Nat. Volatiles & Essent. Oils, 2021; 8(2): 118-124

NVEO
Natural Volatiles &
Essential Oils

The Data De-Duplication In Cloud Storage Management

Dr.E. Mohan[1] , Mr. R. Anandan[2] , Shakthibalan S[3]

[1]Professor, Dhanalakshmi Srinivasan College of Engineering and Technology.

[2]Assistant Professor, Dhanalakshmi Srinivasan College of Engineering and Technology.

[3]Student, Dhanalakshmi Srinivasan College of Engineering and Technology.

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. Moreover, directly applying data de-duplication to primary storage systems in the Cloud will likely cause space contention in memory and data fragmentation on disks. Based on these observations, this project propose a performance-oriented I/O de-duplication, called POD, rather than a capacity oriented I/O de-duplication, exemplified by I Dedup, To improve the I/O performance of primary storage systems in the Cloud without sacrificing capacity savings of the latter.

---

1. INTRODUCTION

Data de duplication is a technique for eliminating duplicate copies of repeating data. A related and somewhat synonymous term is single-instance (data) storage. This technique is used to improve storage utilization and can also be applied to network data transfers to reduce

the number of bytes that must be sent. In the de-duplication process, unique chunks of data, or byte patterns, are identified and stored during a process of analysis. As the analysis continues, other chunks are compared to the stored copy and whenever a match occurs, the redundant chunk is replaced with a small reference that points to the stored chunk. Given that the same byte pattern may occur dozens, hundreds, or even thousands of times (the match frequency is dependent on the chunk size), the amount of data that must be stored or transferred can be greatly reduced.

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing

consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

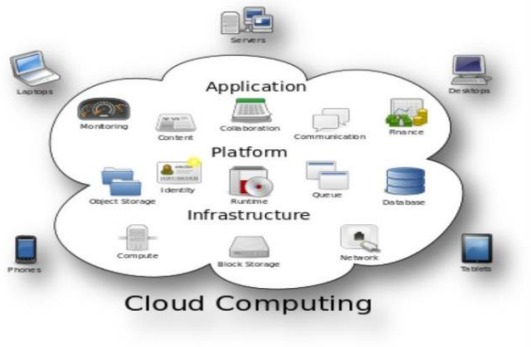

Fig 1.1: Cloud Computing Overview

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer- oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

A. On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.

B. Broad network access:
Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

C. Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multitenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able

to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

D. Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

E. Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

2. MATERIALS AND METHODS

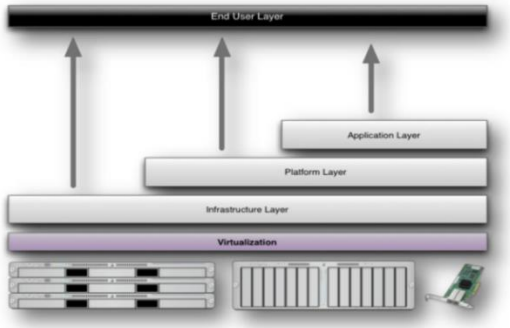

Figure 1.2: Services Model

Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software- as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.

A. Achieve economies of scale – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.

B. Reduce spending on technology infrastructure – Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.

C. Globalize your workforce on the cheap – People worldwide can access the cloud, provided they have an Internet connection.

D. Streamline processes – Get more work done in less time with less people. v Reduce capital costs. There's no need to spend big money on hardware, software or licensing fees.

E. Improve accessibility – You have access anytime, anywhere, making your life so much easier!

F. Monitor projects more effectively – Stay within budget and ahead of completion cycle times. Less personnel training is needed. It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.

G. Minimize licensing new software – Stretch and grow without the need to buy expensive software licenses or programs.

H. **Improve flexibility –** You can change direction without serious "people" or "financial" issues at stake.

3. RESULTS AND DISCUSSIONS

De-duplication technologies are increasingly being deployed to reduce cost and increase space-efficiency in corporate data centers. However, prior research has not applied de-duplication techniques inline to the request path for latency sensitive, primary workloads. This is primarily due to the extra latency these techniques introduce. Inherently, de-duplicating data on disk causes fragmentation that increases seeks for subsequent sequential reads of the same data, thus, increasing latency. In addition, de-duplicating data requires extra disk IOs to access on-disk de-duplication metadata.

In this paper, we propose an inline deduplication solution, I Ded up, for primary workloads, while minimizing extra IOs and seek. Our algorithm is based on two key insights from real world workloads: i) spatial locality exists in duplicated primary data; and ii) temporal locality exists in the access patterns of duplicated data. Using the first insight, we selectively de- duplicate only sequences of disk blocks. This reduces fragmentation and amortizes the seeks caused by deduplication. The second insight allows us to replace the expensive, on-disk, de-duplication metadata with a smaller, in-memory cache. These techniques enable us to tradeoff capacity savings for performance, as demonstrated in our evaluation with real-world workloads. Our evaluation shows that I Ded up achieves 60-70% of the maximum de-duplication with less than a 5% CPU overhead and a 2-4% latency impact.

Data de-duplication has been demonstrated to be an effective technique in reducing the total data transferred over the network and the storage space in cloud backup, archiving, and primary storage systems, such as VM (virtual machine) platforms. However, the performance of restore operations from a deduplicated backup can be significantly lower than that without de- duplication.

In computer science, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment. Many implementations may exist for a given specification or standard. For example, web

browsers contain implementations of World Wide Web Consortium- recommended specifications, and software development tools contain implementations of programming languages.

A. TESTING :

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

(I). **Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

(Ii). Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

(iii). Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

(iv). System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

(v). White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

(vi). Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

4. CONCLUSION AND FUTURE ENHANCEMENT

This system focus on Performance-Oriented De-duplication(POD) scheme, to improve the performance of primary storage systems in the Cloud by leveraging data de-duplication on the I/O path to remove redundant write requests while also saving storage space. It takes a request based selective de-duplication approach (Select-Dedupe) to de-duplicating the I/O redundancy on the critical I/O path in such a way that it minimizes the data fragmentation problem. In the meanwhile, an intelligent cache management (I Cache) is employed in POD to further improve read performance and increase space saving, by adapting to I/O burstiness. Our extensive trace driven evaluations show that POD significantly improves the performance and saves capacity of primary storage systems in the Cloud. POD is an ongoing research project and we are currently exploring several directions for the future research. First, we will incorporate I Cache into other de-duplication schemes, such as I Ded up, to investigate how much benefit I Cache can bring to saving extra storage capacity and improving read performance. Second, we will build a power measurement module to evaluate the energy efficiency of POD. By reducing write traffic and saving storage space, POD has the potential to save the power that disks consume. We will compare the extra power that CPU consumes for computing fingerprints with the power that the storage saves, thus systematically investigating the energy efficiency of POD.

REFERENCES

[1] B. Mao, H. Jiang, S. Wu, Y. Fu, and L. Tian. SAR: SSD Assisted Restore Optimization for Deduplication-based Storage Systems in the Cloud. In NAS’12, Jun. 2012.

[2] B. Mao, H. Jiang, S. Wu, Y. Fu, and L. Tian. Read Performance Optimization for Deduplication-based Storage Systems in the Cloud. ACM Transactions on Storage, 10(2):1–22, 2014.

[3] N. Megiddo and D. Modha. Arc: A self-tuning, low over head replacement cache. In FAST’03, Mar. 2003.

[4] D. Meister, J. Kaiser, A. Brinkmann, T. Cortes, M. Kuhn, and J. Kunkel. A Study on Data Deduplication in HPC Storage Systems. In SC'12, Nov. 2012.

[5] J. Menon. A Performance Comparison of RAID-5 and Log-Structured Arrays. In HPDC'95, pages 167–178, Aug. 1995.

[6] D. T. Meyer and W. J. Bolosky. A Study of Practical Deduplication. In FAST'11, Feb. 2011.

[7] Y. Oh, J. Choi, D. Lee, and Sam H. Noh. Caching less for better performance: Balancing cache size and update cost of flash memory cache in hybrid storage systems. In FAST'12, Feb. 2012.