名词解释题：

1. 网格计算和集群计算

网格计算：分布式计算，是一门[计算机科学](http://baike.baidu.com/view/92404.htm)。它研究如何把一个需要非常巨大的[计算能力](http://baike.baidu.com/view/1751617.htm)才能解决的问题分成许多小的部分，然后把这些部分分配给许多[计算机](http://baike.baidu.com/view/3314.htm)进行处理，最后把这些计算结果综合起来得到最终结果。网格诞生于那些非常需要进行协作的研究和学术社区。研究中非常重要的一个部分是分发知识的能力 —— 共享大量信息和帮助创建这些数据的计算资源的效率越高，可以实现的协作的质量就越好，协作级别也越广泛。网格计算的本质在于以有效且优化的方式来利用组织中各种异构松耦合资源，来实现复杂的工作负载管理和信息虚拟化功能。（注意，一个组织可能会跨越很多部门、物理位置等。我们此处使用的是 “组织” 一词的抽象意义。）

集群计算：关注计算资源，一般包含同种处理器和操作系统。集群包含的处理器和资源的数量通常都是静态的。从物理上看，集群计算也是在同一个地理位置上进行的。

1. 网络的四种体系结构

Host/terminal 主机+终端，多用户系统，host系统

工作站/文件服务器系统 workstation/file server 共享文件服务器上的文件和设备

客户机/服务器方式 client/server 共享服务器资源，包括文件、设备和其他软硬件

对等网络方式（Peer-to-peer network）每台设备都处于对等地位，既做客户机也做服务器

1. 负载共享和负载均衡
2. 深度计算和广度计算

深度：具体一点就是访问一个顶点之后，我继而访问它的下一个邻接的顶点，如此往复，直到当前顶点一被访问或者它不存在邻接的顶点。同样，算法导论采用了“聪明的做法”，用三种颜色来标记三种状态。但这三种状态不同于广度优先搜索：WHITE 未访问顶点，GRAY 一条深度搜索路径上的顶点，即被发现时，BLACK 此顶点的邻接顶点被全部访问完之后——结束访问次顶点

广度：具体一点就是每个顶点只访问它的邻接节点（如果它的邻接节点没有被访问）并且记录这个邻接节点，当访问完它的邻接节点之后就结束这个顶点的访问。广度优先用到了“先进先出”队列，通过这个队列来存储第一次发现的节点，以便下一次的处理；而对于再次发现的节点，我们不予理会——不放入队列，因为再次发现的节点：无非是已经处理完的了或者是存储在队列中尚未处理的。

1. 死锁和饿死

相同点：二者都是由于竞争资源而引起的，与资源的分配策略有关，因而防止饿死与死锁可从公平性方面考虑如FCFS先到先服务算法。

不同点：①从进程状态考虑，死锁进程都处于等待态（等待某一不可被剥夺资源被释放）,饿死进程可能处于忙式等待(就绪队列上等待可剥夺处理机资源)。（忙式等待：不进入等待状态的等待实际状态为”运行“或者”就绪“忙式等待空耗处理器资源因而是低效的,进程无法向前推进等待某一事件，但不主动放弃处理器而是不断循环检测资源是否可用）。②死锁进程等待永远不会被释放的资源，饿死进程等待会被释放但却不会分配给自己的资源。③死锁一定发生了循环等待，而饿死则不然，这也表明通过资源分配图可以检测死锁存在与否，但不能检测是否有进程饿死。④死锁一定涉及多个进程，而饥饿或被饿死的进程可能只有一个。



* 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
* 分析任务划分粒度的大小对任务调度的影响
* 对网格计算发展趋势的看法

﻿ ﻿网格计算专门针对复杂科学计算的新型计算模式。这种计算模式是利用互联网把分散在不同地理位置的计算机组织成一个虚拟的超级计算机，每一台参与计算的计算机就是一个节点，而整个计算是由成千上万个节点组成的一张网格。这样组织起来的虚拟的超级计算机有 2 个优势：①数据处理能力超强，网格计算是分布式计算的一种，参与工作的是计算机网络，显然比以个人计算机为的单位的计算方式具有更加强大的数据处理能力。②能充分利用网上的闲置处理能力，网格计算模式首先把要计算的数据分割，通常实现的软件是一个预先编制好的屏幕保护程序，然后，不同节点的计算机可以根据自己的处理能力下载一个或多个数据片断和这个屏幕保护程序，只要位于节点的计算机的用户不使用计算机时，屏保程序就会工作网格计算至少需要具备以下 3 种基本功能：①任务管理：用户提交任务，为任务指定所需资源，删除任务并监测任务运行状态②任务调度：用户提交的任务由该功能按照任务类型、所需资源、可用资源等安排运行日程和策略。③资源管理：确定并监督网络资源状况，收集任务运行时的资源占用数据。为实现网格计算的目标，必须重点解决以下 3 个问题，这也是目前 Internet 普遍存在的问题：①异构性：由于网格由分布在广域网上下同管理域的各种计算资源组成，怎样实现异构机器间的合作和转换是首要问题。②可扩展性：要在网络资源规模不断扩大、应用不断增长的情况下，不降低性能。③动态自适应性。

* 设计一套软件能够在前端计算机录入成绩，在后端输出该学生信息和总分
* 动态负载均衡有哪三种实现模式？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？

基于DNS负载均衡，基于硬件负载均衡，基于软件负载均衡。三种方案各有优劣，DNS负载均衡可以实现在地域上的流量均衡，硬件负载均衡主要用于大型服务器集群中的负载需求，而软件负载均衡大多是基于机器层面的流量均衡。在实际场景中，这三种是可以组合在一起使用。

1. 使用DNS做负载均衡的方案，天然的优势就是配置简单，实现成本非常低，无需额外的开发和维护工作。但是也有一个明显的缺点是：当配置修改后，生效不及时。这个是由于DNS的特性导致的，DNS一般会有多级缓存，所以当我们修改了DNS配置之后，由于缓存的原因，会导致IP变更不及时，从而影响负载均衡的效果。另外，使用DNS做负载均衡的话，大多是基于地域或者干脆直接做IP轮询，没有更高级的路由策略，所以这也是DNS方案的局限所在。
2. 硬件的负载均衡，性能是非常的好，每秒能处理的请求数达到百万级，即几百万/秒的负载，价格昂贵。因为这类设备一般用在大型互联网公司的流量入口最前端，以及政府、国企等企业。采用 F5 这类硬件做负载均衡的话，主要就是省心省事，买一台就搞定，性能强大，一般的业务不在话下。而且在负载均衡的算法方面还支持很多灵活的策略，同时还具有一些防火墙等安全功能。
3. 软件负载均衡是指使用软件的方式来分发和均衡流量。软件负载均衡，分为7层协议 和 4层协议。网络协议有七层，基于第四层传输层来做流量分发的方案称为4层负载均衡，例如 LVS，而基于第七层应用层来做流量分发的称为7层负载均衡，例如 Nginx。这两种在性能和灵活性上是有些区别的。基于4层的负载均衡性能要高一些，一般能达到 几十万/秒 的处理量，而基于7层的负载均衡处理量一般只在 几万/秒 。基于软件的负载均衡的特点也很明显，便宜。在正常的服务器上部署即可，无需额外采购，就是投入一点技术去优化优化即可，因此这种方式是互联网公司中用得最多的一种方式。

* 比较二层和三层C/S计算的优缺点
* 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？

