活动数据集中的混合重复数据删除和压缩

学生姓名：李改潮

学　　号：2017282110218

专　　业：计算机技术

班　　级：硕士五班

目 录

1 概述

1.1 背景介绍

随着数字信息量的爆炸式增长，数据占用空间越来越大，在过去的 10 年里,很多行业提供的存储系统容量从数十 GB 发展到数百 TB，甚至数 PB。随着数据的指数级增长，企业面临的快速备份和恢复的时间点越来越多，管理保存数据的成本及数据中心空间和能耗也变得越来越严重。

研究发现，应用系统所保存的数据中高于 60%是冗余的,而且随着时间的推移越来越多。为了缓解存储系统的空间增长问题，缩减数据占用空间，降低成本，最大程度地利用已有资源，数据缩减技术已成为一个热门的研究课题。一方面，利用数据缩减技术可以对存储空间的利用率进行优化，以消除分布在存储系统中的相同文件或者数据块。另一方面，利用数据缩减技术可以减少在网络中传输的数据量，进而降低能量消耗和网络成本,并为数据复制大量节省网络带宽。

1.2 数据缩减方法介绍

数据缩减技术（Data Reduction）是所有能够有效地删除数据冗余、提升存储空间利用率的技术的统称，根据消除冗余数据的方法不同分为数据压缩（Data Compression）、Delta 编码（Delta Encoding）和重复性数据删除（Deduplication）三种策略。

1.2.1 数据压缩技术

数据压缩是指采用一定的编码方式对数据进行重新组织，用更少的数据位表示数据对象内原始信息的技术，包含有损压缩和无损压缩两大类。无损压缩是指使用压缩后的数据进行重构，重构后的数据与原来的数据完全相同，常用的无损压缩算法有霍夫曼（Huffman）算法和 Lempel-Ziv（LZ）系列算法，并被广泛地应用于 WinRAR、GZIP 和 GIF 等压缩软件。有损压缩是指使用压缩后的数据进行重构，重构后的数据与原来的数据有所不同，但不影响对原始资料表达的信息，适

用于图像和声音处理等多媒体应用中。数据压缩方法由于是用一种更接近信息本质的描述代替原有冗余数据描述，作用在数据对象内部，只能消除对象内部的冗余，因此只能获得有限的数据缩减率。

1.2.2 Delta 编码技术

Delta 编码是一种通过开发数据对象间的相似性来压缩数据对象存储空间的技术，通过定位两个相似的数据对象间的相同数据内容，按它们的存储先后分为旧版本和新版本，采用差分算法获得数据对象新旧版本之间的差异，并用获得的差异表示数据的新版本并进行存储。由于 Delta 对象编码后可以远小于原数据对象，因此该方法能够有效地缩减数据对象的存储空间。在读取数据时，数据对象的新版本可以由它们之间的差异数据内容和旧版本联合生成，由于存在解码过程，因此读数据性能会受到一定影响。

Delta 编码过程主要分为相似性检验和编码。针对 Delta 编码的研究过程进行相似性检测的算法有很多，比如计算数据对象内所有固定分块中的相似哈希值，以此作为特征来比较和判断两个数据对象是否相似，或者基于 Shingling、Bloom Filter 和模式匹配等技术监测数据的相似性。能否准确、搞笑地判断数据对象之间的相似性，对于 Delta 编码技术的性能是否理想具有决定性作用。相比于相似性检验，有关相似数据编码技术的研究就相对较少，最常见的有三种实现：Unix 的 Diff、Bdiff、V delta，其中 Vdelta 将 Delta 编码和数据压缩技术，这也是 Delta 编码技术中十分常用的结合方式，V delta 的编码和解码效率明显高于其他，而压缩效率和 bdiff 不相上下，但是三种编码方式都需要扫描原始文件，在 V delta 的基础上出现了 Z delta，其消冗效果比 V delta 更理想。

1.2.3 重复数据删除技术

重复数据删除技术是近几年兴起的一种能消除粗粒度数据冗余的数据缩减技术，被广泛应用于基于硬盘的备份、容灾和归档系统中。重复数据删除技术最早由普林斯顿大学李凯教授提出，称之为全局数据压缩（Global Compression），并作为容量优化存储推广到商业应用。它通过将数据流划分为若干数据对象，并对各个数据对象进行哈希计算和基于指纹的索引查找检测出数据流中的重复数

据，只传输或者存储唯一的数据对象副本，并采用指向唯一副本的指针代表其他副本的方法消除冗余数据。根据对数据流进行划分操作粒度的差异，重复数据删除技术可以分为文件级、块级、和字节级重复数据删除，字节级重删即为 Delta 编码，在此主要介绍其他几种重删策略。

文件级重删也叫全文件分块（Whole File Chunking，WFC），以完整的文件作为分块粒度进行重删，应用全文件分块技术的存储系统称为单实体存储，包括微软的 Windows 2000 上的 SIS（Single Instance Storage）以及 EMC 的 Centera 在内的存储系统的都采用全文件分块技术实现重删，虽然全文件分块技术不能发现文件内部以及文件之间更小粒度的数据冗余，但是该方法简单有效，在适宜的负载条件下节省空间量能够达到最优块级划分策略的 3/4，适用于由小文件构成的数据集。

块级重删根据划分快的长度是否可变，分为定长分块重删和变长分块重删。定长分块（Fixed Size Chunking，FSC）又称为静态分块（Static Chunking，SC），以固定大小的数据内容作为粒度分割文件，被应用于 Venti 和 Oceanstore 等基于内容寻址的存储系统（Content Addressable Storage，CAS）。定长分块重删由于操作比较简单，因此效率较高，但是对修改、插入以及删除内容比较敏感，适用于更新操作少的静态数据集。Nath 等人的研究表明，在 CAS 系统中选用 1KB 或者 2KB 的数据对象最节省存储空间。

变长分块又称为基于内容分块（Content Defined Chunking，CDC），应用滑动窗口技术（Sliding Window Technique）基于文件的内容确定分割点，并通过计算窗口 Rabin Finger 值确定边界，在 LBFS 和 Pastiche 等系统中得到应用，该算法可以克服静态分块对数据更新敏感的缺点，重删效果较好。但是 CDC 算法也存在局限性，它划分的粒度绝大部分取决于期望块的设定，如果该值设置的较小，虽然粒度较细、重复数据查找较为精确，但是元数据开销很大；如果该值设置得较大，则粒度过粗，重复数据删除的效果不好，如何权衡精确查找和额外开销在 CDC 算法中是一个难点。

2 活动数据集的数据缩减

许多现代存储系统提供通常某种形式的数据缩减（如压缩、重复数据删除）。过去这些数据缩减方法主要应用于非活动数据中（如备份、容灾和归档系统），在活动数据中的应用较少。此外，各种数据缩减方法对活动数据集的应用效果目前也知之甚少。对活动数据集的数据缩减不仅提高存储效率，同时维持并潜在地改善在广泛的数据类型上的整体性能。

实验问题。本文我们将展示各种数据类型的大规模数据缩减实验的结果，尝试去解决的问题如下：1、单独的压缩还是重复数据删除是否足够好，或者是否需要结合两种方法使用；2、可以期望主动数据的数据缩减效率是多少；3、各种重复数据删除可以获得的相对收益是多少；4、将这些还原方法应用于数据的正确顺序是什么？

算法选择。由于我们专注于适用于存储系统的方法，所以我们选择了文件结构可能未知的通用方法。对于压缩，我们选择了基于 LZ1 的算法，因为它是广泛有效的。分别测试了两种类型的重复数据删除算法：基于块的重复数据删除和全文件重复数据删除（仅重复数据删除相同的文件）。

测试方法。我们对所有测试的估计方法是递归地扫描给定的一组文件目录，并且对于每个文件：根据正在测试的重复数据删除方法将文件内容分成块，同时计算各个块的散列并且执行单个散列的查找/插入一个哈希目录，并且压缩步骤一次将每个文件压缩到一个块，就像一个 1MB 的缓冲区一样。我们通过一系列不同的数据集来运行这些测试，每个数据集代表的数据环境（例如虚拟服务器，数据库，工作站）。

3 基于块的活动数据集数据缩减

本节比较了固定分块、变长分块与 LZ1 压缩的重复数据删除效果。目标是告知工程评估应该在如何最好地应用 I/O 栈数据简化方法，以及如何进行数据简化。分块大小设定为 4KB（固定分块为 4 KB，变长分块估计为 4 KB），接近主存储系统中经历的典型 I/O 请求大小。

固定分块设定在每个块长度（4KB）边界上分割文件。变长分块采用内容敏感的边界选择的方法（如 Rabin 指纹）。固定分块方法适用于通过插入和删除相关的文件系列，而变长分块涉及更多的工程难度，所以须估计它与固定分块相比的增量收益。

用户级压缩/重复数据删除工具实现了递归遍历文件系统目录，并通过组合压缩，固定大小的重复数据删除和可变大小的重复数据删除来应用数据缩减的七种变体。压缩使用了 gzip 中体现的 LZ1 算法。试验了下列 7 种数据缩减组合：

1、压缩（使用 gzip）

2、固定大小的重复数据删除

3、可变大小的重复数据删除

4、固定大小的重复数据删除，然后压缩

5、可变大小的重复数据删除，然后压缩

6、压缩，然后固定分块的重复数据删除

7、压缩，然后变长分块的重复数据删除

3.1 不同数据环境的缩减效率

不同数据环境中数据缩减效率由数据缩减因子表示，数据缩减因子=缩减数据量/原始数据量，如图 1 至 7 所示。缩减因子越小表示缩减效果越好。

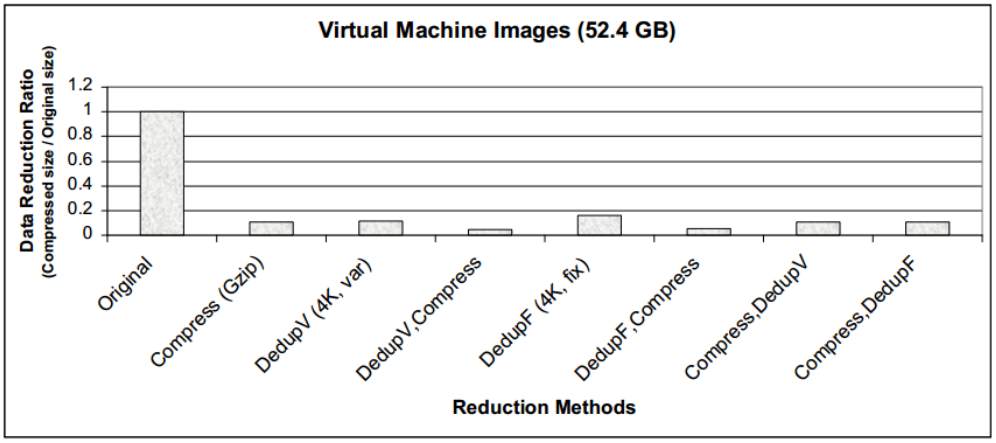

图 1：虚拟机映像

原始虚拟机映像是最容易被裁减的数据类型之一，压缩和重复数据删除（尤其是组合）都会产生重大影响。虽然单独的可变大小的分块在这里比固定的大小更有效，但包括压缩步骤消除了大部分差异。压缩之后的重复数据删除效果不如反之。

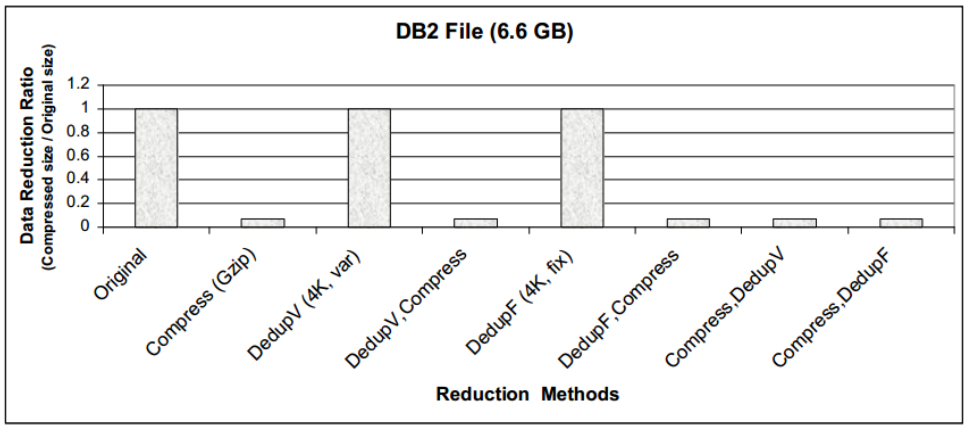

图 2：数据库文件

对于 DB2 数据库文件（不包含 DB2 级别的压缩文件并且主要包含文本数据使用重复数据删除效果较差，而使用 LZ1 算法可以压缩效果较好。

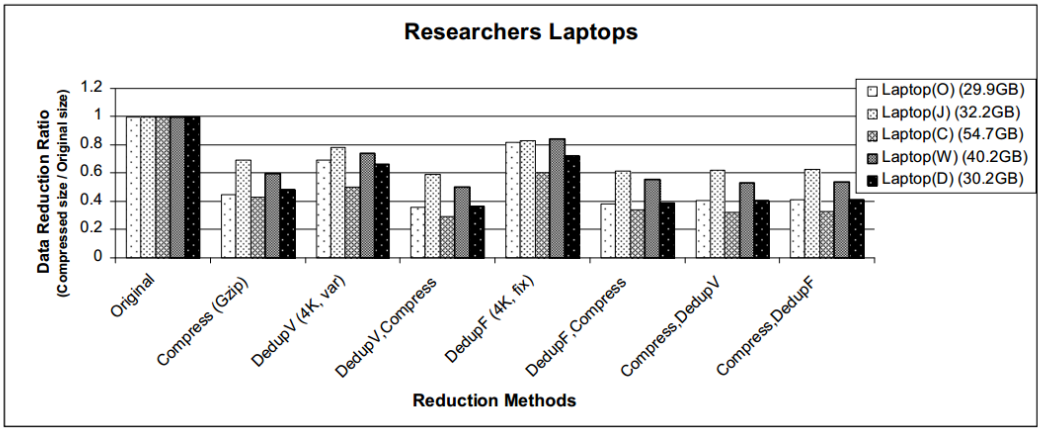

图 3：笔记本电脑

每个笔记本电脑映像包括操作系统（Windows），应用程序和数据。应用压缩和重复数据删除的顺序几乎没有区别，变长分块比固定分块更有效。

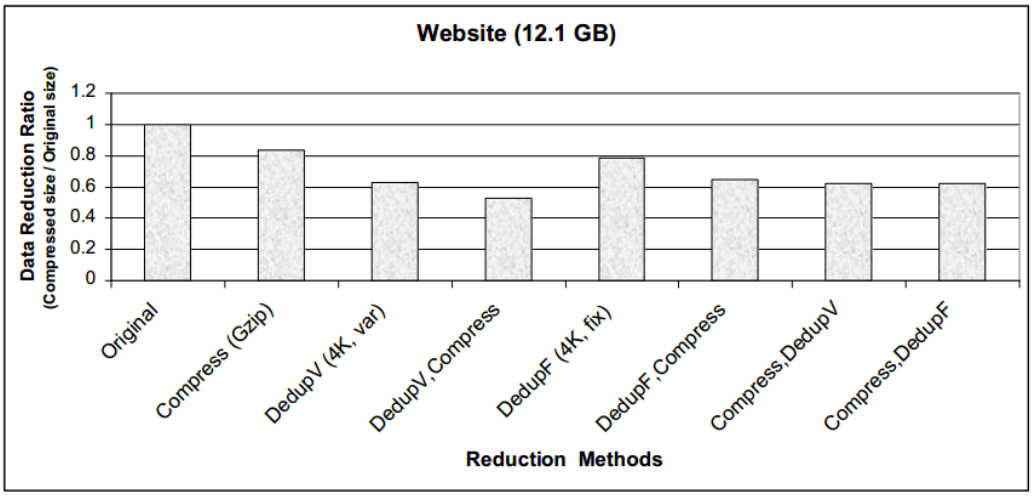

图 4：网站数据

这是“平衡”数据的另一个例子，其中压缩和重复数据删除都具有显著的影响，先应用重复数据删除再进行压缩，缩减效率更高。变长分块比固定分块更有效，但与压缩结合时有一些不同。

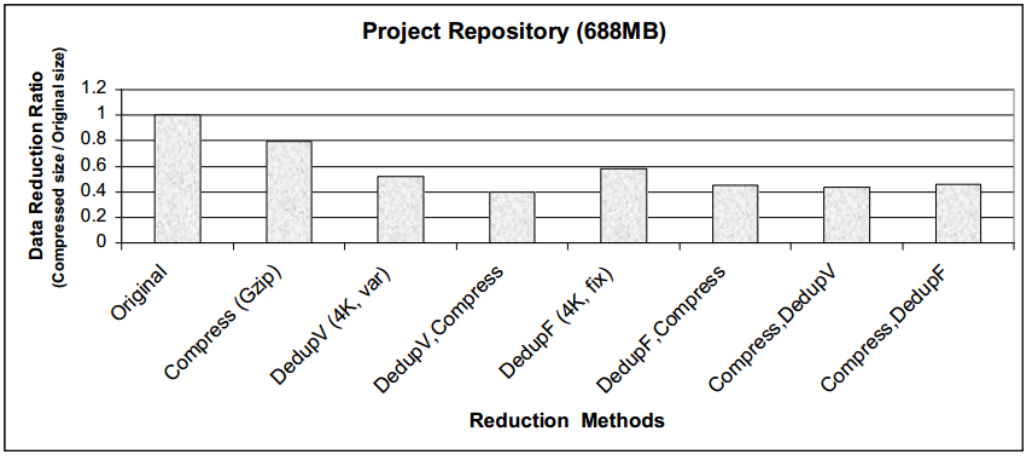

图 5：项目库

项目库中包含大量 Powerpoint、PDF、Word 文件以及 Java、C、LaTeX 源文件的 Subversion 版本库。项目库中重复数据删除比压缩更有效，变长分块的效果比固定分块更有效，但进行压缩后效果相差不大。

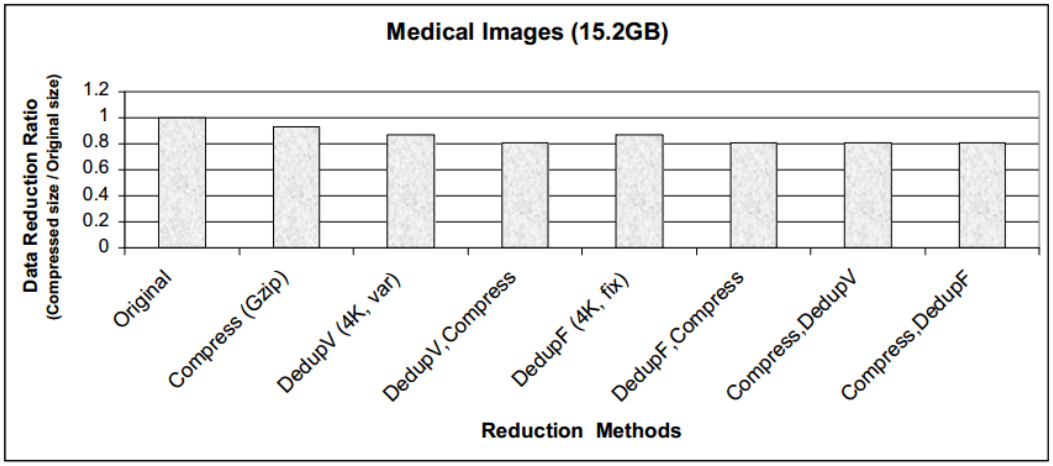

图 6：医学图像

医学图像是缩减效果最差的一种数据类型。它由小块的灰度医学图像组成（非常嘈杂），所以使用无损压缩没有发现太多的冗余。使用重复数据删除（包括变长分块和固定分块）技术的缩减量为 20%。

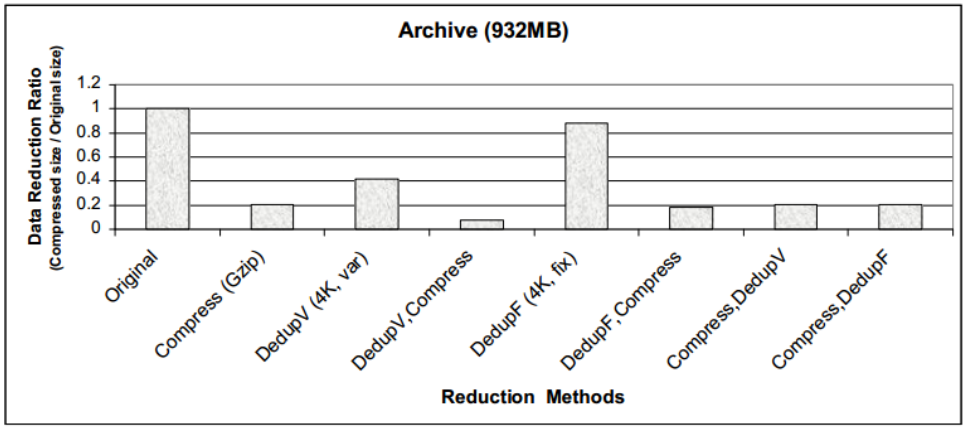

图 7：归档数据

归档数据通常包含某些类型数据（软件包，文档）的各版本。对归档数据来说，变长分块效果更好，而固定分块效果不理想。但由于数据集具有高度的可压缩性，无论是分块方法还是压缩方法都可以实现有效的数据缩减

3.2 总结分析

3.2.1、不同数据环境数据缩减效果对比

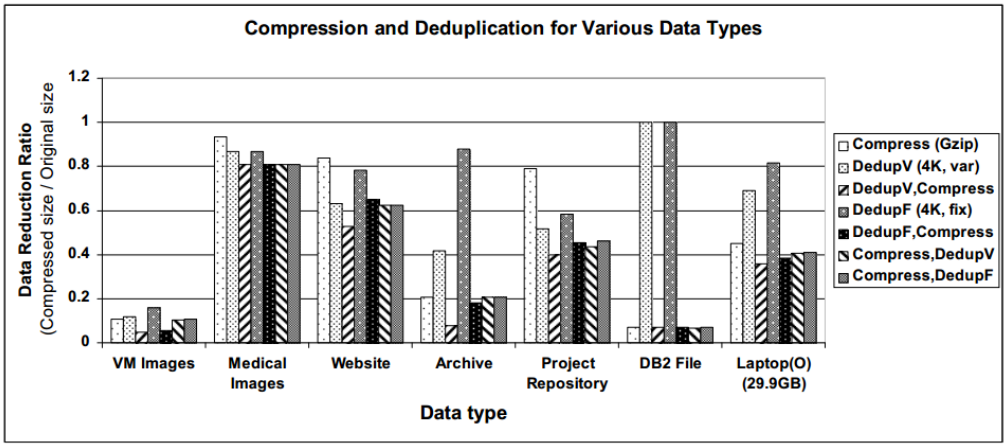

图 8：各种数据的压缩和重复数据删除结果

总结如下：

1、大多数预计可以实现 0.8 到 0.3 的数据缩减（即释放磁盘占用量的 20％到 70％）。

2、重复数据删除和压缩都为数据压缩做出了重大贡献，但在某些情况下，某种技术应用效果较差，某些情况下，两者效果几乎相同。

3、大多种数据类型中，变长分块比固定分块更加有效（例如 0.59 比 0.5），而当与压缩相结合时，二者之间效果差异不大（例如 0.33 与 0.29 相比）。

4、归档文件（例如.tar 和.zip）是上述一般性声明的一个例外。首先，变长分块比固定分块效果更好，在与压缩相结合时，变长分块效果仍然更好。

3.2.2 缩减效率总结分析

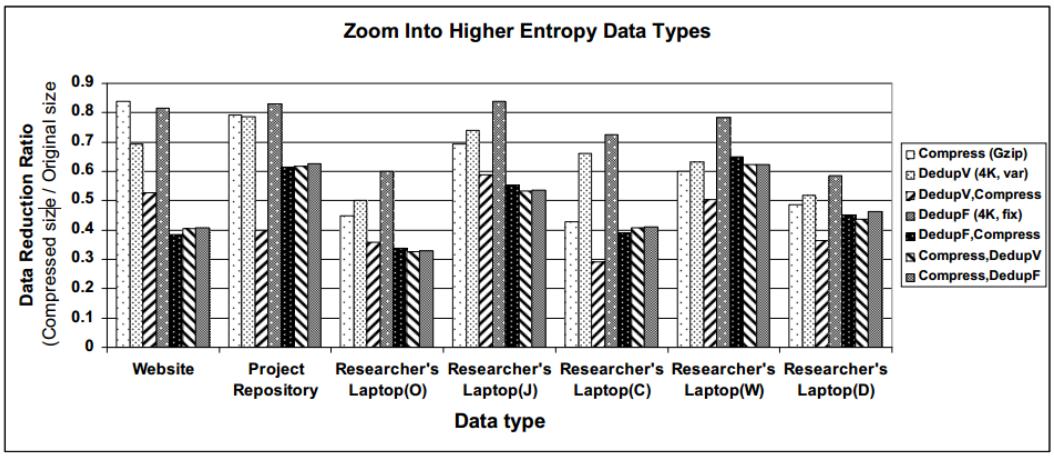

图 9：较少的可减少类型的数据压缩和重复数据删除结果

Technique	Data reduction range
Compression	0.42 - 0.84
Fixed-size dedup	0.59 - 0.84
Variable-size dedup	0.50 - 0.79
Compress-then-variable	0.32 - 0.62
Compress-then-fixed	0.32 - 0.62
Variable-then-compress	0.29 - 0.59
Fixed-then-compress	0.33 - 0.64

表 1：表格总结了图 9 中的可缩减数据较少的图表。

总结如下：

1、在进行重复数据删除之前是否进行压缩对缩减效果并没有显著影响。

2、压缩之前的重复数据消除有可能以较少的 MIPS 资源使用率实现相同的数据缩减。

3、具有高熵医学图像的数据集可能从数据减少中受益最少，但机会（20%）仍然很重要。

4 全文件重复数据删除和压缩

在文件系统中，消耗资源较少的重复数据删除方式是完整文件（FF）级别，将每个文件视为单个块进行 FF 重复数据删除分析，并与固定块长重复数据删除的结果进行比较。

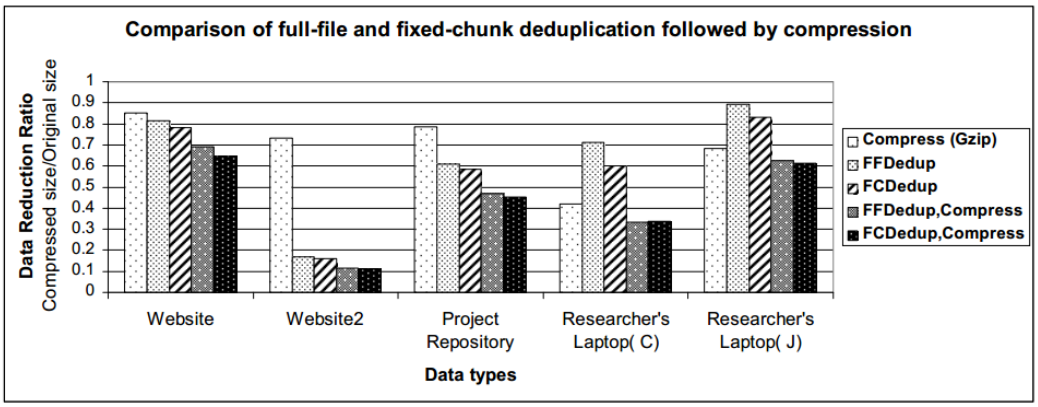

图 10：完整文件和固定块长重复数据删除，然后压缩三种类型的工作负载：Web 数据，项目存储和笔记本电脑数据的比较。

在图 10 中比较使用 FF 和 FC 重复数据删除技术实现的数据缩减，以及是否使用压缩技术。在所有案例中，FC 重复数据删除消除的大部分重复都是由于 FF 重复。FF 重复数据删除在网站数据（有许多媒体文件）和项目存储库上应用效果很好。对于某些数据集（例如笔记本电脑），FC 重复数据删除确实比 FF 更好，但与 FF 差异几乎被压缩步骤所消除。

FF 重复数据删除与 FC 重复数据删除相比，更具竞争力的一点是 FF 重复文件的元数据量要小得多。图 11 中列出的第二个文件是 FF 副本，FF 重复数据删除将其折叠为一个块的 20 个字节的元数据，而 FC 为每个 4KB 块保留单独的元数据，总计 107 个，226 个字节。

为了更好的对比 FF 和 FC 重复数据删除的效果，通过文件类型查看多种数据环境中的重复数据删除结果，各文件类型依次确定 FF 重复数据删除和 FC 重复数据删除的应用效果。分别创建以下三类文件：1、FFDup：FF 和 FC 重复数据删除

都有帮助；2、FCDup：FC 重复数据删除可以发现重复但 FF 不能；3、NoDup：没有发现重复。

不同的文件类型在三个类别之间有不同的分布，如图 12 所示。 例如，类型为.dll 或.exe 的二进制可执行文件（总计 17GB）分为大约 25％的 FFDup、50％的 FCDup、25％的 NoDup；.pdf 文件（超过 16GB）大多是 FFDup；.nsf 类别由图 11 中记录的一个 FCDup 文件支配，同时该文件是高度可压缩的。

Largest Files in LaptopC Data

File type	orig_size	compSize	FCDedupSize	FFDedupSize	FCDedup,compress	FFDedup,compress
.tar	920,852,480	189,279,501	809,106,466	920,852,480	165,159,682	189,279,501
.avi	512,870,004	99,235,083	107,226	20	107,226	20
.avi	512,870,004	99,235,083	499,376,416	512,870,004	96,919,805	99,235,083
.nsf	*147,062,784*	*415,795*	*309,315*	*147,062,784*	*274,161*	*415,795*
F	*134,217,728*	*2,700,830*	*6,889,806*	*134,217,728*	*2,582,912*	*2,700,830*
.tar	114,595,840	18,903,020	114,656,196	114,595,840	19,043,203	18,903,020

图 11：LaptopC 数据中最大的五个文件：压缩后的原始大小，压缩大小，全文和固定块重复数据删除大小和大小（\F“表示没有文件类型扩展名）。

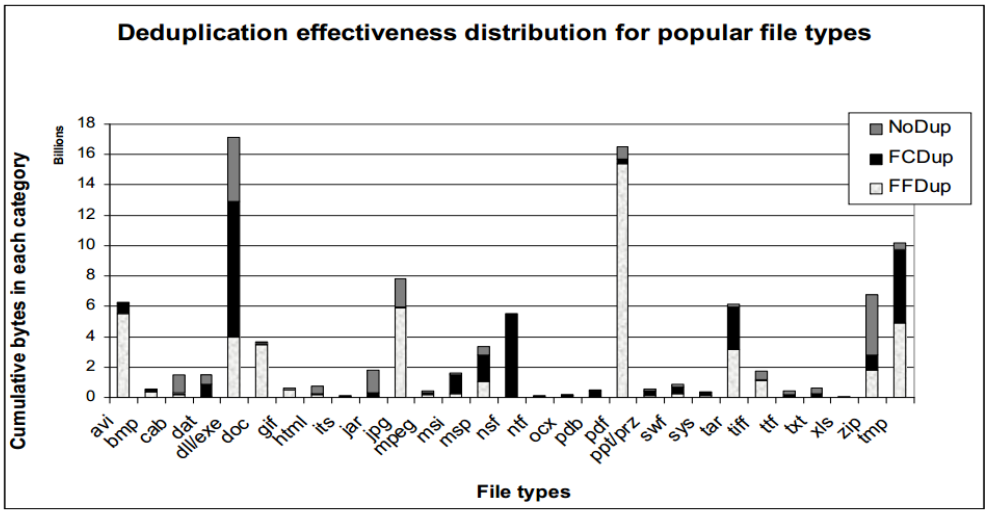

图 12：流行文件类型的重复数据删除效果

通过对全文件重复数据删除的分析，我们得出结论：**全文件重复数据删除（特别是与压缩相结合时），可以为许多数据环境提供非常有效的数据缩减**。

5 结语

5.1 结论

1、重复数据删除和压缩都为数据缩减做出了重大贡献，但在某些数据类型中，某一种技术效果不显著。

2、主动数据的数据缩减程度可以达到预期大多数主存储环境的成本节约。

3、在进行重复数据删除之前是否进行压缩对缩减效果并没有显著影响。

4、在很多数据类型中，完整文件重复数据删除与压缩结合使用时几乎与基于块的数据缩减效果相同，但在复合文件（如虚拟机映像，.tar，.zip，.nsf 等）中基于块的重复数据删除效果更好。

5.2 思考

1、准确预测各种数据缩减技术在大多数数据环境中的有效性将是数据中心选择缩减方法的关键。

2、对于特定数据类型而言，内容感知，结构感知或优化的算法的缩减效果是否更好值得研究。

3、对于主动数据而言，随机访问性能是确定数据简化解决方案的决定因素之一。因此，研究与各种活动数据环境相关的动态访问模式和顺序以及它们如何与在这些环境中存储和传输的数据内容交互至关重要。

致谢

本课程论文是在何老师的启发下完成的。何老师严谨的教学和工作态度给了我极大的帮助，尤其是对计算机存储原理和当下流行计算机技术的理解，让我能够进一步深入了数据存储据在计算机中的重要性和演变规律，让我对数据存储有了更深刻的理解，同时也感谢同学们给我的宝贵建议。