# 论文综述

**论文题目：**Solar: Towards a Shared-Everything Database on Distributed Log-Structured Storage（solar :在分布式日志结构存储上实现共享一切数据库）

**作者及单位：**Tao Zhu, East China Normal University; Zhuoyue Zhao and Feifei Li, University of Utah; Weining Qian and Aoying Zhou, East China Normal University; Dong Xie and Ryan Stutsman, University of Utah; Haining Li, Bank of Communications; Huiqi Hu, East China Normal University; Bank of Communications

**摘要：**大型数据库上的高效事务处理是许多任务关键型应用程序的关键要求。虽然现代数据库已经通过水平分区获得了良好的性能，但是当必须执行跨分区分布式事务时，它们的性能会下降。论文介绍了Solar，这是一个分布式关系数据库系统，在一家大型商业银行得到了成功的测试。太阳能的主要特点包括:

1)基于两层日志结构合并树的共享一切架构；

2)一种新的并发控制算法，可与日志结构存储一起使用，即使存储层压缩后台节点之间的数据，该算法也能确保高效、无阻塞的事务处理；

3)细粒度的数据访问，以有效地最小化和平衡集群内的网络通信。

## 1.研究现状

NoSQL系统的成功展示了横向扩展架构的优势，可实现近线性可扩展性。但是，由于分布式数据存储，很难支持它们中的事务，这是大型数据库的基本要求。例如，Bigtable仅支持单行事务，而其他像Dynamo 则根本不支持事务。为了满足对事务支持的需求，NewSQL系统被设计用于在具有分布式数据存储的集群上进行高效的在线事务处理（OLTP）。

分布式事务处理很难，因为需要节点之间的高效同步以确保ACID属性并保持良好的性能。 尽管许多最近提出的系统取得了重大进展和成功，但它们仍然具有各种局限性。 例如，依赖于无共享架构和2PC（两阶段提交）的系统严重受到跨分区分布式事务的影响，因此需要针对给定工作负载进行仔细的数据分区。另一方面，像Tell 这样的分布式共享数据系统需要大规模不常用的特定硬件支持。

也就是说，如果不能事先假设事务工作负载，并且没有特殊的硬件支持，那么在商用集群上实现高性能事务处理仍然是一个具有挑战性的问题。同时，先前的研究还表明，通过探索多核和多插槽（例如，NUMA）架构，可以在单个节点上设计高性能事务引擎。Silo和Hekaton 都使用单个服务器进行事务处理，并展示了高吞吐量。然而，这样的系统可能无法满足其数据不能适合单个节点的大数据应用的需要，因此需要支持分布式数据存储。

## 2.solar的设计考虑

Solar使用共享所有架构设计，其中服务器节点（称为T节点）被保留用于内存中事务处理，多个存储节点（称为S节点）用于数据存储和读取访问。本质上，Solar中的S节点形成分布式存储引擎，T节点充当主内存事务引擎。分布式存储引擎利用节点集群来实现数据库容量和服务并发读取的能力的可扩展性。事务引擎提供有效的事务处理，并通过其内存提交列表临时存储已提交的更新。通过在后台运行的数据压缩过程，定期将T节点上最近提交的数据项合并回S节点，而不会中断正在进行的事务。总体而言，Solar旨在实现高性能事务处理和可扩展数据存储。

为了加速系统中的更新操作，T节点上的内存提交列表和来自所有S节点的磁盘存储共同形成分布式双层日志结构合并树设计。此外，引入称为P单元的处理层以执行来自S节点的数据访问和事务中所需的任何计算，使得T节点可以摆脱协调数据访问和执行业务逻辑计算的负担。 存储和计算分离的这种设计还使系统能够利用所有CPU资源进行事务调度和验证。

## 3.贡献

 （1）为了实现高性能事务处理，提出了具有T节点，一组S节点和P单元的分布式共享一切架构。

（2）研究了一种称为MVOCC的混合并发控制方案，该方案结合了OCC（乐观并发控制）和MVCC（多版本并发控制）方案。

（3）数据合并算法作为MVOCC的一部分，旨在有效地将T节点上的提交列表定期合并回S节点，而不会中断T节点上的事务处理。

（4）研究了几种优化以改善整体性能，例如，通过P单元分离计算和存储过程，在一个事务中对多个数据访问操作进行分组，维护比特数组以避免对分布式存储引擎的不必要的数据访问。

## 4.系统架构

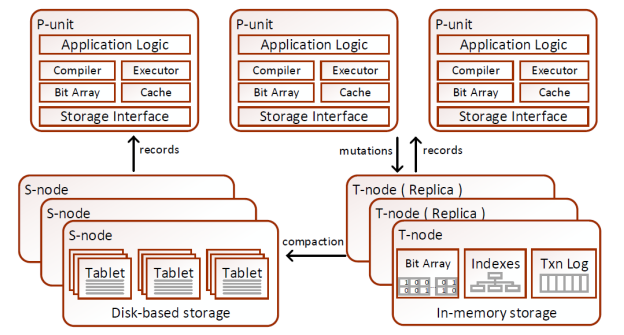


图4.1 solar的系统架构

图4.1概述了Solar的架构。 Solar使用多版本乐观并发控制协议将事务处理分为计算，验证和提交阶段。可以在任何一个P单元上启动事务，这些P单元除了用于数据访问优化的几个小数据结构之外不存储任何数据。P单元处理来自T节点或S节点的所有数据提取以及事务处理。写入在P单元缓冲，直到事务提交或中止。当事务准备好提交时，P单元将写集发送到T节点以进行验证和提交。一旦T节点完成验证，它就会将更新写入其内存存储器，并且还会写入预写日志以确保持久性。最后，如果事务成功提交，T节点通知P单元。 P单元可以在集群内部或外部的任何机器上实例化（通常在S节点或客户端）。它们从T节点卸载了大部分计算负担，因此T节点可以专用于事务管理。集群信息（例如，所有节点的状态，数据分布）由管理器节点维护，并由其他节点高速缓存。

## 5.solar的事务管理

（1）处理

P单元执行事务tx的用户定义逻辑，并从T节点和S节点中读取tx涉及到的数据，事务tx首次与T节点时获得读取时间戳RTX，P节点从T和S节点获取到的数据版本的时间戳均应小于RTX。P节点执行完业务逻辑代码之后，将tx的写集wx提交给T节点验证。

（2）验证

T节点主要验证事务tx和其他事务之间潜在的写冲突。T节点锁定T节点的Memtable上事务tx的写集中的所有数据（记为wx），并检查 Memtable 中是否存在时间戳大于RTX 的任何记录 r ∈ wx 的更新版本，若没有更新版本，则提交，否则终止。

（3）提交

为wx中的每个记录创建新版本。通过递增全局计数器获得tx的提交时间戳CTX，并为新版本数据加入标识符CTX。为防止丢失更新，在（RTX，CTX）范围内，属于wx的记录r被其他事务更新，则终止事务tx的提交。

## 6.系统优化

（1）利用缓存

P单元的缓存池保存从S节点获取到的记录，为相同记录的数据提供访问服务，为防止缓存数据过期，每次T节点执行数据压缩之后，清空缓存池。

（2）利用比特数组

P单元首先从T节点的Memtable获取数据，若没有，再访问S节点的SStable，这样造成对T节点空读，浪费T节点资源。

Bit数组b记录块中的一列是否被修改，若块T中的第C列被修改，则bit数组中的（T, C）位置记录为1。P单元保存数组b在t时刻的备份bt，访问数据之前，首先查询bt表，若值为1，则从Memtable读取数据，否则直接从Sstable读取数据，由于编码的粒度，查询 bt 会导致误报，上述方法对于读密集型或只读列最为有效。

## 7.实验结果

根据对TPC-C，Smallbank和实际工作负载的实证评估，当分布式交易接近或超过5％时，Solar的性能优于现有的无共享系统，最高可达50倍。

## 8.项目支持

Tao Zhu,Weining Qian and Aoying Zhou are supported by 863 Program (2015AA015307), National Key R&D Plan Project (2018YFB1003303),

NSFC (61432006 and 61332006). Feifei Li, Zhuoyue Zhao and Dong Xie are supported in part by NSF grants 1619287 and 1443046.

Feifei Li is also supported in partby NSFC grant 61729202.

Ryan Stutsman is supported in part by NSF grant CNS-1750558.