5. MPI 编程

```
#include<stdio.h>
#include "mpi.h"

main(int argc,char *argv[]){
    int i,tmp,sum =0,n;
    int group_size; //进程总数
    int my_rank;    //进程id
    MPI_Status status;
    MPI_Init(&argc,&argv);  //初始化MPI
    MPI_Comm_size(MPI_COMM_WORLD,&group_size); //得到总的进程数
    MPI_Comm_rank(MPI_COMM_WORLD,&my_rank);  //得到自己的进程数
    if(my_rank == 0){ //第一个进程发送消息
        MPI_Send(&n,1,MPI_DOUBLE,1,99,MPI_COMM_WORLD); //发送消息
        for(i=my_rank;i<n;i+=group_size) sum += i;
        MPI_Recv(&tmp,1,MPI_DOUBLE,1,88,MPI_COMM_WORLD,&status);
        sum += tmp;
        printf("\n The result = %d",sum);
    }
    else{
        MPI_Recv(&n,1,MPI_DOUBLE,0,99,MPI_COMM_WORLD,&status);
        for(i=my_rank;i<n;i+=group_size) sum += i;
        MPI_Send(&sum,1,MPI_DOUBLE,0,88,MPI_COMM_WORLD);
    }
    MPI_Finalize(); //结束main
}
```

6. 说明 Map\Reduce 技术原理。试解释如何在 Hadoop 环境下，通过 Map\Reduce 技术，实现大文本文件中每个单词出现的次数。

1. MapReduce 模型主要包含 Mapper 类和 Reducer 类两个抽象类。Mapper 将输入键值对(key/value pair)映射到一组中间格式的键值对集合。; Reducer 将与一个 key 关联的一组中间数值集归约（reduce）为一个更小的数值集。

2. 在 hadoop 中，输入的数据是<文本具体地址，文本内容>，之后 map 函数将文本内容进行分词，变成<文本来源，具体词汇>的键值对的集合，同是筛除文本中的不必要成分，比如英语中的 is、a，中文中的“的”，“是”等，之后由 reduce 函数对词频进行统计，用一个<具体词汇，出现总数>的键值对集合记录所有词的词频，每多统计到一个词，就在特定的项上的“出现总数”上加一。最后，根据 reduce 得到的这个键值对集合，输出对应词的词频.

1. MapReduce 模型主要包含 Mapper 类和 Reducer 类两个抽象类。Mapper 类主要负责对数据的分析处理，最终转化为 key-value 数据对；Reducer 类主要获取 key-value 数据对，然后处理统计，得到结果。

2. 在 hadoop 中，<k1,v1>键值对可以是<行编号，文件中的一行的数据>，经过 Map 函数映射之后，形成一批中间结果<单词，出现次数>，最后 Reduce 函数则可以对中间结果进行处理，将相同的单词出现次数进行累加，得到每个单词的总的出现次数。

1. 什么情况会出现死锁和饿死？

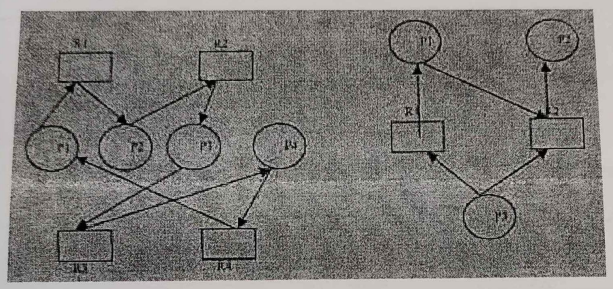
a) 死锁是指两个或两个以上的进程在执行过程中，由于竞争资源或者由于彼此通信而造成的一种阻塞的现象，若无外力作用，他们都将无法进行下去。此时系统处于死锁状态，这些永远在互相等待的进程称为死锁进程。

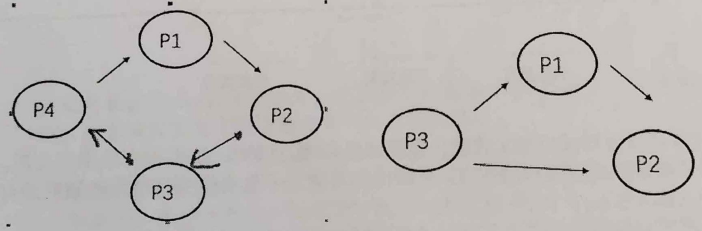
b) 饿死是资源占用者均在有限时间内释放，但由于调度策略的不公平，仍有进程持续申请不到资源，无法进行下去，导致饿死。

死锁是指在多道程序系统中，一组进程中的每一个进程均无限期等待被该组进程中的另一个进程所占有且永远不会释放的资源。

饥饿是指系统不能保证某个进程的等待时间上界，从而使该进程长时间等待，当等待时间给进程推进和响应带来明显影响时，称发生了进程饥饿。当饥饿到一定程度的进程所赋予的任务即使完成也不再具有实际意义时称该进程被饿死。

2. 将下面两个资源分配图转换为等待图。





3. 假设使用 AND 模型，确定以上两个图是否存在死锁？

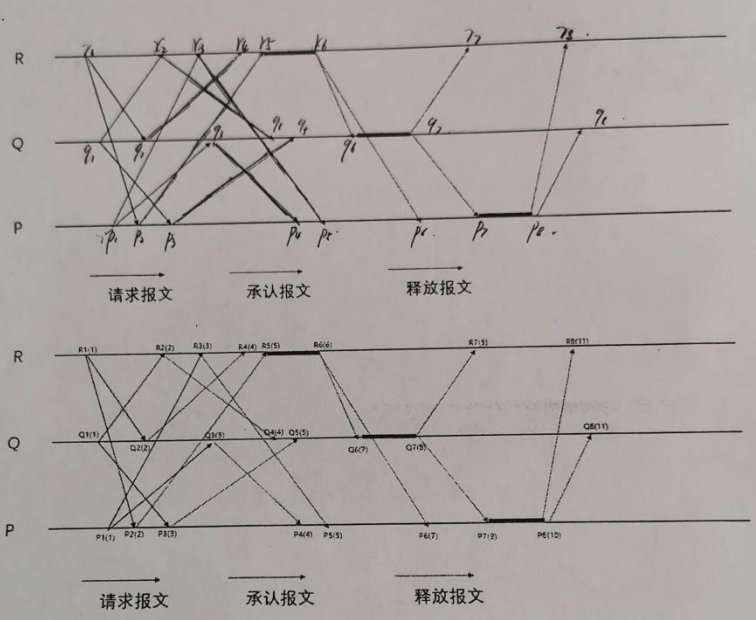
在使用 AND 条件的系统中，死锁条件是在等待图中存在回路。所以图一中存在回路 P1—P2—P3—P4,故图一存在死锁。图二则没有闭合的回路，所以不存在死锁。

4. 一个分布计算系统中存在 3 个进程 P、Q、R，PQ 和 R 分别在 p1,q1 和 r1 时刻（逻辑时钟 p1>q1>r1）发出进去同一临界区的请求，系统采用 Lamport 时

间戳互斥算法进行互斥操作。

1. 画出时空图，标出各进程之间所有报文传递情况，并注明报文类别。找出 PQ 发出请求后分别进入临界区的时间段。
2. 设定 PQR 上各事件的标量逻辑时钟。
3. 你能够改进临界区调度算法以提高性能？

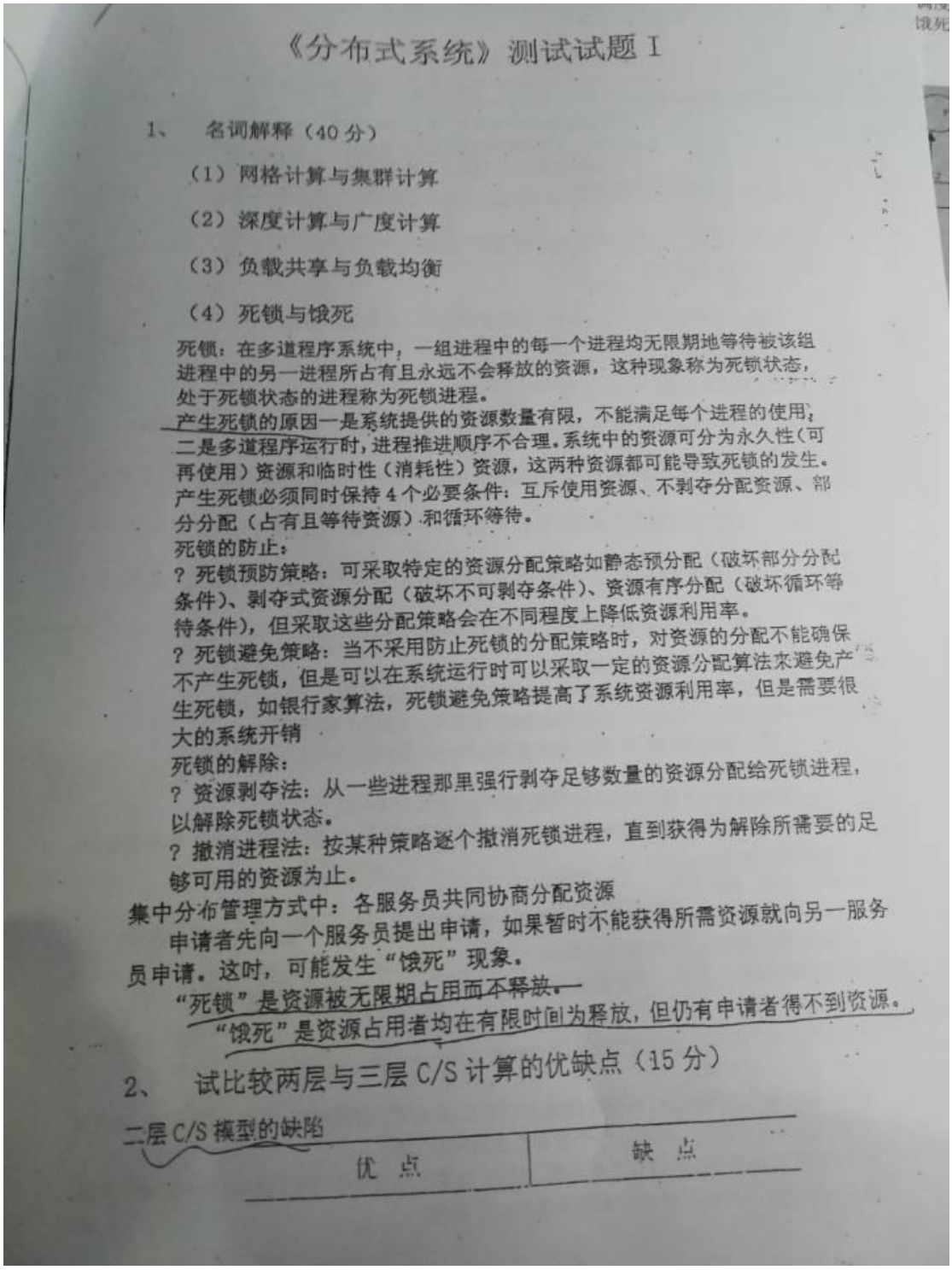
1.



3. 可以通过减少报文数量来提高算法性能。例如：①进程在发出报文后，不需要得到所有进程的回应，只需得到其请求子集中的所有进程的回应即可进入临时区。

②释放临界区时只需向其请求子集中的所有进程发送释放报文。

Part 2



《分布式系统》测试试题 I

1、 名词解释（40 分）

（1）网格计算与集群计算

（2）深度计算与广度计算

（3）负载共享与负载均衡

（4）死锁与饿死

死锁：在多道程序系统中，一组进程中的每一个进程均无限期地等待被该组进程中的另一进程所占有且永远不会释放的资源，这种现象称为死锁状态，处于死锁状态的进程称为死锁进程。

产生死锁的原因一是系统提供的资源数量有限，不能满足每个进程的使用；二是多道程序运行时，进程推进顺序不合理。系统中的资源可分为永久性（可再使用）资源和临时性（消耗性）资源，这两种资源都可能导致死锁的发生。产生死锁必须同时保持 4 个必要条件：互斥使用资源、不剥夺分配资源、部分分配（占有且等待资源）和循环等待。

死锁的防止：

？死锁预防策略：可采取特定的资源分配策略如静态预分配（破坏部分分配条件）、剥夺式资源分配（破坏不可剥夺条件）、资源有序分配（破坏循环等待条件），但采取这些分配策略会在不同程度上降低资源利用率。

？死锁避免策略：当不采用防止死锁的分配策略时，对资源的分配不能确保不产生死锁，但是可以在系统运行时可以采取一定的资源分配算法来避免产生死锁，如银行家算法，死锁避免策略提高了系统资源利用率，但是需要很大的系统开销

死锁的解除：

？资源剥夺法：从一些进程那里强行剥夺足够数量的资源分配给死锁进程，以解除死锁状态。

？撤消进程法：按某种策略逐个撤消死锁进程，直到获得为解除所需要的足够可用的资源为止。

集中分布管理方式中：各服务员共同协商分配资源

申请者先向一个服务员提出申请，如果暂时不能获得所需资源就向另一服务员申请。这时，可能发生“饿死”现象。

“死锁”是资源被无限期占用而不释放。

“饿死”是资源占用者均在有限时间为释放，但仍有申请者得不到资源。

2、 试比较两层与三层 C/S 计算的优缺点（15 分）

二层 C/S 模型的缺陷

优 点	缺 点

《分布式系统》测试试题 II

1、名词解释（40 分）

（1）网格计算与集群计算 P7

（2）网络的 4 种体系结构 P6

（3）负载共享与负载均衡 P7

2、试分析为什么三层客户/服务计算模式可以减少数据库系统 License 个数（10 分） P9

3、在分布式数据库系统中什么情况下会产生饿死和死锁现象？（15 分） P10. P7.9

4、试分析任务划分粒度的大小对任务调度的影响（15 分） P9

5、谈谈你对网络计算发展趋势的看法（10 分）

6、设计一套软件能够在前端计算机上录入学生成绩，在后端计算机上能输出该学生信息和总分（10 分）

《分布式系统》期末考试试题 III

专业_____ 班级_____ 学号_____ 姓名_____

1、名词解释：

（4） 深度计算与广度计算

（5） 死锁与饿死

2、动态负载均衡有哪三种实现模型？它们的基本思想分别是什么？每种实现模型各有什么优缺点？

3、什么是三层链式模型？什么是三层层次模型？这二种模型的根本区别是什么？三层客户/服务模型中各层的功能分别是什么？

4、在一个无错的立方体中，各节点的负载分布如图 1 所示，试利用 DE 算法进行负载平衡，画出在每一维中的负载平衡图。

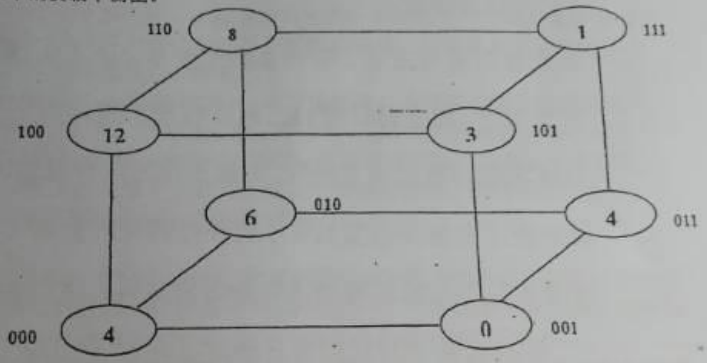


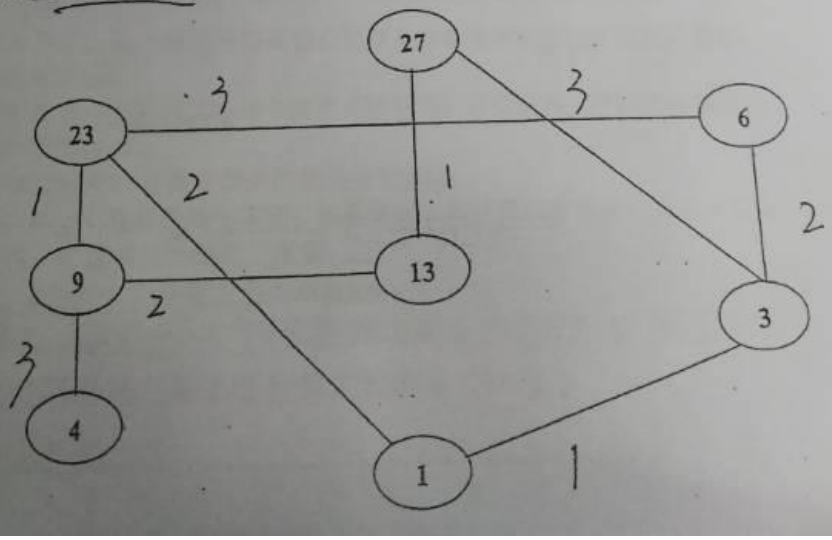
图 1

5、对如图 2 所示的图：

(1) 求边着色的最小.

(2) 对着色图应用扩展维交换方法，给出每个回合后每个节点的负载。

(3) 用程序语言描述(2)中实现节点间负载平衡过程的算法。



1. 程序设计简单，所有程序设计工作均可在客户机上完成	1. 应用变化，牵涉到客户/服务计算模型的全面变化
2. 对前端程序开发售货员素质要求不太高，数据权限工作由网络服务器和数据库系统完成。	2. 数据访问安全性存在不足
	3. 规模受限，系统规模增大时系统性能会直线下降
	4. 系统软件投资大

2.3 三层 C/S 模型

1. 设想　　加入中间层

优势：规模可变、安全性高、降低软件投资、软件实现仍然简单。

2. 定义（P23—24）成份图 3-2，体系结构图 3-3

3. 逻辑平台与物理平台（P30）图 3-6

4. 开发工具与接口（P32）图 3-7

3、 多层客户/服务器计算模型的实现技术有哪些？试介绍其中一种技术的规范或体系结构，并做出你对这种技术的评价（15 分）

4、 试分析任务划分粒度的大小对任务调度的影响（15 分）

5、 逻辑备份可以实现数据库的全面恢复吗？逻辑备份的优点是什么？逻辑备份是否可以替代物理备份？（10 分）

逻辑备份

—— 数据库的逻辑备份包含读一个数据库记录集和将记录集写入文件。

—— （1）输出（Export）输出可以是整个数据库、指定用户或指定表。

—— （2）输入（Import）输入将输出建立的二进制转储文件读入并执行其命令。

—— 2. 物理备份

—— 物理备份包含拷贝构成数据库的文件而不管其逻辑内容。

—— Oracle 支持两种不同类型的物理文件备份：脱机备份（offline backup）和联机备份（online backup）。

—— （1） 脱机备份

—— 脱机备份用在当数据库已正常关闭，数据库处于"offline"时，要备份下列文件：

所有数据文件

所有控制文件

所有联机日志

init.ora（可选的）

—— （2） 联机备份

—— 联机备份可用来备份任何运作在 ARCHIVELOG 方式下的数据库。在这种方式下，联机日志被归档，在数据库内都建立一个所有作业的完整记录。

—— 联机备份过程几条很有力的功能。第一，提供了完全的时间点（point-in-time）恢复。第二，在文件系统备份时允许数据库保持打开状态。

《分布式系统》测试试题 I

1、 名词解释（40 分）

（1）网格计算与集群计算

（2）深度计算与广度计算

（3）负载共享与负载均衡

（4）死锁与饿死

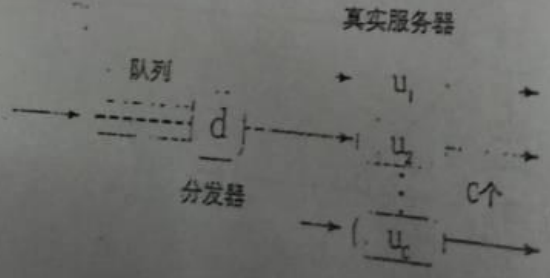
2、 试比较两层与三层 C/S 计算的优缺点（15 分）

3、 多层客户/服务器计算模型的实现技术有哪些？试介绍其中一种技术的规范或体系结构，并做出你对这种技术的评价（15 分）

4、 试分析任务划分粒度的大小对任务调度的影响（15 分）

5、 逻辑备份可以实现数据库的全面恢复吗？逻辑备份的优点是什么？逻辑备份是否可以替代物理备份？（10 分）

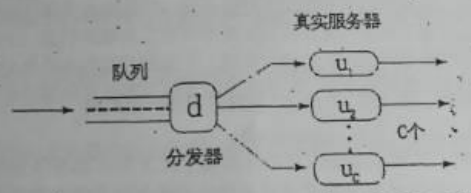
6、 基于分发器的集群结构如下图所示，处于前端位置的分发器将任务分发给真实服务器。试设计一负载均衡算法使系统性能较优。（5 分）



—— 备份方式特性比较

方式	类型	恢复特性
Export	逻辑	可以将任何数据库对象恢复到输出时的状态
Offline Backups	物理	可把数据库恢复到关闭的状态；若数据库运行在 ARCHIVELOG 方式，就可恢复到任何时间点的状态。
Online Backups	物理	可把数据库恢复到任何时间点

6、 基于分发器的集群结构如下图所示，处于前端位置的分发器将任务分发给真实服务器。试设计一负载均衡算法使系统性能较优。（5 分）



3、什么是三层链式模型？什么是三层层次模型？这二种模型的根本区别是什么？三层客户/服务模型中各层的功能分别是什么？

三层层次模型定义：

在三层 C/S 计算模型中，如果每个中间应用服务器负责处理且仅处理物理上它所连接的所有客户机对系统中所有数据库服务器的操作，则称这种三层 C/S 计算模型为三层 C/S 层次模型。

1.2 三层链式模型定义：

在三层 C/S 计算模型中，如果每个中间应用服务器负责且仅负责处理系统中所有客户机对系统中一个数据库服务器的单一数据操作，每个中间应用服务器负责分析与其物理连接的所有客户机的数据操作请求，并调应相应的中间应用服务器提供服务，则称这种三层 C/S 计算模型为三层 C/S 链式模型。

(1) 网络的 4 种体系结构 { Host/Terminal, Workstation / File Server, Client/Server, Peer-to-Peer . . . .

1.1 计算机网络发展现状

1.1.1 体系结构

1. Host 系统 多用户系统 主机+终端
2. 工作站/文件服务器系统 共享文件服务器上的文件和设备
3. 客户机/服务器方式 共享服务器资源，包括文件、设备和其他软硬件
4. 对等网络方式（Peer-to-peer Newark）每台设备处于对等地位，既作客户机又作服务器