Log-structured Write-Back Cache

Introduction

RAM cache has been used to boost applications just as long as the modern computing itself exists. However, now, IT environments need something better than good old caching strategies. The problem is, the cache cannot ensure consistent performance under intense virtualized workloads which are prone to random I/O. Also, the traditional caching approach does not grant the decent resiliency as it relies heavily on the volatile memory.

With Log-structured Write-Back Cache (LSWBC) – StarWind VSAN feature that optimizes the data flows for your underlying storage – we revolutionize the way data is cached. The proposed caching strategy is an effective combination of RAM and a small SSD pool that is designed to provide your applications both resiliency and performance they require.

Problem

The traditional caching does not provide any optimization under intense virtualized workloads. To understand why this is happening, let's take a closer look at how data flows. Each virtual machine writes data to the storage, and all of this data has to be written immediately. So the overall data flow gets highly randomized I/O. This data goes to the cache allocated in fast storage like RAM or Flash, and then gets flushed to the underlying storage. All without any optimization to make it lighter on the underlying storage. That's why the regular cache doesn't bring any consistent performance improvement over the uncached storage.

You see, disk-based storage cannot ensure sustained performance under small random writes. Its latency shoots up, and, as a result, your performance-hungry applications do not work as fast as you expect them. It's absolutely true that you can ensure high performance with all-flash storage. But, an all-flash array may be overkill: what if you never use all that jaw-dropping performance after all?

Data loss in the event of a blackout is another major concern. Users face a dilemma of the traditional cache retention policies: you are to choose between a fast cache wiping out all your data in case of a blackout, or a resilient one without any performance gain. Write-back caching is fast, but speed comes with a reliability tradeoff. You see, with that caching strategy in place, data is written to the underlying storage through RAM. If all of a sudden a host or entire cluster goes down, all the data that has not been de-staged to the non-volatile storage is gone. Distributed write-back caching protects the data against node failure, but it won't save it from a catastrophic or blackout event when you, basically, have the whole cluster down.

Apart from data loss, full synchronization is another thing you should worry about in the event of a blackout. The post-blackout recovery process is associated with lengthy synchronization end error checking of the entire storage pool. The thing is, both these processes induce immense load on the system and system administrators. As a result, your applications do not perform as you want and resiliency is degraded until the systems are fully synchronized. Additionally, there is a good risk of a complete failure and even data loss.

Solution

With LSWBC, we overcome the traditional cache limitations. We present a cache tailored to intense virtualized workload needs. It is an effective combination of RAM and a small storage pool purpose-built to deliver your applications the performance and resiliency they require. We also bring Log Structuring into the play to optimize how data is written to the underlying storage.

Let's look under LSWBC hood. StarWind Virtual SAN writes the data to the RAM cache first. Then, the data is flushed sequentially to the log device formed out of a tiny portion of your disk storage. The log keeps a track of consistent states for quick recovery in the event of a blackout or a cluster-wide failure Subsequently, all the data is sent to the underlying storage where it eventually resides. In this way, LSWBC ensures high application performance even under highly randomized I/O loads.

By developing a unique caching architecture, we eradicate any possible risk of data loss due to a blackout. Data retention time in RAM is so momentary that there's, basically, no data to lose! Additionally, there is a copy on the log disk that always survives as that device resides on the non-volatile memory.

Another nice thing about LSWBC: it eliminates post-blackout full synchronization. This technology allows tracking all the data generated by the virtual machines. If all of a sudden a node or entire cluster goes down, the log disk is the only place that nodes need to check to restore data integrity. No full sync is needed. Nodes just quickly match log disks and the cluster is good to go! With all that being said, LSWBC dramatically simplifies cluster maintenance and makes recovery processes faster than ever before.

Conclusion

By changing the way the data is cached, Log-structured Write-Back Cache ensures high performance and data integrity for your virtualized applications. This feature improves the overall performance with even mere flash storage requirements. LSWBC also almost zeroes out the cluster integrity restoration time and removes the manual recovery requirement.