毕业设计（论文）

译文及原稿

译文题目 FairRide：接近最优的公平缓存共享机制

原稿题目 FairRide: Near-Optimal, Fair Cache Sharing

原稿出处

Qifan Pu, Haoyuan Li (UC Berkeley), Matei Zaharia (MIT), Ali Ghodsi, Ion Stoica (UC Berkeley) 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’16).

FairRide：接近最优的公平缓存共享机制

摘要—

对许多系统来说内存缓存仍然是一个关键组件。近年来，更大量的数据储存在主内存中，特别是在内存共享的例如云这样的环境中。此类环境的性质要求资源分配系统为多个用户/应用程序提供性能隔离，并保证系统的高可用性。我们研究了使用共享文件为多个用户公平分配内存缓存的问题。研究发现，目前没有内存分配策略可以提供通常可由其他类型的资源（例如CPU或网络）实现的所有三个期望属性（隔离保证，策略证明和帕雷托最优）。我们也注意到了目前存在实现三个属性中任意两个的策略。其中阻塞是实现隔离保证和策略证明的唯一方法，我们在一项名为FairRide的新策略中有效地进行了调整。我们在常用的内存中心存储系统中使用高效的阻塞形式实现该策略，命名为预期延迟，并证明FairRide可以在许多情况下提高缓存效率（2.6倍过度分离缓存）并保证公平性。

1简介

高速缓存是大多数计算机系统的关键组件，其特征在于两个特征：它们对应用程序性能有显著的影响，以及与数据总量相比他们的容量有限。随着内存缓存越来越多地被用于大规模数据处理集群以及数据库和键值存储，缓存在今天的多用户云中也起着关键作用。但是，在具有多个用户的共享环境中，管理缓存的问题变得更加困难：提供商应该如何在多个用户之间分配空间，每个用户都希望将自己的数据集保存在内存中吗？

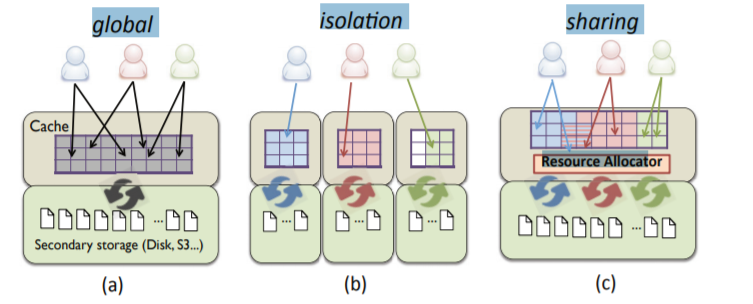
遗憾的是，传统的缓存策略无法为此问题提供令人满意的答案。大多数高速缓存管理算法（例如，LRU，LFU）都关注于高速缓存的全局效率（图1a）：它们旨在最大化总命中率。无论在当今的云服务缓存系统（Redis ，Memcached）和大数据存储（HDFS Caching）中常用，这在共享环境中存在两个问题。首先，读取数据间隔时间长的用户可能从缓存中获得很少甚至无法获得收益，因为它们的数据很可能被内存淘汰策略删除。其次，应用程序也可以通过虚假访问来提高其访问速率，从而轻易滥用此类系统。没有动机阻止用户在云环境中执行此操作，更重要的是，即使非恶意应用。我们稍后会说明，有策略的用户可以通过对她的文件进行虚假访问来超过普通用户2.9倍。

图1：不同的方案。

全局（global）：单个内存池，用户或应用程序不可知;

隔离(isolation)：多个用户之间的内存静态分配，可能利用不足（空白单元），无共享;

共享(sharing)：允许在用户之间动态分配内存，以及共享文件的一个副本（条型单元）。

另一种常见的方法是为每个用户提供隔离的缓存（图1b）。这为每个用户提供了性能保证，并且在实际生产环境中有许多例子，例如，为每个客户虚拟机设置单独的缓冲区缓存的虚拟机管理程序，启动的Web托管平台每个用户都有一个单独的memcached实例。但是，提供这种性能保证是以降低高效率利用率为代价的。

这种低效率不仅是因为用户没有充分利用其分配的缓存，还因为一次可以由多个用户访问该缓存文件并且隔离缓存导致这种共享文件创建了多个副本。 我们发现这种非排他性共享定义了高速缓存分配的一个方面，而其他资源通常是独占的，例如，CPU时间片或通信链路一次只能由单个用户使用。 在实践中，在许多工作负载中的用户之间共享了大量文件，例如，我们观察到来自生产HDFS日志的至少两个用户共享了超过30％的文件。 随着更多工作负载转移到多租户环境，这种共享可能会增加。

在本文中，我们将研究如何在访问共享文件的多个用户之间共享缓存空间。为了解决问题，我们首先确定我们希望分配策略具有的理想属性。在共享CPU和网络资源的公共属性的基础上，我们确定了三个这样的属性：

•隔离保证：如果缓存空间在所有用户之间静态且平均分配（即假设n个用户和相等的共享，每个用户将获得1 / n的缓存空间），则用户不应该获得比缓存空间少的缓存空间。这也意味着用户的高速缓存性能（例如，高速缓存未命中率）不应该比隔离差。

•策略证明：用户通过利用系统（例如通过虚假访问文件）来改善其分配或缓存性能时不能以牺牲其他用户为代价。

•帕累托最优：系统应该是高效的，因为在不降低其他用户的分配的情况下不可能增加一个用户的缓存分配。该属性表面了操作员实现高利用率的愿望。

这些属性是适用于大多数资源共享方案的分配策略的常见功能，包括通过抽取或调度进行CPU共享，通过max-min公平性进行网络链接共享，甚至为计算任务分配多个资源。出乎意料的是，没有用于分配满足所有三个属性的缓存空间的等效策略。如前所示，全局共享策略（图1a）缺乏隔离保证和策略证明，而静态隔离（图1b）不是帕累托最优。

我们发现的第一个令人惊讶的结果是这种缺陷并非偶然：实际上，对于共享缓存资源，没有策略可以实现所有三个属性。简单来说，这是因为缓存数据可以在多个用户之间共享，允许用户通过“搭便车”对其他人缓存的文件进行利用，或者通过缓存流行文件来优化使用。这在帕累托效率和策略证明之间创造了强有力的权衡。

虽然没有内存分配策略可以满足这三个属性（表1），但我们表明有些策略在实践中接近实现这三个属性。 于是，我们提出了FairRide，这种策略既提供隔离保证（因此它总是表现不比隔离的缓存更差）和策略证明（因此用户不会被鼓励作弊），并且在实践中效率达到全球的前4％。 FairRide通过概率阻止（第3.4节）将每个用户的收益 - 成本比率与使用偏好匹配，即，如果一个文件不在用户的缓存习惯中，则概率性地禁止用户访问缓存文件。 我们在第5节中的证明表明，需要阻塞来实现策略验证，并且FairRide可以实现具有最小阻塞的属性。

实际上，可以使用预期的延迟（第4.1节）有效地实现概率阻塞，以减轻I / O开销并防止更复杂的欺骗模型。 我们在Tachyon [26]上实施了FairRide，这是一个以内存为主的存储系统，并使用云服务和大数据工作负载评估系统。 我们的评估显示，FairRide效率达到全球缓冲系统的前4％，同时阻止了策略用户，同时比孤立的缓存减少了2.6倍的作业运行。 在用户做作弊的非法环境中，FairRide在效率方面优于最高最低公平性至少27％。 值得注意的是，FairRide将支持可插拔替换策略，因为它仍然遵循每个用户的缓存首选项，这允许用户选择最适合其工作负载的不同替换策略（例如，LRU，LFU）。

2背景

今天的大多数缓存系统都不知道访问数据的实体（用户）：CPU缓存不关心哪个线程访问数据，Web缓存不关心哪个客户端读取网页，以及基于内存的系统如Spark [41 ]不关心哪个用户读取文件。 相反，这些系统旨在最大化系统效率（例如，最大化命中率），并且因此有利于用户在牺牲其他用户资源的前提下提供更多资源以提高效率（例如，用户以更高的速率访问数据）。

为了说明这些缓存系统的不公平性，请考虑托管服务的典型设置，如图2a所示。 我们设置了多个托管站点，它们共享一个Memcached [7]缓存系统，以加快对后端数据库的访问。 假设A和B的负荷最初是相同的。 在这种情况下，正如预期的那样，两个站点的平均请求延迟大致相同（参见图2b中的左侧栏）。 接下来，假设站点A的负载显着增加。 尽管B的负载保持不变，但其请求的平均延迟显着增加（2.9倍），并且A请求的延迟意外的下降了！ 因此，A的负载增加会改善A的性能，但会降低B的性能。这是因为A更频繁地访问数据，并且作为响应，缓存系统开始从A中加载更多结果，同时淘汰B的结果

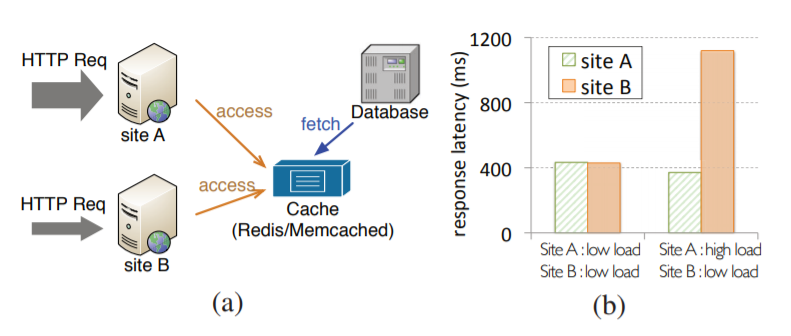


图2：（a）Web服务器的典型缓存设置。

（b）站点B由于不公平的高速缓存共享而遭受高延迟。

虽然该示例基于非真实的Web工作负载数据，但这个问题非常真实，正如技术论坛[6]上发布的许多问题所证明的那样，当使用Redis [9]或Memcached [7]时，如何跨多个站点实现资源隔离。 事实证明，两种流行的缓存系统都没有为性能隔离提供任何保证。 这包括来自主流云服务提供商的定制发行版，例如Amazon ElastiCache [1]和Microsoft Azure Redis Cache [3]。 正如我们将在第7节中所示，对于这样的缓存系统，策略用户很容易通过虚假访问文件来提高性能同时损害其他用户的性能（性能差距为2.9倍）

为了提供性能隔离，当今云缓存服务上下文中的通行方法是为每个用户或每个应用程序设置单独的缓存实例。这违背了整合原则并且成本很高。此外，高速缓存隔离将消除共享高速缓存文件的可能性，这使得隔离更加昂贵，因为要共享的文件比例越来越大。我们研究了一家来自互联网公司的生产HDFS日志，观察到31.4％的文件至少由两个用户/应用程序共享。与非共享文件相比，共享文件往往更频繁地被访问，例如，查看访问量最大的10％的文件，共享文件占访问量的53％。22％的用户至少有50％的文件被另一个用户访问。假设文件大小相等，如果我们为每个用户分配隔离的实例，我们需要至少31.4％的空间，并且由于工作集中共享文件的百分比更大，因此额外缓存的成本更高。