**Leveraging Data Deduplication to Improve the Performance of Primary Storage Systems in the Cloud**

**ABSTRACT**

With the explosive growth in data volume, the I/O bottleneck has become an increasingly daunting challenge for big data analytics in the Cloud. Recent studies have shown that moderate to high data redundancy clearly exists in primary storage systems in the Cloud. Our experimental studies reveal that data redundancy exhibits a much higher level of intensity on the I/O path than that on disks due to relatively high temporal access locality associated with small I/O requests to redundant data. Moreover, directly applying data deduplication to primary storage systems in the Cloud will likely cause space contention in memory and data fragmentation on disks. Based on these observations, we propose a performance-oriented I/O deduplication, called POD, rather than a capacity-oriented I/O deduplication, exemplified by iDedup, to improve the I/O performance of primary storage systems in the Cloud without sacrificing capacity savings of the latter. POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select-Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic. We have implemented a prototype of POD as a module in the Linux operating system. The experiments conducted on our lightweight prototype implementation of POD show that POD significantly outperforms iDedup in the I/O performance measure by up to 87.9 percent with an average of 58.8 percent. Moreover, our evaluation results also show that POD achieves comparable or better capacity savings than iDedup.

**EXISTING SYSTEM**

The existing data deduplication schemes for primary storage, such as iDedup and Offline-Dedupe, are capacity oriented in that they focus on storage capacity savings and only select the large requests to deduplicate and bypass all the small requests (e.g., 4 KB, 8 KB or less). The rationale is that the small I/O requests only account for a tiny fraction of the storage capacity requirement, making deduplication on them unprofitable and potentially counterproductive considering the substantial deduplication overhead involved. However, previous workload studies have revealed that small files dominate in primary storage systems (more than 50 percent) and are at the root of the system performance bottleneck. Furthermore, due to the buffer effect, primary storage workloads exhibit obvious I/O burstiness.

**Disadvantages of Existing System:**

1. From a performance perspective, the existing data deduplication schemes fail to consider these workload characteristics in primary storage systems, missing the opportunity to address one of the most important issues in primary storage, that of performance.
2. Our experimental studies suggest that directly applying data deduplication to primary storage systems will likely cause space contention in the main memory and data fragmentation on disks. This is in part because data deduplication introduces significant index-memory overhead to the existing system and in part because a file or block is split into multiple small data chunks that are often located in non-sequential locations on disks after deduplication. This fragmentation of data can cause a subsequent read request to invoke many, often random, disk I/O operations, leading to performance degradation.

**PROPOSED SYSTEM**

To address the important performance issue of primary storage in the Cloud, and the above deduplication-induced problems, we propose a Performance-Oriented data Deduplication scheme, called POD, rather than a capacity-oriented one (e.g., iDedup), to improve the I/O performance of primary storage systems in the Cloud by considering the workload characteristics. POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select-Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic.

**Advantages of Proposed System:**

1. POD significantly improves the performance and saves capacity of primary storage systems in the Cloud

**SYSTEM ARCHITECTURE**



**MODULES**

In this implementation we have 4 modules,

1. Data Deduplicator Module
2. Request Redirector Module
3. Access Monitor Module
4. Swap Module

**Module Description:**

**Data Deduplicator:**

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.

**Request Redirector:**

Based on Data Deduplicator 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.

**Access Monitor:**

The Access Monitor module is responsible for monitoring the intensity and hit rate of the incoming read and write requests.

**Swap:**

Based on Access Monitor 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.

**SYSTEM CONFIGURATION**

# Hardware Configuration

# Processor - Pentium –IV

* Speed - 1.1 Ghz
* RAM - 256 MB(min)
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGASoftware Configuration

**SOFTWARE REQUIREMENTS:**

* Operating System : Windows XP
* Programming Language : JAVA/J2EE