NVM in ICDE2018

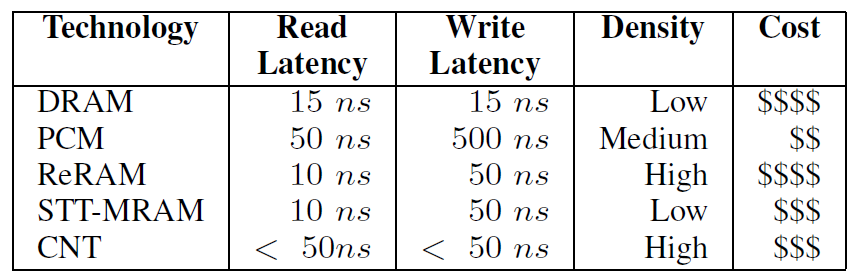
NVM的出现深刻改变现有的软件技术栈，涉及到操作系统、数据库等系统的方方面面。它作为ICDE2018的一个主题，包括一个keynote，一篇长文、一篇workshop论文和一篇短文。这里详细介绍这个keynote，它相当于一个简短的综述。另外3篇文章跟keynote里面的特定细节相关。

Keynote：An NVM Carol: Visions of NVM Past, Present, and Future（细节详见[论文](http://conferences.computer.org/icde/2018/pdfs/ICDE2018-1k0M6MMYnBWkOvJ3BzdLQf/1hj8a20c8d8lrdekOP2eez/7GRWSngv5HPDeQS1kgm9hi.pdf)）

作者简介：[Margo Seltzer](http://www.eecs.harvard.edu/~margo/)是哈佛大学教授，右图是他的照片。她的研究兴趣包括与系统相关的多个方面：文件系统、数据库、事务处理系统、存储和分析图形结构数据的系统等。

2015年7月份，Intel宣布的3D XPoint产品标志着Non-volatile memory (NVM)技术取得巨大进步。它具有DRAM相近的延迟和flash相似的持久性。它底层基于Phase Change Memory(PCM)技术。与PCM相互竞争技术的比较如下表所示。与之前的持久性存储相比，NVM最大的不同之处在于它是字节寻址。这项技术也带来巨大的挑战：数据在任何时候都是持久化的。下面从NVM的过去、现状和未来三个方面介绍这项技术。

表 1 NVM相关技术对比



1. NVM的过去

基于NVM建立系统会引入大量的挑战，但幸运的是，诸多需要解决的问题可以借鉴于之前的工作，其中有下面4个与NVM紧密相关的工作，涉及到体系结构设计、编程模型和事务处理等多个计算机系统的基础内容。

1. Bubble memory （磁泡存储器）

“如果不记得过去，就有可能重蹈覆辙”。人们都期望这些新技术能够替代持久存储，改善数据库、操作系统和编程的架构。但是是否如愿以偿呢？在上个世纪六十年代，bubble memory被广泛研究，并且取得一些成功。然而它最终被淘汰的历史警告我们不要对NVM太乐观，因为最终由价格/性能曲线决定。NVM能否流行取决于它能否有DRAM类似的性能和Flash差不多的价格。

1. Single level store （单级存储）

虚拟内存的出现无缝地整合了易失的主存和持久存储。其中的虚拟地址既可以指向主存中的数据又能访问持久存储中的数据。这使得任意两个存储的组合看起来似乎是一个整体。比较这个单级存储的历史和现在的研究是有启发意义的。persistence包括两个基本的概念：naming和storage。当今的文件系统中的namespace和protection非常适合于persistence。它促成一个两部分的解决方案：整合传统的文件系统中的naming与protection机制和通过进程地址的之间访问。其中有一个值得反思的问题是，传统的文件系统中的naming和protection机制是否适合于持久性内存？

1. Persistent object （持久性对象）

当我们模糊内存对象和持久对象的界限，我们联系到持久对象系统和面向对象数据库。这些领域的技术方法很值得借鉴。比如PS-Algol跟今天NVM中的工作很相似，都秉承一个信念：给一种语言增加持久性但是最小程度地修改这种语言是一种可行方案。OPAL和PC++调用自己的系统并且设计了特定的数据模型。还有Java语言在持久性方面的探索。这些系统都面临在易失的和持久的存储中移动对象。这个问题也在NVM中也有所涉及。

1. Transactional memory （事务性内存）

在NVM中，任何时候的写操作可以被持久化。这个硬件事实是之前的持久性对象系统和当今的NVM最大的差别。这为硬件事务性内存提供了可能。但是仍然有一些根本上的限制。但是这些挑战在软件的（STM）和混合实现中有些探索。

1. NVM的现状

NVM的发展现状跟之前的历史有不同之处。首先，NVM不再是单一技术，而是有许多与商业相关的可行的技术。其次，有许多的商业合作围绕着这项新技术展开。最后，除了硬件产商，像HP，IBM这样的系统公司也很活跃。相比而言，学术界更加的兴奋，围绕着NVM展开多项研究，综合于下面4个方面。

1. Synchronization，consistency和programming model （同步、一致和编程模型）

传统的内存中和持久存储中的同步、一致性机制不适用于NVM。亟待一种能够保序和细粒度同步的编程模型。Intel的ISA和addressable persistent file system（BPFS）正在保序方面探索。NVM direct library和NVM programing model from SNIA提供不同的同步机制。作者相信未来的高级库会封装对NVM的直接操作。

1. Naming （命名）

Phil Karlton说过缓存失效和naming是计算机科学的两大难题。通过分层的路径名来访问持久性数据是当下通用的方式。但是对于NVM而言，有没有必要在NVM中实现文件系统的命名空间。或者混合两种不同的方式使用NVM：首先通过文件系统实现naming，然后采用直接访问存储的方式获得较好性能。那有没有更好的方案呢？

1. Replacing persistent storage componets （替换持久存储的部件）

与之前的基于硬件驱动之上从事系统设计不同，也可以在现有系统中使用NVM，例如日志、文件系统和数据库。那就要仔细研究如何利用NVM中的基于字节访问的机制，这不同于原来的基于块访问的机制。

1. Warn restart （热启动）

一旦重启，机器上的任何程序都有一个缓存。但是这些中间状态是否跟重启之前的完全一致呢？以及如何处理持久性的bugs？

1. NVM的未来

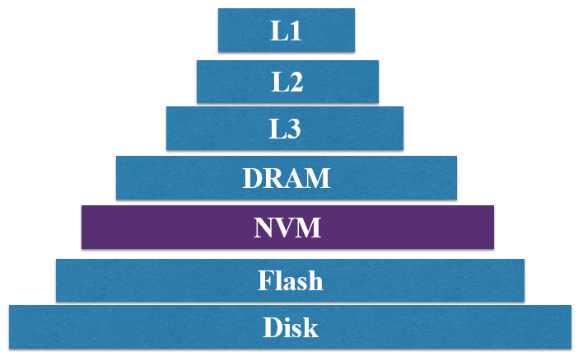


图 1新的系统硬件架构图

如果NVM变成标准的持久性存储（一种可能的架构如上图所示），仍然有许多问题需要解决。我们遗漏的两个问题是整个系统的持久性和嵌入式系统。另外最大的开放性秘密是需要redundancy和replication以保证高可用和容错。还有versioning、encryption和naming需要进一步研究。



图 2 现场照片

[Full paper](http://conferences.computer.org/icde/2018/pdfs/ICDE2018-1k0M6MMYnBWkOvJ3BzdLQf/1akXgsItqY7YP26IcHG62m/5DL0Ym9p9YgfZCdDXSkLWL.pdf): Easy Lock-Free Indexing in Non-Volatile Memory

这篇文章探索在NVRAM数据库中实现无锁、持久以及容易恢复的索引，具体包括双向跳跃表、Bw-tree和B+-tree。

[Workshop paper](http://conferences.computer.org/icde/2018/pdfs/ICDEW2018-5mLRt2m8Bnczm3e7LJVm03/4EUDCPqHkveiz3Dh9LdK6t/2W7t9YY4G9Rs3QLZLtJzOD.pdf): An NVM-aware Storage Layout for Analytical Workloads

这篇文章提出一个NVM感知的table存储设计。它使用一个多维聚集的方法和类似块的结构来使用整个内存栈（框架图如下）。该架构支持优化非键值的查询。

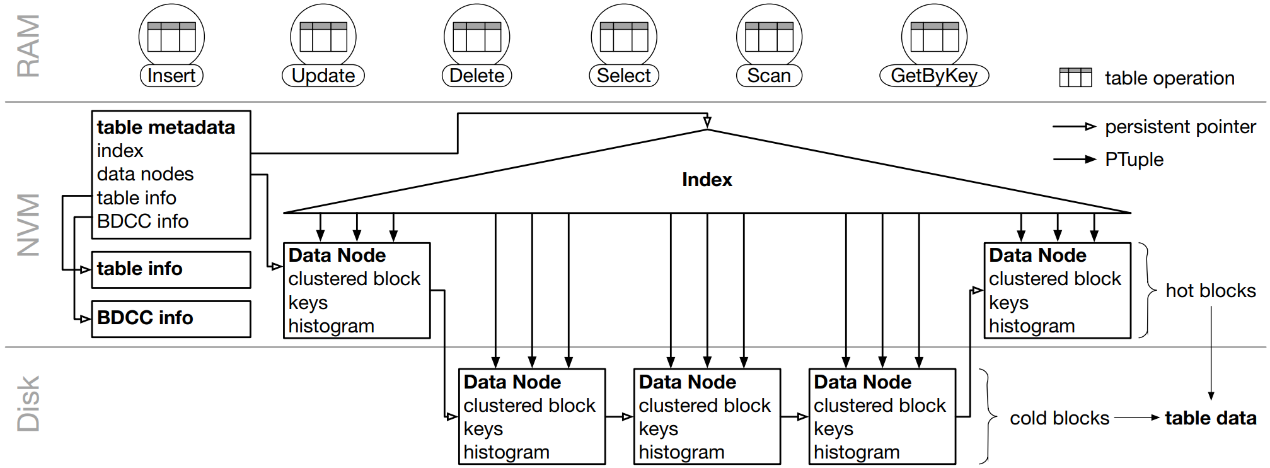


图 3 NVM感知的table存储架构

[Short paper:](http://conferences.computer.org/icde/2018/pdfs/ICDE2018-1k0M6MMYnBWkOvJ3BzdLQf/35bmHMAR6ylRhFeX33vH54/6EbztrxhipQls2RntGA0ir.pdf) On Configuring a Hierarchy of Storage Media in the Age of NVM

这篇论文利用NVM优化数据缓存中间件。它解决的一个问题是选择哪种介质以及决定该介质的容量来构成一个定量的缓存中间件，另外一个问题是何时拷贝数据存储于多种介质。