

# DAMON-based Pages Migration for {C,G,X}PU [un]attached NUMA nodes

SeongJae Park (SJ) <[sj@kernel.org](mailto:sj@kernel.org)> <[sjpark@crusoe.ai](mailto:sjpark@crusoe.ai)>



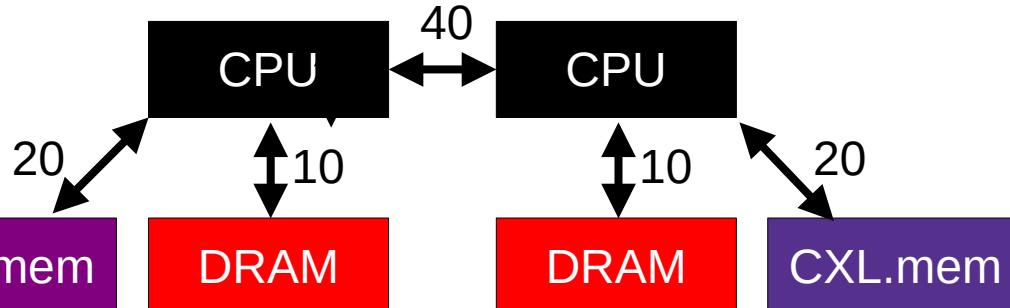
# Table of Contents

- Background (3 mins)
  - NUMA for Dummies
  - DAMON in One Minute
- DAMON for Memory Tiering (4 mins)
- DAMON for Holistic NUMA Migration (8 mins)
- Discussion Time (5 mins)

# Background

# NUMA for Dummies

- Access speed of memory depends on
  - Accessor (CPUs, GPUs, or ?PU),
  - Memory address, etc (e.g., compression rate)



Possible NUMA topology

	CPU1	CPU2
DRAM1	10	50
CXL.mem1	20	60
DRAM2	50	10
CXL.mem2	60	20

Relative access speed  
from CPU to memory

# DAMON in One Minute

- “Subsystem for efficient data access monitoring and access-aware operations”
- Inform monitored accesses
  - Location, Recency, frequency, stability
- DAMOS: Automate access-aware ops
  - e.g., reclaim cold memory

# DAMON for Memory Tiering

# SK Hynix: Capacity Expansion

- Migrate hot pages CXL → DRAM
- Migrate cold pages DRAM → CXL
- Hot/cold threshold: tuned for their case
- A part of SK Hynix HMSDK
- ~94% llama.cpp speedup

# Meta: TPP-DAMON

- DAMON-based **TPP** idea implementation
- Auto-tune hot/cold thresholds for TPP goal
  - Put as many hot data as possible in upper tier
- ~4.42% Taobench score **increase**



# Micron: Dynamic Interleaving

- Change interleaving weights in runtime
  - For dynamic access pattern changes
- ~25% benchmark speedup

# And More

- Non-public success stories and WIP
- Academic researches

# TPP-DAMON in More Details

- The idea is simple and scalable
  - Promote hot pages to the upper node, aiming high utilization of the upper node
  - Demote cold pages to lower node, aiming headroom free space of the node
- Applicable to N tiers

# TPP-DAMON in More Details

```
damo start \
    --numa_node 0 --monitoring_intervals_goal 4% 3 5ms 10s \
        --damos_action migrate_cold 1 --damos_access_rate 0% 0% \
        --damos_apply_interval 1s \
        --damos_quota_interval 1s --damos_quota_space 200MB \
        --damos_quota_goal node_mem_free_bp 0.5% 0 \
        --damos_filter reject young \
    --numa_node 1 --monitoring_intervals_goal 4% 3 5ms 10s \
        --damos_action migrate_hot 0 --damos_access_rate 5% max \
        --damos_apply_interval 1s \
        --damos_quota_interval 1s --damos_quota_space 200MB \
        --damos_quota_goal node_mem_used_bp 99.7% 0 \
        --damos_filter allow young \
        --damos_nr_quota_goals 1 1 --damos_nr_filters 1 1 \
    --nr_targets 1 1 --nr_schemes 1 1 --nr_ctxs 1 1
```

# TPP-DAMON in More Details

```
damo start
  --numa_node 0
    --damos_action migrate_hot 0 --damos_access_rate 5% max \
    --damos_apply_interval 1s \
    --damos_quota_interval 1s --damos_quota_space 200MB \
    --damos_quota_goal node_mem_free_bp 0.5% 0 \
    --damos_filter reject young \
  --numa_node 1 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_action migrate_hot 0 --damos_access_rate 5% max \
    --damos_apply_interval 1s \
    --damos_quota_interval 1s --damos_quota_space 200MB \
    --damos_quota_goal node_mem_used_bp 99.7% 0 \
    --damos_filter allow young \
    --damos_nr_quota_goals 1 1 --damos_nr_filters 1 1 \
    --nr_targets 1 1 --nr_schemes 1 1 --nr_ctxs 1 1
```

Start DAMON in kernel.

Dear DAMON user-space tool,

# TPP-DAMON in More Details

```
damo start \
  --numa_node 0 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_action migrate_cold 1 --damos_access_rate 0% 0% \
    --damos_apply_interval 1s \
    Create one DAMON \
    worker thread, monitoring \
    for node 0 (upper tier) \
    --damos_quota_interval 1s --damos_quota_space 200MB \
    --damos_quota_goal node_mem_used_bp 99.7% 0 \
    --damos_filter allow young \
    --damos_nr_quota_goals 1 1 --damos_nr_filters 1 1 \
  --nr_targets 1 1 --nr_schemes 1 1 --nr_ctxs 1 1
```

Create one DAMON worker thread, monitoring for node 0 (upper tier)

With auto-tuned monitoring intervals.

# TPP-DAMON in More Details

```
damo start \
--numa_node 0 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_action migrate_cold 1 --damos_access_rate 0% 0% \
    --damos_apply_interval 1s \
    --damos_quota_interval 1s --damos_quota_space 200MB \
    --damos_quota_goal node_mem_used_bp 99.7% 0 \
    --damos_filter allow young \
    --damos_nr_quota_goals 1 1 --damos_nr_filters 1 1 \
--nr_targets 1 1 --nr_schemes 1 1 --nr_ctxs 1 1

    And ask it to migrate un-accessed (access_rate 0%)
    memory to node 1 (lower tier), coldest one first.
    IOW, demote cold memory.
```

# TPP-DAMON in More Details

```
damo start \
    --numa_node 0 --monitoring_intervals_goal 4% 3 5ms 10s \
        --damos_action migrate_cold 1 --damos_access_rate 0% 0% \
        --damos_apply_interval 1s \
        --damos_quota_interval 1s --damos_quota_space 200MB \
        --damos_quota_goal node_mem_used_bp 99.7% 0 \
        --damos_filter reject young \
--numa_node 1 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_action migrate_hot 0 --damos_access_rate 5% max \
    --damos_apply_interval 1s \
    --damos_quota_interval 1s --damos_quota_space 200MB \
    --damos_quota_goal node_mem_used_bp 99.7% 0 \
    --damos_filter allow young \
    --damos_nr_quota_goals 1 1 --damos_nr_filters 1 1 \
--nr_targets 1 1 --nr_schemes 1 1 --nr_ctxs 1 1
```

And do that once per second.

# TPP-DAMON in More Details

```
damo start \
--numa_node 0 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_action migrate_cold 1 --damos_access_rate 0% 0% \
    --damos_apply_interval 1s \
    --damos_quota_interval 1s --damos_quota_space 200MB \
    --damos_quota_goal node_mem_free_bp 0.5% 0 \
    --damos_filter reject young \
--numa_node 1 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_quota_space 200MB \
    --damos_quota_goal node_mem_free_bp 0.5% 0 \
    --damos_filter reject young \
--numa_node 2 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_quota_space 200MB \
    --damos_quota_goal node_mem_free_bp 0.5% 0 \
    --damos_filter reject young \
--numa_node 3 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_quota_space 200MB \
    --damos_quota_goal node_mem_free_bp 0.5% 0 \
    --damos_filter reject young \
--nr_targets 1 1 --nr_schemes 1 1 --nr_ctxs 1 1
```

Within *one second*, do the migration only up to *200 MiB* of memory.

# TPP-DAMON in More Details

```
damo start \
    --numa_node 0 --monitoring_intervals_goal 4% 3 5ms 10s \
        --damos_action migrate_cold 1 --damos_access_rate 0% 0% \
        --damos_apply_interval 1s \
        --damos_quota_interval 1s --damos_quota_space 200MB \
        --damos_quota_goal node_mem_free_bp 0.5% 0 \
        --damos_filter reject young \
--numa_node 1 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_quota_goal node_mem_free_bp 99.7% 0 \
    --damos_filter allow young \
    --damos_nr_quota_goals 1 1 --damos_nr_filters 1 1 \
    --nr_targets 1 1 --nr_schemes 1 1 --nr_ctxs 1 1
```

Under the 200MiB/1s hard limitation, auto-tune the real soft limitation to make node 0's free memory ratio becomes 0.5% soon.

# TPP-DAMON in More Details

```
damo start \
    --numa_node 0 --monitoring_intervals_goal 4% 3 5ms 10s \
        --damos_action migrate_cold 1 --damos_access_rate 0% 0% \
        --damos_apply_interval 1s \
        --damos_quota_interval 1s --damos_quota_space 200MB \
        --damos_quota_goal node_mem_free_bp 0.5% 0 \
        --damos_filter reject young \
--numa_node 1 --monitoring_intervals_goal 4% 3 5ms 10s \
    --damos_ac
    --damos_rate_hot 0 --damos_access_rate 5% max \
    --
    --
    Btw, if the page is PG_Young, don't do the migration.
    DAMON might identify young page as not accessed.
    --
    --
    --damos_filter allow young \
    --damos_nr_quota_goals 1 1 --damos_nr_filters 1 1 \
    --nr_targets 1 1 --nr_schemes 1 1 --nr_ctxs 1 1
```

TP

damo

Create another DAMON worker thread that monitors *node 1* (lower tier) with auto-tuned monitoring intervals.

Ask it to migrate memory shown  $\geq 5\%$  access rate to *node 0* (upper tier), hottest first. IOW, **promote** hot pages.

Do the migration once per second, up to 200 MiB of memory.

Under the 200 MiB/s hard limit, adjust the real soft limit aiming *node 0* memory utilization ratio 99.7%.

Btw, to that only if the page is PG\_Young.

```
--damos
--damos_node_mem_tree_bp 0.5% 0 \
--damos_filter reject young \
--numa_node 1 --monitoring_intervals_goal 4% 3 5ms 10s \
--damos_action migrate_hot 0 --damos_access_rate 5% max \
--damos_apply_interval 1s \
--damos_quota_interval 1s --damos_quota_space 200MB \
--damos_quota_goal node_mem_used_bp 99.7% 0 \
--damos_filter allow young \
--damos_nr_quota_goals 1 1 --damos_nr_filters 1 1 \
--nr_targets 1 1 --nr_schemes 1 1 --nr_ctxs 1 1
```

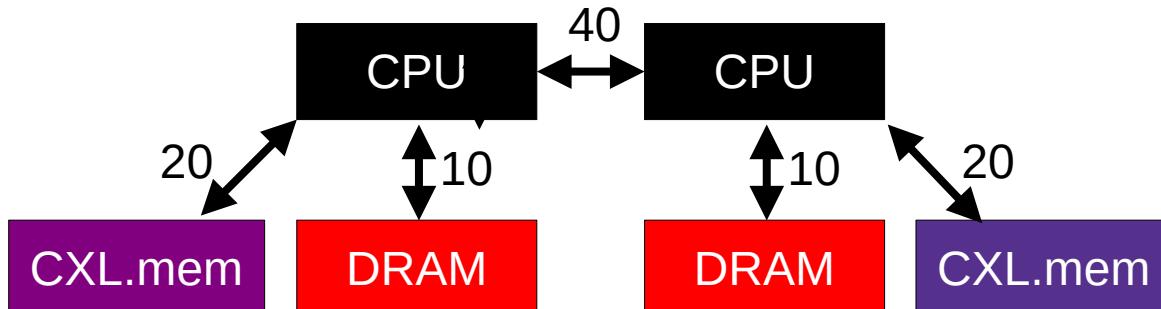
# DAMON for Holistic NUMA Page Migration

# Idea is Simple

- Tiering: special case NUMA management
  - All CPU [un]attached nodes
  - Single promotion/demotion path
- General NUMA: N Accessors
  - CPU, GPU, \*PU
  - N promotion/demotion paths
  - Idea: run accessor-aware TPP for each path

# Idea in Example

- TPP1 migrate CPU1-hot/cold memory
  - On DRAM1 → CXL.mem1 → DRAM2 → CXL.mem2 hierarchy
- TPP2 migrate CPU2-hot/cold memory
  - On DRAM2 → CXL.mem2 → DRAM1 → CXL.mem1 hierarchy



Possible NUMA topology

	CPU1	CPU2
DRAM1	10	50
CXL.mem1	20	60
DRAM2	50	10
CXL.mem2	60	20

Relative access speed  
from CPU to memory

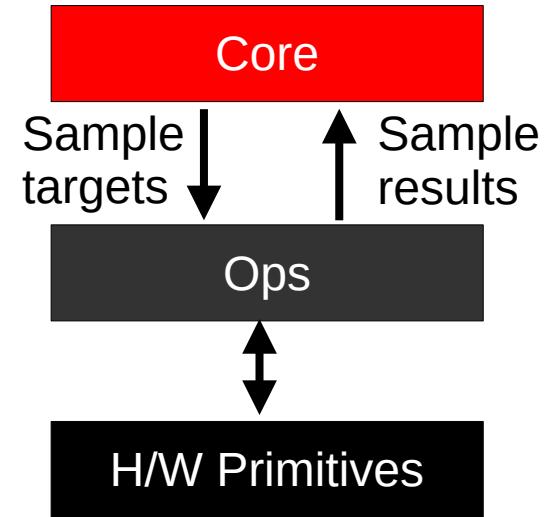
# Can DAMON Do That for Me?

- No, since it cannot do per-CPU monitoring
- DAMON is **willing** to do that in future

# Required DAMON Changes for Holistic NUMA Page Migration (Or, per-CPUs Monitoring)

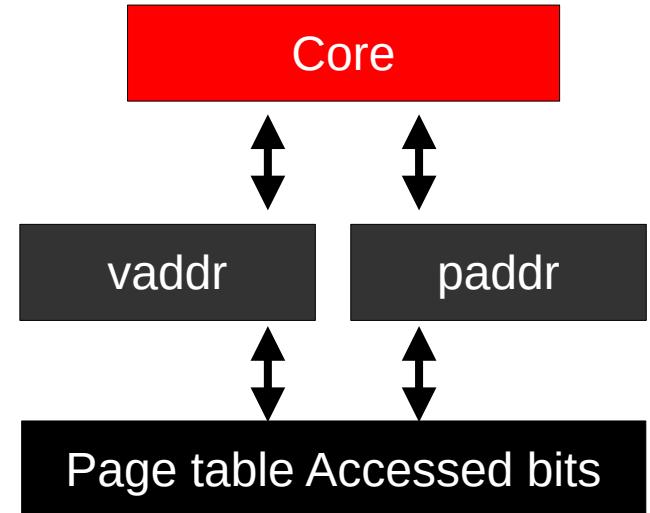
# DAMON Core and Ops Layers

- Core: For pure software logic
  - Adaptively find pages to check access for
  - Assemble small samples to accurate pattern
- Ops: For h/w specific primitives
  - Check if core-specified samples got accessed
  - Using their favorite primitives
  - DAMON callers can implement their own ops



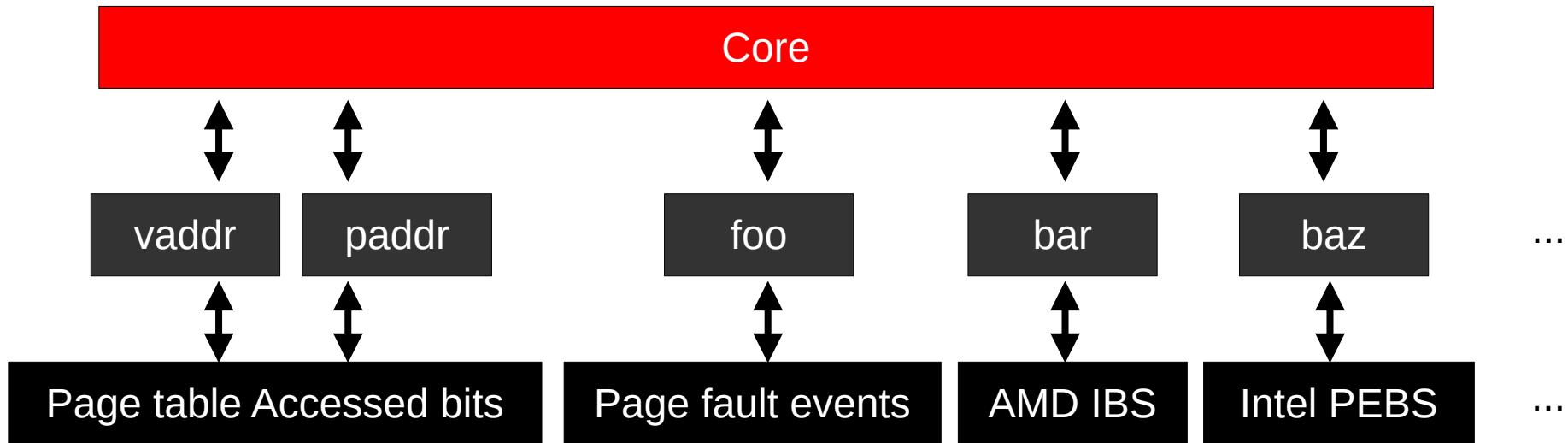
# Existing DAMON Op Impls

- ‘vaddr’ and ‘paddr’
  - For virtual/physical address spaces monitoring
  - Both use page table Accessed bits
    - Lacks accessor CPU info



# How Can We Make It?

- Add new ops for appropriate primitives
  - e.g., Page fault events or instruction sampling



# Can It Be Upstreamed?

- New op: good for PoC but Maintenance
- Too many ops having duplications
- Upstream-aimed work in progress

# damon\_report\_access()

- Let any kernel components  
(e.g., page fault handler, GPU driver, etc)  
report their observed access information  
to DAMON core  
with single function call

# DAMON Sample Control

- API/ABI for setup of primitives to use
- API/ABI for filtering sampling results based on additional info

```
# damo start --sample_primitives page_fault \
             --sample_filter allow cpu 0-4 \
             --sample_filter allow threads 777,888,999 \
             --sample_filter reject write
```

# Progress

- Per-CPUs/threads/read/write patchset
  - RFC v3 is [posted](#)

# TODO

- Upstream per-CPUs monitoring
- Test damo-based holistic NUMA migration
- Add simple-interface kernel module  
(say, DAMON\_NUMA\_MIGRATE)

# Challenges

- Page fault handling part change
  - Need alignment between stakeholders
    - NUMA balancing
    - Page fault handling
    - DAMON
  - Aim to make progress by LSFMMBPF'26

# Wrapup

- DAMON was useful for specific NUMA case
  - Memory tiering
- TPP-DAMON can be extended
  - For holistic NUMA cases
- Works in progress, challenges exist

# Discussion Time!

- Feel free to continue after this session on
  - Hallway
  - Mailing lists
  - DAMON Beer/Coffee/Tea [chats](#)
  - [sj@kernel.org](mailto:sj@kernel.org)