

Starfish: Supplementary Material

Yanwen Xia^{1,2}, Benyong Deng³, Hong Huang^{1,2}, Yuzheng Wang⁴, Quanxi Li^{1,2},
Xingda Wei⁵, Xiaobing Feng^{1,2}, Huimin Cui^{1,2}, Chenxi Wang^{1,2}

¹SKLP, Institute of Computing Technology, CAS

²University of Chinese Academy of Sciences ³Huazhong University of Science and Technology

⁴SCS Peking University ⁵Shanghai Jiao Tong University

1 Clean Evictions Are Common

We first measure how often evicted objects remain clean after being fetched under Non-FT in our evaluation. A clean eviction means the object has not been modified since its last fetch, so a valid retained remote copy can serve as its recovery path without rebuilding protected state. Table 1 shows that clean evictions are common in our evaluated workloads, especially read-dominated ones: 4 of 8 workloads have clean-eviction ratios above 90%. This means many evictions do not inherently need new data and parity writes; existing FT systems pay this cost because they discard or invalidate the previous remote state on fetch.

2 Fine-Grained Accesses Are Common

Carbink’s span-based design is well matched to workloads whose memory footprint is dominated by large objects. However, recent datacenter applications also contain many fine-grained-object workloads. We run two representative datacenter benchmark suites, DCPerf [2] and TailBench [1], to demonstrate this. Figure 1 reports access amplification: for each fetched page, we measure how many bytes are actually used by the application. By this metric, 4 of 6 DCPerf applications and 5 of 8 TailBench applications use only a small fraction of fetched pages, so page/span-granularity movement fetches many unnecessary bytes.

Figure 2 reports object size on the x-axis and the cumulative fraction of total memory on the y-axis. A point at size x means that objects no larger than x account for that fraction of the application’s memory footprint. This metric approximates the best case for Carbink’s hot/cold span construction: even if Carbink perfectly groups objects with similar hotness before eviction, an object smaller than a span can still reside in remote memory and force span-granularity reads or updates. The figure shows that 3 of 6 DCPerf applications and 4 of 8 TailBench applications place most of their memory footprint in objects smaller than a page, leaving room for read and write amplification even under optimal span packing.

Table 1. Clean-eviction ratio across evaluated workloads.

Workload	BFS	LLM	MG	WC	KV-B	KV-A	KV-S	NQ
Clean eviction	95.1%	97.3%	44.2%	16.8%	94.6%	49.3%	4.7%	98.1%

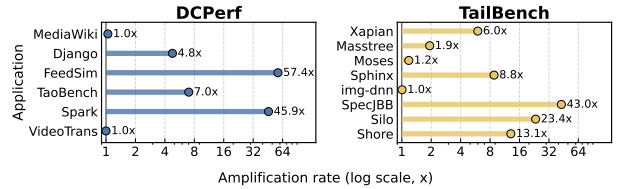


Figure 1. Access amplification in modern datacenter benchmark suites. Many applications use only a small fraction of each fetched page, exposing a mismatch between remote-access granularity and useful data.

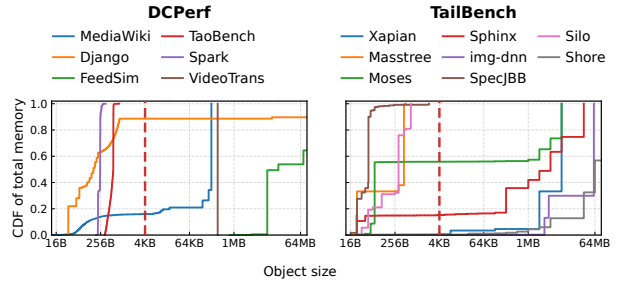


Figure 2. Cumulative memory footprint by object size. For each object-size threshold, the CDF reports the fraction of the application’s total memory footprint contributed by objects no larger than that threshold.

References

- [1] Harshad Kasture and Daniel Sanchez. 2016. Tailbench: a benchmark suite and evaluation methodology for latency-critical applications. In *2016 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 1–10.
- [2] Wei Su, Abhishek Dhanotia, Carlos Torres, Jayneel Gandhi, Neha Gholkar, Shobhit Kanaujia, Maxim Naumov, Kalyan Subramanian, Valentin Andrei, Yifan Yuan, and Chunqiang Tang. 2025. DCPerf: An Open-Source, Battle-Tested Performance Benchmark Suite for Datacenter Workloads. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*. 1717–1730. doi:10.1145/3695053.3731411