

学号 2017282110269

密级 公开

武汉大学研究生课程论文

访问特性引导读写成本调整，
提高闪存性能

院（系）名 称：计算机学院

专 业 名 称 ： 计算机技术

学 生 姓 名 ： 贾明媚

指 导 教 师 ： 何水兵 教授

二〇一七年十二月

访问特性引导读写成本调整，提高闪存性能

摘要

写操作成本相对较高，已成为闪存性能瓶颈。写入成本是指使用增量步骤脉冲编程（ISPP）编程闪存页面所需的时间，而读取成本指的是从存储中检测和传输页面所需的时间。如果在 ISPP 处理期间通过使用更精细的步长来以更高的成本写入闪存页面，则由于在感测和传输中节省了时间，所以可以以相对较低的成本读取闪存页面，反之亦然。

我们引入了 AGCR，这是一种利用这种权衡来提高闪存性能的访问特性引导成本调节方案。基于工作负载特性，接收更多读取的逻辑页面将使用更精细的步长来写入，从而降低其读取成本。类似地，接收更多写入的逻辑页面将使用更粗略的步长写入，从而降低其写入成本。我们的评估显示，与以前的方法相比，AGCR 的开销微乎其微，同时平均提升了 15% 的性能。

1 引言

NAND 闪存由于其轻量级，高性能和小尺寸等优势而在嵌入式系统，个人电脑，移动设备和服务器领域的应用越来越广泛。尽管基于闪存的存储性能优于大多数基于磁性的存储，但是写入和读取操作的成本仍然是当前系统的性能瓶颈。闪存的写入操作由增量步进脉冲编程（ISPP）方案执行。

ISPP 旨在迭代地增加编程电压，并使用小的验证电压来可靠地将闪存单元编程到指定的电压。在每次迭代中，程序电压增加一个预定义的程序步长。一般来说，写入成本比读取成本高很多，通常是 10 到 20 倍，并被认定为主要的性能瓶颈。与此同时，随着比特密度和技术规模的扩大，阅读成本也大大增加。

读写成本的下列折衷已被确定。首先，读取成本高度取决于数据页的最大原始误码率（RBER）和部署的纠错码（ECC）。对于特定的 ECC，最大 RBER 越低，读取成本就越低。其次，可以通过增加 ISPP 进程中的程序步长来降低写入成本，代价是增加编程页面的最大 RBER。这又降低了最大保留时间并增加了编程页面的读取成本。

先前的方法在用于选择将以不同的写入成本写入的数据页面的策略方面不同。Pan 等人和 Liu 等人提出了通过保留时间放松对数据页面应用低成本的写入（即使用更粗的步长）。与预定义的保留时间相比，他们的方法受到多个工作负载中大多数数据的寿命显着短的驱动。另一方面，高成本写入降低了以下读取到同一页面的成本。基于这个特点，Li 等人提出了在有几个排队请求时应用低成本的写入，否则就是高成本的写入。建议应用高成本的写入时，根据他们的估计，下一个访问操作将不会由于成本增加而延迟。但是，这些作品都没有利用工作负载的访问特性。

这项工作是第一个利用工作负载的访问特性进行成本监管的。我们定义了三个页面访问特性：如果几乎所有对数据页面的访问都是读取（写入）请求，则该页面被定义为只读（只写）。如果对数据页面的访问与读取和写入交错，则该页面被表征为交织访问。我们的方法是基于这样一种观察：来自主机的大多数访

问都是在只读或只写页面上执行的。我们利用这一观察来调节访问成本：对于被认定为只读的访问，低成本读取是优选的；对于标识为只写的访问，低成本写入是优选的。所提出的方法选择低密度奇偶码（LPDC）作为 ECC，这是高级闪存存储系统的最佳选择[6] [14]。对于以低成本编写的页面，只要得到的错误率不超过部署的 LPDC 代码的错误纠正能力，所需的保留时间仍然可以保持。因此，可以调节读写成本以显著改善整体性能。我们的主要贡献如下：

- 我们建议 AGCR，一个全面的方法来规范写和读的成本；
- 我们进行初步研究以显示 AGCR 潜在的绩效改善；
- 我们提出 AGCR 的一个有效的实施与可忽略的开销。在 12 个工作负载的跟踪驱动模拟器上的实验结果表明，AGCR 实现了显著的性能改进。

本文的其余部分安排如下。第 2 节介绍动机和问题陈述。第 3 节介绍我们提出的方法。第 4 节介绍了实验和分析。第 5 节总结了这项工作。

2 动机与初步研究

2.1 在闪存中读写

写操作是闪存的性能瓶颈，不仅因为它们需要比读操作更多的时间，而且还因为它们阻塞了等待操作。而且，写入操作的成本隐含地决定了后续读取的成本。闪存技术使用增量步进脉冲编程（ISPP）来编程页面[3]。ISPP 旨在使用迭代编程验证算法可靠地将闪存单元编程到其特定的电压电平。迭代算法在每个迭代步骤中有两个阶段：首先用增量编程电压编程闪存单元，然后验证单元的电压。如果电压低于预定义的阈值，则该过程继续。在每次迭代中，程序电压增加 ΔV_{pp} 的步长。步长 V_{pp} 确定写入成本 - 较粗的步长表示较小的步数（因此较低的写入成本），但是在编程页面中导致较高的原始误码率（RBER），反之亦然。

低密度奇偶校验码（LDPC）被部署为最先进的闪存技术的默认 ECC。对于高 RBER，LDPC 需要细粒度的存储单元感测，这是通过比较一系列 N 个参考电压而实现的。LDPC 的纠错能力随着 N 的增加而增加，但是 N 越大，读取成本就越高。

因此，ISPP 和 LDPC 之间在闪存的读写成本上存在着很强的关系。在 ISPP 处理中步长较大时，写入成本降低，但是 RBER 和读取成本增加，反之亦然。步长还会影响 Flash 页面的保留时间。对于使用更精细的步长编写的页面，预计会有更长的保留时间。在这项工作中，我们专注于读写成本的交互，通过调整步长来调整最小的保留时间。

2.2 读写成本调节器

最近提出了几种策略来利用闪存的可靠性特性来调节读写成本。潘等人。Liu 等人提出通过放宽编程页面的保留时间要求来降低写入成本。Wu 等人提出应用高成本的写入来降低在同一页面上执行的读取请求的成本。高成本的编写是在即将到来的请求不会因成本增加而延迟的前提下进行的。Li 等人提出，当存在排队请求以降低排队延迟时应用低成本写入，并且否则应用高成本写入，从而允许以低成本读取页面。但是，这些作品都没有利用成本监管的准入特征。以前的工作中已经研究了几种类型的工作负载访问特性，包括冷热访问，相互参照间隙，

时间和空间局部性。这些访问特性已被用来指导缓冲区高速缓存和闪存翻译层（FTL）的设计。但是，这些特征意味着访问频率，不能被用于读写成本调节。

在这项工作中，我们应用了 Li 等人提出的成本调节器，在以前的研究中已经通过仿真验证。监管机构由写成本调节器和读取成本调节器组成。写入成本与 ISPP 的程序步长成反比。因此，写成本调节器被定义为：

$$WC(\Delta V_{pp}) = \gamma \times \frac{1}{\Delta V_{pp}}$$

where $RB ER(\Delta V_{pp}) < CBER_{LDPC}(N)$; (1)

$WC(\Delta V_{pp})$ 表示程序步长为 ΔV_{pp} 时的写入成本，是一个变量。较粗的步长会导致写入成本较低，但 $RB ER$ 较高。另外，由于所产生的 $RB ER$ 应该在具有 N 个参考电压的所部署的 LDPC 码的纠错能力 $CBER_{LDPC}(N)$ 内，因此写入成本的降低受到限制。

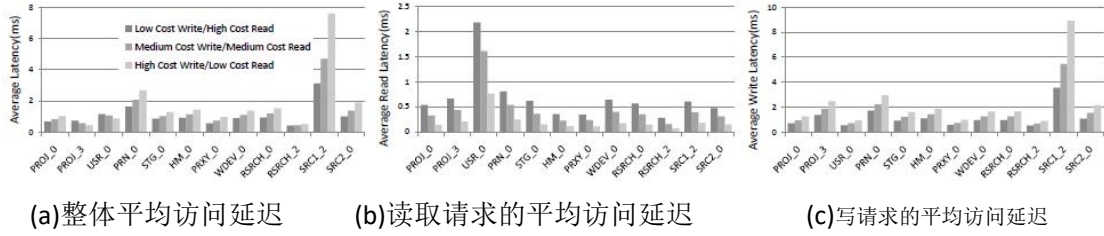


图 1：具有不同访问成本的读取和写入请求的 IO 延迟比较。

读取成本包括检测与 N 成比例的页面的时间，以及从页面向控制器传输数据的时间，其与传输信息的大小成比例。因此，读取成本调节器被定义为：

$$RC(N) = \alpha \times N + \beta \times \lceil \log(N + 1) \rceil$$

where $RB ER(\Delta V_{pp}) < CBER_{LDPC}(N)$; (2)

其中 $RC(N)$ 表示在所部署的 LDPC 码中具有 N 个参考电压的读取成本， α 和 β 是两个变量。随着 N 越大，需要更长的片上存储器感测时间和更长的闪存 - 控制器数据传输时间，这导致读取成本增加。

基于调节器的两个部分，低成本的写入使页面具有更高的 $RB ER$ ，从而需要具有更多参考电压的 LDPC，这进一步导致更高成本的读取。另一方面，高成本写入页面的 $RB ER$ 较低，因此需要具有较少参考电压的 LDPC，这进一步导致较低成本的读取。请注意，监管机构确保在使用不同成本编写数据时始终满足保留时间要求。

2.3 初步研究

在初步研究中，我们评估了三种读写成本组合：1) 所有的写入都是以高成本进行的，其次是低成本读取（HCW / LCR）；2) 所有的写入都是以中等成本执行的，其次是中等成本读取（MCW / MCR）；3) 所有的写入都是以低成本执行的，然后是低成本读取（LCW / HCR）。调节器的详细设置可以在 4.1 节中找到。

图 1 显示了来自 Microsoft Research (MSR) Cambridge 的 12 个代表性企业服务器痕迹的访问延迟比较。与默认的 MCW / MCR 相比，HCW / LCR 将读取性能提高了 54%，LCW / HCR 提高了 26% 的写入性能。显著的性能改进来自相应的访问成本的降低。比较 LCW / HCR

和 HCW / LCR，其读写潜伏期的差异分别为 114% 和 61%。绩效差距表明应该认真应用读写成本监管机构。虽然 LCW / HCR 能够提高整体性能，但它引入了最差的读取性能，如图 1 (b) 所示。然而，阅读操作总是处于关键路径，这激发了我们的方法。

3 AGCR：准入特征引导成本法规

在本节中，我们提出 AGCR，一个访问特性的引导成本调节计划。我们首先介绍几种工作负载的访问特性。然后，基于本研究的观察，我们提出了一个全面的方法，包括一个高精度的访问特征识别方法和一个引导成本调节方案。最后，我们介绍实施和开销分析。

3.1 访问工作负载的特征

为了指导读写成本管理，工作负载的访问特性非常重要。例如，如果访问数据页面（主机系统处的逻辑页面）由读取请求支配，则以低成本读取访问数据页面可显著提高性能。

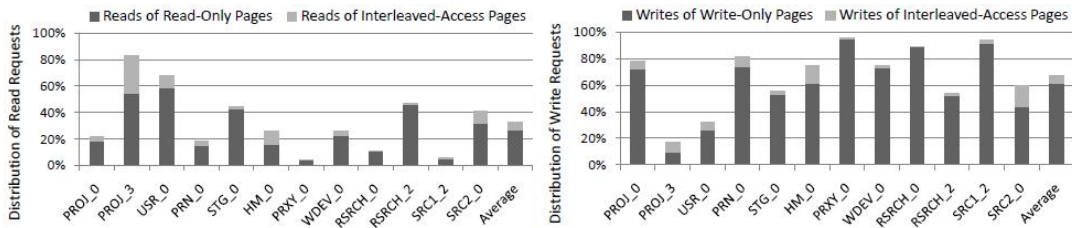
图 2 显示了在我们的初步研究中分析的工作量的统计结果。我们收集了主机系统中所有数据页面的访问特性，并区分了三种类型的数据访问：

- 1.只读：如果几乎所有对数据页面的访问（> 95%）都是读取请求，我们将此页面定义为只读。访问媒体文件或其他只读文件时，这是典型的；
- 2.只写：如果几乎所有对数据页的访问（> 95%）都是写请求，我们将此页描述为只写。这是典型的定期数据从内存系统刷新到存储一致性维护；
- 3.交错访问：如果访问数据页面的操作与读写交错，我们将此页面描述为交错访问。

图 2 显示了这些逻辑数据页访问特性的请求分布。图 2 (a) 显示了只读页面和交错访问页面的读请求分布。类似地，图 2 (b) 示出了只写页面和交错存取页面的写请求的分布。对于每个轨迹，图 2 (a) 和 2 (b) 的访问总和为 100%。我们从图 2 中做出三个观察：

- 观察 1 - 大多数读取请求访问只读页面，平均超过 85%；
观察 2 - 大多数写入请求访问只写页面，平均超过 91%；
观察 3 - 只有一小部分请求访问交错访问页面。

这些观察指导我们的读写成本监管的设计。我们首先提出一个简单而准确的访问特征识别方法。



(a) 分发阅读请求

(b) 分发书面要求

图 2：三种访问特性的读写请求分布

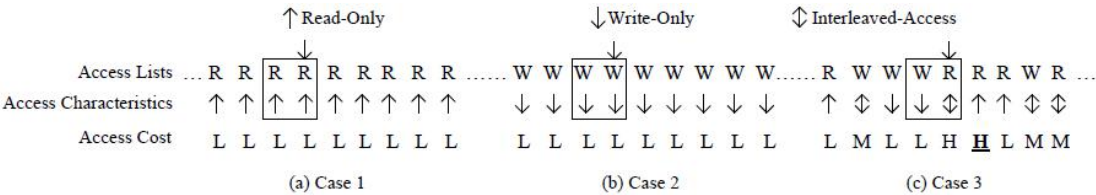


图 3：访问特性识别和成本监管示例

3.2 接入特性标识

我们根据使用历史记录窗口记录的最近的请求和即将到来的请求来识别数据页面的访问特性。如果最近的请求和即将到来的请求都是读取(写入),则页面被表征为只读(只写)。否则,该页面被描述为交织访问。本质上,识别方法寻找连续的读取或写入来表征页面。

图 3 显示了识别方法的例子。第一行显示访问列表。情况 1 仅具有读取请求,情况 2 仅具有写入请求,情况 3 具有交错读取和写入请求。历史窗口的长度在示例中设置为 2。因此,它包括即将到来的请求(由窗口上方的箭头指示)以及对该页面的最近请求。访问列表下方的箭头表示已识别的页面访问类型。可以看出,标识在每次访问页面到达时被更新。如果页面的访问特性确实是只读的或只写的,则识别是准确和稳定的,如图 3(a)和 3(b)所示。但是,如果页面的访问特性是交错的,如图 3(c)所示,识别是不稳定的,并且可能会不时变化。

3.2 接入特性标识

我们根据使用历史记录窗口记录的最近的请求和即将到来的请求来识别数据页面的访问特性。如果最近的请求和即将到来的请求都是读取(写入),则页面被表征为只读(只写)。否则,该页面被描述为交织访问。本质上,识别方法寻找连续的读取或写入来表征页面。

图 3 显示了识别方法的例子。第一行显示访问列表。情况 1 仅具有读取请求,情况 2 仅具有写入请求,情况 3 具有交错读取和写入请求。历史窗口的长度在示例中设置为 2。因此,它包括即将到来的请求(由窗口上方的箭头指示)以及对该页面的最近请求。访问列表下方的箭头表示已识别的页面访问类型。可以看出,标识在每次访问页面到达时被更新。如果页面的访问特性确实是只读的或只写的,则识别是准确和稳定的,如图 3(a)和 3(b)所示。但是,如果页面的访问特性是交错的,如图 3(c)所示,识别是不稳定的,并且可能会不时变化。

这种简单的识别方法效果很好,因为如上所述,大多数访问是 **writeonly** 或只读。识别方法的准确性在第 3.4 节进一步评估,我们分析了历史窗口大小的影响。

3.3 访问成本法规

基于所识别的页面访问特征,可以调节即将到来的请求的成本。主要思想是对只写页面应用低成本写入,对低成本读取(通过应用高成本写入启用),对只读页面和对于交错存取页面的中等成本访问。

写成本监管。写入的成本在发行时受到管制。如图 3 所示,写成本调整有两种情况。在写入请求到达后,该页面被表征并且其成本被调节如下:

- 使用低成本写入来写入只写页面以提高写入性能;
- 如图 3(c)所示,交错存取页面将以中等成本写入,以避免读取性能的影响。

当第一次写入页面时,将执行高成本的写入。

阅读成本法规。如果页面写入的代价高昂,只能发出低成本的读取。因此,高成本的写入将被插入到读取列表中以在即将到来的读取之前重新写入页面,如果页面之前没有以高成本写入的话。如图 3 所示,有三种阅读成本监管的情况。在读取请求到达后,该页面被表征,其成本被调整如下:

- 对于低成本阅读的只读页面,不应该做任何事情;

- 对于交叉访问页面，读取成本不受限制;
- 对于高成本读取的只读页面，将高成本重新写入操作插入到重新写入队列中并在空闲时间执行，以降低即将到来的读取的成本（在图 3（c）中用 H 表示）。

3.4 实施和开销

图 4 显示了闪存控制器中 AGCR 的实现。我们添加三个新的组件：访问特性标识，成本调节器和重写队列。此外，FTL 中的每个映射条目都扩展了两个字段，如图 5 所示。第一个是访问历史记录，第二个是 1 位低成本写入标签。当低成本的写入标签被设置为只读页面时，这表示页面必须被重写。当一个 I/O 请求被发出时，该页面被描述，其成本按照 3.3 节的规定进行调整。如果读取成本调节需要重新写入操作，则将数据页面的逻辑地址添加到重新写入队列中。在空闲期间，重写操作被触发以高成本对队列中的数据进行了编程，这保证了在这些只读页面上的低成本读取。重写开销在实验中被评估。

这种实现招致三种类型的开销：存储，硬件和固件开销。存储开销包括对于每个页面来说，访问历史记录的位数和用于低成本写入标记的一位。假设访问历史记录被设置为一位（只记录最近的页面请求），则对于具有 4KB 页面的 64GB 闪存，存储开销为 4MB。硬件开销包括支持三套不同成本的读写所需的电压阈值。根据以前的研究，这个开销可以忽略不计。固件开销包括访问特性描述，成本调整和重写排队等过程。这些简单的过程的开销可以忽略不计。由于读写成本监管机构的限制，AGCR 不会引入可靠性问题。另外，附加部件的能耗可以忽略不计。

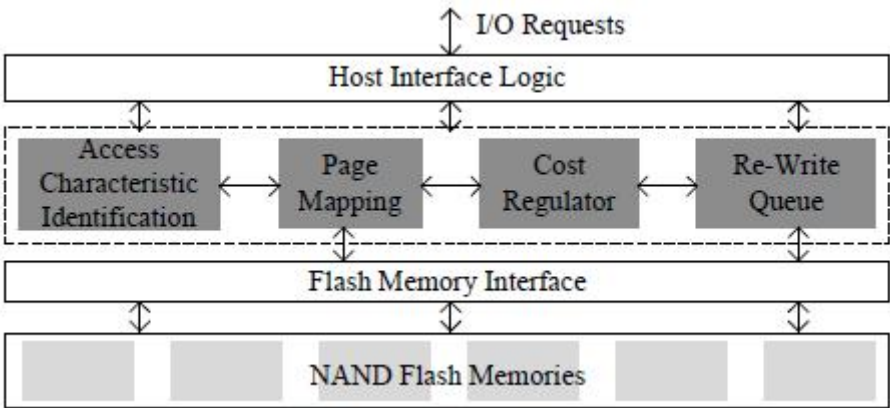


图 4：提出的方法的实施。

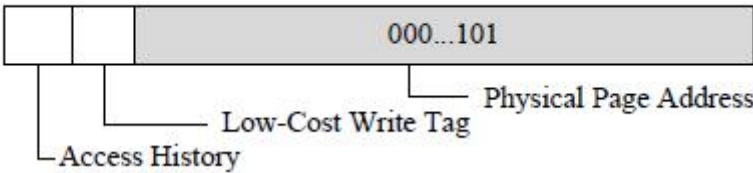


图 5：FTL 映射条目示例

4 评估

4.1 实验设置

表 1: 访问成本配置。

Read Cost (μs)	Low	Medium	High
	70	170	310
Write Cost (μs)	High	Medium	Low
	800	600	450
Erase Cost (μs)	3000	3000	3000

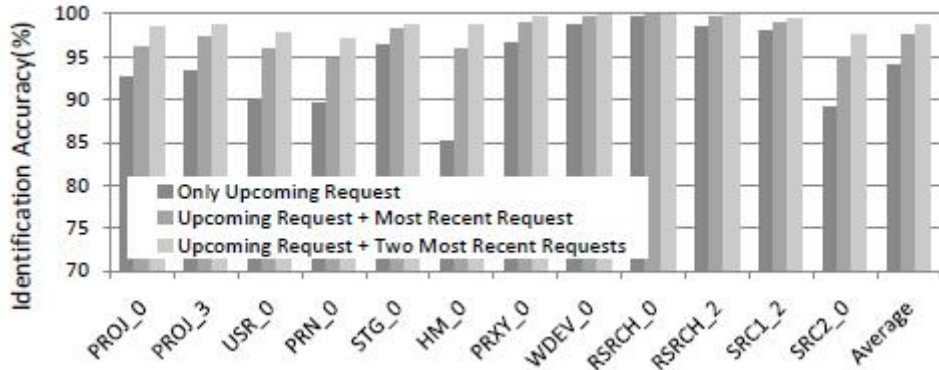


图 6: 窗口大小对识别准确性的影响

我们使用 SSDsim 和 MSR 的工作负载来评估我们的成本监管方法。从我们的研究中，我们发现 MSR 中的所有工作负载具有相似的访问特征。我们从图 2 中选择 12 个代表性曲线进行评估。模拟存储系统配置每个单元 MLC 闪存两位。它有八个通道，每通道八个芯片，每个芯片四个平面。每个芯片有 2048 块，每块有 64 个 4KB 的页面。基于 FTL 的默认页面映射，垃圾收集和耗损均衡在模拟器中实现，代表着最先进的存储系统。这些机制的内部读写操作是以规定的访问成本进行处理的。为了公平比较，我们使用 Li 等人的成本调节器的参数。表 1 显示了访问成本。擦除操作的成本不取决于写入成本。

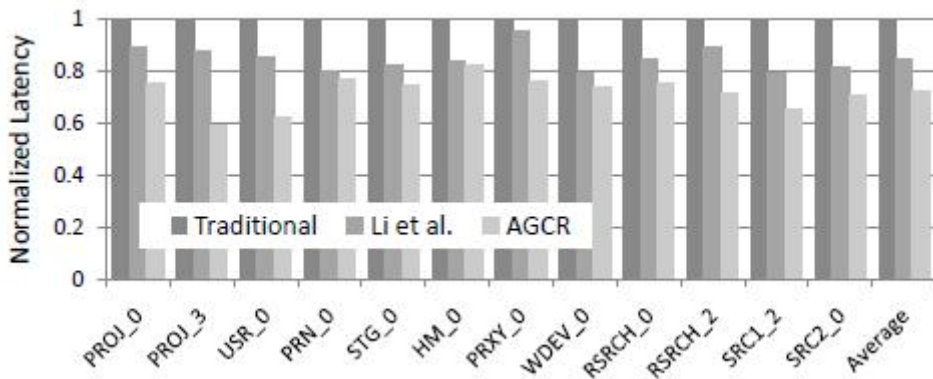


图 7: 整体性能对比

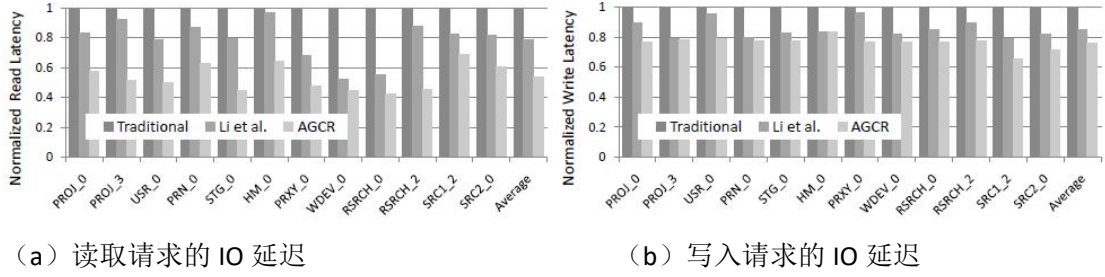


图 8: 与传统情况相比, 归一化访问延迟和 Li 等[13]。

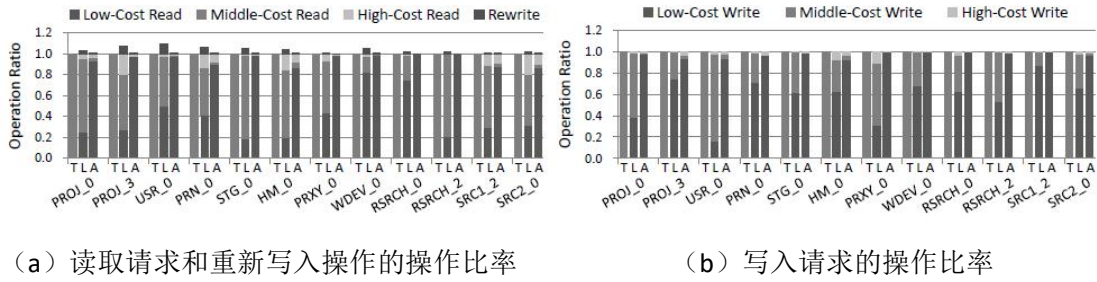


图 9: 不同成本运作的分布: T-Traditional, L-Li 等和 A-AGCR。

其中最重要的参数之一就是识别方法中历史窗口的大小。 它会影响存储开销以及识别的准确性。图 6 显示了三种窗口尺寸的识别精度。结果显示更大的窗口导致更高的准确度。但是,精度的增加随着尺寸的增加而减小。 在下面的实验中,我们只考虑最近的请求和即将到来的识别请求来平衡存储开销和识别精度。

4.2 实验结果

在本节中,对 AGCR 的阅读,写作和整体表现进行评估,并与传统的情况和李等人的最新研究成果进行比较。 在传统的条件下,所有的读写操作都是以中等成本进行的,而 Li 等人 使用低成本读取或写入来减少排队延迟。 成本监管高度依赖于 AGCR 的访问历史。 如果访问页面没有历史记录,则写入请求使用高成本写入,读取请求使用低成本读取。 图 7 和图 8 显示了与传统的非监管成本(图 1 中的 MCW / MCR)和 Li 等人的归一化访问延迟。图 7 显示,与李等人相比,AGCR 平均获得了最好的总体性能提升 15%。

另外,如图 8 (a) 和 8 (b) 所示,与李等人的研究相比,AGCR 分别实现了 48.3%和 20.4%的读写延迟。

为了理解性能变化,图 9 显示了不同成本的操作分布,包括快速,中等和慢速读写,以及重写。 与李等人的工作相比,AGCR 发出了更多的低成本的读写操作,因此取得了很大的进步。 此外,重写操作的百分比可以忽略不计,主机发出的所有访问不超过 1%,这是由于交织访问页请求的一小部分。

5 结论和未来工作在本文中,我们介绍了 AGCR,闪存的成本调节方法,以提高访问性能。 所提出的方法是由广泛研究的工作量得出的观察结果驱动的。 基于这些观察,我们提出了准确的访问特征描述方法,并规定访问成本来提高性能。 我们的仿真结果表明 AGCR 是有效的,与最先进的方法相比,对于读写请求分别减少了 I/O 延迟 48.3%和 20.4%。