**MODULES:**

* Data deduplication
* POD
* Select-Dedupe
* iCache

**MODULE DESCRIPTIONS:**

**Data deduplication:**

Data deduplication has been demonstrated to be an effective technique in Cloud backup and archiving applications to reduce the backup window, improve the storage-space efficiency and network bandwidth utilization. The Data deduplication technique to detect the redundant content on disks but does not eliminate them on the I/O path. This allows the disk head to service the read requests by pre-fetching the nearest blocks from all the redundant data blocks on disk to reduce the seek latency. The write requests are still issued to disks even if their data has already been stored on disks.

**POD:**

POD resides in the storage node and interacts with the *File Systems* via the standard read/write interface. Thus, POD can be easily incorporated into any HDD-based primary storage systems to accelerate their system performance. Moreover, POD is independent of the upper file systems, which makes POD more flexible and portable than whole-file deduplication and iDedup. It can be deployed in a variety of environments, such as virtual machine images that are mostly identical but differ in a few data blocks.

POD has two main components: Select-Dedupe and iCache.

**Select-Dedupe:**

The request-based *Select-Dedupe* includes two individual modules: Data Deduplicator and Request Redirector. The *Data Deduplicator* module is responsible for splitting the incoming write data into data chunks, calculating the hash value of each data chunk, and identifying whether a data chunk is redundant and popular. Based on this information, the *Request Redirector* module decides whether the write request should be deduplicated, and maintains data consistency to prevent the referenced data from being overwritten and updated.

**iCache:**

The *iCache* module also includes two individual modules: Access Monitor and Swap Module. The *Access Monitor* module is responsible for monitoring the intensity and hit rate of the incoming read and write requests. Based on this information, the *Swap* module dynamically adjusts the cache space partition between the index cache and read cache. Moreover, it swaps in/out the cached data from/to the back-end storage. iCache helps request-based Select-Dedupe deduplicate as many redundant data blocks as possible and improves the read performance by expanding the read cache size in face of read bursts.