

DAMON: Current Status and Future Plans

SeongJae Park <sj@kernel.org>

https://damonitor.github.io

Notices

• The views expressed herein are those of the speaker; they do not reflect the views of his employers

I, SeongJae Park (SJ)

- Working on AWS
- Maintaining Linux kernel DAMON subsystem and its user-space tool, DAMO



Overview

- DAMON in a Nutshell (10 mins)
- Updates Since Kernel Summit 2022 (10 mins)
- Future Plans (15 mins)
- DAMON Community (5 mins)
- Conclusion and Remaining QnA

• Each time-specified topic will get its own QnA

DAMON in a Nutshell

Why? Increasing Demands/Costs and No-free-lunch H/W Solution

- Memory demands increase faster than the price is decreasing
- New H/W will arrive, but as a new hierarchy rather than a perfect drop-in replacement
 - No free lunch
- Need access-aware system operation, but how can we be access-aware?



Figure 3: Memory as a percentage of rack TCO and power Figure 2: Latency characteristics of memory technologies. across different hardware generations of Meta.

Retrieved from "TPP: Transparent Page Placement for CXL-Enabled Tiered-Memory", https://dl.acm.org/doi/pdf/10.1145/3582016.3582063

How Data Access Would Look Like, Over Time

• Accesses are made sometime, somewhere



How Data Access Would Look Like, Over Time

• Accesses are made sometime, somewhere



How Data Access Would Look Like, Over Time

• Accesses are made sometime, somewhere



Ideal Data Access MONitor

- Capture all access
- Space granularity: bit (or, electron?)
- Time granularity: 1 sec / CPU freq / # CPUs (or, speed of light?)
- Record from: From the boot (or, since Unix timestamp 0 (1970-01-01)?)



Fixed Granularity Monitoring

- Let user define the time/space granularity ('nr_min_regions' and 'sample_interval')
 - 10 and 5ms by default for the two parameters
- Check access to only one page per region
 - Pages in each region is assumed to have similar access frequencies
 - 'nr_min_regions' could be "monitoring target address space size / PAGE_SZ"



Sampling Results Aggregation

- Introduce new user-specifiable time interval, "aggregate interval" (100ms by default)
 - Count number of access-detected sampling intervals per aggregate interval ("nr_accesses")
- Amount of the record is reduced
 - A bool per sampling interval \rightarrow One counter per aggregate interval
 - Create snapshot per aggregation interval



Sampling Results Aggregation

- Introduce new user-specifiable time interval, "aggregate interval" (100ms by default)
 - Count number of access-detected sampling intervals per aggregate interval ("nr_accesses")
- Amount of the record is reduced
 - A bool per sampling interval \rightarrow One counter per aggregate interval
 - Create snapshot per aggregation interval



Merging Regions

• Definition of region: address range that having similar access frequencies to pages in it



Merging Regions

- Definition of region: address range that having similar access frequencies to pages in it
- Merge adjacent regions of similar access frequency, at the end of the aggregation interval



Split Regions

• Access pattern may change over time



Split Regions

- Access pattern may change over time
- Randomly split regions at the beginning of aggregation interval
- Some would be merged at the end of the aggregation interval



Continuous Merge/Split: Adaptive Regions Adjustment

- Split reverts unnecessary merge, vice versa
- One page per region sampling still reasonable
- Users can set 'nr_{min,max}_regions'
 - DAMON stops merge/split if the range can be violated
- Users can control accuracy and overhead

Age Counting

- Age: Number of last aggregation intervals that similar nr_accesses were kept
- Snapshot contains some history



DAMON: Access Monitoring Results Snapshot Generator

- Answer to "Which memory region is how frequently accessed for how long time?"
 - With controllable overhead and accuracy

	addr [127.601 TiB , 127.601 TiB) (7.715 MiB) access 0 % age 59.600 s
	000000000000000000
	000000000000000000000000000000000000000
	000000000000000000000000000000000000000
	000000000000000000000000000000000000000
	000000000000000000000000000000000000000
	addr [127.601 TiB , 127.990 TiB) (398.565 GiB) access 0 % age 200 ms
	000000000000000000000000000000000000000
	000000000000000000000000000000000000000
8	addr [127.990 TiB , 127.990 TiB) (1.879 MiB) access 0 % age 47.500 s
9	addr [16.000 EiB , 16.000 EiB) (4.000 KiB) access 0 % age 200 ms
	144444444444444444
	addr [85.967 TiB , 85.967 TiB) (8.000 KiB) access 50 % age 600 ms
	addr [85.967 TiB ,85.967 TiB)(8.000 KiB)access 60 % age 0 ns
	9999999999999999999999999999
	addr [85.967 TiB , 85.967 TiB) (4.000 KiB) access 100 % age 17.200 s
	999999999999999999999999999
	9999999999999999999999999
	addr [85.967 TiB , 85.967 TiB) (9.543 MiB) access 100 % age 2 s
	length: age (represents [0 ns, 1 m 15.500 s] with [1, 30] columns in logscale)
	color: access_rate (represents [0 %, 100 %] with [0, 9] number and color (0 256789) in linearscale)
	neight: size (represents [4.000 KiB, 398.565 GiB] with [1, 5] rows in logscale)

DAMON: Access Monitoring Results Snapshot Generator

- Answer to "Which memory region is how frequently accessed for how long time?"
 - With controllable overhead and accuracy
- Wait, isn't this information enough to make kernel just works?



DAMOS: DAMON-based Operation Schemes

- Find regions of interesting access pattern from the snapshot and apply a requested action
 - "Page out pages of regions that not accessed for >= 2 mins
 - "Use THP for pages of regions that having >= 10% access rate for >= 1 minute"
- Multiple requests (called schemes) can be made

damo start --damos_action pageout --damos_access_rate 0% 0% --damos_age 2m max
damo start --damos_action thp --damos_access_rate 10% max --damos_age 1m max

DAMOS: Target Access Pattern and Action

- Basic ways to specify the request
- Target access pattern
 - Ranges of size, access rate, and age of the region of the interest
- Action
 - System action that DAMOS will make to the regions of the pattern
 - pageout, thp, nothp, ...

DAMOS Quota

- DAMOS target access pattern is hard to tune
 - min/max for 3 ranges = 6 parameters
 - Optimum tuning depends on the characteristics of the system and workloads
- Quota: user-specifiable maximum resource that DAMOS can use for applying the action
 - e.g., Apply the action to only up to 100 MiB of regions per second
- Under the limit, DAMOS prioritizes regions based on access pattern, following the context
 - If the action is pageout, colder pages are prioritized
 - If the action is thp, hotter pages are prioritized

QnA for DAMON/DAMOS Basics

- Sampling-based access frequency monitoring
- Adaptive regions adjustment
- DAMOS, w/ access pattern
- DAMOS quota

DAMON Updates Since Kernel Summit 2022

Overview

- DAMON in a Nutshell (10 mins)
- Updates Since Kernel Summit 2022 (10 mins)
- Future Plans (15 mins)
- DAMON Community (5 mins)
- Conclusion and Remaining QnA

• Each time-specified topic will get its own QnA

DAMOS Tried Regions

- Expose DAMOS-found target regions
- Expected Usages
 - Debugging and tuning DAMOS schemes and/or DAMOS itself
 - Query-like efficient monitoring results collecting
- Provide kernel API, tracepoint, and sysfs interfaces
- Merged in v6.2
- https://lore.kernel.org/damon/20221101220328.95765-1-sj@kernel.org/

DAMOS Filters

- Non-access pattern information based DAMOS target filtering
 - Type of backing content of the page (file or anon)
 - Belonging memory cgroups
 - Address range
 - Belonging process
- E.g., "Apply this DAMOS scheme to anon pages of these cgroup, if it's in the address range of NUMA node X, but exclude those of these processes"
- Merged in v6.3, later expanded supporting types in v6.6
 - https://lore.kernel.org/damon/20221205230830.144349-1-sj@kernel.org/
 - https://lore.kernel.org/damon/20230802214312.110532-1-sj@kernel.org/

Psuedo Moving-Average Access Rate-based Snapshot Generation

- DAMON snapshot is prepared per aggregation interval (100 ms by default)
- Problematic with long aggregation interval (e.g., 20 secs)
 - Long aggregation interval for high accuracy and/or lower overhead
- Generate a snapshot with pseudo moving average access rate per sampling interval
- Merged in v6.7-rc1
- https://lore.kernel.org/damon/20230915025251.72816-1-sj@kernel.org/

moving_sum(n) = moving_sum(n - 1) - last_non_moving_sum / len_window + new_value

https://lore.kernel.org/damon/20230915025251.72816-4-sj@kernel.org/

DAMOS Apply Interval

- DAMOS applies the action to regions every aggregation interval
 - Since the snapshot is complete only at that time
- Psuedo-moving access rate allows them be independent
- Use a dedicated time interval for DAMOS
- Merged in v6.7-rc1
- https://lore.kernel.org/damon/20230916020945.47296-1-sj@kernel.org/

DAMO (Data Access Monitoring Operator) v2

- DAMO is a DAMON user-space tool
- Available on many distros
- Initially designed for static offline monitoring usage
- re:Designed to support online DAMON usages
- Released just before OSSummit EU 2023 (Sep 2023)

QNA for DAMON Updates since KernelSummit 2022

- DAMOS Tried Regions
- DAMOS Filters
- Pseudo moving average access rate based snapshot generation
- DAMOS apply interval
- DAMO v2

Overview

- DAMON in a Nutshell (10 mins)
- Updates Since Kernel Summit 2022 (10 mins)
- Future Plans (15 mins)
- DAMON Community (5 mins)
- Conclusion and Remaining QnA

• Each time-specified topic will get its own QnA

DAMON Future Plans

Aim-oriented Feedback-driven DAMOS Aggressiveness Auto-tuning

https://lore.kernel.org/damon/20231112194607.61399-1-sj@kernel.org/

DAMOS Tuning Difficulty

- Quota reduces DAMOS tuning complexity by removing number of knobs (6 to 1)
- Optimum quota value still depends on systems and workloads
- Especially difficult for balancing two conflicting schemes
 - Number of knobs still increase with multiple schemes

Aim-oriented Feedback-driven Aggressiveness Auto Tuning

- DAMOS quota is good for controlling aggressiveness
 - The prioritization mechanism provides a best effort
- Idea: Allow users feed and tame DAMOS
 - Ask what users want from DAMOS, instead of how DAMOS should work
 - Easier to know for users who don't know DAMOS
 - Separate the policy and the mechanism
 - DAMOS somehow make it; Users provide feedback
- Implementation: A simple feedback loop algorithm

f(n) = max(f(n - 1) * ((target_score - current_score) / target_score + 1), 1)

Progress and Test Results

- First idea was shared on ksummit 2022; First RFC patchset has sent for ksummit 2023
 - Presentation-driven development works
- Proactive reclamation aiming last 10 secs 0.5% memory pressure stall has tested
 - Memory saving and performance overhead similar to an "offline tuned" ones (DAMOOS)
 - Aggressiveness auto-tuning achieves best PSI saving among all



Not-tuned Offline-tuned Online-tuned

QnA for DAMOS Aggressiveness Auto-tuning

- Tuning difficulty
- Aim-oriented feedback-driven aggressiveness auto-tuning
- Test results

Access/Contiguity-aware Memory Auto-scaling (ACMA)

https://lore.kernel.org/damon/20231112195114.61474-1-sj@kernel.org/

Collaborative Memory Over-subscribed VM systems

- Guest voluntarily reports pages that the host can reuse
 - Free pages reporting
- The host detects guests' access to reported pages (page fault) and allocate new one



Guest Requirements

- Being memory frugal without performance impact
 - To allow higher over-subscription ratio
- Report time free pages contiguity
 - To minimize reporting overhead
- Reported pages contiguity
 - If the host uses large page size, to avoid returning whole host-page (large) for single guest-page (small) fault
- Minimizing metadata for reported pages
 - To maximize the over-subscription

Possible Solutions and Challenges

- Being memory frugal: DAMON-based proactive reclamation
- Report-time contiguity: Proactive compaction
 - Compaction could fail due to isolation/migration failures
 - More-than-required granularity compaction waste resource
- Post-report contiguity: We found no good solution
- Minimizing metadata for reported pages: Memory hot-remove
 - Memory-block granularity isolation/migration is slow and fails frequently
- Orchestrating multiple kernel features that not designed together from user space
 - Complex and inefficient

ACMA: Access/Contiguity-aware Memory Autoscaling

- A new kernel feature designed for the requirements
- Aims
 - Provide better solutions for each problem if possible
 - Efficiently orchestrate the solutions
 - Provide easy-to-use user interface (kernel that just works)

ACMA: New Metric and Operation

- New metric: DAMON-working set
 - Memory regions that DAMON has shown access to, within a user-specifiable time threshold
- New operation: Stealing
 - Migrate pages in given physical address range out and take the pages of the range
 - Do nothing with the pages but report those pages to the host as free to use
 - If an entire memory block is stolen, hot-remove the block, free metadata, report the freed pages
 - Maybe similar to virtio-mem's memory reduction operation

ACMA: Workflow

- If DAMON-working set to free memory ratio is higher than a threshold (high, e.g., 200%)
 - Steal report-gran-contiguous regions from last available memory block, colder regions first
- If the ratio is becomes lower than a threshold (middle, e.g., 100%)
 - Stop stealing
 - Run DAMON-based proactive reclamation, until the ratio reaches the threshold (middle)
- If the ratio is lower than yet another threshold (low, e.g., 50%)
 - Start returning stolen pages, stolen pages closer to not-yet-stolen memory block first
 - Hot-add previously hot-removed memory block if needed
 - Continue until the ratio reaches the threshold (low)

ACMA: Expectations, or Hopes

- System gets free memory of a size that relative to working set
 - 50-100% in above example
- Compaction (migration) for only report-granularity contiguity
- Less compaction/hot-remove failures, due to colder pages first approach
- Easy to use: Set only three thresholds
- More hopes, or crazy thoughts
 - Useful for general memory auto-scaling (for DRAM's power consumption saving?)
 - Expand to be yet another contiguous memory allocator (Access-aware CMA?)

Progress

- No implementation at all
- Detailed RFC idea is sent to the mailing list

QnA for ACMA

DAMOS Auto-tuning Based Tiered Memory Management

https://lore.kernel.org/damon/20231112195602.61525-1-sj@kernel.org/

Various DAMOS-based Tiered Memory Management Approaches

- DAMOS is not supporting tiered memory management at the moment
 - The maintainer willing to, but has no good test setup so far
- First DAMOS patch for tiered memory management was sent 2 years ago
 - No new revision so far, though
- A few various downstream approaches made
 - Results also vary
- Maybe better to have a public approach or idea to discuss

DAMOS-based Tiered Memory Management

- For each CPU-independent NUMA node,
 - If the node has a lower node,
 - Demote cold pages of the current node to the lower node, aiming little fraction (e.g. 5%) of free memory of the current node
 - If the node has a upper node,
 - Promote hot pages of the current node to the upper node, aiming big fraction (e.g., 96%) of used memory of the _upper_ node

node 0 (fast) No lower node, do nothing

DAMOS-based Tiered Memory Management

- For each CPU-independent NUMA node,
 - If the node has a lower node,
 - Demote cold pages of the current node to the lower node, aiming little fraction (e.g. 5%) of free memory of the current node
 - If the node has a upper node,
 - Promote hot pages of the current node to the upper node, aiming big fraction (e.g., 96%) of used memory of the _upper_ node

node 0 (fast) Demote cold pages in node 0 aiming 5% free memory of node 0 node 1 (slow) Promote hot pages in node 1 aiming 96% used memory of node 0

DAMOS-based Tiered Memory Management

- For each CPU-independent NUMA node,
 - If the node has a lower node,
 - Demote cold pages of the current node to the lower node, aiming little fraction (e.g. 5%) of free memory of the current node
 - If the node has a upper node,
 - Promote hot pages of the current node to the upper node, aiming big fraction (e.g., 96%) of used memory of the _upper_ node

node 0 (fast) Demote cold pages in node 0 aiming 5% free memory of node 0
node 1 (slow) Promote hot pages in node 1 aiming 96% used memory of node 0
Demote cold pages in node 1 aiming 5% free memory of node 1
node 2 (sloow) Promote hot pages in node 2 aiming 96% used memory of node 1

Expectations, or Hopes

- High utilization of upper nodes, with hotter pages
- Low utilization of lower nodes, with colder pages
- Auto-tuning-based speed control and overlapping memory util/free goals
 - keep slow but continuous promotion/demotion
- Easy to be applied for multiple tiers
- Possible future extensions
 - General NUMA balancing
 - Extend DAMON to capture accesses maker CPU
 - Combination with ACMA
 - Automatically remove/add tiers depending on real (or, DAMON-) workingset

Progress

- No implementation at all
- Detailed RFC idea is sent to the mailing list

QnA for DAMOS-based Tiered Memory Management

Overview

- DAMON in a Nutshell (10 mins)
- Updates Since Kernel Summit 2022 (10 mins)
- Future Plans (15 mins)
- DAMON Community (5 mins)
- Conclusion and Remaining QnA

• Each time-specified topic will get its own QnA

DAMON Community

Community Members

- DAMON is a community-driven project
- Everyone interested in DAMON is a member
 - Amazon Linux ported initial version of DAMON in their >=v5.4 kernels
 - Android common kernel ported and enabled DAMON_RECLAIM
 - Some companies published their research on DAMON
 - Some academic papers are addressing DAMON

Collaborations

- Collaborating with a number of AWS internal/external people (DAMON community)
- In v6.1..v6.7-rc1, 27 Amazon-external people contributed 51/192 patches for DAMON
 - For v5.15..v6.1, 39 people, 90/163 patches (DAMON is collapsing, or ... stabilized?)
- There were significant contributions to the user space tool (DAMO), too

range	AWS	non-AWS	AWS/non-AWS
v6.1v6.2	28	6	17.65 %
v6.2v6.3	32	16	33.33 %
v6.3v6.4	0	5	100.00 %
v6.4v6.5	19	8	29.63 %
v6.5v6.6	20	7	25.93 %
v6.6v6.7-rc1	42	9	17.65 %
v6.1v6.7-rc1	141	51	26.56 %

Communication Channels

- DAMON-dedicated open mailing list
- Bi-weekly community meetup series
 - Second in-person version will be held as an LPC BoF, at 4:30 pm, today
- Presenting DAMON in conferences since 2019
 - Striving to present for both kernel and user space developers
 - LSFMM, LinuxCon NA/EU, and Kernel Summit in 2023
- Having occasional/regular private meetings on demand

DAMON Community is Waiting For Your Voices

- DMAON is evolving
 - It might not perfectly fit for your use case
- Don't forgive it or wait for someone to implement it; make your voice
 - Report your use case/test results and challenges
 - Ask questions and request features
 - Show your interest to known future works
 - Send patches

Overview

- DAMON in a Nutshell (10 mins)
- Updates Since Kernel Summit 2022 (10 mins)
- Future Plans (15 mins)
- DAMON Community (5 mins)
- Conclusion and Remaining QnA

• Each time-specified topic will get its own QnA

Conclusion

- DAMON answers "which memory is how frequently accessed for how long?"
- DAMOS makes the kernel *just works* in an access-aware manner
- Continuous development is being made
- Please participate in making it better for the community

Questions?

- You can also use
 - The maintainer: sj@kernel.org
 - Project webpage: https://damonitor.github.io
 - Kernel docs for admin and programmers
 - DAMON mailing list: damon@lists.linux.dev
 - DAMON Beer/Coffee/Tea Chat

Backup Slides

DAMON Stack: The Whole Picture of The Stack



depends on