存储虚拟化研究综述

舒继武 李思阳 张广艳
清华大学

关键词：存储虚拟化 软件定义存储 存储超融合

存储虚拟化是与存储系统相伴而生的。早期人们使用磁盘时，是根据它的逻辑块地址 (Logical Block Address, LBA) 进行存储访问，也就是一种从线性逻辑地址空间到三维空间（柱面、磁道、扇区）的虚拟化。随着存储系统规模的扩大，网络存储技术的发展，存储需求的变化和数据中心的构建等，存储虚拟化作为一种技术，发生了很大的变化。它已由早期的经典存储虚拟化，发展到目前的软件定义存储和存储超融合等，而软件定义存储和存储超融合也往往是与数据中心联系在一起的。存储虚拟化一般在专门的硬件设备上使用，而软件定义存储则没有设备的限制，是以存储虚拟化为基础，其存储具有服务的数据管理功能。超融合架构主要就是计算加存储的一体化方案，提供尽可能的存储就近计算，超融合架构的一个核心组成部分就是软件定义存储，并通过容器和应用的关系，催生了存储的应用感知和超融合存储的应运而生，其核心也是存储虚拟化的新发展。

存储虚拟化

目前，对于“存储虚拟化”这一概念尚未形成统一的权威定义。简单来讲，存储虚拟化就是对物理存储系统的逻辑抽象。目前的存储虚拟化已应用到存储系统层面，早已大大突破了早期存储虚拟化的内涵和外延。

存储虚拟化的分类与特点

按照国际存储网络工业协会 (Storage Networking Industry Association, SNIA) 的分类方法 [1]，存储虚拟化可以按照三个标准进行分类：虚拟化对象，虚拟化发生位置，虚拟化实现方式（如图 1）。虚拟化对象可以包括：磁盘、磁带系统、数据块、文件系统、文件或记录。按照虚拟化发生位置来分，存储虚拟化分为基于主机的虚拟化、基于存储的虚拟化和基于网络的虚拟化。就虚拟化实现方式来说，根据存储虚拟化中传输元数据的控制路径与传输数据的数据路径是否相同，可以分为带内 (in-band) 虚拟化和带外 (out-of-band) 虚拟化。

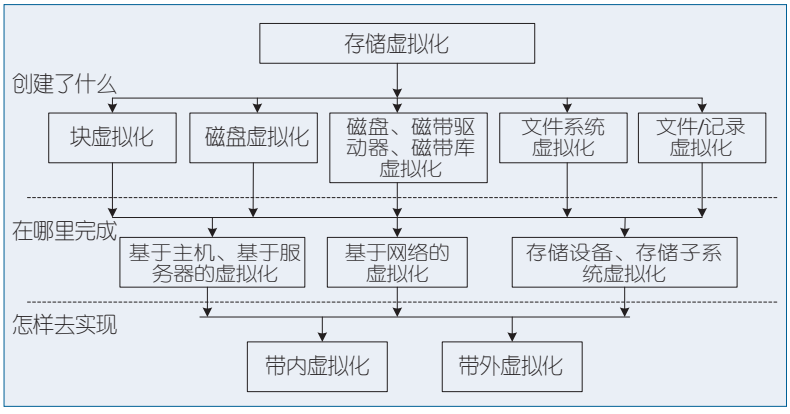

图1 SNIA对存储虚拟化的分类法[1]

SNIA 对存储虚拟化的分类法，表明存储虚拟

化正在以不同形式适应着各种用户环境的需要，也反映了存储虚拟化技术的复杂性。但是，各类存储虚拟化也存在着基本的共性：提供一套方法，隐藏底层存储部件的复杂，提供一些高级存储服务。

存储虚拟化可使逻辑设备的能力不再受单个物理设备的限制，可动态扩展逻辑存储设备的容量和性能，并可采用多种方法对整个系统的存储资源进行优化使用，降低总体拥有成本[2]，从而能够全面提升存储系统的服务质量。

存储虚拟化的研究进展

存储虚拟化催生了许多新技术、新产品和新公司。早在1987年，加利福尼亚大学伯克利分校的帕特森(Patterson)等人就提出了独立冗余磁盘阵列(RAID)技术[3]，其最初目标是增强存储性能，并提供磁盘失效后的数据可恢复性。RAID技术的提出，是存储虚拟化发展的里程碑。

存储虚拟化因其广泛采用而吸引了大量研究者的目光。这些主要研究工作可分为三类：

1.数据布局优化

已有的数据布局优化大致可为两种：(1)在所有磁盘之间均匀分布数据块和校验块，例如，RAID-5战胜RAID-4；(2)利用数据局部性，例如，左对称RAID-5可以将k个连续的数据块访问分散到k个不同的磁盘上[4]。

研究者提出的校验分散(parity declustering)布局[5]，利用尽量少的磁盘来进行数据恢复，以便其余的磁盘能够服务应用I/O请求。这种布局被进一步扩展和优化，例如，被用在Panasas文件系统中。

也有一些工作设计对负载特征或者应用场景敏感的数据布局。比如磁盘缓存磁盘DCD[6]使用额外的一块磁盘作为缓存，将小的随机写转化成大的日志写。惠普公司的AutoRAID[7]将存储空间分割成RAID-1和RAID-5两种，通过区别对待读写操作来实现提高存储带宽且降低数据冗余开销。ALIS[8]和BORG[9]识别出频繁访问的数据块和数据块序列，并把它们以连续的方式放在一块专用区域里。

2.存储重构优化

认识到缩短脆弱的数据重构窗口的重要性，已有研究提出了改善重构性能的诸多方法。首先，一些研究工作集中于在盘组范围内设计更好的数据布局。例如，卡恩(Khan)等人提出了一种循环Reed-Solomon编码[10]来最小化数据恢复和降级读所需的I/O操作。梅农(Menon)和马特森(Mattson)提出了一种称作分布式空闲盘的技术[11]，利用并行的空闲盘来提高存储性能。其次，还有一些方法优化存储重构的工作流程，如面向磁盘的重构(DOR)[12]和流水化重构(PR)[13]等方法。最后，一些任务调度技术[14]可以用于优化存储重构的速率控制。国内学者也提出了基于优先级的多线程重构优化(PRO)[15]、并行的倾斜子阵列(S2-RAID)结构[16]、降级RAID集的快速重构方法[17]等存储重构优化。

3.存储扩展优化

存储系统的扩展效率也是一些研究者一直在探索的问题。当有新盘加入一个存储系统中，需要迁移数据来重新获得一致的数据分布。第一类存储扩展方法是使用随机RAID[18]，这类方法显著减少了数据迁移量，但是多次扩展之后就会产生数据分布不均衡的问题。

更多的存储系统使用确定性的布局来组织数据。如冈萨雷斯(Gonzalez)等提出一种逐步同化的算法来控制RAID-5的扩展执行开销[19]，但该方法为保证数据一致性而采取的逐一、串行的数据迁移方式和元数据的同步更新，导致数据迁移效率较低。

清华大学张广艳等人发现了循环RAID扩展过程中的可乱序窗口特性[20]，进而提出了一系列高效扩展方法[20~24]。该方法显著提高了存储系统的扩展效率。最近，米兰达(Miranda)等提出了一种利用负载信息的扩展方法CRAID[25]，用一个专用的缓存系统来捕获频繁访问的数据，扩展时只将这部分热数据在所有磁盘之间进行重新分布即可。

软件定义存储

随着存储系统的不断发展，多样性（存储介质的多样性，存储协议的多样性[26]，存储形式的多样

性）和复杂性使得对于存储系统的管理越来越复杂，特别是存储系统在软件和硬件上的多样性，都加剧了存储系统管理和扩展的复杂性，这主要体现在如下几个方面：首先是现有的磁盘设备难以进行扩展，如对于存储域网络 (Storage Area Network, SAN) 存储的扩展 [27] 往往要花费昂贵的资金更换机头；其次是存储虚拟化导致的软件栈层次变多，增加了额外的处理开销，从而对系统的服务质量 (QoS) 产生负面的影响。更为重要的是，在传统的模式下，软件针对不同的存储接口需要在底层对整个存储进行替换，如文件接口、对象接口和块接口都分别对应着不同的存储介质。软件定义存储 (Storage Defined Storage, SDS)[28] 就是用来处理传统数据中心面临的诸多挑战的。软件定义存储将数据的控制层和存储层分离。控制层主要负责管理存储的软件资源，而数据层则负责管理存储的基础架构，对数据中心的不同存储设备进行抽象，为整个数据中心提供通用的、软件定义的存储访问接口，从而减少数据中心管理的复杂性。软件定义存储为数据中心复杂的存储系统提供了抽象的存储接口，使得用户不必要去关心底层复杂的系统架构，从而成为很多存储厂商研究的热点。

软件定义存储的层次与特点

如图 2 所示，现有的软件定义存储 [29] 主要分为三个层次：应用层，控制层，存储架构层。存储架构层负责将不同类型的存储介质和系统进行抽象和整合，为控制层提供抽象的存储资源。控制层负责对存储资源进行分类和划分，根据数据中心的需求对存储资源进行分配。应用层为各种应用提供不同的存储接口，以满足各种应用对于存储的需要。

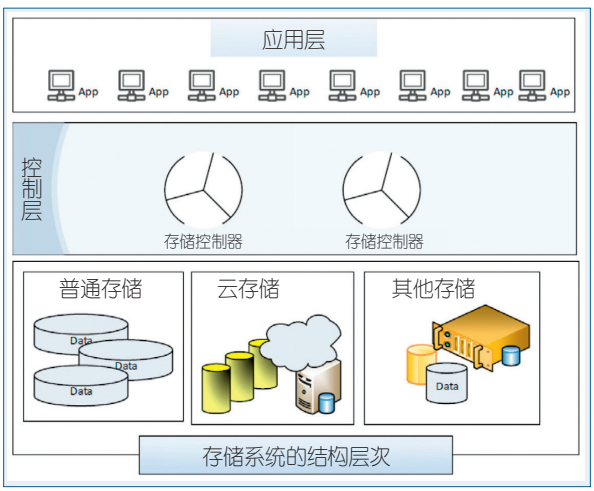

图2　软件定义存储的层次[29]

现有的软件定义存储具有如下特点：

(1) 横向扩展的架构：能够支持存储空间动态的增加和减少；(2) 统一的硬件：能够支持加入不同的存储资源，包括本地的存储设备和网络的存储设备 [30]；(3) 基于资源池的管理：所有的存储资源都应该被整合为一个统一的逻辑空间，并且可以根据资源的需求动态地进行分配；(4) 存储资源抽象：能够将所有的存储设备整合为一个统一的资源，并且支持各式各样的存储接口；(5) 自动化：能够根据用户需求自动地定义不同类型的存储并进行动态扩展；(6) 可编程：提供众多的 API 接口供控制访问存储资源，能够让用户的应用程序自动化地完成对于存储资源的管理；(7) 自定义的存储策略：可以对不同的用户提供不同的安全性、可靠性和服务质量。

软件定义存储的现状

现有的软件定义存储方案都将控制层和数据管理层分开，存储公司或厂商的不同在于其性能、容量和扩展性等方面，主要有以下几种解决方案：

IBM 的 Storwize [31]，其主要特点是有众多的 API 来支持虚拟化环境，帮助企业高效处理海量增长的数据。在存储层，提供了文件和块的存储访问接口，并为云存储提供了可扩展的存储管理。

EMC 提出了存储即服务的概念，主要的特点是简化了存储管理的复杂性，可在存储资源池中直接提供文件、对象和块的访问，并能够动态地加入新的存储阵列。对用户而言，不同的虚拟机为不同的用户提供抽象的存储访问。此外，EMC 的存储虚拟化软件平台 ViPR 提供了开放访问的 API 接口，不同于 IBM 提供的上层接口，EMC 还提供了下层的接口，供不同的存储厂商或者企业适配各自的硬件存储产品。在抽象的存储访问接口方面，ViPR 还支

持用于大数据分析的 ViPR-Hadoop 接口。

Nexenta 提出了 SMARTS[33] 的解决方案，其主要特点是提供了丰富的安全和可靠性接口，实现了存储设备的克隆、快照、备份、端到端的校验，自适应数据恢复等诸多功能。此外，还提供用于用户访问的 GUI 接口，提供较好的扩展性，且能够集成到 OpenStack、VMware 等云服务基础架构设施中。

Atlantic USX[34] 是一个主要为虚拟机提供存储平台的软件定义存储方案，通过在存储介质和虚拟机之间构建一个虚拟层，用 HyperDup Content-aware 服务实现数据的重复数据删除和压缩，并且通过优化虚拟机和物理机之间的 I/O 通道，实现虚拟机 I/O 的高效访问。此外，USX 提供了整合内存和外存的方案，能够为如远程桌面、XenApp 和威睿 (VMware) 的 Horizon 提供高效的访问。

Ceph[35] 是一个开源软件定义存储解决方案，其特点是利用 crush 算法实现了存储的高效可扩展性，通过在底层实现一个虚拟的对象存储层，能够为上层提供文件、对象和块存储。Ceph 支持对于数据的压缩、克隆、容错等诸多方案，在存储介质方面，可以方便地利用各种异构的存储组成一个统一的虚拟存储空间。Ceph 目前被广泛应用于 OpenStack，能够为虚拟机提供高效的存储访问。

Gluster[36] 也是一个开源软件定义存储解决方案，其特点在于支持各种各样的模块，不同模块之间可以任意组合。Gluster 是一个没有元数据的分布式存储，通过基于目录的动态哈希算法，可以将节点扩展至上万个，被广泛应用于超大规模的超算中心和数据存储中心。

除了上述的软件定义存储之外，还有如 Maxta[37]、HITACHI[38]、Datacore[39] 和 CloudBytes[40] 等软件定义存储方案。但这些方案还不能完全覆盖软件定义存储的需求。例如，并不是所有的软件定义存储都能够提供完整的文件、对象和块的访问，或者提供一个统一的存储访问空间。

总之，现有软件定义存储都尝试在可扩展性、安全性、可靠性和经济性等方面提供解决方案，但依然面临一些挑战。

软件定义存储的挑战

到目前为止，软件定义存储还没有明确的定义。构建一个软件定义存储对整个资源的整合和协调能力提出了巨大的挑战，主要包括：(1) 动态地分配不同的数据接口，提供块存储、文件存储和对象存储；(2) 支持数据的迁移；(3) 支持数据的可靠性；(4) 支持高级的 API 供用户使用；(5) 支持数据压缩和副本；(6) 支持存储服务质量保证；(7) 提供高效的元数据访问；(8) 提供容错性和可靠性；(9) 提供系统监控。

在未来，随着存储硬件和互联网的发展，软件定义存储还将面临新的问题。如与**新型存储设备的结合**。以 3D Xpoint[41] 为代表的新型非易失存储将被广泛使用到存储系统中，随着存储硬件延时的不断降低，软件因存储带来的开销将越来越大。与**高速网络设备结合**。软件定义存储也需要适应一些高速网络设备的发展而做新的调整。与**物联网的结合**。传统的数据存储系统将很难适应物联网对于数据高并发、高速持久化的需求。

存储超融合

在数据中心中，往往需要构建单独的虚拟计算平台和单独的存储平台，计算平台和存储平台相互独立。其中最为典型的就是 Ceph[35] 和 OpenStack[42] 的虚拟化和存储整合方案。它们最大的特点就是存储和计算分离，是独立的模块。这种架构的优点是可以分别对存储和计算进行扩展。但缺点也是显而易见的，首先，分别部署两套系统，增加了额外的管理负担和成本；其次，存储计算分离的架构使得数据 I/O 通道变长，系统性能难以优化。构建一个数据中心往往需要综合考虑各个软件堆栈的层次协调，消耗巨大的管理成本。为此，业界提出了超融合 (Hyper-converged) 的概念，其中超 (Hyper) 的本质意义是虚拟化，主要体现在由虚拟机提供计算资源，由软件定义网络 (SDN)[43] 对虚拟机进行组网，由软件定义存储 (SDS) 为虚拟机提供存储。

存储超融合的特点

在传统的虚拟机云平台方案中，计算虚拟化和网络虚拟化是融合的，如 OpenStack 云平台中就实现了软件定义存储和虚拟机计算的融合。所以超融合中最为本质的是加入了软件定义存储，主要以分布式文件系统、分布式块存储、网络附加存储 (Network Attached Storage, NAS) 集群等为代表。但超融合并不是将传统的软件定义存储直接与虚拟机计算组合，而是将计算、网络和存储融合在一个统一的平台中，减少管理的复杂性。最为典型的就是一台基于 X86 架构的物理设备能够提供整套的虚拟化方案，并且实现计算、网络的同步横向扩展，实现数据中心的快速部署。如图 3 所示，SimpliVity 公司的超融合概念中，将传统的各个层次的软硬件融合到了一台服务器中。

超融合的概念最初由 Nutanix[44] 公司提出，Nuanix 的整合方案是在一台物理机中同时提供分布式存储、虚拟网络和虚拟机，并且多台物理机可以进行横向扩展，支持数据在多台物理机之间进行共享、复制、容错和压缩等。超融合的方案只需要直接增加物理机器就可以实现数据中心的扩容，不需要进行复杂的软件配置和开销。超融合方案为小型计算中心的构建提供了快捷简便的通道，在商业上具有巨大的价值。同时，超融合中需要研究的数据高效访问机制也成为了业界研究的热点。

超融合的现状

根据 IDC 的定义，超融合系统是一种新兴的集成系统，其本身是将核心的存储、计算和网络功能整合到单一的软件解决方案或者设备中。现有的超融合系统主要是为了实现以下几个目标：(1) 按需扩展：数据中心可以根据业务的需求不断更新硬件，实现物理资源的高效利用；(2) 快速部署：将软件和硬件融合在一个单独的机器中，机器交互后即可直接使用，无需繁杂的部署；(3) 易于管理：提供了统一的管理界面，能够对计算、存储和网络资源进行统一管理，无需分部门进行运维；(4) 弹性扩展：超融合必须支持分布式的架构，能够实现性能和容量的线性扩展，无节点限制，无单点故障，支持备份、容错和删除。

为此，现有的超融合方案有两种类型。

一是纯软件的解决方案。其特点是支持在现有的硬件上实现存储资源的整合，部署时，只需要在现有的硬件上配置软件后即可快速使用。其中典型的是 DataCore 公司的 SANsymphony[45] 和 EMC 的

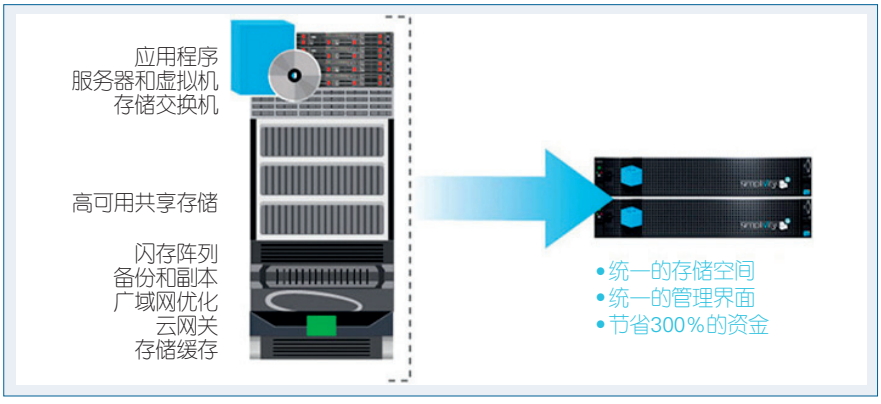

图3　SimpliVity公司的超融合概念

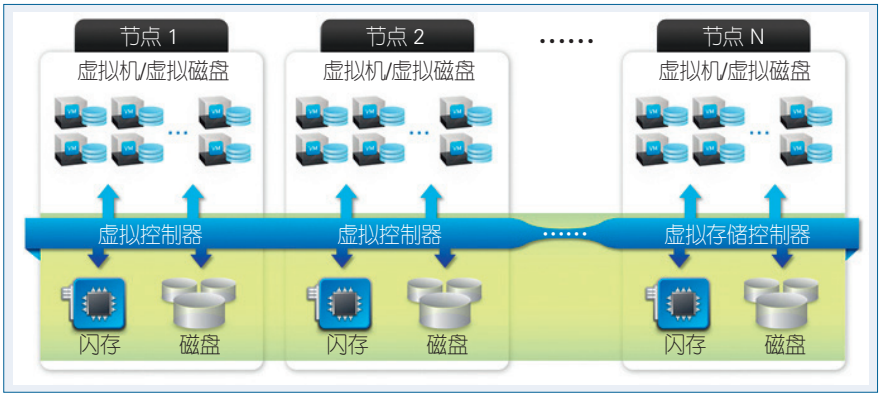

图4　Nutanix的超融合架构[44]

Scale IO[46]。

二是软硬件结合的架构。其特点是将软件和硬件整合到同一台服务器。其中的典型代表是VMware的VSAN[47]和Nutanix[44]。

以Nutanix的超融合架构为例（见图4），现有方案中，最为核心的是提出了虚拟控制器(Controller VM)的概念，其主要用于替代SAN存储中的控制器。这种虚拟控制器的主要作用有两方面：一是多个节点的虚拟控制器能够组成一个分布式的存储管理模块，提供数据的备份、重复数据删除和副本的功能；二是存储控制器通过虚拟机中的数据能够直接访问控制器，从而避免了传统的复杂软件堆栈开销，在虚拟控制器中，往往通过加入闪存作为本地缓存，支持虚拟机对于数据的低延迟、高并发访问。

存储超融合的挑战

现有的存储超融合方案也面临着一些挑战，主要体现在：

1. 存储的横向扩展。超融合架构关键特征之一就是易于扩展、最小部署和按需扩容。超融合中计算能力、存储性能和容量是同步扩容的，无法满足现实中单项能力的扩展，有些厂商还对扩容最小单元有要求，扩展灵活性会受到限制。集群达到一定规模后，系统架构复杂性就会呈非线性增加，集群管理变得更加困难，硬件故障和自修复发生的概率也会大大增加。

2. 存储形态单一。主要体现在仅仅提供了块存储访问接口，这种接口在大多数情况适用于虚拟机的访问。但是在数据中心中，往往还需要提供对于文件和数据库的数据持久化业务，而这些业务往往无法在现有的超融合架构上部署，需要添加额外硬件，构建额外的系统。

3. I/O资源与计算资源的协调。随着网络设备和存储设备性能的提升，网络设备和存储设备对于计算资源的竞争将成为导致虚拟机运行不稳定的首要原因。在高速网络和高速存储中，现有的中断机制将不能满足对于延时的要求，从而导致无法完全发挥现有的存储网络硬件的性能。

针对超融合存在的问题，未来的研究可在两个方面展开。

1. 设计新的CPU资源调度算法，协调网络、存储和计算的处理。

2. 利用基于轮询访问的机制，如DPDK[48]和SPDK[49]，降低存储和网络访问的延迟，提高带宽。 ■

舒继武

CCF杰出会员。清华大学计算机科学与技术系教授。主要研究方向为网络存储/云存储/大数据存储系统与应用、面向新型NVM的存储系统与技术、存储安全与可靠性、并行/分布式处理技术等。
shujw@tsinghua.edu.cn

李思阳

CCF学生会员。清华大学客座博士研究生。主要研究方向为分布式文件系统。
lisiyang@mail.tsinghua.edu.cn

张广艳

CCF专业会员。清华大学计算机科学与技术系副教授。主要研究方向为大数据计算、网络存储、分布式计算。
gyzh@tsinghua.edu.cn

参考文献

[1] Peglar R. Storage Virtualization I: What, Why, Where and How[OL].(2007).http://www.snia.org/education/tutorials/2007/spring/virtualization/Storage_Virtualization_I.pdf.

[2] Brinkmann A, Heidebuer M, Meyer auf der Heide F, et al. V: Drive—Costs and Benefits of an Out-of-Band Storage Virtualization System[C]// *Proceedings of 12th NASA Goddard, 21st IEEE Conf. Mass Storage Systems and Technologies* (MSST '04). 2004:153-157.

[3] Patterson D A, Gibson G A, Katz R H, et al. A case for redundant arrays of inexpensive disks (RAID)[C]// *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*. New York: ACM

Press,1988: 109-116.

[4] Lee E K, Katz R H. The performance of parity placements in disk arrays[J]. IEEE Transactions on Computers, 1993, 42(6): 651-664.

[5] Muntz R R, Lui J C S. Performance analysis of disk arrays under failure[C]// Proceedings of the 16th VLDB Conference. 1990: 162-173.

[6] Yang Q, Hu Y. DCD—Disk Caching Disk: A New Approach for Boosting I/O Performance[C]//Proceedings of the 23rd annual international symposium on computer architecture. New York: ACM Press, 1996: 169-178.

[7] Wilkes J, Golding R A, Staelin C, et al. The HP AutoRAID hierarchical storage system[J]. ACM Transactions on Computer Systems, 1996, 14(1): 108-136.

[8] Hsu W W, Smith A J, Young H C, et al. The automatic improvement of locality in storage systems[J]. ACM Transactions on Computer Systems, 2005, 23(4): 424-473.

[9] Bhadkamkar M, Guerra J, Useche L, et al. BORG: Block-reorGanization and self-optimization in storage systems[R/OL]. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=0B8A311ED574DD158306877C1174895A?doi=10.1.1.89.7216&rep=rep1&type=pdf.

[10]Khan O, Burns R, Plank J, et al. Rethinking erasure codes for cloud file systems: minimizing I/O for recovery and degraded reads[C]// Proceedings of the 10th USENIX Conference on File and Storage Technologies. Berkeley, CA:USENIX Association, 2012:20-20.

[11]Menon J, Mattson D. Distributed sparing in disk arrays[C]// Proceedings of the thirty-seventh International Conference on COMPCON. San Francisco, IEEE CS Press, 1992:410-421.

[12]Holland M C. On-line data reconstruction in redundant disk arrays[D].Carnegie Mellon University, 1994.

[13]Lee J Y B, Lui J C S. Automatic recovery from disk failure in continuous-media servers[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(5):499-515.

[14]LUMB, C. R., SCHINDLER, J., GANGER, G. R., NAGLE, D., AND RIEDEL, E. Towards higher disk head utilization: Extracting ”free” bandwidth from busy disk drives[C]. In 4th Symposium on Operating System Design and Implementation, 2000, 87-102.

[15]TIAN, L., FENG, D., JIANG, H. ,et al : A popularity-based multi-threaded reconstruction optimization for raid structured storage systems[C]. In 5th USENIX Conference on File and Storage Technologies, FAST 2007, February 13-16, 2007, San Jose, CA, USA, 2007, 277-290.

[16]WAN, J., WANG, J., XIE, C., AND YANG, Q. S2-RAID: Parallel RAID architecture for fast data recovery[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25 (6), 1638-1647.

[17]Wu S, Jiang H, Feng D, and et al. Workout: I/O workload outsourcing for boosting RAID reconstruction performance[C]. In 7th USENIX Conference on File and Storage Technologies, 2009: 239-252.

[18]GOEL, A., SHAHABI, C., YUEN DIDI YAO, et al, Scaddar: An efficient randomized technique to reorganize continuous media blocks[C]. In In Proceedings of the 18th International Conference on Data Engineering (ICDE), 2002, 473-482.

[19]GONZALEZ, J., AND CORTES, T. Increasing the capacity of raid5 by online gradual assimilation[C]. In Proceedings of the International Workshop on Storage Network Architecture and Parallel I/Os, 2004, 17-24.

[20]ZHANG, G., SHU, J., XUE, W., AND ZHENG, W. SLAS:an efficient approach to scaling round-robin striped volumes. TOS 3, 1 2007, 3:1–3:39.

[21]ZHANG, G., ZHENG, W., AND SHU, J. Alv: A new data redistribution approach to raid-5 scaling[J]. IEEE Transactions on Computers, 2010, 59(3), 345-357.

[22]Weimin Zheng, Guangyan Zhang (Corresponding author). FastScale: Accelerate RAID Scaling by Minimizing Data Migration[C]. in the Proceedings of the 9th USENIX Conference on File and Storage Technologies (FAST'11), San Jose, CA, 2011.

[23]G Zhang, W Zheng, K Li, Rethinking RAID-5 Data Layout for Better Scalability[J], IEEE Transactions on Computers, 2014,63(11), 2816-2828.

[24]G Zhang, K Li, J Wang, W Zheng, Accelerate RDP RAID-6 Scaling by Reducing Disk I/Os and XOR Operations[J], IEEE Transactions on Computers, 2015, 64(1), 32-44.

[25]MIRANDA, A., AND CORTES, T. CRAID: online RAID upgrades using dynamic hot data reorganization[C]. In Proceedings of the 12th USENIX conference on File and Storage Technologies, FAST 2014, 2014, 133-146.

[26]Xu, Qiumin, et al. Performance analysis of nvme ssds and their implication on real world databases[C]. Proceedings of the 8th ACM International Systems and Storage Conference. ACM, 2015.

[27]Glider, Joseph S., Carlos F. Fuente, and William J. Scales. The software architecture of a san storage control system[J]. IBM Systems Journal 2003,42(2).

[28]Carlson M, Yoder A, Schoeb L, et al. Software defined storage[J]. Storage Networking.

[29] K. Palanivel, B. Li, Anatomy of Software Defined Storage Challenges and New Solutions to Handle Metadata, Report[J], University of Minnesota, 2013.

[30] Gibson, Garth A., and Rodney Van Meter. Network attached storage architecture[J].Communications of the ACM, 2000,43(11).

[31] Choose a storage platform that can handle big data and analytics, Solution Brief TSS03158-USEN-01, IBM Corporation, 2014.

[32] Transform your storage for the software defined data center with emc vipr controller, white paper H11749.4, EMC Corporation, 2015.

[33] Nexentastor solutions guide for mirantis openstack, white paper, Nexenta, 2014.

[34] Atlantis usx, http://www.atlantiscomputing.com/products/atlantis-usx [On- line; accessed Oct-2014].

[35]Weil, Sage A., et al. Ceph: A scalable, high-performance distributed file system[C]. Proceedings of the 7th symposium on Operating systems design and implementation. USENIX Association, 2006.

[36]Subramanian, Krishnan. Gluster Introduces Scale-Out NAS Virtual Storage Appliances for VMware and AWS[OL]. CloudAve online article (2011).

[37]Maxta storage platform (enterprise storage redefined, white paper, Maxta Corporation).

[38]Storage virtualization: How to capitalizeonits economic benefits, whitepaper WP-435-E DG, Hitachi Data Systems, 2015.

[39]Top 3 challenges impacting your data and how to solve them, white paper, DataCore Software Corporation, 2014.

[40]Storage architected for the new-age datacenters, white paper, CloudByte Corporation. Industry Assoc. working draft, 2014.

[41]Cappelletti, Paolo. Non volatile memory evolution and revolution[C]. Electron Devices Meeting (IEDM), 2015 IEEE International. IEEE, 2015.

[42]Sefraoui O, Aissaoui M, Eleuldj M. OpenStack: toward an open-source solution for cloud computing[J]. International Journal of Computer Applications, 2012, 55(3).

[43]Lantz B, Heller B, McKeown N. A network in a laptop: rapid prototyping for software-defined networks[C]// Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. ACM, 2010.

[44]Aron, Mohit, Dheeraj Pandey, and Ajeet Singh. Architecture for managing I/O and storage for a virtualization environment. U.S. Patent No. 8,601,473. 3, 2013.

[45]Symphony, S. A. N. DataCore Software.

[46]EMC. EMC ScaleIO User Guide, Nov 2013.

[47]Hogan, Cormac, and Duncan Epping. Essential Virtual SAN (VSAN): Administrator's Guide to VMware Virtual SAN. VMware Press, 2016.

[48]Pongrácz, Gergely, Laszlo M, and Zoltán Lajos K. Removing roadblocks from SDN: OpenFlow software switch performance on Intel DPDK[C]. Software Defined Networks (EWSDN), 2013 Second European Workshop on. IEEE, 2013.

[49]Kim H J, Lee Y S, Kim J S. NVMeDirect: a user-space I/O framework for application-specific optimization on NVMe SSDs[C]//8th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 16). USENIX Association, 2016.