Precise Control of Page Cache for Containers

Kun WANG, Song WU, Shengbang LI, Zhuo HUANG, Hao FAN, Chen YU, Hai JIN

Frontiers of Computer Science, DOI: 10.1007/s11704-022-2455-0

Problems & Ideas

- Problems of page cache management for containers:
  - Containers can utilize more memory than limited by their cgroup, effectively breaking memory isolation.
  - The OS kernel has to evict page cache to make space for newly-arrived memory requests, slowing down containerized applications.
- Ideas: pCache divides the page cache of containers into private and shared, then controls both kinds of page cache separately and precisely.

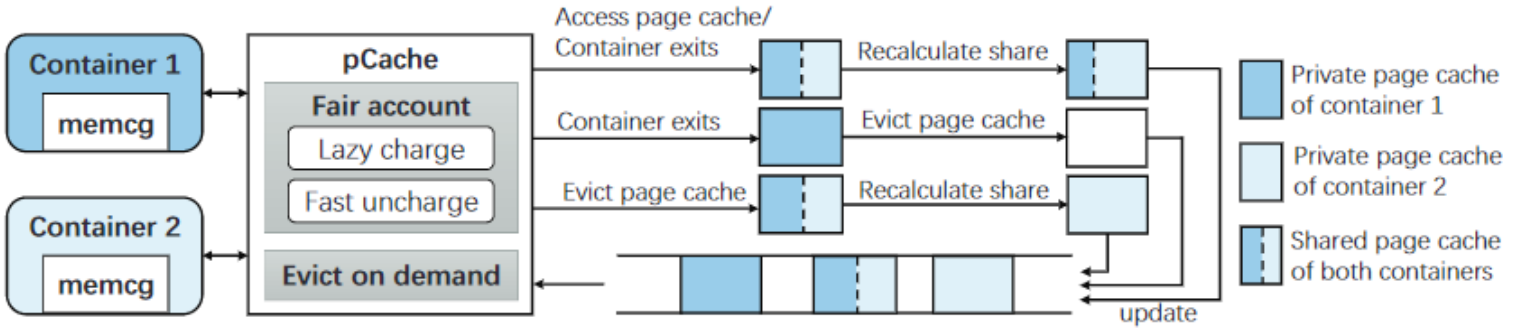

Figure 1 pCache architecture

Split the shared page cache charging based on per-container share to prevent containers from using memory for free;
Charge the per-container share lazily and randomly, preventing attackers from inferring the real-time memory usage;
Reduce unnecessary page cache evictions to avoid the performance impacts.

Main Contributions

- Contributions:
  - An empirical study that shows isolation problems of page cache and reveals the root causes and the impacts on containerized applicaitons;
  - A pricise page cache management scheme to to prevent containers from using memory for free by splitting the shared page cache charging based on per-container share and avoid the performance impacts by reducing unnecessary page cache evictions;
  - Experimental results show that pCache can account for page cache fairly, enhance memory isolation effectively, and achieve performance improvement over the original page cache management policy.

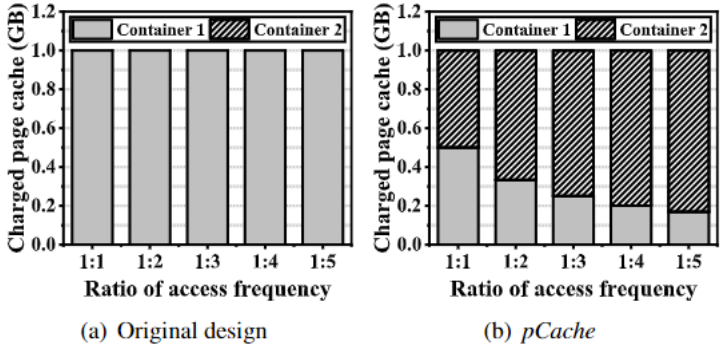

Figure 2 Isolation enhancement

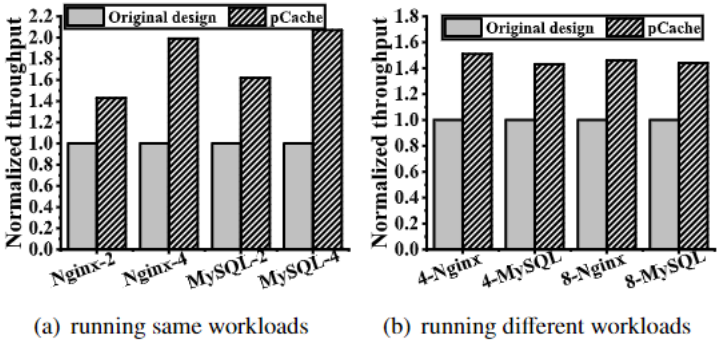

Figure 3 Performance improvement

Page cache isolation and overall performance results yielded by pCache versus original design. Left: Charged page cache of containers under various ratios of page cache access frequency; Right: Normalized throughput of Nginx and MySQL when running same and different workloads in various number of containers (higher is better).