## 用于提高闪存性能的访问特性引导读写成本调整

### 摘要

我选的这篇论文是fast16上的一篇来自重庆大学计算机学院和香港城市大学计算机学院的学生和老师共同发表的论文，这篇论文篇幅不是很长，共9页。但是文章的标题很新颖，通过访问特性引导读写成本调整来提高闪存的性能。也就是这篇论文的标题，接下来介绍这篇论文的摘要。

写操作的成本相对较高，这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来写入，以便降低写入成本。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。

所以这篇文章的核心就是讲如何使用ACGR来提高闪存性能，并且这种方案能够在实际的应用中有多大的提高，对存储领域的研究能够起到的启发和帮助。以及在以后的研究中的应用。

### 简介

NAND闪存是一种比硬盘驱动器更好的存储设备，在不超过4GB的低容量应用中表现得犹为明显。随着人们持续追求功耗更低、重量更轻和性能更佳的产品，NAND被证明极具吸引力。NAND闪存是一种非易失性存储技术，即断电后仍能保存数据。它的发展目标就是降低每比特存储成本、提高存储容量。NAND闪存阵列分为一系列128kB的区块(block)，这些区块是NAND器件中最小的可擦除实体。擦除一个区块就是把所有的位(bit)设置为“1”(而所有字节(byte)设置为FFh)。有必要通过编程，将已擦除的位从“1”变为“0”。最小的编程实体是字节(byte)。一些NOR闪存能同时执行读写操作(见下图1)。虽然NAND不能同时执行读写操作，它可以采用称为“映射(shadowing)”的方法，在系统级实现这一点。这种方法在个人电脑上已经沿用多年，即将BIOS从速率较低的ROM加载到速率较高的RAM上。

NAND闪存由于其优势在嵌入式系统，个人计算机，移动设备和服务器上的部署日益增加，如重量轻，性能高，外形小等特点。虽然基于闪存的存储性能优于大多数基于磁性的存储，但是写入和读取的成本却很高，操作仍然是当前系统的性能瓶颈。闪存的写入操作是通过增量步进脉冲编程完成（ISPP）的计划。 ISPP被设计为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要高得多，通常是10倍到20倍，并被认定为主要性能瓶颈与此同时，随着比特密度的增加和技术的扩大，阅读成本也大大增加。

NAND闪存卡的读取速度远大于写入速度。当芯片磨损，抹除与程序的操作速度会降到相当慢。传递多个小型文件时，若是每个文件长度均小于闪存芯片所定义的区块大小时，就可能导致很低的传输速率。访问的迟滞也会影响性能，但还是比硬盘的迟滞影响小。闪存控制器的质量也是影响性能的因素之一。即使闪存只有在制造时做缩小晶粒(die-shrink)的改变，但如果欠缺合适的控制器，就可能引起速度的降级。不同种类、不同工艺、不同技术水平的NAND闪存在读写速率上存在差异。同时，闪存产品的读写性能也与读写方式有关。一般闪存的数据接口为8位或者16位，其中8位较为常见。如果产品支持多通道并行读写，那么就会有更高的速度。

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。潘和刘等人提出通过保留时间放宽来将低成本写入（即使用较粗步长）应用于数据页面。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，高成本的写入降低了以下读取的成本同一页。基于这一特点，当有多个排队请求时，李等人提议应用低成本写入，否则就是高成本写入。吴等人提出应用高成本的写作时，根据他们的估计，下一个访问操作不会因成本增加而延迟。然而，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：如果几乎所有的访问都是一个数据页面被读取（写入）请求，该页面被描述为只读（只写）。如果对数据页面的访问与读写交错，页面被描述为交织访问。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。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着提高整体性能。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;
* 我们提出一个初步的研究来展示潜力AGCR的业绩改善;
* 我们提出AGCR的一个有效的实施与可忽略的开销。跟踪驱动的模拟器和12个工作负载的实验结果表明AGCR实现了显着的性能提升。

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。

### 动机和问题陈述

这部分主要分为读取和写入闪存、读写成本调节器、初步研究三个部分。分别介绍了读取和写入闪存中步长和成本的问题，分析几种策略来利用闪存的可靠性特性来调节读写成本并应用了Li等人提出的成本调节器，并在进行实验来研究不同的写法之间的差异，并从实际工作量中读取成本，包括PROJ，USR，MSR的HM读取和写入成本的三种组合在初步研究中被评估，并根据微软研究院的17个企业级访问研究的比较，由此激发了我们的想法。

* 增量步进脉冲编程（ISPP）是用来编程Flash页面，
* 程序电压随步长增加
* 完成当电压超过阈值电压Vth
* 程序时间→写入成本

写入成本和RBER有关（Raw Bit Error Rate：原始误码率）

低密度奇偶校验码（LDPC）应用于flash中，具有很强的ECC能力。LDPC码的译码强度取决于输入信息的准确性。读写评估这块的两种方法：低成本的写到高成本的读，高成本的读和低成本的写，

#### 2.1 读取和写入闪存

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，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。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ΔVpp的步长。步长ΔVpp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），但是在编程中导致较高的原始误码率（RBER）页面，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过程中，步长较大，写入成本是减少但是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。预计页面的保留时间会更长用更精细的步长编写。在这项工作中，我们专注于读写成本交互，通过调整步长来调整最小值所需的保留时间。

#### 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。潘和刘等人建议通过放宽编程页面的保留时间要求来降低写入成本。吴等人提出了应用高成本的写法来减少在同一页面上执行读取请求的成本。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。李等人提出应用低成本的写入时，有排队的请求，以减少排队延迟，并应用高成本写入，否则，允许低成本读页面。但是，这些作品都没有利用成本监管的准入特征。有几种类型的工作负载访问特性包括冷热通道，相互参照间隙，时间和空间局部等。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模拟验证。监管机构由写入成本调节器和读取成本调节器。写入成本与ISPP的程序步长成反比。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，其与传输的信息的大小成比例。基于调节器的两个部分，低成本的写入使页面具有更高的RBER，从而需要具有更多参考电压的LDPC进一步导致成本较高的读取。另一方面，高成本写入页面的RBER较低，因此需要具有较少参考电压的LDPC，这进一步导致成本较低的读取。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，然后是中等成本读取（MCW/ MCR）;

3）所有的写入都是以低成本执行的，然后是高成本读取（LCW / HCR）。

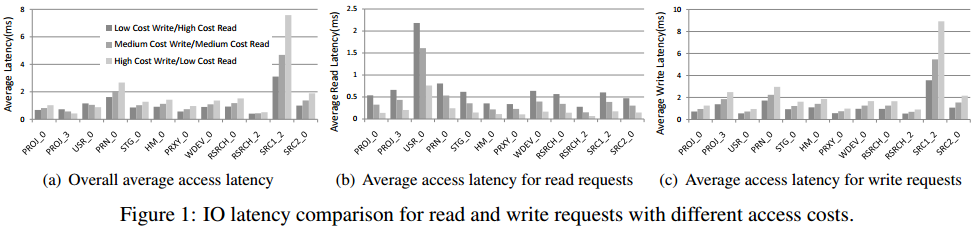


图1显示了来自微软研究院（MSR）剑桥的17个企业服务器代表踪迹的访问延迟比较[17]。相比到默认的MCW / MCR，HCW / LCR将读取性能提高了54％，LCW / HCR将写入性能提高了26％。 显着的性能改进来自相应的访问成本的降低。比较LCW / HCR和HCW / LCR，读写延迟差异分别为114％和61％。绩效差距表明应该认真应用读写成本监管机构。正如图1所示：虽然LCW / HCR能够提高整体性能，但它引入了最差的读取性能。然而，阅读操作总是处于关键路径，这激发了我们的方法。

### ACGR: 接入特性指导下的成本调节

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

为了指导读写成本规则，工作负载的访问特性是非常重要的。例如，如果访问数据页面（主机系统的逻辑页面）受读请求的支配，访问低数据页面成本读取可以显着提高性能。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。我们收集了访问主机系统中所有数据页的特性，并区分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。访问媒体文件时这是典型的或其他只读文件;
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。这在定期的数据冲刷中是典型的从内存系统到存储系统一致性维护;
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叉访问。

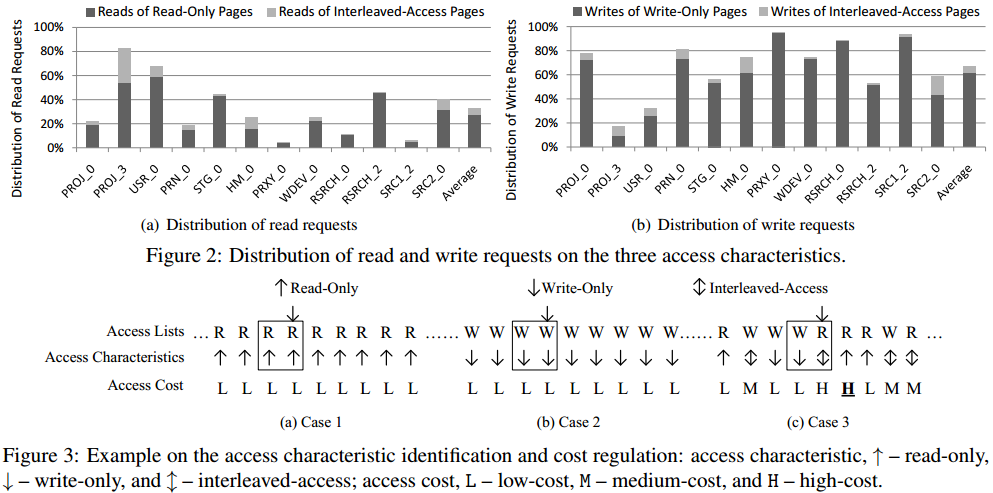


图2显示了这些逻辑数据页访问特性的请求分布。图2（a）示出了只读页面和交织访问页面的读请求的分布。类似地，图2（b）示出了只写页面和交错存取页面的写请求的分布。对于每个轨迹，图2（a）和2（b）的访问总和为100％。我们做图2的三个观察结果：

观察1 - 大多数读取请求访问只读网页平均超过85％

观察2 - 大多数写入请求访问只写平均超过91％;

观察3 - 只有一小部分请求访问交织访问页面。

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

#### 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），该页面被描述为只读（只写）。否则，该页面的特点是交错的访问。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只有读请求，情况2只有写请求，情况3具有交织的读取和写入请求。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（用箭头表示在窗口上方）和最近的请求到这个页面。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，每次访问页面时都会更新标识。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，如果页面的访问特征是交错的，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。识别的准确性方法在3.4节进一步评估，我们在这里分析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。主要思想是将低成本写入应用于只写页面，低成本读取（通过应用高成本写入来启用），用于只读页面，以及用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如下调整：

•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，交错访问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。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重新写入队列，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

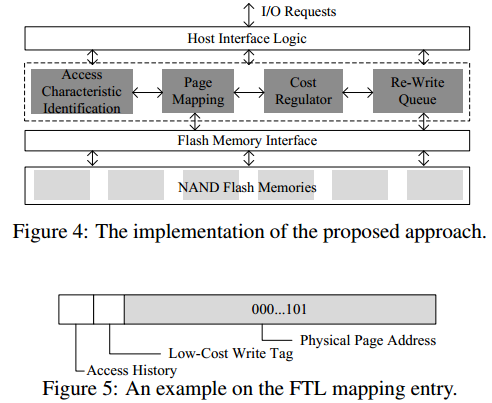
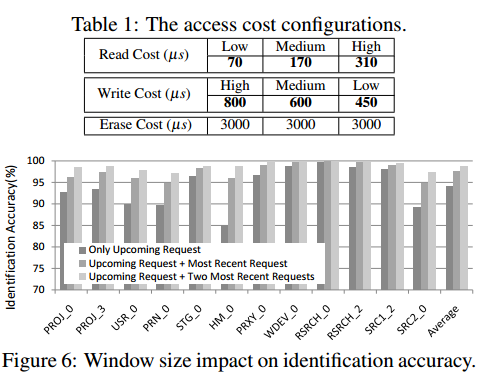


图4显示了闪存控制器中AGCR的实现。我们添加三个新的组件：访问特性标识，成本调节器和重写队列。此外，FTL中的每个映射条目都扩展了两个字段，如图5所示。第一个是访问历史记录，第二个是1位低成本写入标签。当低成本的写入标签被设置为只读页面时，这表示页面必须被重写。当一个I / O请求被发出时，该页面被描述，其成本按照3.3节的规定进行调整。如果读取成本调节需要重新写入操作，则将数据页面的逻辑地址添加到重新写入队列中。在空闲期间，重写操作被触发以高成本对队列中的数据进行编程，这保证了在这些只读页面上的低成本读取。重写开销在实验中被评估。这种实现招致三种类型的开销：存储，硬件和固件开销。存储开销包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），那么对于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。

### 评估

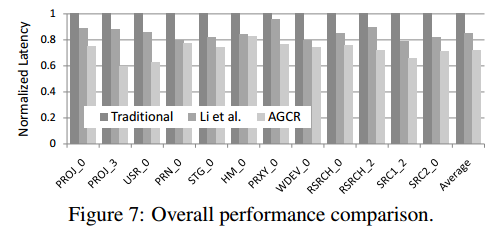
#### 4.1 实验装置

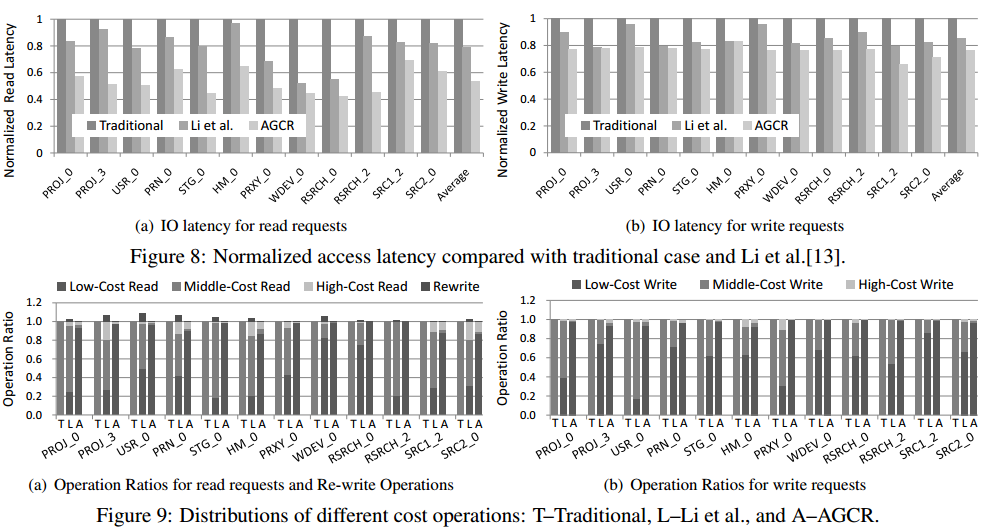
我们使用SSDsim和MSR的工作量来评估我们的成本监管方法。从我们的研究中，我们发现MSR中的所有工作负载具有相似的访问特征。我们选择图2中的12个代表性曲线来进行评估。模拟存储系统配置每个单元MLC两位闪存。它有八个通道，每通道八个芯片，每个芯片四个平面。每个芯片有2048块，每块有64个4KB的页面。基于默认页面映射的FTL，垃圾收集和磨损均衡在模拟器中实现，代表着最先进的存储系统。这些机制的内部读写操作是以规定的访问成本进行处理的。为了公平比较，我们使用Li等人的成本调节器的参数。表1显示了访问成本。擦除操作的成本不取决于写入成本。



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。实验结果。

#### 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。在在传统的情况下，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。成本监管高度取决于AGCR中的访问历史记录。如果访问页面没有历史记录，则写入请求使用高成本写入，读取使用低成本读取要求。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Li等人相比，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（包括快速，中等和慢速读取和写入）的操作分布，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。

### 结论与未来工作

本文的初步研究中，我们提出了一个初步的研究，以显示我们的方法潜在的性能善。并给予了意见，做了观察，大多数读（写）访问只读（只写）页面。方法的途径，提出了一个全面的方法来调节读写成本。评估方法，结果表明，所提出的方法实现了显着的性能改善。

在本文中，我们介绍了AgCr，一种闪存来提高访问性能成本规制的方法。该方法的动机是来自广泛的研究工作的意见。基于这些观察，我们提出了一种精确的访问表征方法，并规范访问成本以提高性能。我们的模拟结果表明，AgCr是有效的，减少I/O由48.3%和20.4%分别为读写请求尽可能多的延迟，与国家的最先进的方法。