1st International Conference on Applied Soft Computing Techniques
22 & 23.04.2017
In association with
International Journal of Scientific Research in Science and Technology

A Survey on Deduplication Workload Resource for Big Data Applications

[1]V. Manochitra, [2]B.Jackline Jose
[1]Department of Information Technology, Bon Secours College for Women, Thanjavur
manokavishna@gmail.com
[2]Department of Information Technology, Bon Secours College for Women, Thanjavur
jackholyangel24@gmail.com

ABSTRACT

Deduplication seems to be a appropriate explanation for data detonation in the big data era by 1) slowing down the data growth speed by removing redundant data, and 2) relieving pressure on disk bandwidth by removing dismissed IO accesses. However, deduplication also introduces above to the system. For example, hash indexing needs be performed for every IO request to classify duplicates, which results in slower IO response time. In addition, extra CPU control is required to compute the hash values in each IO request, which leads to progressive vigor consumption. Since the capacity of IO needs is enormous and increasing in big data workloads, the overall performance and energy capability below different deduplication configurations is valuable to be deliberate methodically.

Keywords: Big Data, Deduplication, Hash Indexing, Resource Allocation, Big Data Analysis

I. INTRODUCTION

The demand for data storage and processing is increasing at a rapid speed in the big data era. Such a tremendous amount of data pushes the limit on storage capacity and on the storage network. The data analysis of the International Data Corporation (IDC), the volume of data in the world will reach 40 trillion gigabytes in 2020. In order to reduce the burden of maintaining big data, more and more enterprises and organizations have chosen to outsource data storage to cloud based big data storage providers. This makes data management a critical challenge for the big data storage providers. To achieve optimal usage of storage resources, many existing cloud storage providers perform deduplication, which exploits data redundancy and avoids storing duplicated data from multiple users.

A substantial helping of the dataset in big data assignments is redundant. As a result, deduplication knowledge, which eradicates replicas, develops an attractive solution to save disk space and traffic in a big data environment. However, the overhead of extra CPU subtraction (hash indexing) and IO budding introduced by deduplication should be painstaking. Therefore, the net outcome of using deduplication for big data loads needs to be examined.

II. WHY DEDUPLICATION IN BIG DATA?

- More than 2.5 quintillion bytes of data are generated every day. 90% of the total data has been bent just in the past few years alone. To cover such a huge amount of data, storage continues to grow at an volatile rate (52% per year) [1][14]. By the end of 2015, the size of the total formed data will surpass 7.9 zettabytes (ZB). This number is expected to reach 35 ZB in 2020, which has established to be too conservative [2][15].
- Big data assignments have joblessness. On even, 44% of the lively data set in our big data assignments is redundant. Organizing an extra VM yields 97% more redundant data. Using a repetition device in the Hadoop distributed file system (HDFS) presents 19% jobless data on average. The

statistics node covers 25% more jobless data than the name node.

- Additional confusion calculation on the CPU leads to extra power ingesting (around 10%), which results in drive overhead (7%). However, for the benchmarks with a high level of joblessness, the overall energy can be protected (by 43% in our experiment).The deduplication reduces assignment implementation time. There is a strong association between drive influence and the degree of redundancy.
- Deduplication helps assignments utilize more disk amount (3X higher when deduplication is on in our experiment), which leads to, at most, a 45% performance development. Though, due to the above of hash indexing, performance can damage by 161% for some benchmarks in an extreme circumstance.
- In a cross SSD/HDD (Solid State Drive / Hard Disk Drive) environment, deduplication can improve the scheme performance (by up to 17% in our experiment) if the SSD relative is adjusted correctly. However, in a pure SSD situation, deduplication costs presentation and liveliness overhead (about 5% and 6% respectively).

III. LITERATURE SURVEY AND RELATED WORK

Dictionary Reduction model

Traditionally, data reduction has been the result of data compression approaches that use a dictionary model to identify redundancy for short strings (e.g., 16 B), such as the classic LZ77/LZ88 algorithms. Most of these approaches first compute a weak hash of strings and then compare the hash-matched strings byte by byte. Because of their time and space complexity, dictionary model-based compression approaches, such as LZO, LZW, DEFLATE, only compress data in a much smaller region, e.g., data within a file or a group of small files, which trades off processing speed against compression effectiveness.

Delta Compression Technique

The technique is to the indexing issue of delta compression either record the resemblance information for files, instead of data chunks, so that similarity index entries can fit in the memory, or exploit the locality of backup data streams in deduplication-based backup/archiving systems, which avoids the global indexing on the disk

Locality-based approach

Locality-based approaches exploit the inherent locality in a backup stream, which is widely used in state-of-the-art deduplication systems such as DDFS, Sparse Indexing, and ChunkStash. The locality in this context means that the chunks of a backup stream will appear in approximately the same order in each full backup with a high probability. Mining this locality increases the RAM utilization and reduces the accesses to on-disk index, thus alleviating the disk bottleneck.

Similarity based approach

Similarity based approaches are designed to address the problem encountered by locality-based approaches in backup streams that either lack or have very weak locality (e.g., incremental backups). They exploit data similarity instead of locality in a backup stream, and reduce the RAM usage by extracting similar characteristics from the backup stream. A well-known similarity-based approach is Extreme Binning that improves deduplication scalability by exploiting the file similarity to achieve a single on-disk index access for chunk lookup per file.

History Aware Re-writing Algorithm

This existing model is to rewrite duplicate but fragmented chunks during the backup via rewriting algorithm, which is a trade-off between deduplication ratio (the size of the non-deduplicated data divided by that of the deduplicated data) and restore performance. These approaches buffer a small part of the on-going backup stream, and identify the fragmented chunks within the buffer. They fail to accurately identify sparse containers because an out-of-order container seems

sparse in the limited-sized buffer. Hence, most of their rewritten chunks belong to out-of-order containers, which limit their gains in restore performance and garbage collection efficiency.

The Capping algorithm

Capping algorithm are recently proposed rewriting algorithms to address the fragmentation problem. Both of them buffer a small part of the on-going backup stream during a backup, and identify fragmented chunks within the buffer (generally 10-20 MB). For example, Capping divides the backup stream into fixed-sized segments (e.g., 20 MB), and conjectures the fragmentation within each segment. Capping limits the maximum number (say the chunks in the *N of T* containers that hold the least chunks in the segment are rewritten.

3.1 RELATED WORK

Qiang Li, Qinfen Hao, Limin Xiao and **Zhoujun Li**

[14] proposed VM-base architecture for adaptive management of virtualized resources in cloud computing. The authors also designed a resource controller named Adaptive Manager that dynamically adjusts multiple virtualized resource utilization to achieve application Service Level Objective (SLO) using feedback control theory. Adaptive Manager is a multi-input, multi-output (MIMO) resource controller which controls CPU scheduler, memory manager and I/O manager based on feedback mechanism. For the periodic measurement of the application performance each Virtual Machine has sensor module which transmits information to the adaptive manager. Authors adopted Kernel based Virtual Machine (KVM) as a tool for infrastructure of virtual machine.

Mayank Mishra, Anwesha Das, Purushottam Kulkarni and **Anirudha Sahoo** [15] discussed that live virtual machine migration plays a vital role in dynamic resource management of cloud computing. The authors mainly focused on efficient resource utilization in non-peak periods to minimize wastage of resources. For the attainment of goals like server consolidation, load balancing and hotspot mitigation, authors discussed three components – when to migrate, which VM to migrate and where to migrate – and approaches followed by different heuristics to apply migration techniques. Authors also discussed virtual machine migration over LAN and WAN with their challenges.

Fan and Bifet [16] pointed out that the terms "big data" and "big data mining" were first presented in 1998, respectively. The big data and big data mining almost appearing at the same time explained that finding something from big data will be one of the major tasks in this research domain. Data mining algorithms for data analysis also play the vital role in the big data analysis, in terms of the computation cost, memory requirement, and accuracy of the end results. In this section, we will give a brief discussion from the perspective of analysis and search algorithms to explain its importance for big data analytics.

Ruijin Zhou, Ming Liu, Tao Li [17] deliberated that the redundancy of big data is measured by three foundations 1) deploying more nodes, 2) increasing the dataset, and 3) using repetition devices. They future about characterize the joblessness of characteristic big data assignments to justify the need for duplication. They deliberated about the analyze and describe the presentation and energy influence brought by repetition under various big data surroundings. In their trials, they identify three bases of redundancy in big data jobs: 1) deploying more nodes, 2) expanding the dataset, and 3) using replication devices.

Min Chen · Shiwen Mao · Yunhao Liu [18] focused on the four phases of the value chain of big data, i.e., data generation, data acquisition, data storage, and data analysis. For each phase, they introduce the general background, discuss the technical challenges, and review the latest advances. Finally they examined the several representative applications of big data, including enterprise management, Internet of Things, online social networks, medial applications, collective intelligence, and smart grid.

Dongchul Park, Ziqi Fan, Young Jin Nam, and David H. C. Du [19] discussed on Data duplication (dedupe for short) is a special data compression technique. It has been widely adopted to save backup time as well as storage space, particularly in backup storage systems. Therefore, most dedupe research has primarily focused on improving dedupe write performance. So backup storage dedupe read performance is also a crucial problem for storage recovery.

T.Thamarai Selvan [20] presented a programmatic cloud suitable medical diagnostic computing application which actually gets a pulse modified into 32 bit form data via sensors and plot it as time varying graph. The author drafted an optimal pulse system measurement algorithm and made it venal into the programmatic application which is very efficient in transfiguring data to its core binary format with lose-less nature and plot graph which can be exported into any Big data applications for further analyzing. The algorithm used in this paper can be referenced for creating a resource allocation model in cloud environment running big data applications as it focuses on a schema where the data is processed in real time and processed.

Pritee Patil, Nitin N. Pise [21] discussed on the greatest test for enormous information from a security perspective is the assurance of client's protection. Enormous information as often as possible contains gigantic measures of individual identifiable data and thusly security of clients is a colossal concern. Be that as it may, encoded information present new difficulties for cloud information deduplication, which gets to be significant for huge information stockpiling and preparing in cloud. Customary deduplication plans can't take a shot at encoded information.

Irfan Ahmad Murali Vilayannur Jinyuan Li [22] propose that DEDE, a block-level deduplication system for live cluster file systems that does not require any central coordination, tolerates host failures, and takes advantage of the block layout policies of an existing cluster file system.

Yinjin Fu, Hong Jiang , Nong Xiao[23] proposed that Σ-Dedupe, a scalable inline cluster deduplication framework, as a middleware deployable in cloud data centers, to meet this challenge by exploiting data similarity and locality to optimize cluster deduplication in inter-node and intra-node scenarios.

Lei Wei, Chuan Heng Foh, Bingsheng He and **Jianfei Cai** [24] proposed a heterogeneous resource allocation approach, called skewness-avoidance multi-resource allocation (SAMR), to allocate resource according to diversified requirements on different types of resources. The work includes a VM allocation algorithm to ensure heterogeneous workloads are allocated appropriately to avoid skewed resource utilization in PMs, and a model-based approach to estimate the appropriate number of active PMs to operate SAMR.

Ricardo Koller Raju Rangaswami [25] discussed that e I/O Deduplication, a storage optimization that utilizes content similarity for improving I/O performance by eliminating I/O operations and reducing the mechanical delays during I/O operations. I/O Deduplication consists of three main techniques: content-based caching, dynamic replica retrieval, and selective duplication.

Lauro Beltrao Costa , Samer Al-Kiswany , Raquel Vigolvino Lopes and Matei Ripeanu [26] discussed about the energy trade-offs brought by data deduplication in distributed storage systemsData deduplication enables a trade-off between the energy consumed for additional computation and the energy saved by lower storage and network load.

Shengmei Luo, Guangyan Zhang, Chengwen Wu, Samee U. Khan[27] proposed data deduplication replaces identical regions of data (files or portions of files) with references to data already stored on the disk. Compared with the traditional compression techniques, data deduplication can eliminate not only the data redundancy within a single file, but also the data redundancy among multiple files.

IV. SUMMARY OF RESOURCE ALLOCATION FOR BIG DATA APPLICATION

Table 1 summarizes the work done by various researchers and future work and/or gaps in their existing work.

Table I. Summary of Resource Allocation Techniques

Year	Author	Technique/ Algorithm	Tools used	Future works and/or Gaps in existing technologies	Discussion in terms of Deduplication Workload Resource Allocation for Big Data Application support (can be made by creating / availed by / integrated to / can take this model or algorithm or procedure)
2009	**Qiang Li, Qinfen Hao, Limin Xiao and Zhoujun Li [14]**	Adaptive Management of Visualized resources using Feedback control.	KVM	Only KVM model,I/O Performance, still better modelling can be done	Support can be made by integration with KVM bbut cannot be expected to make load balancing.
2012	**Mayank Mishra, Anwesha Das, Purushottam Kulkarni and Anirudha Sahoo [15]**	Live Virtual Machine Migration	Not mentioned	Only Load on virtual machine is considered. Consumer requirements and priority of the job is not considered.	Support can be made if we create an API which can integrate with the big data application.
2015	**Fan and Bifet [16]**	Data Mining algorithm	Not mentioned	Author has plan to discover more diverse, larger, and faster.	Big Data may be a hype to sell Hadoop based computing systems.
2013	**Ruijin Zhou, Ming Liu, Tao Li[17]**	Advanced hash indexing algorithm	Not mentioned	Writer has plans to modify the benefit of evading jobless disk admissions and the overhead of hashing.	Big data requests counting web hunt, machine learning, analytical inquiry, and categorization
2014	**Min Chen· Shiwen Mao· Yunhao Liu [18]**	Hash algorithm, Analytical Algorithm	Not mentioned	The author has plan to propose a new algorithm on special multimedia event detection using a few positive training	The growth of big data applications accelerates the revolution and innovation of data centers. Many big data applications have developed their unique architectures and directly promote the development of storage, network, and computing technologies related to data center
2016	**Dongchul Park, Ziqi Fan, Young Jin Nam, and David H. C. Du [19]**	Cache Algorithm (LRU)	Not mentioned	In future the author has to extended design outperforms LRU by an average of 64.3%	Big Data dedupe is a specialized technique to eliminate duplicate data so that it retains only one unique data copy on storage
2011	**T.Thamarai Selvan [20]**	Optimal pulse system	Not mentioned	Author has plans to modify the schema to	Schema of this algorithm can be used to test the efficiency

		measurement algorithm		promote more resource allocation optimisation.	of the application with respect to streaming data.
2016	**Pritee Patil, Nitin N. Pise [21]**	Resource allocation through skewness technique	Not mentioned	Extensive performance analysis and test showed that our scheme is secure and efficient under the described security model and very opportune for sizably voluminous data deduplication	Distributed computing is a rising administration display that gives calculation and capacity assets on the Web.
2009	**Irfan Ahmad Murali Vilayannur Jinyuan Li [22]**	Deduplication algorithm	Not mentioned	Author has plan to explore alternate indexing schemes that allow for greater control of deduplication policy	The evaluation of our deduplication techniques using various microbenchmarks and realistic workloads
2012	**Yinjin Fu, Hong Jiang , Nong Xiao[23]**	Data Routing algorithm	Xen Virtual machine	Author has plan achieve in each node by exploiting similarity and locality in backup data streams.	Managing the data deluge under the changes in storage media to meet the SLA requirements becomes an increasingly critical challenge for Big Data protection
2015	**Lei Wei, Chuan Heng Foh, Bingsheng He and Jianfei Cai [27]**	Heterogeneous resource allocation approach, called skewness-avoidance multi-resource allocation (SAMR)	VMware	Workloads are concentrated but the work allocator bases are left after primary allocation table readiness.	The approach can be studied for the work load handling but cannot be considered as a whole as Big data applications must be controlled in more trivial aspects for better and efficient data processing.
2010	**Ricardo Koller Raju Rangaswami [25]**	Replacement algorithm	Not mentioned	The future direction is to optionally coalesce or even eliminate altogether write I/O operations for content that are already duplicated elsewhere on the disk, or alternatively direct such writes to alternate locations in the scratch space	Needs a lot of implementation as the technique proposed has a lot of variation with regard to certain big data data sets.
2011	**Lauro Beltrao Costa , Samer Al-Kiswany , Raquel Vigolvino Lopes‡ and Matei Ripeanu [26]**	Checkpointing is representative for workloads that can benefit from deduplication	Check point application	Author propose an energy consumption model that highlights the same issues and, in spite of its simplicity, can be used to reason about the energy and performance break-even points when	Data deduplication is a method to detect and eliminate similarities in the data.

				configuring a storage system	
2015	**Shengmei Luo, Guangyan Zhang, Chengwen Wu, Samee U. Khan[27]**	Data routing algorithm and Rabin's fingerprint algorithm.	Not mentioned	Author propose that in future we remove the data deduplication ratio in single node with the help of cache container of hot fingerprint based on access frequency	Computing resources of a cluster efficiently and satisfy the applications' demands of parallel processing on big data in the cloud storage system.

V. CONCLUSION AND DISCUSSION

Big data management strategies and best practices are still evolving, but joining the big data movement has become an imperative for companies across a wide variety of industries. Big data analytics is the process of examining large data sets to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information. The analytical findings can lead to more effective marketing, new revenue opportunities, better customer service, improved operational efficiency, competitive advantages over rival organizations and other business benefits.Many authors have proposed algorithms, methods and techniques for dynamic resource allocation in Deduplication Workload Resource for Big Data Applications environment that support big data applications. We have taken schema from some authors which can be converted to a meaningful insight for our work. In summary, an efficient Resource Allocation Technique should meet following criteria's: Quality of Service (QoS) aware utilization of resources, cost reduction and power reduction / energy reduction.. The ultimate goal of Deduplication Workload Resource Allocation for Big Data Applications is to identify the redundancy of sequences of bytes across very large comparison windows. Sequences of data (over 8 KB long) are compared to the history of other such sequences. The first uniquely stored version of a sequence is referenced rather than stored again.

VI. REFERENCES

[1]. EMC Data Domain. http://www.datadomain.com/.

[2]. IBM ProtecTIER. http://www-03.ibm.com/systems/storage/news/center/deduplication/index.html.

[3]. Acronis. http://www.acronis.com/backup-recovery/deduplication-roicalculator.html

[4]. B. Efron. Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2010.

[5]. U. Fayyad. Big Data Analytics: Applications and Opportunities in On-line Predictive Modeling. http:// big-data-mining.org/keynotes/#fayyad, 2012.

[6]. Dimitrios Zissis and Dimitrios Lekkas, "Addressing cloud computing security issues" in ELSEVIER - Future Generation Computer Systems 28 (2012) 583–592.

[7]. Liang-Jie Zhang, Jia Zhang, Jinan Fiaidhi, J. Morris Chang, "Hot Topics in Cloud Computing" in IEEE Computer Society, ITPro 2012, 1520-9202.

[8]. Robert Grossman, "The Case for Cloud Computing" in IEEE Computer Society, IT Pro 2009.

[9]. B. Efron. Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2010.

[10]. U. Fayyad. Big Data Analytics: Applications and Opportunities in On-line Predictive Modeling. http:// big-data-mining.org/keynotes/#fayyad, 2012.

[11]. D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In SODA, 2013.

[12]. J. Gama. Knowledge Discovery from Data Streams. Chapman & Hall/Crc Data Mining and Knowledge Discovery. Taylor & Francis Group, 2010.

[13]. J. Gantz and D. Reinsel. IDC: The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. December 2012.

[14]. Gartner, http://www.gartner.com/it-glossary/bigdata

[15]. Offshore Oil and Gas Supply. Working Document of the National Petroleum Council, 2011

[16]. The Changing Geospatial Landscape. A Report of the National Geospatial Advisory Committee, 2009

[17]. How Big Data Is Changing Astronomy (Again). The Atlantic, 2012

[18]. W. Cox, M. Pruett, T. Benson, S. Chiavacci, and F. Thompson III. Development of Camera Technology for Monitoring Nests. USGS Northern Prairie Wildlife Research Center, 2012

[19]. http://www.groundcontrol.com/Oil-And-Gas_Satellite.htm.

[20]. Bhagwat D, Eshghi K, Long D D E, et al. Extreme binning: Scalable, parallel deduplication for chunk-based file backupCModeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on. IEEE, 2009: 1-9.

[21]. Dong W, Douglis F, Li K, et al. Tradeoffs in Scalable Data Routing for Deduplication ClustersCFAST. 2011: 15-29.

[22]. You L L, Pollack K T, Long D D E. Deep Store: An archival storage system architectureCData Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on. IEEE, 2005: 804-815.

[23]. Eshghi K, Tang H K. A framework for analyzing and improving contentbased chunking algorithmsJ. Hewlett-Packard Labs Technical Report TR, 2005, 30: 2005.

[24]. Liu C, Lu Y, Shi C, et al. ADMAD: Application-Driven Metadata Aware De-duplication Archival Storage SystemCStorage Network Architecture and Parallel I/Os, 2008. SNAPI'08. Fifth IEEE International Workshop on. IEEE, 2008: 29-35.

[25]. Bobbarjung D R, Jagannathan S, Dubnicki C. Improving duplicate elimination in storage systemsJ. ACM Transactions on Storage (TOS), 2006, 2(4): 424-448.

[26]. Kruus E, Ungureanu C, Dubnicki C. Bimodal content defined chunking for backup streams CProc of the USENIX FAST10, Brekeley, CA:USENIX, 2010: 239-252

[27]. Zhu B, Li K, Patterson R H. Avoiding the Disk Bottleneck in the Data Domain Deduplication File SystemCFast. 2008, 8: 1-14. 29Bloom B H. Space/time trade-offs in hash coding with allowable errorsJ. Communications of the ACM, 1970, 13(7): 422-426. 30Lillibridge M, Eshghi K, Bhagwat D, et al. Sparse Indexing: Large Scale, Inline Deduplication Using Sampling and LocalityCFast. 2009, 9: 111- 123.

[28]. Broder A Z. On the resemblance and containment of documentsC//Compression and Complexity of Sequences 1997. Proceedings. IEEE, 1997: 21-29.

[29]. Debnath B, Sengupta S, Li J. ChunkStash: speeding up inline storage deduplication using flash memoryCProceedings of the 2010 USENIX conference on USENIX annual technical conference. USENIX Association, 2010: 16-16.