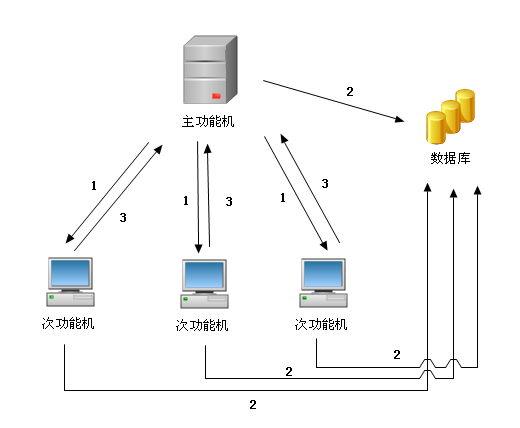
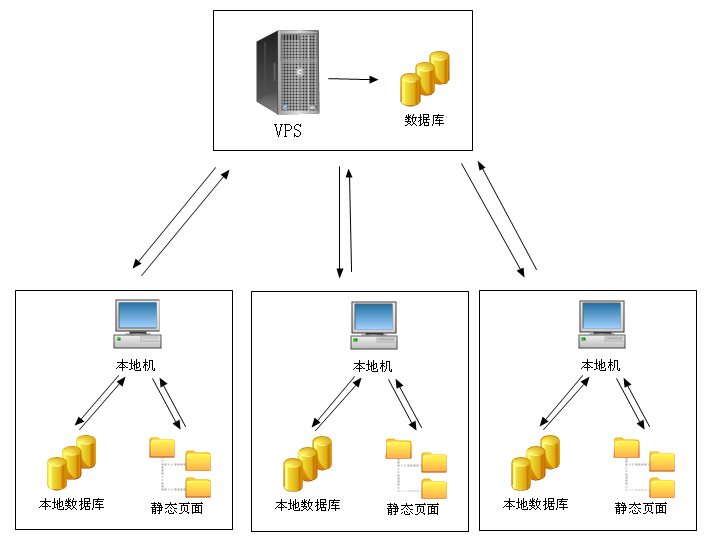
**利用已有单机程序组建分布式模型的分析与实例**（讲稿）

1. 问题产生
   1. 随着从事IT时间越长，自己积累的软件也就越多，相对于IT行业的高速发展，积累的软件中有些已经不能符合市场需求，有些软件的部分功能还适用，如何处理这些鸡肋软件？
   2. 随着学习知识的深入和自身水平的提高，原有的代码已经不能满足于从新利用的需求。如何使这些代码回复青春？
   3. 现在有众多的免费或性价比高的程序可以使用，如何快速使用其中的某些功能？
   4. 而对于一些客户来说，他们积累了不少退出主流市场的计算机，如何利用它们为正在使用的软件进行服务？
   5. 如何利用已有的软硬件，从事快速开发、原型开发——以保证质量和相对稳定性为前提——争取客户源（特别是对于我这样的工作室）至关重要。

（穿插分布式基本概念：而分布式（分布式计算）是一门计算机科学，它研究如何把一个需要非常巨大的计算能力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进行处理，最后把这些计算结果综合起来得到最终的结果。）

1. 案例
   1. 服装厂原有程序的分布式结算。
      1. 需求产生：  
         这是为某家广东中山的服装厂的开发的一个软件，这个软件开发于2001年，使用的是delphi + mssql 2000 进行开发的。这套系统应用于全厂的生产及管理的各个环节，涉及员工的考勤，工资结算，库存管理，飞票打印等。由于这家厂每星期都有员工的进出，所以每个星期的数据处理量都很大，单机处理较难适应其需求。需要一个能快速处理数据的解决方案。
      2. 原系统特点：
         1. 程序太旧，当时限于知识面太窄，很多没有考虑得到。
         2. 随着客户要求的不断变更，程序多次打补丁，系统扩展性不强。
         3. 数据处理量最大的地方是员工的结算。
      3. 分布式解决方案：（结合图展开说明）  
         
         1. 主功能机划分任务，发放给各个次功能机。
         2. 次功能机连接到数据库，读取并处理数据。
         3. 返回通知主功能机任务结束。
   2. CMS系统 静态页面生成。
      1. 需求产生：  
         在我们工作室成立初期，基于资金方面的考虑，我们采取租用VPS类型的主机来做服务器。随着客户的增多和客户数据的增多，限于VPS的cpu配额和内存配额，访问网页的速度非常缓慢，需要一个能提高网站浏览速度的解决方案。
      2. 原系统特点：
         1. 采用Apache + mysql + php形式的WWW服务模式。
         2. 经分析，是CMS系统对的Mysql频繁访问导致整体速度缓慢。
      3. 尝试的解决方案：（说明不可行原因）
         1. 增加内存配额。
         2. 优化mysql
         3. 开启Apache缓存
         4. 更换CMS系统
      4. 分布式解决方案：（结合图展开说明）  
         
         1. 生成静态页面。
         2. 将数据分发给本地机。
         3. 本地机获取数据上传到网站上。
   3. Blender3D 渲染工厂。
      1. 需求产生：  
         工作室需要做一些3D动画，而我们使用的是Blender3D软件（一个开源的好东西）。Blender3D没有内置的网络渲染的系统。需要一个网络渲染的环境来提高动画渲染的速度。
      2. 原系统特点：
         1. 可以使用命令行执行渲染操作。
         2. 可以选择渲染哪一帧。
      3. 尝试解决方案：（说明不可行原因）  
         使用已有的网络渲染工具yadra。
      4. 分布式解决方案：（结合图展开说明，暂缺图）
         1. 任意台机分发任务。
         2. 从共享文件夹获取要进行渲染的文件。
         3. 渲染到共享文件夹。
   4. Blender3D + hypershot 渲染流程。
      1. 需求产生  
         blender3D的内置渲染器比较慢，hypershot速度很快效果也很好。需要将两者结合进行动画渲染。
      2. 原系统特点：
         1. Blender3D内置Python脚本，有很好的API及多种脚本模块。
         2. 经分析，Hypershot的文件格式与Obj文件格式有很多相似的地方。
         3. Blender3D有很好的Obj输出脚本。
      3. 分布式解决方案。（结合图展开说明，暂缺图）
         1. Blender3D输出Obj。
         2. Erlang输出bip。
         3. Erlang写入队列文件。
         4. 各机器进行渲染处理。
   5. 在线网游开发
      1. 目的：  
         这是工作室自行设计开发的一个带实验性质的web game，用到了flash + erlang+php+mysql+…为了快速开发，使用了很多已有的程序。（说明系统特点和为何使用分布式）
      2. 设计方案：（举战斗和A\*寻径的例子，说明为何选用erlang的优点，展开说明）（结合图展开说明，暂缺图）
         1. Erlang服务器接受需求。
         2. 随机向各个功能端发送需求。
         3. 获得需求结果。
         4. 返回给请求机
2. 总结（展开说明）
   1. 分布式处理的类型
      1. 数据分布处理。
      2. 功能分布处理。
   2. 分布式处理的重点注意事项：细化程度
   3. 利用已有程序进行分布式处理的要求：
      1. 处理的数据为外部存储类型。
      2. 数据可以无序处理。
      3. 程序可以由命令行形式进行调用。
      4. 程序可方便的使用或简单的加入网络连接功能。
      5. 源程序程序耦合度低。
   4. 分布式原型、分布式系统、多线程并行处理、虚拟机 简单比较（分析其类同，加图）