

班 级 1503013
学 号 15030130026

西安电子科技大学

本科毕业设计论文



题 目 面向负载均衡的虚拟机

迁移方法研究

学 院 计算机科学与技术学院

专 业 计算机科学与技术

学生姓名 王作栋

导师姓名 赵辉

毕业设计（论文）诚信声明书

本人声明：本人所提交的毕业论文《面向负载均衡的虚拟机迁移方法研究》是本人在指导教师指导下独立研究、写作成果，论文中所引用他人的无论以何种方式发布的文字、研究成果，均在论文中加以说明；有关教师、同学和其他人员对文本的写作、修订提出过并为我在论文中加以采纳的意见、建议，均已在我的致谢辞中加以说明并深致谢意。

本文和资料若有不实之处，本人承担一切相关责任。

论文作者：_____ (签字) 时间： 年 月 日
指导教师已阅：_____ (签字) 时间： 年 月 日

摘 要

随着云计算技术的发展，虚拟化技术已经成为提高云平台资源利用率的有效手段。随着数据中心的服务器不断运行，服务器中虚拟机会不断的开启、关闭，导致某些服务器负载过重，影响服务质量；某些服务器却空闲，浪费服务器资源。通过虚拟机的动态迁移技术，将高负载服务器上运行的虚拟机迁移至空闲或低负载服务器上，从而实现服务器的负载均衡。本文根据蚁群算法的思想设计了一种面向负载均衡的虚拟机动态迁移框架。首先进行框架的总体设计，然后对框架的各个模块进行详细地设计，其中蚂蚁算法模块是框架的核心，其核心思想是利用蚂蚁在搜索过程中获取并记录服务器信息，并根据信息素等信息不断优化搜索路径，找到适合迁移的虚拟机。并通过设置不同的阈值来定义一个服务器的负载状态模型，通过定时计算服务器的负载值来动态观察服务器所处的负载域，当服务器过载便会生成蚂蚁去搜寻可以迁移的服务器。最终通过一定的选择策略来选择待迁移服务器上适合迁移的虚拟机，以及一定的迁移策略来选择适合迁移的目的服务器，最终完成虚拟机的迁移过程。一个数据中心内的所有服务器，不断重复这一迁移过程来达到整个数据中心内的服务器负载均衡。最后通过扩展云计算基础架构和服务的建模和仿真框架 CloudSim，仿真云计算环境下的数据中心、服务器以及虚拟机等云计算基础设施，并应用本文提出的虚拟机迁移框架进行虚拟机迁移的模拟，最终对这个虚拟机迁移框架的性能及其最终的负载均衡效果进行分析与评估。

关键词：云计算 虚拟化 虚拟机迁移 负载均衡

摘要

Abstract

With the development of cloud computing technology, virtualization technology has become an effective means to improve the utilization of cloud platform resources. As the servers in the data center continue to run, the virtual machines in the server are continuously turned on and off, causing some servers to be overloaded and affecting the quality of service; some servers are idle and waste server resources. The virtual machine's dynamic migration technology is used to migrate the virtual machines running on the high-load servers to idle or low-load servers to achieve load balancing of the servers. According to the idea of ant colony algorithm, this paper designs a virtual machine dynamic migration framework for load balancing. Firstly, the overall design of the framework is carried out, and then the various modules of the framework are designed in detail. The ant algorithm module is the core of the framework. The core idea is to use ants to obtain and record server information during the search process, and continuously based on information such as pheromones. Optimize your search path to find a virtual machine that is suitable for migration. And by setting different thresholds to define a load state model of a server, by dynamically calculating the load value of the server to dynamically observe the load domain where the server is located, when the server is overloaded, an ant is generated to search for a server that can be migrated. Finally, a certain selection strategy is adopted to select a virtual machine suitable for migration on the server to be migrated, and a certain migration strategy to select a destination server suitable for migration, and finally complete the migration process of the virtual machine. All servers in a data center continue to replicate this migration process to achieve server load balancing across the data center. Finally, CloudSim, which is a modeling and simulation framework for cloud computing infrastructure and services, is used to simulate cloud computing infrastructure such as data centers, servers, and virtual machines in cloud computing environments, and the virtual machine migration framework proposed in this paper is used for virtual machine migration. Simulation, and finally analyze and evaluate the performance of this virtual machine

Abstract

migration framework and its final load balancing effect.

Key words: cloud computing virtualization virtual machine migration load balancing

目 录

第一章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	1
1.3 本文的主要工作	2
1.4 本文的内容安排	2
第二章 绪论	5
2.1 文字处理	5
2.1.1 字体设置	5
2.1.2 数字转中文	6
2.2 表格	6
2.2.1 普通表格	6
2.2.2 复杂点的表格	6
2.2.3 长表格	7
2.3 图片	8
2.3.1 普通图片的插入	8
2.3.2 图片并排插入	9
2.4 公式	9
2.4.1 普通公式	9
2.4.2 复杂公式	10
2.5 休息一下	11
2.5.1 山水之间	11
2.5.2 念奴娇·赤壁怀古	12
第三章 两个重点	13
3.1 环境	13
3.1.1 定理类环境	13

3.1.2 代码环境	14
3.2 参考文献的引用	14
致谢	17
参考文献	19
附录 A 数据	21
A.1 放松一下	21
A.1.1 惊鸿一面	21
A.1.2 无题	22
A.2 代码	22

第一章 绪论

1.1 研究背景及意义

云是由许多的虚拟化的服务器互联组成的大型分布式系统，它管理着巨大的计算资源、网络资源和存储资源。它们是动态配置的，通过服务提供者和消费者之间地协商建立服务协议，服务消费者可以支付相应的服务费用来动态地获得服务资源。虚拟化是将计算机的资源划分为多个执行的框架或方法环境，通过对底层硬件资源进行抽象，可以将资源抽象为多个互相独立、互不影响的计算环境。虚拟化的方法，它有助于单独物理机器（PM）的资源在各种虚拟机（VM）之间确保执行分离，提高资源利用效率。迁移是现代虚拟机技术提供的最重要的特性之一。它可以将虚拟机实例从一个物理节点移动到另一个物理节点，而不会中断正在迁移的环境。它是一个非常强大的集群管理工具，可以有效提高集群和数据中心在线系统维护、负载均衡的效果。

1.2 国内外研究现状

应用程序性能需求的前期可以通过高性能硬件和相关软件协议的适当配置来满足。高性能的硬件资源造成了成本的限制，通过对云数据中心的适当部署和管理，也可以降低成本。利用云服务模型按使用付费策略的好处，可以控制成本约束。此后，大多数服务提供商决定将其应用程序移入云端，从而将数据中心移入云端。结合硬件和负载平衡算法，应用程序的高响应特性还需要数据中心内部架构的支持，以保持高级别的可用性。并行研究结果表明，仅通过增强负载平衡机制，就可以进一步提高性能。应用程序的高可用性表明数据中心组件的最佳冗余控制。因此，平行研究结果也证明了在许多方法中使用最佳设置。云上的数据中心是所有主要计算和通信硬件组件的集合框架，因此，深入分析和理解所有硬件组件是研究的主要目标，以实现最佳策略。为了实现最高级别的负载平衡，云数据中心的性能需要通过固定格式的性能参数进行验证，而不是根据可用性或高计算速度

或更高存储级别或更高冗余级别来衡量的传统方法。最近的研究成果未能提出一个标准的性能评估矩阵，该矩阵必须提出一组关注数据中心各个方面的参数，以实现高水平的负载平衡性能。就云端数据中心的性能而言，对计算增长的需求以及性能参数的制定是当前研究趋势的需求。

1.3 本文的主要工作

本文对面向负载均衡的虚拟机迁移方法进行了深入研究，设计了一种虚拟机动态迁移框架。主要工作如下：

1. 理解云计算及虚拟化的相关概念，研究虚拟机动态迁移的技术。
2. 设计虚拟机动态迁移框架。借鉴蚁群算法的思想，利用蚂蚁在搜索过程中获取并记录服务器信息，并根据信息素等信息不断优化搜索路径，找到适合迁移的虚拟机。并通过设置不同的阈值来定义一个服务器的负载状态模型，通过定时计算服务器的负载值来动态观察服务器所处的负载域，当服务器过载便会生成蚂蚁去搜寻可以迁移的服务器。最终通过一定的选择策略来选择待迁移服务器上适合迁移的虚拟机，以及一定的迁移策略来选择适合迁移的目的服务器，最终完成虚拟机的迁移过程。
3. 扩展了云计算模拟平台 CloudSim，仿真云计算环境下的数据中心、服务器以及虚拟机等云计算基础设施，并应用本文提出的虚拟机迁移框架进行虚拟机迁移的模拟，最终对这个虚拟机迁移框架的性能及其最终的负载均衡效果进行分析与评估。

1.4 本文的内容安排

本文主要分为五章，内容如下：

第一章，绪论

介绍本课题的研究背景及意义，对虚拟机迁移技术的相关国内外研究现状进行了分析，并对本文的主要工作和论文内容安排进行了说明。

第二章，相关技术研究

对本文应用的相关技术进行了研究，包括虚拟化技术、虚拟机动态迁移技术、蚁群算法等。

第三章，面向负载均衡的虚拟机动态迁移框架设计

提出了面向负载均衡的虚拟机动态迁移框架，并对相关的算法设计，服务器负载域模型以及虚拟机迁移的相关策略进行了详细论述。

第四章，仿真实验与结果分析

扩展了支持云计算系统和应用程序建模和仿真的仿真工具 CloudSim，应用本文提出的虚拟机迁移框架进行虚拟机迁移的模拟，最终对这个虚拟机迁移框架的性能及其最终的负载均衡效果进行分析与评估。

第五章，总结与展望

总结了本文的主要工作和成果，思考可以进一步进行设计优化的方向。

第二章 绪论

不用看了，本文内容其实和章名没多大关系，只是没得名起了，迎合《本科生毕业设计（论文）工作手册》要求而已。

2.1 文字处理

2.1.1 字体设置

西安电子科技大学（Xidian University）简称“西电”或“西军电”，坐落于古都西安。学校是中央部属高校，教育部直属、工信部共建，国家首批“211 工程”，是“985 工程优势学科创新平台”、“111 计划”、“2011 计划”重点建设高校（！！！子！！！！！！！！一的！2011 计！！！！！！），35 所示范性软件学院的高校之一、集成电路人才培养基地的高校之一，56 所获批设立研究生院的重点大学之一，也是**北京高科大学联盟**的重要成员。

1931 年 1 月 28 日，红一方面军总司令朱德、总政委毛泽东于小布总部签发“调学生学无线电的命令”，随后，第一期无线电训练班在小布镇陈家土楼正式开课。后迁移至瑞金，成立中央军委无线电学校，是毛泽东等老一辈革命家亲手创建的第一所工程技术学校。1958 年学校迁址西安，1966 年转为地方建制，1988 年定为西安电子科技大学。该校是中国最早建立信息论、信息系统工程、雷达、微波天线、电子机械、电子对抗等专业的高校之一，开辟了中国 IT 学科的先河，形成了鲜明的电子与信息学科特色与优势。毛泽东曾先后两次为该校题词：“**全心全意为人民服务**”、“**艰苦朴素**”。^①

来点没用的

^①从百度百科粘下来的，就不放进参考文献了。

2.1.2 数字转中文

测试数字：1234.233. 转为中文数字：一千二百三十四点二三三；转为中文字字符串：一二三四点二三三。

2.2 表格

2.2.1 普通表格

先来看一个无标题的普通表格：

标题	标题	标题
1	2	3

如果想要居中可以使用 `center` 环境。

带标题表格。

表 2.1 普通表格 1

标题	标题	标题
1	2	3

来一个表格并排，每个表格一个标题：

表 2.2 并排表格 1

标题	标题	标题
1	2	3

表 2.3 并排表格 2

标题	标题	标题
1	2	3

再来一个表格并排

表 2.4 并排表格

(a) 并排表格 1

标题	标题	标题
1	2	3

(b) 并排表格 2

标题	标题	标题
1	2	3

2.2.2 复杂点的表格

表 2.5 主要用到的是列合并单元格、跨行合并单元格、混合合并单元格、表格横线的自定义粗细以及控制表格横线的自定义连接等。^①

^①表格虽然难看，主要是为了让大家看效果，里面技巧选用。

表 2.5 复杂表格

1	3	4			
5	6	7	8		
9	10	11			
13	15		16	19	20

表格填充颜色，如表 2.6 所示。此次，主要用到的出上述介绍外，有单个单元格填充颜色、整列填充颜色、整行填充颜色；此外，还添加了单元格划分的功能。

表 2.6 填色表格

No.	Title	L-Title	R-Title
1	One	First	
2	Two	Second	
3	Three	Third	

以上，就是普通表格的常用例子了，其他的应用技巧以及功能大家自学吧，毕竟这只是个模板的使用样例，不是 `LATEX` 教案。

2.2.3 长表格

呐，在开始长表格之前，我们应该怎么样呢？对，先说点废话。为什么呢？你想啊，既然是长表格，肯定是能够跨页存在的，不说点废话把它顶下去，顶到换页，咋能对得起它的 NB 功能呢，是吧。

咳咳，严肃点，主角登场了。表 2.7 就是这一小节的主角——长表格了。首先说明一下，这种表格如果放的数据太多的话，就不要在正文里面用了，放到附录里就可以了。

表 2.7 这是一个长表格

行号	标题 1	标题 2	标题 3	标题 4
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16

接下页续表……

续表 2.7 这是一个长表格

行号	标题 1	标题 2	标题 3	标题 4
5	17	18	19	20
6	21	22	23	24
7	25	26	27	28
8	29	30	31	32
9	33	34	35	36
10	37	38	39	40
11	41	42	43	44
12	45	46	47	48
13	49	50	51	52
14	52	54	55	56
15	57	58	59	60
16	61	62	63	64
17	65	66	67	68
18	69	70	71	72
19	73	74	75	76
20	77	78	79	80
21	81	82	83	84
22	85	86	87	88
23	89	90	91	92
24	93	94	95	96
25	97	98	99	100

2.3 图片

2.3.1 普通图片的插入

如同表格一样，插图一般都用浮动体来控制，这样排出来的文章美观。^①比如，图 2.1 所示，是一个图片的样例。

^①本文涉及到的所有图片，除我校校徽等标记外，均为个人拍摄。

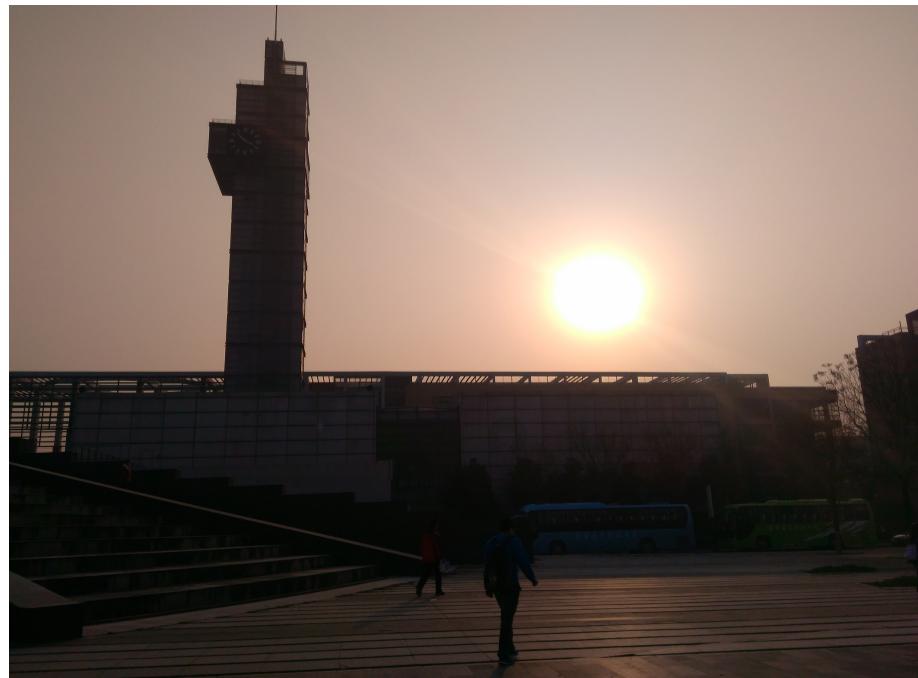


图 2.1 西电 1

2.3.2 图片并排插入

先来看第一个，两个图片分别一个标题，如图 2.2 和图 2.3.

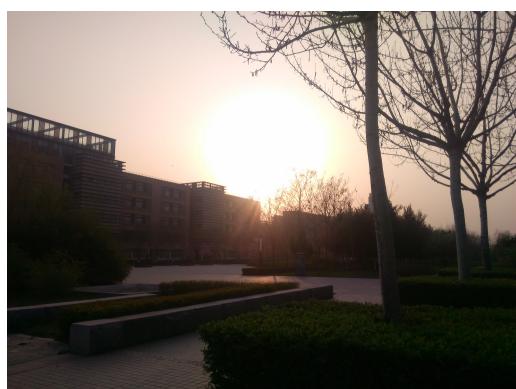


图 2.2 西电 2

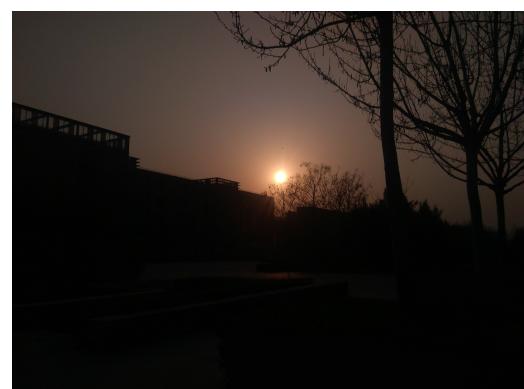


图 2.3 西电 3

再来看一个统一大标题带子标题的，如图 2.4 所示。

插图就介绍到这，其他知识自学。

2.4 公式

2.4.1 普通公式

先来看一个行内公式： $y = x + 1$.



(a) 西电 4

(b) 西电 5

图 2.4 并排插图

下面是一个居中的公式：

$$f(X) = \sum_{i=1}^n \sin \frac{\pi}{2} x_i$$

看一个编号的公式，如式(2-1)：

$$f(X) = \sum_{i=1}^n \sin \frac{\pi}{2} x_i \quad (2-1)$$

2.4.2 复杂公式

- 多行公式

$$\begin{aligned} f(x) &= \sin(a + b) \\ &= \sin(a) \cos(b) + \cos(a) \sin(b). \end{aligned} \quad (2-2)$$

$$f(x) = \begin{cases} -x + 1, & \text{if } x < 0; \\ x + 1, & \text{if } x \geq 0; \end{cases} \quad (2-3)$$

- 矩阵

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}, \quad C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ \vdots & \ddots & \vdots \\ c_{31} & c_{32} & c_{33} \end{bmatrix}.$$

行列式也类似

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}.$$

- 其他公式

$$\mathbb{L} = \int_{x_1}^{x_2} \sqrt{1 + (y')^2} dx = \frac{1}{\alpha} \int_{x_1}^{x_2} y''(x) dx$$

2.5 休息一下

2.5.1 山水之间

山水之间

许嵩

昨夜同门云集 推杯又换盏
 今朝茶凉酒寒 豪言成笑谈
 半生累 尽徒然
 碑文完美有谁看
 隐居山水之间 誓与浮名散
 湖畔青石板上 一把油纸伞
 旅人停步折花 淋湿了绸缎
 满树玉瓣多傲然
 江南烟雨却痴缠
 花飞雨追 一如尘缘理还乱
 落花雨 你飘摇的美丽
 花香氤 把往日情勾起
 我愿意 化浮萍躺湖心
 只陪你 泛岁月的涟漪

2.5.2 念奴娇·赤壁怀古

念奴娇·赤壁怀古

苏轼

大江东去，浪淘尽，千古风流人物。

故垒西边，人道是：三国周郎赤壁。

乱石穿空，惊涛拍岸，卷起千堆雪。

江山如画，一时多少豪杰。

遥想公瑾当年，小乔初嫁了，雄姿英发。

羽扇纶巾，谈笑间樯橹灰飞烟灭。

故国神游，多情应笑我，早生华发。

人生如梦，一尊还酹江月。

这一章就写到这里吧，下一章重点介绍两个内容

第三章 两个重点

这一章，会涉及到环境以及参考文献两个重点东西。

3.1 环境

3.1.1 定理类环境

命题 3.1： 这是一个命题。

它还可以这么写，所有定理类环境，都可这么写。

命题 3.2 (命题名)： 由命题 3.1，这也是一个命题。

假设 3.1： 距离水平面 $100 m$ 以内，重力加速度 g 是不变的。

引理 3.1 (法图引理)： 设 (S, Σ, μ) 为一个测度空间， $(f_n)_{n \geq 0}$ 是一个实值的可正值函数列。那么：

$$\int_S \liminf_{n \rightarrow \infty} f_n d\mu \leq \liminf_{n \rightarrow \infty} \int_S f_n d\mu.$$

其中函数极限是在逐点收敛的意义上的极限，函数取值和积分可以是无限大的。

定理 3.1： 若 $\lim_{n \rightarrow \infty} x_n = a$ ，则 $\{x_n\}$ 的任何子列 $\{x_{n_k}\}$ 都收敛于 a .

证明 因 $\lim_{n \rightarrow \infty} x_n = a$ ，对任意给定的 $\varepsilon > 0$ ，必定存在正整数 N ，当 $n > N$ 时，有

$$|x_n - a| < \varepsilon.$$

今取 $K = N$ ，则对一切 $k > K$ ，有 $n_k > n_K = n_N \geq N$ ，这时就有

$$|x_{n_k} - a| < \varepsilon.$$

□

3.1.2 代码环境

```

1 #include<stdio.h>
2
3 int main()
4 {
5     printf("Hello World!");
6     return 0;
7 }
```

代码 3.1 Java 代码

```

1 package com.stick.test;
2
3 /**
4 * 公主类;
5 * 类名必须与文件名相同
6 */
7 public class Test{
8     public static void main(String[] args) {
9         System.out.println("Hello World!");
10    }
11 }
```

x<-c(1,2,3,4,5,6)

y<-sin(x)

lines (x,y)

3.2 参考文献的引用

以下为常用的参考文献。

期刊^[1,2], 专著 (无页码)^[3-5], 专著 (含页码)^[6-8], 论文集^[9,10]。

硕士^[11]/博士论文^[12], 科技报告^[13]。

以下参考文献, 原有 plain 风格中并无, 为 Stick 本人自己写的。

网络内容引用^[14], 译著^[15]

此外, 多说一句, 本文模板加载的 `natbib` 宏包使得 `\cite` 命令还支持排序等功能。

比如下列文献^[2,4,8,14,15]

致 谢

一转眼，大学四年的生活走到了结尾，心里既有对着四年时光的不舍与留恋，也有对未来生活的憧憬与不安。但更多的是对这四年的经历以及对我支持关怀的老师、同学和父母家人的感谢之情！首先要感谢我的毕设导师赵辉老师，在这段时间给了我们很多的指导与帮助，帮助我们完成了大学的最后一课。还有感谢大学四年所有教过和帮助过我的老师们，是他们让我四年时光有许多成长，感谢他们的辛勤付出与陪伴！感谢四年一起走过的同学们，我们一起欢笑，一起努力，一起成长，一起迈入人生新的阶段，日子还长，未来可期！感谢在背后默默支持我的父母家人，他们默默付出，任劳任怨，无怨无悔，希望用他们的双手为我们撑起一片天，那么我们今后就要独自撑起自己的生活，让他们安心，让他们不再那么辛苦！虽然大学生活告一段落了，但是今后的人生才刚刚开始，学习之路也还远没有结束，希望自己永葆一颗年轻活力、积极进取的心，每当走过一段再回首往事的时候可以无愧于心！

参考文献

- [1] 作者 1, 作者 2, 作者 3, 等. 文献名[J]. 期刊名, 2015, 23(9):1-13.
- [2] Author1, Author2, Author3, et al. Article title[J]. Journal Name, 2015, 23(9):1-13.
- [3] 作者 1, 作者 2, 作者 3, 等. 书名[M]. 二. 出版地: 出版社, 2015.
- [4] Author1, Author2, Author3, et al. Book name[M]. Publisher Address: Publisher Name, 2015.
- [5] 作者 1, 作者 2, 作者 3, 等. 书名[M]. 出版地: 出版社, 2015.
- [6] 作者 1, 作者 2, 作者 3, 等. 书名[M]. 二. 出版地: 出版社, 2015: 2-13.
- [7] 作者 1, 作者 2, 作者 3, 等. 书名[M]. 出版地: 出版社, 2015: 2-13.
- [8] Author1, Author2, Author3, et al. Book name[M]. 2nd ed. Publisher Address: Publisher Name, 2015: 2-13.
- [9] 作者 1, 作者 2, 作者 3, 等. 篇名[M]/编辑/主编. 论文集名. 二. 出版地: 出版社, 2015: 24-57.
- [10] Author1, Author2, Author3, et al. Paper title[M]/Name E. Book Title. 2nd ed. Address: Publisher Name, 2015: 24-57.
- [11] 作者 1, 作者 2, 作者 3, 等. 论文名[D]. 保存者地址: 保存者 (一般是学校), 2015: 2-13.
- [12] 作者 1, 作者 2, 作者 3, 等. 论文名[D]. 保存者地址: 保存者 (一般是学校), 2015: 2-13.
- [13] 作者 1, 作者 2, 作者 3, 等. 报告名[R]. 机构地址: 机构, 2015: 3-13.
- [14] I-am-13. 新浪微博[EB/OL]. 2015. <http://weibo.com/StickCui>.
- [15] (原作者) 作者 1, 作者 2, 作者 3, 等. 译著名[M]. (译者) 作者 1, 作者 2, 作者 3, 等译. 出版地: 出版社, 2015: 3-13.

附录 A 数据

这里是附录的数据部分，其实是瞎写；来看个公式编号对不对，如式(A-1)所示。

$$h_i = \frac{\max_{j=1}^N \left\{ \frac{c_j}{f_j} \right\} - \frac{c_i}{f_i}}{\max_{j=1}^N \left\{ \frac{c_j}{f_j} \right\} - \min_{j=1}^N \left\{ \frac{c_j}{f_j} \right\}} \quad (\text{A-1})$$

A.1 放松一下

A.1.1 惊鸿一面

惊鸿一面^[A1]

许嵩

翻手为云 覆手为雨

金盆洗手止风雨

不恋红尘 却难舍回忆

每一段都有你

年少初遇 常在我心

多年不减你深情

江山如画 又怎能比拟

你送我的风景

柳下闻瑶琴 起舞和一曲

仿佛映当年 翩若惊鸿影

谁三言两语 撩拨了情意

谁一颦一笑 摆曳了风景

纸扇藏伏笔 玄机诗文里

紫烟燃心语 留香候人寻

史书列豪杰 功过有几许

我今生何求 唯你

远山传来清晨悠然的曲笛

晓风掠走光影

残月沉霜鬓里

有了你

恩怨都似飞鸿踏雪泥

A.1.2 无题

无题

李商隐

相见时难别亦难，东风无力百花残。

春蚕到死丝方尽，蜡炬成灰泪始干。

晓镜但愁云鬓改，夜吟应觉月光寒。

蓬山此去无多路，青鸟殷勤为探看。

A.2 代码

代码 A.1 C++ code

```

1  /*
2   Program: Hello world;
3   Author: Stick Cui;
4   Time: 2015/12/04.
5   */
6   #include<stdio.h>
7
8   int main()
9   {
10      printf("Hello world!");

```

```
11     return 0;  
12 }
```

Java code

```
1 /**
2 * Program: Hello world;<br>
3 * Time: 2015/12/04.
4 * @author Stick Cui
5 */
6 public class JavaTest{  
7     /**
8      * The main method.  
9      * @param args The parameter when the method is called.  
10     */  
11    public static void main(String[] args){  
12        System.out.println("Hello world!");  
13    }  
14 }
```

%floyd 算法通用程序，输入 a 为赋权邻接矩阵

%输出为距离矩阵 D, 和最短路径矩阵 path

```
function [D,path]=floyd(a)
```

```
n=size(a,1);
```

```
D=a;
```

```
path=zeros(n,n);
```

```
for i=1:n
```

```
    for j=1:n
```

```
        if D(i,j)~=inf
```

```
            path(i,j)=j;
```

```
    end
```

```

end

end

for k=1:n
    for i=1:n
        for j=1:n
            if D(i,k)+D(k,j)<D(i,j)
                D(i,j)=D(i,k)+D(k,j);
                path(i,j)=path(i,k);
            end
        end
    end
end

% 配合 floyd 算法的后续程序, s 为源点, t 为宿点
% L 为长度, R 为路由

function [L,R]=router(D,path,s,t)
L=zeros(0,0);
R=s;
while 1
    if s==t
        L=fliplr(L);
        L=[0,L];
        L=L(end);
        return
    end
    L=[L,D(s,t)];
    R=[R,path(s,t)];
    s=path(s,t);
    if s==0
        return
    end

```

end

参考文献

[A1] 许嵩. 惊鸿一面 [A]. 海蝶音乐. 不如吃茶去 [C]. 北京: 北京海蝶音乐有限公司, 2014, 8.