

**Original:** Documentation/vm/daemon/design.rst  
**翻译:** 司延腾 Yanteng Si <siyanteng@loongson.cn>  
**校译:**  
**设计**

## 可配置的层

DAMON提供了数据访问监控功能, 同时使其准确性和开销可控。基本的访问监控需要依赖于目标地址空间 并为之优化的基元。另一方面, 作为DAMON的核心, 准确性和开销的权衡机制是在纯逻辑空间中。DAMON 将这两部分分离在不同的层中, 并定义了它的接口, 以允许各种低层次的基元实现与核心逻辑的配置。

由于这种分离的设计和可配置的接口, 用户可以通过配置核心逻辑和适当的低级基元实现来扩展DAMON的 任何地址空间。如果没有提供合适的, 用户可以自己实现基元。

例如, 物理内存、虚拟内存、交换空间、那些特定的进程、NUMA节点、文件和支持的内存设备将被支持。另外, 如果某些架构或设备支持特殊的优化访问检查基元, 这些基元将很容易被配置。

## 特定地址空间基元的参考实现

基本访问监测的低级基元被定义为两部分。:

1. 确定地址空间的监测目标地址范围
2. 目标空间中特定地址范围的访问检查。

DAMON目前为物理和虚拟地址空间提供了基元的实现。下面两个小节描述了这些工作的方式。

### 基于VMA的目标地址范围构造

这仅仅是针对虚拟地址空间基元的实现。对于物理地址空间, 只是要求用户手动设置监控目标地址范围。

在进程的超级巨大的虚拟地址空间中, 只有小部分被映射到物理内存并被访问。因此, 跟踪未映射的地址区域只是一种浪费。然而, 由于DAMON可以使用自适应区域调整机制来处理一定程度的噪声, 所以严格来说, 跟踪每一个映射并不是必须的, 但在某些情况下甚至会产生很高的开销。也就是说, 监测目标 内部过于巨大的未映射区域应该被移除, 以不占用自适应机制的时间。

出于这个原因, 这个实现将复杂的映射转换为三个不同的区域, 覆盖地址空间的每个映射区域。这三个 区域之间的两个空隙是给定地址空间中两个最大的未映射区域。这两个最大的未映射区域是堆和最上面的mmap()区域之间的间隙, 以及在大多数情况下最下面的mmap()区域和堆之间的间隙。因为这些间隙 在通常的地址空间中是异常巨大的, 排除这些间隙就足以做出合理的权衡。下面详细说明了这一点:

```
<heap>
<BIG UNMAPPED REGION 1>
<uppermost mmap()-ed region>
(small mmap()-ed regions and munmap()-ed regions)
<lowermost mmap()-ed region>
<BIG UNMAPPED REGION 2>
<stack>
```

### 基于PTE访问位的访问检查

物理和虚拟地址空间的实现都使用PTE Accessed-bit进行基本访问检查。唯一的区别在于从地址中 找到相关的PTE访问位的方式。虚拟地址的实现是为该地址的目标任务查找页表, 而物理地址的实现则是查找与该地址有映射关系的每一个页表。通过这种方式, 实现者找到并清除下一个采样目标地址的位, 并检查该位是否在一个采样周期后再次设置。这可能会干扰其他使用访问位的内核子系统, 即空闲页跟踪和回收逻辑。为了避免这种干扰, DAMON使其与空闲页面跟踪相互排斥, 并使用 PG\_idle 和 PG\_young 页面标志来解决与回收逻辑的冲突, 就像空闲页面跟踪那样。

## 独立于地址空间的核心机制

下面四个部分分别描述了DAMON的核心机制和五个监测属性, 即 采样间隔、聚集间隔、区域更新间隔、最小区域数和最大区域数。

### 访问频率监测

DAMON的输出显示了在给定的时间内哪些页面的访问频率是多少。访问频率的分辨率是通过设置 采样间隔 和 聚集间隔 来控制的。详细地说, DAMON检查每个 采样间隔 对每个页面的访问, 并将结果汇总。换句话说, 计算每个页面的访问次数。在每个 聚合间隔 过去后, DAMON调用先前由用户注册的回调函数, 以便用户可以阅读聚合的结果, 然后再清除这些结果。这可以用以下简单的伪代码来描述:

```
while monitoring_on:
    for page in monitoring_target:
        if accessed(page):
            nr_accesses[page] += 1
    if time() % aggregation_interval == 0:
        for callback in user_registered_callbacks:
```

```
        callback(monitored_target, nr_accesses)
    for page in monitored_target:
        nr_accesses[page] = 0
    sleep(sampling_interval)
```

这种机制的监测开销将随着目标工作负载规模的增长而任意增加。

### 基于区域的抽样调查

为了避免开销的无限制增加, DAMON将假定具有相同访问频率的相邻页面归入一个区域。只要保持这个假设(一个区域内的页面具有相同的访问频率), 该区域内就只需要检查一个页面。因此, 对于每个采样间隔, DAMON在每个区域中随机挑选一个页面, 等待一个采样间隔, 检查该页面是否同时被访问, 如果被访问则增加该区域的访问频率。因此, 监测开销是可以通过设置区域的数量来控制的。DAMON允许用户设置最小和最大的区域数量来进行权衡。

然而, 如果假设没有得到保证, 这个方案就不能保持输出的质量。

### 适应性区域调整

即使最初的监测目标区域被很好地构建以满足假设(同一区域内的页面具有相似的访问频率), 数据访问模式也会被动态地改变。这将导致监测质量下降。为了尽可能地保持假设, DAMON根据每个区域的访问频率自适应地进行合并和拆分。

对于每个聚集区间, 它比较相邻区域的访问频率, 如果频率差异较小, 就合并这些区域。然后, 在它报告并清除每个区域的聚合接入频率后, 如果区域总数不超过用户指定的最大区域数, 它将每个区域拆分为两个或三个区域。

通过这种方式, DAMON提供了其最佳的质量和最小的开销, 同时保持了用户为其权衡设定的界限。

### 动态目标空间更新处理

监测目标地址范围可以动态改变。例如, 虚拟内存可以动态地被映射和解映射。物理内存可以被热插拔。

由于在某些情况下变化可能相当频繁, DAMON检查动态内存映射的变化, 并仅在用户指定的时间间隔(区域更新间隔)内将其应用于抽象的目标区域。