综述报告

这篇文章发表在2018年的USENIX Annual Technical Conference会议上，会议的宗旨是“将领导性的研究者和实践者带到一起进行的前沿技术探讨” ，通常每届会议录用30篇左右论文，录取率在15%左右，是操作系统、体系结构方面最好的会议之一。会议于2018年7月11-13日在美国波士顿举行。文章由ERC Grant 339539 (AOC).部分支持。

文章是关于基于NVRAM内存中无日志数据结构的算法设计。一些来自惠普、谷歌等公司和知名大学的学者都对NVRAM内存数据结构的研究非常感兴趣。在NVRAM方面，在OOPSLA会议、ASPLOS会议、SYSTOR会议、EuroSys会议等会议上，Kumud Bhandari, Dhruva R Chakrabarti, and Hans-J Boehm等等团队提出了多种基于事务的交互手段，其方法具有良好的通用性和易用性，但是由Seunghee Shin团队在2017年ISCA会议上提出基于事务的方法会造成极大的延时，并提出了一些减轻延时的尝试。Izraelevitz团队在2016年ASPLOS会议上介绍了一种方法，通过在每个线程中可靠地跟踪最后执行的存储指令，可以在重新启动之后简单地完成关键部分的执行。如果写回缓存是持久的，那么这种方法虽然有效，但在其他情况下，需要为关键部分中的每个存储对日志进行写操作，此外Kolli团队也在同会议上提出将重点放在基于锁的应用程序中的静态事务上，并试图最小化持久化依赖，以限制等待时间，还展示了如何在不持有任何锁的情况下执行事务的提交阶段。在次年的ASPLOS会议上提出的DUDETM方法通过首先执行事务并在易失性内存中获得redo日志，然后原子性地将redo日志刷新到持久内存，然后仅修改原始数据，从而优化了重做日志记录。在2017年的EuroSys会议上，Amirsaman Memaripour团队还提出一种写时复制的技术，并避免在关键部分进行日志记录。

许多基于NVRAM工作都采用基于锁的方法，与文章方法相比，会产生额外开销，特别是日志记录的开销是非常大的。关于NVRAM中的索引树问题，在FAST、VLDB等会议上有专家学者提出一些建议，但是他们大都基于某种形式的日志记录，或者会产生内存泄漏问题。在2018年的PPoPP会议上，Michal Friedman团队介绍了用于持久队列的无锁算法，但是该方法也没超出这个数据结构，同时不考虑内存管理。支持无锁算法的其他工作要么假设持久缓存，要么自动生成持久算法，在每次更新后发出同步。NVRAM的一般内存分配和回收问题也受到了关注。通用持久内存框架将分配和回收作为它们提供的事务机制的一部分，从而依赖于日志记录。在2015年ADMS会议上David等人提供了分配和释放持久内存的接口，但是由于细粒度的计算，每次分配和重新分配都会带来大量的开销。此外还有学者在粗粒度级别上跟踪分配器元数据。但是，它们在恢复时需要更高的成本，因为它们需要在整个内存中传递垃圾收集。

其他针对NVRAM的Memcached修改已经被提出，但它们广泛使用事务或者它们不能保证所有已完成的请求都是持久的，而NV-Memcached确保所有已完成的请求都是持久的，并通过使用无日志哈希表将事务限制在slab分配器代码中。