The International ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC), Minneapolis, Minnesota, United States, June 27 – July 1, 2022

DAOS: Data Access-aware Operating System

SeongJae Park — amazon
Madhuparna Bhowmik — University of Illinois Urbana-Champaign
Alexandru Uta — amazon

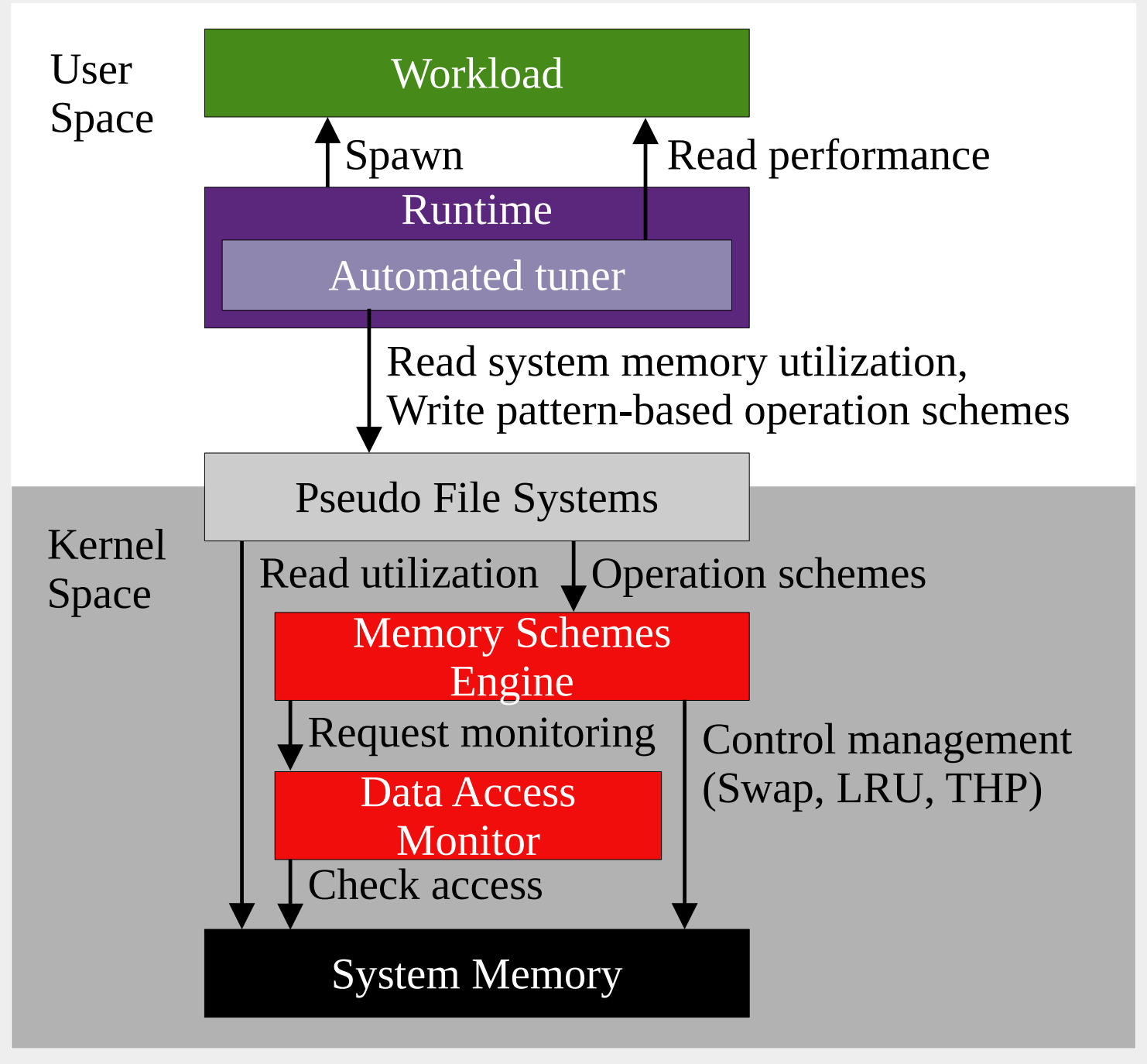
Abstract

In data-intensive workloads, data placement and memory management are inherently difficult: the programmer and the operating system have to choose between (combinations of) DRAM and storage, replacement policies, as well as paging sizes. Efficient memory management is based on fine-grained data access patterns driving placement decisions. Current solutions in this space cannot be applied to general workloads and production systems due to either unrealistic assumptions or prohibitive monitoring overheads.

To overcome these issues, we introduce DAOS, an open-source system for general data access-aware memory management. DAOS provides a data access monitoring framework that utilizes practical best-effort trade-offs between overhead and accuracy. The memory management engine of DAOS allows users to implement their access-aware management with no code, just simple configuration schemes. For system administrators, DAOS provides a runtime system that auto tunes the schemes for user-defined objectives in a finite time. We evaluated DAOS on commercial service production systems as well as state-of-the-art benchmarks. DAOS achieves up to 12% performance improvement and 91% memory saving. DAOS is upstreamed and available in the Linux kernel.

We Need Data Access-aware Operating System

- Because DAOS can predict future memory usage better
- Because it helps making better data management decision
- Because it can improve memory efficiency and performance
- Because DRAM is a major infrastructure expense


User Space
Workload
Spawn
Read performance
Runtime
Automated tuner
Read system memory utilization,
Write pattern-based operation schemes
Pseudo File Systems
Kernel Space
Read utilization
Operation schemes
Memory Schemes Engine
Request monitoring
Control management
(Swap, LRU, THP)
Data Access Monitor
Check access
System Memory

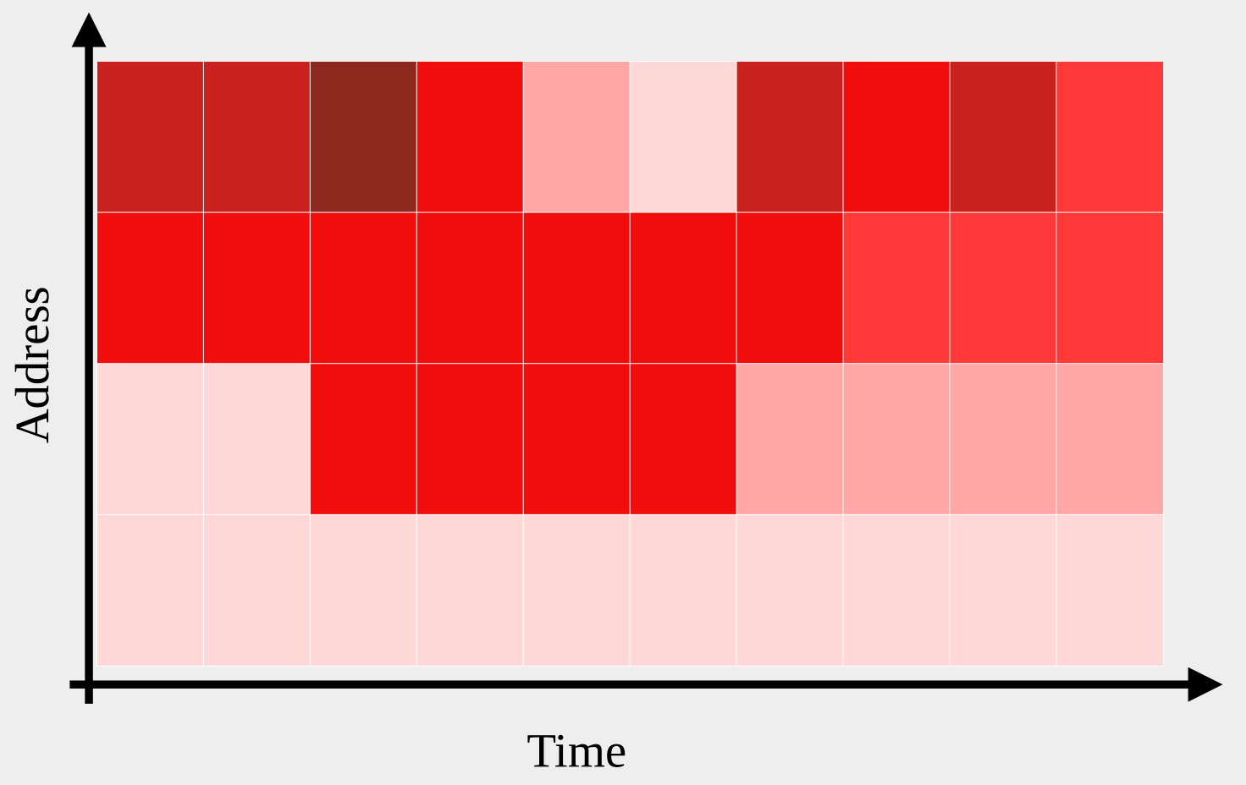
DAMOS: DAMON-based Operation Schemes

- DAMON-based memory management schemes engine for DAOS
- Receives ‘schemes’; each scheme is constructed with
  - Target access pattern: ranges of size, access frequency, and age
  - 1 memory management action
    - Currently supported actions include: `WILLNEED, COLD, PAGEOUT, HUGEPAGE, NOHUGEPAGE`
- DAMOS automatically finds the memory region of the target pattern from DAMON results 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
```

DAMON: Data Access MONitor

- Data access monitoring component for DAOS
  - Provides access frequency of each address range
  - DAOS gets data access patterns of the system online using this component



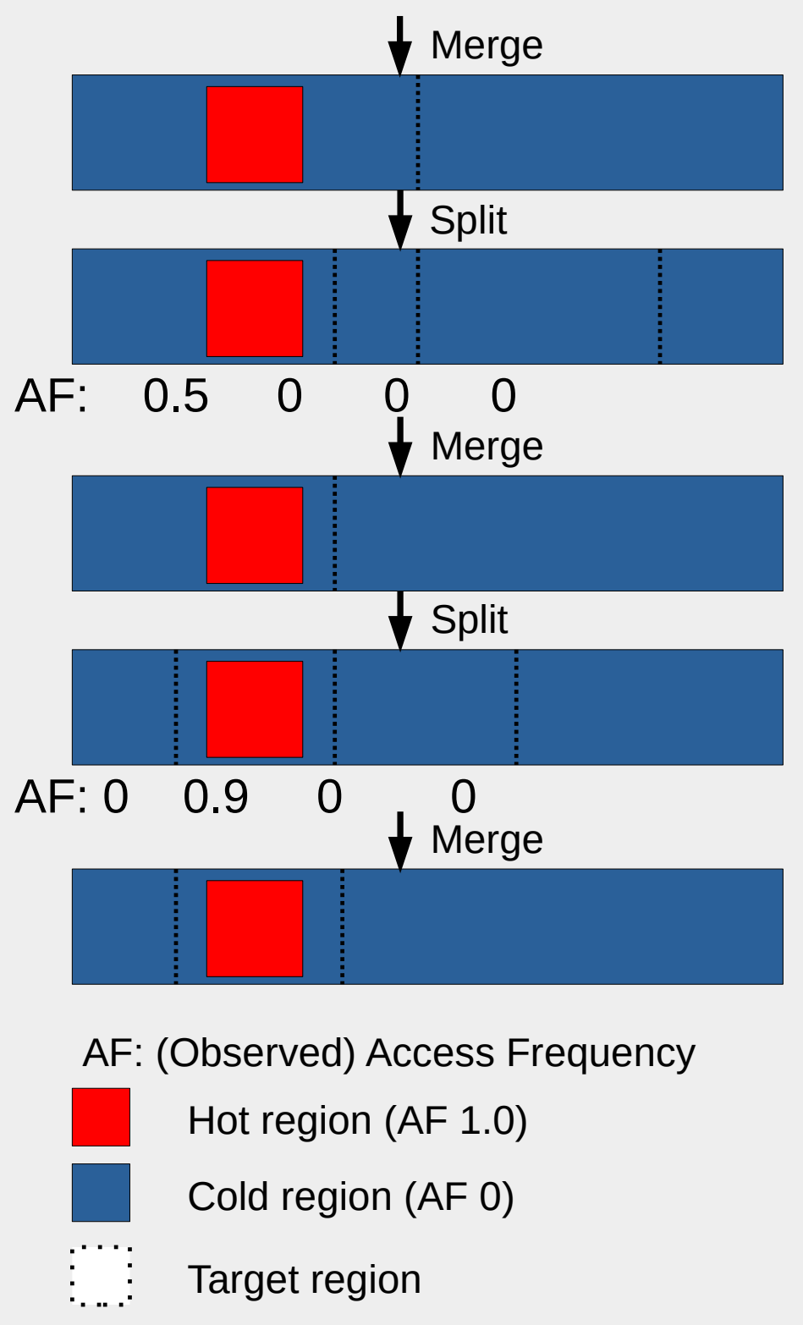
- To be used for DAOS, DAMON needs to fulfill below requirements
  - Accuracy: The monitoring result should be useful for DRAM level MM
  - Overhead: Should light-weight enough for online monitoring
  - Scalability: The upper-bound overhead should be controllable regardless of the size of the monitoring target systems and workloads
- DAMON fulfills the requirement in below steps
  - Straightforward Access Monitoring (collect basic information)
  - Region-based Sampling (optimize overhead)
  - Adaptive Regions Adjustment (make best-effort accuracy)

Region-based Sampling

- Defines data objects in access pattern oriented way
  - “A data object is a contig memory region that all page frames in the region have similar access frequencies”
  - By the definition, if a page in a region is accessed, other pages of the region has probably accessed, and vice versa
  - Thus, checks for the other pages can be skipped
- By limiting the number of regions, we can control the monitoring overhead regardless of the target size
- However, the accuracy will degrade if the regions are not properly setp

Adaptive Regions Adjustment

- Starts with minimum number of regions covering entire target memory areas
- For each aggregation interval,
  - merges adjacent regions having similar access frequencies to one region
  - Splits each region into two (or three, depend on state) randomly sized smaller regions
  - Avoid merge/split if the number of regions might be out of the user-defined range
- If a split was meaningless, next merge process will revert it (vice versa)
- In this way, we can let users control the upper bound overhead while preserving minimum and best-effort accuracy

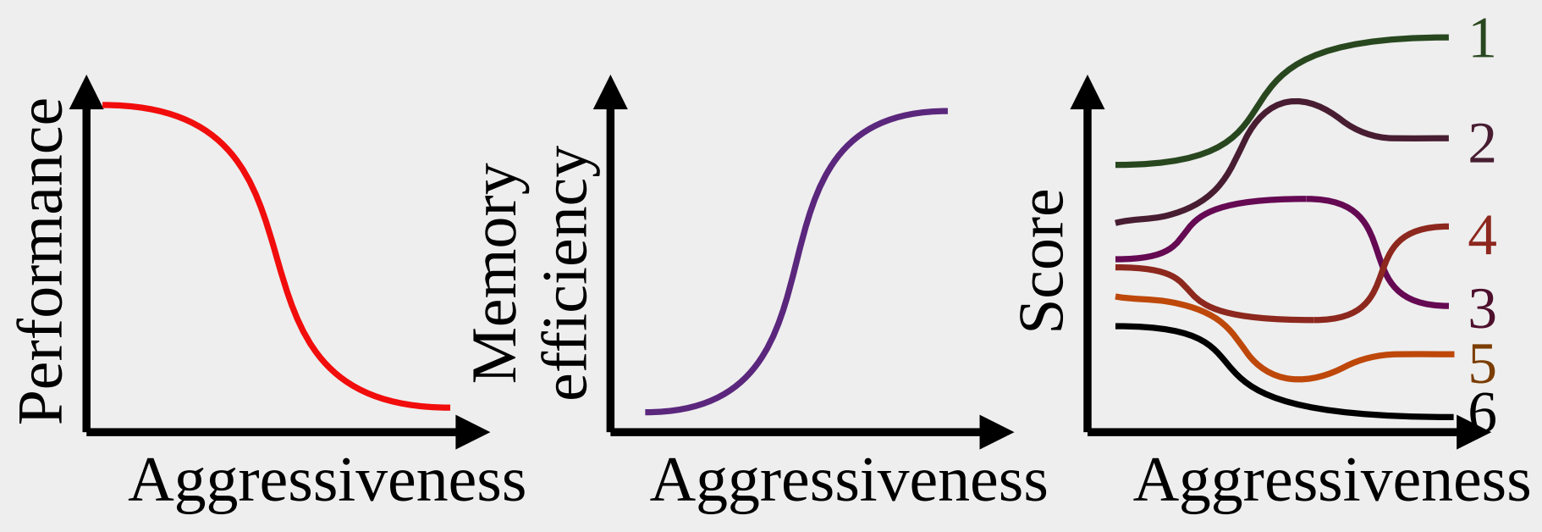


AF: (Observed) Access Frequency
🟥 Hot region (AF 1.0)
🟦 Cold region (AF 0)
⬚ Target region

DAMOOS: DAMON-based Optimal Operation Schemes (Auto-tuning Runtime for DAOS)

Simplifying The Problem

- We care only memory efficiency and performance at last
  - These can be consolidated into one metric (score) with different priorities
  - Giving the priorities could be easy for users (depends on users’ SLO)
- The target access pattern is only the aggressiveness of the scheme
- The multi-dimension search space can be reduced to 2-dimension
  - Aggressiveness as X-axis, Score as Y-axis
- We can further expect six simple patterns in common cases

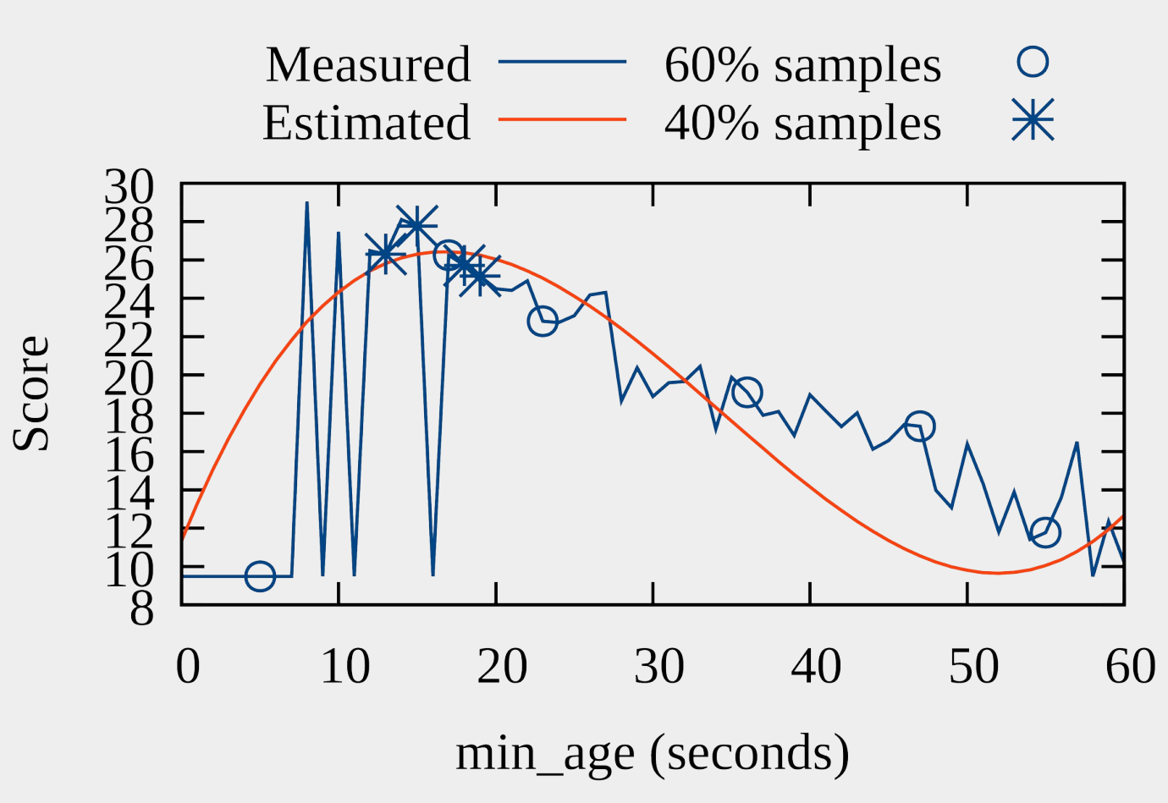


Sampling

- Calculate how many times we can measure the score for different aggressiveness (‘nr_samples’)
  - The user-specified tuning time limit divided by the unit work time
- Run the workload with 60% of ‘nr_samples’ schemes having random aggressiveness and measure one score for each scheme
- Run the workload with 40% of ‘nr_samples’ schemes having random but near to the best of the 60% sample results aggressiveness

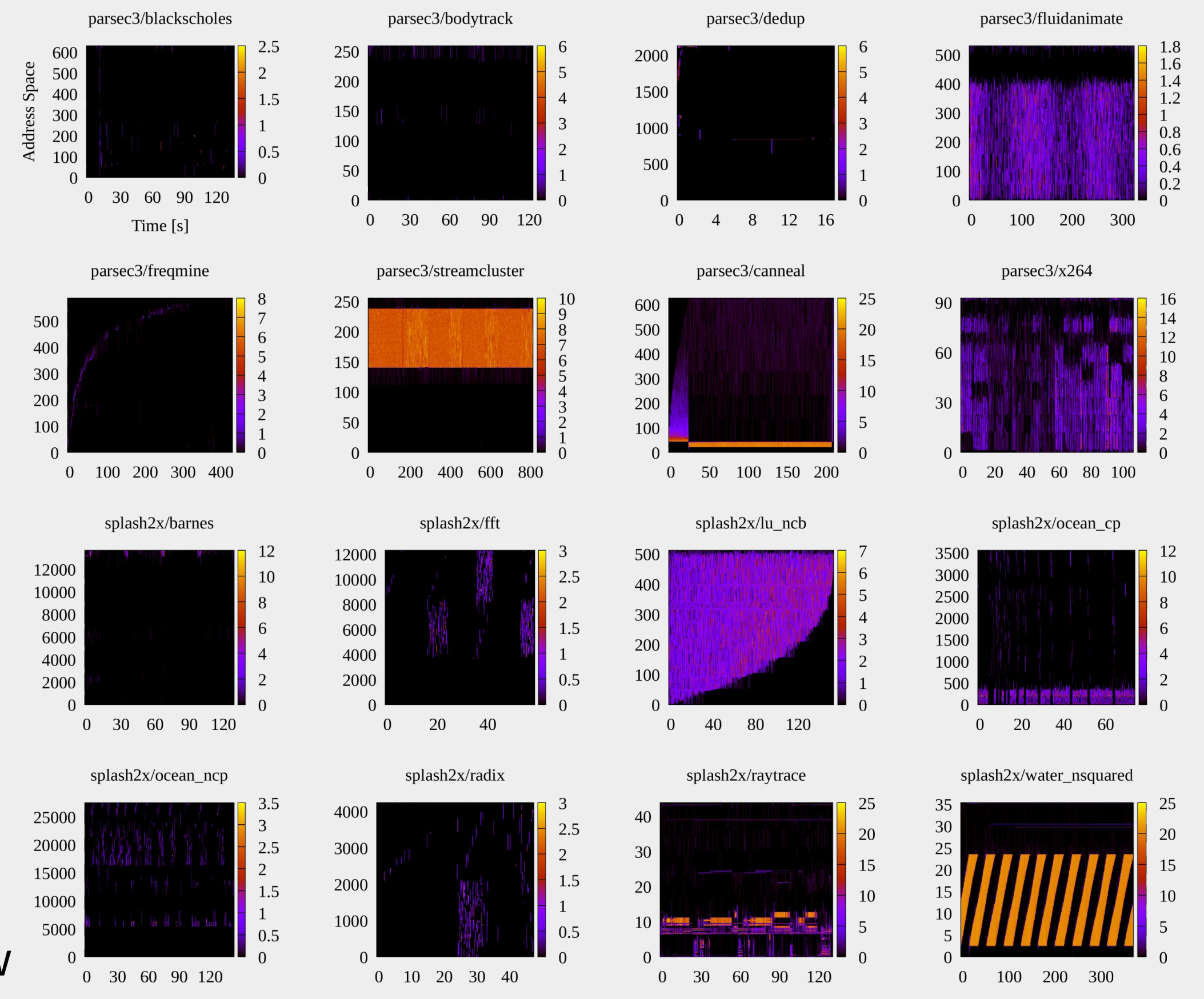
Estimation and Selection

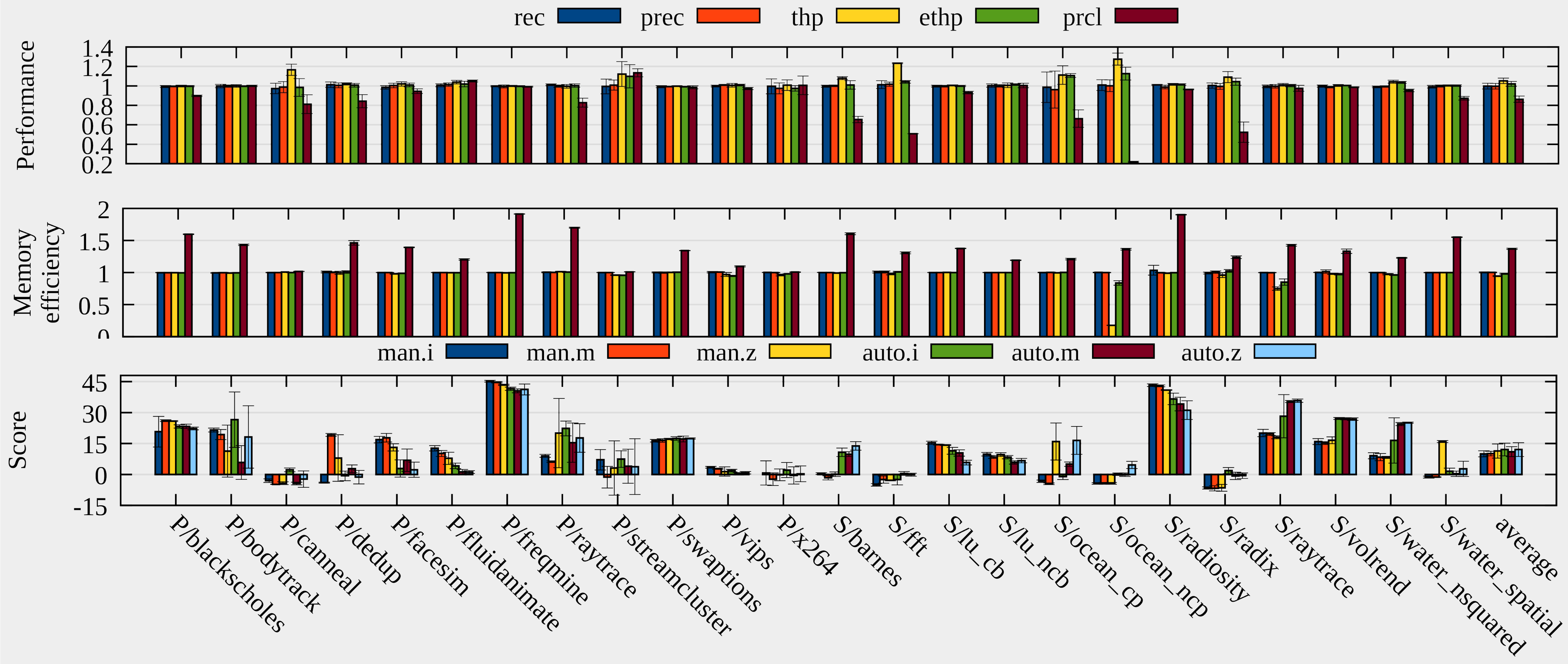
- Find the relationship between the aggressiveness and score by applying Polynomial curve fitting to the ‘nr_samples’ data points
- On the curve, we find an aggressiveness value that generates maximum score and use it as the best scheme aggressiveness



Evaluation

- DAMON consumes <=2% single CPU time for all cases
- DAMON monitoring results that visualized in heatmap format shows reasonable results
- ‘ethp’ applies THP for regions having >=5% access frequency and applies regular pages for regions >=2MB and not accessed for >=7 seconds
  - On average, preserves 39% of ‘thp’ speedup while removing 64.28% of ‘thp’ memory overhead
- ‘prcl’ pages out regions not accessed for >=5 seconds
  - On average, saves 37.10% memory with 13.66% slowdown
- Auto-tuning runtime obtains score improvements of 20.02%, 6.16%, and 6.25% on three different types of h/w





DAOS is OpenSource

- Kernel parts of DAOS are merged in the Linux; User-space parts of DAOS are available at https://github.com/damonitor
- Visit https://damonitor.github.io for quick start