

一种基于重复数据删除的镜像文件存储方法研究

李 锋¹ 陆婷婷¹ 郭建华²

(东华大学计算机科学与技术学院 上海 201620)¹ (上海大学计算机科学与技术学院 上海 200444)²

摘 要 在云计算环境中,基础设施即服务的日益发展导致虚拟机和虚拟机镜像的急剧增加,例如,Amazon Elastic Compute Cloud(EC2)有 6521 个公共虚拟机镜像文件,这给云环境的管理带来了极大的挑战,特别是大量镜像文件带来的重复数据的空间存储问题。为解决这一问题,提出一种基于固定分块的镜像文件重复数据删除的存储方案。当存储一个镜像文件时,先计算该镜像文件的指纹,并与指纹库的指纹比较,若存在则用指针替代,否则采用固定分块对镜像文件分割存储。为此,可以设计镜像文件元数据格式和镜像文件 MD5 索引表来解决上述问题。实验结果表明,内容相同的镜像文件只是元数据的开销并实现秒传,而相同版本、相同系统、不同软件的镜像组的重删率约达到 58%。因此,本方案是非常有效的。

关键词 云计算,重复数据删除,镜像文件存储

中图分类号 TP399 文献标识码 A

Effective Image File Storage Technique Using Improved Data Deduplication

LI Feng¹ LU Ting-ting¹ GUO Jian-hua²

(Department of Computer Science and Technology, Donghua University, Shanghai 201620, China)¹

(Department of Computer Science and Technology, Shanghai University, Shanghai 200444, China)²

Abstract In the cloud computing environment, the increasing development of Infrastructure as a Service leads to the sharp increase of virtual machine and virtual machine image. For example, Amazon Elastic Compute Cloud(EC2) has 6521 public virtual machine image files, which bring a great challenge to the management of the cloud environment. In particular, the spatial storage of duplicate data brought by a large number of mirror images. In order to solve this problem, this paper proposed a storage scheme for a fixed block of the image file based deduplication. When an image file is stored, we should calculate the image file's fingerprint first and compared with the fingerprint database. If it exists in the fingerprint database, we should replace it with pointer, else using fixed block to split and storage image file. To this end, we designed the image file metadata format and mirror file MD5 index table. The experiment shows that the same content image file is just the cost of metadata and the second pass. And the same version of the same system, but different software's mirror group, whose deletion rate is about 58%. As a result, our scheme is very effective.

Keywords Cloud computing, Deduplication, Image file storage

1 引言

根据 Gartner 的统计,2015 年底,中国公共云服务市场规模达到了近 70 亿美元,并有望在 2018 年年底增长至 207 亿美元。由于移动互联网用户众多以及公共云服务市场的蓬勃发展,许多中国企业已经计划部署云计算,以顺应网络驱动、超连接数字化的电子商务业务的增长趋势^[1]。

云计算服务的日益发展导致虚拟机(Virtual Machine)和虚拟机镜像(VM images)急剧增加,这给镜像文件的管理带来了极大的挑战。例如,Amazon Elastic Compute Cloud(EC2)就有 6521 个公共虚拟机镜像文件^[2]。

镜像文件存储已成为云计算研究的关键问题^[3-5]。镜像文件存储系统中存在着大量重复数据。从文件级看,镜像文件之间有着大量相同的数据。一方面由于镜像文件存储系统

中存在大量相同类型的镜像文件。如都是 ubuntu12.04 的系统,只是在一个镜像中安装了 Mysql,另一个镜像中安装了 Apache。文献[6]指出在安装相同操作系统(Fedora9)和不同应用程序(office,SDK 和 Web)的镜像文件中存在 90%左右的重复数据。另一方面由于镜像文件中的安装的系统类似。如都是 linux 系统,一个为 Redhat,另一个为 CentOS,或者系统相同但版本不同,一个为 ubuntu13.04,一个为 ubuntu13.10。从文件内看,在虚拟机镜像文件中存在大量的零填充块^[7]。文献[8]对 525 个基于 Linux 和 IBM AIX 操作系统的虚拟机镜像文件进行了分析,发现零填充块在镜像文件中的比例高达 35%~55%。文献[6]指出,在刚安装完操作系统的单个虚拟机镜像文件内部中存在约 5%的重复数据。因此,重复数据删除技术能大幅度地减少用于存储虚拟机镜像文件的存储空间^[6,8,9]。

李 锋 教授,主要研究方向为嵌入式及深度学习;陆婷婷 硕士生,主要研究方向为深度学习、云计算;郭建华 硕士,主要研究方向为大数据并行化处理。

本论文对当前云计算环境下大量虚拟机运行所带来的镜像文件存储问题进行了研究,探讨对镜像文件进行重复数据删除处理。具体来说,本论文讨论了镜像文件存储的元数据设计及组织方式,并评估了存储元数据的空间开销;为镜像文件进行修改和删除,设计了镜像块引用计数;镜像块的备份管理,指出引用计数对于镜像库可靠性的影响,并为高引用度镜像块做了备份管理,以免由于高引用导致度镜像块的损坏从而造成的大量镜像文件的不可用。

2 相关工作

重复数据删除技术的思路是:在存储数据时,具有相同内容的数据块只在物理存储媒介中存储其一份,以避免由于重复存储相同的数据而造成的存储空间开销。只要有重复数据存在,重复数据删除技术就可以发挥其技术优势。因此,有不少学者对虚拟磁盘文件的重复数据删除进行了研究。文献[6]部署了基于 Venti 的 CAS(Content Addressable Storage)系统用于存储镜像文件。文献[9]讨论了重复数据删除技术在虚拟机在线迁移方面的影响。K Takahashi 等人^[10]使用数据删除技术实现一种快速的虚拟存储迁移,通过可重用磁盘页来减少主机 A 和主机 B 之间来回迁移的数据传输量。Nath 等人^[11]使用 CAS 的折中方案对企业规模级的虚拟机进行了量化,发现使用基于块的重复数据删除对于减少存储和网络资源开销效果显著。文献[12]设计了一个用于容错的云环境下的镜像文件存储系统。文献[13]采用静态分块策略对桌面虚拟机镜像去重。文献[6]指出在相似的虚拟机中,运用重复数据删除技术,镜像文件存储和传输的减少量可高达 96%。

3 相关动机

3.1 基于文件的重复数据删除

基于文件的重复数据删除也被称为完全文件检测技术(Whole File Detection, WFD),是以粗粒度的文件为处理对象。首先,直接使用散列函数来计算文件的指纹,其后,以指纹来判断文件是否为重复文件,并决定是否存储该文件,其流程如图 1 所示。例如,微软在其分公司部署了 Windows Storage Server 2003 R2 之后,其共享服务器的数据存储量减少了 40%^[15]。

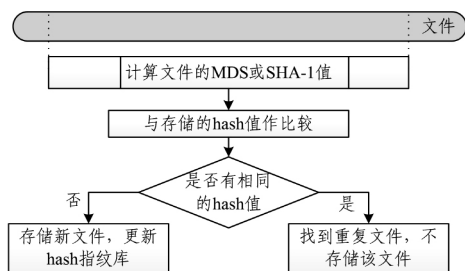


图 1 基于文件的重复数据删除过程示意图

该方法的主要缺点为:1)即使两个文件只有一个字节不同,系统也会将两个文件都存储起来;2)重复数据删除率相对不高,因为该方法无法将同一个文件中的重复数据删除。

3.2 基于数据块的重复数据删除

固定长度分块(Fixed-Sized Partition, FSP)^[16]先将源数据划分成不重叠的、固定尺寸的数据块,源数据划分后最后一

个数据块往往不满足固定尺寸要求,此时,需要对该数据块进行填充空数据处理,如图 2 所示。固定长度分块算法实现简单,对数据的处理速度快。解决了基于文件的重复数据删除不能检测文件内部重复数据的缺点,能够删除文件内部及文件间的重复数据。

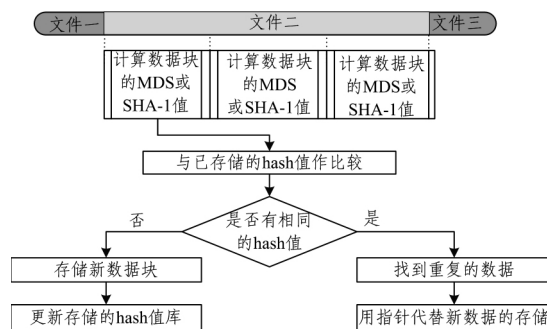


图 2 固定长度分块重复数据删除过程示意图

但同时其对文件的修改操作非常敏感,当对文件进行插入或删除处理时,都可能因为插入或删除数据大小的不同而影响后面的数据,进而影响整个重复数据删除率。

将基于文件和数据块删除的优势结合起来,提出一种基于重复数据删除的镜像文件存储方式。

4 基于重复数据删除的镜像文件存储设计与实现

本节讨论镜像文件重复数据删除模块的设计与实现。讨论为镜像块及元数据设计新的组织方式,评估存储元数据的空间开销,在设计包文件时,引入对镜像块的引用计数,以达到能够对镜像文件进行修改和删除的目的。

为了讨论镜像块重复删除的效果,给出如下 2 个定义。

定义 1 镜像文件内部的重复数据删除比率(简称重复率,记为 P)指单个镜像文件经重复数据删除处理之后,原镜像文件自身数据量(S)和剩余的数据量(R)之差,与原镜像文件自身数据量的比值。记为: $P = \frac{S-R}{S} \times 100\%$ 。

定义 2 镜像文件之间的重复数据删除比率(简称重删率,记为 D_g)指一组镜像文件经重复数据删除之后,原镜像文件组数据量(S_g)和所剩余的数据量(R_g)之差,与原镜像文件组数据量的比值。记为: $D_g = \frac{S_g - R_g}{S_g} \times 100\%$ 。

4.1 镜像文件元数据设计

由于传输镜像文件时通常需要传输整个镜像文件,如果能把镜像块在磁盘中按顺序存储,连续读取数据块的开销将会明显减少。并且如果元数据和镜像块在相邻的位置,在读取元数据后,再次读取镜像块时,不会耗费多少旋转延迟。基于此,按照图 3 所示的方式来组织镜像文件元数据及镜像块。

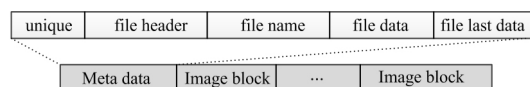


图 3 镜像文件元数据及镜像块组织形式

从图 3 中可以看到,镜像文件元数据和镜像块按顺序组织在一起,元数据包括 unique, file header, file name, file data 和 file last data 5 个字段。其中 unique 字段用以记录整个镜像文件的 MD5 码, fileheader 中记录了文件名的长度、镜像块大小、镜像块数量及最后一个数据块大小等信息, filename 记

录了镜像文件名, filedata 记录了镜像文件的各个镜像块的存储地址, filelastdata 字段记录着镜像文件的最后一个镜像块, 因为这个镜像块的尺寸通常比正常镜像块小, 导致它的重复概率极小, 因此需要单独保存。

为了从整体上对镜像文件切分后的镜像块进行统一且有效的管理, 设计了包文件, 以统一存储与镜像文件相关的信息。

图 4 显示了包文件的内部布局。包文件由 5 个部分构成: 包头 (header)、所有镜像文件的唯一镜像块地址组 (uniqueblockset)、引用数组 (reference)、索引组 (Index) 和指纹信息 (fingerprint)。第一部分的包头信息记录了总的镜像块个数、镜像文件 MD5 码个数、总的镜像块指纹个数、指纹大小和到各个字段的偏移信息等; 第二部分的 uniqueblockset 存储了所有镜像文件的唯一镜像块的地址; 第三部分的 reference 记录唯一镜像块引用次数的引用数组 (reference), 它和 uniqueblockset 中的唯一镜像块一一对应, 用以标记每个唯一镜像块的引用次数; 第四部分的索引组 (Index) 记录所有镜像文件的 MD5 码和所有镜像文件元数据地址; 第五部分指纹记录了所有唯一镜像块的指纹。

header	unique block set	reference	Index	fingerprint
--------	------------------	-----------	-------	-------------

图 4 包文件的内部布局

4.2 存储元数据的空间开销

对于 1G 的镜像文件, 假设用大小为 4kB 的固定块来进行重复数据删除, 采用 ATA-6 规范的 48 位 LBA 地址^[12]将会产生约 1.5M 的元数据。这仅仅多占了约千分之一的空间。因此文件元数据的开销十分有限, 即使存储 1TB 的镜像文件空间开销也仅为 1.5GB (假设镜像文件的大小均为 1GB 级别)。因此完全可以将文件元数据加载到内存之中。同样, 若仅仅存储镜像文件的 MD5 码, 1TB 的镜像文件仅仅需要 16kB 的空间。

4.3 镜像块引用计数

考虑到在用户不使用虚拟机时可能删除镜像文件, 或者用户可能对镜像文件进行修改, 在设计时对每个镜像块设置了引用计数 (如图 3 中的 reference), 这样就可以记录下每个镜像块在文件中的出现次数, 当向镜像库中添加镜像文件时, 如果镜像文件的镜像块在镜像库中存在, 则将对对应镜像块的引用计数加 1, 反之, 当由于删除镜像文件导致要删除某些镜像文件块时, 将对应的镜像文件块引用计数减 1。删除镜像块的原则是引用计数数值为 0。

4.4 改进的镜像文件重复数据删除方案

镜像文件不同于其他类型的文件, 其特点是文件大、重复数据量多。因此在设计时, 采用与传统重复数据删除方案有所不同的处理方式。

主要考虑出发点是: 镜像文件尺寸较大, 当要存储的镜像文件已经存储在镜像存储服务器中时, 采用传统的重复数据删除方案, 就需首先对该镜像文件分块, 然后在指纹库中比对每个分块的指纹, 这种处理方式会花费大量的分块处理时间和指纹比对时间。因此, 为了能够快速检测镜像文件是否已存储, 避免对那些和已存储镜像相同的镜像做切分, 在切分镜像文件之前, 首先为镜像文件生成一个 MD5 码, 来唯一标识该镜像文件。图 5 说明了本文设计的改进的镜像文件重复

数据删除处理方案的流程图。

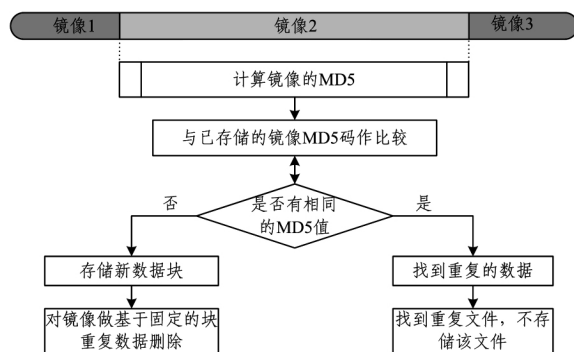


图 5 镜像文件删除流程图

这样该设计方案中会有两个指纹库, 为区别它们, 在文中用“指纹库”指镜像文件切块后生成的指纹集合, 用“镜像文件 MD5 码库”指为镜像文件 (而不是镜像块) 生成的指纹集合。

整个处理流程为: 首先, 为整个镜像文件生成 MD5 码; 然后, 与已存储的镜像文件 MD5 码比对, 若已经存在该 MD5 码, 则不需存储该文件, 若找不到该 MD5 码, 对该镜像文件进行传统的基于固定块的重复数据删除处理。

5 实验

5.1 实验环境

为了验证本文的设计方案, 并做数据分析, 在 1 台物理机上做了一系列实验。这台物理机的配置如下: 型号为 DELL PowerEdge M610, 处理器类型和频率为 Intel® Core™2 Quad CPU Q8300 @2.50GHz processors, 内存为 3.92GB 内存, 硬盘为 Seagate 500GB 7200 RPM, 操作系统为 Ubuntu Server 12.04, Linux 内核为 3.5.0。

5.2 镜像文件重复数据评估

在本次实验中, 主要关注镜像文件内部和镜像文件之间的重复数据删除率。W3Techs 报告^[14]指出, 目前市场上服务器系统占有率最高的系统为 Linux 和 Windows。因此在实验中使用 11 个镜像文件, 共包含两种操作系统分别为 Windows 和 Linux。具体类型及大小如表 1 所列。

表 1 测试的镜像文件

镜像号	系统类型及软件	镜像大小 (GB)
1	Windows XP SP3	1.4
2	Windows 7	9.1
3	Ubuntu 12.04 Server	2.9
4	Ubuntu 12.04 Server+apache2	3.2
5	Ubuntu 12.04 Server+mysql	3.3
6	Ubuntu 13.04 Server	3.1
7	RedHat 5	2.3
8	RedHat 5+php5	2.5
9	CentOS 5.4	2.4
10	Fedora 17	2.5
11	Debian 5	1.8

如图 6 所示, 分别对编号为 1、3、7、9、10 和 11 的镜像文件用固定分块为 4kB 和 8kB 两种方式做重复数据删除。从图中可以看到镜像文件自身的重复率并不高, 在分块为 4kB 时, 重复率最高为 9.5% (RedHat), 最低为 4.2% (Windows XP), 随着分块的增长, 重复率呈下降趋势, 这也符合本文的直观思维, 因为在数据量越大的情况下, 出现不同数据的可能性就越大。

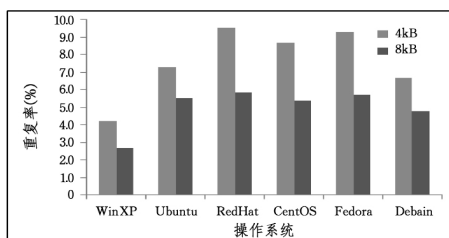


图6 镜像文件内部重复率

如图7所示,对11个镜像文件分别用固定分块为4kB、8kB、16kB、32kB 4种方式做重复数据删除。从图中可以看到随着分块的增长,重删率也呈现下降趋势,并且可以看到重复率下降越来越严重,当分块为32k时,重删率已从原来的54.7%下降到16.8%。

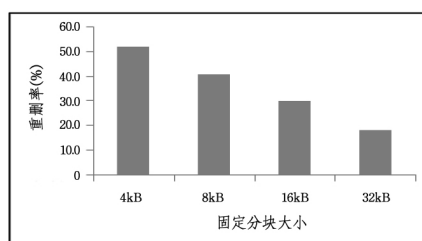


图7 不同固定分块大小下镜像组的重删率

图8中,1+2表示对镜像号为1和2的镜像组进行重复数据删除,同理,3+4+5表示对镜像号为3、4和5的镜像组进行重复数据删除。在图8中,分别对不同版本相同系统的镜像组(如都是Windows的1+2和都是Ubuntu的3+6)、相同版本相同系统不同软件的镜像组(如3+4+5)、不同系统的镜像(如1+3、1+7)、相近系统的镜像(如7+9+10中的RedHat、CentOS和Fedora均属RedHat系列)和相似系统的镜像(如3+7中的Ubuntu和RedHat均属于Linux操作系统)进行了固定块大小为4kB的重复数据删除。从图中可以看出,相同版本相同系统不同软件的镜像组具有最高的重删率,达到了约58%;不同系统的镜像具有最低的重删率约为7%;相近系统的镜像也有不错的重删率约20%。在进一步分析后,发现相同版本相同系统不同软件的镜像之间重复数据约高达88%,而且可以相信在更多镜像文件的环境下重删率能进一步提高。

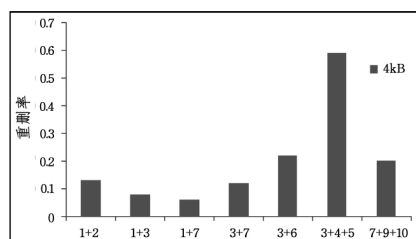


图8 不同镜像组的重删率

结束语 设计实现了基于重复数据删除的镜像文件存储方案。设计新的元数据格式,并针对镜像块的修改和删除设计了引用计数。实验结果表明,使用基于固定分块的重复数据删除技术可以有效地消除镜像文件间的重复数据。在实验中,节约的存储空间达到了54.7%,而且实验中发现具有相同系统相同版本不同软件的镜像和具有相似系统的镜像重复数据量较高。另外,通过本地镜像块也可以明显地加快虚拟机的部署速度。

未来需要对基于镜像重复数据删除的可靠性以及指纹的快速索引进行深入的研究。对于可靠性方面,下一步可以从镜像块引用度、镜像库可靠性、镜像块副本数以及由于备份引起的额外存储开销这4个方面进行权衡,找出最佳的对于高引用度镜像块的备份方案。对于指纹索引方面,下一步需要找到一种能够应对上亿级别的指纹数的快速定位机制。

参考文献

- [1] <http://www.gartner.com/newsroom/id/2603623>
- [2] Cloud A E C. Amazon elastic compute cloud (amazon ec2)[J]. 2013
- [3] Beloglazov A, Buyya R. OpenStack Neat: a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds[J]. Concurrency and Computation: Practice and Experience, 2014, 27(5): 1310-1333
- [4] Ferdaus M H, Murshed M, Calheiros R N, et al. Network-Aware Virtual Machine Placement and Migration in Cloud Data Centers [J]. Emerging Research in Cloud Distributed Computing Systems, 2015, 42
- [5] Peng C, Kim M, Zhang Z, et al. VDN: Virtual machine image distribution network for cloud data centers[C]// INFOCOM, 2012 Proceedings IEEE. IEEE, 2012: 181-189
- [6] Lent A F, Morrisette P M, Clayton-Luce T J. System and method for storage and deployment of virtual machines in a virtual server environment[P]. U. S. Patent 8,943,203, 2015-1-27
- [7] Xu J, Zhang W, Zhang Z, et al. Clustering-based acceleration for virtual machine image deduplication in the cloud environment [J]. Journal of Systems and Software, 2016, 121: 144-156
- [8] Jayaram K R, Peng C, Zhang Z, et al. An Empirical Analysis of Similarity in Virtual Machine Images[C]// Proceedings of the Middleware 2011 Industry Track Workshop (Middleware'11). New York, NY, USA. ACM, 2011: 1-6
- [9] Wang J, Zhao Z, Xu Z, et al. I-sieve: An Inline High Performance Deduplication System Used in Cloud Storage I-sieve: An Inline High Performance Deduplication System Used in Cloud Storage [J]. Tsinghua Science and Technology, 2015, 20(1): 19-29
- [10] Takahashi, Kazushi, Sasada K, et al. A fast virtual machine storage migration technique using data deduplication [C]// The Third International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2012). 2012
- [11] Lewis R, Hartman J H. Accordion: multi-scale recipes for adaptive detection of duplication[C]// 7th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 15). 2015
- [12] Li T, Zhou R. Method and apparatus for virtual machine live storage migration in heterogeneous storage environment[P]. U. S. Patent 9,195,401, 2015-11-24
- [13] Fu Yin-jin, et al. Deduplication based storage optimization technique for virtual desktop[J]. Journal of Computer Research and Development, 2012(S1)
- [14] <http://www.boot-us.com/gloss11.htm>
- [15] Padhye J D, Kandula S, Bahl P. Flyways in data centers[P]. U. S. Patent 8,972,60, 2015-3-3
- [16] Reddy R, Kathpal A, Basak J, et al. Data layout for power efficient archival storage systems[C]// Proceedings of the Workshop on Power-Aware Computing and Systems. ACM, 2015: 16-20