Contribution ID: **33** Type: **not specified**

DAMON: Data Access Monitoring Framework for Fun and Memory Management Optimizations

Wednesday, 26 August 2020 10:00 (45 minutes)

Background

In an ideal world, memory management provides the optimal placement of data objects under accurate predictions of future data access. Current practical implementations, however, rely on coarse information and heuristics to keep the instrumentation overhead minimal. A number of memory management optimization works were therefore proposed, based on the finer-grained access information. Lots of those, however, incur high data access pattern instrumentation overhead, especially when the target workload is huge. A few of the others were able to keep the overhead small by inventing efficient instrumentation mechanisms for their use case, but such mechanisms are usually applicable to their use cases only.

We can list up below four requirements for the data access information instrumentation that must be fulfilled to allow adoption into a wide range of production environments:

- Accuracy. The instrumented information should be useful for DRAM level memory management. Cache-level accuracy would not highly required, though.
- Light-weight overhead. The instrumentation overhead should be low enough to be applied online while making no impact on the performance of the main workload.
- Scalability. The upper-bound of the instrumentation overhead should be controllable regardless of the size of target workloads, to be adopted in general environments that could have huge workloads.
- Generality. The mechanism should be widely applicable.

DAMON: Data Access MONitor

DAMON is a data access monitoring framework subsystem for the Linux kernel that designed to mitigate this problem. The core mechanisms of DAMON called 'region based sampling' and 'adaptive regions adjustment' make it fulfill the requirements. Moreover, its general design and flexible interface allow not only the kernel code but also the user space can use it.

Using this framework, therefore, the kernel's core memory management mechanisms including reclamation and THP can be optimized for better memory management. The memory management optimization works that incurring high instrumentation overhead will be able to have another try. In user space, meanwhile, users who have some special workloads will be able to write personalized tools or applications for deeper understanding and specialized optimizations of their systems.

In addition to the basic monitoring, DAMON also provides a feature dedicated to semi-automated memory management optimizations, called DAMON-based Operation Schemes (DAMOS). Using this feature, the DAMON users can implement complex data access aware optimizations in only a few lines of human-readable schemes descriptions.

Overhead and Performance

We evaluated DAMON's overhead, monitoring quality, and usefulness using 25 realistic workloads on my QEMU/KVM based virtual machine.

DAMON is lightweight. It increases system memory usage by only -0.39% and consumes less than 1% CPU time in the typical case. It slows target workloads down by only 0.63%.

DAMON is accurate and useful for memory management optimizations. An experimental DAMON-based operation scheme for THP removes 69.43% of THP memory overhead while preserving 37.11% of THP speedup. Another experimental DAMON-based reclamation scheme reduces 89.30% of residential sets and 22.40% of system memory footprint while incurring only 1.98% runtime overhead in the best case.

Current Status of The Project

Development of DAMON started in 2019, and several iterations were presented in academic papers[1,2,3], the kernel summit of last year[4], and an LWN article[4]. The source code is available[6] for use and modification, the patchsets[7] are periodically being posted for review.

Agenda

I will briefly introduce DAMON and share how it has evolved since last year's kernel summit talk. I will introduce some new features, including the DAMON-based operation schemes. There will be a live demonstration and I will show performance evaluation results. I will outline plans and the roadmap of this project, leading to a Q&A session to collect feedback with a view on getting it ready for general use and upstream inclusion.

[1] SeongJae Park, Yunjae Lee, Yunhee Kim, Heon Y. Yeom, Profiling Dynamic Data Access Patterns with Bounded Overhead and Accuracy. In IEEE International Workshop on Foundations and Applications of Self-*Systems (FAS* 2019), June 2019. https://ieeexplore.ieee.org/abstract/document/8791992
[2] SeongJae Park, Yunjae Lee, Heon Y. Yeom, Profiling Dynamic Data Access Patterns with Controlled Overhead and Quality. In 20th ACM/IFIP International Middleware Conference Industry, December 2019. https://dl.acm.org/citation.cfm?id=3368125
[3] Yunjae Lee, Yunhee Kim, and Heon. Y. Yeom, Lightweight Memory Tracing for Hot Data Identification, In Cluster computing, 2020. (Accepted but not published yet)
[4] SeongJae Park, Tracing Data Access Pattern with Bounded Overhead and Best-effort Accuracy. In The Linux Kernel Summit, September 2019. https://linuxplumbersconf.org/event/4/contributions/548/
[5] Jonathan Corbet, Memory-management optimization with DAMON. In Linux Weekly News, February 2020. https://lwn.net/Articles/812707/
[6] https://github.com/sjp38/linux/tree/damon/master
[7] https://lore.kernel.org/linux-mm/20200525091512.30391-1-sjpark@amazon.com/

I agree to abide by the anti-harassment policy

I agree

Primary author: Dr PARK, SeongJae (Amazon)

Presenter: Dr PARK, SeongJae (Amazon)

Session Classification: Kernel Summit

Track Classification: Kernel Summit