，GECO-G和GECO）
- **实验问题**
  - 高效性：对比现有系统，本发明能否取得更低平均时延与更高吞吐量？
  - 可伸缩性：本发明能否高效扩展到更大数据集？
- **天然满足的目标**
  - 数据机密性：在传输、存储和计算过程中，非索引数据使用全同态加密，索引数据使用确定性加密和保序加密，保障明文不泄露与不被篡改。

有益效果 - 本发明实现更高吞吐量和更低平均时延

- 图 20 和图 21 使用 TPC-C 对比了 SQL 数据库场景下各系统性能，其中 item 表的记录数为 10^6。所有事务请求均为随机生成，设置50%事务为关联事务。

- 可以看出：对比基线全同态数据库 MySQL-FHE，本发明带来约3.3-19.3倍吞吐量提升和最高36%平均时延降低
  - （保护点1）关联事务融合可带来约最高2.3倍吞吐提升和最高17.4%平均时延降低
  - （保护点2）GPU三维加速全同态事务可带来约最高10.7倍吞吐提升和最高8.2%平均时延降低

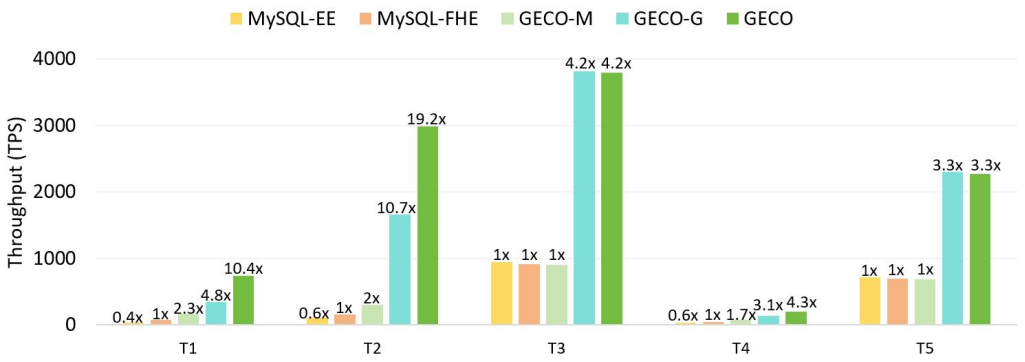


图20：基于TPC-C数据集的SQL密态数据库吞吐量对比

TPC-C	MySQL-EE		MySQL-FHE		GECO-M		GECO-G		GECO	
	tput	latency	tput	latency	tput	latency	tput	latency	tput	latency
T1: New Order	28	2130	71	960	165	886	342	933	737	855
T2: Payment	99	398	155	248	302	239	1663	255	2988	240
T3: Order Status	940	73	912	95	903	86	3817	99	3792	88
T4: Delivery	29	1887	45	1840	76	1574	138	1690	195	1361
T5: Stock Level	720	179	694	213	683	176	2304	219	2271	180

图21：基于TPC-C数据集的SQL密态数据库端到端性能

有益效果 - 本发明可高效伸缩

- 图 23 展示了 GECO 在 TPC-C 测试基准不同数据量下的端到端吞吐量变化。
- 可以看出：
  - 本发明**支持高效横向伸缩**：GECO 可在**数据量**显著增加的场景下，维持事务执行吞吐量基本稳定。
  - 本发明**支持高效纵向伸缩**：GECO 可在**数据库表结构**显著复杂的场景下，维持事务执行吞吐量基本稳定。

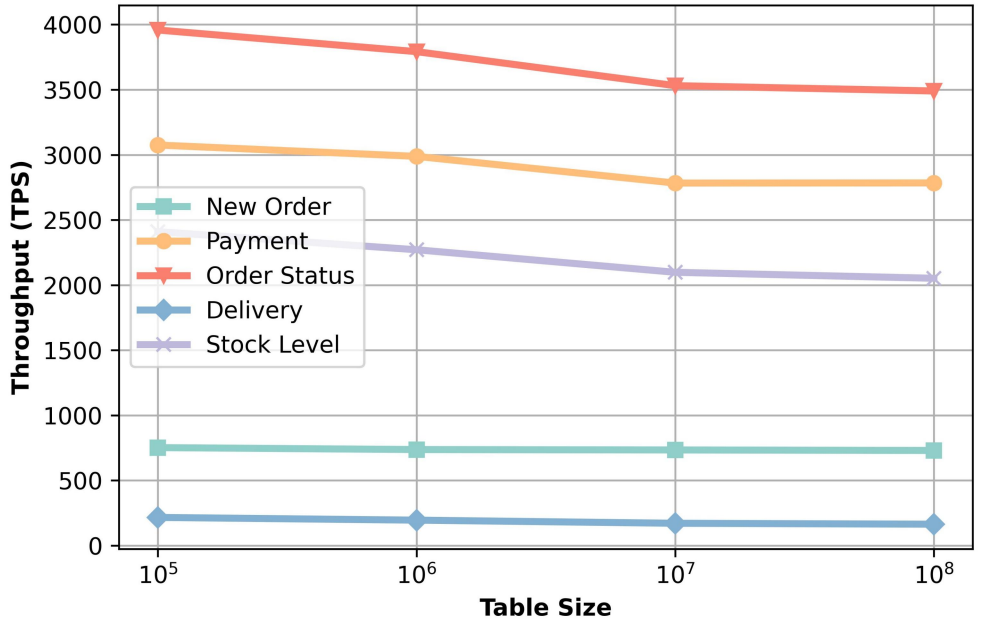


图22：GECO 在 TPC-C 不同数据量下的端到端吞吐量

结论：申请者认为本发明有重大专利价值

- 云端密态数据库目前广泛应用于金融、政务、电商等多个领域，蕴藏极高的**商业价值**和**落地潜力**。

- 现有系统**无法同时满足**事务执行**高效性**与**数据机密性**这两个核心目标，极大阻碍了来自不同行业的客户（例如政府或企业）采纳云端密态数据库。

- 本发明利用全同态加密数据库的**密文可算性**的特点，基于对全同态运算特征和事务关联性的观察，通过使用**关联事务融合**和**GPU三维加速数据库全同态事务**执行，并且基于**多元加密**和**中间件**的系统设计，实现了**安全、高性能**的全同态数据库。

- 本发明有益于华为云、GaussDB、数据存储产品线等部门，**作为现有数据库的有力补充**，守护核心机密数据（例如技术方案实施案例）。

检索情况

国家/地区	检索网站	检索公式 (关键词及组合方式)	文献数量(指依据检索关键字直接获得的文献数，不需要列出具体的文献信息)
CN/US/EP	http://www.incopat.com （推荐使用。账号获取：直接通过主页“IP登录”进入） 使用教程及密码： http://3ms.huawei.com/hi/group/2033427/wiki_5014231.html		
其它网站	https://scholar.google.com（适用于使用**专利号**直接检索） 标准网站等其它网站，请自行增加。	GPU-accelerated fully homomorphic encryption database transaction Dependency-based scheduling for fully homomorphic encryption database transaction	262 21

Thank You

www.huawei.com

www.huawei.com

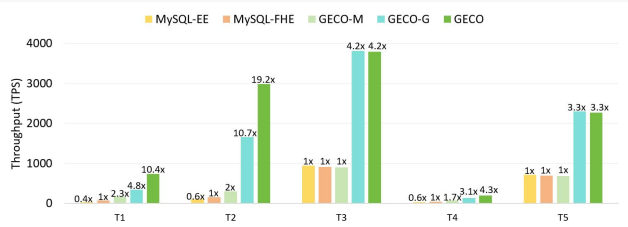
HUAWEI

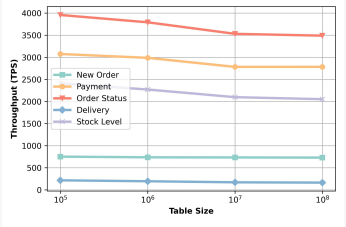
一种基于关联事务融合和GPU三维加速的全同态数据库中间件

- **现有方案一：现有全同态数据库使用全同态加密（例如CKKS）**
- ☒ 缺点1：**密文运算性能极差**。全同态算术运算开销高昂
- ☒ 缺点2：**密文比较性能极差**。涉及全同态密文比较的数据查询需要进行**复杂度高达O(MN)的全表扫描**

- **现有方案二：传统密态数据库使用传统加密算法（例如RSA）**
- ☒ 缺点1：**性能低下**。密文事务执行牵涉**先解密、后加密**的繁琐流程
- ☒ 缺点2：**明文易泄**。密文事务执行期间会导致数据**明文暴露**。

- **有益效果：使用TPC-C进行端到端性能测试**
- ☑ GECO带来端到端约3.3-19.3倍吞吐量提升和最高36%平均时延降低
  - ☑ 关联事务融合协议可带来约最高2.3倍吞吐量提升和最高17.4%平均时延降低（贡献1）
  - ☑ GPU三维加速全同态事务可带来约10.7倍吞吐量提升和最高8.2%平均时延降低（贡献2）

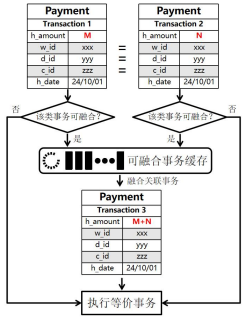




本发明解决方案

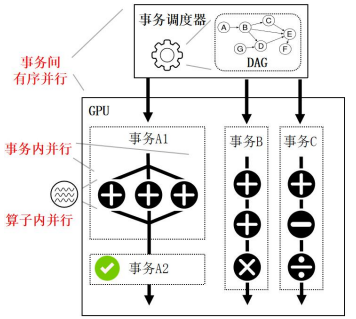
- **贡献1：关联事务融合**

实现目标：通过在关联事务静态声明和动态融合，将多个关联事务融合为单一等价事务，通过减少全同态事务实际执行数量，从而减少显著降低极耗时的全同态事务实际执行数量，提高端到端吞吐和显著降低时延。



- **贡献2：GPU三维加速全同态事务**

实现目标：使用GPU和事务调度器实现三个层级并行加速(事务间有序并行、事务内并行、算子内并行)，在提高并发度的同时消除事务并发冲突引发的回滚重试，显著提高吞吐量和降低时延。



- **技术方案实施例：基于中间件架构实现本专利加速传统非密态云端数据库**