DAMON Requirements for (More) Access-aware MM of Future

SeongJae (SJ) Park <sj@kernel.org>

https://damonitor.github.io

Recent Proposals for (More) Access-aware MM of Future

- Workingset reporting
- CXL Hotness Monitoring Unit
- Promotion of unmapped page cache folios
  (or, folio_mark_accessed()-based pages promotion)
- PTE A bit-based pages promotion kernel thread (kmmscand)
- AMD IBS-based pages promotion kernel thread (kpromoted)
- MGLRU-based pages promotion kernel thread (klruscand)
- Memory tiering per-cgroup fairness

Requirements for (More) Access-aware MM of Future

- Multiple sources of access information
  - (MG)LRU, file I/O, Accessed bits, CXL HMU, AMD IBS, …
- Fancy controls of management operations
  - Unblock main workloads’ progress
  - Tasks/cgroups level fairness

How DAMON Could Help Multiple Sources Information Requirement

- DAMON Operations Set layer:
  - Address space/fundamental feature (A-bit, LRU activeness, IBS, ...) specific operations layer
  - Core layer pulls the information and distills those into DAMON-region abstraction
    - Use of DAMON-region is not mandated (hacky, though)
  - API callers can implement and use their operations set with the sources

```
/**
 * struct damon_operations - Monitoring operations for given use cases.
[...]
 * @prepare_access_checks should manipulate the monitoring regions to be
 * prepared for the next access check.
 * @check_accesses should check the accesses to each region that made after the
 * last preparation and update the number of observed accesses of each region.
[...]
        void (*prepare_access_checks)(struct damon_ctx *context);
        unsigned int (*check_accesses)(struct damon_ctx *context);
```

How DAMON Could Help Fancy Memory Management Operations

- DAMOS (auto-tuned) quotas for resource usages
  - Give quotas tuning feedback loop input with appropriate data
  - Users can provide arbitrary input or use DAMON self-retrievable metrics
- DAMOS filters for fine target pages selection

How DAMON in Future Could Better Help

- Be a better DAMON: less overhead and more accuracy
  - Efforts are ongoing: Page level monitoring, Intervals auto-tuning, ...
- Optimized API for access reporting (pushing information)
- Less hacky Optional (dis)use of DAMON region abstraction
  - For both monitoring and memory management operations
- Direct folio-level use of controlled memory management operations engine
- DAMOS tuning goal metrics and filter for more use cases including per-cgroup per-node memory usages

DAMON API for Pushing Access Information

```
+/**
+ * damon_report_access - Report identified access events to DAMON.
+ * @pid:              The PID of the virual address space of the address.
+ *                    NULL if it is of the physical address.
+ * @addr:             The start address of the access made region
+ * @size:             Size of the region that the access events were made.
+ * @nr_accesses:      Number of detected accesses to the address range.
+ * @node_id:          NUMA node that made the accesses.
+ *
+ * Report access events to DAMON.  Can be called from any context.
+ */
+void damon_report_access(struct pid *pid, unsigned long addr, unsigned long size,
+              unsigned int nr_accesses, int node_id)
```

DAMOS API for Folio Level Async/Controlled Memory Operations

```
+/**
+ * damos_add_folios - Add a list of folios as highest priority target for given
+ *                    damos.
+ * @scheme:    DAMOS scheme for the specific access-aware operation.
+ * @folios:    List of folios to apply the access-aware operation.
+ *
+ * If a kernel component finds folios that eligible for a specific memory
+ * management operation (e.g., CXL-promotion or demotion), execution of the
+ * operation can be requested to be done by DAMOS using this function.  The
+ * execution of the operation will be asynchronously done by DAMOS worker
+ * thread.  DAMOS features for resource control (DAMOS quotas) will also be
+ * applied.
+ */
+void damos_add_folios(struct damos *scheme, struct list_head *folios)
```

Current Plans

- `damon_report_access()` will be “RFC PATCH”-ed in near future
  - Use cases: per-CPU, write-only monitoring
  - Milestones: Page faults-based operations set, AMD IBS-based one, other h/w feature based ones
- `damos_add_folios()` plan is not clear
  - Not sure if there is a real use case, make your voice

Discussion Points

- We may need this first
- We ain’t gonna need this
- You’re missing this