1. INTRODUCTION

Recent technological advancements in cloud computing, internet of things and social network, have led to a deluge of data from distinctive domains over the past two decades. Cloud data centers are awash in digital data, easily amassing petabytes and even Exabyte of information, and the complexity of data management escalates in big data. However, IDC data shows that nearly 75% of our digital world is a copy.

Data deduplication, a specialized data reduction technique widely deployed in disk-based storage systems, not only saves data storage space, power and cooling in data centers, also decreases significant administration time, operational complexity and risk of human error. It partitions large data objects into smaller parts, called chunks, represents these chunks by their fingerprints, replaces the duplicate chunks with their fingerprints after chunk fingerprint index lookup, and only transfers or stores the unique chunks for the purpose of improving communication and storage efficiency.

Data deduplication has been successfully used in various application scenarios, such as backup system, virtual machine storage, primary storage, and WAN replication. Big data deduplication is a highly scalable distributed deduplication technique to manage the data deluge under the changes in storage architecture to meet the service level agreement requirements of cloud storage. It is generally in favor of source inline deduplication design, because it can immediately identify and eliminate duplicates in datasets at the source of data generation, and hence significantly reduce physical storage capacity requirements and save network bandwidth during data transfer. It performs in a typical distributed deduplication, framework to satisfy scalable capacity and performance requirements in massive data.

The framework includes inter-node data assignment from clients to multiple deduplication storage nodes by a data routing scheme, and independent intra-node redundancy suppression in individual storage nodes. Unfortunately, this chunk-based inline distributed deduplication framework at large scales faces challenges in both inter-node and intra-node scenarios. First, for the inter-node scenario, different from those distributed deduplication with high overhead in global match query, there is a challenge called deduplication node information island. It means that deduplication is only performed within individual nodes due to the communication overhead considerations, and leaves the cross-node redundancy untouched. Second, for the intra-node scenario, it suffers from the chunk index lookup disk bottleneck. There is a chunk index of a large dataset, which maps each

chunk's fingerprint to where that chunk is stored on disk in order to identify the replicated data. It is generally too big to fit into the limited memory of a deduplication node, and causes the parallel deduplication performance of multiple data streams to degrade significantly due to the frequent and random disk index I/Os.

There are several existing solutions that aim to tackle the above two challenges of distributed deduplication by exploiting data similarity or locality. Locality means that the chunks of a data stream will appear in approximately the same order again with a high probability. Locality only based approaches distribute data across deduplication servers at coarse granularity to achieve scalable deduplication throughput across the nodes by exploiting locality in data streams, but they suffer low duplicate elimination ratio due to high cross-node redundancy.

Similarity in this context means that two segments of a data stream or two files of a dataset share many chunks even though they arrive in a random order. The most similar stored segments or files are perfetched to deduplicate the processing segment or file in low-locality workloads by exploiting a property called logical locality. Similarity-only based methods leverage data similarity to distribute data among deduplication nodes to reduce cross-node duplication, while they also often fail to obtain good load balance and high intra-node deduplication ratio by fingerprint based mapping and allowing some duplicate chunks to be stored.

In recent years, researchers exploit both data similarity and locality to strike a sensible tradeoff between the conflicting goals of high deduplication effectiveness and high performance scalability for distributed deduplication. However, all these schemes are oblivious to the content and format of application files, and cannot find the redundancy in files with complex format, like video and audio files. Hence, their space efficiency can be further improved by exploiting application awareness.

This is a codesign of storage and application to optimize deduplication based storage systems when the deduplicated storage layer has extensive knowledge about the file structures and their access characteristics in the application layer. As shown in Table1.1, the conventional deduplication schemes always improve performance in single-node scenario or distributed scenario without considerations on application awareness. In the latest research works, application aware duplicate detection has been adopted to single-node deduplication to improve deduplication efficiency with low system overhead.

AppDedupe is a scalable source inline distributed deduplication framework with leveraging application awareness, as a middleware deployable in data centers, to support big data management in cloud storage. This solution takes aim at large-scale distributed deduplication with thousands of storage nodes in cloud datacenters which would most likely fail in the traditional distributed methods due to some of their shortcomings in terms of global deduplication ratio, single-node throughput, data skew, and communication overhead. The main idea behind AppDedupe is to optimize distributed deduplication by exploiting application awareness, data similarity and locality in streams. More specifically, it performs two-tiered routing decision which firstly dispenses application metadata at file level with an application-aware routing to keep application locality, then assigns chunk fingerprints of intra-app similar data to the same storage node at the super-chunk (i.e. consecutive smaller chunks) granularity using a handprinting-based stateful data routing scheme to maintain high global deduplication efficiency without cross-node deduplication, meanwhile balances the workload of nodes from clients. Finally, it performs application-aware deduplication in each node independently and in parallel. To reduce the overhead of resemblance detection in each node, we build an application-aware similarity index to alleviate the chunk index lookup disk bottleneck for the deduplication processes in individual nodes. The client only needs to send the unique chunks of the super-chunk to the target node, because duplicate detection process is performed in the target node before data transfer.

The proposed AppDedupe distributed deduplication system has the following salient features that distinguish it from the state-of-the-art mechanisms:

– To the best of our knowledge, AppDedupe is the first research work on leveraging application awareness in the context of distributed deduplication.

– It performs two-tiered routing decision by exploiting application awareness, data similarity and locality to direct data routing from clients to deduplication storage nodes to achieve a good tradeoff between the conflicting goals of high deduplication effectiveness and low system overhead.

– It builds a global application route table and independent similarity indices with super-chunk handprints over the traditional chunk-fingerprint indexing scheme to alleviate the chunk lookup disk bottleneck for deduplication in each storage node.

– Evaluation results show that it consistently and significantly outperforms the state-of-the-art schemes in distributed deduplication efficiency by achieving high global deduplication

effectiveness with balanced storage usage across the nodes and high parallel deduplication throughput at a low inter-node communication overhead.

TABLE 1.1: Applications of Deduplication

Type	Application Oblivious	Application Awareness
Single-node Deduplication	DDFS, SiLo, ChunkStash, D2D, RevDedup	ViDeDup, ADMAD, AA-Dedup, ALG-Dedupe
Distributed Deduplication	ExtremeBinn, EMC, DEBAR, CALB, MAD2, IBM, DEDIS, HYDRAstor, Symantec, Produck, ∑-Dedupe.	AppDedupe