

# Actionable Data Access Monitoring Output Data and Format

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



# Table of Contents

- Why Data Access Pattern
- DAMON in Nutshell
- Available Access Pattern Formats
- Discussion

# Why Data Access Pattern

- Restricted seats, various audiences
- To see how we are doing
- To fore-see what we should do
- To get actionable insights and make it

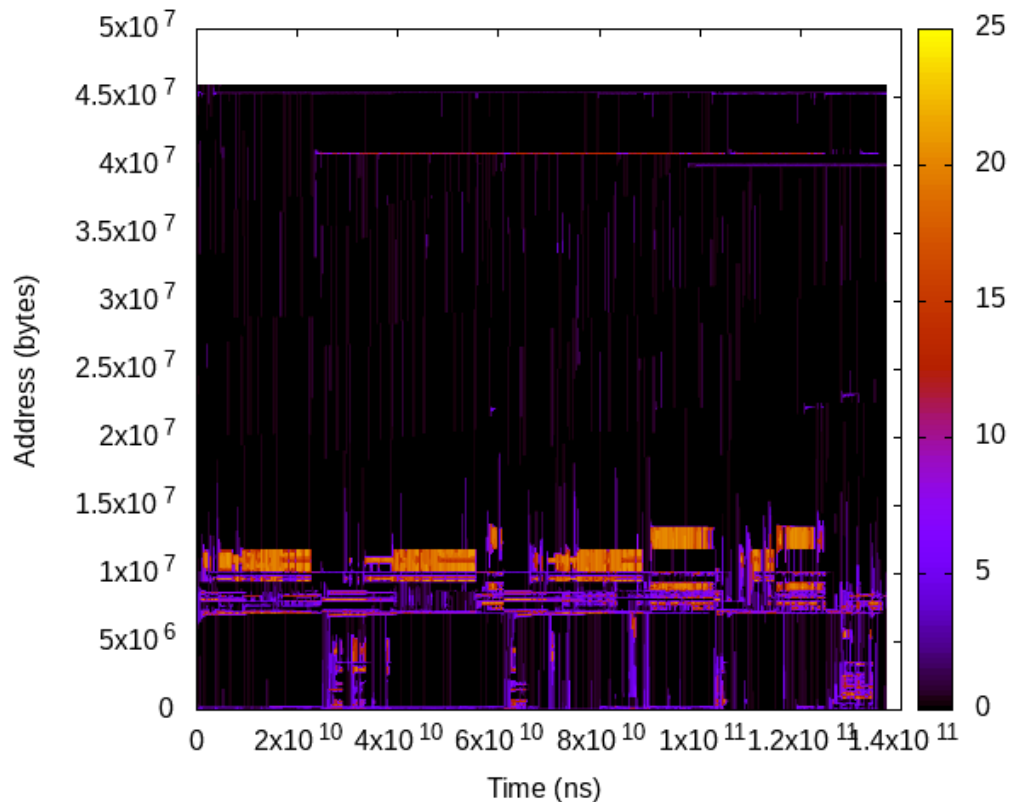
# DAMON in Nutshell

# What It Does

- Spawn a kernel thread that
- See if each page is accessed, every 5ms
- Inform users the findings, every 100ms
- Time intervals can be [auto-]tuned

# Providing Access Information

- Location
- Frequency
- Stability
- Recency



Access pattern heatmap of  
Splash2x/Raytrace

# Overhead and Accuracy

- Utilize adaptive sampling/aggregation
- The upper-limit overhead is tunable
  - Regardless of the memory size
- 0.0x% single CPU use on real workloads
- Light, accurate, scalable, automated

# Availability and Usages

- Available on  $\geq 5.15$  upstream kernels
- Enabled on most **distro** kernels
- Being used by products and researches



# DAMOS: Second Face of DAMON

- DAMon-based Operation Schemes
  - “Page out cold memory”
  - “Use huge pages for hot memory”
- Turn DAMON into access-aware memory management system operations engine

# DAMON in One Sentence

“DAMON is a Linux kernel subsystem for efficient access monitoring and access-aware system operations.”

# Community

- Waiting for any *\*selfish\** discussion
- Public channels
  - Mailing list: [damon@lists.linux.dev](mailto:damon@lists.linux.dev)
  - Project website: <https://damonitor.github.io/>
- Private channels
  - Maintainer email: [sj@kernel.org](mailto:sj@kernel.org)
  - DAMON Beer/Coffee/Tea Meetup

# Available Formats (Access Pattern Visualizations)

# It's Format, Stupid!

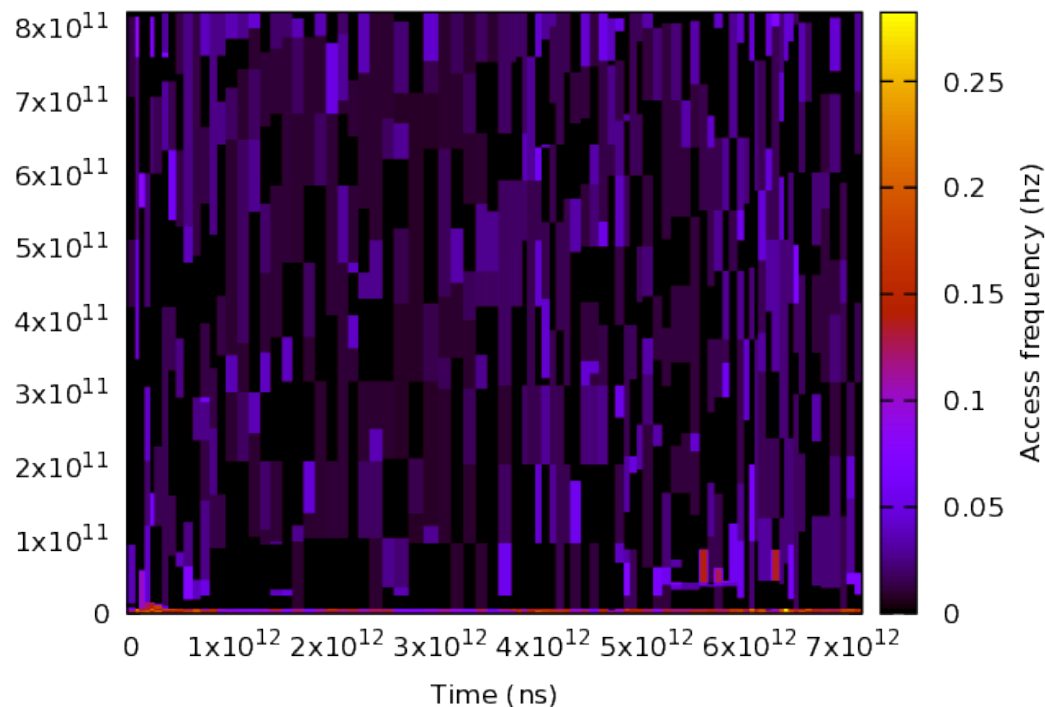
- Having data does nothing
- Same data causes different insights
- Most challenging part
  - Convincing “human”, not “machines”
  - Anyway I’m keeping trying...

# Formats for All DAMON Info

- Straightforward (or, lazy) format
  - For location, frequency, stability, and recency
- Namely, heatmap and snapshot

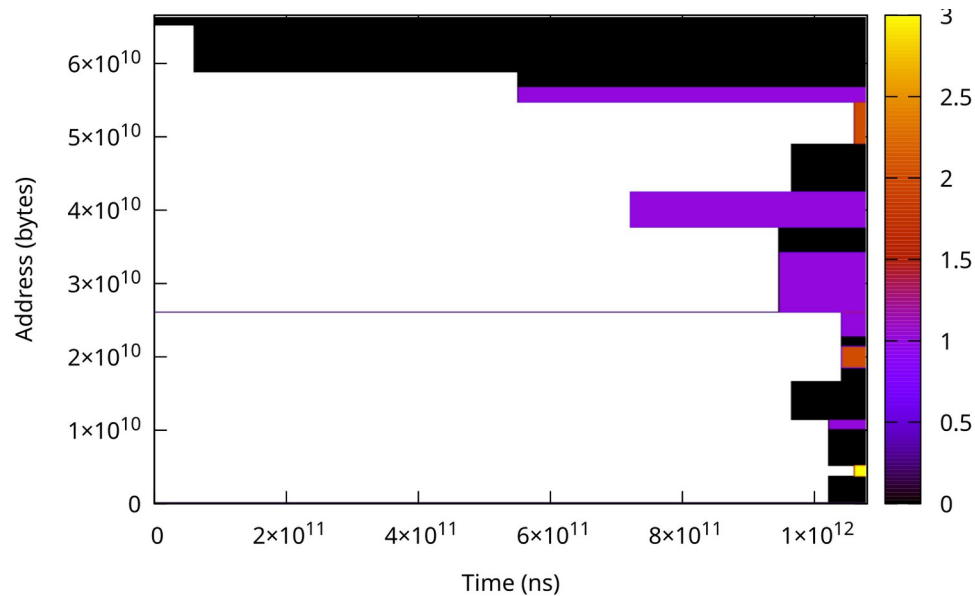
# Heatmap

- X-axis: time
- Y-axis: address
- Color: frequency
- Scoping is important (or, challenging)



# Snapshot

- A slice of heatmap
- Scoping is still the thing





# Formats for DAMON++ Data

- DAMON cannot capture the world alone
- Use it together with other tools to record
  - Memory alloc/free/RSS data
  - Hotspot functions
- And show all together as time series data

# Holistic Snapshot[s]

- Ani-GIF

```
# Heapmap
# target 0, address range 94399443111936-94399839154176
# x-axis: space (94399443111936-94399839154176: 377.695 MiB)
# y-axis: time (11983476845000-11984286267000: 809.422 ms)
# resolution: 80x5 (4.721 MiB and 161.884 ms for each character)
# target 0, address range 140283056488448-140283168743424
# x-axis: space (140283056488448-140283168743424: 58888884 MiB)
# y-axis: time (11983476845000-11984286267000: 809.422 ms)
# resolution: 80x5 (1.338 MiB and 161.884 ms for each character)
# target 0, address range 140723243646976-140723243782144
# x-axis: space (140723243646976-140723243782144: 131.953 KiB)
# y-axis: time (11983476845000-11984286267000: 809.422 ms)
# resolution: 80x5 (1.649 KiB and 161.884 ms for each character)

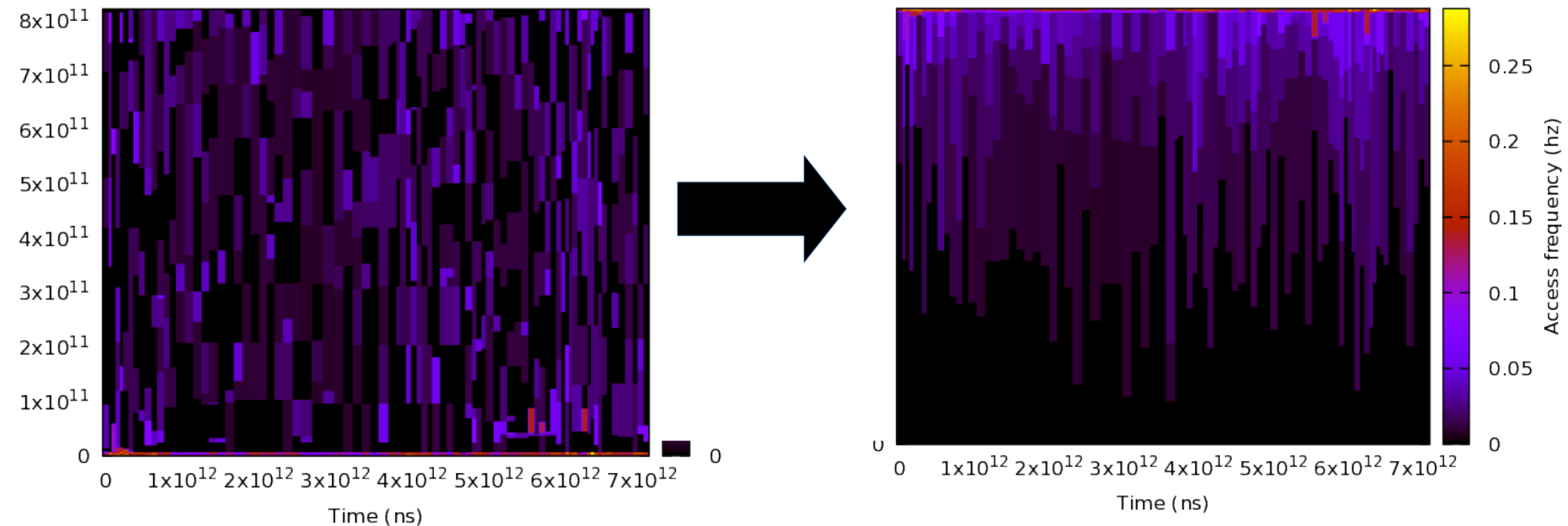
# Memory Footprints Distribution
percentile      0          25          50          75         100
wss             9.652 MiB    9.660 MiB    12.855 MiB    23.418 MiB    43.359 MiB
rss            100.867 MiB   100.867 MiB   100.867 MiB   100.867 MiB   100.867 MiB
vsz            104.535 MiB   104.535 MiB   104.535 MiB   104.535 MiB   104.535 MiB
sys_usd         2.241 GiB    2.241 GiB    2.271 GiB    2.271 GiB    2.271 GiB

# Hotspot functions
# Samples: 20K of event 'cpu-clock:ppph'
# Event count (approx.): 511700000
#
# Overhead  Command      Shared Object      Symbol
# .....
#
# 81.56% swapper      [kernel.vmlinux]   [k] pv_native_safe_halt
# 17.63% masim        [kernel.vmlinux]   [.] do_seq_wq
# 0.04% ps            [kernel.vmlinux]   [k] do_syscall_64
# 0.03% ps            [kernel.vmlinux]   [k] memset_orig
# 0.03% ps            [lib.so.6]         [.] read
# 0.03% ps            [kernel.vmlinux]   [k] kmem_cache_alloc_noprof
# 0.02% ps            [kernel.vmlinux]   [k] mod_objcg_state
# 0.02% ps            [kernel.vmlinux]   [k] _d_lookup_rcu
# 0.02% ps            [kernel.vmlinux]   [k] do_task_stat
# 0.01% ps            [lib.so.6]         [.] _close
```

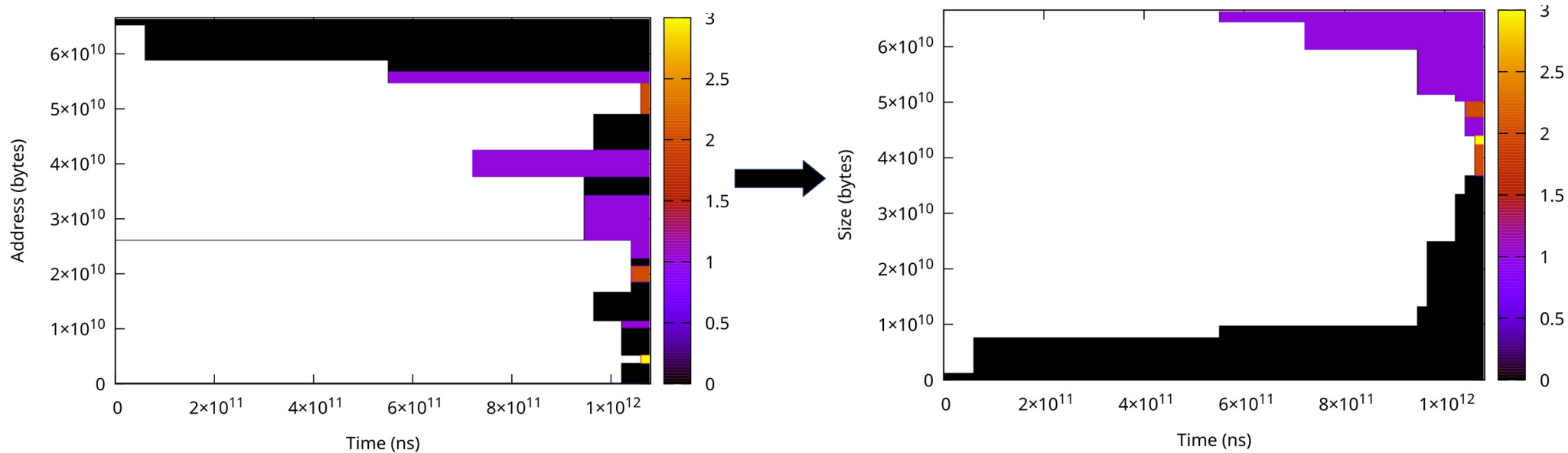
# Formats for DAMON-- Data

- DAMON data is just right for machines
- Too much for human in some cases
- Visualize after dropping some information
  - Specifically, “location”
- Namely, hotness-sorted heatmap and snapshot

# Hotness-sorted Heatmap



# Hotness-sorted Snapshot



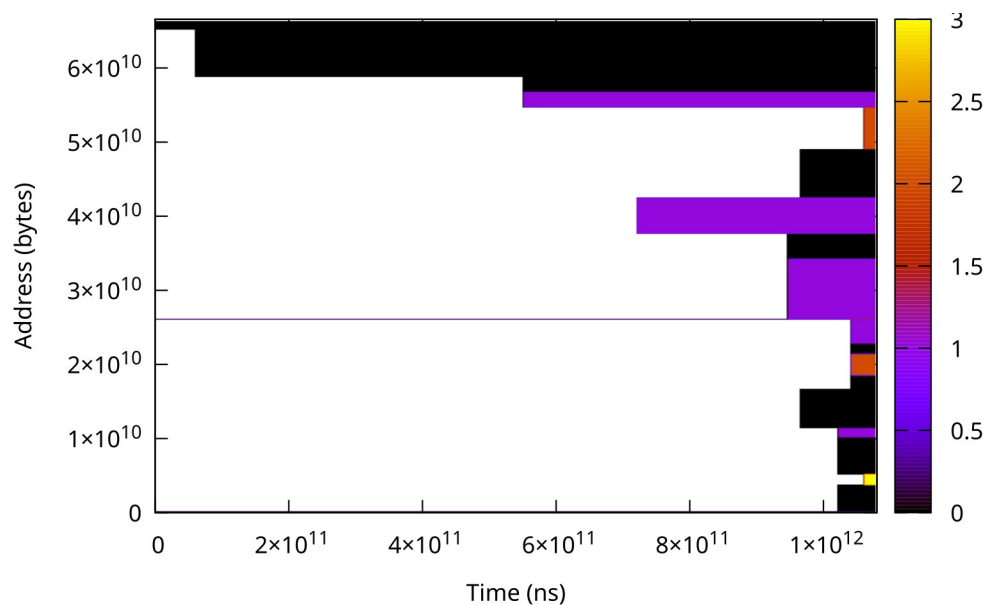
# For Coarse-grain DAMON-- Data

- Lines: informative
  - Encourage “creative” interpretation
- Discrete numbers are *alarm*-able
  - Better for busy “fleets”

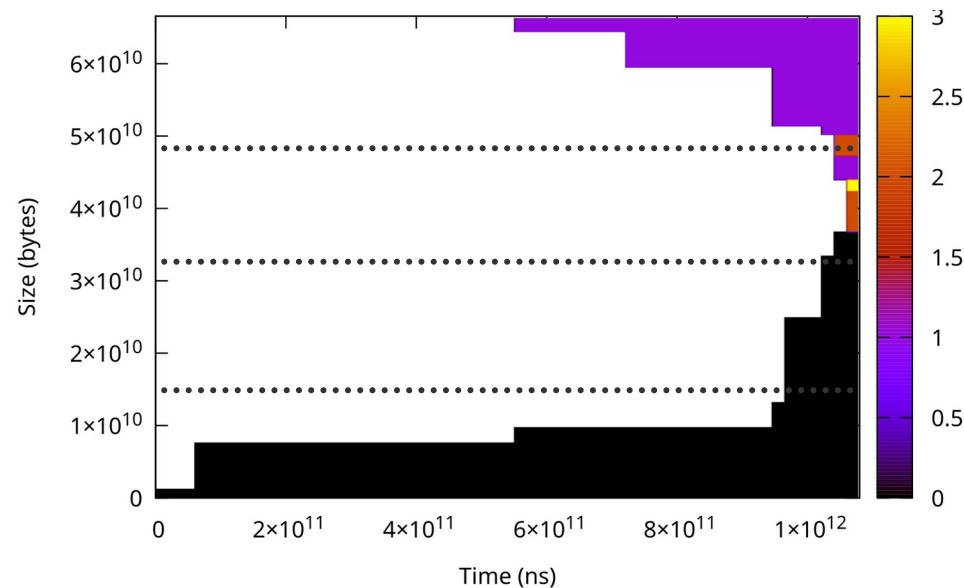
# Idle Time Percentiles: Concept

- Idle time: how long it was not accessed
- Percentile: that of the statistics

Unsorted snapshot



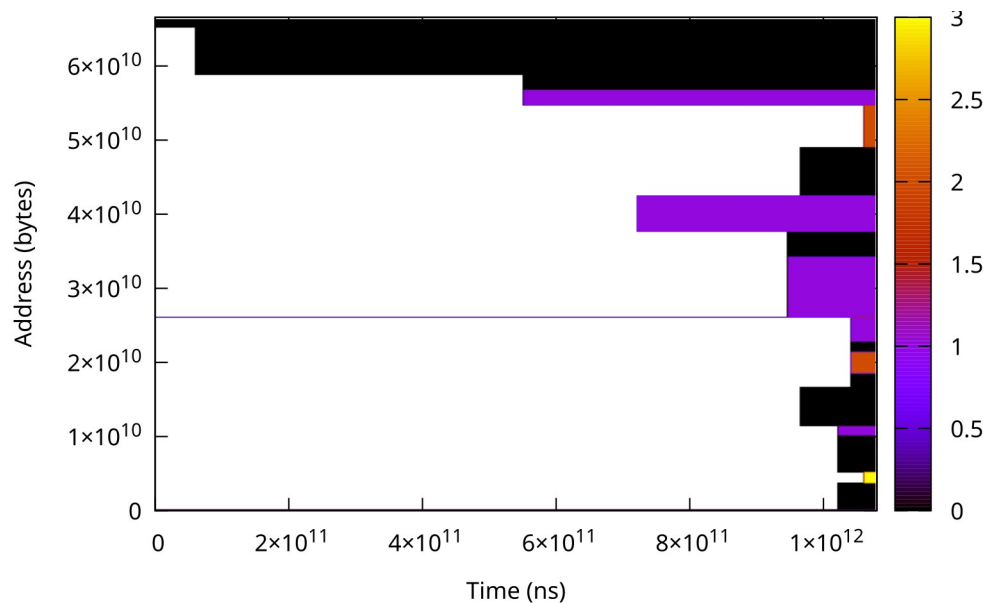
Sorted by access frequency



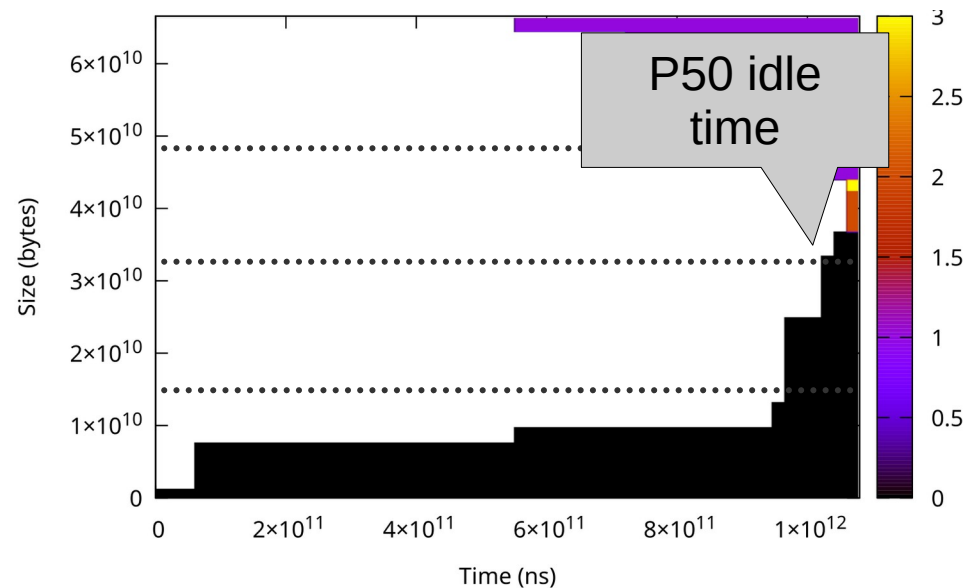
# Idle Time Percentiles: Concept

- Idle time: how long it was not accessed
- Percentile: that of the statistics

Unsorted snapshot



Sorted by access frequency

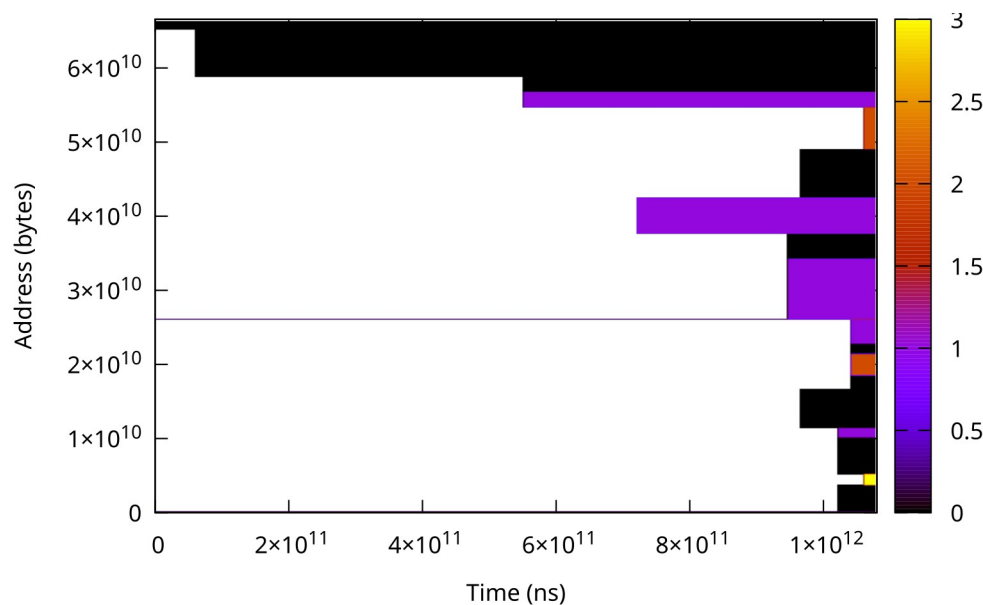




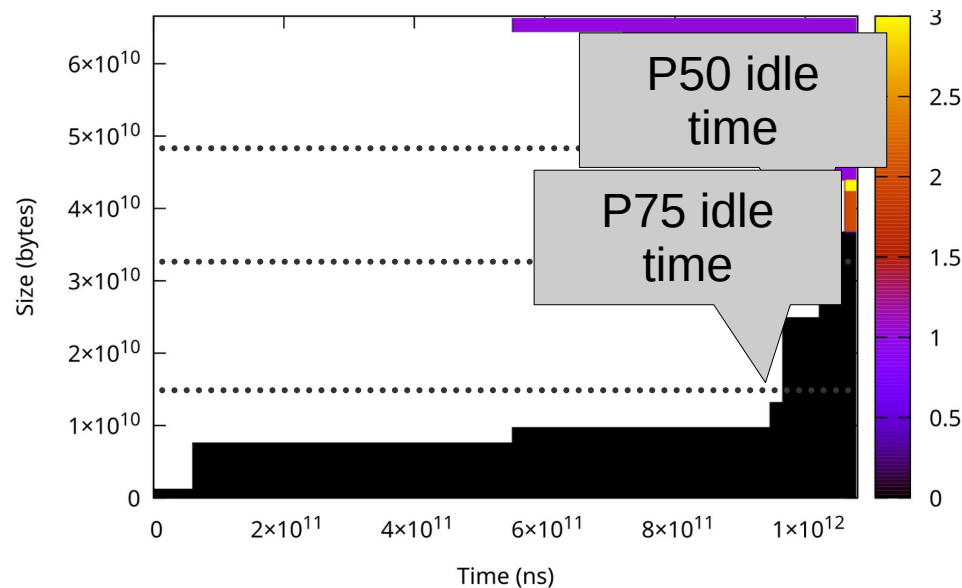
# Idle Time Percentiles: Concept

- Idle time: how long it was not accessed
- Percentile: that of the statistics

Unsorted snapshot



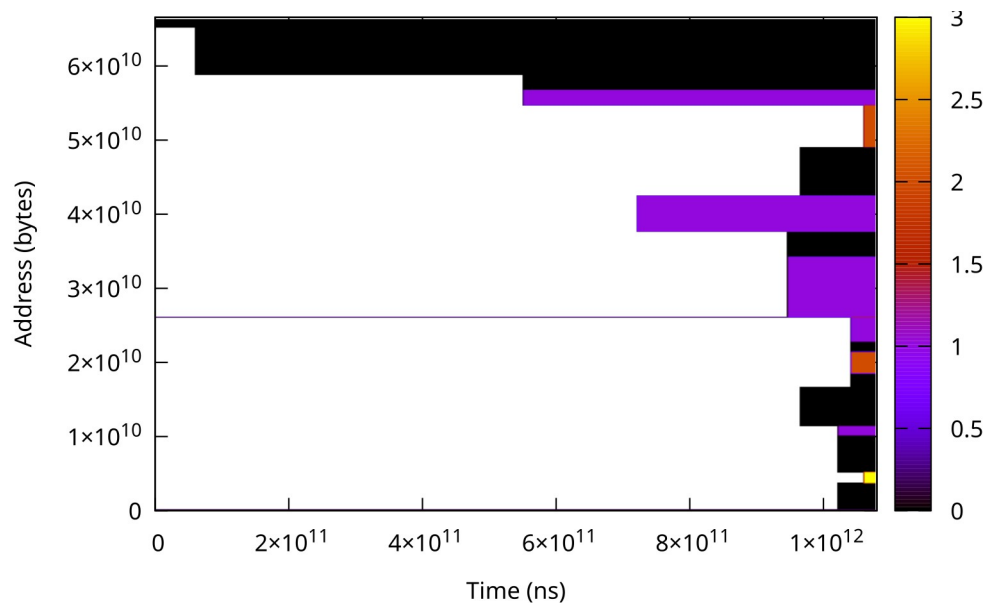
Sorted by access frequency



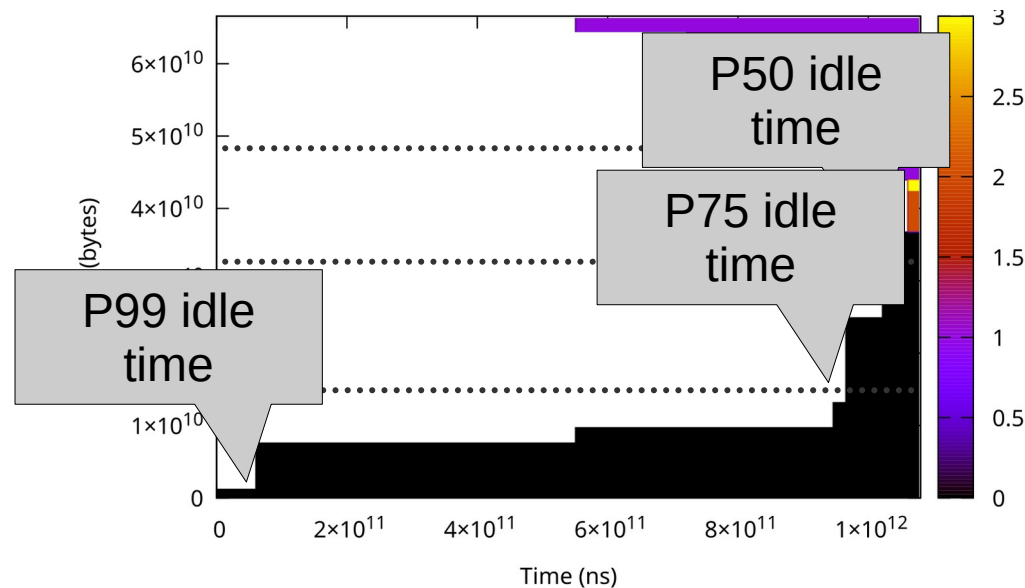
# Idle Time Percentiles: Concept

- Idle time: how long it was not accessed
- Percentile: that of the statistics

Unsorted snapshot

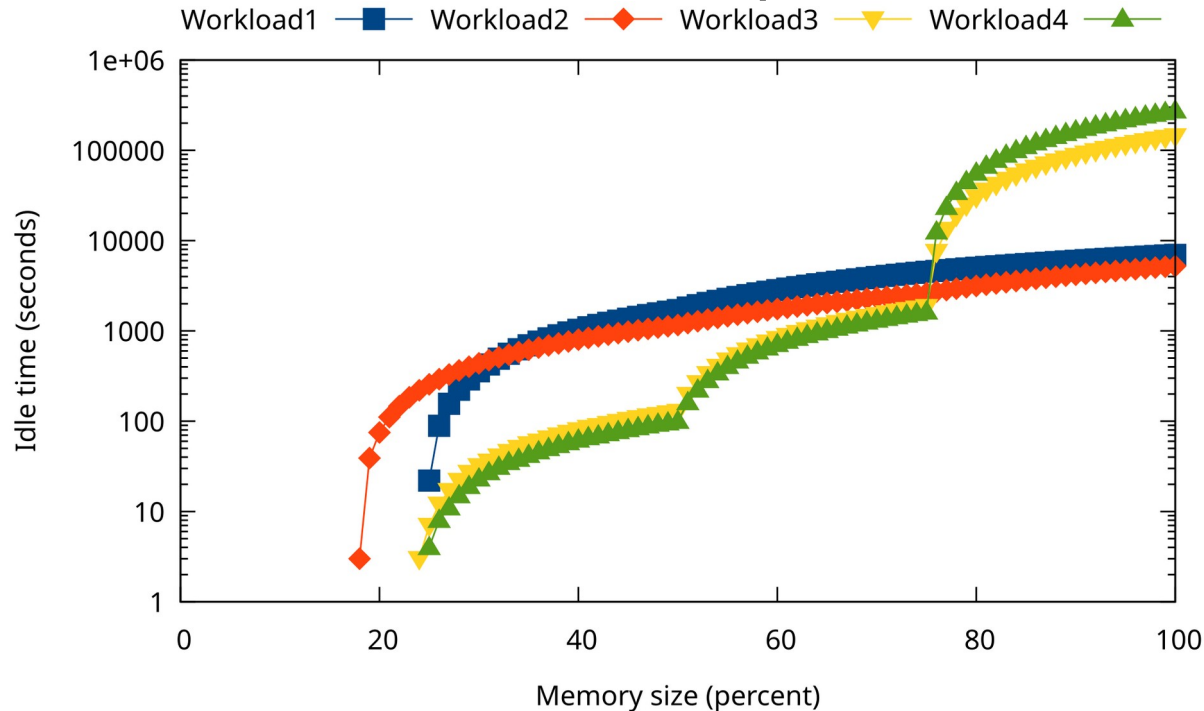


Sorted by access frequency



# Idle Time Percentiles in Real

- Note: Y-axis in logscale
- Show common and different patterns of workloads

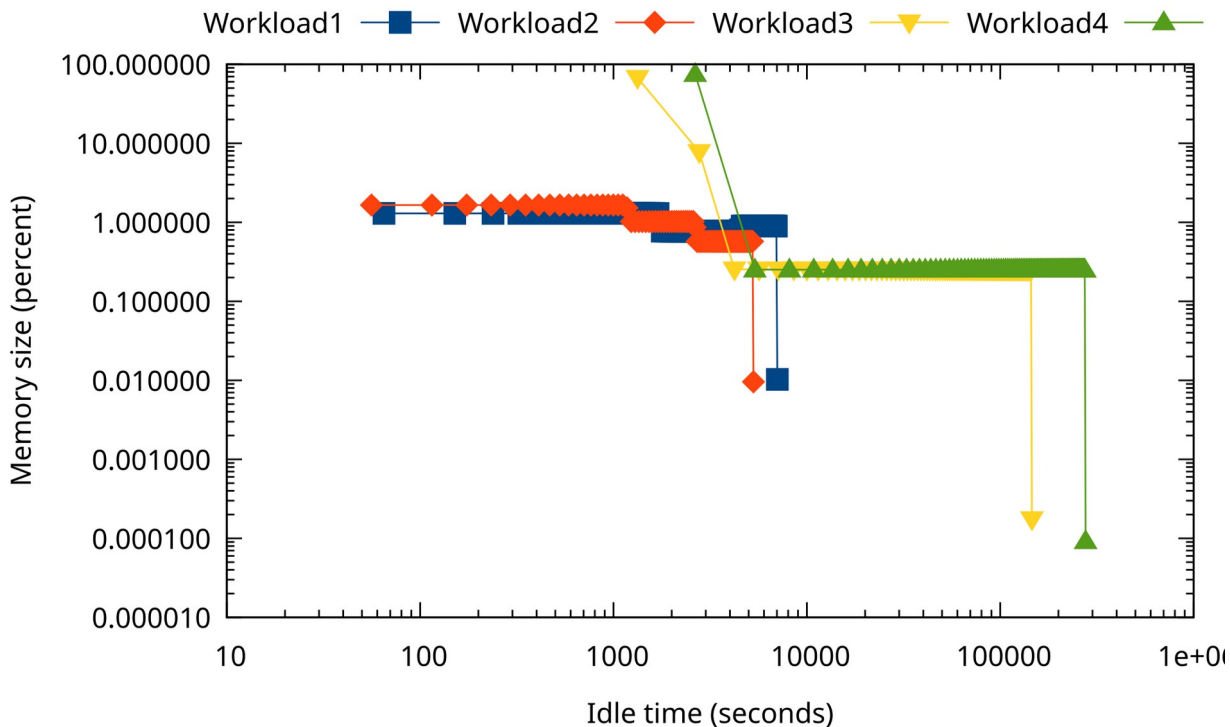


# Cold Memory Tail

- X/Y-axis inversion of idle time percentiles
- X axis: Idle time
- Y axis: Size of memory of the idle time

# Cold Memory Tail in Real

- Note: Both X/Y-axis in logscale



# Estimated Memory Bandwidth

- A region in DAMON snapshot
  - of size 1 GiB
  - that was accessed 10 times per second
- Estimation: 10 GiB/s bandwidth use
- Shouldn't be accurate but best effort

# Formats for Brave People

- Turn on DAMOS-based operations
  - e.g., proactive reclamation, memory tiering
- Hear if someone yells
- Surprisingly most successful use case
- Here is the help: DAMOS stats

# Wrapup

- DAMON provides access information
- There are 2+ ways to show that
  - Maybe good for fans of oysters but snakes



# Discussion Time

- Any of the formats seems useful? Why?
- Any new format suggestions?
- More info or less info?
- What is reasonable “actionable” goal?