DAMON-based Reclamation

DAMON-based Reclamation (DAMON_RECLAIM) is a static kernel module that aimed to be used for proactive and lightweight reclamation under light memory pressure. It doesn't aim to replace the LRU-list based page_granularity reclamation, but to be selectively used for different level of memory pressure and requirements.

Where Proactive Reclamation is Required?

On general memory over-committed systems, proactively reclaiming cold pages helps saving memory and reducing latency spikes that incurred by the direct reclaim of the process or CPU consumption of kswapd, while incurring only minimal performance degradation [1] [2] .

Free Pages Reporting [3] based memory over-commit virtualization systems are good example of the cases. In such systems, the guest VMs reports their free memory to host, and the host reallocates the reported memory to other guests. As a result, the memory of the systems are fully utilized. However, the guests could be not so memory-frugal, mainly because some kernel subsystems and user-space applications are designed to use as much memory as available. Then, guests could report only small amount of memory as free to host, results in memory utilization drop of the systems. Running the proactive reclamation in guests could mitigate this problem.

How It Works?

DAMON_RECLAIM finds memory regions that didn't accessed for specific time duration and page out. To avoid it consuming too much CPU for the paging out operation, a speed limit can be configured. Under the speed limit, it pages out memory regions that didn't accessed longer time first. System administrators can also configure under what situation this scheme should automatically activated and deactivated with three memory pressure watermarks.

Interface: Module Parameters

To use this feature, you should first ensure your system is running on a kernel that is built with `CONFIG_DAMON_RECLAIM=y`.

To let sysadmins enable or disable it and tune for the given system, DAMON_RECLAIM utilizes module parameters. That is, you can put `damon_reclaim.<parameter>=<value>` on the kernel boot command line or write proper values to `/sys/modules/damon_reclaim/parameters/<parameter>` files.

Note that the parameter values except `enabled` are applied only when DAMON_RECLAIM starts. Therefore, if you want to apply new parameter values in runtime and DAMON_RECLAIM is already enabled, you should disable and re-enable it via `enabled` parameter file. Writing of the new values to proper parameter values should be done before the re-enablement.

Below are the description of each parameter.

enabled

Enable or disable DAMON_RECLAIM.

You can enable DAMON_RCLAIM by setting the value of this parameter as `Y`. Setting it as `N` disables DAMON_RECLAIM. Note that DAMON_RECLAIM could do no real monitoring and reclamation due to the watermarks-based activation condition. Refer to below descriptions for the watermarks parameter for this.

min_age

Time threshold for cold memory regions identification in microseconds.

If a memory region is not accessed for this or longer time, DAMON_RECLAIM identifies the region as cold, and reclaims it.

120 seconds by default.

quota_ms

Limit of time for the reclamation in milliseconds.

DAMON_RECLAIM tries to use only up to this time within a time window (quota_reset_interval_ms) for trying reclamation of cold pages. This can be used for limiting CPU consumption of DAMON_RECLAIM. If the value is zero, the limit is disabled.

10 ms by default.

quota_sz

Limit of size of memory for the reclamation in bytes.

DAMON_RECLAIM charges amount of memory which it tried to reclaim within a time window (quota_reset_interval_ms) and makes no more than this limit is tried. This can be used for limiting consumption of CPU and IO. If this value is zero, the limit is disabled.

128 MiB by default.

quota_reset_interval_ms

The time/size quota charge reset interval in milliseconds.

The charget reset interval for the quota of time (quota_ms) and size (quota_sz). That is, DAMON_RECLAIM does not try reclamation for more than quota_ms milliseconds or quota_sz bytes within quota_reset_interval_ms milliseconds.

1 second by default.

wmarks_interval

Minimal time to wait before checking the watermarks, when DAMON_RECLAIM is enabled but inactive due to its watermarks rule.

wmarks_high

Free memory rate (per thousand) for the high watermark.

If free memory of the system in bytes per thousand bytes is higher than this, DAMON_RECLAIM becomes inactive, so it does nothing but only periodically checks the watermarks.

wmarks_mid

Free memory rate (per thousand) for the middle watermark.

If free memory of the system in bytes per thousand bytes is between this and the low watermark, DAMON_RECLAIM becomes active, so starts the monitoring and the reclaiming.

wmarks_low

Free memory rate (per thousand) for the low watermark.

If free memory of the system in bytes per thousand bytes is lower than this, DAMON_RECLAIM becomes inactive, so it does nothing but periodically checks the watermarks. In the case, the system falls back to the LRU-list based page granularity reclamation logic.

sample_interval

Sampling interval for the monitoring in microseconds.

The sampling interval of DAMON for the cold memory monitoring. Please refer to the DAMON documentation (:doc:`usage`) for more detail.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\mm\damon\[linux-master][Documentation][admin-guide][mm][damon]reclaim.rst, line 154); *backlink***
>
> Unknown interpreted text role "doc".

aggr_interval

Aggregation interval for the monitoring in microseconds.

The aggregation interval of DAMON for the cold memory monitoring. Please refer to the DAMON documentation (:doc:`usage`) for more detail.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\mm\damon\[linux-master][Documentation][admin-guide][mm][damon]reclaim.rst, line 162); *backlink***
>
> Unknown interpreted text role "doc".

min_nr_regions

Minimum number of monitoring regions.

The minimal number of monitoring regions of DAMON for the cold memory monitoring. This can be used to set lower-bound of the monitoring quality. But, setting this too high could result in increased monitoring overhead. Please refer to the DAMON documentation (:doc:`usage`) for more detail.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-**

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\mm\damon\[linux-master][Documentation][admin-guide][mm][damon]reclaim.rst, line 170); *backlink*

Unknown interpreted text role "doc".

max_nr_regions

Maximum number of monitoring regions.

The maximum number of monitoring regions of DAMON for the cold memory monitoring. This can be used to set upper-bound of the monitoring overhead. However, setting this too low could result in bad monitoring quality. Please refer to the DAMON documentation (:doc:`usage`) for more detail.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\mm\damon\[linux-master][Documentation][admin-guide][mm][damon]reclaim.rst, line 180); *backlink*

Unknown interpreted text role "doc".

monitor_region_start

Start of target memory region in physical address.

The start physical address of memory region that DAMON_RECLAIM will do work against. That is, DAMON_RECLAIM will find cold memory regions in this region and reclaims. By default, biggest System RAM is used as the region.

monitor_region_end

End of target memory region in physical address.

The end physical address of memory region that DAMON_RECLAIM will do work against. That is, DAMON_RECLAIM will find cold memory regions in this region and reclaims. By default, biggest System RAM is used as the region.

kdamond_pid

PID of the DAMON thread.

If DAMON_RECLAIM is enabled, this becomes the PID of the worker thread. Else, -1.

nr_reclaim_tried_regions

Number of memory regions that tried to be reclaimed by DAMON_RECLAIM.

bytes_reclaim_tried_regions

Total bytes of memory regions that tried to be reclaimed by DAMON_RECLAIM.

nr_reclaimed_regions

Number of memory regions that successfully be reclaimed by DAMON_RECLAIM.

bytes_reclaimed_regions

Total bytes of memory regions that successfully be reclaimed by DAMON_RECLAIM.

nr_quota_exceeds

Number of times that the time/space quota limits have exceeded.

Example

Below runtime example commands make DAMON_RECLAIM to find memory regions that not accessed for 30 seconds or more and pages out. The reclamation is limited to be done only up to 1 GiB per second to avoid DAMON_RECLAIM consuming too much CPU time for the paging out operation. It also asks DAMON_RECLAIM to do nothing if the system's free memory rate is more than 50%, but start the real works if it becomes lower than 40%. If DAMON_RECLAIM doesn't make progress and therefore the free memory rate becomes lower than 20%, it asks DAMON_RECLAIM to do nothing again, so that we can fall back to the LRU-list based page granularity reclamation.

```
# cd /sys/modules/damon_reclaim/parameters
# echo 30000000 > min_age
# echo $((1 * 1024 * 1024 * 1024)) > quota_sz
# echo 1000 > quota_reset_interval_ms
# echo 500 > wmarks_high
```

```
# echo 400 > wmarks_mid
# echo 200 > wmarks_low
# echo Y > enabled
```


[1] https://research.google/pubs/pub48551/

[2] https://lwn.net/Articles/787611/

[3] https://www.kernel.org/doc/html/latest/vm/free_page_reporting.html