To Avoid Deduplication Using Content Similarity And Checksum Algorithm With Job Scheduling Method

N.R.Bhuvaneswari
Department of Computer Science and Engineering
A.V.C College of Engineering

Mrs.B.Muthulakshmi
Department of Computer Science and Engineering
A.V.C College of Engineering

***Abstract* - Experimentally, magnetic tape items has been used for database backup. With the explosion in disk capacity, it is now impossible to use disk for data backup. Cloud storage is used for the database backup. Data deduplication is a specialized data compression technique for eliminating duplicate copies of repeating data. To identify similar segments, use content similarity and a sparse index. Choose a small portion of the chunks in the stream as samples. Storage-based data deduplication reduces the amount of storage needed for a given set of files. It is most effective in applications where many copies of very similar or even identical data are stored on a single disk. Virtual cloud servers benefit from deduplication because it allows nominally separate system files for the cloud to be concluded in a single storage space. The proposed method can be allocating the resource can be based on the dependencies and the particular job execution and its weight of the each job and content similarity avoid the de-duplications**

Index Terms - Cloud storage, on-demand access, collective I/O, De-duplication.

I. INTRODUCTION

One of the main features that has contributed to the growing popularity of Infrastructure-as-a-Service (IaaS) cloud computing is the elastic on-demand provisioning of resources: users can bring up a whole virtual cluster and reconfigure it dynamically with a simple click of a button. However, as the user interface grows simpler and the types of workloads diversify,achieving efficient on-demand VM provisioning is a non-trivial task.

A particularly difficult challenge in this context is the *collective on-demand read* pattern, i.e., provisioning a large number of inter-dependent VMs (e.g. part of the same virtual cluster running a large scale distributed application) that concurrently read (typically) a part of the content from the same VM (virtual machine) disk image (e.g., boot and launch applications) or from a large dataset (e.g., shared input data). This pattern is often encountered in the context of largescale HPC (high performance computing) and data-intensive applications. Obviously, there is a need to minimize the provisioning time and guarantee scalability despite a growing number of VMs, otherwise users do not perceive IaaS as truly on-demand and lose interest, while at the same time cloud providers lose potential profit by not efficiently leveraging their computational resources.

Despite widespread need for scalable, high-performance solutions that handle the collective on-demand read pattern, IaaS cloud providers offer limited support in this regard. Most often, in an attempt to avoid any bottlenecks due to I/O contention to the storage service where the VM images and datasets are stored, it is very common to broadcast the full content to the local storage of the VM instances before allowing any read.

However, most of the time, this approach is sub-optimal because of two reasons: (1) not all content is actually read; and (2) reads need to wait for the whole broadcast to finish. Thus, approaches that deliver content onthe-fly as needed in order to eliminate these two disadvantages saw increasing adoption, despite the added complexity of having to deal with the I/O contention to the storage service. One major direction that addresses the problem of I/O contention for on-the-fly data delivery during collective reads is the use of peer-to-peer collaborative techniques. In this class of solutions, the VM instances are aware of each other's previously accessed data that is locally available and prefer to exchange the needed data among themselves rather than interact with the decoupled storage service, which risks the creation of bottlenecks due to I/O contention. Although related to pre-broadcast techniques (which are typically implemented as BitTorrent-like protocols), the focus in this context falls on how to detect and anticipate what content is actually needed during the runtime of a VM instance, in order to be able to pre-fetch it from the other VM instances as early as possible.

However, despite the success of such techniques to improve the performance and scalability of collective reads, most of the time they require foreknowledge about what VM instances are related and what dataset or VM image they share and read in a concurrent fashion. This is a significant limitation for large IaaS cloud datacenters where a large number of users share the infrastructure simultaneously, because there are multiple opportunities for VM instances to collaborate and exchange identical pieces of data even if they belong to different users for which the relationship between the VM instances, their access pattern and the data they are reading is unknown. This aspect is particularly important in light of several studies that confirm a large amount of

redundancy among VM images, with the data duplication degree reported up to 94%

II. RELATED WORK

Content similarity detection is typically performed by means of deduplication, which is broadly classified into static and content-defined. Static approaches split the input data into equally sized chunks, which are then compared among each other (either byte-by-byte or, for increased performance, based on their hash values) in order to identify and eliminate duplicates. While simple and fast, static approaches suffer from misalignment issues (i.e insertions or deletions lead to the impossibility to detect duplicates). To deal with such misalignment issues, content defined approaches were proposed. Essentially, they involve a sliding window over the data and that hashes the window content at each step using Rabin's fingerprinting methodMany storage systems have adopted and refined deduplication techniquesTechniques to fetch data from storage services to VM instances are broadly classified into pre-broadcast and on- demand. Pre-broadcast techniques use various scalable mechanisms (e.g., multi-cast application level broadcast- trees to peer-to-peer protocols to deliver a shared dataset from the storage service to multiple VM instances in advance, such that it can be used later without worrying about bottlenecks due to I/O bandwidth contention.

However, on the downside, the broadcast can take a long time to finish and potentially delivers more content than is actually needed during runtime. On-demand techniques on the other hand eliminate both disadvantages at the cost of dealing with the I/O bandwidth contention during runtime. This approach is widely used in IaaS datacenters for virtual disk images using copy-on-write: a locally stored QCOW2image is instantiated from a shared backing image that is located remotely on the image store (e.g. NFS server). In an attempt to alleviate the I/O contention, various solutions ranging from decentralizing the storage (e.g. by using parallel file system to using dedicated repositories and specialized prefetching technique have been proposed. In a broader sense, collaborative caching has been explored in the MPI-IO context. Our own previous work explores how to improve collective reads to a shared virtual disk image by means of pushing accessed chunks among the members of the group, in an attempt to anticipate and avoid direct access to the storage service. This paper focuses on exploiting content similarity on-the- fly in order to enable multiple VM instances, even if they be- long to different dissemination groups, to collaborate, identify and exchange identical chunks of data in order to minimize the I/O pressure on the storage service under concurrency. To our best knowledge, we are the first to focus on this aspect in particular.

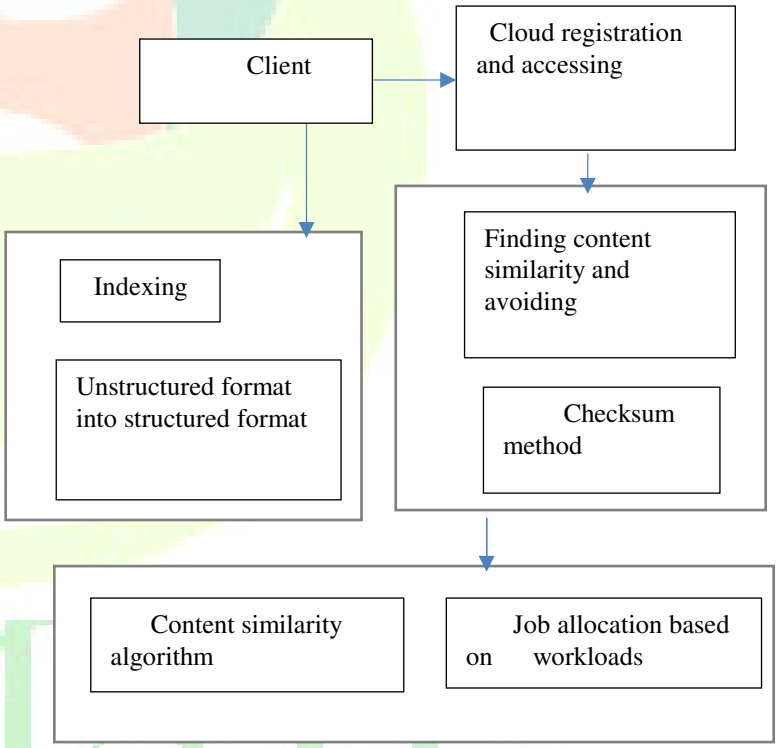
III. EXISTING SYSTEM

In Existing method the major direction that addresses the problem of I/O contention for on-the-fly data delivery during collective reads is the use of peer-to-peer collaborative techniques. In this class of solutions, the VM instances are aware of each other's previously accessed data that is locally available and prefer to exchange the needed data among themselves rather than interact with the decoupled storage service, which risks the creation of bottlenecks due to I/O contention. The data transformation is done by deduplication to avoid the traffic and to improve I/O request performance the content similarity technique is used.

Disadvantages:

- Content Similarity is Used
- Reduces Amount of Data Transfer and No of Transfer
- Lack of Resource Scheduling
- Workload Balance is not Maintained
- Lack of User Interface

IV. SYSTEM ARCHITECTURE



A particularly difficult challenge in this context is the *collective on-demand read* pattern, i.e.,

provisioning a large number of inter-dependent VMs (e.g. part of the same virtual cluster running a large

spelling and grammar. Use high resolution (300dpi or above) figures, plots, drawings and photos for best printing result.

TABLE I
TYPE SIZE FOR PAPERS

Type size (pts.)	Appearance		
	Regular	Bold	Italic
6	Table superscripts		
8	Section titles[a], references, tables, table names[a], table captions, figure captions, footnotes, text subscripts, and superscripts		
9		Abstract, Index Terms	
10	Authors' affiliations, main text, equations, first letter in section titles[a]		Subheading
11	Authors' names		
22	Paper title		

[a]Uppercase

B. *Preparing Your PDF Paper for IEEE Xplore©*

Detailed instructions on how to prepare PDF files of your papers for IEEE *Xplore*© can be found at

http://www.ieee.org/pubs/confpubcenter

PDF job setting files for Acrobat versions 4, 5 and 6 can be found for downloading from the above webpage as well. The instructions for preparing PDF papers for IEEE *Xplore*© must be strictly followed.

II. HELPFUL HINTS

A. *Figures and Tables*

Try to position figures and tables at the tops and bottoms of columns and avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be centered below the figures; table captions should be centered above. Avoid placing figures and tables before their first mention in the text. Use the abbreviation "Fig. #," even at the beginning of a sentence.

Figure axis labels are often a source of confusion. Use words rather than symbols. For example, as shown in Fig. 1, write "Magnetization," or "Magnetization (M)" not just "M." Put units in parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization (A□m^{-1})." Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)," not "Temperature/K."

Multipliers can be very confusing. Write "Magnetization (kA/m)" or "Magnetization (10^3 A/m)." Figure labels should be legible, at 8-point type.

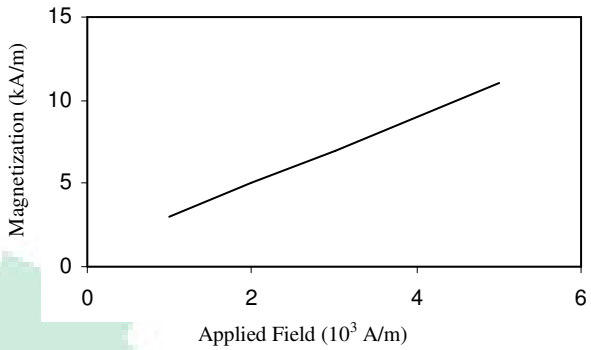


Fig. 1 Magnetization as a function of applied field. Note how the caption is centered in the column.

B. *References*

Number citations consecutively in square brackets [1]. Punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]. Use "Ref. [3]" or "Reference [3]" at the beginning of a sentence: "Reference [3] was the first …"

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes (see Table I). *IEEE Transactions* no longer use a journal prefix before the volume number. For example, use "*IEEE Trans. Magn.*, vol. 25," not "vol. MAG-25."

Give all authors' names; use "et al." if there are six authors or more [4]. Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. In a paper title, capitalize the first word and all other words except for conjunctions, prepositions less than seven letters, and prepositional phrases.

For papers published in translated journals, first give the English citation, then the original foreign-language one [6].

C. *Abbreviations and Acronyms*

Define abbreviations and acronyms the first time they are used in the text, even if they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title unless they are unavoidable.

D. *Equations*

Number equations consecutively with equation numbers in parentheses flush with the right margin, as in (1). To make your equations more compact, you may use the solidus (/) and the exp function, etc. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use an en dash (–) rather than a hyphen for a minus sign. Use parentheses to avoid ambiguities in denominators. Punctuate equations with commas or periods when they are part of a sentence, as in

$$\frac{e^{ix}}{2} = \frac{\cos x + i \sin x}{2} \Rightarrow \exp(ix)/2 = (\cos x + i \sin x)/2 . \quad (1)$$

Symbols in your equation should be defined before the equation appears or immediately following. Cite equations

using "(1)," not Eq. (1)" or "equation (1)," except at the beginning of a sentence: "Equation (1) is …"

E. Other Recommendations

The Roman numerals used to number the section headings are optional. Do not number ACKNOWLEDGEMENT and REFERENCES and begin Subheadings with letters. Use two spaces after periods (full stops). Hyphenate complex modifiers: "zero-field-cooled magnetization." Avoid dangling participles, such as, "Using (1), the potential was calculated." Write instead, "The potential was calculated using (1)," or "Using (1), we calculated the potential."

Use a zero before decimal points: "0.25," not ".25." Use "cm^3," not "cc." Do not mix complete spellings and abbreviations of units: "Wb/m^2" or "webers per square meter," not "webers/m^2." Spell units when they appear in text: "…a few henries," not "…a few H." If your native language is not English, try to get a native English-speaking colleague to proofread your paper. Do not add page numbers.

III. UNITS

Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive."

Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

IV. SOME COMMON MISTAKES

The word "data" is plural, not singular. In American English, periods and commas are within quotation marks, like "this period." A parenthetical statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.) A graph within a graph is an "inset," not an "insert." The word alternatively is preferred to the word "alternately" (unless you mean something that alternates). Do not use the word "essentially" to mean "approximately" or "effectively." Be aware of the different meanings of the homophones "affect" and "effect," "complement" and "compliment," "discreet" and "discrete," "principal" and "principle." Do not confuse "imply" and "infer." The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen. There is no period after the "et" in the Latin abbreviation "et al." The abbreviation "i.e." means "that is," and the abbreviation "e.g." means "for example." An excellent style manual for science writers is [7].

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g." Try to avoid the stilted expression, "One of us (R. B. G.) thanks …" Instead, try "R.B.G. thanks …" Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

[1] M. King, B. Zhu, and S. Tang, "Optimal path planning," *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.
[2] H. Simpson, *Dumb Robots*, 3rd ed., Springfield: UOS Press, 2004, pp.6-9.
[3] M. King and B. Zhu, "Gaming strategies," in Path Planning to the West, vol. II, S. Tang and M. King, Eds. Xian: Jiaoda Press, 1998, pp. 158-176.
[4] B. Simpson, et al, "Title of paper goes here if known," unpublished.
[5] J.-G. Lu, "Title of paper with only the first word capitalized," *J. Name Stand. Abbrev.*, in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Translated J. Magn. Japan*, vol. 2, pp. 740-741, August 1987 [*Digest 9th Annual Conf. Magnetics Japan*, p. 301, 1982].
[7] M. Young, *The Technical Writer's Handbook*, Mill Valley, CA: University Science, 1989.