

# 新兴云存储与云计算技术

院（系）名 称：计算机学院

专 业 名 称 ： 计算机应用技术

学 生 姓 名 ： 蔡诗晗

二〇一七年十二月

## 摘 要

主机端闪存缓存已经成为云计算系统虚拟机（VM）存储可扩展性问题的有前途的解决方案，但其容量和耐用性仍然面临着严重的限制。基于云的存储提供了可靠性和易管理性。不幸的是，即使使用基于块的重复数据删除和增量压缩等技术，对于存储和传输数据也会产生巨大的成本。提供对以前版本数据的访问的当前趋势加剧了这两种成本。在本文中，我们首先探求和研究了 2016 与 2017 关于云存储和云计算的论文，然后集中对其中体现出较好优越性的一种按需缓存管理解决方案——CloudCache 进行介绍。CloudCache 提出了重用工作集（RWS）和一种动态缓存迁移方法，可满足虚拟机缓存需求，并将缓存磨损降至最低。CloudCache 的有效性将使用现实世界的痕迹来展示全面的实验评估。

**关键词：**云存储；虚拟机；

# ABSTRACT

Host-side flash caching has emerged as a promising solution to the scalability problem of virtual machine (VM) storage in cloud computing systems, but it still faces serious limitations in capacity and endurance. Cloud-based storage provides reliability and ease-of management. Unfortunately, it can also incur significant costs for both storing and communicating data, even after using techniques such as chunk-based deduplication and delta compression. The current trend of providing access to past versions of data exacerbates both costs. In this paper, we first pursue the papers about cloud computing and cloud-based storage of recent years. Then we focus an on-demand cache management solution—CloudCache. CloudCache proposed a new cache demand model, Reuse Working Set (RWS) and a dynamic cache migration to meet VM cache demands and minimize cache wear-out. The effectiveness of CloudCache is demonstrated by comprehensive experimental evaluations using real-world traces.

**Key words:** Cloud-based storage; virtual machine;

# 目 录

1、 引 言 .....	1
1.1 基于云存储的新技术 .....	1
1.2 云计算技术 .....	2
1.3 基于闪存的存储器 .....	3
1.4 论文的组织结构 .....	4
2、 新按需缓存管理解决方案 .....	5
2.1 体系结构 .....	5
2.2 按需缓存分配 .....	5
2.2.1 基于 RWS 的缓存需求模型 .....	6
2.2.2 在线缓存需求预测 .....	7
2.2.3 缓存分配和准入 .....	8
3、 实验 .....	9
3.1 预测精确度 .....	9
3.2 分段策略 .....	10
3.3 WSS 与基于 RWSS 的缓存分配 .....	11
3.4 动态缓存迁移 .....	12
3.4.1 实时缓存迁移 .....	12
3.4.2 迁移率限制 .....	13
4、 结论与展望 .....	14

# 1、 引言

## 1.1 基于云存储的新技术

主机端闪存缓存在虚拟机（VM）主机上采用基于闪存的存储作为其远程存储的缓存，以利用数据访问位置并提高虚拟机性能。近年来备受关注，这可以归结为两个重要的原因。首先，随着云计算系统的整合水平不断提高，共享虚拟机存储服务器的可扩展性成为一个严重的问题。其次，基于闪存的存储的出现使得闪存缓存成为解决 IO 可扩展性问题的一个有希望的选择，因为访问本地闪存缓存比通过网络访问远程存储快得多。

但是，由于闪存设备的容量和成本的限制，与虚拟机的数据集大小相比，可以在主机上使用的闪存高速缓存的数量是非常有限的，尤其是考虑到增加强度的工作负载和越来越多的数据通过虚拟化工作负载整合到主机。因此，为了实现闪存缓存的潜力，根据实际需求在竞争的虚拟机之间分配共享缓存容量是非常重要的。此外，闪存设备因写入而磨损，并且面临严重的耐久性问题，事实上由于缓存的使用而加剧了这一问题，因为工作负载固有的写入和错过缓存的读取都会导致磨损。因此，高速缓存管理还需要小心，不要承认对工作负载性能无用的数据，只会损害耐久性。

有几个相关的闪存缓存管理解决方案。S-CAVE 在估计虚拟机的缓存需求时考虑重用块的数量，并使用多种启发式方法分配缓存。vCacheShare 通过最大化一个统一函数来分配一个只读缓存，该函数捕获虚拟机的磁盘延迟，读写比，估计缓存命中率和分配缓存容量的重用率。Centaur 使用 MRC 和延迟曲线根据其 QoS 目标为虚拟机分配缓存。但是，这些解决方案将所有引用的数据都纳入缓存中，包括那些时态局部性较弱的缓存，并且会导致不必要的缓存使用和磨损。而且，他们都没有考虑动态缓存迁移，以满足缓存过载时的需求。

HEC 和 LARC 研究了高速缓存许可政策，以滤除时间局部性较弱的的数据，并减少了闪存的磨损。但是，他们没有考虑如何将共享缓存容量分配给并发工作负载的问题。

Bhagwat 等人研究了如何允许被迁移的虚拟机向其先前主机上的缓存请求数据，这与本文提出的按需缓存迁移方式相同。但是，单独使用按需迁移不能保证迁移的虚拟机具有良好的性能。它在性能干扰和缓存利用率方面也对源主机有着持久的

负面影响。当迁移的虚拟机的数据在源主机上被逐出时，性能变得更糟，因为请求必须由源主机转发到主存储器。VMware 的 vSphere 闪存读取缓存[6]也支持后台缓存迁移。虽然它的细节是未知的，没有我们提出的 RWS 模型，类似的解决方案将不得不迁移虚拟机的整个缓存痕迹。这需要更长的迁移时间，并对性能造成更大的影响。

与这些处理器和内存缓存工作相比，闪存缓存管理提出了一系列不同的挑战。低地点数据不仅对闪存高速缓存的性能有害，而且对它的生命周期也是有害的，特别是在闪存高速缓存层。虽然虚拟机迁移可用于在主机之间迁移工作负载，但大量的缓存数据不能简单地翻转或轻松发送。

## 1.2 云计算技术

存储系统的两个趋势是共谋增加存储和检索数据的成本。首先，由于管理，成本效益和可靠性方面的益处，企业和消费者将更多的数据存储基于云的存储提供商处。然而，从远程网站写入和读取数据可能会导致网络通信的巨大成本。

其次，客户越来越多地希望获得更多的数据。本地存储解决方案，如苹果公司的 Time Machine，保留了用户数据的多个版本，使得访问这些数据变得非常容易。云存储提供商也纷纷效仿，例如 GoogleDrive，微软 OneDrive 和 DropBox 都存储并允许用户访问其文件的旧版本。

过去的版本有很多用途，例如，恢复丢失或重写的的数据，再现创建数据的过程，审计以及取证故障排除。版本的好处随着更多版本的保留而增加。例如，如果在每次关闭文件时仍然保留，则用户通常保证每次保存操作或应用程序终止时的文件的快照。但是，许多应用程序仅在终止时保存数据；在这种情况下，应用程序使用过程中创建的所有中间数据都不可用于恢复和分析。保存每个文件系统修改的数据会产生更频繁的检查点，但不能恢复内存中永远不会写入文件系统的瞬态，重要的是，不捕获修改存储器映射文件。最重要的是，用户应该能够重现过去的状态文件系统或应用程序内存中，我们称之为 eidetic 版本的属性。

版本控制的成本还取决于版本的保留频率。例如，在每一个文件修改中保留一个版本会导致更大的存储成本，而且每个关闭的存储都会减少，而客户将通过减少更多的存储来消耗更多的带宽。版本化策略必须平衡这些好处和成本。许多当前的系统选择偶发版本作为结果。

等价原则（价值与计算之间）可以表示由计算产生的数据的原理，或者是通过值或者重现计算所需的输入的记录。它已被观察到并在许多环境中使用；例如容错，状态机复制，数据中心存储管理和状态同步。

使用等价原则，即减少分布式文件系统中客户端和服务端之间的通信开销。Lee 等人首先在 Coda 文件系统中应用这个原则，并创造了术语运输航运。客户端登录并将用户操作（例如，shell 命令）发送到服务器代理，以重播操作以重新生成数据。Chang 等人 扩展这个想法来记录和发送用户活动，如键盘和鼠标输入。

虽然运行运输的基本思想是强大的，但是先前的系统记录和运输非确定性的类型非常有限，因此不能保证通过日志接收到的状态与原始状态相匹配。shell 命令的日志和用户活动日志都不足以重现通用程序的计算。研究人员认识到了这个缺点，并通过补充重复计算与前向纠错和补偿行动，使用散列检测剩余的差异和恢复到价值运输补偿。不幸的是，转向多处理器和多线程程序意味着许多程序在这些现有系统中不能处理的方式是不确定的。此外，由于这些现有系统处理的非确定性输入非常有限，所以它们在记录和重放方面需要相同的环境，这在许多客户端 - 服务器设置中是不现实的。

### 1.3 基于闪存的存储器

基于闪存的存储器的出现极大地催化了在基于 DRAM 的主存储器和基于 HDD 的主存储器之间采用新的基于闪存的缓存层。它有潜力解决公共和私有云计算系统等高度整合的系统所面临的严重的可扩展性问题。这些系统通常使用共享网络存储来存储分布式虚拟机主机虚拟机镜像，以提高资源利用率，方便虚拟机管理。虚拟机主机上闪存缓存的可用性可以使用本地闪存设备上缓存的数据来加速虚拟机数据访问，这比访问基于硬盘的网络存储要快得多。即使随着闪存设备作为主存储器的采用越来越多，闪存技术的多样性允许使用更快且更小的闪存设备作为用于较慢但较大的闪存设备的缓存用作主存储器。

为了实现闪存缓存的潜力，采用按需缓存管理是至关重要的，即根据其需求在竞争的工作负载之间分配共享的缓存容量。商品闪存设备的容量通常远远小于单个主机上的虚拟机的数据集大小。虚拟机如何共享有限的缓存容量，不仅关系到它们的性能，还关系到闪存设备的耐用性。一方面，如果无法有效缓存工作负载的必要数据，则会从存储服务器获取错过的数据，并且延迟服务器的其他工作负载，从

而导致数量级的延迟。另一方面，如果工作负载占用了不必要数据的缓存，则浪费了宝贵的缓存容量，并危及了需要该空间的其他工作负载。与 CPU 分配不同的是，工作负载不能超过其需求，活动缓存工作负载可占用超出其实际需求的所有分配空间，从而损害其他工作负载的性能以及闪存设备的耐用性。

CloudCache 解决方案，通过按需缓存管理来解决闪存缓存中的问题。具体来说，它回答了两个具有挑战性的问题。首先，如何根据缓存需求为虚拟机分配闪存缓存？为了解决这一难题，一个新的缓存需求模型——重用工作集（RWS）被提出，来捕获具有良好时间局部性的数据，并且使用重用工作集大小（RWSS）来表示工作负载的缓存需求。基于此模型，可以进一步使用预测方法在线估计工作负载的缓存需求，并使用新的缓存允许策略来仅允许 RWS 进入缓存，从而在减少磨损的同时为工作负载提供了良好的性能。

第二个问题是如何处理虚拟机缓存需求超过闪存缓存容量的情况。为了解决这个问题，一种动态缓存迁移方法被提出，通过将缓存的数据与虚拟机一起实时迁移来平衡主机之间的缓存负载。它使用脏数据的按需迁移，为迁移虚拟机提供零停机时间，以及 RWS 的后台迁移，以便快速预热虚拟机的缓存，从而最大限度地降低对性能的影响。

基于块级虚拟化的 CloudCache 被实际实现。这个实验使用从现实世界中的云系统收集的一组长期痕迹来评估它。结果表明，基于 RWSS 的缓存分配可以在最坏的情况下以很小的性能损失为代价，大幅降低缓存使用率和磨损。

CloudCache 是第一个提出 RWSS 模型，用于从具有良好的本地性的数据中捕获工作负载的缓存需求，并指导闪存缓存分配以实现良好的性能和耐久性。它也是第一个提出动态缓存迁移来平衡跨分布式闪存缓存的负载，并通过优化来最大限度地减少缓存数据传输的影响。

## 1.4 论文的组织结构

本论文组织如下：

第一章主要介绍今年来的云计算和云存储的新进展

第二章介绍一种新的按需分配的管理解决方案——CloudCache；

第三章展示了实验和结果

第四章将对全文进行总结，并且对后续的研究方向进行展望。



## 2、 新按需缓存管理解决方案

### 2.1 体系结构

CloudCache 支持基于图 2.1 所示的典型闪存缓存架构的按需缓存管理。CloudCache 按需分配闪存缓存到本地虚拟机，动态虚拟机和缓存在主机之间迁移，以满足虚拟机的缓存需求。CloudCache 将以缓存虚拟机磁盘文件的方式设法在缓存虚拟机块设备上上进行分配和迁移。

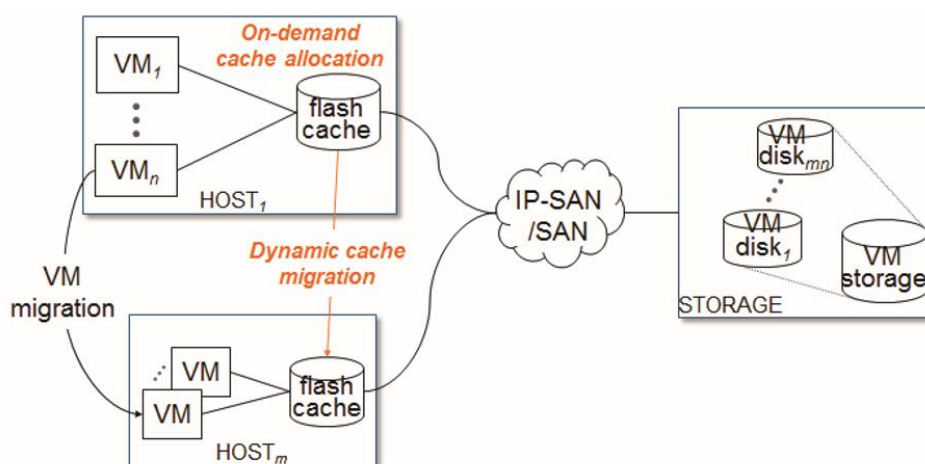


图 2.1 CloudCache 体系结构

CloudCache 支持不同的写入缓存策略：（1）写入无效（2）写入（3）回写。写入无效策略对于写入密集型工作负载的性能不佳。当写入异步提交到存储服务器并且服务器的负载较轻时，直接写入策略的性能接近于回写。

在接下来的几节中，我们将介绍 CloudCache 的两个组件，按需缓存分配和动态缓存迁移。

### 2.2 按需缓存分配

CloudCache 解决了关于按需缓存分配的两个关键问题。首先，如何建模工作负载的缓存需求？云工作负载包括具有不同级别的时间局部性的 IO，其不同地影响高速缓存命中率。一个好的缓存需求模型应该能够捕获对于工作负载性能来说真正重要的 IO，以便最大限度地提高性能，同时最大限度地减少缓存利用率和闪存磨损。其次，如何使用缓存需求模型来分配缓存并将数据导入缓存？为了指导缓存的分配，我们需要准确预测工作负载的缓存需求，并且只允许有用的数据进入缓存，

这样分配就不会超过缓存。在本节中，我们将介绍 CloudCache 解决这两个问题的方法。

### 2.2.1 基于 RWS 的缓存需求模型

工作集 (WS) 是经常用来估计工作负载的缓存需求的经典模型。时间  $t$  的工作集  $WS(t, T)$  被定义为时间间隔  $[t - T, t]$  中由工作负载引用的一组不同的 (地址方式) 数据块。该定义使用局部性原则来形成工作负载将接下来要访问的块集合的估计值，并将其保存在缓存中。工作集大小 (WSS) 可用于估计工作负载的缓存需求。

为了解决 WS 模型的局限性，我们提出了一个新的缓存需求模型 Reuse Working Set,  $RWSN(t, T)$ ，它被定义为一组工作负载重用的不同的 (地址方式) 数据块在时间间隔  $[t - T, t]$  内至少  $N$  次。与 WS 模型相比，RWS 仅捕获具有时间局部性的数据块，这将有益于工作负载的缓存命中率。当  $N = 0$  时，RWS 减少到 WS。然后，我们建议使用重用工作集大小 (RWSS) 作为工作负载的缓存需求的估计。由于 RWSS 忽略低地点数据，因此可以更准确地捕获工作负载的实际高速缓存需求，并减少高速缓存污染和由此类数据引起的不必要磨损。

为了证实 RWS 模型的有效性，我们使用不同的  $N$  值来分析 MSR Cambridge 轨迹，并评估对高速缓存命中率的影响，高速缓存使用发送到缓存设备的写入数量与缓存接收的 IO 数量。我们假设一个数据块只有在被访问了  $N$  次之后才被允许进入缓存，即我们只缓存工作负载的  $RWSN$ 。图 2.2 显示了来自 36 个 MSR 痕迹的这些度量的分布，使用显示四分位数的晶须的箱形图。

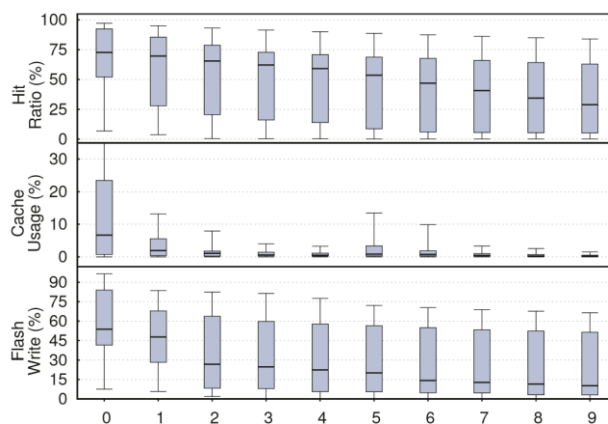


图 2.2 使用不同  $N$  值分析 RWS

这些结果证实了使用 RWSS 估计高速缓存需求的有效性——它能够以较小的命中率成本大幅降低工作负载的高速缓存使用率及其导致的磨损。系统管理员可以通过为 RWS 模型选择合适的 N 来平衡性能与缓存使用和耐用性之间的关系。在本文的其余部分，我们使用  $N=1$  来进行基于 RWSS 的缓存分配。而且，在考虑共享缓存不能适应所有工作负载的工作集的云使用场景时，使用 RWS 模型来分配缓存容量可以实现更好的性能，因为它阻止了低地点数据将有用数据从缓存中移出。

为了衡量工作量的 RWSS，我们需要确定合适的时间窗口来观察工作量。这里有两个相关的问题。首先，如何跟踪窗口？我们使用基于实时的窗口来观察工作负载的 RWSS。

第二个问题是如何决定时间窗口的大小。我们在一段时间内完成工作量的分配，并根据使用不同大小的窗口测量的 RWSS 来为工作负载分配空间时模拟缓存命中率。然后我们在这个命中率对比窗口大小模型的“拐点”处选择窗口

我们给出一个使用两周的 Moodle 轨迹来估计窗口大小的例子。图 2.3 显示，随着窗口大小的增加，命中率迅速增加。我们选择 24 到 48 小时作为测量此工作负载的 RWSS 的窗口大小，因为较大的窗口大小无法获得足够的命中率来证明工作负载的缓存使用率进一步增加，如果我们基于观察到的 RWSS 分配缓存。如果工作负载中的命中率随着窗口大小的增加而缓慢增长，但没有明显的拐点，窗口大小应设置为一个很小的值，因为它不会影响命中率，但可以节省缓存空间的其他工作负荷，并有明确的拐点。

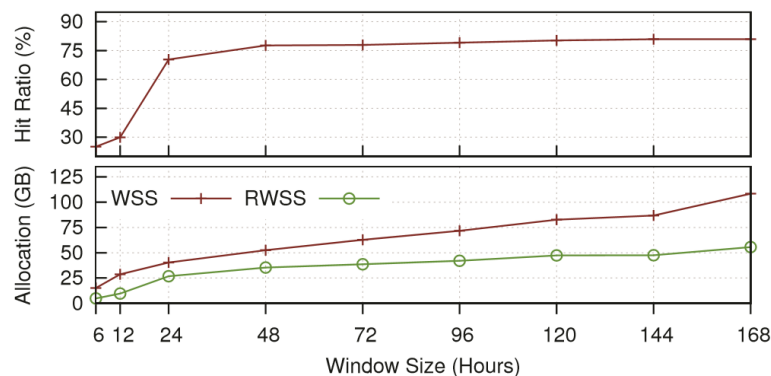


图 2.3 Moodle 痕迹的时间窗口分析

### 2.2.2 在线缓存需求预测

基于 RWSS 的缓存分配的成功与否取决于我们是否能够基于从先前的窗口观

察到的 RWSS 值准确地预测下一时间窗口的缓存需求。为了解决这个问题，考虑经典的指数平滑和双指数平滑方法。

图 2.4 显示了三个星期的 Web 服务器跟踪的 RWSS 预测的示例。

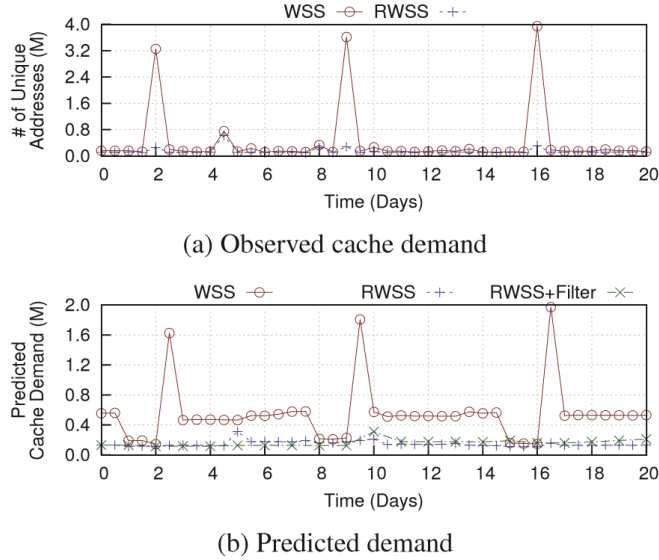


图 2.4 基于 RWSS 的缓存需求预测

### 2.2.3 缓存分配和准入

基于 RWSS 的高速缓存分配方法还需要一个基于 RWSS 的高速缓存许可策略，只允许重用的数据块进入高速缓存。CloudCache 使用主存的一小部分作为引用地址的暂存区域，这是实现缓存准入的常用策略。一个块只有在被访问了  $N$  次后才被允许进入缓存，无论它们是读取还是写入。暂存区域的大小是有限的，当它满了时，使用 LRU 驱逐暂存的地址。我们将这种方法称为仅寻址主内存中地址分段的分段策略。

CloudCache 还考虑了缓存准入的数据分段策略，该策略将候选块的地址和数据都存储在分段区域中，并使用 LRU 管理它们。因为主内存不是持久的，所以更确切的说，只有读请求返回的数据在内存中暂存，而写内容只是它们的地址被暂存。这种策略可以在读取访问缓存之前，通过从临时区域提供读取访问来减少读取错过。权衡是因为数据块比地址大得多，对于相同的分段区域，数据分段可以追踪比地址分段少得多的参考，并且可能遗漏具有良好时间局部性的数据。

CloudCache 考虑了第三种混合分段策略，其中暂存区域分为存储地址和数据，地址和数据分区域分别由 LRU 管理。这种策略通过使用数据分段来减少具有小重用距离的块读取错失，并通过使用地址分段来允许具有相对较大复用距离的块。

### 3、 实验

CloudCache 通过向虚拟机提供虚拟块设备并透明地将其数据访问缓存到通过网络访问的远程块设备而在块级虚拟化中创建。它包括一个实现虚拟块设备、监视虚拟机 IO、强制缓存分配和许可的内核模块，以及测量和预测 RWSS 并确定虚拟机的缓存份额的用户空间组件。内核模块将最近观察到的 IO 存储在一个小的循环缓冲区中供用户空间组件使用。CloudCache 基于 Linux，还创建了一个用户级的 CloudCache 缓存模拟器。

#### 3.1 预测精确度

在第一组实验中，我们评估了之前考虑的不同 RWSS 预测方法：（1）用  $\alpha=0.3$  指数平滑，（2）Exp self，指数平滑的自调整版本，（3）DExp （4）DExp self，双指数平滑的自调整版本，以及（5）Last 值，这是一种简单的方法，它使用最后观察到的 RWSS 值作为新窗口的预测值。

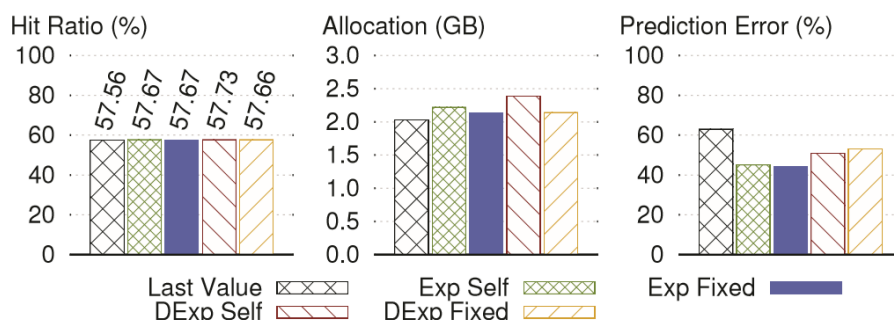


图 3.1 预测精确度

图 3.1 比较了三种不同的预测方法：（1）命中率，（2）缓存分配和（3）预测误差——预测 RWSS 和观测 RWSS 之差除以观测 RWSS 的差值的绝对值。预测误差会影响其他两个度量标准——预测不足会增加缓存未命中而过度预测使用更多缓存。该图显示了整个 9 个月 Web 服务器跟踪的所有时间窗口中这些度量的平均值。

结果表明，在不同的预测方法中，命中率的差异很小，但是在缓存分配方面却相当可观。

### 3.2 分段策略

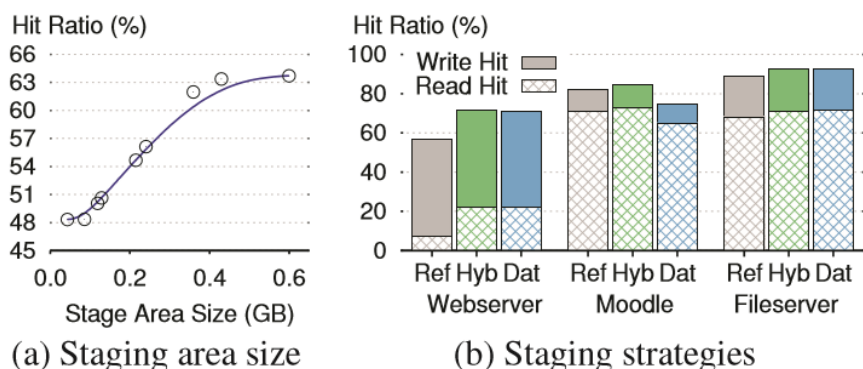


图 3.2 分块策略分析

在第二组实验中，我们评估了 CloudCache 的分段策略。首先，我们研究分期面积大小的影响。图 3.2a 给出了一个示例，说明在使用地址分段时，分段区域分配如何影响 Web 服务器工作负载的命中率。数据分级的结果是相似的。

我们还将相关工作与高耐用性高速缓存 (HEC) [33] 相比较。HEC 的触摸计数 (TC) 技术使用内存位图来跟踪所有的缓存块 (默认为 4MB)，只允许重用的块进入缓存。相比之下，CloudCache 只跟踪少量最近访问的地址，以限制内存使用量，并防止过早访问的数据块进入缓存。HEC 的选择性顺序排斥 (SSEQR) 技术跟踪访问的顺序，并拒绝长序列 (默认情况下，长度超过 4MB 的顺序)。相比之下，CloudCache 使用分段区域自动过滤掉长扫描序列。

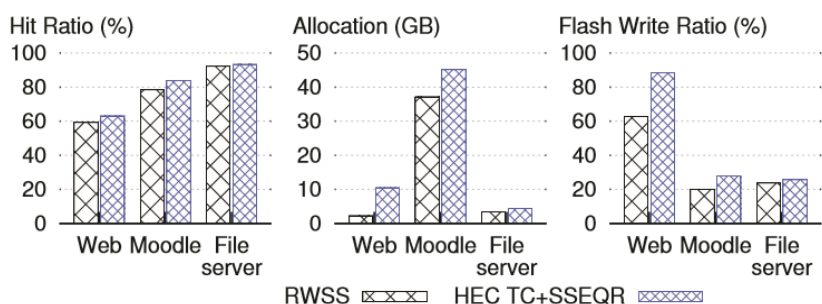


图 3.3 与 HEC 比较

图 3.3 显示了使用不同轨迹的比较。结果证实，RWSS 方法能够自动拒绝扫描序列，而 HEC 需要使用固定阈值。



### 3.3 WSS 与基于 RWSS 的缓存分配

在第三组实验中，我们使用相同的预测方法比较基于 RWSS 的缓存分配和基于 WSS 的缓存分配，指数平滑和自调整。在这两种情况下，都会严格执行缓存分配，并且在每个窗口开始时，工作负载超出其新分配的额外缓存使用率将立即下降。

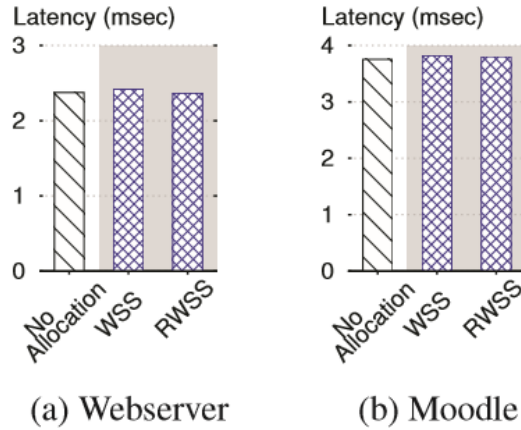


图 3.4 虚拟机延迟率比较

图 3.4 显示了这些不同方法之间的比较。为了评估命中率降低会造成多少性能损失，我们重播这些痕迹并测量它们的 IO 延迟。我们考虑一个月份的 Web 服务器和 Moodle 痕迹。他们在之前指定的真实虚拟机存储和缓存设置中重播。我们比较不同的缓存管理方法在第 95 百分位的 IO 延迟。图 3.5 显示了基于 RWSS 的方法提供了类似于替代方法的性能，同时使用更少的缓存并导致更多写入缓存设备，如前面的结果所示。

结果证实，我们提出的基于 RWSS 的缓存分配确实可以大大降低工作负载的缓存使用率和相应的磨损，只需很小的性能成本。

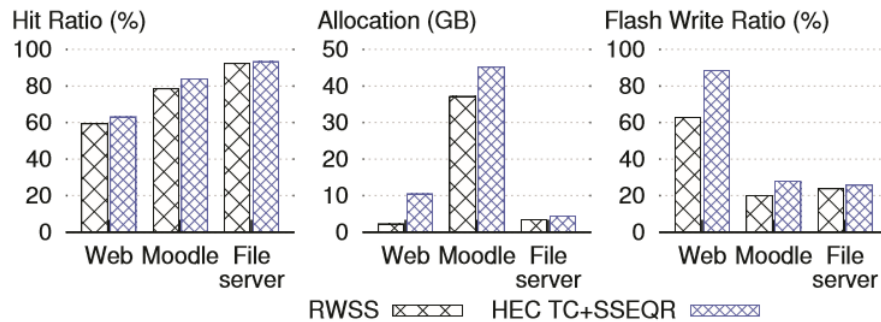


图 3.5 分配方法

### 3.4 动态缓存迁移

本节将介绍动态高速缓存迁移方法，通过动态迁移工作负载的缓存数据以及虚拟机来平衡不同主机之间的高速缓存负载。它还考虑技术来优化迁移虚拟机的性能，并尽可能减少迁移过程中对其他虚拟机的影响。

#### 3.4.1 实时缓存迁移

实时虚拟机迁移允许工作负载在虚拟机中运行时在物理主机之间透明地迁移。在 CloudCache 中，我们建议使用实时虚拟机迁移来平衡虚拟机主机的闪存缓存上的负载——当主机的缓存容量不足以满足本地虚拟机的总缓存需求（由其预测的 RWSS 估计）时，一些虚拟机可以被迁移到具有空闲缓存容量的其他主机以满足其缓存需求。

基于虚拟机迁移的高速缓存负载平衡提出了两个挑战。首先，迁移源主机上的迁移虚拟机脏缓存数据必须先同步到目标主机，然后再由虚拟机访问。一个天真的方法是将所有脏数据转移到迁移虚拟机的远程存储服务器。根据脏数据量和可用的 IO 带宽，浮点运算可能非常耗时，并且在浮点运算完成之前，虚拟机不能恢复其活动。闪存也会导致存储服务器的 IO 负载激增，影响其他共享服务器的虚拟机的性能。其次，迁移虚拟机需要预热目标主机上的缓存，这也可能需要很长时间，并且在缓存被预热之前会经历大量的性能下降。

为了应对这些挑战，CloudCache 的动态缓存迁移方法结合了被动迁移技术和主动迁移技术：

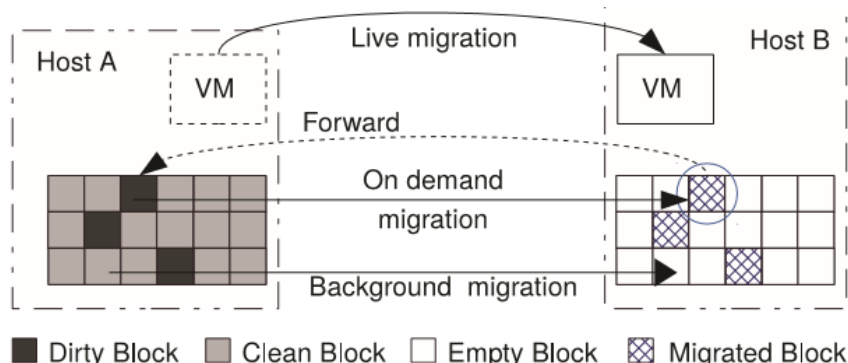


图 3.6 动态缓存迁移体系结构

按需迁移允许迁移的虚拟机快速访问其脏块，但不能传输多个块。后台迁移可以有效地传输批量数据，但可能无法为迁移的虚拟机正在等待的当前请求提供服



务。因此,这两种迁移策略的结合可以优化虚拟机的性能。图 3.6 显示了 CloudCache 如何执行缓存迁移。

CloudCache 根据云系统中虚拟机的唯一 ID 为虚拟块设备分配唯一的名称。在迁移之前,在目标主机上创建从虚拟块设备到物理设备的映射,在迁移之后,源主机上的对应设备将被删除。

### 3.4.2 迁移率限制

虽然按需迁移和后台迁移相结合可以优化迁移虚拟机的性能,但也需要考虑对源主机和目标主机上其他虚拟机的影响。高速缓存迁移要求读取源主机的高速缓存并写入目标主机的高速缓存,这可能会降低其他共同托管的虚拟机的高速缓存 IO。除了虚拟机内存迁移已经消耗的带宽之外,还需要网络带宽,并影响其他虚拟机的网络 IO 性能。

为了控制对共同托管虚拟机的性能干扰,CloudCache 能够限制高速缓存迁移的传输速率。考虑到速率限制,它强制每个时间周期从源主机到目标主机传输的数据块的最大数量,包括按需迁移和后台迁移。一旦达到限制,迁移线程将休眠并等待下一个时间段继续数据传输。如果需要请求睡眠时间,在线程唤醒后,它们将被延迟并立即服务。速率可以根据虚拟机的优先级和迁移虚拟机的 RWSS 等因素设置。CloudCache 允许系统管理员调整速率,以最大限度地减少对共同托管虚拟机的高速缓存迁移影响,并且仍然足够快地迁移 RWS 以满足高速缓存需求。

## 4、 结论与展望

本文首先对近年的云计算和云存储技术进行分析和研究，然后重点介绍了 CloudCache，这是一个针对这些问题的按需缓存管理解决方案。它采用新的缓存需求模型 Reuse Working Set (RWS) 来捕获具有良好时间局部性的数据，根据预测的重用工作集大小 (RWSS) 分配缓存空间，并且只允许 RWS 进入分配的空间。其次，为了处理缓存过载的情况，CloudCache 采用了一种新的缓存迁移方法，该方法使用缓存数据实时迁移虚拟机，以满足虚拟机的缓存需求。基于现实世界的痕迹的广泛评估证实，基于 RWSS 的缓存分配方法可以为虚拟机实现良好的缓存命中率和 IO 延迟，同时大幅降低缓存使用率和闪存磨损。

## 参考文献

- [1] Arteaga D, Cabrera J, Sundararaman S, et al. CloudCache: on-demand flash cache management for Cloud Computing[C]// Usenix Conference on File and Storage Technologies. USENIX Association, 2016:355-369.
- [2] Xianzheng D. Peter M. C., Jason F.: Knockoff: Cheap Versions in the Cloud[C]// Usenix Conference on File and Storage Technologies. USENIX Association, 2017:73-88