

# Achieving Cost/Performance Balance Ratio Using Tiered Storage Caching Techniques: A Case Study with CephFS

*Tuesday 11 October 2016 16:30 (15 minutes)*

As demand for widely accessible storage capacity increases and usage is on the rise, steady IO performance is desired but tends to suffer within multi-user environments. Typical deployments use standard hard drives as the cost per/GB is quite low. On the other hand, HDD based solutions for storage are not known to scale well with process concurrency and soon enough, high rate of IOPs create a “random access” pattern killing performance. Though not all SSDs are alike, SSDs are an established technology often used to address this exact “random access” problem. Whilst the cost per/GB of SSDs has decreased since inception, their costs are still significantly more than standard HDDs. A possible approach could be the use of a mixture of both HDDs and SSDs coupled with a caching mechanism between the two types of drives. With such approach, the most performant drive technology can be exposed to the application while the lower performing drives (in IOPs performance metric) used for storage capacity. Furthermore, least used files could be transparently migrated to the least performing storage in the background. With this agile concept, both low cost and performance may very well be achieved. Flashcache, dm-cache, and bcache represents a non-exhaustive list of low-level disk caching techniques that are designed to create such tiered storage infrastructure.

In this contribution, we will first discuss the IO performance of many different SSD drives (tested in a comparable and standalone manner). We will then be discussing the performance and integrity of at least three low-level disk caching techniques (Flashcache, dm-cache, and bcache) including individual policies, procedures, and IO performance. Furthermore, the STAR online computing infrastructure currently hosts a POSIX-compliant Ceph distributed storage cluster - while caching is not a native feature of CephFS (but only exists in the Ceph Object store), we will show how one can implement a caching mechanism profiting from an implementation at a lower level. As our illustration, we will present our CephFS setup, IO performance tests, and overall experience from such configuration. We hope this work to service the community’s interest for using disk-caching mechanisms with applicable uses such as distributed storage systems and seeking an overall IO performance gain.

## Tertiary Keyword (Optional)

Object stores

## Secondary Keyword (Optional)

Distributed data handling

## Primary Keyword (Mandatory)

Storage systems

**Primary authors:** Dr LAURET, Jerome (Brookhaven National Laboratory); POAT, Michael (Brookhaven National Laboratory)

**Presenter:** POAT, Michael (Brookhaven National Laboratory)

**Session Classification:** Posters A / Break

**Track Classification:** Track 4: Data Handling