

**本科生毕业设计（论文）开题报告**

题　　目：基于透明压缩的数据布局管理方案

|  |  |
| --- | --- |
| 院 系 | 计算机科学与技术 |
| 专业班级 | CS1904 |
| 姓 名 | 胡志洋 |
| 学 号 | U201915043 |
| 指导教师 | 童薇 |

2023年2月

**开题报告填写要求**

1. 开题报告主要内容：
   * + 1. 课题来源、目的及意义；
       2. 国内外研究现况及发展趋势；
       3. 课题研究的内容和技术方案；
       4. 可行性与风险分析；
       5. 课题研究进度安排；
       6. 主要参考文献。
2. 报告内容用小四号宋体字编辑，采用A4号纸双面打印，封面与封底采用浅蓝色封面纸（卡纸）打印。要求内容明确，语句通顺。
3. 指导教师评语、教研室（系、所）或开题报告答辩小组审核意见用蓝、黑钢笔手写或小四号宋体字编辑，签名必须手写。
4. 理、工、医类要求字数在3000字左右，文、管类要求字数在2000 字左右。
5. 开题报告应在第八学期第二周之前完成。

# 课题来源、目的及意义

## 课题来源

随着SSD固态硬盘的发展， SSD在现在的硬盘市场上随处可见。固态硬盘（Solid State Drive），顾名思义就是用固态电子存储芯片阵列而制成的硬盘。SSD由控制单元和存储单元（DRAM芯片，FLASH芯片）组成，而FLASH闪存主要有两种类型，即NAND型和NOR型闪存。NOR型闪存通常部署为代码存储，因为它提供字节I/O和快速读取操作【1】。然而，就单位字节的成本而言，NOR型闪存比NAND型闪存更昂贵，因此NAND型闪存越来越广泛地用于大数据存储。但是相比于HDD，SSD的单位字节成本还是要更昂贵的。所以对于SSD有限的存储容量来说，应该得到有效的管理。

压缩技术的出现使得对于用户来说，同样数据现在可以使用更小的空间存储，提高了硬盘的耐久度和变相地减缓了由于多次擦除而导致响应速度的下降。压缩常被实现在三个层面，分别是应用程序、文件系统、存储设备的硬件。而现如今的两个技术趋势使得SSD上的压缩越来越普遍。一个是闪存的物理特性。闪存的存储单元像水桶储水一样保存电子，来表达和记录数据。反复的擦除和写入会逐渐对闪存单元形成不可逆转的磨损。这一工作原理决定了闪存会限制固态硬盘的寿命。另一个是摩尔定律，随着技术的不断发展，逻辑门尺寸的持续改进导致计算的成本越来越低。这两个趋势使得压缩这一技术变得越来越常见，越来越重要。研究证明实现在存储设备的硬件上的压缩效果是最好的[1]。对于主机端压缩来说，本层次实现的压缩会抢占CPU资源；对于文件系统的压缩来说，块设备跟文件系统的接口协议固定了传输数据块的大小，因此压缩产生的变长数据块会造成性能的下降，而且文件系统上频繁的压缩和解压也会提高读写的延迟，影响性能；对于压缩卡实现的硬件压缩，压缩卡有限的带宽是影响性能的瓶颈。这些问题都会随着压缩实现在SSD内部而解决。然而，与现有的用于NOR型闪存[4]、[5]的压缩技术不同，NAND型闪存中的数据压缩很难利用，因为它只支持页面I/O[6]。 如果按照原来的数据存放方式进行数据的存储，那么压缩带来的益处将会在很大程度上被削减。这是因为当压缩数据的大小小于页面大小时，NAND闪存上的数据页会出现内部碎片（没有被写入数据的部分），这严重降低了压缩的有效性。这个时候就需要一种合理的数据布局方式来充分利用压缩带来的用户更少的空间使用、更小的写放大等优点。

## 课题目的及意义

随着压缩越来越普遍和位置越来越偏向于底层，如果按照原本数据管理模式，SSD内部压缩后的变长数据块会使得NAND闪存面临数据页碎片化的问题。随着压缩的下移以及摩尔定律的失效，一种适合于压缩后变长数据块的数据布局管理方案将会越来越重要。本课题研究可以让我深入学习SSD的内部压缩，熟悉压缩的流程，并且寻找到一种合理的数据布局管理方案来管理SSD 内嵌压缩产生的变长数据块，降低压缩数据页内部的碎片化程度，充分发挥压缩所带来的优势。

# 国内外研究现况及发展趋势

现有的SSD内嵌压缩的数据布局方式有如下几种，CaFTL中chunk-based方案【2】、IPS方案【3】和compaction方案【1】。

CaFTL中chunk-based方案中使用大小为4个页面大小的chunk来作为写基本单元刷新到flash里面。它使用一个缓冲区来缓存写请求，按照I/O次序将写请求压缩然后放入chunk中，当chunk填满了就被刷新到闪存里面。由于一个chunk里面可能含有多个被压缩之后的数据页，因此该方案中除了页映射级别的FTL表之外还需要一个压缩后的数据块管理表Data Chunk Table来存储压缩后的数据（文中称为数据块）的信息。

IPS方案中同样设置有缓冲区，可以对于多个I/O单元分别进行压缩，并且在写缓冲区中临时维护一组压缩后的页。然后采用三种方法，分别是first-fit，best-fit，worst-fit对压缩后的页面进行分组。first-fit策略将压缩的页面存储到第一个有足够空间的组中。best-fit策略贪婪地将压缩的页面放到一个组中，该组在执行分组后会产生最小的内部碎片。worst-fit策略将压缩的页面放到一个组中，该组在执行分组后会产生最大的内部碎片。文章中通过实验证明best-fit策略是三种方案中最优策略。当由于容量限制而无法将传入的压缩页存储到任何写缓冲组中时，内部打包方案将对具有最小内部碎片的组之一执行写操作。

Compaction方案使用一个小页面大小的NV -RAM缓冲区，一旦每个主机写入的块从主机到达，它就会分别压缩。它将压缩块复制到NV -RAM缓冲区的末尾。如果在复制过程中缓冲区溢出，它会将完全满的缓冲区写入flash页面，清除缓冲区，并将压缩块的末尾复制到缓冲区的开头。Compaction方案想要充分利用flash的空间，每一个被压缩后的数据块都紧紧跟在前一个数据块的后面，不留缝隙。然而这样操作可能导致一个数据块被存储在两个flash页面上，而且这两个flash页面通常是不连续的，这也就导致本方案需要使用更为复杂的FTL映射表。

后面有人在IPS方案上提出了两种优化的方案，分别是重新排序方案【1】和IPS{our}【4】方案。重新排序方案是指当bin（大小为页面大小，用来对压缩后的数据分组）中没有足够的空间来存放下一个被压缩后的数据块时，对现有的压缩后的数据块进行重新排序，如果产生足够大的空间就将其插入，若不行则将bin刷新到闪存中。IPS{our}则类似compaction方案，将压缩后的数据连续存储，两个压缩后的数据块之间不留空间。

Compaction由于其具有最小的浪费空间（总是小于一个页大小）的特性被较为广泛的使用。国防科学技术大学的刘芳等人设计的FCTL方案【5】中也是使用了跟Compaction一样的思想，将压缩后的数据连续存放。FCTL中对于存在于一个物理闪存页的数据使用一个三元组来记录，而对于存在于两个物理闪存页的数据采用两个三元组来记录。Yifan Qiao等人在2022年中利用透明压缩提升B+ tree性能的相关工作【6】中使用的也是Compaction思想的数据布局管理方案。ScaleFlux研究团队在研究透明压缩来简化数据管理软件的工作中也使用了Compaction思想的方案。

# 研究内容与技术方案

## 课题研究内容

本课题研究存储设备上的硬件压缩，寻找一种合适的SSD内嵌的压缩方案。研究闪存页面中压缩后变长数据块的布局对于SSD的存储数据减少、SSD的寿命、SSD读写性能的影响。

## 课题预期目标

本课题研究可以让我深入学习SSD的内部压缩，熟悉压缩的流程。在学习和理解的过程中复现现有的工作，并且寻找到一种新的合理的数据布局方式来管理SSD 内嵌压缩产生的变长数据块，降低压缩数据页内部的碎片化程度，充分发挥压缩所带来的优势。

## 技术路线与方案

### 关键技术

（1）在服务器上配置和构建FEMU项目，使用FEMU模拟器来进行课题的研究，通过编写对应功能的代码实现对于压缩后数据管理的工作。

（2）在FEMU上复现IPS{best-fit}方案，即实现FCL层。

（3）在压缩模块里增加读取缓冲区模块来保证一定的I/O带宽。

（4）修改FTL里的映射表来实现对于变长数据块的管理。

### 总体方案

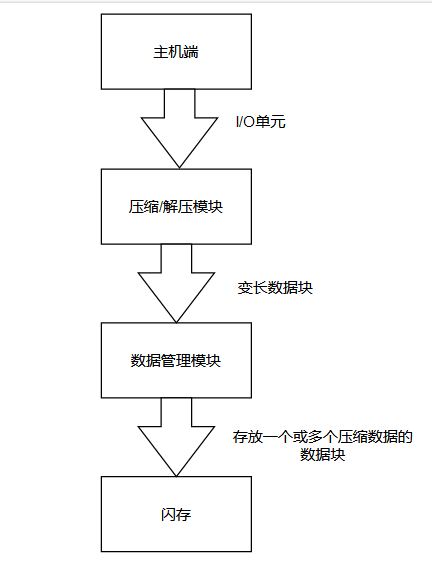
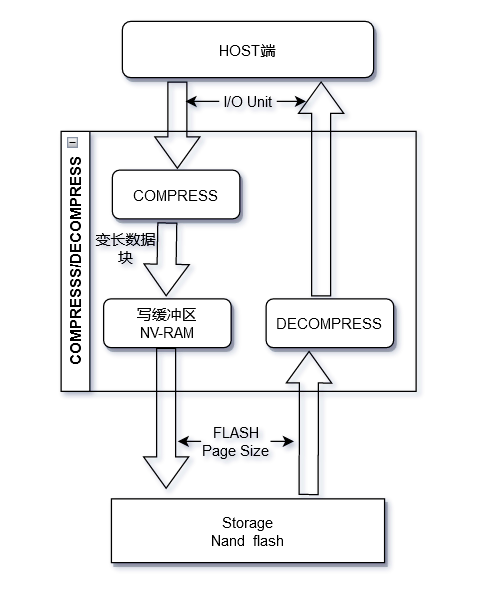


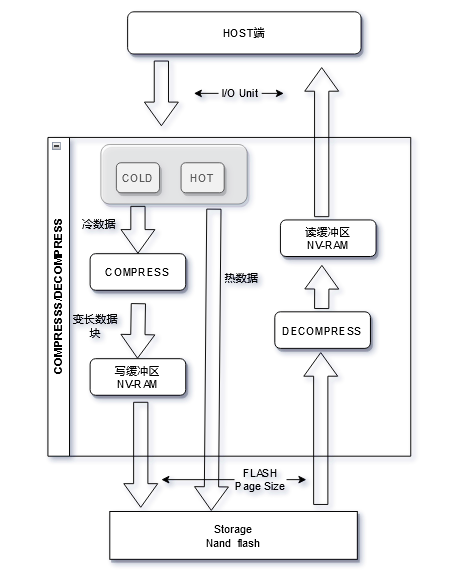
图1 系统总体架构图

目前对于本课题的构思架构如图1所示分为两个模块，一个是实现压缩模块，另一个是数据布局管理模块。

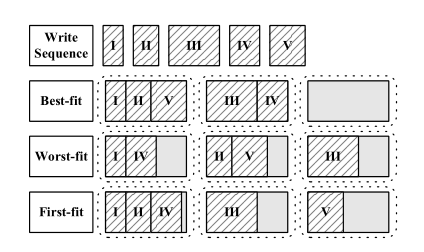


IPS方案的闪存压缩层的简易框图如图二所示。当主机发起写操作时，HOST端的数据被向下传入到压缩硬件里，然后数据被压缩成不定长的数据页，接着按照一定的规则分到缓冲区里的缓冲组里面，最后当缓冲区满的时候将数据刷新到闪存里面。读取时，从NAND闪存里的指定页的数据被取出，往上传到解压缩硬件里面进行解压，然后继续向上传到主机端。对于压缩模块来说，我打算选择\*\*\*\*算法，因为其具有良好压缩率、使用较少硬件资源、具有较快的压缩和解压缩速度等特点，符合本课题的研究需求以及有利于后续的优化工作的开展。

压缩模块里解压缩时可能会解压大量数据，因此引起读取性能降低。针对这个问题，我觉得可以对压缩模块进行一定的优化，例如可以增加冷热数据区分模块或者增加读取缓冲区模块。两种放大都是减少对于热数据解压缩次数来减少解压缩引起额外读取延迟。



对于数据管理模块，准备选择IPS方案作为本课题的基准方案。其中，IPS方案中对于压缩后数据页分到缓冲组的三种分组规则如图3所示。虽然Best-fit的分组策略具有最低的平均内部碎片率【8】，但是该策略会打破数据页之间的次序性，这会引起额外的延迟。以一个具有n页的数据文件（压缩率为50%）来作为例子，IPS方案压缩后数据页的分布类似于概率，因此连续排列的概率为p=1/n！。对于这n页数据的读取来说，最坏的情况下需要n个初始化延迟和n个读取时间。而如果按照文件原有的次序来存储数据页的话只需要n/2个初始化延迟和n个读取时间。IPS打破数据页之间的次序性而引起额外初始化延迟的问题同样可以通过设置读取缓冲区或者优化分组策略方式来改善。第一种方案的基本思路是我们可以将一个page里的所有压缩数据都解压缩到读取缓冲区里来减少多次解压缩而增加的初始化延迟。由于解压缩的速度是比I/O总线的速度要快，因此也不会因为额外的解压缩而增加过多的开销。第二种方案是在数据文件的次序性和闪存内部碎片率做出一种权衡，例如可以分组时采用Compaction的方案就可以保持文件的持续性【7】。实现上述的优化方案后，优化后的方案框图则如图4所示。



本课题打算在模拟器上来进行实现和验证。我选择了FEMU项目来作为本课题的研究平台。首先，我先在服务器上配置和构建FEMU项目，阅读项目代码，找到主机端跟存储I/O的模块。在两者之间先实现我们的简易方案，复现压缩方案IPS{best-fit}。如果有需要的话还需要修改FTL里逻辑地址到物理地址的映射关系。然后在原来的基础上实现IPS{best-fit}方案的优化方案。最后进行两种方案比较跟分析。最后期望可以提出一种新的、更合适的数据布局管理方案。

# 可行性与风险分析

## 制约因素与风险

这部分简要说明你在方案设计和实现、技术选择或平台选择等方面，为减少你所开发或设计的系统实施后可能存在的对社会、健康、安全、法律、文化及环境等方面带来的不利影响，你在设计中采取的相关措施。包括设置提示信息、进行身份认证、采用安全保密措施等等。这方面的内容大家过去可能关注的不多，相关的案例和设计思考可以在网上查到一些资料。

另外还需要给出本课题实施过程中可能面临的各种风险。

## 成本估算与可行性

这一部分首先根据相关模型评估一下课题的软件开发成本，根据成本估算以及前面分析的风险，如何应对可能存在的风险，整体评估项目可行性。

软件开发成本估算主要指软件开发过程中所花费的工作量及相应的代价。 不同于传统的工业产品，软件成本不包括原材料和能源的消耗，主要是人的劳动的消耗。另外软件也没有一个明显的制造过程，它的开发成本是以一次性开发过程所花费的代价来计算的。因此，软件开发成本的估算，应是从软件计划、需求分析、设计、编码、单元测试、集成测试到认证测试，整个开发过程所花费的代价作为依据的。常见的软件开发成本估算的经验模型有Putnam 模型 , COCOMO模型(constructive cost model) 等，大家也可选择其他你熟悉的模型，这里不需要详细介绍选用的模型原理，只需要说明你用的什么模型，为什么选这个模型，然后对其中各的参数结合你的项目进行分析说明，最后给出一个计算结果就可以！

### Putnam 模型 （阅读后删除）

1978年Putnam提出的，一种动态多变量模型。

L = Ck \* K1/3 \* td4/3 (3)

其中：

L:源代码行数(以LOC计)

K：整个开发过程所花费的工作量（以人年计）

Td：发持续时间（以年计）

Ck：技术状态常数，它反映“妨碍开发进展的限制”，取值因开发环境而异，见下表1.

表1 开发因素对CK值的影响

|  |  |  |
| --- | --- | --- |
| Ck的典型值 | 开发环境 | 开发环境举例 |
| 2000 | 差 | 没有系统的开发方法，缺乏文档和复审 |
| 8000 | 好 | 有合适的系统的开发方法，有充分的文档和复审 |
| 11000 | 优 | 有自动的开发工具和技术 |

从上述方程加以变换，可以得到估算

工作量的公式： K = L3/(Ck3\*td4) (4)

估算开发时间： td = [L3/(Ck3\*K)]1/4 (5)

### COCOMO模型(constructive cost model) （阅读后删除）

这是由TRW公司开发，Boehm提出的结构化成本估算模型。是一种精确的、易于使用的成本估算方法。COCOMO模型中用到以下变量：

DSI：源指令条数。不包括注释。1KDSI = 1000DSI。

MM：开发工作量（以人月计） 1MM = 19 人日 = 152 人时 =1/12 人年

TDEV：开发进度。(以月计)

　 　COCOMO模型中，考虑开发环境，软件开发项目的类型可以分为3种：

**1)组织型**(organic): 相对较小、较简单的软件项目。开发人员对开发目标理解比较充分，与软件系统相关的工作经验丰富，对软件的使用环境很熟悉，受硬件的约束较小，程序的规模不是很大（<50000行）

**2)嵌入型**(embedded): 要求在紧密联系的硬件、软件和操作的限制条件下运行，通常与某种复杂的硬件设备紧密结合在一起。对接口，数据结构，算法的要求高。软件规模任意。如大而复杂的事务处理系统，大型/超大型操作系统，航天用控制系统，大型指挥系统等。

**3)半独立型**（semidetached）： 介于上述两种软件之间。规模和复杂度都属于中等或更高。最大可达30万行。

估算公式：

基本COCOMO模型估算工作量和进度的公式如下

工作量： **MM = r\*(KDSI)c**

进度： **TDKV = a(MM)b**

其中经验常数 r, c, a, b 取决于项目的总体类型。

COCOMO模型按其详细程度可以分为三级：**基本COCOMO模型**，**中间COCOMO模型**，**详细COCOMO模型**。其中基本COCOMO模型是是一个静态单变量模型，它用一个以已估算出来的原代码行数(LOC)为自变量的经验函数计算软件开发工作量。 中级COCOMO模型在基本COCOMO模型的基础上，再用涉及产品、硬件、人员、项目等方面的影响因素调整工作量的估算。详细COCOMO模型包括中间COCOMO模型的所有特性，但更进一步考虑了软件工程中每一步骤（如分析、设计）的影响。

|  |  |  |
| --- | --- | --- |
| 总体类型 | 工作量 | 进度 |
| 组织型 | MM = 0.4\*(KDSI)1.05 | TDKV = 10.5(MM)0.38 |
| 半独立型 | MM = 3.0\*(KDSI)1.12 | TDKV = 10.5(MM)0.35 |
| 嵌入型 | MM = 3.0\*(KDSI)1.20 | TDKV = 10.5(MM)0.32 |

# 课题研究进度安排

由于本课题是一个研究性的课题，构思和实验的进度难以确定。大致的进度安排如表2所示。

表2 课题研究进度安排表

|  |  |
| --- | --- |
| **月份** | **工作任务** |
| 2022年1月  ~2022年2月 | 接受课题，搜索并阅读相关文献，准备开发测试环境 |
| 实现前人的算法，阅读代码和结果并从中获得启发 |
| 2022年2月  ~2022年4月 | 完成翻译和开题报告 |
| 自主学习课题涉及到的陌生知识，完成整体构思 |
| 2022年5月 | 编写代码反复实验验证效果 |
| 2022年6月 | 书写最终报告，准备最终答辩 |

# 主要参考文献

1. Mithun, N. C.; Paul, S.; and Roy-Chowdhury, A. K. 2019. Weakly supervised video moment retrieval from text queries. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 11592–11601.
2. Mun, J.; Cho, M.; and Han, B. 2020. Local-Global VideoText Interactions for Temporal Grounding. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2765–2775.
3. Nan, G.; Qiao, R.; Xiao, Y.; Liu, J.; Leng, S.; Zhang, H.; and Lu, W. 2021. International Video Grounding with Dual Contrastive Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2765–2775.
4. Zeng, R.; Xu, H.; Huang, W.; Chen, P.; Tan, M.; and Gan, C. 2020. Dense Regression Network for Video Grounding. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) ,2765–2775.
5. Lin, Z.; Zhao, Z.; Zhang, Z.; Wang, Q.; and Liu, H. 2020. Weakly-supervised video moment retrieval via semantic completion network. Proceedings of the American Association for Artificial Intelligence,2765–2775.

**华中科技大学本科生毕业设计（论文）开题报告评审表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | 胡志洋 | **学号** | U201915043 | **指导教师** | 郭德纲 |
| **院（系）专业** | | 计算机科学与技术学院CS1904 | | | |
| **指导教师评语**   1. 学生前期表现情况。 2. 是否具备开始设计（论文）条件？是否同意开始设计（论文）？ 3. 不足及建议。 | | | | | |
| 评 分： 指导教师（签名）：  2023年2月21 日 | | | | | |
| **教研室（系、所）或开题报告答辩小组审核意见** | | | | | |
| 教研室（系、所）或开题报告答辩小组负责人（签名）：    2023年2月25 日 | | | | | |