分　数：	
评卷人：	

華中科技大學

研究生（数据中心技术）课程论文（报告）

题　目：基于 NVM 和 RDMA 的分布式存储系统综述

学　　号　M202076709

姓　　名　刘旭冉

专　　业　电子信息

课程指导教师　施展　童薇

院（系、所）　武汉光电国家研究中心

2020 年 12 月 28 日

基于 NVM 和 RDMA 的分布式存储系统综述

刘旭冉[1)]

[1)](华中科技大学 武汉光电国家研究中心，武汉 430074)

摘 要 近年来，速度快、容量大、字节编址且提供非易失性存储的新型存储器非易失性存储器 NVM（Non-volatile Memory）和高带宽、高吞吐、低延迟的新型网络传输技术远程直接数据存取 RDMA（Remote Direct Memory Access）正在大数据领域被广泛应用。NVM 和 RDMA 技术以其优秀的性能为提高分布式存储系统的性能带来了新的机遇，然而直接用两者替换现有系统中的传统存储模块和网络通信模块将面临一些问题，不能充分利用它们带来的优秀硬件性能。本文首先简要介绍 NVM 和 RDMA 技术，然后说明为什么不能直接简单地用两者替换现有系统中的传统存储模块和网络通信模块，最后以三个相关工作为例，介绍如何在分布式存储系统中通过减少软件逻辑冗余和充分利用 NVM 和 RDMA 的硬件特性来优化系统。

关键词 非易失性存储器；远程直接数据存取；分布式存储系统

Survey on Distributed Storage Systems Based on NVM and RDMA

Liu Xuran[1)]

[1)](Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074)

Abstract Nowadays, byte-addressed Non-volatile Memory with high speed and large capacity and Remote Direct Memory Access with high bandwidth, high throughput and low latency are applied in bigdata area. Their exceptional performance provide opportunity for distributed systems to achieve better performance. However, simply replacing the communication module and storage module will not make full use of their advance in performance. In this article, we will first introduce NVM and RDMA, then explain why not simply replacing the corresponding modules, last show how to optimize distributed storage system by reducing software redundancy and exploiting hardware feature using three related systems as examples.

Key words Non-volatile Memory; Remote Direct Memory Access; Distributed Storage System

1 背景介绍

近年来，速度快、容量大、字节编址且提供非易失性存储的新型存储器非易失性存储器 NVM（Non-volatile Memory）和高带宽、高吞吐、低延迟的新型网络传输技术远程直接数据存取 RDMA（Remote Direct Memory Access）正在大数据领域被广泛应用。NVM 和 RDMA 技术以其优秀的性能为提高分布式存储系统的性能带来了新的机遇，本章将简要介绍 NVM 和 RDMA 技术。

1.1 NVM

非易失性存储器 NVM（Non-volatile Memory）或称持久存储 PM（Persistent Memory）具有速度快、容量大、字节编址且提供非易失性存储的特点。NVM 提供接近内存的访问速度和远超硬盘的巨大容量。

持久内存产品 Intel 傲腾持久内存 Optane DCPMM（Intel Optane DC Persistent Memory Module）于 2019 年 4 月发布。

Optane DCPMM 安装在内存总线上并由可以 CPU 的 Load/Store 指令访问，具有不对称的读写性能：总的读带宽可达 37.6 GB/s，写带宽的峰值仅为 13.2 GB/s，并且 Optane DCPMM 的读访问延迟(~300 ns)稍高。

除此之外，CPU 对 PM 的原子写操作是以 8 字

节为单位的，这与 HDD 和 SSD 的 512B 或 4KB 的原子写操作不同。这使得编程人员需要采取额外的技巧，例如 undo/redo log 来保证原子性。

在 PM 写操作的过程中，写操作先被缓存到 CPU cache 中，然后以随机的顺序被写入 PM 中，这使得编程人员需要明确这些写操作真正被执行的顺序，才能保证数据一致性。保证 PM 写操作的执行顺序有两种做法：一种是使用 clflush、clwb 或 clflushopt 指令；一种是在每个 PM 写操作后使用 mfence。

1.2 RDMA

远程直接内存访问(remote direct memory access，RDMA)技术正在大数据领域被越来越广泛地应用，它支持在对方主机 CPU 不参与的情况下远程读写异地内存，提供高带宽、高吞吐和低延迟的数据传输特性。

RDMA 提供两种原语，双向原语和单向原语。双向原语以 Send 和 Recv 为代表，类似传统的网络套接字，在发送消息时，接收方需提前调用 Recv 原语，用于指定接收消息的存放地址等。单向原语包括 Read，Write 以及相应的变种（例如带有立即数的指令），能在远端 CPU 不介入的情况下直接读取或更新远端内存。

下面以远程写操作为例，如图 1 所示，介绍 RDMA 网络数据收发过程：首先本地 CPU 直接以 MMIO 方式向网卡发远程写命令，并传递相应参数（待写入数据块基地址、远端内存地址、写入数据块大小、远端注册内存密钥等）；然后，本地网卡收到命令之后，根据参数本地数据块基地址将数据块从主存以 DMA Read 的方式读取到网卡缓存，并发送到远端；最后，远端网卡接收到数据块之后，以 DMA Write 的方式直接将数据写入内存对应地址。不同于传统网络中数据需要在进入内核中的多层网络协议栈中进行拷贝传递，整个过程中，RDMA 实现了内核旁路和零拷贝的跨节点数据传输。

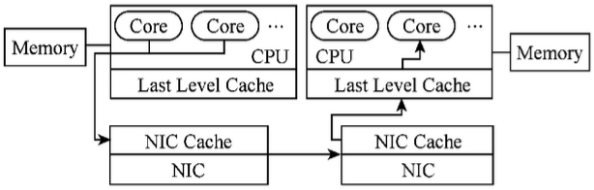

图 1 RDMA 网络数据收发过程

RDMA 与传统网络相比有着明显的优势。RDMA 实现了内核旁路，这使得应用程序可以直接在用户态执行数据传输，不需要在内核态与用户态之间做上下文切换。RDMA 的数据传输过程是零拷贝的，这使得数据能够被直接发送到缓冲区或者能够直接从缓冲区里接收，而不需要被复制到网络层。这些特征使得 RDMA 达到了与传统网络相比较而言更优秀的性能：更高的带宽和更低的延迟，基于 RDMA 的分布式存储统将为满足大数据高时效处理和存储带来新的机遇。

2 问题与挑战

NVM 和 RDMA 技术以其优秀的性能为提高分布式存储系统的性能带来了机会，但直接用 NVM 和 RDMA 技术替换现有系统中的传统存储模块和网络通信模块将面临一些问题，不能充分利用它们带来的优秀硬件性能。本章将说明为什么不能直接简单地用 NVM 和 RDMA 技术替换现有系统中的传统存储模块和网络通信模块。

2.1 软件逻辑冗余

现有软件系统的模块化设计减少了软件开发的工作量，提高了系统代码的可读性，方便了系统功能的扩展，但同时也导致系统的软件逻辑中存在大量的冗余，降低系统的效率。在传统系统中，网络延迟较大、硬盘的传输速度较慢，是系统性能的主要瓶颈。然而有了高速的网络传输模块 RDMA 和高速的存储介质 NVM 之后，系统的软件开销变得不可忽略，需要对系统软件进行优化。

2.2 硬件特性闲置

传统系统的网络传输和数据存储模块针对传统的网络技术和存储介质进行设计，未充分考虑到 NVM 和 RDMA 硬件提供的特性，利用这些特性，有机会实现更好的性能。例如，NVM 存储器是字节编址的，不同于传统硬盘，针对传统硬盘设计的系统存储模块未能充分利用这一特性；同时，NVM 又不同于传统内存，是非易失性的，针对传统内存设计的系统存储模块未能充分利用这一特性。RDMA 技术不同于传统网络传输模块的内核旁路等特性亦未被现有系统充分利用。

3 相关研究

本章以三个相关工作 RDMP-KV、pDPM 和 TH-DPMS 为例，介绍如何在基于 NVM 和 RDMA

的存储系统中通过减少软件逻辑冗余和充分利用 NVM 和 RDMA 的硬件特性来优化系统。

3.1 RDMP-KV

3.1.1 介绍

RDMP-KV（Remote Direct Memory Persistence based Key-Value stores）[1]，是一种针对带 NVM 的 KV 存储系统的 RDMA 通信架构，使得客户端可以直接访问和持久化地更新服务端 PMEM 里存储的 KV 对，如图 2 所示。这种设计的目标是利用 NVM 字节编址的特征避免在 RDMA communication buffers 和 PMEM 之间拷贝数据(DRAM-to-PMEM staging)。

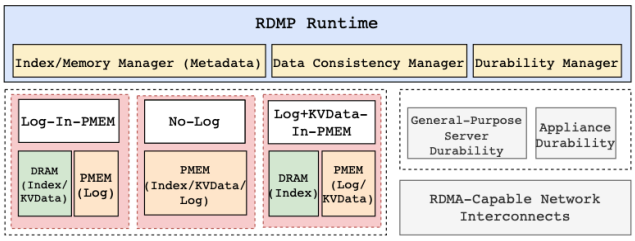

图 2 RDMP-KV 架构

3.1.2 系统设计

不同于其他不针对 NVM 的字节编址特点进行优化的 RDMA 通信方式，RDMP-KV 设计了一种巧妙的策略避免服务端在 DRAM 中的 RDMA 通信缓冲区和持久存储 PMEM 之间进行数据的拷贝，该过程论文称之为 DRAM-to-PMEM staging。

以 GET 过程为例，在其他不针对 NVM 的字节编址特点进行优化的 RDMA 通信过程中，首先客户端向服务端发送包含 KV 对的键值 K 的请求，然后服务端根据 K 在 DRAM 中查找 index，获得在 PMEM 中存储的数据的元数据，接着根据 index 在 PMEM 中读取目标数据，最后发送给客户端，如图 3 所示。

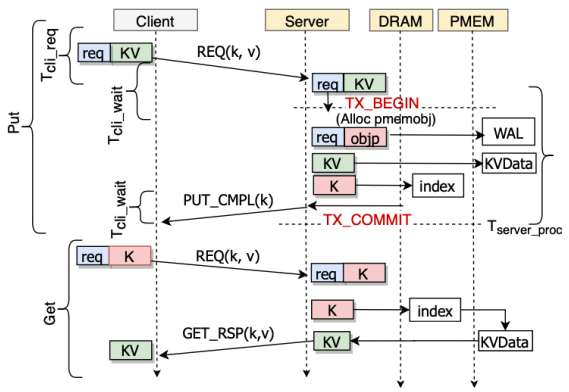

图 3 Pmem-Redis 中的 PUT 和 GET 流程

而在 RDMP-KV 中，GET 过程分为两个阶段：server-reply 阶段中，客户端同样向服务端发送请求，服务端同样查找 index，但服务端不再将数据全部读取发给客户端，而是仅将找到的元数据信息包括数据在 PMEM 中的地址发送给客户端；在下一个阶段 server-bypass 阶段中，客户端根据收到的数据地址，直接通过单边 RDMA 从 PMEM 中读取数据，如图 4 所示。

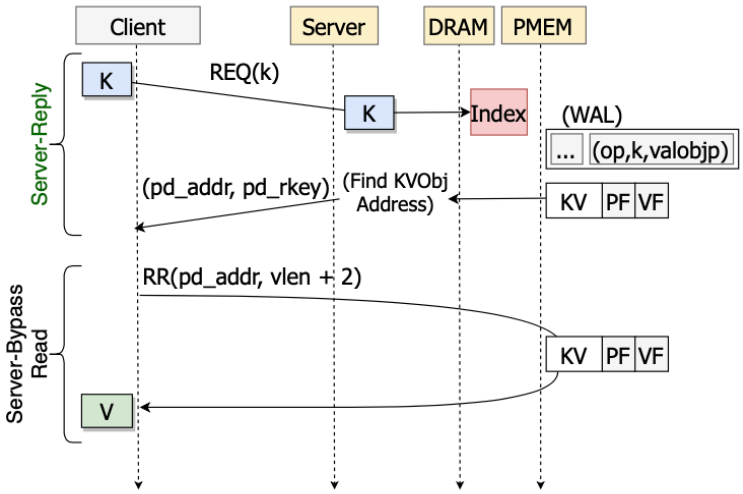

图 4 RDMP-KV 的 GET 流程

PUT 过程与 GET 类似，在其他不针对 NVM 的字节编址特点进行优化的 RDMA 通信过程中，首先客户端将带有 KV 对数据的请求发送到服务端，然后服务端为 KV 对分配存储空间，接着服务端将数据写入 PMEM 中的对应地址，最后向客户端返回完成的信号，如图 3 所示。

而在 RDMP-KV 中，PUT 过程与 GET 类似，服务端并不负责将客户端发来的数据写入 PMEM，而是仅根据客户端提供的数据长度为 KV 对分配存储地址，将地址返回给客户端，由客户端使用单边 RDMA 将数据写入对应地址，如图 5 和图 6 所示。接下来，与 GET 不同的是，PUT 操作需要进行写入数据的持久化，RDMP-KV 针对不同的实际硬件设计了两种不同的持久化方式，分别称为 RDMP-SA（Server-Assisted RDMP Protocol）和 RDMP-CC（Client-Centric RDMP Protocol）。在 RDMP-SA 模式下，服务端根据客户端发送的带有立即数的写指令中提供的需要刷写的数据长度等信息用 flush 指令刷写数据，完成后向客户端发送持久化完成的信号。RDMP-CC 模式需要硬件提供持久化功能如 RDMA Commit，该模式完全旁路服务端 CPU，客户端在写数据时使用 RDMA Commit

来实现写入数据的持久化。

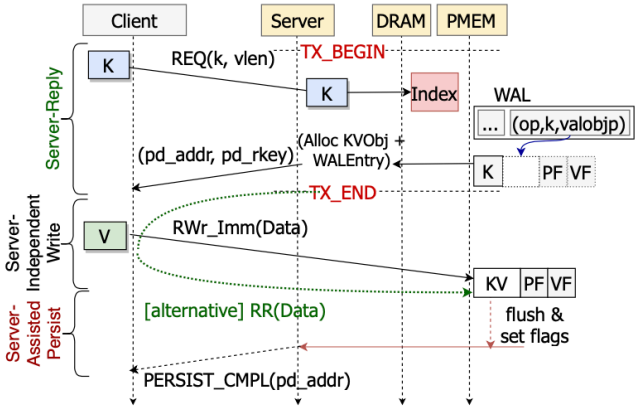

图 5 RDMP-SA 的 PUT 流程

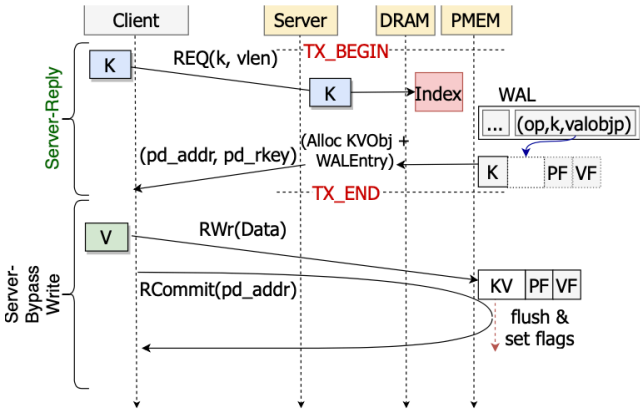

图 6 RDMP-CC 的 PUT 流程

3.1.3 实验结果

实验结果表明RDMP-KV在延迟和吞吐率等方面优于不针对 PMEM 优化的通信机制如 HERD，如图 7 所示，并且优于其他 RDMA-over-PMEM 架构如 Octopus 和 Forca，如图 8 所示。

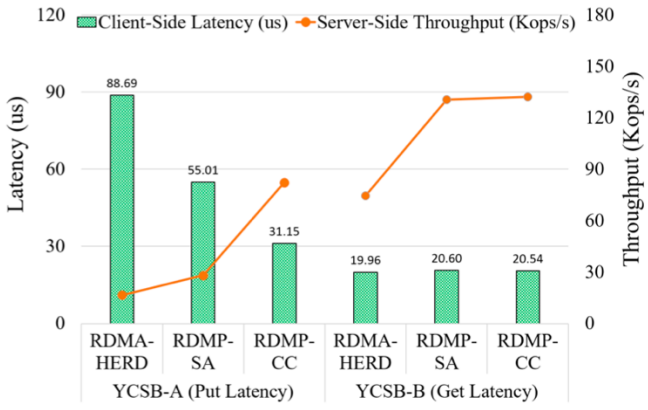

图 7 YCSB 任务测试结果对比

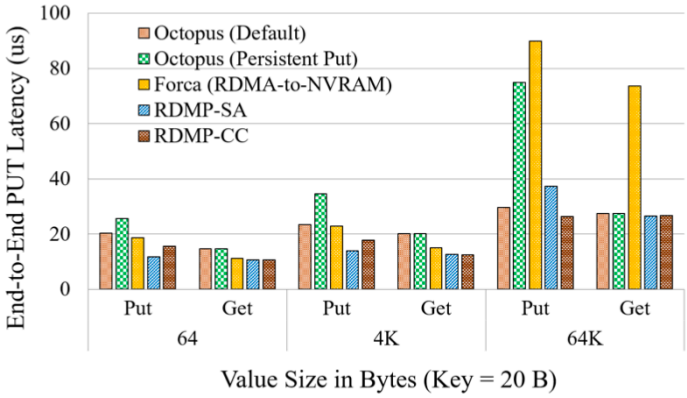

图 8 端到端延迟测试结果对比

3.2 pDPM

3.2.1 介绍

被动式分离持久内存 pDPM[2]（passive disaggregated persistent memory）模型是论文提出的一种新型的分离管理 PM 资源的模型：PM 资源配置在单独的数据节点上，数据节点无需 CPU，仅需 RDMA 网卡，计算节点通过 RDMA 远程管理 PM 资源。

与非分离式系统如图 9 所示相比，对 PM 资源进行存算分离管理有助于提升可扩展性和资源利用率，同时降低成本。但现有的分离式管理 PM 的模型没有利用 RDMA 内核旁路的特性，利用该特性设计分离式管理 PM 资源的系统可以进一步降低成本。论文将现有系统中用于分离管理 PM 资源的模型称为主动式分离持久存储，这些模型将存储节点配置为存储服务器，在存储节点本地配置 CPU 来管理节点内的 PM 资源，如图 10 所示。

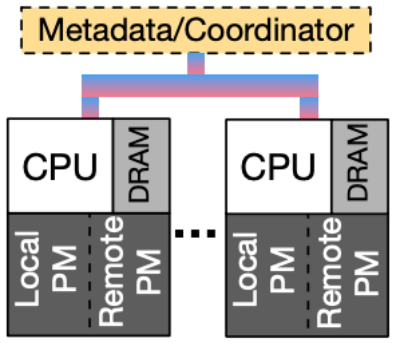

图 9 非分离式 PM 管理（本节图片中蓝色连线表示双边 RDMA 通信，红色连线表示单边 RDMA 通信，红蓝混合的连线表示单边双边混合使用）

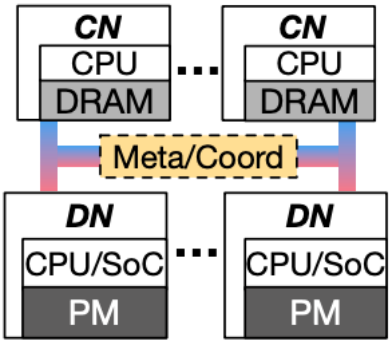

图 10 传统主动式分离 PM 管理 aDPM

3.2.2 系统设计

针对在存储节点以外的何处管理数据这一问题的三种不同解决策略，论文设计实现了三种 pDPM 架构的 KV 存储系统，分别是：由计算节点管理数据，称为 pDPM-Direct；由中央 Coordinator 管理数据，称为 pDPM-Central；混合方法，称为 clover。三种系统的读写过程如图 11 所示。

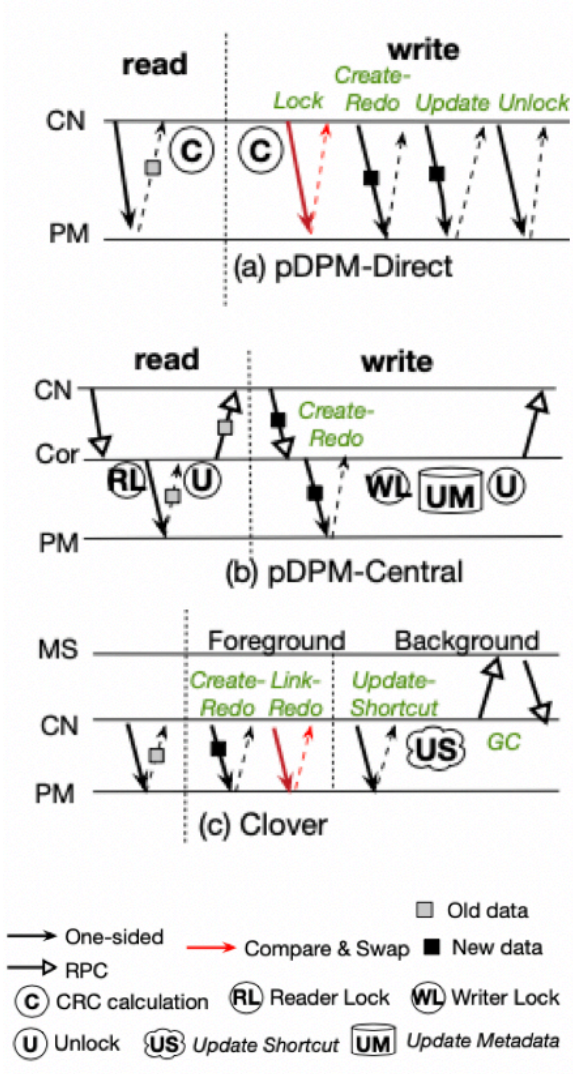

图 11 三种 pDPM 系统的读写过程

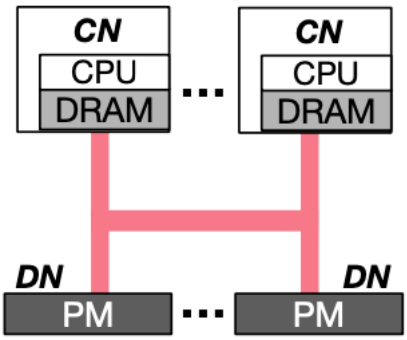

图 12 pDPM-Direct 系统架构

如图 12 所示，pDPM-Direct 系统由计算节点 CN 管理数据，CN 使用单边 RDMA 直接访问 DN 进行管理，数据和元数据都由 CN 处理。为了实现分布式的管理，该系统使用远程锁协调写操作，基于 CRC 冗余校验保证读操作获得有效数据。该系统设计得到的效果包括获取远程锁的开销导致写操作较慢，数据规模较大时计算 CRC 的开销导致读操作较慢，并且高并发时扩展性差。

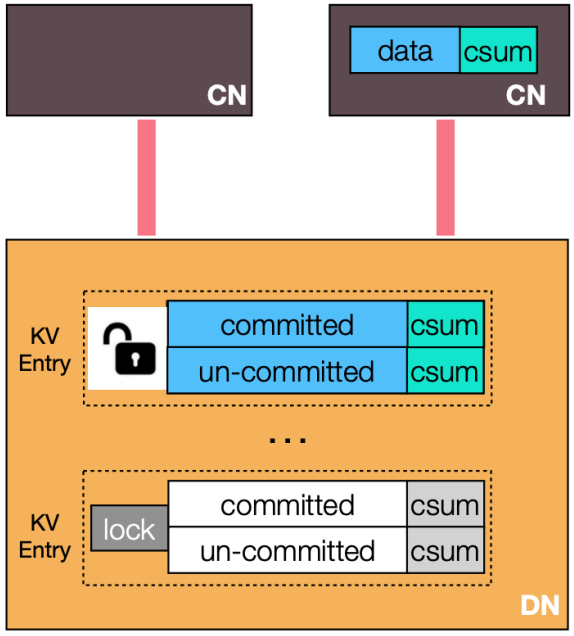

图 13 pDPM-Direct 详细设计

如图 13 所示，pDPM-Direct 系统为每个 KV 对都预先在 PMEM 上分配了一个锁和两个存储空间，两个存储空间一个已提交，一个未提交，每个空间存储的数据后都带有 CRC 冗余校验码。写操作时，计算节点首先用一个 RTT 获取存储节点上的远程锁，然后用一个 RTT 将数据和对应校验码写入未提交空间，再用一个 RTT 将数据和对应校验码写入已提交空间，最后用一个 RTT 释放远程锁。在无冲突的最佳情况下写操作需要 4 个 RTT 和计算冗余校验

码的时间。读操作时计算节点读取已提交空间中的数据和冗余校验码，接着检查校验码判断数据的有效性，如果校验失败，重试。在高并发的情况下可能会多次重试，表现较差。

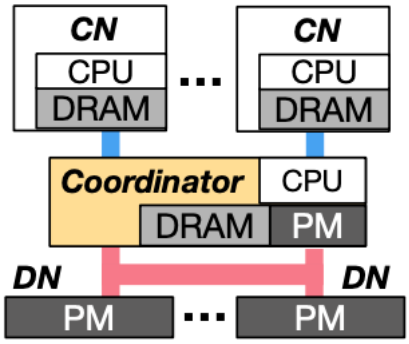

图 14 pDPM-Central 系统架构

如图 14 所示，pDPM-Central 在系统中设置了中央调度器 Coordinator 来管理数据。Coordinator 负责管理 DN 的空间并且使用本地锁序列化处理 CN 的请求。Coordinator 使用单边 RDMA 直接访问 DN，同时和 CN 之间使用双边 RDMA 通信。和 pDPM-Direct 相比，pDPM-Central 的写操作不再需要获取远程锁，读操作也不需计算 CRC，并发请求被 Coordinator 序列化，不再导致各计算节点的多次重复无效操作，性能有所提升。但这种设计仍然存在缺陷：读操作时中央调度器在计算节点和数据节点之间来回传递数据，需 2 个 RTT 并且产生了冗余的拷贝；此外，负载集中在 Coordinator 处，使它成为了整个系统的性能瓶颈。

如图 15 所示，pDPM-Central 在中央调度器 Coordinator 中维护一个映射表，为每个数据条目记录存放的位置并为该位置数据分配一个本地锁。写操作时，首先由计算节点向 Coordinator 发送带有数据的请求，Coordinator 然后为写操作分配空间，把数据写入分配的空间，此时已经写好的数据是一个 redo copy，最后 Coordinator 利用本地锁来原子性地更新数据条目存放的位置。在所有情况下，写操作都需要计算节点向 Coordinator 发请求，Coordinator 向存储节点写数据返回，Coordinator 向计算节点发送完成信号一共 2 个 RTT。在读操作时，计算节点首先向 Coordinator 发送请求，Coordinator 锁定本地位置映射表中的对应条目，然后根据数据存放位置从存储节点读取数据，回复给计算节点。读操作同样在所有情况下都需要 2 个 RTT 的时间，中央调度器在计算节点和数据节点之间来回传递数据，并且产生了冗余的拷贝。此外，负载集中在 Coordinator 处，使它成为了整个系统的性能瓶颈。

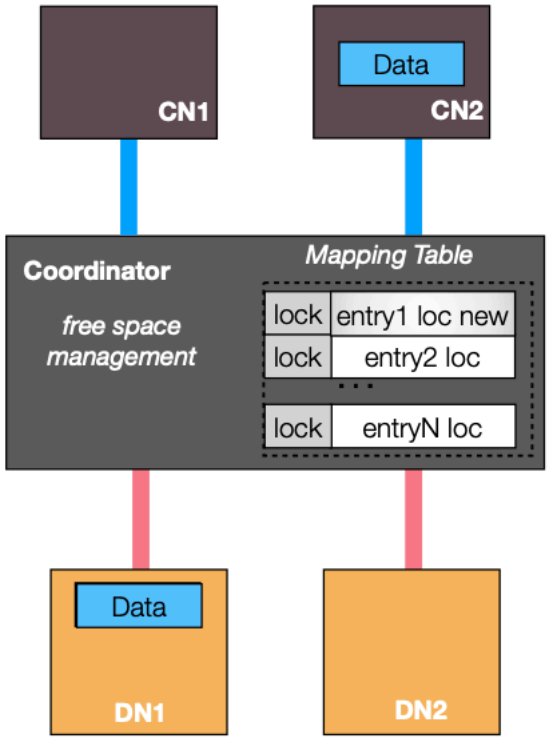

图 15 pDPM-Central 详细设计

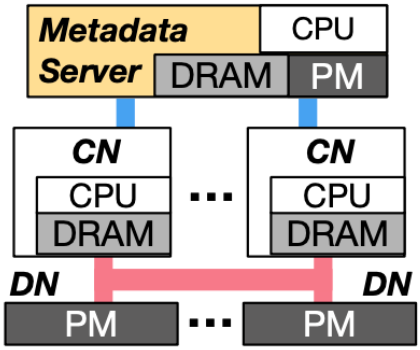

图 16 Clover 系统架构

如图 16 所示，Clover 混合了前述两种数据管理策略，设置了一些元数据服务器来管理元数据，同时让计算节点可以直接访问数据。Clover 分离了数据和元数据：数据层面上，计算节点用单边 RDMA 直接访问存储节点；元数据层面上，计算节点用双边 RDMA 和元数据服务器通信。这样的设计保证了 Clover 优秀的读写性能和可扩展性。

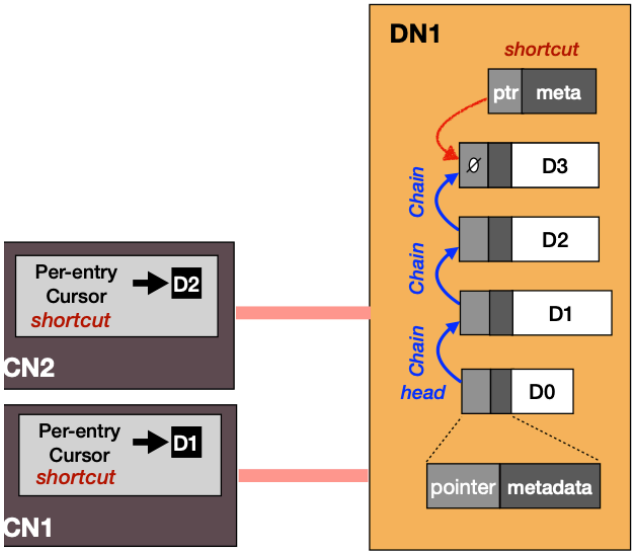

图 17 Clover 系统数据层面设计

如图 17 所示，Clover 在存储节点中将同一个数据条目的不同的版本以链表形式链接起来，每个版本中保存了指向更新版本的指针和数据长度等元数据，最新的数据版本的指针域是空的，同时存储节点中还有一个很可能指向最新数据的捷径指针；计算节点对每个数据条目缓存一个指向该条目某一版本的指针。写操作时，首先在存储节点中分配空间写入数据，此时创建了一个新的版本，然后使用 c&s 方式将新版本链接到原来的最新版本上，如果链接失败，更新条目对应的指针并重试。最好情况下写操作需要写入数据和链接版本 2 个 RTT。读操作时，读取指针所指的版本，如果不是最新版本，更新指针并重试，直到走到链表的末尾，获得最新版本的数据；同时读取捷径指针，根据捷径指针读取数据。两种读操作同时进行，两者中更快的一种操作完成时返回。最好情况下计算节点保存的指针指向最新版本，读操作需要 1 个 RTT。

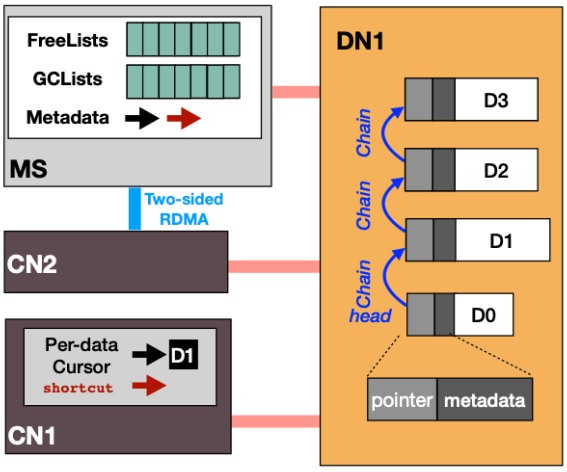

图 18 Clover 系统元数据层面设计

如图 18 所示，Clover 设计了元数据服务器来管理元数据，实现空间管理，垃圾回收和负载均衡的功能。元数据服务器中维护空闲列表、回收列表、数据条目的指针和存储节点的捷径指针。分配空间时，计算节点每次向元数据服务器申请大量空闲空间，元数据服务器以考虑负载均衡的策略从空闲列表中分配空间。垃圾回收的方式是，在写操作后，计算节点异步地回首一批旧版本，元数据服务器将这些空间加入空闲列表中。

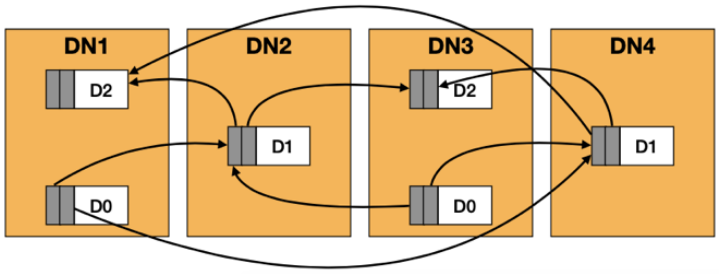

图 19 Clover 系统数据可靠性保障机制

此外，如图 19 所示，Clover 通过将一个版本链接向下一版本的所有数据备份来保证数据的可靠性。通过设置多个 shadow 元数据服务器来保证元数据的可靠性。通过两个层面上的方法来实现负载均衡：元数据服务器和计算节点均控制数据的存储位置；同一个版本的数据备份在不同的存储节点。

3.2.3 实验结果

根据实验结果，pDPM 的读写性能远远优于非分离式系统，如图 20 所示，并且在财务成本大幅降低的同时性能接近主动式分离 PM 系统，如图 21 和图 22 所示。

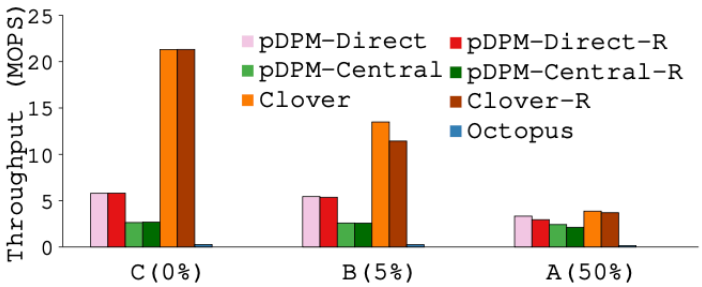

图 20 YCSB 任务吞吐率 pDPM 系统与非分离式系统对比

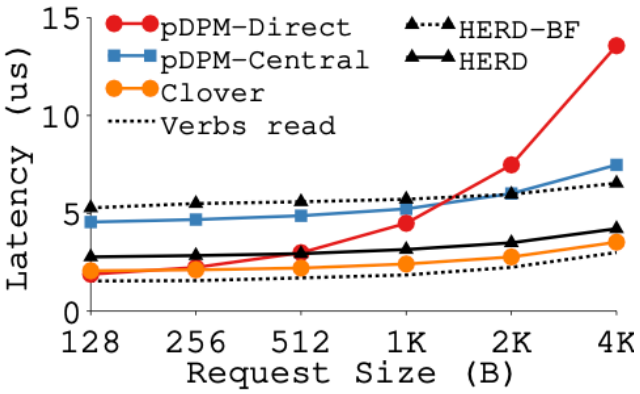

图 21 读延迟 pDPM 系统与 aDPM 系统对比

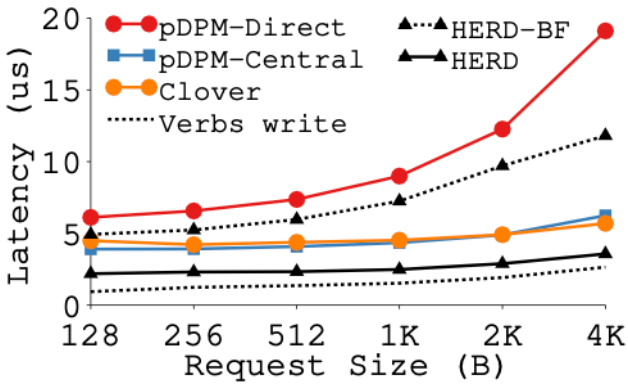

图 22 写延迟 pDPM 系统与 aDPM 系统对比

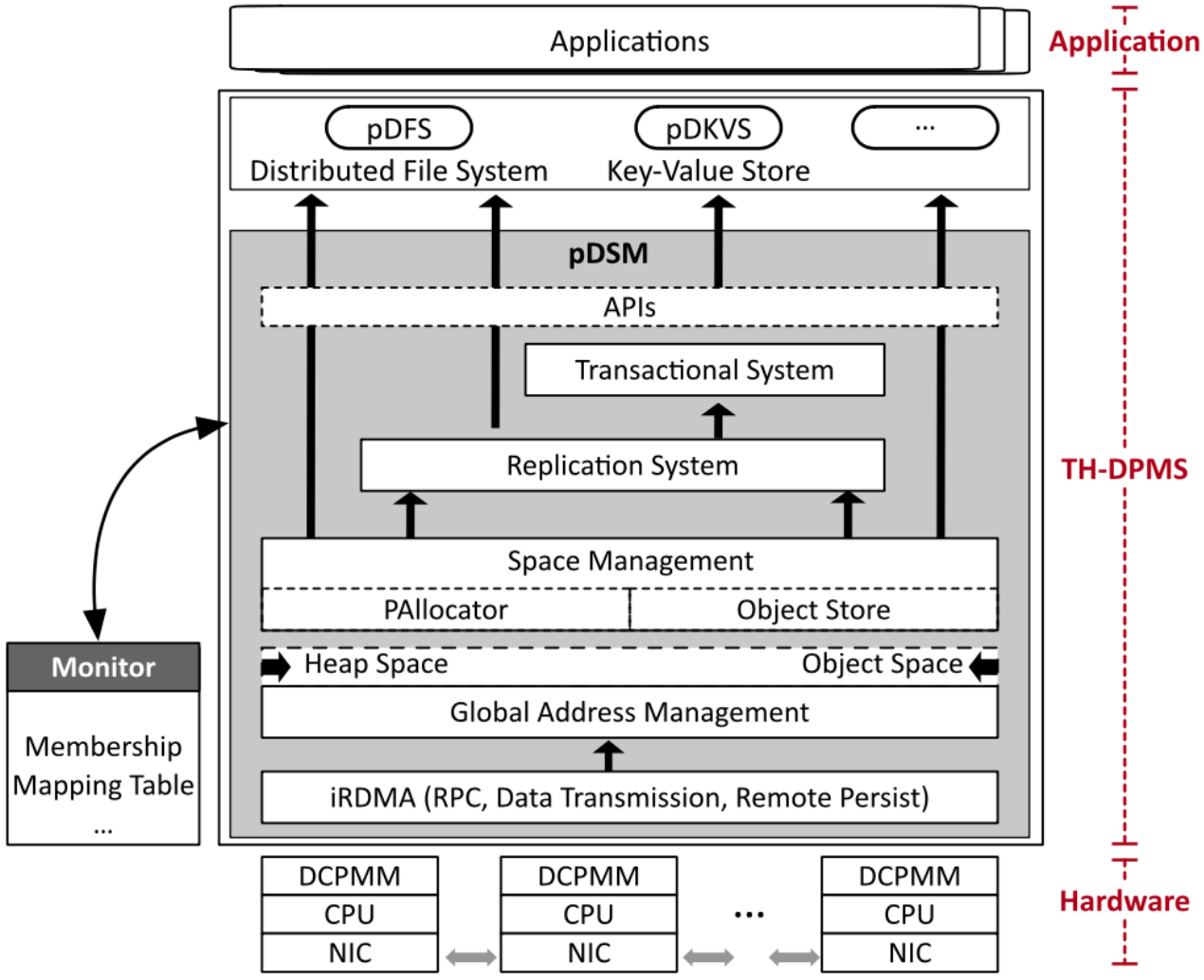

图 23 TH-DPMS 的软件架构

3.3 TH-DPMS

3.3.1 介绍

TH-DPMS[3] (TsingHua Distributed Persistent Memory System) 如图 23 所示，是一个综合分布式存储系统。考虑到数据中心中的文件系统和 KV 存储等不同接口分别实现了相似的功能，例如空间分配、崩溃一致性的保障等，可以将其统一协调以减少冗余。此外，由于 NVM 的数据持久化需要通过

CPU 指令将数据从 CPU 缓存中刷到 PM，而 RDMA 单边访问会旁路远端 CPU，这给 NVM 的 RDMA 单边写操作的持久化带来了问题，需要统一的协调调度以确保系统的一致性。同时，为了充分利用硬件的优秀性能，需要更快的软件设计。TH-DPMS 系统设计了基于 RDMA 网络管理分布式 PM 的抽象层 pDSM (persistent distributed shared memory)，并在其基础上实现了传统接口包括文件系统和 KV 存储。

3.3.2 系统设计

TH-DPMS 的软件架构如图 23 所示，pDSM 实现了一系列的基础功能，包括基于 RDMA 的网络通信（包括远程的数据持久化功能）、全局地址管理、空间管理、备份系统和事务系统，并提供一组应用程序可以直接调用的高速接口。同时，在 pDSM 的基础之上，TH-DPMS 实现了分布式文件系统接口 pDFS 和 KV 存储系统接口 pDKVS 以支持传统应用。

3.3.3 实验结果

实验将 TH-DPMS 与针对普通网络设计的高性能分布式内存对象缓存系统 RDMA-memcached 对比，如图 24 所示。

根据实验结果，在数据规模较小时，RDMA-memcached 系统的数据可以全部保存在内存中，但 TH-DPMS 的性能仍然优于 RDMA-memcached，原因可能包括：首先，RDMA-memcached 系统的 GET/PUT 操作都需要多个 RTT，而 TH-DPMS 的客户端通过巧妙优化实现了零拷贝的 RPC 原语能在一个 RTT 内访问 KV 存储；此外，RDMA-memcached 系统在内存分配和维护 LRU 链表等方面存在扩展性瓶颈。

随着数据规模的扩大，RDMA-memcached 的内存容量不足以存放全部数据，系统不得不将一部分冷数据存放在 SWAP 设备中，因而性能有了剧烈的下降，与此同时 TH-DPMS 在较大数据规模下表现优秀，性能并未显著下降。

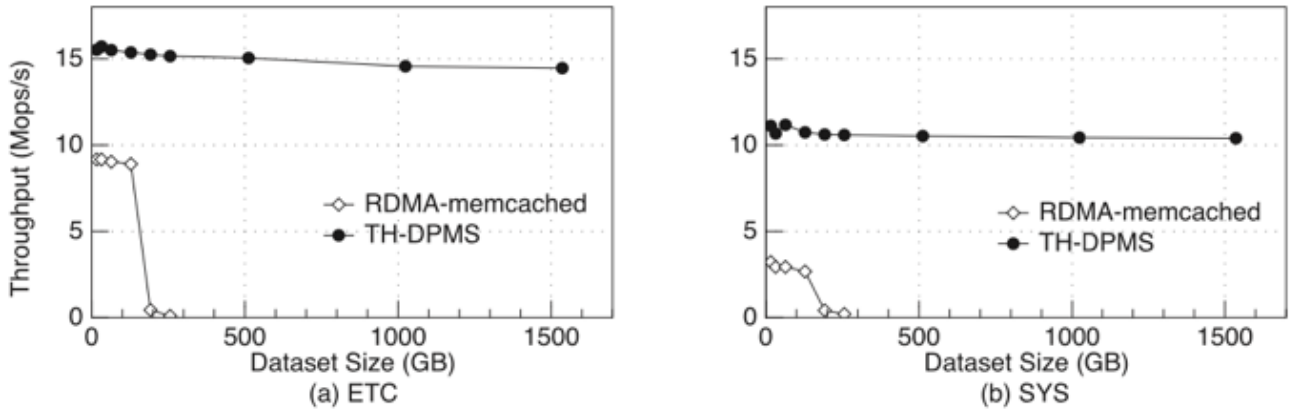

图 24 TH-DPMS 与 RDMA-memcached 对比

4 总结

NVM 和 RDMA 技术提供了优秀的硬件性能，给分布式存储系统性能的提升带来了机遇。高速的网络传输和高速的存储改变了分布式存储系统的设计中，软件开销占比极小，可以忽略的局面。此外，两者的硬件特性，一方面为进一步优化系统提供了可能，例如 NVM 字节编址的特性可以与 RDMP-KV 的新型通信方式结合，提供更优的系统性能， RDMA 内核旁路特性可以用于构建 pDPM 模型的系统，在保障性能的前提下大幅降低财务成本；另一方面，也对系统软件所要实现的功能提出了新的要求，例如 NVM 在写操作时先被 CPU 缓存的特性要求软件设计考虑数据的持久化问题。这都要求分布式系统的软件设计人员考量它们的特点，有针对性地重构软件，并且尽可能地减少系统软件的冗余，才能充分发挥 NVM 和 RDMA 的优势，提高系统的整体性能。

参 考 文 献

[1] T. Li, D. Shankar, S. Gugnani and X. Lu. RDMP-KV: Designing Remote Direct Memory Persistence based Key-Value Stores with PMEM// International Conference for High Performance Computing, Networking, Storage and Analysis (SC). Atlanta, US, 2020: 722-735

[2] Shin-Yeh Tsai, Yizhou Shan and Yiying Zhang. Disaggregating Persistent Memory and Controlling Them Remotely: An Exploration of Passive Disaggregated Key-Value Stores//Annual Technical Conference. 2020: 33-48

[3] Shu Jiwu, Chen Youmin, Wang Qing, Zhu Bohong, Li Junru, Lu Youyou. TH-DPMS: Design and Implementation of an RDMA-Enabled Distributed Persistent Memory Storage System. ACM Trans. Storage, 2020, 16: 31