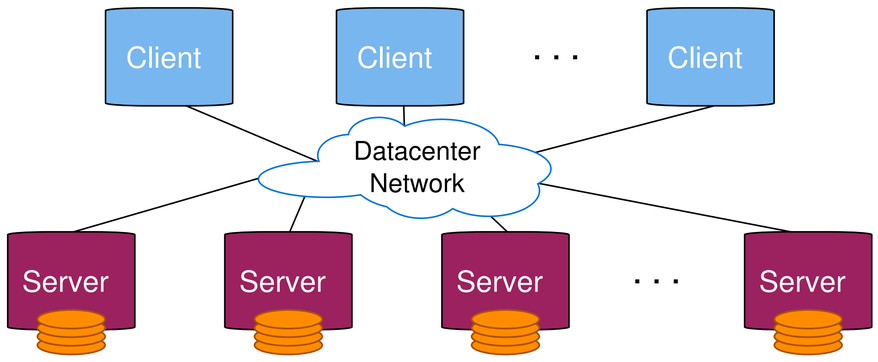
Log-structured Memory for DRAM-based Storage

Steve Rumble, Diego Ongaro, Ryan Stutsman, Ankita Kejriwal, Mendel Rosenblum, John Ousterhout

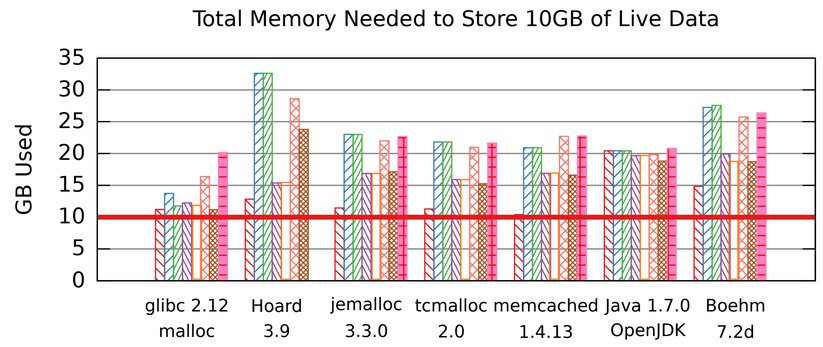
RAMCloud Overview



- Datacenter storage system
- All data stored in DRAM at all times
- *Low-latency*: 5 - 10µs small RPCs across datacenter
- Large scale: 1,000 - 10,000 servers
- *Goal*: Enable novel applications with 100 - 1,000x decrease in storage latency / increase in ops/second

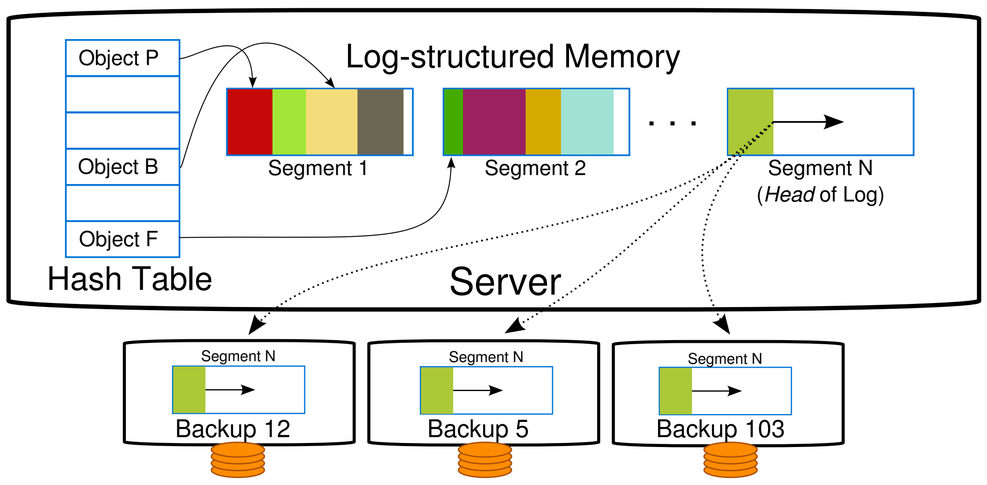
Problem with Conventional Memory Allocators

- 50% of RAMCloud hardware cost is DRAM, but
- Existing memory allocators don't use memory efficiently, particularly when access patterns change:



- Non-copying allocators (e.g. malloc) suffer fragmentation
- Language-based copying garbage collectors waste memory to improve performance
- How to get high memory utilization and high performance?
  - Exploit restricted use of pointers in storage systems

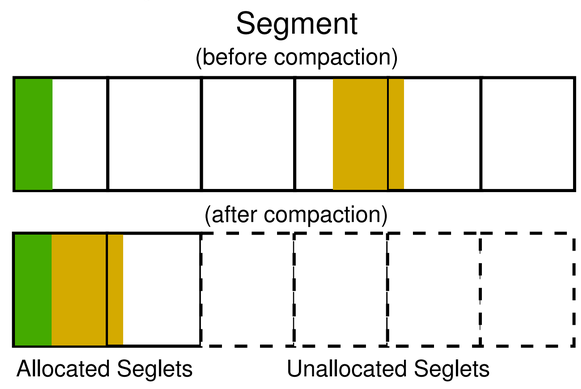
RAMCloud Structure: Pervasive Log



- Memory treated as large contiguous array: a *log* structure
  - New/updated objects append to end, replicated on backups
- Log is *cleaned* (defragmented) to reclaim dead object space
- Hash table provides fast map from key to object in log
  - One pointer to check object liveness when cleaning
  - One pointer to update when object is relocated

Two-Level Cleaning

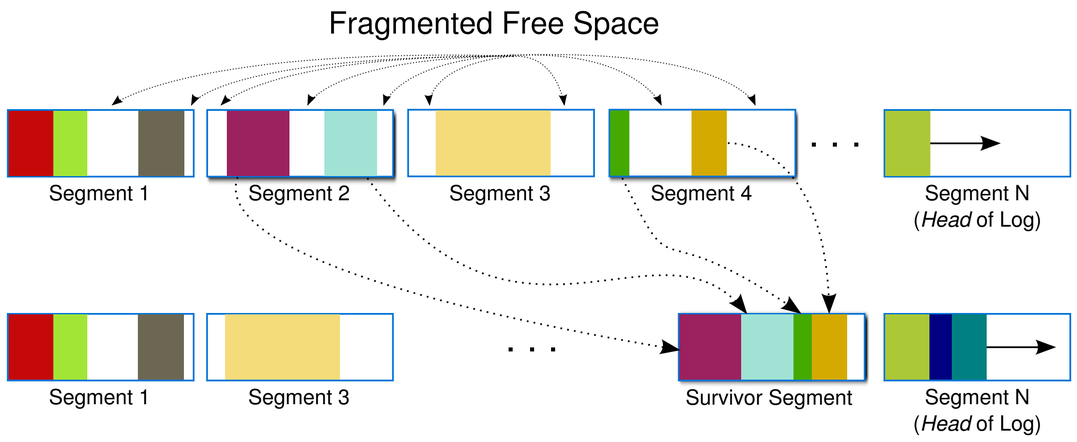
- Level 1: Compact segments in memory (no cleaning on disk)
- Level 2: Clean segments on disk (see "Parallel Log Cleaning")
- 30-100x more memory bandwidth than net/disk bandwidth



- Divide segments into *seglets*, cleaning compacts segments
- Play to strengths/weaknesses of each medium
  - RAM: High bandwidth, low capacity, local access
  - Disk: Low bandwidth, high capacity, remote access
- More segments with fewer live objects = cheap disk cleaning

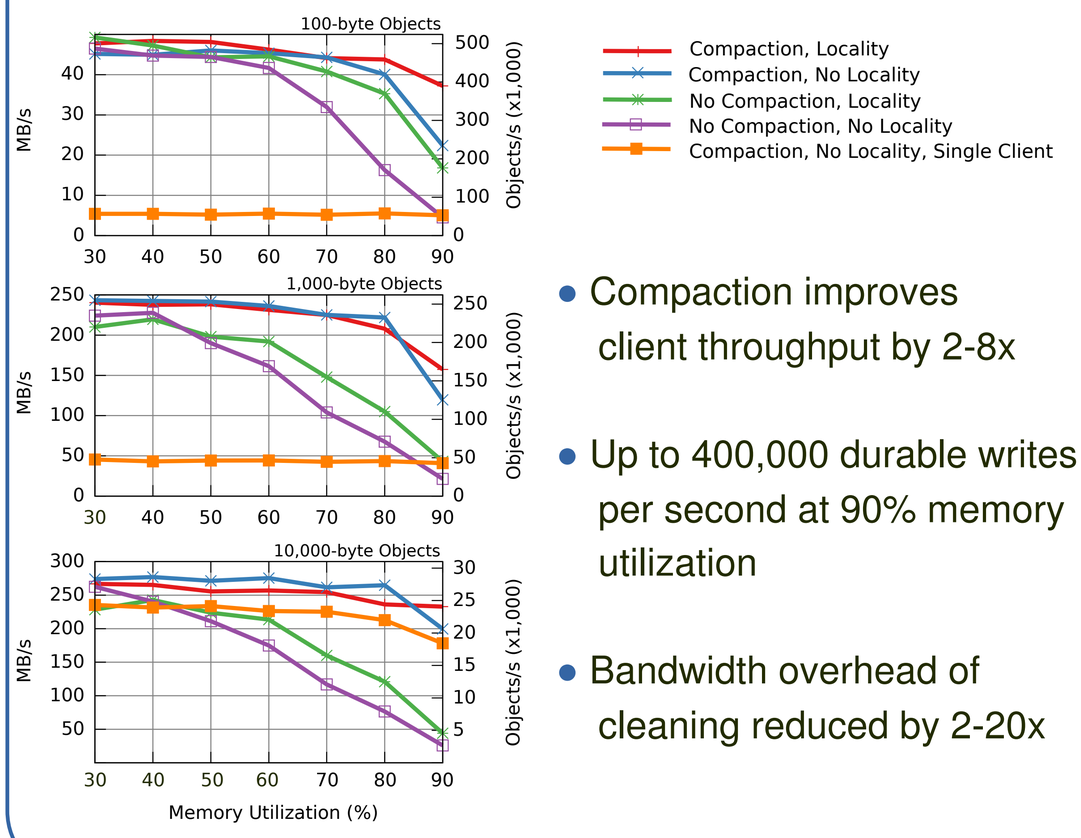
Parallel Log Cleaning

- Fragmented space reclaimed in parallel w/ normal operation
  - Cleaner defragments space freed by deletes & overwrites
- Reclaims space by coalescing live data, writing to new survivor segments, freeing cleaned segments for head of log



- Segments 2 and 4 cleaned into single survivor segment
- Memory for 2 and 4 returned to free list (for future log heads)

Client Write Performance



- Compaction improves client throughput by 2-8x
- Up to 400,000 durable writes per second at 90% memory utilization
- Bandwidth overhead of cleaning reduced by 2-20x