Adaptive Write-Back Destaging for Server-Side Caches

Gokul Nadathur, Venkatesh Kothakota, Chethan Kumar,
Mahesh Patil, Karthik Dwarakinath, Vanish Talwar
PernixData, Inc.

Problem Statement: We consider a virtualized server side storage acceleration architecture wherein IOs from each host are accelerated by a local cache in write back mode [1]. Each host is connected to a variety of NAS and SAN storage backends with different performance capabilities. Writing dirty data to backend storage is done by the destager by batching a fixed number of IOs. Since reads are mostly cached, this fundamentally alters the workload to the storage from mixed read writes to a purely bursty write biased workload of varying IO sizes. Maintaining a constant batch size in the destager that is agnostic of the actual capability of the storage backend results in sub-optimal write throughput accompanied by high latencies. The problem is further compounded when multiple destagers in the cluster write to the same shared storage. Finally, every destager IO also involves reading from an acceleration tier the behavior of which should also be factored.

Adaptive Destager: We address these problems by designing an adaptive destager that optimizes destager write throughput and latency in a diverse storage environment. The scheme also works with multiple destagers in the system without any explicit state sharing. Specifically, we introduce an adaptation module to the destager consisting of (i) a monitoring component to collect throughput and latency stats of the flash device and backend storage. These are fed into (ii) an adaptor component that applies a learning algorithm to dynamically pick the optimal batch size of the IOs issued by the destager. This optimal size is then enforced by (iii) a controller that actuates the batch size in the overall destager workflow. The adaptor's learning algorithm robustly quantifies and detects the optimal write throughput (OWT) available to a destager in the cluster. It runs continuously and dynamically updates the OWT to respond to changing IO conditions in the cluster. We calculate OWT in 2 ways - (i) as the knee point of batch size vs observed write throughput curve [2], and (ii) as the knee point of write batch size vs observed write throughput power curve, where throughput power is the ratio of throughput to latency [3]. The latency is aggregated across acceleration and storage tier accommodating negative changes in both tiers. The batch size at OWT is used as the optimal batch size for the destager. The destager starts with a conservative value for the write batch size which is then increased by the adaptation module at regular intervals based on OWT calculation and checks to see if the knee has been reached. Once a knee is determined, the system stays at this point until certain counter conditions are triggered. When a counter condition is hit, the system backs off from the current knee and the whole detection process is repeated again. Every destager in the cluster executes the above algorithm. Since the changing conditions of the shared storage such as throughput loss or latency increase is visible to all the destagers, the different destagers in the cluster reach an equilibrium without explicit state sharing. Our results so far show upto 30% predictable improvement in throughput and 50% reduction in latency under overloaded conditions.

Related Work: PARDA [4] enforces proportional-share fairness among distributed hosts accessing a storage device at a latency threshold using flow control mechanisms similar to FAST TCP. PARDA operates at the VM level and is not designed to optimize aggregate IO throughput. Our work achieves the best possible balance between storage backend throughput and latency in a stateless algorithm with no user intervention. Similarly, while some of the techniques from other adaptation papers may be applicable, the overall architecture and constraints are different.

[1] BHAGWAT D., et. al. A practical implementation of clustered fault tolerant write acceleration in a virtualized environment. In FAST 2015.

[2] SATOPAA V., et. al. Finding a kneedle in a haystack: Detecting knee points in system behavior. In ICDCSW 2011.

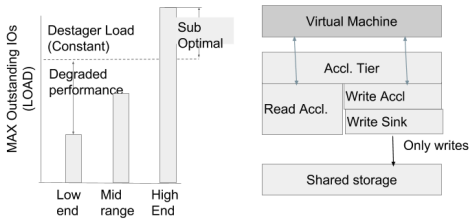
[3] KLEINROCK L. Power and deterministic rules of thumb for probabilistic problems in computer communications. In International Conference on Communications 1979.

[4] GULATI A., et. al. PARDA: proportional allocation of resources for distributed storage access. In FAST 2009.

Adaptive Write-Back Destaging for Server-Side Caches

Motivation

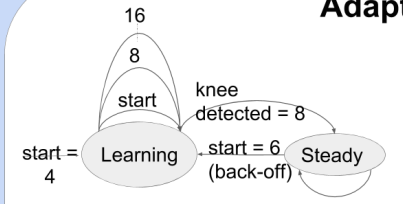
- Server side caching alters workload to the storage backend
- With write back storage becomes a write sink for data at rest
- Multiple hosts writing data back in large batches overwhelms storage
- Non-linear increase in latency and loss of throughput



Problem Statement

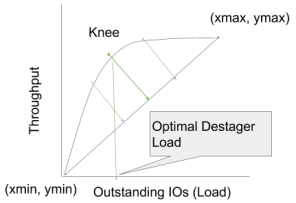
- What is optimal write batch size ? Needs to work with diverse storage arrays
- Write back destager needs to maximize write bandwidth without adverse latency effects resulting in VM stalls
- Destagers also need to play fair with peers in the cluster

Adaptive Destaging State Machine



- Learning state: Determine knee point of batch size vs throughput or power
- Steady state: Stay until counter condition is hit and back off.

Knee of the curve detection



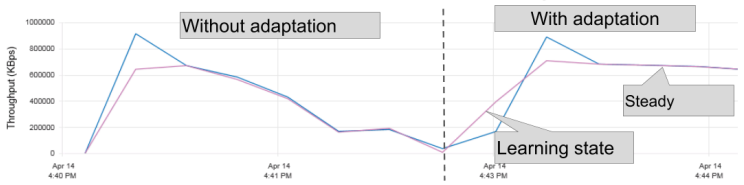
Throughput Power P:

$$P = \frac{Avg.\ Throughput}{Avg.\ Latency} = \frac{Avg.\ destager\ tput}{Avg.\ destager\ IO\ latency}$$

- Knee of the curve = identifies optimal batch size
- Use power for latency sensitivity
- Adapt to changing IO sizes by optimizing over-all throughput
- Stateless distributed Algorithm

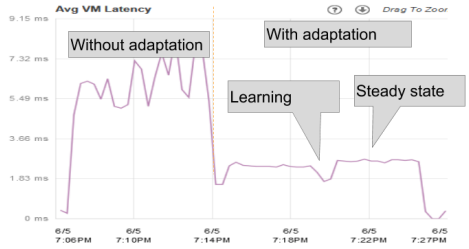
Evaluation

Knee Metric: 30% improvement in throughput



- Workload: 1 MB writes from VM backed by NAS.

Power Metric: Latency reduced by 50%



Future Work

- Model HBA queue in the adaptation algo
- Tune cluster wide adaptation & backoff
- Broaden scope of solution to encompass end-to-end IO control for VMs

*We acknowledge the contributions from the R&D, performance, and system test teams in PernixData Engineering for their help with experiments and reviews.