

本科生毕业设计

|  |
| --- |
| 基于生成对抗网络的闪存仿真器设计与实现 |

|  |  |
| --- | --- |
| 院 系 | 计算机科学与技术 |
| 专业班级 | 计算机1604 |
| 姓 名 | 孟嵩淼 |
| 学 号 | U201614613 |
| 指导教师 | 吴非 |

2020年06月10日

**学位论文原创性声明**

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密囗，在 年解密后适用本授权书

2、不保密囗 。

（请在以上相应方框内打“√”）

作者签名： 年 月 日

导师签名： 年 月 日

摘 要

理想的存储体系结构应该提供安全性、跨平台数据共享、高性能、以及可扩展性，然而当前广泛使用的三种存储结构（DAS，SAN和NAS）都不能兼顾所有这些需求。因此，工业界和学术界正在努力改变目前的存储技术，使存储设备从非智能的、外部管理发展成为智能的、自管理的设备，并且能够感知设备所服务的应用。基于对象的存储（Object-Based Storage）被认为是该问题的解决之道。

以SCSI OSD协议为基础，建立属性的两种传递机制。一种是调用专门的应用程序接口直接提供属性信息，存储设备根据属性直接做出存储策略选择。另一种是应用程序提供文件系统层的信息，设备内部根据文件的属性做出相关预测，并用于策略选择。比较而言，前者容易实现，而后者要求新的OSD文件系统，好处是不需要修改应用程序。同时还给出了三个属性页的定义作为OSD标准扩展提议。设计实现了符合T10 OSD标准协议的原型系统（iSCSI-OSD-RAID）。利用iSCSI协议作为OSD命令的传输层，加快了系统的实现过程。同时以一个正在进行的RAID控制器项目为基础，增加了新的对象存储管理模块，最终实现了OSD接口的磁盘阵列，并验证了属性控制的数据放置策略。实验测试结果表明….

**关键词**：生成对抗网络；闪存仿真器；闪存块错误数据；

摘要撰写说明，阅后焚毁！！

中文摘要是对论文内容的高度概括，应用精练的语言概述论文的主要研究内容、目的意义、设计过程、实验手段及取得的成果等。摘要一般分为2~3段，第一段简要介绍背景知识，尽量简洁，切中要害，不要说些任何人都知道的无信息量的语句；后面可以用一到两段介绍毕业设计工作，主要体现自己的工作，摘要不要太长，但主要工作部分应该比第一段背景知识长，不要本末倒置。

摘要中不得出现“本文共有X章，第一章…，第二章…”之类的表述。摘要严禁出现“本文”，“我”，“我们”这样的第一人称主语，尽量采用动宾结构，比如设计了….实现了…利用了….技术…..进行了….实验…..实验结果表明……，如果需要设置主语可以用文中涉及的系统或研究机制等。

摘要不要超过一页，关键词与摘要在同一页，数量3～7个。中文关键词须用汉字，尽量不使用英文单词或其缩写，例如“DBMS”不能作为中文关键词，必须用对应的中文表述：“数据库管理系统”。关键词之间用逗号分隔，最后一个关键词后不用标点。关键词应该具体，不得用过于泛化的词做关键词，如“音乐、视频”，形容词不能作关键词。

Abstract

Failure resilience is of pivotal importance in practical network function virtualization (NFV) systems, but has been mostly absent in the existing ones. The absence is mainly due to the challenge of patching source code of the existing NF software for extracting important NF states, a necessary step toward flow migration and replication to provide failure tolerance.

This paper proposes NFActor, a novel NFV system that uses the actor programming model to provide transparent resilience, high scalability and low overhead in network flow processing. In NFActor, a set of efficient APIs are provided for constructing NFs, with inherent support for scalability and resilience. A per-flow management principle is advocated, different from the existing practice, which provides dedicated micro service chain services for individual flows, enabling decentralized flow migration and scalable flow replication. We implement NFActor and show that it achieves good scalability, prompt flow migration and failure recovery with large numbers of concurrent flows. We also show that NFActor can enable applications such as live NF update and correct MPTCP subflow processing, which cannot be efficiently achieved in previous systems.

**Keywords:** Network Function Virtualization (NFV), Failure resilience, Actor model, Flow migration, Service Chain

撰写说明，阅后删除！！！

英文摘要在中文摘要定稿后再开始撰写，否则会多次修改，不要过于依赖百度翻译等工具。

Abstract是论文的英文摘要，一般对照中文摘要翻译，要求另起一页。Keywords对照中文关键词翻译，与英文摘要在同一页，以“Keywords：”另行顶左开始。Keywords之间用“,”隔开，最后一个Keyword后不用加任何标点符号。

目 录

[摘 要 I](#_Toc23945441)

[Abstract III](#_Toc23945442)

[1 绪 论 1](#_Toc23945443)

[1.1 课题背景 1](#_Toc23945444)

[1.2 国内外研究现状 2](#_Toc23945445)

[1.3 研究目的和主要内容 3](#_Toc23945446)

[1.4 论文结构 4](#_Toc23945447)

[1.5 课题来源 4](#_Toc23945448)

[2 方案论证（或具体背景技术概述） 5](#_Toc23945449)

[2.1 系统需求分析 5](#_Toc23945450)

[2.2 系统可行性分析 6](#_Toc23945451)

[2.3 开发工具分析及选择 6](#_Toc23945452)

[2.4 关键技术分析 6](#_Toc23945453)

[2.5 基本方案制定 7](#_Toc23945454)

[2.6 本章小结 7](#_Toc23945455)

[3 XXX系统设计 8](#_Toc23945456)

[3.1 功能需求 8](#_Toc23945457)

[3.2 系统总体设计 9](#_Toc23945458)

[3.3 功能模块设计 13](#_Toc23945459)

[3.4 本章小结 16](#_Toc23945460)

[4 XXX系统实现 17](#_Toc23945461)

[4.1 过滤器实现 17](#_Toc23945462)

[4.2 属性管理模块实现 18](#_Toc23945463)

[4.3 数据迁移模块实现 19](#_Toc23945464)

[4.4 本章小结 22](#_Toc23945465)

[5 性能测试与分析 23](#_Toc23945466)

[5.1 测试环境 23](#_Toc23945467)

[5.2 功能测试 23](#_Toc23945468)

[5.3 系统界面 23](#_Toc23945469)

[5.4 性能测试 23](#_Toc23945470)

[5.5 本章小结 25](#_Toc23945471)

[6 总结与展望 26](#_Toc23945472)

[致 谢 27](#_Toc23945473)

[7 毕业设计模板基本框架 28](#_Toc23945474)

[7.1 封面 28](#_Toc23945475)

[7.2 原创性声明页 29](#_Toc23945476)

[7.3 摘要 29](#_Toc23945477)

[7.4 目录 29](#_Toc23945478)

[7.5 参考文献 30](#_Toc23945479)

[7.6 附录 30](#_Toc23945480)

[7.7 毕业设计任务书 30](#_Toc23945481)

[7.8 成绩评定页 31](#_Toc23945482)

[8 毕业设计撰写要求 32](#_Toc23945483)

[8.1 图的格式 32](#_Toc23945484)

[8.2 表的格式要求 36](#_Toc23945485)

[8.3 公式 38](#_Toc23945486)

[8.4 流程图 39](#_Toc23945487)

[8.5 常见格式问题 39](#_Toc23945488)

[参考文献 41](#_Toc23945489)

[附录：大学期间发表或提交的论文 44](#_Toc23945490)

撰写说明，阅后删除！！！

目录不会自动更新，排版变动后必须更新目录，更新目录方式：鼠标点击目录，按F9键或者右键更新域，选择更新整个目录即可。

目录仅包含一级标题和二级标题和标题样式，目录是全论文的纲要。中文摘要、Abstract、论文正文的各级标题（不包括第三级）、致谢、参考文献、附录等都应编入目录，标注其页码对照关系，但目录本身不出现在其中。中文摘要、Abstract、目录等使用希腊数字“I、II、…”编连续页码；论文正文、致谢、参考文献、附录等使用“1，2，3，… ”编连续页码。

# 绪 论

本章介绍了本文的选题背景，说明了本设计课题的来源，目的，意义与应解决的主要问题及应达到的技术要求，分析了国内外研究现状及存在的问题。

## 课题背景

绪论主要介绍本文的选题背景：说明本设计课题的来源、目的、意义、应解决的主要问题及应达到的技术要求；简述本课题在国内外发展概况及存在的问题，本设计的指导思想等。（请在开题报告中完成第一章绪论部分的内容）

### 研究背景和趋势

生成对抗网络(GAN)是一种非监督式学习的方法，由一个生成器和一个判别器构成，生成器以一组随机取样值作为输入，其输出的数据需要尽可能的接近训练集中的真实数据。判别器接受真实数据或者生成数据作为输入，并需要区分真实数据与生成的假数据并输出输入数据的标签值。通过对生成器和判别器不断地进行训练，两者在动态博弈中达到平衡，最终生成器可生成足以媲美真实数据的假数据，使得判别器无法给出确切的判断。GAN网络通常用于生成以假乱真的图片数据，视频数据或者3D模型。

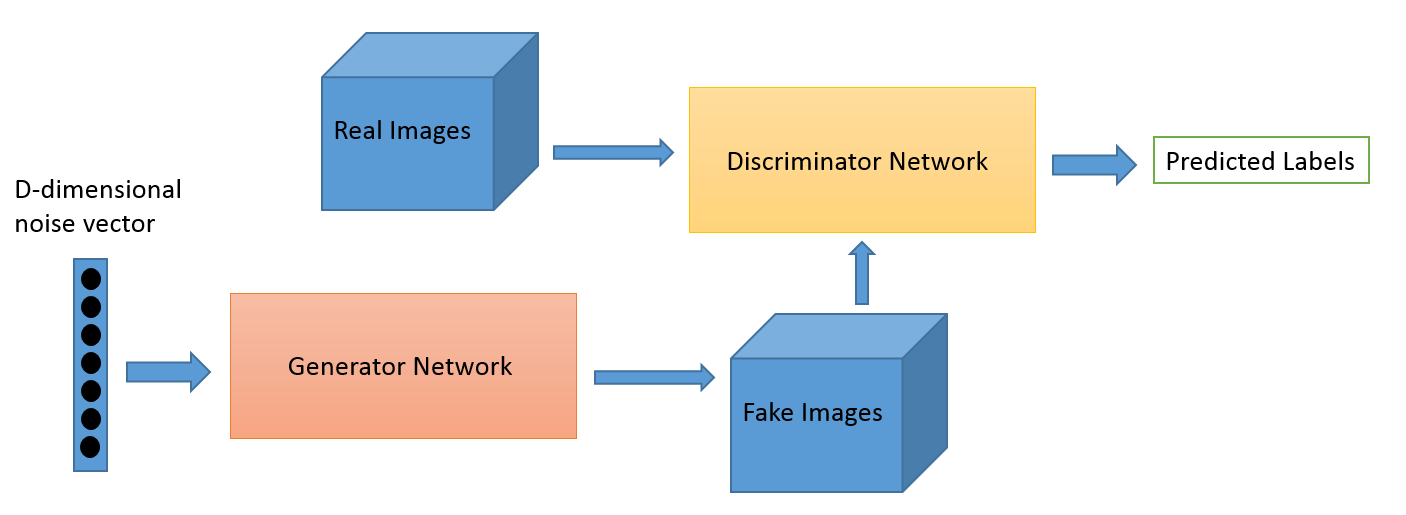


图1‑1生成对抗网络(GAN)

闪存在进行寿命预测以及其他工作的过程中，需要针对闪存在不同的测试条件下对闪存块的错误信息进行采集，这种针对真实闪存介质的测试与信息采集需要耗费较多的时间，因此考虑使用软件模拟仿真的方式来获得可以使用的闪存块错误信息数据。

### 面临的问题和挑战

测试所用的闪存，一个闪存块(Block)包含2304个页(Page)，每个页又被分为16个区域，对闪存块进行测试采样时，需要对每个块收集这230416个位置的错误次数，因此若想使用传统的方式对这生成一个块的2304个假数据较为困难。

原始的GAN网络本身则存在着许多问题，包括训练不稳定，需要小心的平衡生成器和判别器的训练程度，以防止其中一方的训练程度远超另一方（通常是判别器的训练程度较高）；生成器的生成样本容易缺乏多样性；并且生成器的结果由输入的随机值确定，因此输出结果具有随机性不易于控制。

## 国内外研究现状

### 生成对抗网络的改进与变种

生成对抗网络自提出以来便得到了广泛的关注，研究者们对原始的GAN网络进行了改进并产生了许多不同的变种，以适应不同问题情形下的需要。

Alec Radford等人提出了深度卷积生成对抗网络(DCGAN)，为GAN网络的一个变种。在判别器的内部使用了卷积神经网络来代替全连接神经网络，在生成器内部则使用了转置卷积进行上采样来生成假数据，并对网络内部的架构进行了精心的设计来解决原始GAN网络的问题，卷积网络的引入则提高了模型对于图像数据的学习能力。

Martin Arjovsky提出了WGAN从数学原理层面对GAN网络进行改进，它去掉了判别器的Sigmoid层，去掉了损失函数的log层，每次更新判别器的参数后将其截断到一个区间[-c, c]（c为一固定常数），并且不再使用基于动量的优化算法，最终解决了GAN网络训练不稳定的问题，增加了生成样本的多样性。

Jun-Yan Zhu提出了循环对抗生成网络(CycleGAN)，模型存在两个判别器分别针对数据域A和数据域B进行鉴别，同样存在两个生成器，分别负责将数据从A映射到B和从B映射到A。CycleGAN根据其特性常用于图像翻译转换工作，例如将普通马匹的图像转换为斑马的图像。

### 条件生成对抗网络模型简介

Mehdi Mirza等人提出了条件生成对抗网络(CGAN)，为GAN网络的一个变种，对于生成器，除了一组随机取样值*z*外，还需添加一组值*y*用于控制生成结果的类型；同样对于判别器除了以待判别数据作为输入外，也需要额外添加指定数据类型的标签值*y*。CGAN网络训练至收敛后，可以通过输入网络的标签值来控制产生数据的类型，有别于GAN网络随机生成假数据；

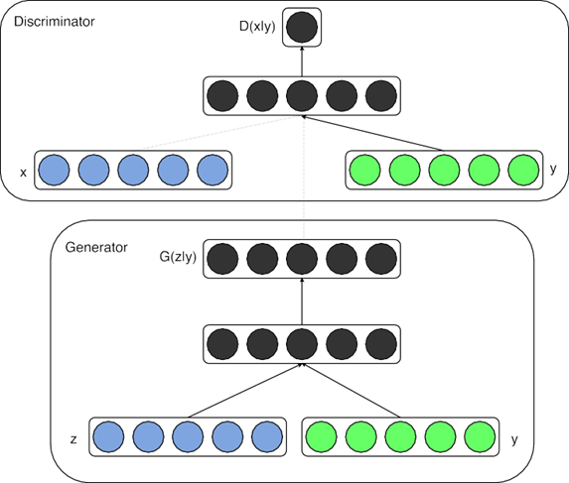


图1‑2条件生成对抗网络(CGAN)

## 研究目的和主要内容

本课题将基于生成对抗网络设计并实现一个闪存仿真器，用以仿真闪存的性能、可靠性等行为，支持不同干扰特性定量组合的闪存数据生成。

研究该课题的直接意义在于，它可以对不同特征进行定量组合，生成可媲美测试数据的结果，大大降低3D闪存测试的时间成本。例如对于闪存寿命预测等问题需要针对闪存进行大量的测试来获得真实的块错误信息，进行这些测试需要花费大量时间，其次当一个块的P/E次数达到一定量时，该闪存块就会损坏。因此如果能使用仿真器生成的数据代替真实采集的数据，就可以节省采集数据所花费的时间。

## 论文结构

本文的主要内容如下：

第一章介绍了本研究的目的和意义，介绍了GAN网络的定义，及其在不同应用条件下相关变种，以及如何将其应用于闪存块错误信息生成。

## 课题来源

该课题是本人在大四上学期在武汉光电国家研究中心实习时选择的课题，该课题由华中科技大学吴非带领研究。。

# 方案论证（或具体背景技术概述）

上一章对于论文的研究背景与趋势进行了分析，并阐明了本课题所要解决的问题。本章将对课题内容进行具体分析，并设计一种方案进行实现。

## 系统需求分析

块（Block）作为闪存的一个基本存储单元，它由2304个页（Page）组成，每个页又被划分为16个区域。块在多次P/E的过程中，这2304 × 16个位置均会产生一定量的错误。现对闪存进行测试，得到了一系列闪存记录文件，其中详细记录了块的在闪存中的位置，测试条件，以及块中每个位置测错误次数等信息。从这些记录文件中取得大量的块在不同P/E次数下的错误信息组织成真实数据集。要求通过GAN网络使用真实数据集进行训练，最终得到的生成器可以产生足以媲美真实数据的生成数据。

如图2-1，为真实数据的形式，记录了在一定P/E次数下，一个闪存块每个位置的错误次数，最终目标生成数据也为这种形式，即尺寸为2304×16的整形矩阵。

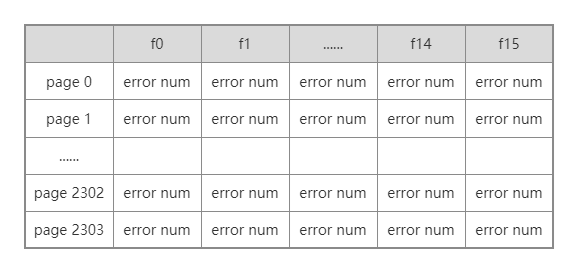


图2-1数据形式

## 开发工具分析及选择

目前较流行的机器学习的开源框架有Tensorflow，Pytorch，LightGBM，Keras等等。考虑到Pytorch接口定义简洁明晰，容易学习和掌握，且多用于学校的学术研究之中，因此选择Pytorch来构建整个系统，选择编程语言Python与IDE PyCharm。由于Anaconda3集成了大量的用于科学计算的工具，因此选择Anaconda3作为使用Python的一个基本环境。

## 基本方案制定

根据系统需求，需要在不同的测试条件下对闪存错误数据进行生成，因此应选择条件生成对抗网络（CGAN）作为系统构建的模型。

方案1：块错误数据维度为2304×16，判别器与生成器内部使用全连接神经网络，考虑直接将块错误数据平铺为一维数据进行输入，因此生成器的输出层与判别器的数据输入部分的尺寸为36,864‬（2304×16）。以这种最直接的方式处理数据进行训练，训练效果不佳，原因在于数据的尺寸过大，使用全连接神经网络导致模型拥有大量的参数使得计算压力过大，且生成器与判别器训练不均衡。

方案2：将原始数据处理为尺寸48×48×16作为输入，它相当于分辨率为48×48的16通道图像，块错误数据中的f0-f15分别为一个通道，而在page的维度2304组织成48×48。因此在CGAN的内部使用卷积神经网络，来处理这种具有图片特点的数据。

方案3：对每个块错误数据进行处理，统计出块中每页的错误总数，将所有的页错误数除以该块中的最大页错误数，得到该块所有页的相对错误分布（数据尺寸为(2304, )）。整个方案分为三部分：

（1）使用块的相对错误分布数据集对CGAN进行训练，使得最终的生成模型可根据条件（P/E次数）产生对应的相对错误分布。

（2）每个P/E次数都对应一个正态分布，使用该正态分布产生块的错误总数。

（3）整合（1）（2）得到块的各页错误总数，针对每一页错误总数，生成和为页错误总数的16个均匀分布的随机数，最终得到块每个位置的错误数据。

最终选择使用方案3，该方案的合理性将在后文论证。

## 本章小结。

本章分析了整个系统的需求，并从需求出发选择了要使用的开发工具与技术，并制订了多个方案进行对比，从中选择了一种方案进行实现。

# 系统设计

前面分析了系统的需求和方案，本章则根据方案的功能需求，对系统进行整体设计。

## 功能需求

（1）设计文本解析模块，能够对闪存的记录文本进行解析，从中提取闪存块的各项属性，测试条件，以及块每个位置的错误次数，并以页为单位导入数据库或者以二进制格式保存至本地。

（2）设计数据库连接模块，能够连接数据库服务器，能够从数据库中读取块错误数据等信息。

（3）设计数据统计模块，可以对原始数据进行统计与分析，根据统计分析结果辅助方案的设计。

（4）设计块相对错误分布生成模块，应用生成对抗网络，可以对真实数据集进行学习，训练后的模型能够生成媲美真实错误分布的假数据；

（5）设计块错误总数生成模块，在不同的P/E下，能够生成符合真实数据分布的块错误总数。

（6）能够对相对分布和错误总数进行整合，得到完整的块错误数据。

## 系统总体设计

系统整体设计如图3-1所示，原始的闪存块错误信息以文本文件的格式存储，首先通过文本解析模块将块错误信息从文本中提取出来，或导入数据库，或以二进制文件格式保存在本地。可以对这些处理得到的数据集进行统计与分析，得到数据的一些特征来辅助系统设计。对原始数据集进行处理来训练CGAN网络，最终可得到错误相对分布模型，同时块错误生成模块可以生成块的错误总数，最终整合这些信息，得到最终的块错误数据。

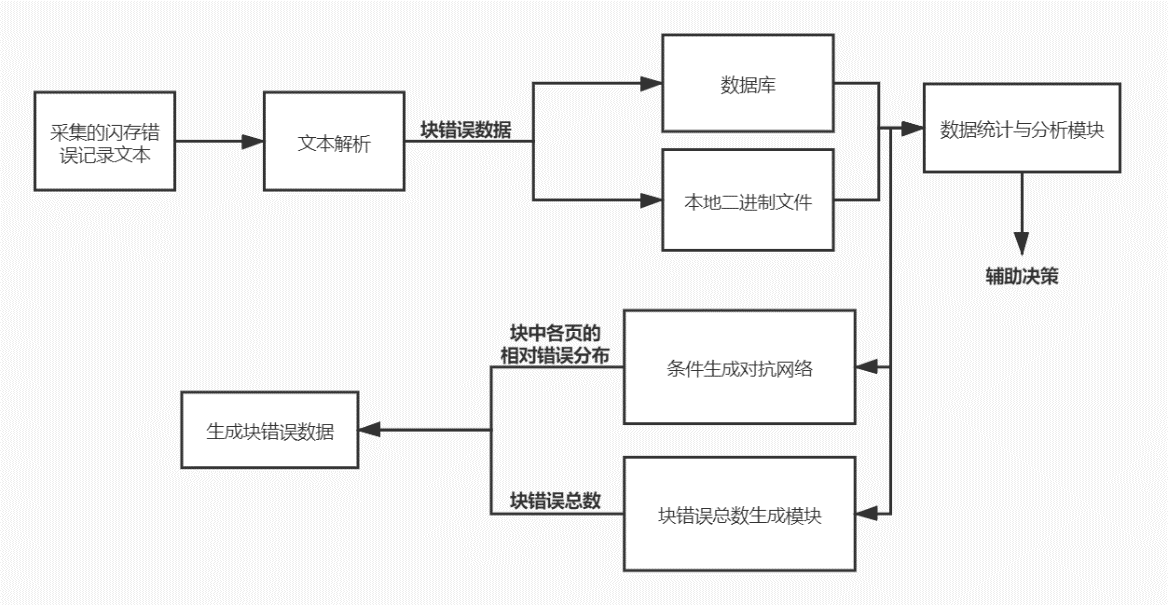


图3-1系统总体设计

## 本章小结

本章给出了系统的功能需求分析，说明了系统分为哪些模块以及各个模块的功能，并将这些模块联系在一起，阐明了系统的总体设计思路，并给出了系统总体设计图。

# 系统实现

上一章给出了系统的总体设计，本章将给出系统中各个模块的具体实现，包括文本解析模块，数据库连接模块，数据统计与分析模块，块相对错误分布生成模块，块错误总数生成模块。

## 文本解析模块

### 设计

*typedef struct \_DEVICE\_T{*

*ADD\_DEVICE\_FUN AddDeviceFun; //过滤器增加设备通知函数*

*DELETE\_DEVICE\_FUN DeleteDeviceFun; //删除设备*

*DELETE\_DEVICE\_FUN PrevDeleteDeviceFun; //预删除设备*

*DELETE\_DEVICE\_FUN CanDeleteDeviceFun; //设备是否可删除*

*struct \_list\_t{ //各层过滤器连接*

*void \*next; //该层过滤器的下层过滤器*

*void \*prev; //该层过滤器的上层过滤器*

*}list\_entry;*

*char MyData[1]; //该层过滤器私有数据*

*}DEVICE\_T;*

## 数据库连接模块

### 设计

## 数据统计与分析模块

## 块相对错误分布生成模块

### 数据集定义

由于块相对错误分布生成模块的目的为生成一个块内各页的相对错误分布，因此该模块所用到的数据集为块的页错误数及该块对应的P/E次数。图4-1展示了数据的格式，每条数据尺寸为(2304, )，即包含了每页的错误总数，共2304页。

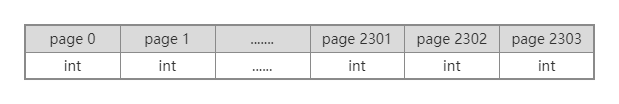


图4-1数据格式

为使模型学习到错误次数的相对分布，需要对数据集进行一定的处理。数据集中的每条数据均为尺寸为(2304, )的矩阵，记录了每页的错误次数，将矩阵中的每个值除以该矩阵中的最大值，意味着将错误次数矩阵转换为了错误相对分布矩阵。

矩阵中的每个浮点值表示了该页的错误次数相对于最大的页错误次数的比率，取值范围为[0.0, 1.0]，当某页对应的值为1.0时，表明该页的错误次数为其所在块内的最大值。

其次为方便模型进行训练，需对数据集进行归一化，处理公式为：

最终矩阵内元素的取值范围为[-1, 1]。

### 模型定义

根据前文，选用**条件生成对抗网络(CGAN)**来学习块错误相对分布。CGAN模型分为两部分，生成器与判别器。

**a. 生成器定义**

生成器的输入端接受一组随机噪音值*z*和该块的生成条件P/E次数（P/E次数需要经过归一化和正则化）。输出一尺寸为(2304, )的浮点型矩阵。其内部的具体结构如图4-2所示，为全连接神经网络，选择LeakyReLU作为激活函数，系数设置为0.2；在线性层与激活层之间添加批归一化层用于加快模型收敛速度，且在一定程度上缓解了梯度弥散的问题，使得模型的训练过程更加稳定。最终输出层选择Tanh函数将生成结果取值限制在[-1, 1]。

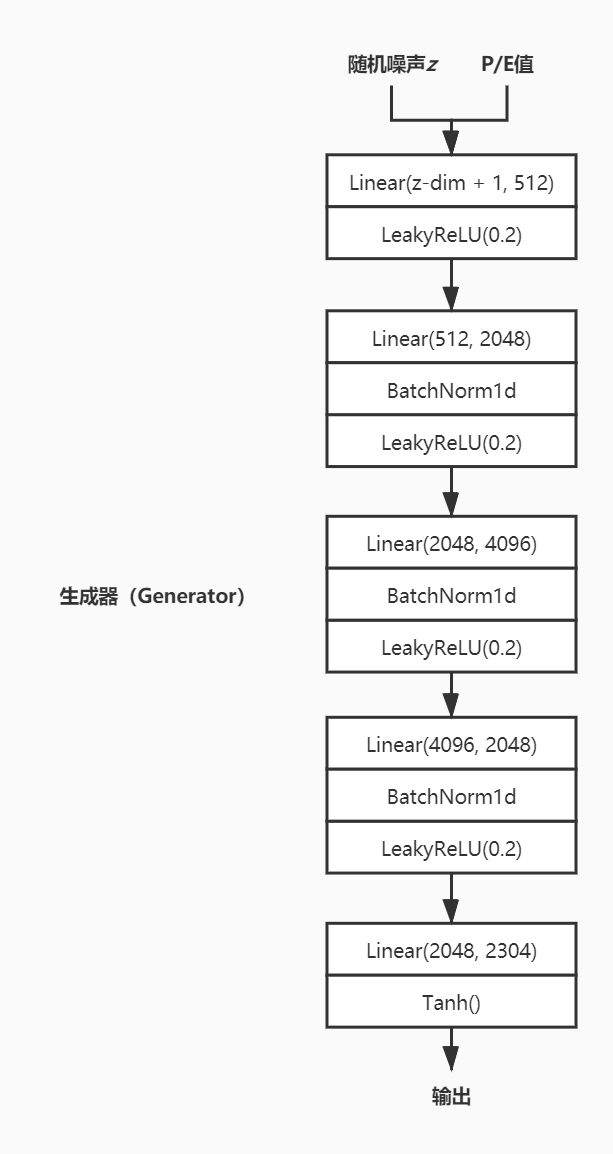
****

图4-1生成器

**b. 判别器定义**

判别器的具体结构如图4-1所示，其输入端接受块数据与P/E次数，输出判别结果为单个值，“**1**”代表为真实数据，“**0**”代表为假数据。参考Wasserstein GAN网络，去掉判别器最后一层的Sigmoid函数；在线性层与激活函数之间添加Dropout层来防止过拟合；由于在GAN网络训练过程中易出现训练不均衡的现象，通常是判别器的训练程度过高所导致，因此在设计时，判别器的层数与神经元数量要少于生成器，通过简化判别器来均衡训练过程。

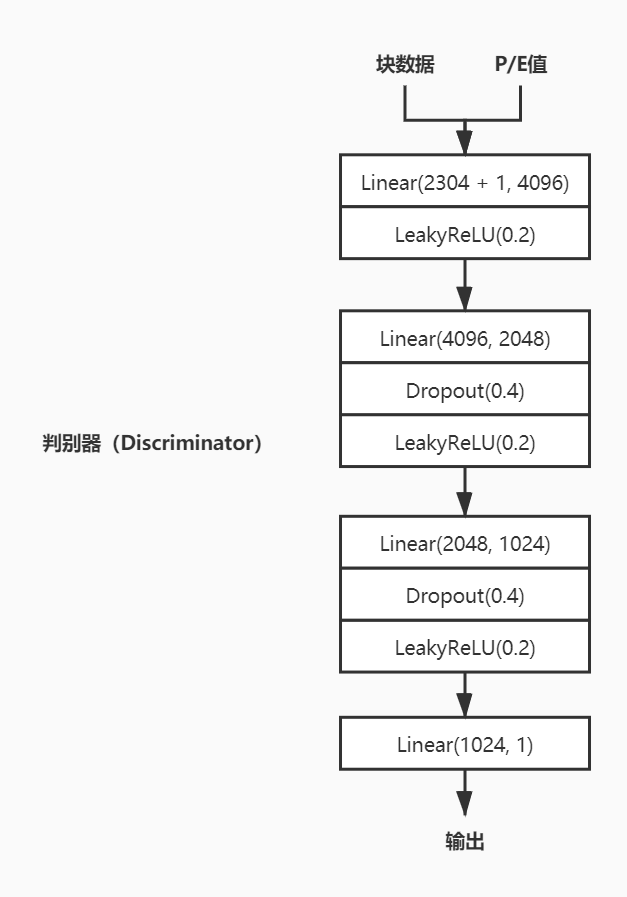


图4-1判别器

c. **训练过程**

模型训练流程如图4-x所示，整体思路为利用生成器生成的假数据与来自真实数据集的真实数据训练判别器，再从判别器对生成数据的输出结果反向传播，训练生成器，两者交替训练，直到完成训练轮次。生成器与判别器的优化算法均选用Adam算法，损失函数则使用均方损失函数（MSELoss）。

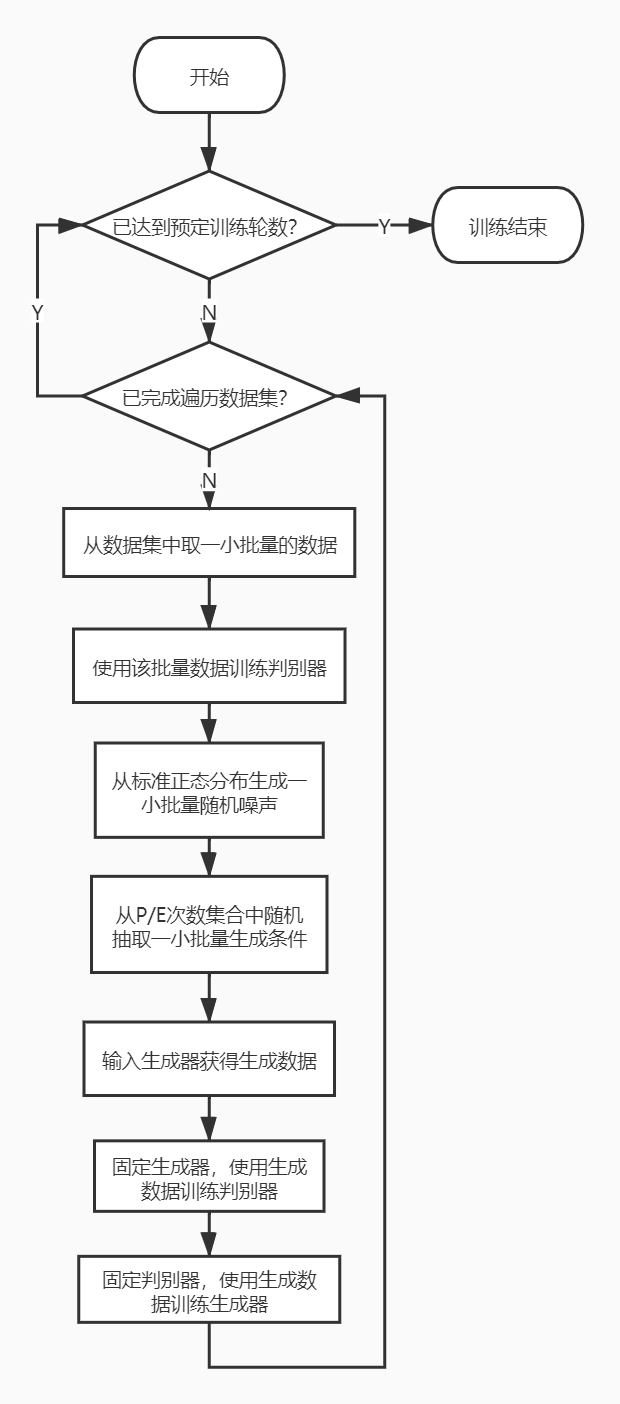


图4-1训练流程

## 块错误总数生成模块

根据数据统计与分析的结果，使用正态分布来拟合块的错误总数在不同P/E次数条件下的真实分布。

**a. 统计与记录**

对于拥有的全部块错误数据，按P/E次数进行分类，分为若干个集合，统计每个集合内块错误总数的平均值与标准差，并与其对应的P/E次数一同保存下来。

**b. 生成块错误总数**

由于数据集所收集的块记录对应的P/E集合为{1, 1000, 2000, … 16000, 17000}，或者是以其他间隔值来确定P/E集合，总之所统计的均值与标准差为有限个P/E对应的结果，例如若要针对P/E等于1200的条件进行生成，就无法从统计结果中获得确切的值。因此考虑使用三次样条插值，以统计所得的有限个点获得拟合函数，即以点集{, , … }进行插值，获得P/E🡪mean的拟合函数，标准差则同理；最终若要生成的P/E值不在统计范围内，则用拟合函数对应的值代替。

设某P/E值对应的均值为mean，标准差为std，根据之前的分析结果，该

P/E条件下使用正态分布来拟合块错误总数的真实分布。

使用Python中的标准正态分布来生成一组值*y*，则即为正态分布的对应值。

## 结果整合

根据以上设计，设通过块相对分布生成获得了一个块的各页相对分布矩阵

## 本章小结

本章。

# 性能测试与分析

## 测试环境

我们首先对RAID软件进行了评估。测试平台配置为：CPU Intel Xeon 1-8GHz，512MB内存，LSI Logic 21320-R (1030) ULTRA320 SCSI HBA，连接4块Seagate Cheetah Ultra320磁盘（MODEL ST373307LC），一块Agilent 5221A FC HBA（2Gb），与主机（启动器）的Qlogic 2310F FC卡通过光纤连接。为评价磁盘阵列控制器最终性能，我们利用Iometer软件对阵列控制器进行了详细的性能评测，具体测试环境见**错误!未找到引用源。**：

表5‑1 光纤通道启动器与阵列控制器配置

|  |  |  |
| --- | --- | --- |
| 配 置 | 启动器 | 阵列控制器 |
| CPU | Xeon 2.0 | Intel 80321 |
| Memory | 512MB | 1GB |
| Fiber Channel HBA | Qlogic 23102G | Agilent DX2 |
| OS | Windows 2003 | ARM Linux |

## 功能测试

## 系统界面

不是必须

## 性能测试

其中阵列控制器挂接6个Maxtor STM3320820AS磁盘，利用6个磁盘分别构建RAID0、RAID10、RAID5三种不同的阵列级别，分条大小为128K字节，缓存大小为512MByte，LUN大小为500GByte，阵列控制器采取写回策略。



图5‑1 磁盘阵列控制器RAID5顺序读写IOPS性能

**错误!未找到引用源。**为6个磁盘构建RAID5时阵列控制器顺序读写IOPS(I/O Per Second)性能示意图，从图中可以看出，IOPS随着数据块大小变大而变小，当数据块大小为512Byte时，阵列控制器处理读请求的性能可以达到每秒15000个，写性能由于包含冗余校验信息计算，IOPS略低，约为12000，此指标高于同类美国阵列nStor 4502F，其写IOPS最大为7000左右。



图5‑2 磁盘阵列控制器RAID5读写性能

为6个磁盘构建RAID5时阵列控制器数据传输率示意图，从图中也可以看出，随着主机端I/O请求数据块大小逐渐增大，阵列控制器数据传输率逐渐增加，当数据块大小达到128Kbyte时，系统性能最佳，读性能可以达到185MByte/s，写性能约为167MByte/s。**（有坐标轴的图必须给出单位，坐标轴意义，所有测试图应该有详细分析）**

从总体性能评测结果看，本文最终设计的FC-SATA磁盘阵列控制器基本发挥了Intel IOP处理器的最佳性能，同比性能优于同类进口磁盘阵列，基本达到了设计要求，如需要更高的数据传输率，必须同时提升主机通道以及IOP本身的性能，才能有实质性的提高。

## 本章小结

本章给出了控制器完整的软硬件设计方案，提出了主机板与通道板分离的架构，可以灵活构建全系列阵列控制器产品线；提出了磁盘阵列软件分层异步体系结构，可以有效提高I/O请求处理的并行度，减少I/O处理同步等待时间；最后提出了一种针对阵列层次性结构的细粒度测试方法，可单独测试评价各模块性能，为系统性能精确评测，以及故障诊断提供决策依据，最后给出了阵列控制器性能综合评测。

# 总结与展望

基于对象的存储是为了克服当前基于块的存储存在的诸多难题，在存储接口和结构层次的重要发展。基于对象的存储设备可以具有更多智能性，可以根据应用负载选择优化的存储策略。实现属性控制的存储策略，关键需要解决两个方面的问题：第一，属性表达什么信息，怎样影响存储策略？第二，属性怎样从用户应用传递给存储设备？围绕这两个问题，作了如下几点研究和工作：

1. 提出了对象属性控制存储策略的模型。OSD协议作为SCSI的扩展集，可以在TCP/IP网络和iSCSI协议之上来传输，基于对象的存储设备接收OSD命令并进行处理。
2. 结合对应用负载特征的分析，提出根据对象属性描述的负载特征动态选择RAID级别和分条单元大小的数据放置策略。I/O请求较小的对象放在RAID-10中，同时分条单元尺寸较小；反之则放在RAID-5中。
3. 建立了属性的传递机制，一种是应用程序接口静态传递；另一种是动态机制，即根据属性当中包含的文件系统信息预测文件的分类，并根据分类做出进一步的存储策略选择。
4. 论文总结必须用这样的条目的形式给出，总结是对全文的总结，不是对毕业设计的心得，心得部分不要写。

本文为更好地认识基于对象的存储以及OSD的实现进行了有益的探索。该领域还有很多研究工作有待完成，包括：

* 以OSD为节点构建大规模（PB级）存储系统
* 属性控制的caching/prefetching策略
* 实现Device-aware（表达设备能力，按需分配资源）
* 设备的学习能力，以文件系统和历史访问信息等为提示进行预测

随着存储需求的爆炸性增长，下一代互联网络必须有新的存储结构来应对构建和管理更大规模存储系统的挑战。基于对象的存储是最有希望的突破点之一。相信存储技术的发展一定能够满足人们不断增长的存储需求。

致 谢

论文完成之际，首先要感谢我的导师吴非教授。他站在学科发展的前沿，从论文的选题，研究工作逐步深入，到论文的撰写，都给我以细致的指导和建设性的意见，使我得以圆满而顺利地完成……。XXX严谨的治学态度、诲人不倦的师德和一丝不苟的工作作风将会给我留下不可磨灭的记忆……。

在…… 四年里，得到了许多老师和同学的大力帮助和支持，在此表示深深的谢意。感谢XXX等老师对我的关心和帮助。在课题研究和项目开发过程中，我与很多同学一起度过了一段段难忘的时光，值得怀念。我们大家共同创造的良好的学术氛围，将给我以永远而美好的回忆。

……

最后，我要深深地感谢我的父母，他们给予我无尽的关怀和无止境的爱，….

致谢属于论文的辅文部分。使用第一人称，采用散文体，对指导教师以及协助完成设计的有关人员表示谢意，并可简述自己通过本次毕业设计的体会，注意只写是查重最容易出问题的地方，请千万不要看别人写的，照搬。

# 毕业设计模板基本框架

2015年3月华中科技大学教务处发布了的《关于进一步加强本科生毕业设计（论文）规范化的通知》，该文对毕业设计开题报告，译文翻译，毕业设计论文的撰写提出了具体的规范化要求，该规范全部采用文字描述，实际操作有些困难，为方便计算机科学与技术学院全体学生撰写毕业设计论文，特开发此计算机科学与技术学院毕业设计标准模板供大家使用，本模板是对学校教务处标准的实例化，整体基本遵循学校标准，但也适当进行美化，如与学校教务处标准有冲突，请以此模板为主。

## 封面

本模板中封面中所有需要用户填写部分全部采用插入文档部件中的文档属性完成，如图7‑1中的姓名显示为作者，在开题报告中、成绩评定页中均包括学生姓名，采用文档属性的方式可以做到一次修改多处同时变化，避免数据重复输入和不一致性，注意不要删除文档属性框另外输入，日期部分不需要修改，会自动更新为正确时间。



图7‑1 封面模板中的文档属性

## 原创性声明页

最终定稿的毕业设计论文需要提交正式胶印版一份，双面黑白打印，这个版本需要将所有签字完成，其他未定稿的版本均不需要签字。学位论文原创声明页需要作者本人签名两次，提交导师签字之前请学生本人将自己的签名完成，再找导师签字，否则非常不礼貌。

## 摘要

中文摘要是对论文内容的高度概括，应用精练的语言概述论文的主要研究内容、目的意义、设计过程、实验手段及取得的成果等。摘要一般分为2~3段，第一段简要介绍背景知识，尽量简洁，切中要害，不要说些任何人都知道的无信息量的语句；后面可以用一到两段介绍毕业设计工作，主要体现自己的工作，摘要不要太长，但主要工作部分应该比第一段背景知识长，不要本末倒置。

摘要中不得出现“本文共有X章，第一章…，第二章…”之类的表述。摘要严禁出现“本文”，“我”，“我们”这样的第一人称主语，尽量采用动宾结构，比如设计了….实现了…利用了….技术…..进行了….实验…..实验结果表明……，如果需要设置主语可以用文中涉及的系统或研究机制等。

摘要不要超过一页，关键词与摘要在同一页，数量3～7个。中文关键词须用汉字，尽量不使用英文单词或其缩写，例如“DBMS”不能作为中文关键词，必须用对应的中文表述：“数据库管理系统”。关键词之间用逗号分隔，最后一个关键词后不用标点。关键词应该具体，不得用过于泛化的词做关键词，如“音乐、视频”，形容词不能作关键词。

## 目录

目录不会自动更新，排版变动后必须更新目录，更新目录方式：鼠标点击目录，按F9键或者右键更新域，选择更新整个目录即可。

目录仅包含一级标题和二级标题和标题样式，目录是全论文的纲要。中文摘要、Abstract、论文正文的各级标题（不包括第三级）、致谢、参考文献、附录等都应编入目录，标注其页码对照关系，但目录本身不出现在其中。中文摘要、Abstract、目录等使用希腊数字“I、II、…”编连续页码；论文正文、致谢、参考文献、附录等使用“1，2，3，… ”编连续页码。

## 参考文献

参考文献必须25篇论文以上，尽量避免网页链接引用。英文10篇以上。

## 附录

附录部分非必须内容，如果有一些需要特别交代的内容可以在这里列出，如本人参加的项目，提交的论文等，如没有特别闪光的地方建议删除本章。

## 毕业设计任务书

### 任务书封面

毕业设计任务书封面内容均可根据本模板封面数据自动生成，无需没写，但如果题目超过两行，请自行输入题目到两行中，避免出现图7‑2中尴尬的情况，该封面中所有时间都已经固定好，不需要修改。



图7‑2 任务书封面标题过长

### 任务书内容

课题内容和要求请按毕业设计互选系统中的内容酌情修改完善后填充，内容尽可能丰满完善一点，尽可能的细化，不要干瘪的一句话，参考文献部分应列出最终论文中最重要的5~6篇参考文献，尽可能的将表格充满，注意列表格式应与原格式一致。如有同组设计者请填写姓名，没有填无。

### 任务书签字流程

任务书封面部分需要指导老师所在单位教学负责人签字审查，并经教学院长批准，请各班集中收集任务书封面单页后，到教务科集中签字盖章审核，为减少学生和相关负责人工作量，原则上不针对单个学生进行该项业务。

计算科学理论研究所： 石柯 并行分布式计算研究所： 陆枫

信息安全研究所： 付才 数字媒体研究所： 李丹

数据存储研究所： 谭志虎 数据工程研究所： 潘鹏

## 成绩评定页

成绩评定页需要导师和答辩组就论文和答辩情况进行综合评定，参照硕士论文答辩流程，请学生**事先草拟导师意见**，提交导师参考手写到评定页，评分并签字。**答辩时请携带答辩评审意见空白页**（含姓名，班级，打印论文时单独取出，不要装订）请**答辩评议组当场给出评审意见**并签字打分。

# 毕业设计撰写要求

## 图的格式



图8‑1 图注必须在图的正下方

正文中所有插图要求图面整洁，布局合理，线条粗细均匀，大小适中。插图必须是矢量图，不能是从他处直接拷贝位图，所有曲线、图表、线路图、流程图、程序框图、示意图必须按国家规定标准或工程要求采用visio等工具绘图后复制到word。插图在论文中采用的是正文样式，所以会自动缩进，请调整标尺取消缩进，让插图严格居中，图表中的文字大小必须小于正文字号。

### 图的题注

所有插图均应有图号和图名（office中称题注，图注），图注出现在插图的正下方，小四黑体居中，在本模板中可以直接使用样式库中的题注样式即可规范格式，图号和图名后面不加标点符号，也不得加参考文献引用。

图号按章编编号，如第2章的图为图2-1、图2-2、…，第3章的图为图3-1、图3-2、…等，第一个编号是章号，第二个编号是本章的图的序号，常见错误是出现二级编号如：图2-1-1；第二个图名是插图的名称，扼要概括图的内容，字数不宜太多。

图的题注可在引用菜单栏通过插入题注的方式插入**（最简单的方法是复制其他图的图注然后更新域即可）**，引用时采用交叉引用引用，注意题注，交叉引用是word很重要的概念，是office排版的高级技巧，撰写毕业设计论文时应研究学习一下此功能，如果插入题注没有对应的图题注，只有figure等，可以自己创建，具体流程如图 8‑2，图 8‑3所示，需要注意的是，由于office不是为中文排版设计，图编号前面会增加一个空格，建议手工删除。

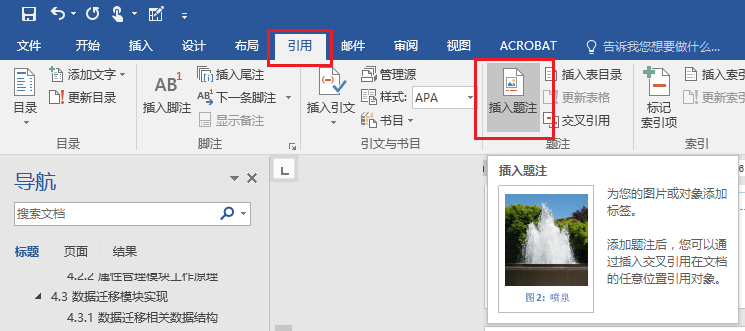


图 8‑2 插入图的题注

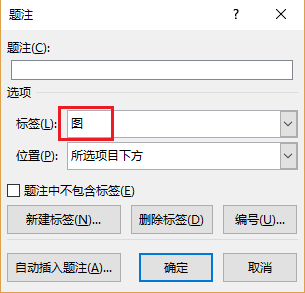


图 8‑3 插入题注对话框

### 图的引用

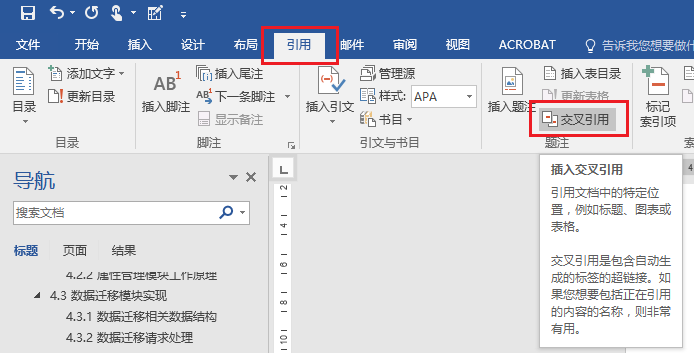


图8‑4 交叉引用

所有插图均应在正文中予以引用。引用某插图时，一般写为“…如图2-1”或“图3-2是…”。正文中的插图均须安排在文中第一次引用到该图的正文下面，一般要求先见文，后见插图，且图一般不跨页绘制。

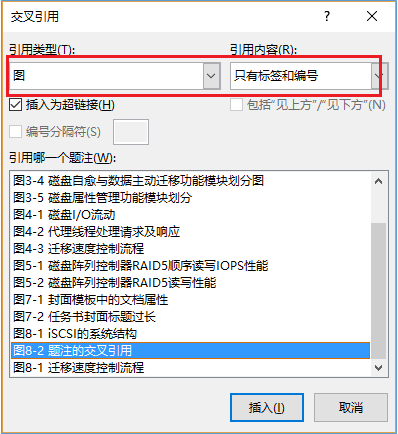


图8‑5 插入交叉引用对话框

手工输入图引用的方式非常不方便，如果图编号发生更改要全文修改编号，Office中提供交叉引用的功能进行题注的自动引用，方便编号自动更新，如图8‑4所示，点击交叉引用后得到图8‑5所示对话框，选择合适的图编号即可，注意引用内容选择只有标签和编号。

### 错误的插图格式

图8‑6所示图的排版中插入时未取消左侧的格式缩进，右侧超出了页边距，专业术语称之为“出血”现象，请一定要仔细核对插图以及图注的缩进问题，插图和图注应该严格居中，插图不得超出页边距。另外文章红色字体过大，一般也是有问题，建议所有图标中的文字字体均小于正文字体。

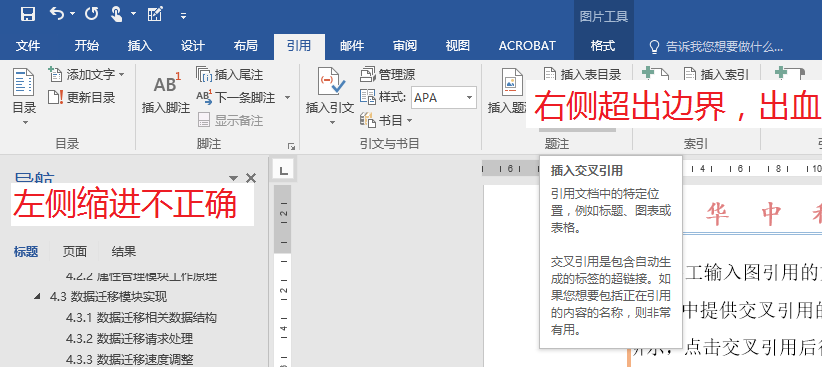


图8‑6 缩进错误，出血错误实例

不得出现下面这样大段的空行或留白现象，每章最后一节除外。

## 表的格式要求

表的格式要求基本和图一致，注意图表中的文字均需要小于正文字体大小，与图注位置相反，表的题注必须在表的正上方。

表8‑1 表头必须在表的正上方

|  | Type 0 | Type 1 | Type 5 | Type 7 |
| --- | --- | --- | --- | --- |
| RAID级别 | 0 | 10 | 5 | N/A |
| 磁盘个数 | 8 | 8 | 8 | N/A |
| 数据块大小 | 512KB | 16KB | 64KB | 1KB |
| 数据块个数 | 8K | 8K | 8K | 8K |

所有表均应有表号和表名（office中称题注，表注），表注出现在表的正上方，小四黑体居中，在本模板中可以直接使用样式库中的题注样式即可规范格式，表号和表名后面不加标点符号，也不得加参考文献引用。

表格由表号、表名、表头、表身等组成。表号按章编，如第2章的表为表2-1、表2-2、…，第3章的表为表3-1、表3-2、…等。表名是表格的名称，扼要概括表的内容，字数不宜太多。

表头包括栏头、行头，与表身一起构成表格的主体。表中的竖称为栏，横格称为行。表身的内容，一般包括：数据、文字、公式和表图等。表内的数据对应位要对齐。少数表有表注，表注写在表下面且字号应比表号、表名的字小一号。

所有表格均应在正文中予以引用，具体引用方式和图的引用方式类似，如图8‑7所示。引用某表格时，一般写为“…见表1-2”或“表3-1是…”。表格应尽量靠近正文的叙述，一般应先见文，后见表，表不跨节。表格允许转页。表格转页部分可以不写表号和表名，但要重复书写表头，具体设置如图8‑8所示。

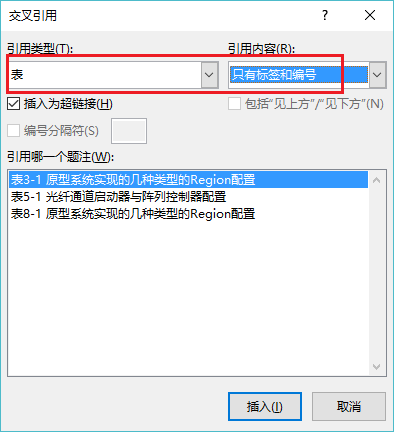


图8‑7 插入表注对话框

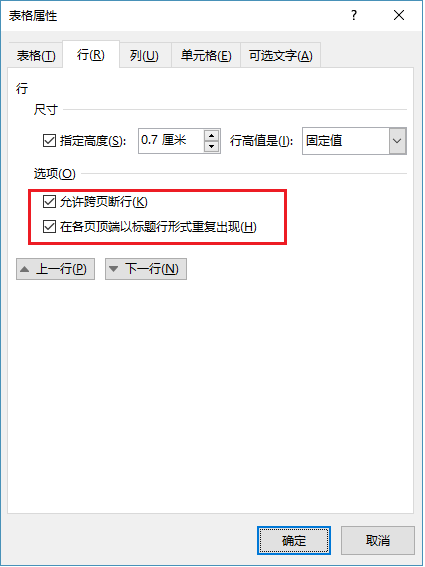


图8‑8 表格跨行设置

## 公式

常见公式排版格式如下：

(8‑1)

公式必须进行标注，公式中的变量必须用斜体，标注要居右，公式居中，要做到这一点可以在标号前面加空格进行调整。

公式一般另行居中写，公式末不加标点。若公式前有文字，如例、解、证、假定等，文字顶格写，公式仍居中写。一行写不下时，公式允许转行。公式转行需处理得当，做到既意义正确，又使版面美观匀称。

公式要有编号，公式编号按章编，如第2章的公式为(2-1)、(2-2)、…，第3章的公式为(3-1)、(3-2)、…等。公式编号写在公式右侧行末顶边线，并加圆括号。模板中公式编号已经按照图题注的方式给出，可以直接复制然后更新域自动排序。。

公式一般应在正文中予以引用，引用时以公式编号指示公式。正文中常有公式中表示量的符号说明，采用“式中”二字作为标志。一般可写成接排形式，如“式中，A指……；B指……”。

## 流程图



图8‑9 迁移速度控制流程

（标准的流程图一定要有开始和结束当然有些程序没有结束，请注意流程图的画法，Yes/No分支应该清晰的标注在菱形分支附近，而不是很远。

## 常见格式问题

|  |
| --- |
| 未定义标签 |
| 各章之间未分页 |
| 标题格式不正确 |
| 章节标题不合适 |
| 缩进不规范 |
| 行距不统一 |
|  |
|  |
| 不必要的留白和空行 |
| 图表中字体明显大于正文 |
| 图表比例失调，甚至有出血现象 |
| 表注不规范 |
| 表格排版欠美观 |

参考文献

1. Palkar S, Lan C, Han S, et al. E2: a framework for NFV applications[C]. Symposium on Operating Systems Principles. ACM, 2015:121-136.
2. Bremler-Barr A, Harchol Y, Hay D. OpenBox: A Software-Defined Framework for Developing, Deploying, and Managing Network Functions[C]. Conference on ACM SIGCOMM 2016 Conference. ACM, 2016:511-524.
3. Sekar V, Egi N, Ratnasamy S, et al. Design and implementation of a consolidated middlebox architecture[C]. Usenix Conference on Networked Systems Design and Implementation. 2011:24-24.
4. Anderson J W, Braud R, Kapoor R, et al. xOMB: extensible open middleboxes with commodity servers[C]. Eighth ACM/IEEE Symposium on Architectures for NETWORKING and Communications Systems. ACM, 2012:49-60.
5. A. Gember, R. Grandl, A. Anand, T. Benson, and A. Akella. Stratos: Virtual Middleboxes as First-class Entities[R]. Technical report, UW-Madison 2012.
6. Zhang W, Liu G, Zhang W, et al. OpenNetVM: A Platform for High Performance Network Service Chains[C]. The Workshop on Hot Topics in Middleboxes and Network Function Virtualization. ACM, 2016:26-31.
7. Gember-Jacobson A, Viswanathan R, Prakash C, et al. OpenNF: enabling innovation in network function control[C]. ACM Conference on SIGCOMM. ACM, 2015:163-174.
8. Rajagopalan S, Dan W, Jamjoom H, et al. Split/merge: system support for elastic execution in virtual middleboxes[C]. Usenix Conference on Networked Systems Design and Implementation. 2013:227-240.
9. J. Khalid, A. Gember-Jacobson, R. Michael, A. Abhashkumar, and A. Akella. Paving the Way for NFV: Simplifying Middlebox Modifications Using StateAlyzr[C]. In Proc. of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI’16), 2016.
10. Ballani H, Costa P, Gkantsidis C, et al. Enabling End-Host Network Functions[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4):493-507.
11. Paxson V. Bro: a system for detecting network intruders in real-time[C]. Conference on Usenix Security Symposium. USENIX Association, 1998:3-3.
12. Yoan N. Iptables[M]. Miss Press, 2013.
13. Sherry J, Gao P X, Basu S, et al. Rollback-Recovery for Middleboxes[C]. ACM Conference on Special Interest Group on Data Communication. ACM, 2015:227-240.
14. Chrome G. HTTP persistent connection[J]. 2015.
15. Surhone L M, Tennoe M T, Henssonow S F, et al. Ffmpeg[M]. 2010.
16. Housley R, Hoffman P. Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP[J]. Ietf Rfc Sri Network Information, 1999, 11(3):82--89.
17. Maheshwari A, Sharma A, Ramamritham K, et al. TranSquid :Transcoding and Caching Proxy for Heterogenous E-Commerce Environments[C]. International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/e-Business Systems. IEEE Computer Society, 2002:50.
18. J. Khalid, A. Gember-Jacobson, R. Michael, A. Abhashkumar, and A. Akella. Paving the Way for NFV: Simplifying Middlebox Modifications Using StateAlyzr[C]. In Proc. of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI’16), 2016.
19. NFV Paper [EB/OL].https://portal.etsi.org/NFV/NFV White Paper2.pdf,2017-05-05.
20. Hwang J, Ramakrishnan K K, Wood T. NetVM: high performance and flexible networking using virtualization on commodity platforms[C]. Usenix Conference on Networked Systems Design and Implementation. USENIX Association, 2014:445-458.
21. S. Han, K. Jang, A. Panda, S. Palkar, D. Han, and S. Ratnasamy. SoftNIC: A Software NIC to Augment Hardware[R]. Technical report, EECS Department, University of California, Berkeley, 2015..
22. Martins J, Ahmed M, Raiciu C, et al. ClickOS and the art of network function virtualization[C]. Usenix Conference on Networked Systems Design and Implementation. USENIX Association, 2014:459-473.
23. Panda A, Han S, Jang K, et al. NetBricks: taking the V out of NFV[C]. Usenix Conference on Operating Systems Design and Implementation. USENIX Association, 2016:203-216.
24. A. Gember, R. Grandl, A. Anand, T. Benson, and A. Akella. Stratos: Virtual Middleboxes as First-class Entities[R]. Technical report, UW-Madison 2012.
25. Palkar S, Lan C, Han S, et al. E2: a framework for NFV applications[C]. Symposium on Operating Systems Principles. ACM, 2015:121-136.
26. Rajagopalan S, Dan W, Jamjoom H. Pico replication:a high availability framework for middleboxes[C]. Symposium on Cloud Computing. 2013:1-15.
27. Sherry J, Gao P X, Basu S, et al. Rollback-Recovery for Middleboxes[C]. ACM Conference on Special Interest Group on Data Communication. ACM, 2015:227-240.
28. Qazi Z A, Tu C C, Chiang L, et al. SIMPLE-fying middlebox policy enforcement using SDN[J]. Computer Communication Review, 2013, 43(4):27-38.
29. Zhang W, Hwang J, Rajagopalan S, et al. Flurries: Countless Fine-Grained NFs for Flexible Per-Flow Customization[C]. International on Conference on Emerging NETWORKING Experiments and Technologies. ACM, 2016:3-17.
30. Bremler-Barr A, Harchol Y, Hay D. OpenBox: A Software-Defined Framework for Developing, Deploying, and Managing Network Functions[C]. Conference on ACM SIGCOMM 2016 Conference. ACM, 2016:511-524

附录：大学期间发表或提交的论文

1. A. Gember, R. Grandl, A. Anand, T. Benson, and A. Akella. Stratos: Virtual Middleboxes as First-class Entities[R]. Technical report, UW-Madison 2012.
2. Palkar S, Lan C, Han S, et al. E2: a framework for NFV applications[C]. Symposium on Operating Systems Principles. ACM, 2015:121-136.

此页以后的内容为装订材料！先仔细阅读！

撰写论文时此页以后的内容全部删掉！

装订毕业论文时应将任务书原件装订到最后



**本科生毕业设计任务书**

|  |  |
| --- | --- |
| 题 目 | 基于生成对抗网络的闪存仿真器设计与实现 |
|  |  |

（任务起止日期：2020年1月1日～2020年6月10日）

|  |  |
| --- | --- |
| 院 系 | 计算机科学与技术 |
| 专业班级 | 计算机1604 |
| 姓 名 | 孟嵩淼 |
| 学 号 | U201614613 |
| 指导教师 | 吴非 |

教研室（系、所）负责人 2020年1月6日 审查

院（系）负责人 2020年1月9日 批准

|  |
| --- |
| 课题内容 |
| 实现一个简单的分布式网络系统，实现文件的分布式网络化存储。，实现文件的分布式网络化存储。，实现文件的分布式网络化存储。，实现文件的分布式网络化存储。，实现文件的分布式网络化存储。，实现文件的分布式网络化存储。，实现文件的分布式网络化存储。，实现文件的分布式网络化存储。 |
| 课题任务要求 |
| 研究学习文件系统的相关原理及实现研究分布式系统概念及应用研究网络文件系统的原理、架构及实现。研究学习文件系统的相关原理及实现研究分布式系统概念及应用研究网络文件系统的原理、架构及实现。研究学习文件系统的相关原理及实现研究分布式系统概念及应用研究网络文件系统的原理、架构及实现。 |
| 主要参考文献 |
| 1. 董晓明，谢长生．基于对象的进化存储系统研究．计算机科学，2005，32(11): 223~226 2. 庞丽萍编．操作系统原理（第二版）．武汉：华中理工大学出版社，1994.9．225~270 3. （美）Nils J. Nilsson著；郑扣根等译．人工智能（Artificial Intelligence）．北京：机械工业出版社，2000.9．177~194 4. （美）Tom Mitchell著；曾华军等译．机器学习（Machine Learning）．北京：机械工业出版社，2003.1．38~56 5. 谢长生，董晓明，万继光，谭志虎，刘瑞芳．磁盘阵列控制器的设计与原型实现．小型微型计算机系统，2006, 27(1): 173~176 6. （美）Nils J. Nilsson著；郑扣根等译．人工智能（Artificial Intelligence）．北京：机械工业出版社，2000.9．177~194 |
| 同组设计者 |
| 无 |

**成 绩 评 定**

**指导教师评定意见**

装订论文时应将成绩评定页放最后，导师评语放前面！

一、对毕业设计（论文）的学术评语（应具体、确切、实事求是）

|  |
| --- |
|  |

二、对毕业设计评分

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 评分项目  (分值) | 调研论证  (10分) | 外文翻译  (5分) | 设计(论文)撰写质量  (10分) | 学习态度  (10分) | 基本理论和基本技能  (50分) | 创 新  (15分) | 合 计  (100分) |
| 得分 | **8** | **4** | **8** | **8** | **40** | **12** | **80** |

指导教师签字：**吴非** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 年 月 日

**答辩小组评定意见**

答辩小组意见和综合评定页放最后，双面！

一、评语（根据学生答辩情况及其设计（论文）质量综合评价）

|  |
| --- |
|  |

二、评分

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 评分项目  (分值) | 答 辩 情 况 | | 论 文 质 量 | | 合 计  (100分) |
| 答辩情况  (15分) | 回答问题情况  (25分) | 规范要求与文字表达  (20分) | 学术水平  (40分) |
| 得分 |  |  |  |  |  |

答辩小组长签字：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 年 月 日

**毕业答辩及成绩评定说明**

1. 毕业答辩
2. 答辩前，答辩小组应详细审阅每个学生的毕业设计（论文），为答辩做好准备。
3. 严肃认真组织答辩，开好答辩会。
4. 指导教师应参加所指导学生的答辩会，但评定其成绩时宜回避。
5. 答辩中要做好记录以供成绩评定时参考。
6. 成绩评定
7. 答辩前每个学生都要将自己的毕业设计（论文）在指定时间内交给指导教师，由指导教师审阅，写出评语并预评分。
8. 答辩工作结束后，答辩小组应举行专门会议进行讨论，在参考指导教师预评结果的基础上，结合学生毕业设计（论文）质量和学生答辩情况，综合评定每个学生的成绩。
9. 院（系）对专业答辩小组提出的优秀和不及格的毕业设计（论文），要组织院（系）级答辩，最终确定成绩，并向学生公布。
10. 各专业学生的最后成绩应符合正态分布规律。
11. 请用蓝、黑钢笔手写或五号宋体字编辑，签名须手写，A4纸双面打印。

**毕业设计（论文）成绩评定**

|  |
| --- |
| 班号：**计算机1604** 学生姓名：**孟嵩淼**  综合成绩：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_分（折合等级\_\_\_\_\_\_\_\_\_\_\_\_\_\_）  答辩小组长（签名）：\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 年 月 日 |