学号 2017282110225

部分奇偶校验缓存和数据缓存管理方法来提高基于 SSD 的 RAID 的性能

院（系）名 称 ： 计算机学院

班　　级　　： 2017 级硕士 5 班

学 生 姓 名 ： 陈星晖

二〇一七年十二月

摘　要

论文中提出了一种部分奇偶校验缓存和数据缓存管理方法，用于降低基于固态盘（SSD）的廉价磁盘冗余阵列（RAID）系统的奇偶校验更新成本，从而可以使（I/O）RAID 系统的输入/输出性能得到提高。与硬盘驱动器相比，SSD 有许多优点。但是，将 SSD 直接添加到 RAID 系统是不可取的，因为这会降低 SSD 的性能和使用寿命。在 RAID-5 系统中，奇偶校验生成包括对 SSD 的读写操作。每当有新的写入请求到 RAID 时，相关的奇偶校验都必须更新并写入 SSD；这种频繁的奇偶校验更新导致 RAID 性能不佳，缩短了 SSD 的使用寿命。

本文主要基于 2014 年 VLSI 论文《Partial parity cache and data cache management method to improve the performance of an ssd-based raid》，论文中将现有方法和所提出的高效的缓冲区管理方法与数据缓存相结合，所提出的方法减少了用于在 RAID 系统中生成奇偶校验的读取和写入操作的数量。实验结果表明，采用该方法可以使 RAID-5 系统的 I/O 性能提高 76%。

目　录

1　绪论 1
1.1 研究背景 1
1.2 主要研究工作 3
2　RAID 架构耗损问题研究 4
2.1 RAID 架构 4
2.2 奇偶校验和数据缓存管理方案 5
3　有效的缓冲管理研究 7
3.1 缓冲区和缓存的结构 7
3.2 PPC 方法 8
3.3 写入缓存和数据缓冲区的操作 10
4　PPC 实验 13
4.1 实验准备 13
4.2 奇偶校验的开销 14
4.3 RAID/IO 的整体性能 15
4.4 差分参数模拟 17
5　总结 21
参考文献 22

1 绪论

1.1 研究背景

个人电脑（PC）的中央处理器（CPU）的设计已经取得了很多进展。CPU 的内核数量从一个增加到了多个，因此 CPU 的性能有了明显的提高。而且，诸如动态随机存取存储器（DRAM）和视频卡的个人计算机的其他组件也经历了实质性的改进。然而，由于某些物理限制，硬盘驱动器（HDD）（由盘片组成并需要磁头致动器机构）的改进并不明显。近年来，由于闪存芯片价格相对较低，固态硬盘（SSD）变得越来越流行[1]。

如图 1 所示，SSD 由许多闪存芯片组成，并且没有头部致动器机构。比 HDD 更具优势，例如更好的抗振性，更低的功耗和更快的性能。

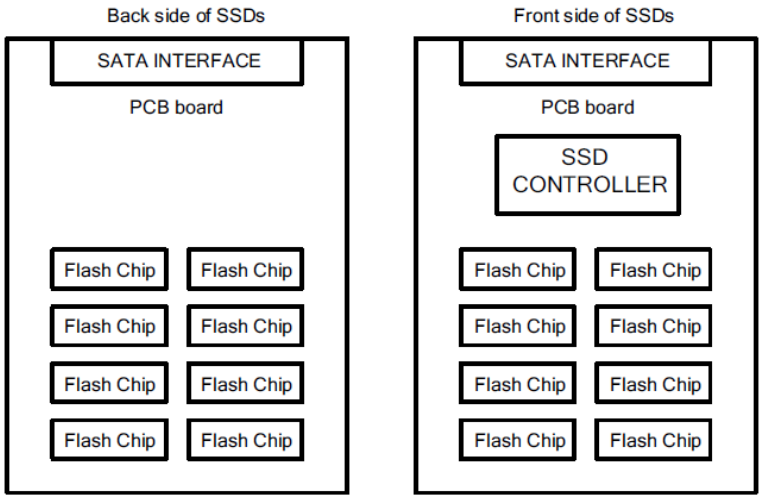


图 1 SSD 的正面和背面

闪存是 SSD 的基本组件。闪存具有以下特点：首先，用于读写操作的单元是页面，但擦除操作的单元是块[2] - [4]。因此，不同操作的速度差别很大，如表 1 所示。其次，在每次擦除操作之后，只能一次写入相同的物理页面。第三，每个块的擦除次数有限。

表 1 NAND FLASH 芯片的规格[5]

Hynix 32GB NAND Flash Chip	
Data Integrity	100,000 erase cycles
Page Read	0.025ms
Page Program Time	0.2ms
Block Erase Time	2ms

当数据写入闪存时，空闲页面的数量变小，因此闪存控制器必须擦除块以回收空闲页面。在控制器擦除块之前，控制器必须将块中的有效数据复制到另一个空闲块；这个操作被称为垃圾收集（GC）。表 1 显示块擦除时间比页面编程时间长 10 倍。因此，写操作，擦除块操作和 GC 需要花费相当多的周期才能完成。因此，减少写入次数的方法对 SSD 来说非常重要。当写入次数减少时，SSD 的寿命延长并且输入输出（I / O）处理速度得到改善。

在数据中心中，由于 HDD 价格低廉，HDD 被用在廉价磁盘冗余阵列（RAID）中作为存储系统。但是，HDD 的功耗和散热是关键问题。例如，为了冷却数据中心，需要全部功耗的 40％[5]。由于固态硬盘比容量相同的固态硬盘价格昂贵，固态硬盘无法取代硬盘，即使耗电量较低，I / O 性能也比硬盘快。因此，有一些公司正在为 SSD 开发 RAID 控制器。未来，当闪存芯片价格大幅下降时，SSD 可以作为存储系统用于数据中心，以降低数据中心冷却成本，提高 I / O 处理速度。此外，数据中心总是使用 RAID 技术来提高性能并确保数据的完整性。

日立和三星等一些公司已经提出了涉及在数据中心中使用 SSD 的解决方案[6]，[7]。日立[6]通过使用串行连接 SCSI（SAS）接口和光纤通道（FC）和 SSD 来开发数据中心。Reinsel 和 Janukowicz [7]提到过，未来数据中心面临的挑战是 SSD 的可靠性和价格。

目前市面上有 RAID-0 SSD 的产品。RAID-0 的 I / O 处理速度非常高，但是 RAID-0 不够可靠。当其中一个闪存芯片损坏时，存储的数据将丢失。从可靠性的角度来看，RAID-4 和 RAID-5 是更好的解决方案，因为当一个存储设备被损坏时，它们可以使用奇偶校验来恢复数据。

表 1 给出了 NAND 闪存芯片的规格[5]。编程时间和擦除时间与读取时间相比非常慢。但是，在 RAID-4 或 RAID-5 中，奇偶校验更新有许多写入操作。但是，在 RAID-4 或 RAID-5 中，奇偶校验更新有许多写入操作。

只要将新数据写入 RAID-4 和 RAID-5 的存储系统，就必须用读取操作更新奇偶校验，以生成新的奇偶校验。然后，新的奇偶校验将写入存储系统。存储系统必须减少奇偶校验生成开销和奇偶校验写入次数。因此，论文中需要一个高效的奇偶校验缓存管理方案来提高 RAID 系统的 I / O 处理速度，同时确保系统的可靠性。

1.2 主要研究工作

论文中结合了先前的部分奇偶校验缓存（PPC）方法和所提出的有效的缓存管理方法和一个新的数据缓存来合并奇偶校验。实验结果表明，使用所提出的方法，读写操作都减少了。此外，文章中添加一个保留旧数据的特殊数据缓冲区。该数据缓冲器可以减少 SSD 的读取操作以进行部分奇偶校验更新，从而可以进一步提高 I / O 处理速度。一种有效的缓冲区管理方法可以减少奇偶校验写入次数，并且所提出的数据缓存可以减少奇偶校验更新所需的读取操作次数。此外，所提出的方法还延长了 SSD 的使用寿命。

论文的其余部分安排如下。第二节讨论了 RAID 架构。第三节讨论相关的作品和他们的问题。第四节介绍了提出的奇偶校验和数据缓存管理方法及其操作。第五节讨论实验结果。第六节提出结论。

2 RAID 架构耗损问题研究

RAID 技术通常用于工作站和数据中心。由于并行数据访问方案，不仅提高了 I / O 性能，而且通过添加奇偶校验数据来保证数据的完整性。

2.1 RAID 架构

RAID-0 使用块级剥离，并将数据同时写入不同的存储设备。RAID-0 具有高速 I / O 性能，RAID-0 可以充分利用存储设备的全部容量。但是，由于在磁盘崩溃时无法使用冗余数据来恢复数据，因此 RAID-0 不够可靠。RAID-1 同时将数据复制到两个不同的设备。因此，可用容量只有总容量的 50％。因此，RAID-1 的成本相当高。

RAID-2 使用汉明码的位级剥离和错误收集。如果有一位出错，那位可以被恢复。然而，错误收集的硬件逻辑方案是复杂的。RAID-3，RAID-4 和 RAID-5 使用额外的存储设备来存储奇偶校验，以便它们可以容忍存储设备的故障。RAID-3 使用带奇偶校验的字节级条带。RAID-4 和 RAID-5 都使用带有奇偶校验的块级条带。RAID-3，RAID-4 和 RAID-5 中的最小存储设备数量为 3。在这三个存储设备中，两个负责存储数据，第三个负责将奇偶校验数据存储在 RAID-3，RAID-4 和 RAID-5 中。

RAID-3 和 RAID-4 将奇偶校验放置到固定的存储设备上。但是，当新数据写入 RAID 系统时，必须计算和更新相关奇偶校验。因此，存储奇偶校验的存储设备具有大量的写入操作，并且该设备一直非常繁忙。由于每块 SSD 的擦除次数有限，存储奇偶的存储设备将在短时间内崩溃。

RAID-5 将奇偶校验放置到每个存储设备中，以便将奇偶校验写入操作分成不同的存储设备。RAID-5 的存储容量利用率和性能是可以接受的。因此，论文中采用 RAID-5 架构是因为上述考虑。

RAID-5 的瓶颈是频繁的奇偶更新。例如，如图 2 所示，奇偶校验 P0 由 D0⊕D1⊕D2⊕D3 生成。符号“⊕”代表异或运算符。当新的数据被写入时，例如 D1，奇偶校验 P0 必须被更新。因此，无论在同一条带中更新的数据量如何，都必须重新计算奇偶校验，并写入存储设备。奇偶校验生成的成本包括从存储设备读取旧数据的读取操作和写入操作。

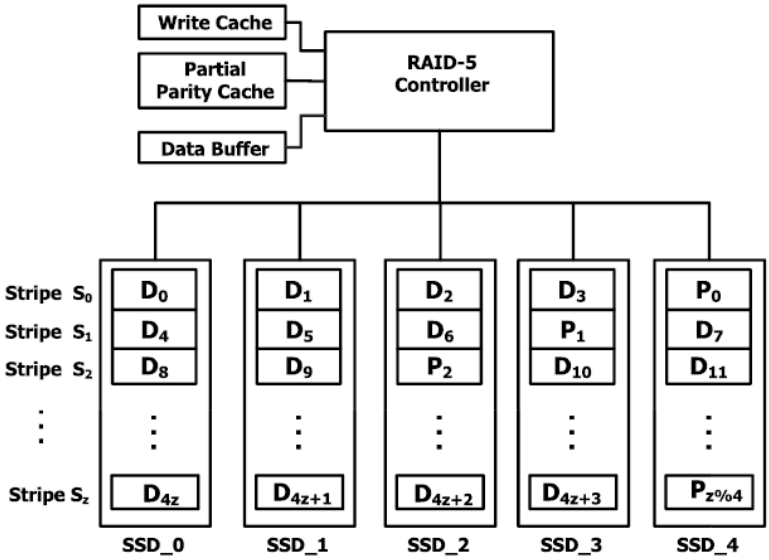


图 2 推荐的 4 + 1 RAID-5 架构

论文总将 RAID-5 表示为 n + 1，其中 n 表示数据存储设备的数量，“1”表示奇偶校验存储器。例如，图 2 中显示了 4 + 1 RAID-5 架构。

2.2 奇偶校验和数据缓存管理方案

一些早期的作品使用 SSD 和 HDD 来构建混合 RAID 系统。例如，基于混合奇偶校验的磁盘阵列（HPDA）使用 SSD 来存储数据，使用两个 HDD 来存储奇偶校验，并成为写入缓冲区。SSD 和 HDD 是使用 RAID-4 架构构建的。奇偶校验磁盘 HDD 和另一个 HDD 的剩余空间是通过使用 RAID-1 作为写入缓冲区来构建的。如果写入请求是连续的，则这些请求将直接写入到 RAID-4 中。相反，如果这些请求是随机访问的，则这些请求被写入写入缓冲区。当 I / O 空闲时，写入缓冲区中的请求被写回到 RAID-4。HPDA 使用硬盘来解决 RAID-4 的奇偶校验磁盘频繁的奇偶更新问题。但是，由于 HDD 的 I / O 处理速度比 SSD 慢，从而延长了 SSD 的使用寿命，但 RAID 性能却下降了。

当 SSD 直接部署到 RAID 架构时，存在许多问题。在 RAID-4 架构的情况下，奇偶校验盘比其他盘有更高的写入次数。实验结果表明，写操作集中在奇偶校验磁盘上，因此他们的解决方案使用 HDD 来替代 SSD 奇偶校验磁盘。

在 RAID-5 架构的情况下有一种磨损均衡的方案，动态地放置奇偶校验数据，并创建一个用于记录奇偶写入次数的 k 位映射表。当一个磁盘的写入时间大于特定值时，磨损均衡方案将这些奇偶校验数据与具有较低写入时间的其他奇偶校验

数据交换。耗损均衡方案仅平衡 SSD 的写入次数，但是频繁奇偶校验更新问题的根本原因尚未解决。

为了解决 RAID-5 架构中频繁的奇偶校验数据更新问题，以前的研究增加了一个奇偶校验缓冲区来减少奇偶校验数据的写入次数。这种方法被称为“延迟奇偶校验更新方案”。奇偶校验数据保存在奇偶校验缓冲区中，直到满足某些特定条件。延迟奇偶校验更新方案肯定可以减少奇偶写入次数。它们减少了奇偶校验数据的写入次数，但不考虑奇偶校验生成开销。

PPC 方案也采用延迟奇偶校验更新方案，但是它产生了部分奇偶校验并将其存储到缓存中。当缓存已满时，必须将部分奇偶校验重建为完整奇偶校验并写入 SSD。然而，奇偶校验写入时间几乎与其他奇偶校验缓存方式相同。写操作需要相当多的周期，如表 1 所示。因此，如果可以减少奇偶校验数据的写入次数，则可以显着改善整个 RAID 性能。而且，当 PPC 方案更新部分奇偶校验时，需要从存储设备读取旧数据，因此部分奇偶校验更新的开销应当减少。

3 有效的缓冲管理研究

推荐的基于 4 + 1 SSD 的 RAID-5 架构如图 2 所示。主机系统向存储接口发送读取或写入请求，并且 RAID-5 控制器处理写入缓存并分配对 SSD 的数据访问。写入缓存保存来自主机系统的数据，并且使用非易失性随机存取存储器（NVRAM）作为写入缓存。PPC 存储了部分奇偶校验，它还使用 NVRAM 来避免在突然断电的情况下丢失数据。数据缓冲区用于降低部分奇偶校验更新的成本。数据缓冲区中存储的数据可以从 SSD 重新加载，因此可以使用静态随机存取存储器（SRAM）或 DRAM 来实现该缓冲区。

部分奇偶校验仅包含完整奇偶校验的一部分。PPC 可以帮助合并奇偶校验写入操作，例如，如果在高速缓存中存在 P0 至 P16 部分奇偶校验。写入到条 S0-S16 的新数据可以合并它们在 PPC 中的奇偶校验写入操作。因此，SSD 的写入操作的数量可以显着减少。

论文中提出了一种有效的缓冲区管理方法，以避免 PPC 被占用。当 PPC 已满时，必须选择其中一个部分奇偶校验并将其写回 SSD。但是，RAID 控制器需要读取 SSD 中的关联数据以建立完整的奇偶校验。另外，奇偶写操作需要相当多的周期。因此，所提出的有效的缓冲区管理方法可以降低奇偶校验更新的成本。

3.1 缓冲区和缓存的结构

图 3 显示了在所提出的 RAID-5 控制器中的 PPC，数据缓冲器和写入缓存的数据结构。变量 m 表示 PPC 和数据缓冲器中的条目数量，w 表示写入缓存中的条目数量。T 代表 RAID-5 中的总条数，n 代表 n + 1 RAID-5 架构中的存储设备数量。

部分奇偶校验的数据大小是一页。n 位字段表示与部分奇偶校验相关的数据。另外，关联的数据被存储在数据缓冲器中。例如，条带号（S0）的 4 位二进制值“1100”意味着用 D0 和 D1 产生部分奇偶性（P0）。另外，D0 和 D1 存储在数据缓冲区中。

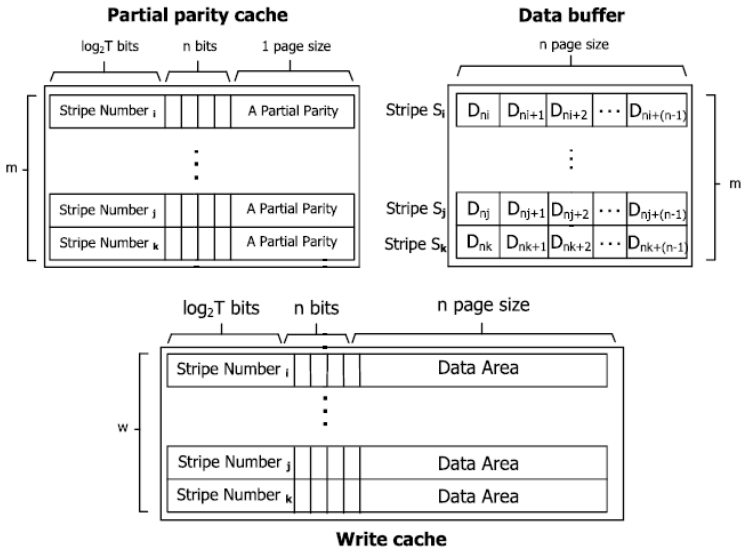


图 3 PPC，数据缓冲区和写入缓存

3.2 PPC 方法

图 4 显示了 PPC 提出的操作流程图。主机将数据传输到 RAID 控制器。RAID 控制器确定写入缓存是否空闲。如果写入缓存空闲，则数据可以缓存在写入缓存中，并且从主机系统的角度来看，写入操作完成。但是，如果写入缓冲区已满，则 RAID 控制器必须从写入缓存中选择 victim 条带，并将数据写回存储设备。

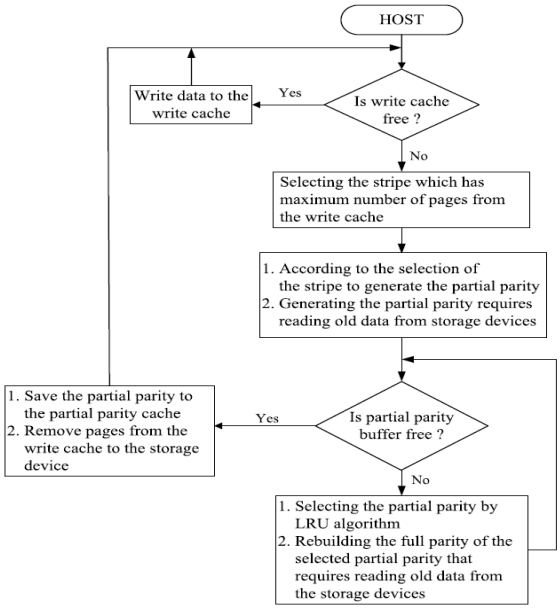


图 4 PPC 流程图[10]

victim 条带的选择涉及找出写入缓存中的哪个条带包含最大量的数据。当条带包含大量数据时，可以将这些数据并行写入到 SSD 中。另外，所选条带中的数据

用于产生部分奇偶校验。

在产生部分奇偶校验后，控制器检查 PPC 是否空闲。如果 PPC 空闲，则将部分奇偶校验写入 PPC，然后将选择的条带中的数据写入存储设备。当数据从写缓存中删除，写缓存释放一些空闲空间。

如果 PPC 已满，则 RAID 控制器通过使用最近最少使用（LRU）算法来选择 victim 部分奇偶校验。然后，控制器重建所选部分奇偶校验的完整奇偶校验。然后，控制器重建所选部分奇偶校验的完整奇偶校验。此操作可能需要从 SSD 中读取不属于所选部分奇偶校验的数据，然后 PPC 可释放一些空闲空间。

图 5 显示了所提出的有效缓冲区管理方法的流程图。PPC [10]和提出的方法之间的主要区别在于选择 victim 条带。所提出的方法中的 victim 条带选择规则是，如果写入缓存中的条带包含最大页数，并且其部分奇偶校验可以与现有的部分奇偶性合并，则将首先选择该条带。否则，论文中遵循 PPC 的选择规则。所提出的方法可以避免部分 PPC 经常被占用。当 PPC 已满时，RAID 控制器需要计算完整的奇偶校验，并将奇偶校验数据写回存储设备，因此开销非常大。

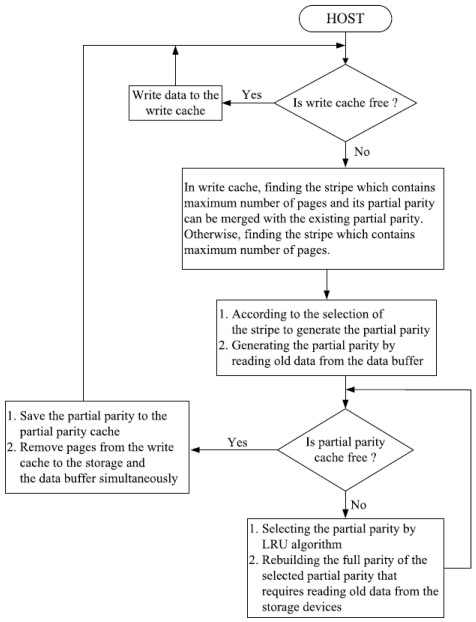


图 5 拟议方案的流程图

而且，论文中使用数据缓冲区来降低部分奇偶校验更新的成本。在 PPC 中，RAID 控制器需要从存储设备读取旧数据来更新部分奇偶校验。在所提出的方法中，这些旧数据被存储在数据缓冲器中，并且因此可以显着减少来自 SSD 的读取操作次数，并且可以进一步改善 RAID 系统的 I / O 处理速度。

3.3 写入缓存和数据缓冲区的操作

本节讨论 PPC，写入缓存和数据缓冲区的操作。写入缓存存储来自主机系统的数据。当写入缓存满时，RAID 控制器从写入缓存中选择受害者条带。例如，图 6 显示论文中选择条带 S0 作为受害者条带，因为 S0 包含最大量的数据（即三页）。其他条纹 S5 和 S8 仍然保留在写缓存中以包含更多数据。

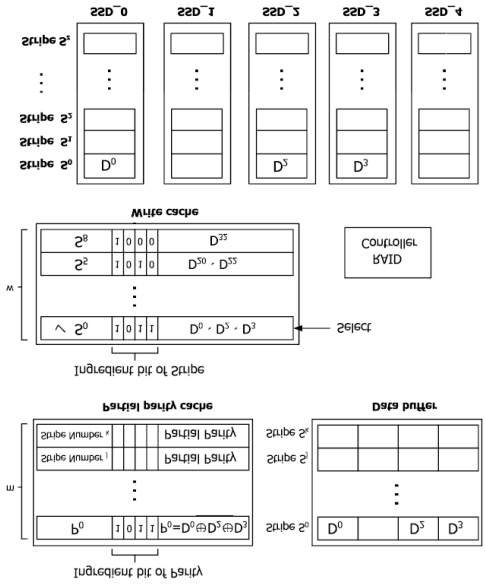


图 6 拟议方案的运作

当选择条带 S0 时，RAID 控制器执行以下操作：

1）通过使用 D0，D2 和 D3 生成部分奇偶校验 P0，并将该成分比特存储到 PPC；
2）将 D0，D2 和 D3 写入数据缓冲区的相应位置；
3）将 D0，D2 和 D3 写入 SSD 的相应位置。

图 6 显示了系统复位时的情况，因此在 PPC 中没有部分奇偶校验，RAID 控制器直接创建一个部分奇偶校验 P0。部分奇偶校验 P0 保存在 PPC 中，不写回到 SSD。当 P0 仍然在 PPC 中时，写入 D0，D1，D2 和 D3 没有进一步的完全奇偶校验更新。因此，PPC 可以帮助减少 RAID-5 架构中的完整校验写入次数。数据缓冲区总是存储最新的数据，新选中的数据直接覆盖数据缓冲区中的数据。

由于主机系统不断向 RAID 系统写入数据，因此 PPC 和写入缓存都没有足够的可用空间。然后，他们执行频繁的受害者条纹选择和许多部分奇偶校验更新。图 7 显示了部分奇偶校验的合并。在这个例子中，在写入缓冲区中有很多条包含三页数据。此外，论文中假设 PPC 中没有可用空间。

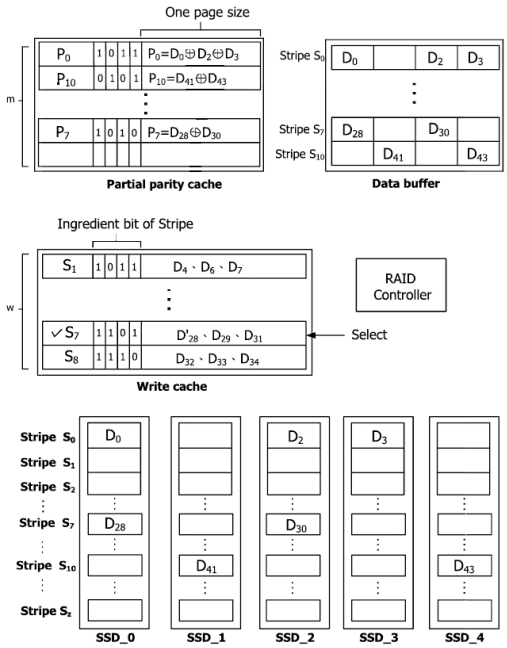


图 7 计算部分奇偶校验的操作

在 PPC 中，RAID 控制器随机选择任何包含最大数据量的条带作为 victim 条带。如果选择 S1 作为受害者条带，由于 PPC 中没有空闲空间，RAID 控制器需要使用 LRU 算法从 PPC 中去除部分奇偶校验，以便为新的部分奇偶校验 P1 释放一个空闲空间。如果 RAID 控制器决定从 PPC 移除部分奇偶校验 P7，则 P7 的全奇偶校验生成成本是对 SSD 的两个读取操作和一个写入操作。由于部分奇偶校验 P7 由 D28 和 D30 生成，所以执行两次读取操作以从 SSD 读取 D29 和 D31。执行一次写入操作以将生成的完整奇偶校验写回到 SSD。

因此，在这个例子中，由于在 PPC 中已经存在相关的部分奇偶校验 P7，所以所提出的有效的缓冲区管理方法选择 S7 作为受害者条带以便降低奇偶校验更新的成本。随后，RAID 控制器执行以下操作：

1）计算新的部分奇偶校验$P7'$如下：$P7'$= P7⊕D28⊕$D28'$⊕D29⊕D31（D28 从数据缓冲区获取）；

2）将$D28'$，D29 和 D31 写入数据缓冲区（$D28'$将被 D28 替换）；

3）将$D28'$，D29 和 D31 写入 SSD（在 SSD_0 中，D28 将被$D28'$覆盖）。

新的部分奇偶校验$P7'$代替旧的部分奇偶校验 P7，并且不占用 PPC 中的新空间。因此，论文中可以减少完整奇偶校验生成操作的次数以及完整奇偶校验写入 SSD 的次数。另外，RAID 控制器计算新的部分奇偶校验时，可以从数据缓冲区中获取 D28。因此，所提出的方法不仅减少了 SSD 的奇偶校验写入时间，而且减少了 SSD 的读取时间。

图 8 显示了在所有可能情况下重建完整校验的开销。有两种方法可以重建如下完整的奇偶校验。

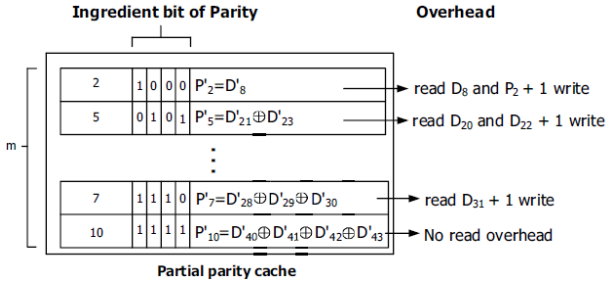


图 8 重建完整的平价的开销

方法 1：从 SSD 读取相应的旧数据和旧的完全校验以重建新的完整校验。

方法 2：从 SSD 中读取不属于部分奇偶校验的数据以重建新的完整奇偶校验。

当部分奇偶校验$P2'$需要被写回到 SDD 时，RAID 控制器采用方法 1 来重建完整的奇偶校验，而不是方法 2。这种偏好的原因是方法 2 具有三个读取成本（读数 D9，D10 和 D11），这比使用方法 1 的成本要高。新的完整奇偶校验 P2 可以计算为$P2' \oplus D8 \oplus P2$。

RAID 控制器将决定哪种方法更高效。当部分奇偶校验$P5'$和$P7'$必须写回 SSD 时，RAID 控制器采用方法 2 来重建完整奇偶校验。新的完整奇偶校验 P5 可以计算为$P5' \oplus D20 \oplus D22$，新的完整奇偶校验 P7 可以计算为$P7' \oplus D31$。当 RAID 控制器需要重建新的完整奇偶校验 P10 时，由于部分奇偶校验也是完整奇偶校验，因此不会有读取开销。

4 PPC 实验

具有所提出的高效缓冲区管理方案的 RAID-5 控制器已由 Socle Technology Corporation MDK-3D 开发板验证。CPU 为 ARM1176JZF，频率高达 1GHz。先进的微控制器总线架构先进的高性能总线频率高达 200 MHz，并支持 NOR 闪存/ NAND 闪存/ DDR2 存储器。

4.1 实验准备

论文中采用现场可编程门阵列实现了 RAID-5 控制器和提出的高效缓冲区管理方案。另外，论文中建立了仿真环境来评估所提出的 RAID-5 控制器的性能，如图 9 所示。整个系统由一个 RAID-5 控制器和 SSD 组成[5]。

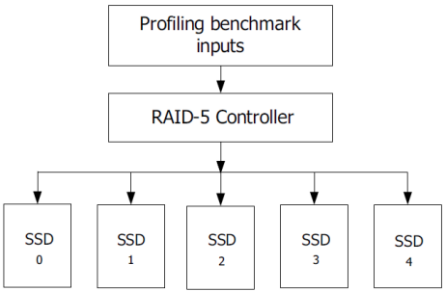


图 9 4 + 1 RAID-5 模拟环境

RAID-5 控制器接受来自基准分析结果的输入。条带编号和 SSD 编号是通过将逻辑地址除以数据存储设备的总数来生成的，商和剩余部分分别被用作条带编号和 SSD 编号。例如，如果 RAID 系统是 4 + 1 RAID-5 架构并且逻辑地址是 2045，则 2045 除以 4，商和余数分别是 511 和 1。因此，具有逻辑地址 2045 的数据在条带 S511 被写入 SSD_1。

RAID 控制器还管理写入缓存，PPC 和数据缓冲区。论文中介绍两个基准，Iozone 和 Postmark 作为输入。在分析结果的基础上，论文中分别使用 Iozone 和 Postmark 进行顺序写作测试和随机写作测试。论文中选择这两个基准的原因是由于大多数应用程序包含混合顺序写入和随机写入。结果，这两个基准可以测试所提出的有效缓冲区管理方案的最佳和最差性能。

在模拟环境中，SSD 的页面大小为 2kB，写入缓存的大小为 16kB。数据缓冲区的大小和 PPC 的大小是每个 16 kB。此外，论文中重建四种类型的 RAID 架构进行比较如下。

1）RAID-5（只有写入缓存）。

2）FPC（具有全奇偶校验缓存的 RAID-5）。
3）PPC（带 PPC 的 RAID-5 ）。
4）建议（提出了 PPC 和数据缓冲区的 RAID-5）。

RAID-5 是具有写入缓存的传统 RAID-5 架构。FPC 使用完整的奇偶校验缓存来减少奇偶校验写入次数。在图 10，图 11，图 13，图 14 和图 16-18 中，论文中显示了通过 4000 个写入请求的总输入来标准化 SSD 的读取和写入次数。

4.2 奇偶校验的开销

图 10 显示了用 Postmark 基准输入生成奇偶校验的平均写入请求开销。传统的 RAID-5 方案不具有奇偶校验高速缓存，因此用于产生奇偶校验的读取操作和写入操作的次数最高。当常规 RAID-5 方案的写入缓存已满时，RAID 控制器选择受害者（victim）条带并将数据写回到 SSD。根据条带的选择，RAID 控制器建立关联完全奇偶校验，并直接将完整奇偶校验写回到 SSD。当 RAID 控制器产生完整的奇偶校验时，需要读取受害者条带的剩余数据，这些读取操作是全部奇偶校验生成的开销;因此在常规 RAID-5 中读取操作的数量也是最高的方案。因此，直接将 SSD 添加到传统的 RAID-5 架构是不合适的。

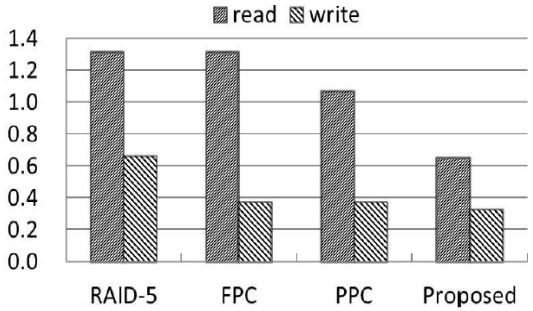


图 10 生成奇偶校验（Postmark）的平均写请求开销

FPC 方案具有完整的奇偶校验缓存。FPC 和 RAID-5 的主要区别在于，在奇偶校验缓存满之前，奇偶校验缓存中将保留完整的奇偶校验。因此，可以将一些完整的奇偶校验合并到这个高速缓存中，并且可以减少奇偶校验的写入次数。然而，奇偶校验生成的读取时间几乎与常规 RAID-5 方案中的读取时间相同。其原因是 FPC 中的全部奇偶校验生成与传统的 RAID-5 方案相同。

PPC 方案采用部分奇偶校验方案，这可以减少奇偶校验生成的读取次数。奇偶校验的写入时间几乎与 FPC 方案中的写入次数相同。论文中可以从前面提到的实验结果得出两个结论。

1）奇偶校验缓存可以减少奇偶校验数据的写入次数。
2）部分奇偶校验方案可以减少产生完整奇偶校验的读取次数。

所提出的使用具有数据缓冲器的有效缓冲区管理方法的方案可以减少对 SSD 的读取时间和写入时间。RAID-5，FPC，PPC 的奇偶校验生成的标准化写入时间分别为 0.66,0.37,0.37 和 0.32。与 PPC 和 RAID-5 相比，所提出的方案将奇偶校验的写入次数分别减少了 13％和 51％。RAID-5，FPC，PPC 的奇偶校验生成的标准化读取时间和所提出的方案分别是 1.31,1.31,1.068 和 0.65。与 PPC 和 RAID-5 相比，所提出的方案将奇偶校验生成的读取次数分别减少了 39％和 50％。

图 11 显示了 Iozone 基准的模拟结果。奇偶校验生成 RAID-5，FPC，PPC 和提出的方案的规范化写入时间分别为 0.3,0.19，0.19 和 0.18。与 PPC 和 RAID-5 相比，所提出的方案将奇偶校验的写入次数分别减少了 5％和 40％。由于使用 PPC 和有效的缓冲方法，写入时间减少。RAID-5，FPC，PPC 的奇偶校验生成的标准化读取次数分别为 0.28,0.28,0.5 和 0.32。与 PPC 相比，所提出的方案能够有效地将产生奇偶校验的读取次数减少 36％。

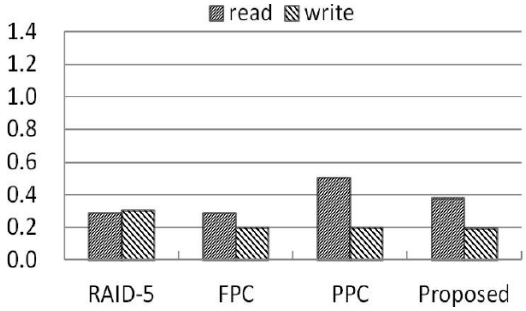


图 11 生成奇偶校验的平均写入请求开销（Iozone）

然而，在 PPC 和所提出的方案中，奇偶校验生成的读取次数大于 FPC 和 RAID-5。原因是由于 Iozone 基准测试的输入是连续的，有许多文件重写请求，因此有很多部分奇偶校验更新。

PPC 方案中的读取时间包括对 SSD 进行部分奇偶校验更新的读取操作以及对完整奇偶校验回写的 SSD 写入操作。相反，在所提出的方案中，可以从数据缓冲器获得用于部分奇偶校验更新的旧数据，并且因此可以显着减少用于部分奇偶校验更新的 SSD 的读取操作次数。如图 11 所示，PPC 中的标准化读取时间包括 0.32（对于部分奇偶校验更新）和 0.18（对于全奇偶校验生成）。因此，所提出的数据缓冲器可以减少用于部分奇偶校验更新的读取次数，而具有少量附加硬件成本。

4.3 RAID/IO 的整体性能

图 12 显示了不同 RAID 体系结构的 I / O 性能比较。性能标准化为传统的

RAID-5 方案。RAID-5，FPC，PPC 和 Postmark 基准输入的 RAID I / O 性能分别为 1.0,1.13,1.47 和 1.76。与 RAID-5 和 PPC 相比，该方案的 I / O 性能分别提高了 76％和 19％。

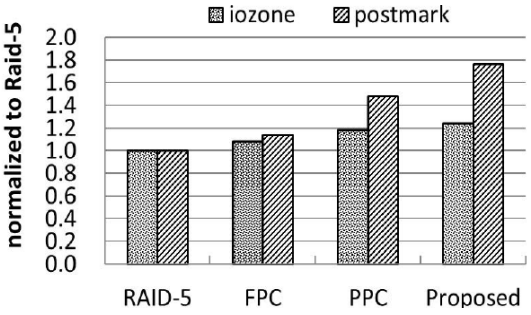


图 12 RAID I / O 性能

在对 RAID 系统的许多写请求之后的总执行周期的详细分析如下。

W_{cycle} 在写入缓存中花费的周期数；

D_{cycle} 将数据写入 SSD 和数据缓冲区的周期数；

R_{cycle} 读取操作中用于产生完全奇偶校验或更新部分奇偶校验的周期数；

WP_{cycle} 将完整奇偶校验写回 SSD 的周期数；

EBF 在建议的有效缓冲区管理方法中花费的周期数；

LRU在最近最少使用的方法中花费的周期数；

T_{cycle} 总执行周期数。

在所提出的方法中，执行周期的总数可以表示如下：

$$T_{cycle} = cycle + D_{cycle} + R_{cycle} + WP_{cycle} + \text{EBF} + \text{LRU} \quad (1)$$

在 PPC [10]中，执行周期的总数可以表示如下：

$$T_{cycle} = W_{cycle} + D_{cycle} + R_{cycle} + WP_{cycle} + \text{LRU} \quad (2)$$

最后，RAID-5 和 FPC 的执行周期总数可以表示如下：

$$T_{cycle} = W_{cycle} + D_{cycle} + R_{cycle} + WP_{cycle} \quad (3)$$

随机写入请求比 RAID 系统中的顺序写入请求更经常发生。所提出的方案很好地处理随机写入请求，因为奇偶校验合并通常在具有随机输入的 PPC 中发生。该方案的R_{cycle}和WP_{cycle}小于 PPC 。

Iozone 基准输入的 4 + 1 RAID-5 I / O 性能对于不同的 RAID 体系结构几乎是相同的，因为 RAID 控制器可以轻松地找到带有顺序写入请求的四页数据的条带。当 RAID 控制器选择一个带有四页数据的条带时，与该条带相关的部分奇偶校验也被认为是完全奇偶校验，因此 PPC 方案和所提出的方案都不需要重建完整奇偶

校验。在这种情况下，每种 RAID 架构中的R_{cycle}和WP_{cycle}几乎相同。

4.4 差分参数模拟

图 13 和 14 分别显示了用于产生具有不同条带大小的奇偶校验的平均读取次数和写入次数。由于所提出的高效的缓冲区管理方法，所提出的用于产生奇偶校验的方案的写入次数小于 PPC 的写入次数，如图 5 所示。当条带大小增加时，在 PPC 和提出的方案中用于产生奇偶性的写入次数都减少。这是因为部分奇偶校验可以将更多的数据与相对较大的条带大小合并。例如，4 + 1 RAID-5 系统中的部分奇偶校验 P0 可以表示 D0 到 D3。但是，7 + 1 RAID-5 系统中的部分奇偶性 P0 可以表示 D0 至 D7。这意味着合并部分平价的可能性会增加。

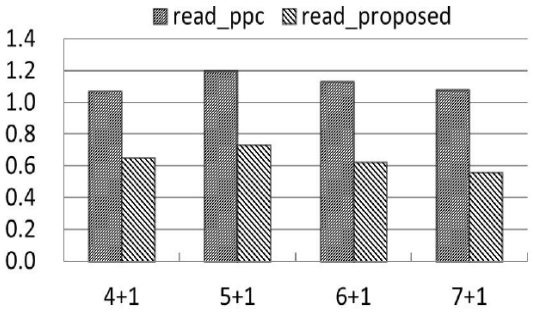


图 13 用于生成具有不同条带大小的奇偶校验（Postmark）的平均读取时间

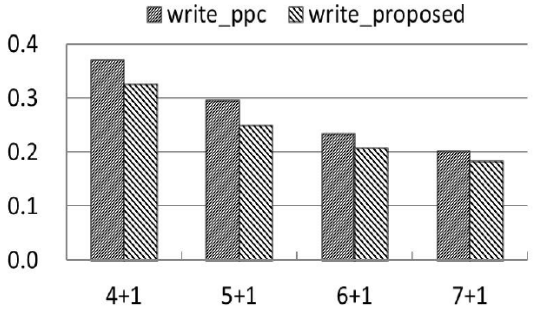


图 14 用于生成具有不同条带大小的奇偶校验（Postmark）的平均写入次数

在所提出的方案中，用于产生奇偶性的读取次数也小于具有不同条带大小的 PPC 。在提出的方案中，产生奇偶校验的读取次数相对较少的事实有两个原因：首先，所提出的方案的产生奇偶校验的写入次数小于 PPC，从而也减少了重建全部奇偶校验的相关读取次数。其次，所提出的方案增加了数据缓冲器来存储用于部分奇偶校验更新的旧数据，从而可以进一步减少产生奇偶校验的读取次数。

图 15 显示了具有不同条带大小的所提出的方案的 RAID I / O 性能。性能标准化为 4 + 1 RAID-5 系统。4 + 1，5 + 1，6 + 1 和 7 + 1 RAID 系统的 I / O 性能分别为 1.0,1.15,1.29 和 1.47。如图 13 和 14 所示，用于产生奇偶校验的读取次数和写入

次数随着条带大小的增加而逐渐减小。另外，RAID 控制器在条带大小相对较大的情况下可以很容易地将数据写入到 SSD 中。结果，当条带大小增加时，I / O 性能也提高。

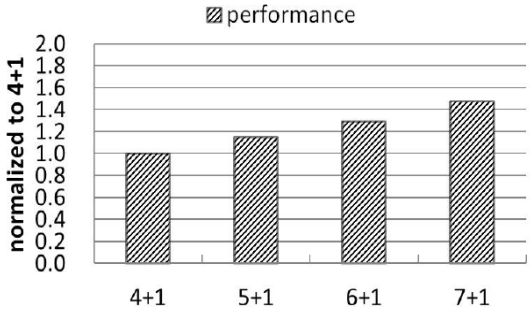


图 15 具有不同条带大小的 RAID I / O 性能（Postmark）

图 16（a）和（b）显示了 PPC 方案和提议方案的读取时间与邮戳基准输入的比率。在 PPC 中的完全校验回写操作的情况下，比率是 0.73,0.79,0.64 和 0.51。在 PPC SSD 部分奇偶校验更新的情况下，比率分别为 0.33,0.40,0.48 和 0.56。在 PPC 的情况下，部分奇偶校验更新的读取时间比率随着条带大小的增加而逐渐增加，这是因为在数据条带相对较大的情况下，部分奇偶校验与更多的数据相关联。所提出的方案增加了数据缓冲器以保留用于部分奇偶校验更新的旧数据，因此不存在用于部分奇偶校验更新的 SSD 读取操作，如图 16（b）所示。

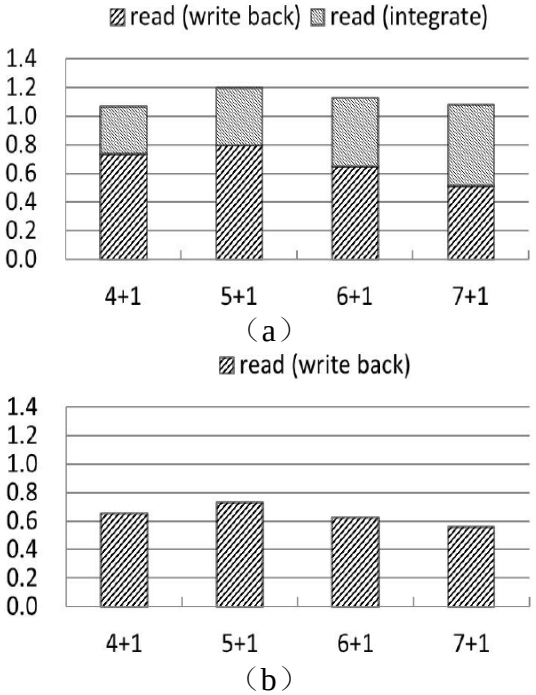


图 16 生成奇偶校验（Postmark）的读取时间比率。（a）PPC（b）推荐的计划

图 17（a）和（b）显示了 PPC 方案和提出的方案的读取时间与 Iozone 基准输入的比率。在 PPC 的完全校验回写操作中，比率分别为 0.18,0.64,0.53 和 0.34。在

PPC SSD 中进行部分奇偶校验更新的情况下，比率分别为 0.32,0.39,0.47 和 0.54。连续输入的全部奇偶校验生成的读取次数比随机写入输入的读取次数少。在图 17（b）中，论文中看到，通过添加数据缓冲器，所提出的方案中的总读取时间减少。

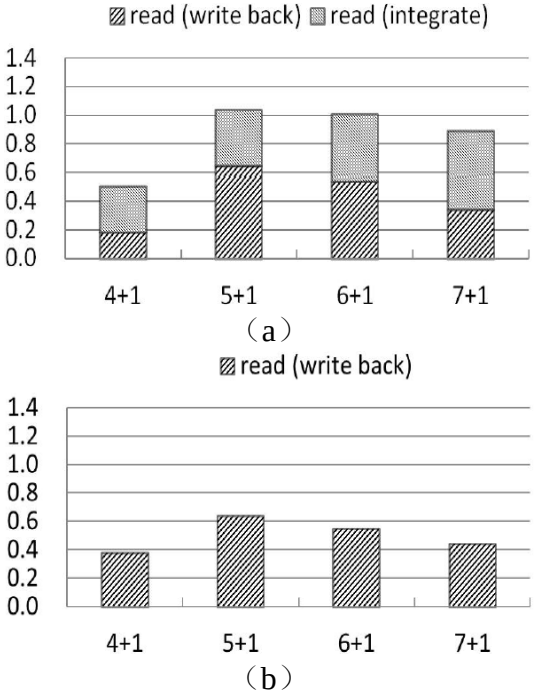
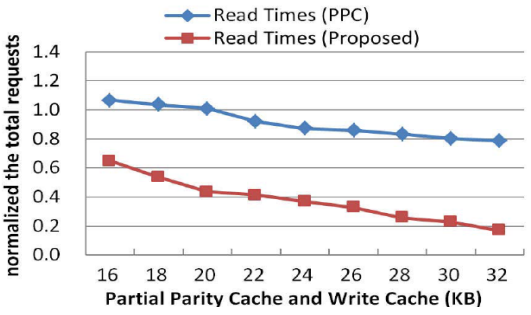
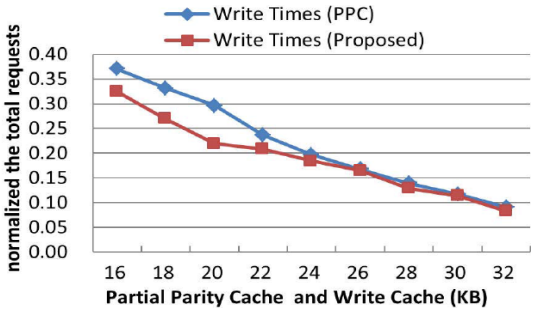


图 17 产生奇偶校验的读取时间比率（Iozone）。（a）PPC（b）推荐的计划

当 RAID 控制器从写缓冲区中选择带有 Iozone 和 Postmark 基准的受害者条带的平均页面大小分别是 3.0 和 1.3 页。就 Iozone 基准测试而言，RAID 控制器平均需要 1.0 次读取操作才能生成完整校验码。在 Postmark 基准测试中，RAID 控制器可能平均需要 2.7 次读取操作才能生成完整校验码。因此，如图 16 和 17 所示，在邮戳输入的情况下的读取操作的数量大于在 Iozone 输入的情况下的读取操作的数量。





（b）

图 18 用不同高速缓存大小生成奇偶校验的平均读取次数和写入次数。

图 18（a）和（b）显示了用于产生具有不同高速缓存大小的 PPC 方案和所提出的方案的奇偶校验的读取次数和写入次数。图 18（a）显示了用于产生所提出的方案的奇偶校验的读取次数总是小于 PPC 的情况。在所提出的方案中，读取时间分别是 0.65 和 0.16，PPC 和写入缓存大小分别为 16kB 和 32kB。PPC 方案的读取时间分别是 1.06 和 0.80，PPC 和写入缓存大小分别为 16kB 和 32kB。较大的 PPC 可以更有效地合并部分奇偶校验。而且，所提出的方案中的数据缓冲器有助于减少产生奇偶校验所需的读取时间。

图 18（b）显示了对 PPC 方案 SSD 和具有不同高速缓存大小的所提出方案的全奇偶校验回写操作的写入次数。所提出的方案即使使用相对较小的缓存也可以减少写入时间。因此，论文中可以在有限的硬件资源的情况下很好地处理完全校验回写操作。在较大的缓存大小的情况下，PPC 方案和提出的方案都可以减少写入时间。然而，在所提出的有效的缓冲区管理方案中，所提出的方案的写入时间将总是小于 PPC 方案的写入时间。

5 总结

原论文中提出了一种 PPC 和数据缓存管理方法来提高基于 SSD 的 RAID 系统的性能。由于固态硬盘的特性与传统硬盘不同，因此在 RAID-5 架构中增加固态硬盘有很多考虑因素。

提出的具有有效的缓冲区管理方法的 PPC 可以更有效地合并部分奇偶校验数据。建议的 RAID 控制器从写入缓存中选择合适的受害者条带，以防止 PPC 经常被占满。当 PPC 已满时，RAID 控制器使用 LRU 算法选择牺牲部分奇偶校验，重新构建所选奇偶校验的完整奇偶校验，并将其写回到 SSD。论文中还添加了一个数据缓冲区来减少部分奇偶校验更新开销。实验结果表明，所提出的方案减少了用于产生奇偶校验的 SSD 的写操作次数和写操作次数。

参考文献

[1] Dirik, Cagdas, and B. Jacob. "The performance of PC solid-state disks (SSDs) as a function of bandwidth, concurrency, device architecture, and system organization." ACM, 2009:279-289.

[2] Lee, Chul, S. H. Baek, and K. H. Park. "A Hybrid Flash File System Based on NOR and NAND Flash Memories for Embedded Devices." IEEE Transactions on Computers 57.7(2008):1002-1008.

[3] Chang, Li Pin. "A Hybrid Approach to NAND-Flash-Based Solid-State Disks." IEEE Transactions on Computers 59.10(2010):1337-1349.

[4] Chung, Ching Che, and N. M. Hsueh. "A low-complexity high-performance wear-leveling algorithm for flash memory system design." IEICE Electronics Express 9.24(2012):1874-1880.

[5] Im, Soojun, and D. Shin. "Flash-Aware RAID Techniques for Dependable and High-Performance Flash Memory SSD." IEEE Transactions on Computers 60.1(2010):80-92.

[6] Kim, Hyojun, and U. Ramachandran. "FlashLite: A User-Level Library to Enhance Durability of SSD for P2P File Sharing." IEEE International Conference on Distributed Computing Systems IEEE, 2009:534-541.

[7] latexhtml. "Design Tradeoffs for SSD Performance. " Security & Privacy IEEE 7.2(2008):57-70.