

DAMON: Long-term Plans for Kernel That {Just Works, Extensible}

Seongjae Park <sj@kernel.org>



**Kernel
Memory Management
MicroConference @**

LINUX

PLUMBERS

CONFERENCE Vienna, Austria / Sept. 18-20, 2024

Notices

- The views expressed herein are those of the speakers;
They do not reflect the views of their employers
- This is for long-term plan discussions rather than presentations



DAMON in One minute

- DAMON is a data access pattern snapshot generator informing
 - which address range is how frequently accessed for how long time
- Supports virtual address spaces and physical address space
- Designed for low/controllable overhead and easy extensions
- DAMOS: access-aware system optimizations engine
 - e.g., reclaim memory regions that not accessed for ≥ 2 minutes
- Being used in real-world products including Aurora Serverless v2



For Kernel That Is Extensible

- For a sort of user-space control plane
 - Kernel knows many things but not everything
- Expose DAMON functionality via sysfs as much as possible
 - For new features, add files and/or directories
 - Aim to be a long-term stable ABI
 - Humans should use a user-space tool instead of ABI
 - damo aims to be a long-term stable user-space tool
- Open to BPF integration (damon_ext?)



For Kernel That Just Works

- For reasonable benefits without additional efforts
 - Users know many things but not necessarily everything
 - Not necessarily optimum for all case
- Add DAMON modules for common usages with simple interface
 - DAMON_RECLAIM, DAMON_LRU_SORT, ...
 - Aim to assist rather than replace
- Continue simplifying UI by *adding* more auto-tuning features/knobs
 - Manual driving handle will continue be provided
- Eventually, a single knob, say, CONFIG_DAMON_X=y
 - Run all auto-tuned DAMON modules by default



What I'm [not] gonna do

I'm gonna do (unless you stop me)

- Make DAMON extensible and just works
- Add more sysfs files under `/sys/kernel/mm/daemon/`

I'm *not* gonna do (unless you ask me to)

- Introduce regressions (by purposes)
- Add intrusive changes to places other than `mm/daemon/`



Discussion Time!

- Is pursuing two goals a bad idea?
- Is sysfs interface a bad idea?
- Will this disturb other kernel developers?
- Should DAMON modules be replacements rather than assistance?
- How to determine DAMON is ready for CONFIG_DAMON_X=y?
- Should the verbal promises be documented somewhere?



Backup Slides



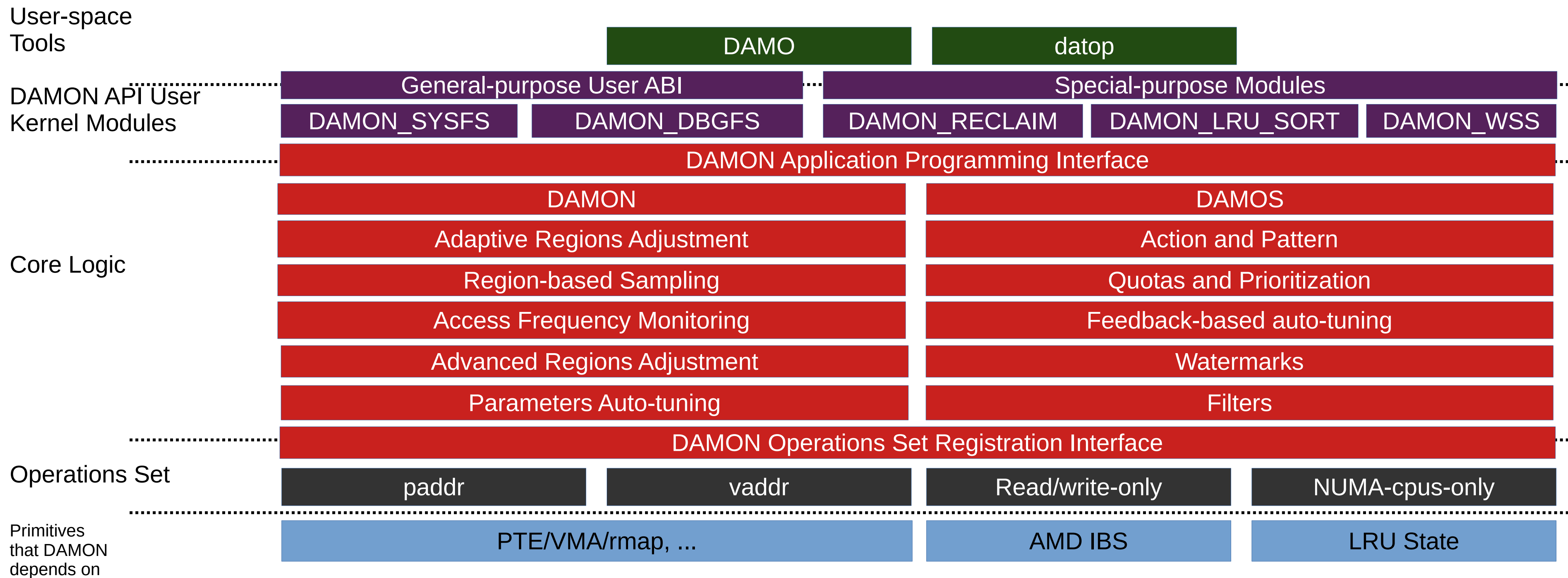
**Kernel
Memory Management
MicroConference @**

LINUX

PLUMBERS

CONFERENCE Vienna, Austria / Sept. 18-20, 2024

DAMON Stack, In a Future



DAMON sysfs interface

- Following sysfs philosophy
- Extensible
- *Strive* to be long-term stable

```
/sys/kernel/mm/daemon/admin
| kdamonds/nr_kdamonds
| | 0/state,pid
| | | contexts/nr_contexts
| | | | 0/avail_operations,operations
| | | | | monitoring_attrs/
| | | | | | intervals/sample_us,aggr_us,update_us
| | | | | | nr_regions/min,max
| | | | | targets/nr_targets
| | | | | | 0/pid_target
| | | | | | | regions/nr_regions
| | | | | | | | 0/start,end
| | | | | | | | ...
| | | | | ...
| | | | schemes/nr_schemes
| | | | | 0/action,target_nid,apply_interval_us
| | | | | | access_pattern/
| | | | | | | sz/min,max
| | | | | | | nr_accesses/min,max
| | | | | | | age/min,max
| | | | | | quotas/ms,bytes,reset_interval_ms,effective_bytes
| | | | | | | weights/sz_permil,nr_accesses_permil,age_permil
| | | | | | | goals/nr_goals
| | | | | | | | 0/target_metric,target_value,current_value
| | | | | | | watermarks/metric,interval_us,high,mid,low
| | | | | | | filters/nr_filters
| | | | | | | | 0/type,matching,memcg_id
| | | | | | | stats/nr_tried,sz_tried,nr_applied,sz_applied,qt_exceeds
| | | | | | | | tried_regions/total_bytes
| | | | | | | | | 0/start,end,nr_accesses,age
| | | | | | | | | ...
| | | | | ...
| | | | ...
| | | ...
| | ...
| ...
```

<https://www.kernel.org/doc/html/latest/admin-guide/mm/daemon/usage.html#files-hierarchy>

