|  |  |
| --- | --- |
| 论文题目 | 《An Analysis of Persistent Memory Use with WHISPER》 |
| 论文摘要 | 本文介绍了NVM技术的特点，然后介绍并分析PM基准套件WHISPER，该套件收集了10个PM应用程序 ，可以覆盖当前所有接口。然后基于分析提出了HOPS来自追踪硬件中PM的更新。HOPS比当前的持久化方法在性能上提高了24.3%。 |
| 研究目的 | 通过WHISPER基准套件，为PM系统研究打下坚实的基础。WHISPER包含我们搜集的十个PM应用程序：Echo、N-store、Redis、C-tree、Hashmap、Vacation、Memcached、NFS、Exim、MySQL。这些应用程序覆盖了各种各样的PM接口，例如数据库、内存数据存储和持久堆 |
| 论文内容 | **分析PM的特征：**  ①再PM感知应用程序中，只有4%写入PM，剩下的全部都写入非易失性内存。  ②软件事务通常需要使用5到50个排序点来执行并且只有在最后一个排序点之后才需要持久化  ③75%的epochs直接在一个64B缓存行上进行更新，并不需要在相同的缓存行  ④来自相同线程的80%epochs取决于相同线程的之前的epochs，只有少部分epochs取决于其他的线程。  **PM编程存在的问题：**  ① 数据只有到PM之中才会实现持久化；易失性处理器缓存中的数据会在断电之后丢失。因此，应用程序需要持久性就必须确保数据离开缓存行并且等待数据到达PM。  ②在持久性数据结构修改中，系统崩溃包括断电可能会导致恢复时出现不一致的状态。  ③写回处理器缓存行能将PM中的更新重排序，这就意味着即使是已经排序的更新在到达PM时会乱序  **介绍WHISPER**  WHISPER的访问层：Mnemosyne、NVML、PMFS  WHISPER的应用程序：N-store、Echo  **HOPS系统**  HOPS系统组成：硬件扩展，PB，ISA原语，ofence和dfence |
| 性能评估 | ①对于x86-64实现，PWQ平均减少15.5%的运行时。  ②HOPS（NVM）比x86-64（NVM）的性能提高24.3%，比x86-64（PWQ）提高10%。  ③HOPS（PWQ）比HOPS（NVM）只能提高1.4%。 |
| 论文结论 | PM是一个低延迟性的接口。基准套件WHISPER包含实际的PM应用程序。HOPS通过跟踪和执行PM应用程序在硬件上的顺序和持久性约束，在应用程序性能上获得24.3%的增益。 |
| 论文题目 | 《PMTest A Fast and Flexible Testing Framework for Persistent Memory Programs》 |
| 论文摘要 | 最近的非易失性存储技术（如3DXPoint和NVDIMM）启用了可直接在内存中操作持久性数据的持久性内存(PM)系统。内存技术的这一进步推动了一套用于PM的新的崩溃一致性软件(Ccs)的开发。本文开发一个测试框架，帮助程序员识别CCS中的崩溃一致性错误。  以前的测试工具缺乏通用性，①它们只适用于一个特定的CCS或内存持久性模型；②引入重要的性能开销。  本文提出了PMTest。在现有的8个项目中，检测出45bugs，和3个PMFS和PMDK开发的程序中的新bugs。 |
| 论文目的 | 目标是设计一个灵活又快速崩溃一致性测试框架。 这个框架既适用于所有可持久模型又能快速检测的工具。 |
| 论文思路 | 测试崩溃一致性的关键思想：  ①灵活性：通过两个类型的操作来保证持久性和顺序性保证：强制内存地址持久化和强制执行持久性之间的顺序。  ②快速性：关键思想是在运行时跟踪PM操作，减少写入的时间间隔。  PMTest的过程：  ①offline阶段：程序员按照崩溃一致性机制的程序规范使用低级或高级检查器对CCS进行注释。  ②online阶段：在执行期间，PMTest会追踪程序中的PM操作并传递给检测引擎，检测引擎根据检测机制对这些信息是否满足checker的要求进行检测，最后检测引擎会报出警告和错误回馈，并附带checker的文件名和代码行 |
| PMTest接口 |  |
| 性能评估 | 由于在实际的工作负载中访问PM没有那么密集，所以PMTest在实际负载中是有效的。 |
| 论文结论 | PMTest是一个兼具灵活性和快速性的工具，通过测试为PM系统开发的CCS，并在用户空间应用程序和内核空间文件系统中检测到了新的bug，从而证明了PMTest的有效性。PMTest可以扩展到支持文献中提出的不同的持久性模型 |