OpenEuler

DAMON特性介绍

目录

CONTENT

OpenEuler

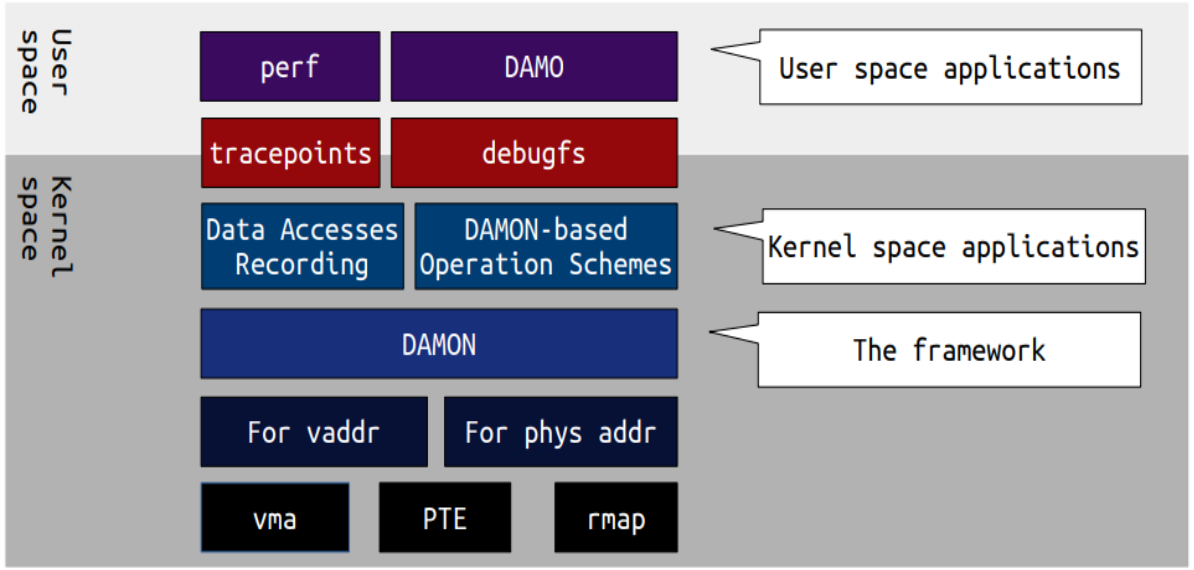
特性简介

1. 简介

DAMON是一个内核模块，允许用户在线轻量级地监控一个特定进程空间的实际内存访问情况。其目标是：

- 数据准确：保证获取的数据是正确的
- 足够轻量：不能带来太多额外的工作开销
- 通用性：能够监控多个进程，虚拟地址和物理地址等多场景

2. 总体架构



工作机制

1. Access frequency monitor - 访问频率监控

- 采样间隔：采样间隔内统计每页的访问次数。
- 聚合间隔：每个聚合间隔后，调用定义的回调函数读取聚合的结果。

2. Region Based Sampling – 基于region的频率采样

- 将具有相同访问频率的相邻页分组到一个region，每个region中随机选取一个页面，检查该页面访问情况。
- 因此，通过设置region个数，可以控制monitor的开销情况，用户可以设置最小和最大region数。

3. Adaptive Regions Adjustment – 自适应region调整

- 为了保持相同region内的页面具有相似的访问频率，DAMON要自动的合并和切割region：每个聚合间隔，比较相邻region的访问频率，如果频率差较小，则进行合并。
- 如果总region数不超过用户配置的最大region数，则将每个region拆分为2~3个region。

工作机制

4. Dynamic Target Space Updates Handling – 动态更改监控区域

- Monitor的目标地址范围可以动态更改。在某些情况下更改可能非常频繁，DAMON会在每个region更新时间间隔内对于监控region进行更新检查。

5. 提供参数：

monitor 属性	含义	用法
Sampling interval	采样间隔	检查每个页面的访问
Aggregation interval	聚合间隔	统计访问每个页面的次数
Regions update interval	region 更新间隔	region 的自适应合并和切割
Minimum number of regions	最小 region 数	监控的最小 region 数
maximum number of regions	最大 region 数	监控的最大 region 数

OpenEuler

功能接口

1. 用户态工具 DAMO[1]

提供用户态工具DAMO,可以进行数据访问模式的记录，可视化分析以及具体的内存管理和优化的策略制定。

- Record – 记录内存访问数据

a) 监控具体的程序：**./damo record "./masim "**

b) 监控具体的进程i：**./damo record $pid**

c) 监控特定关键字paddr(默认监控“/proc/iomem”文件中指定的最大“system RAM”区域)：**./damo record paddr**

```
kdamond.0 31216 [000] 16418.095816: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=17 140727093895168-140728437059584: 0
kdamond.0 31216 [000] 16418.095816: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=17 140728437059584-140729886818304: 0
kdamond.0 31216 [000] 16418.095817: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=17 140729886818304-140731549777920: 0
kdamond.0 31216 [000] 16418.095817: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=17 140731549777920-140732701061120: 0
kdamond.0 31216 [000] 16418.095817: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=17 140732701061120-140734641201152: 0
kdamond.0 31216 [000] 16418.095817: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=17 140734641201152-140734833086464: 0
kdamond.0 31216 [001] 16418.201908: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=19 4194304-12091392: 0
kdamond.0 31216 [001] 16418.201910: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=19 12091392-23941120: 1
kdamond.0 31216 [001] 16418.201910: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=19 23941120-27889664: 5
kdamond.0 31216 [001] 16418.201911: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=19 27889664-89776128: 1
kdamond.0 31216 [001] 16418.201911: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=19 89776128-95305728: 2
kdamond.0 31216 [001] 16418.201911: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=19 95305728-103600128: 1
kdamond.0 31216 [001] 16418.201911: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=19 103600128-135860224: 0
kdamond.0 31216 [001] 16418.201912: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=19 1866694344704-1867768086528: 0
kdamond.0 31216 [001] 16418.201912: damon:damon_aggregated: target_id=18446617476134678656 nr_regions=19 140715645054976-140717639958528: 0
```

[1]https://github.com/awslabs/damo

功能接口

Heat形式：时间，访问地址，地址的访问

1. 用户态工具 DAMO[1]

提供用户态工具DAMO,可以进行数据访问模式的记录，可视化分析以及具体的内存管理和优化的策略制定。

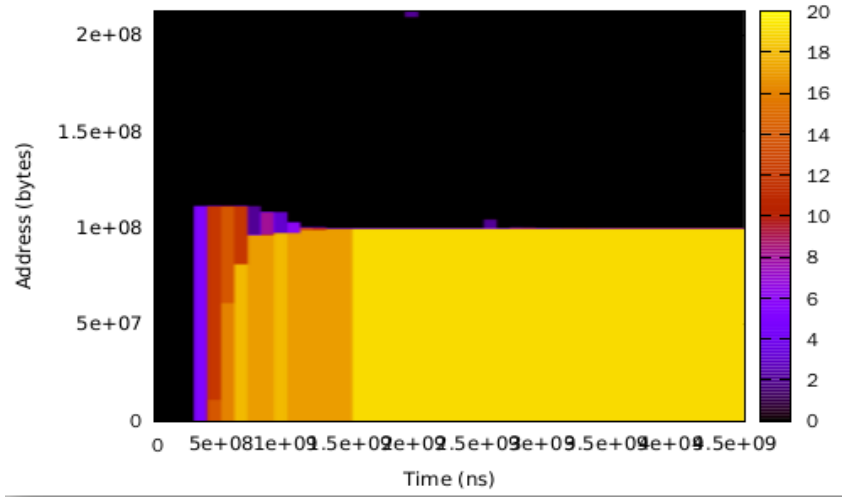
- Report – 可视化分析内存访问数据

Raw 文本格式

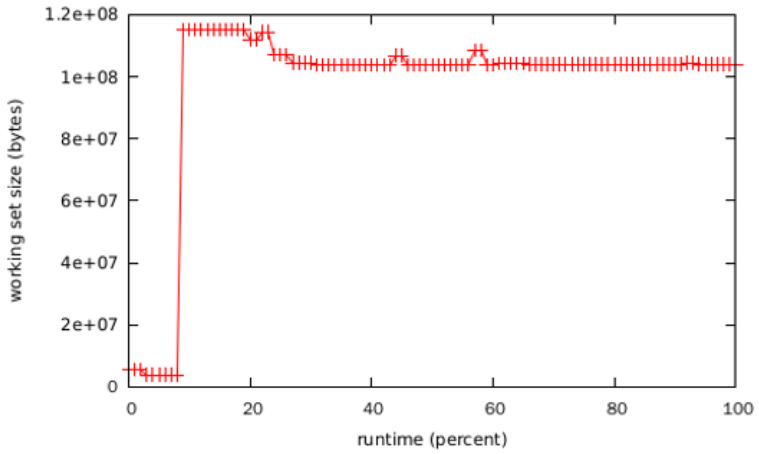
```
7f7eb8b76000-7f7eba947000(  29.816 MiB):        0
7f7eba947000-7f7ebc581000(  28.227 MiB):        0
7f7ebc581000-7f7ebe48d000(  31.047 MiB):        0
7f7ebe48d000-7f7ebf68d000(  18.000 MiB):        0
7ffcec6f2000-7ffcec7f1000(1020.000 KiB):        0

monitoring_start:              3.699 s
monitoring_end:                3.802 s
monitoring_duration:         103.422 ms
target_id: 18446617473488935168
nr_regions: 95
000000400000-000000402000(    8.000 KiB):        7
000000402000-000000403000(    4.000 KiB):        0
000000403000-000000405000(    8.000 KiB):        11
000000406000-000000407000(    4.000 KiB):        0
000000407000-000000408000(    4.000 KiB):        19
000000408000-000000411000(   36.000 KiB):        9
000000411000-000000417000(   24.000 KiB):        13
000000417000-000000438000(  132.000 KiB):        10
000000438000-000000453000(  108.000 KiB):        7
000000453000-000000465000(   72.000 KiB):        9
000000465000-000000469000(   16.000 KiB):        10
000000469000-00000046b000(    8.000 KiB):        8
00000046b000-000000481000(   88.000 KiB):        10
```

Heat形式：时间，访问地址，地址的访问频率



wss形式：随着时间变化，进程运行过程中访问内存区域的大小变化



[1]https://github.com/awslabs/damo

OpenEuler

功能接口

1. 用户态工具 DAMO[1]

提供用户态工具DAMO,可以进行数据访问模式的记录，可视化分析以及具体的内存管理和优化的策略制定。

- Schemes： scheme子命令能够在进程运行时对内存空间进行分配优化，scheme 格式如下
  - **<min-size> <max-size> <min-acc> <min-age> <max-age> <action>**

示例 1：进程运行的某个区域存在不小于 100ms 的较高访问频率，则将该区域存放在 LRU 列表的头部。

min max 80 max 100ms max willneed

min-size/max-size	进程运行中被监控区域的最小/最大的内存空间；默认值用 min/max 表示； 单位表示形式为 B/K/M/G/
min-acc/max-acc	被监控区域的最小/最大访问频率；默认值用 min/max 表示； 单位表示形式为 us/ms/s/m/h/d
min-age/max-age	被监控区域的最小/最大访问时间，默认值用 min/max 表示
action	scheme 可以进行的操作，一共五种 willneed, cold, pageout, hugepage, nohugepage，与对应系统调用 madvise() 的 flag 值

Action	对应系统调用
willneed	madv_willneed
cold	madv_cold
pageout	madv_pageout
hugepage	madv_hugepage
nohugepage	madv_nohugepage
stat	return 0

[1]https://github.com/awslabs/damo

OpenEuler

功能接口

2. Debugfs挂载

- Attrs:设置进程的sample interval（采样时间间隔），aggregation interval（聚合时间间隔），regions update interval（区域更新时间间隔）以及目标监控区域数目的最大最小值。

  echo 5000 100000 1000000 10 1000 > /sys/kernel/debug/damon/attrs

- Monitor_on: monitor开关

  echo on > /sys/kernel/debug/damon/monitor_on

- Target_ids: 指定监控进程号，支持多进程

  echo 5043 43 > /sys/kernel/debug/damon/target_ids

- Record: 记录访问数据指定监控进程号，支持多进程

  echo “4096 /damon.data” > /sys/kernel/debug/damon/record

- Schemes: 设置内存优化策略（格式见前面功能接口 DAMO Schemes）

  echo “4096 8192 0 5 10 20 2” > /sys/kernel/debug/damon/schemes

3. 内核接口开发

- 内核编程开发

应用场景举例

冷页换出：查找特定时间内没有被访问的内存区域，进行pageout

- 场景：Zram/zswapd， ssd写入会消耗大量的cpu去做频繁的page out操作，对于这种场景，可以配置对应的时间或者区域大小的上限，damon_reclaim会优先page out那些长时间未访问的内存区域。
- 测试结果：在ZRAM 5.15-rc5 linux内核上，配上50ms/s的时间上限，damon_reclaim模块节约了38.58%的内存，运行时开销只有1.94%，单CPU时间消耗了4.97%.

a) 测试环境：qemu/kvm 环境，i3.metal AWS instance, has 130GiB memory, and runs a linux kernel built on latest -mm tree[1] plus this patchset., utilizes a 4 GiB ZRAM swap device.

b) 测试数据：PARSEC3 and SPLASH-2X benchmark suites中24个realistic workloads 的系统内存占用空间，page fault的次数和runtime，重复5 次测试取平均值。

parsec3典型测试集

splash2x/ocean_ncp	orig	thp	thp overhead	ethp	ethp overhead
Runtime(s)	188.626	95.922	-49.15%	146.884	-22.13%
MemTotal – MemFree (kB)	4538442	7720563	70.11%	4558585	0.44%

说明：
orig: /sys/kernel/mm/transparent_hugepage/enabled设置为madvise，关闭DAMON策略
thp: /sys/kernel/mm/transparent_hugepage/enabled设置为always
ethp: /sys/kernel/mm/transparent_hugepage/enabled设置为madvise，开启DAMON策略，attrs为默认配置，schemes配置为0 4294967295 5 4294967295 0 4294967295 3 2097152 4294967295 0 0 7 4294967295 4。
含义为对于访问频率大于5的区域，执行MADV_HUGEPAGE，对于大于2M且在7s内没有被访问的区域，执行MADV_NOHUGEPAGE。

结论：
一直使用大页性能提升49.15%，但内存占用增加70.11%，使用DAMON策略，性能可以提升22.13%，且内存占用没有明显增加。

OpenEuler

mysql + tpcc性能提升场景测试

指标	warehouse	runMins	tpmTotal	TPS (相对orig提升比)	mysql_VmRSS (kB)	MemTotal-MemFree(kB)	系统内存占用（相对orig提升比）	配置信息
orig	1000	10	99590.25		11843410	24276163		
thp	1000	10	112013.94	12.47%	11980477	25985577	7.04%	
ethp	1000	10	109795.64	10.25%	12122212	24593106	1.31%	min-size:0 max-size:4294967295 min-acc:1 max-acc:4294967295 min-age:0 max-age:4294967295 action:3(hugepage) min-size:2097152(2M) max-size:4294967295 min-acc:0 max-acc:0 min-age:40 max-age:4294967295 action:4(nohugepage)
orig	1000	10	119825.27		32047942	43603310		
thp	1000	10	129710.42	8.25%	32445349	46656401	7.00%	
ethp	1000	10	128048.58	6.86%	32314835	43640292	0.08%	min-size:0 max-size:4294967295 min-acc:1 max-acc:4294967295 min-age:0 max-age:4294967295 action:3(hugepage) min-size:2097152(2M) max-size:4294967295 min-acc:0 max-acc:0 min-age:10 max-age:4294967295 action:4(nohugepage)

结论：

mysql buffer大小为10G场景下，使用DAMON特性性能提升可达到10.25%，且内存占用仅增加1.31%；

mysql buffer大小为30G场景下，使用DAMON特性性能提升可达到6.86%，且内存占用仅增加0.08%。

OpenEuler

mysql + tpcc节省内存场景测试

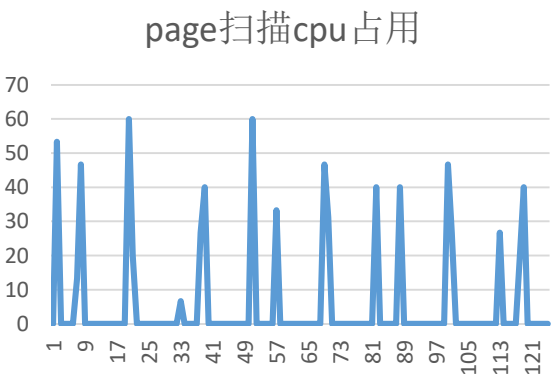
指标	warehouse	runMins	tpmTotal	TPS (有无memig下降比)	mysql_VmRSS (kB)	mysql_VmSwap (KB)	VmRSS (下降比)	配置信息
NOMIGRATE	1000	10	131077.09		40088991	0		
WITHMIGRATE	1000	10	108184.43	17%	19211571	20055222	52%	sample_interval:5000 aggr_interval:100000 regions_update_interval:1000000 migrate_interval:80000000 min_nr_regions:100 max_nr_regions:1000 migrate_thres:1

结论：
使用DAMON扫描策略对冷页执行swap操作，可以节省52%内存，同时性能下降17%以内。

OpenEuler

mysql + tpcc节省内存场景底噪对比

指标	warehouse	runMins	tpmTotal	TPS (有无memig下降比)	memigd/kdamond_%CPU	mysql_VmRSS (kB)	mysql_VmSwap (KB)	VmRSS (下降比)	配置信息
基线	1000	20	195095.8		0	39250948	0		
region scan + migrate	1000	20	183988.45	6%	65.008	30869304	3420964	21%	sample_interval:1000000 aggr_interval:30000000 regions_update_interval:60000000 migrate_interval:1000000 min_nr_regions:100 max_nr_regions:1000 migrate_thres:2
page scan + migrate	1000	20	180560.97	7%	65.096	30535973	8763291	22%	loop:1 interval:1 sleep:1 T:2
region scan	1000	20	195591.08	-	0	39503213	0	-	sample_interval:1000000 aggr_interval:30000000 regions_update_interval:60000000 min_nr_regions:100 max_nr_regions:1000
page scan	1000	20	185196.83	-	5.4144	39791099	0	-	loop:1 interval:1 sleep:1



结论：
使用DAMON扫描策略相比于每个page扫描统计，在节省相同内存，且性能下降差不多的情况下，使用DAMON扫描可以极大降低CPU占用率，避免cpu冲高的情况。

OpenEuler

DAMON特性与etmem结合

etmem实现内存分级扩展技术，通过扫描页面，对页面进行分级，将分级后的内存冷数据从内存介质迁移到高性能存储介质中，达到内存容量扩展的目的，从而实现内存成本下降。

1、etmem当前仅支持每个页面扫描的方式，可以结合DAMON特性，同时支持region扫描的方式
2、etmem当前仅支持识别出的冷数据迁移出内存，可以结合DAMON特性，支持热数据使用透明大页，以及当透明大页热数据变为冷数据时，取消使用透明大页，优化透明大页的分配
3、etmem当前的分级策略是设置一个水线值，访问次数小于水线值识别为冷页，结合DAMON特性，可以同时设置水线值以及访问时间，对内存分级。

Thank you

OpenEuler