1. 各位老师下午好，我的毕设题目是CG树顶端节点集群的设计与实现，指导教师是卢伟老师。
2. 首先介绍一下CG树。CG树是一个分布式集群的模型，其特点就是将集群的节点分级分层，以前端调度节点为根节点。这幅图就是一颗CG树的结构图，从图中我们可以看出，这些叶子节点的父节点其实就是在叶子节点中选择一个节点作为父节点，我们一般选择性能最优的作为父节点，这一层也是一样的。。。就是这样组成一个树状结构。CG树目前主要用于高清视频点播。
3. 由CG树组成CG森林，顶端节点位于整个系统的最顶端。在视频点播系统中，视频资源存放在CG树叶子节点上，由一个数据库保存所有的资源信息。当客户端发送请求时，由顶端节点查询数据库，进行调度，然后把保存资源的叶子节点的地址告知客户端。  
   如果CG森林的规模扩大到一定程度，顶端节点就会成为整个系统的瓶颈。我在(2.66GHz, 四核CPU)的PC上做过测试，顶端节点每秒钟只能处理大约一万次左右的资源请求，如果系统面向全世界服务，显然这样的性能是不能满足需求的。
4. 为了解决顶端节点性能瓶颈的问题，提出了将顶端节点改为顶端节点集群的想法。具体做法就是使用双层结构，第一层是负载均衡器，仅仅负责客户请求的转发，第二层是若干台真实服务器，负责数据库的操作，资源的调度等等。
5. 下面是顶端节点集群的具体设计与实现。集群采用了心跳机制，心跳信息由第二层的真实服务器发送给第一层的负载均衡器，使负载均衡器及时了解每台服务器的工作状态。因为心跳的发送比较频繁，因此使用UDP协议，在未来的实现中也可以在心跳中加入其它信息比如每台服务器的负载信息等等。  
     
   整个系统使用MySQL作为数据库。因为数据库更新操作不频繁，而且处理客户端的请求全部是查询操作，因此我采用的是每台真实服务器上维持一个内容相同的数据库副本，牺牲空间来换取时间上的性能。但是如此要考虑数据库一致性的保持。  
   对于数据库的一致性问题，我实现方式是当数据库需要更新时，由负载均衡器使用TCP将消息发送给每一个真实服务器，由真实服务器执行数据库更新操作。当然也可以使用更加成熟的MySql cluster或者 slave/master数据库集群的方式来实现。另外，为了提高数据库操作的效率，我实现了一个简单的数据库连接池。
6. 负载均衡器转发客户请求需要一个调度算法。我实现了三种调度算法，首先是轮转调度，这是最简单的一种，然后是源地址哈希调度，即通过哈希函数将客户端地址映射到唯一的一台服务器上。最后是加权轮转调度，就是为每台服务器设置一个权值，根据权值的比例发送到每台服务器。
7. 这个流程图是负载均衡器的启动流程图，读取配置文件，初始化，然后创建五个线程，分别用于。。。
8. 这个流程图是真实服务器的启动流程图。。。。。
9. 最后，是对整个集群的性能的测试。测试环境是利用实验楼404的机器和千兆网的网络环境，总共使用了6台机器。这是详细配置。
10. 因为客户端的请求使用的是UDP,所以先测试丢包率，确定服务器的最大处理能力。测试方法就是使用三台PC机不断的发送请求，服务器简单的记录收到的请求数。测试结果就是当发送速率在每秒钟50万次左右的时候，丢包率是百分之是可以接受的。
11. 然后分别测试改进前的顶端节点和改进后的顶端节点集群的性能，测试方法是首先建立一个200万条记录的数据库，然后由客户端不间断的发送请求，每次请求1-10个视频资源，记录满负载状态下整个服务器的处理能力。  
    我测试了改进以前的顶端节点还有分别挂载了2、3、4台真实服务器的情况下的整个系统的请求处理能力，从表中可以看出，随着服务器数量的增加，性能的增长基本上是呈线性的。
12. 这幅图是满负荷状态下真实服务器的资源使用情况。可以看出CPU的使用率在90%以上。
13. 这幅图是使用四台真实服务器而且满负荷的情况下负载均衡器的负载情况，可以看出真实服务器满负载时，负载均衡器的CPU使用率还是比较低的，集群的规模完全可以继续扩大。
14. 然后是数据库更新操作的测试，这个数据随着语句查询条件的不同波动比较大，但是也基本可以说明整个集群的数据库操作性能。
15. 最后是结论，由测试结果可以看出，顶端节点集群处理客户端请求的能力明显提高。  
    系统的缺点  
    第二层的所有服务器都使用相同的数据库，有较大的数据冗余。另外，该系统只能用于数据库更新不频繁的情况下。  
    解决办法就是可以考虑将第二层的服务器分组，同时数据库的内容也可以分组存放，这样就可以进一步提高系统的效率。
16. 最后是致谢。

因为整个系统的演示需要至少四台计算机，条件有限，所以在这里我只能使用进程来模拟每台计算机，我在这里是把请求的内容还有处理结果打印出来了。