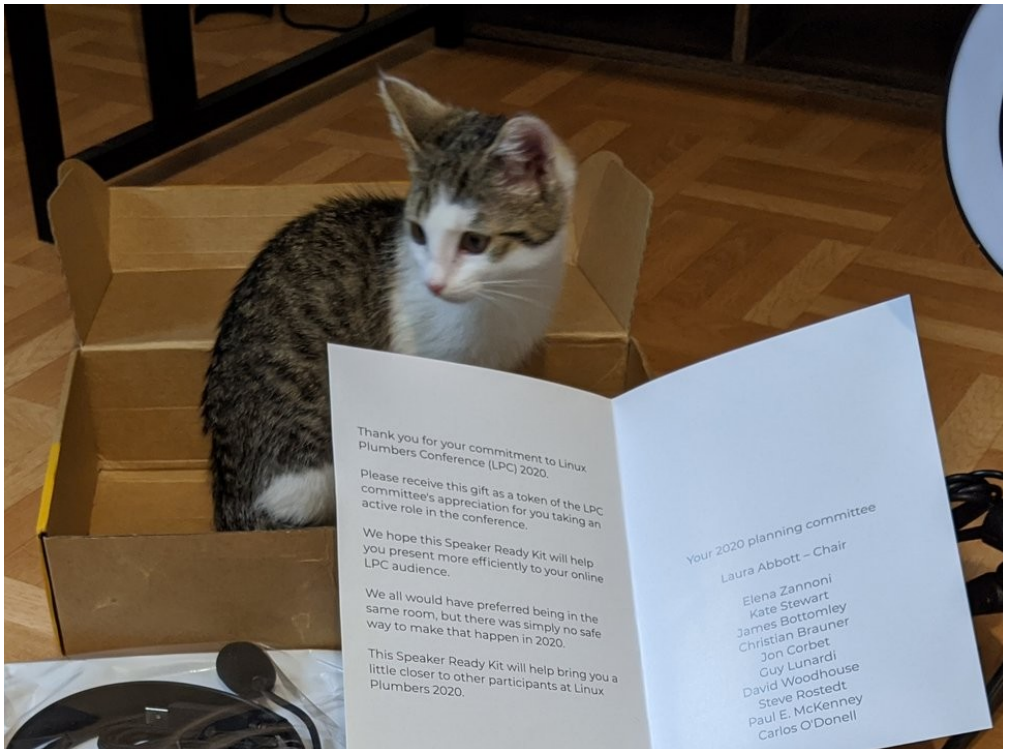
Kernel Summit @
Linux
Plumbers Conference | Dublin, Ireland Sept. 12-14, 2022

DAMON:
Current Status and Future Plans

SeongJae Park <sj@kernel.org>

Disclaimer

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



https://twitter.com/sjpark0x00/status/1295387149018300419/photo/1

I, SeongJae Park <sj@kernel.org>

- Just call me SJ, or whatever better for you to pronounce
- Kernel Development Engineer at Amazon Web Services
- Interested in the memory management and the parallel programming
- Maintaining DAMON

Overall Motivation: Access-aware Linux Kernel

- DRAM is a major infrastructure expense
- We can reduce the expense by improving memory efficiency
- We can do that if we can make better data management decision
- We can do that if we can estimate data access patterns
- We can do that if we can aware of current access patterns, and operate the system with the information
- Linux is an Access-aware OS kernel, but apparently could do better
- And, we could make it highly *user-space-controllable* while *just works*

Overview

- Current Status
  - DAMON
  - DAMOS
  - DAMON Modules
  - User-space, Testing, and Community
- Future Plans
  - DAMON/DAMOS Extension and Improvements
  - DAMON Auto-tuning and Modules Unification
  - Misc Kernel-side TODOs
  - More User-space Components and Testing
- Conclusion

DAMON: Motivation

- Data access monitoring overhead is high and unbounded
  - High overhead can also affect monitoring results accuracy
    - Maybe the uncertainty principle can be applied here
  - Unbounded overhead is inherent in fixed granularity monitoring
    - As the amount of data to monitor increases, overhead grows
- High overhead is inevitable for precise large amount of information
  - Do we really paying for only what we need? No one price plan fits all
- No simple and centralized interface for data access monitoring

DAMON: Data Access MONitor

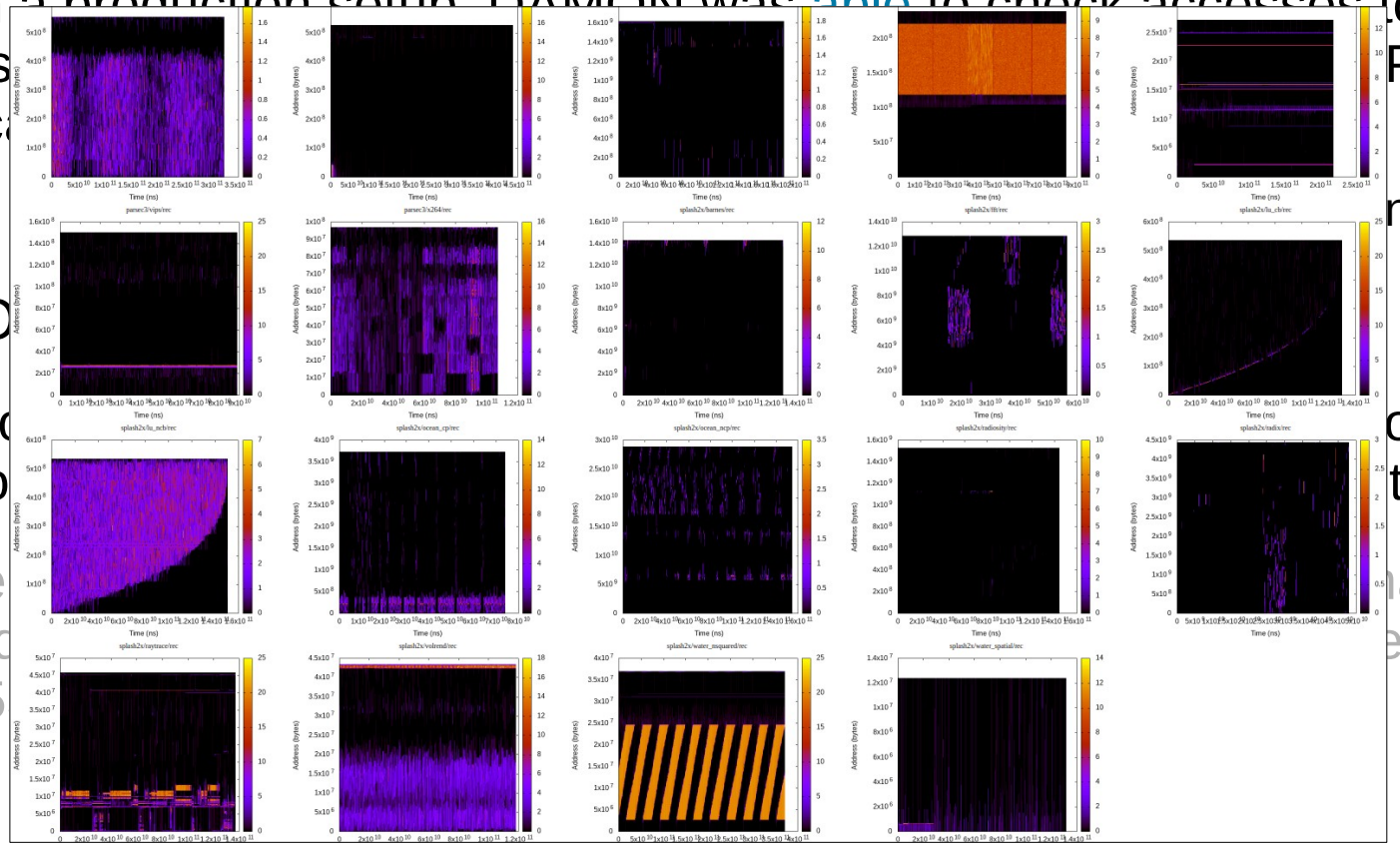
- A data access monitoring framework of Linux kernel
  - Aims to be the simple and centralized access monitoring interface
- Allows users to know which memory area is how frequently accessed
  - Let users set the min accuracy and max overhead
  - DAMON makes the best-effort accuracy / overhead trade-off
    - Multiple mechanisms could be used for this
    - At the moment, Adaptive access pattern-oriented regions adjustment mechanism is the default and single mechanism for this
      (For details, refer to the doc or ksummit'20 DAMON talk)
    - The default mechanism allows bounded overhead regardless of the monitoring target memory size
    - Experimental efficient page granularity monitoring mechanism was already implemented; Not merged in due to the absence of real use case so far
- Allows easy extension and flexible customization

DAMON: Evaluation

- DAMON is lightweight
  - On a production setup, DAMON was able to check accesses to entire system memory (68.60 GB) every 5 msecs with <1% single CPU time (scans 68.6GB / (5ms * 1%) = 1,372 TB/s)
    - Note: DAMON overhead does not increase as the memory size increases
- DAMON is accurate
  - Shows sane monitoring results with realistic benchmark workloads and production workloads (found 7GB working set and 4KB hottest region)
  - Identifying hot memory regions from DAMON results with human eyes and modifying the program to do `mlock()` the regions achieves up to 2.55x speedup under memory pressure[1,2]

DAMON: Evaluation

- DAMON is lightweight
  - On a production setup, DAMON was able to check accesses to entire sys PU time (sc
    - ncreases
- DAMO
  - Sho oads and pro t region)
  - Ide an eyes and es up to 2.5



DAMON: Evaluation

- DAMON is lightweight
  - On a production setup, DAMON was able to check accesses to entire system memory (68.60 GB) every 5 msecs with <1% single CPU time (scans 68.6GB / (5ms * 1%) = 1,372 TB/s)
    - Note: DAMON overhead does not increase as the memory size increases
- DAMON is accurate
  - Shows sane monitoring results with realistic benchmark workloads and production workloads (found 7GB working set and 4KB hottest region)
  - Identifying hot memory regions from DAMON results with human eyes and modifying the program to do `mlock()` the regions achieves up to 2.55x speedup under memory pressure[1,2]

DAMON: History and Status

- 2020-01: RFC v1 posted
- 2021-05: Merged in Amazon Linux 2 >=5.4 Kernels
- 2021-09: Merged in v5.15-rc1
- 2022-04: Enabled in Android GKI

Overview

- Current Status
  - DAMON
  - DAMOS
  - DAMON Modules
  - User-space, Testing, and Community
- Future Plans
  - DAMON/DAMOS Extension and Improvements
  - DAMON Auto-tuning and Modules Unification
  - Misc Kernel-side TODOs
  - More User-space Components and Testing
- Conclusion

DAMOS: Motivation

- Now DAMON-based optimizations are available (by both kernel and user)
  - Common steps for the optimizations would be…
  - Run DAMON, get/analyze monitoring results, and make some memory management decisions (e.g., reclaim cold pages, use THP for hot pages, …)
- Not so ideal because
  - Steps could be repetitive
  - User-space approach could be inefficient due to kernel-user context switches
    - BPF-like approaches would be available, though
- Couldn’t we offload the works to DAMON in the kernel?
  - Don’t repeat yourself
  - Use the information directly from where it generated

DAMOS: DAMON-based Operation Schemes

- A core feature of DAMON for reducing the repetitive works
- Receives ‘schemes’; each scheme is constructed with
  - Target access pattern
    - Ranges of size, access frequency, and age of memory regions
  - Memory management action, e.g., PAGEOUT, HUGEPAGE, ...
- DAMOS automatically finds the memory region of the target pattern using DAMON and applies the action to the region
- Now users can make DAMON-based optimizations with no-code

```
# format is:
# <min/max size> <min/max frequency (0-100)> <min/max age> <action>
#
# if a region of size >=4KB didn’t accessed for >=2mins, page out
4K max       0 0       2m max       pageout
```

DAMOS: Evaluation

- Implemented main ideas of two state-of-the-arts works with DAMOS
  - Access-aware THP collapse/split
    - The original work was accepted to a top-tier conf (OSDI'16)
    - Re-implemented with two lines of DAMOS config
    - Our version removes 76.15% of THP memory bloats while preserving 51.25% of THP speedup
  - Proactive reclamation
    - The original work was accepted to another top-tier conf (ASPLOS'19) and being used for Google fleets
    - Re-implemented with one line of DAMOS config
    - reduces 93.38% of residential sets and 23.63% of system memory footprint while incurring only 1.22% runtime overhead in the best case.
  - For more details, please read the report

DAMOS: More Features For Production

- For productions that safety-critical, DAMOS provides additional features
- Time/space quota per a given time interval
  - DAMOS try to use CPU time no more than the given time quota
  - DAMOS try to apply the action to memory no more than the space quota
- Regions prioritization
  - Under the quota, DAMOS applies the action to prioritized regions first
  - Prioritization logic can be customized for different DAMOS actions
    - In case of RECLAIM, older and colder pages are prioritized by default
- Three watermarks (high, mid, low) with user-specified metric (e.g., freemem)
  - Deactivate if the metric > high_watermark or metric < low_watermark
  - Activate if the metric < mid_watermark and metric > low_watermark
  - Avoid DAMOS using any resource under a peaceful or a catastrophic situation
- With online tuning, these opens some interesting capabilities

DAMOS: History and Status

- 2020-02: RFC v1 posted
- 2021-05: Merged in Amazon Linux 2 >=5.4 Kernels
- 2021-11: Merged in v5.16-rc1
- 2022-04: Enabled in Android GKI

Overview

- Current Status
  - DAMON
  - DAMOS
  - DAMON Modules
  - User-space, Testing, and Community
- Future Plans
  - DAMON/DAMOS Extension and Improvements
  - DAMON Auto-tuning and Modules Unification
  - Misc Kernel-side TODOs
  - More User-space Components and Testing
- Conclusion

DAMON Modules: Motivation

- A common question: Policy in user-space or kernel-space?
  - Only user-space policies could make ultimate result
    - True for some cases, might not be true for some other cases
    - Requires non-trivial user-space efforts investment, though
  - Kernel-space policies could be worthy to try
    - Useful for someone who cannot get help from the user-space ultimate policy
    - Kernel just works at least reasonably
- DAMON aims to convince both; It hence provides
  - Highly tunable and flexible user interface via sysfs, and
  - DAMOS application static kernel modules for general use cases
    - Supposed to be useful in general without complex tuning
    - Can still be highly tuned via parameters, but the default values are supposed to make no fantastic benefits but some, or at least *no harm*

DAMON_RECLAIM

- DAMON-based proactive reclamation kernel module
  - Basically same to the previously shown DAMOS schemes
- Written using DAMOS' simple kernel API
  - Excepting the code for module parameters, only 188 lines of code
  - SJ live-coded almost similar one in 10 minutes at ksummit'21
- Aims to be used on the production
  - Ensure the safety using the quotas and watermarks
  - The quotas and watermarks are conservatively tuned
    - Aims to make no fantastic benefits but some with no harm
    - Can be tweaked via module parameters

DAMON_LRU_SORT

- Yet another DAMON module
- Sorts pages in LRU lists for better performance
  - Can be used for proactive reclamation improper situations
    - E.g., Having slow or restrictive storage devices
  - `mark_page_accessed()` hot pages and `deactivate_page()` cold pages
    - We could do more fine-grained LRU lists-adjustment in a future
- Reduces 10-20% memory pressure stall time (memory PSI, some) under artificial memory pressure

DAMON Modules: History and Status

- 2021-06: DAMON_RECLAIM RFC v1 posted
- 2021-10: DAMON_RECLAIM merged in Amazon Linux 2 >=5.10 kernel
- 2022-01: DAMON_RECLAIM merged in v5.16-rc1
- 2022-04: DAMON_RECLAIM Enabled in Android GKI
- 2022-05: DAMON_LRU_SORT RFC v1 posted
- 2022-08: DAMON_LRU_SORT merged in v6.0-rc1

Overview

- Current Status
  - DAMON
  - DAMOS
  - DAMON Modules
  - User-space, Testing, and Community
- Future Plans
  - DAMON/DAMOS Extension and Improvements
  - DAMON Auto-tuning and Modules Unification
  - Misc Kernel-side TODOs
  - More User-space Components and Testing
- Conclusion

DAMON User-space Applications

- DAMON provides two user interfaces based on debugfs and sysfs
  - Those are for user space tools implementation, not for manual use
  - The debugfs interface will be deprecated after next LTS is released
- SJ develops and maintains his own user space tool
  - Available at Github (https://github.com/awslab/damo) and PyPI (https://pypi.org/project/damo/)
  - Provides all DAMON features with human-friendly interface
- Alibaba developed their own DAMON user-space tool: https://github.com/aliyun/data-profile-tools

Testing DAMON

- Maintains open source tests suite for DAMON: https://github.com/awslabs/damon-tests
  - Provides functionality tests and performance tests
    - Functionality tests are based on selftests and kunit tests
    - Performance tests use PARSEC3/SPLASH-2X workloads
  - We run
    - Functionality tests for every update of SJ’s hacking tree, >=v5.15.y, mm-unstable, and the mainline
    - Performance tests for latest SJ’s hacking tree every day
- *Did* some fuzzing tests
  - Made DAMON debugfs interface syzkaller description and upstreamed
  - No DAMON sysfs interface syzkaller description yet, though

DAMON Community

- DAMON is developed by its open and (hopefully) inclusive community
- Several people, institutes, and companies are participating
- Dedicated open mailing list: damon@lists.linux.dev
- Open, regular, and informal virtual bi-weekly meeting series
  - https://lore.kernel.org/damon/20220810225102.124459-1-sj@kernel.org/
  - We will have an in-person instance of it 17:00 today at Meeting Room 9
  - Aims to be used for aligning participants' conflicting goals and prioritizing TODO items

Overview

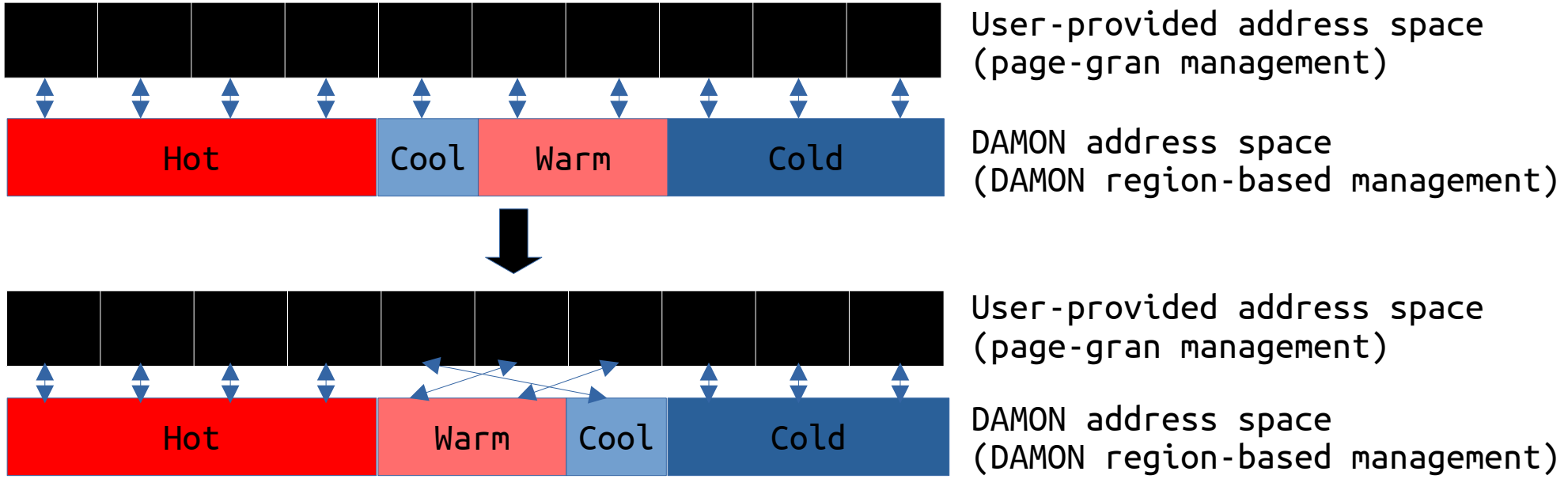
- Current Status
  - DAMON
  - DAMOS
  - DAMON Modules
  - User-space, Testing, and Community
- Future Plans
  - DAMON/DAMOS Extension and Improvements
  - DAMON Auto-tuning and Modules Unification
  - Misc Kernel-side TODOs
  - More User-space Components and Testing
- Conclusion

Extending DAMON

- Can easily extend for various address spaces and use cases
  - Need to implement new monitoring operations set for the use case
- Currently, monitoring operations for virtual address spaces and the physical address space are available
- Imaginable extensions include
  - More efficient page-granularity system monitoring
    - Current page-granularity monitoring is only for proof of concepts
    - We already implemented an experimental version of this
  - Monitoring operations using IBS or LRU-position
  - Read-only, write-only, cgroups, for only specific file-backed memory, ...

Improving DAMON Accuracy/Overhead

- Adaptive monitoring attributes adjustment and regions splitting
  - Find struggling regions and apply aggressive adaptation
  - Page-gran monitoring will be re-implemented to be used for comparison
- Remapping regions based on monitoring results, to sorted by hotness
  - The spatial locality assumption of memory regions will be more reasonable
  - DAMON-internal address space would be needed for usual cases



DAMOS Allow/Deny-list

- DAMON is simple
  - No special care of anon, file-backed, cgroups, but only access pattern
- Users know some important memory areas that shouldn’t be distracted by DAMOS
  - Users could do `madvise()` or `mlock()` or somewhat
  - DAMOS could have allow-list and deny-list for such regions
    - E.g., “do the access-aware proactive reclamation for system memory but these processes, these files, and these cgroups”

More DAMOS Actions and Modules

- DAMON might be able to be used to help
  - Efficient but performant THP promotion/demotion
  - Page migration target (for compaction or CMA) selection
    - Cold pages might be not pinned
  - Tiered-memory management

Overview

- Current Status
  - DAMON
  - DAMOS
  - DAMON Modules
  - User-space, Testing, and Community
- Future Plans
  - DAMON/DAMOS Extension and Improvements
  - DAMON Auto-tuning and Modules Unification
  - Misc Kernel-side TODOs
  - More User-space Components and Testing
- Conclusion

DAMON Modules Unification

- DAMON modules are exclusive at the moment
- Would be better to unify all the modules
  - So that all modules can be enabled in one step: Just working kernel
    - Say, CONFIG_ACCESS_AWARE_MM?
  - But, should still be able to be used separately (no behavior/interface change)
  - Support of multiple DAMON contexts on single kdamond is required

DAMON Auto-tuning

- DAMON provides not small number of knobs, tuning is not so trivial
  - Adding yet other more intuitive knobs for auto-tuning other knobs could help
  - E.g., a monitoring time quota for `nr_{min,max}_regions` auto-tuning
- Transparent Memory Offloading (TMO) like Auto-tuning of DAMON_RECLAIM would be possible and seems promising
  - Monitor PSI and adjusts DAMOS quota
  - Running (too-)simplified and aggressive TMO implementation with DAMON_LRU_SORT reduces PSI 10-20%
- Might be able to be generalized for general DAMOS
  - Already available via online tuning, but why not make kernel just work?
  - Provides feedback interface, let users put metric like PSI that really matters to them; DAMON does the auto-tuning based on the metric value
  - Allow special keywords for kernel-better-collecting metrics like PSI

Overview

- Current Status
  - DAMON
  - DAMOS
  - DAMON Modules
  - User-space, Testing, and Community
- Future Plans
  - DAMON/DAMOS Extension and Improvements
  - DAMON Auto-tuning and Modules Unification
  - Misc Kernel-side TODOs
  - More User-space Components and Testing
- Conclusion

Misc Kernel-side TODO Items

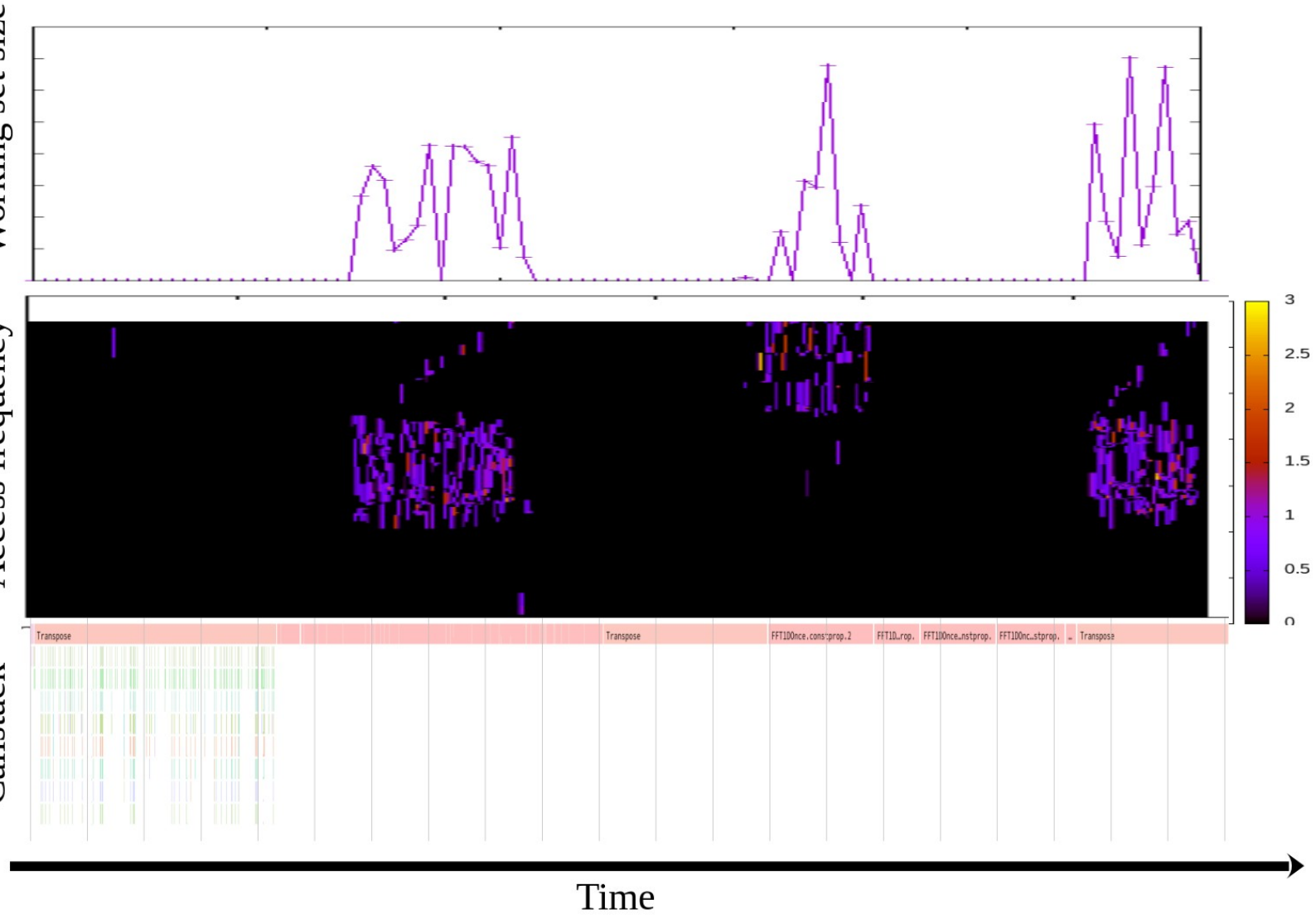
- Non-root user DAMON interface
- DAMON running on no kdamond but a user-process
- Efficient DAMON monitoring results exposing
- Querying regions of specific access pattern

More User-space Components

- Out of SJ’s hand; waiting for user space experts’ contributions
- User-space library, say, `libdamon`
  - While preparing for DAMON system call, let’s hide the gap
- Hand over of the official DAMON user-space tool
  - SJ is bad at user space programming
  - Alibaba’s `datop` might be the one candidate?
- More tools
  - Showing access pattern with callstack will give us interesting profiling cap
  - User-space optimizer/auto-tuner? RSS/WSS based OOM killer?
- More services
  - DAMON/DAMOS as-a Service
    - Working set size monitoring/alarming/auto scaling/...

More User-space Components: Imaginable Profiler

- O
- Us
  - Ha
  - M
  - M



More User-space Components

- Out of SJ’s hand; waiting for user space experts’ contributions
- User-space library, say, `libdamon`
  - While preparing for DAMON system call, let’s hide the gap
- Hand over of the official DAMON user-space tool
  - SJ is bad at user space programming
  - Alibaba’s `datop` might be the one candidate?
- More tools
  - Showing access pattern with callstack will give us interesting profiling cap
  - User-space optimizer/auto-tuner? RSS/WSS based OOM killer?
- More services
  - DAMON/DAMOS as-a Service
    - Working set size monitoring/alarming/auto scaling/...

Testing

- DAMON tests suite
  - Use of more workloads
  - More functionality tests case
  - More unit tests
  - Integrating in the kernel CI systems
- Fuzzing Support
  - DAMON sysfs syzkaller description should be made

Summary

- Current Status
  - In the kernel: DAMON, DAMOS, DAMON Modules
  - In userspace: >=2 Tools
  - In community: open and (hopefully) inclusive culture and chat series
- Future Plans
  - Extended and improved DAMON/DAMOS/DAMON Modules
  - Convenient and intuitive semi-auto tuning of DAMON/DAMOS
  - Unified DAMON Modules: Access-aware LInuX Assistant
  - More user-space libraries, tools, and services
  - The community!

Questions?

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

https://kids.nationalgeographic.com/content/dam/kids/photos/animals/Birds/A-G/adelie-penguin-jumping-ocean.ngsversion.1396530997321.adapt.1900.1.jpg