**1 概述**

**1.1 神龙架构的特点**

阿里云官方文档对于神龙架构的描述如下：  
保留了普通云服务器的资源弹性，并因嵌套虚拟化技术让弹性裸金属服务器保留了物理机的体验。

**1.2 理解上的难点**

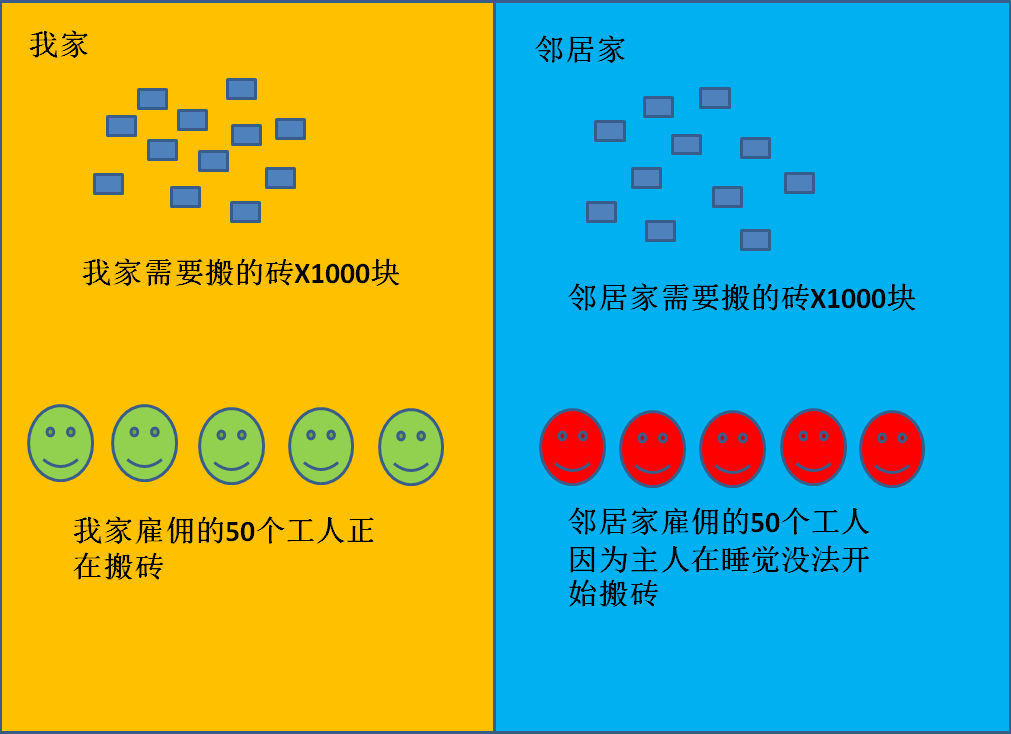
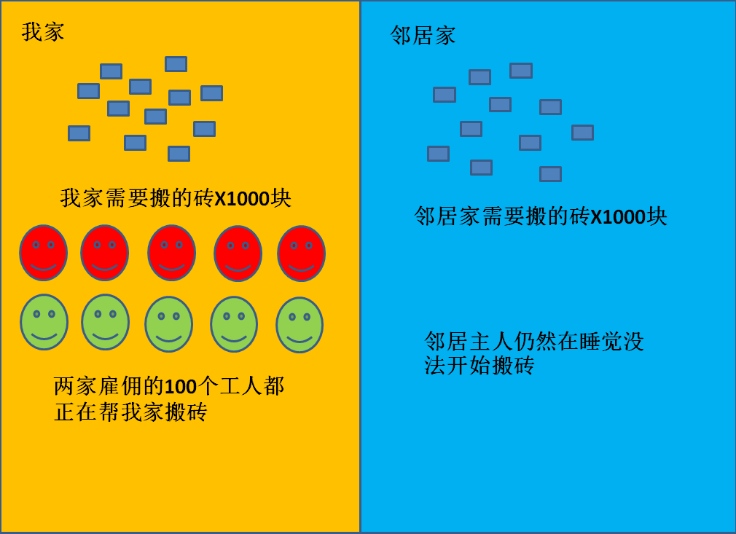
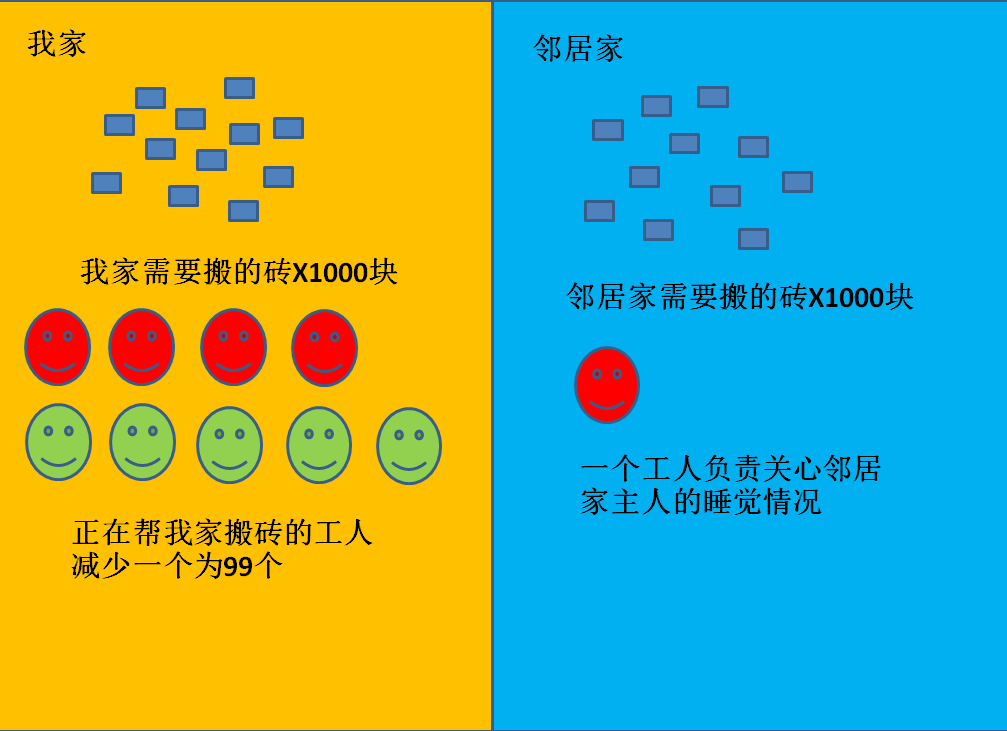
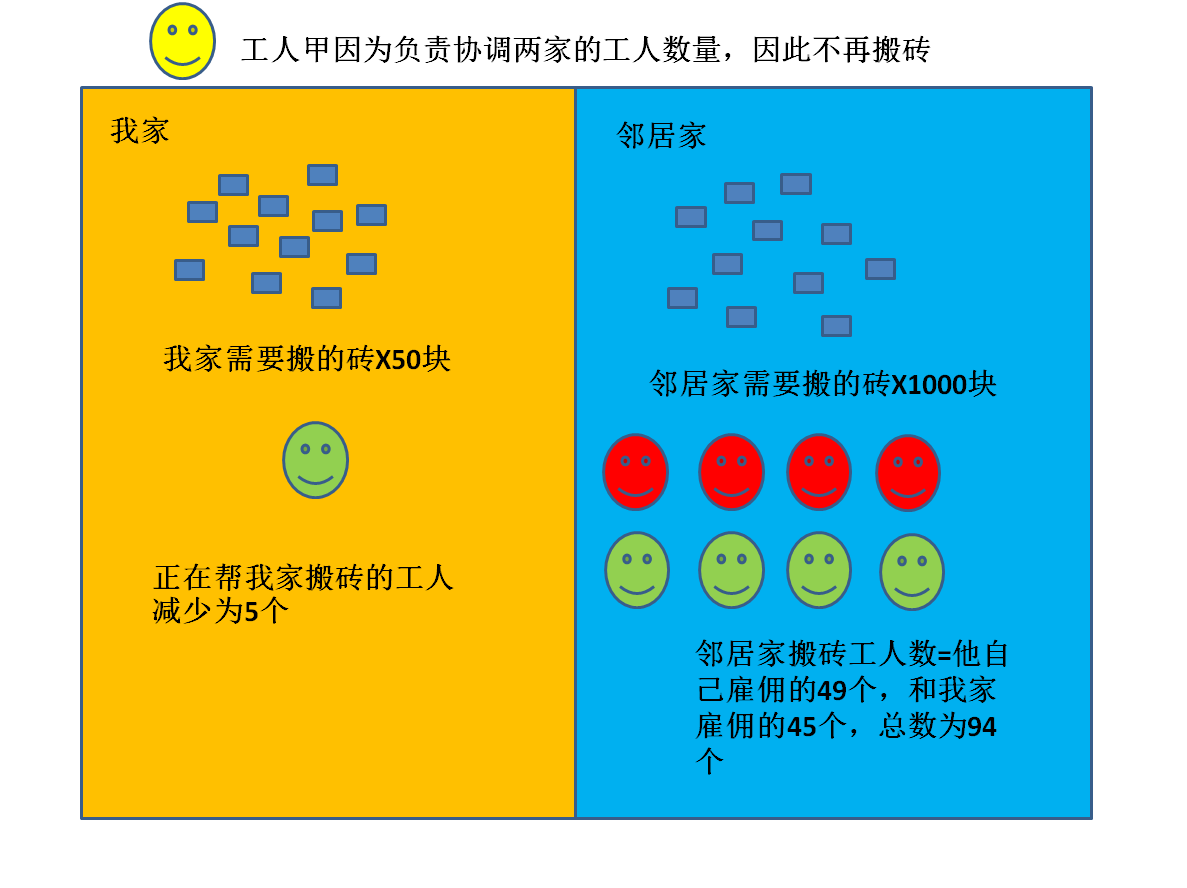
同时拥有云服务器的资源弹性和保留了物理机体验的特点容易让用户在需要深入了解神龙架构时产生一个疑问：神龙架构到底是虚的还是实的，如果虚实融合又怎么来理解什么是虚实融合？通过什么手段做到的？

**1.3 本文重点说明的问题**

结合以上神龙架构的特点和理解上的难点，本文详细的对于神龙架构进行研究分析，说明神龙架构是如何做到同时拥有云服务器的资源弹性和保留了物理机体验的目标的。

**2 为什么需要发明神龙架构**

**2.1 以搬砖为例说明虚拟化技术的特点**

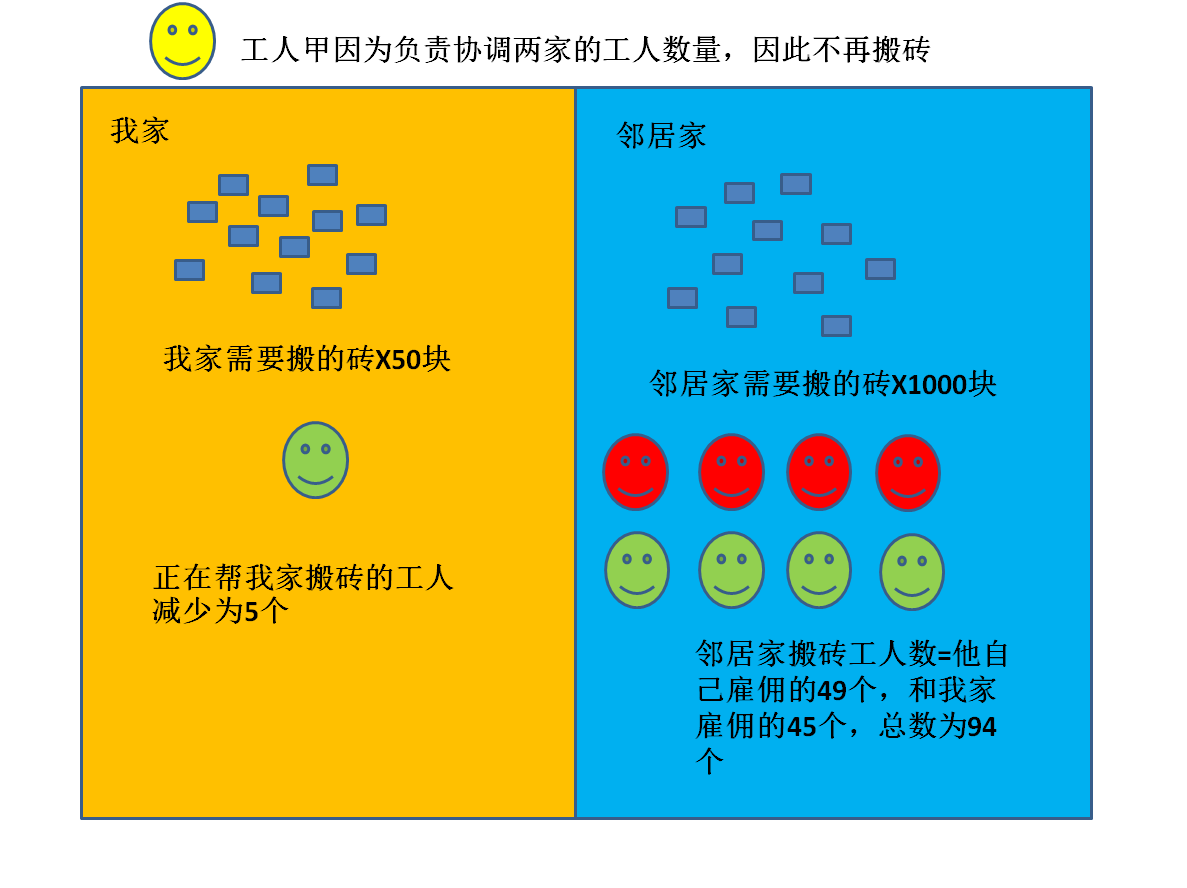
把物理机变成虚拟机的这个技术，就是“虚拟化”。比如我家里装修有100块砖需要搬运，邻居家也在装修同样有100块砖需要搬运，我们各请了50个搬运工，当工人到达时发现邻居家的主人在睡觉，因此他家的50个工人只能等他睡醒再搬砖，我家请的50个工人则可以直接帮我开始搬砖，情况如下图所示：  
  
正好两家的工人来自于同一个公司于是包工头过来看了一下，发现邻居家的工人在空闲状态觉得效率很低。于是决定既然邻居家的工人目前空闲于是一起来帮我家搬砖。和我商量费用并不增加，工人增加50个，我自然非常开心，觉得多给了我家50个工人。于是邻居家的工人也过来开始帮我家搬砖如下图所示，我们称这个100个工人为计算节点：  
  
包工头心里在想一个事情，他马上需要去其他工地，现在100个工人都在帮我家搬砖，因此进度很快，但是邻居万一睡醒了也要开始搬砖怎么办，于是他抽了一个工人甲出来看着邻居家动静，一旦邻居家醒了需要开始搬砖，则把暂时帮我家搬砖的工人还给他并且工人数量至少50个。  
于是甲离开了搬砖行动，专门看着邻居家主人防止他突然醒过来，帮我家搬砖的工人数目前为99个。这个负责关心邻居家主人睡觉情况并负责后续把工人还给他的甲，我们称他为管理节点。  
  
邻居家主人睡醒了，甲于是立即从我家将50个工人安排到邻居家开始搬砖，同时和我商量，因为之前我家搬砖的劳动力多了一倍，因此1000块砖被搬了只剩50块了，而邻居家的砖还是1000块，因此除了邻居雇佣的50个工人外能否我家只留5个工人，我自己雇佣的45个工人也帮邻居家搬砖，我欣然同意，因此两家搬砖的工人数再一次改变如图所示：  
  
这个整个过程即为弹性计算的本质，前提即是虚拟化，如果缺少了虚拟化技术，代表我和邻居家雇佣的工人来自于两个公司，没有人来统筹决定每家搬砖的工人数，因此即使邻居在睡觉，他雇佣的工人空闲着也无法过来帮我搬砖，能够做到搬砖的工人灵活调配的前提就是将两家人家雇佣的工人进行统筹考虑再进行分配。对于用户的好处在于，我和邻居家都有1000块砖要搬，但是搬砖的时间不同，我在搬砖的时候他在睡觉而他睡醒需要搬砖的时候我家的砖已经快搬完了，同样100个工人的劳动力在不同的时间段里被我们用到了极致。

**2.2 虚拟化技术的瓶颈**

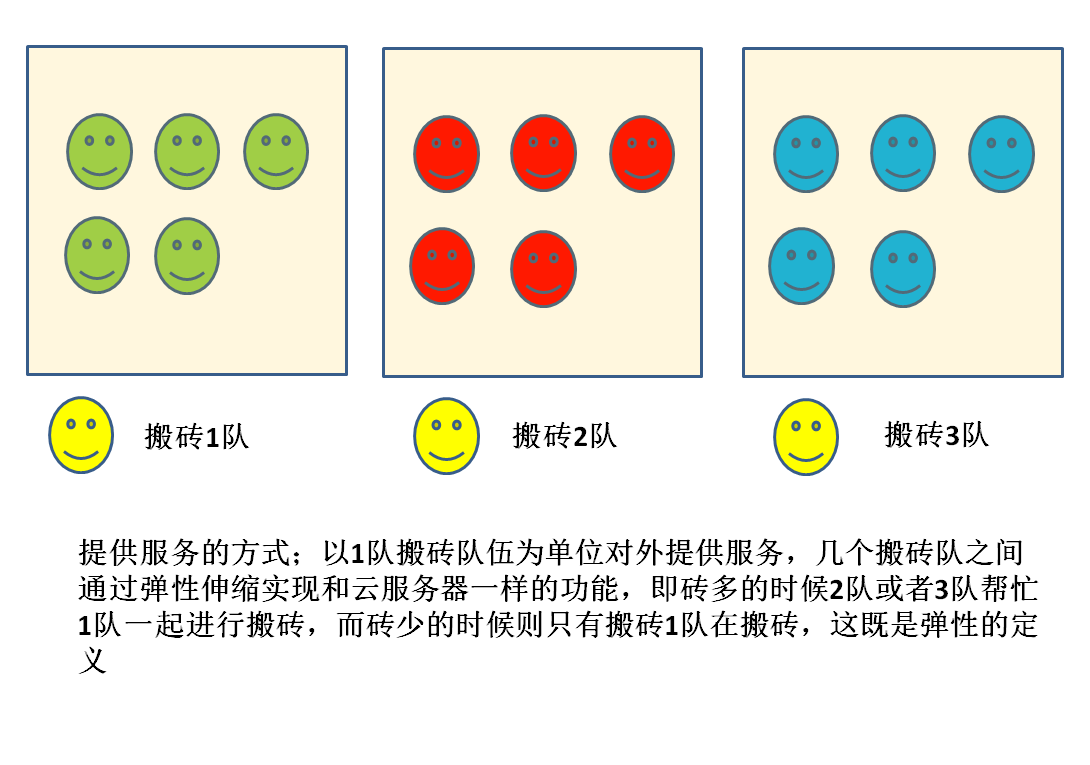
从以上搬砖的例子中可以发现，因为工人甲负责协调我和邻居家搬砖的工人安排因此他本身不再负责搬砖，也就是100个劳动力中抽调了1个工人的劳动力来做管理工作，实际搬砖的劳动力为99个。按照原来雇佣的劳动力，我家雇佣了50个工人，邻居家雇佣了50个工人，总的劳动力为100人，因此实际搬砖的劳动力少了1个，但因为我和邻居家搬砖时间的错开并且以我们的感受都享受到了远大于50个工人的劳动力（实际我家99个，邻居醒来后他家为94个）因此满足我们的需求，也就并不太在意100个工人中有1个来作为协调我们两家工人数的管理人员。隐患在于如果我家砖搬完了，邻居家的搬砖工人上升到99个，他发现需要再快一点，要求100个工人搬砖，这时候我和邻居将同时发现劳动力因为有人去做管理工作而少了一个，我们两家总共花了100个工人的钱，却总共只能享受到99个工人的劳动力。  
事实上这1个管理人员确实是整个体系中无法解决的瓶颈，代表只要采用虚拟化和弹性计算，就代表100个劳动力必须选择1个管理人员，实际上只能有99个劳动力进行搬砖。**反映到云计算上就是只要物理服务器采用虚拟化技术，就必须配置管理节点，因此单台物理服务器所提供的计算力在原来的基础上需要打折扣，造成物理服务器基础上采用虚拟化技术后生成的云服务器的计算性能必然比物理服务器要差**。虽然用户因为云服务器集群的弹性计算功能未必能感受到。  
这个瓶颈原来在云服务提供商中都存在，似乎成为了必然，因为觉得没有办法解决因需要管理节点而造成的总计算力损失因此也没有云服务商去讨论深究这个问题。而阿里云神龙架构即破天荒的在这个瓶颈问题上开始动刀子，想做到的目标就是既然100个工人搬砖，就要全部搬砖，但同时也需要有手段来管理和控制我家和邻居家不同时间搬砖的工人数。

## 3 神龙出世

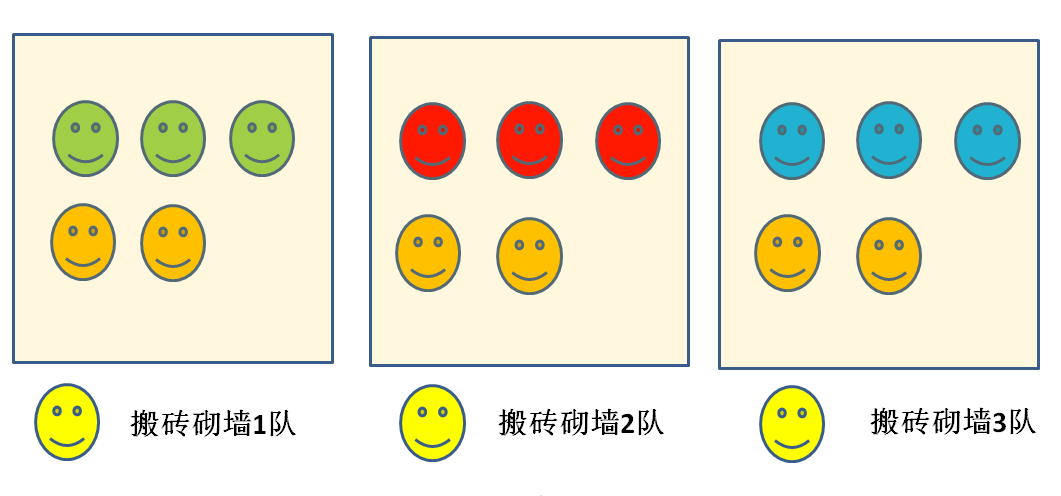
### 3.1 继续说我们的搬砖问题

  
第2章中指出只要采用虚拟化和弹性计算，就代表100个劳动力必须选择1个管理人员，实际上只能有99个劳动力进行搬砖。而神龙想做到的目标就是既然100个工人搬砖，就要全部搬砖，但同时也需要有手段来管理和控制我家和邻居家不同时间搬砖的工人数。以上图为例就是让黄色的那个被抽出来负责管理工作的工人回去仍然搬砖去。  
包工头看着目前的情况想，如果要维持两家搬砖的工人弹性灵活，就需要100个工人抽1个工人做管理工作，那如果1000个工人就需要损失10个，10000个工人就需要损失100个。工程量越大则损失的劳动力就越多，当业务得到大规划发展时这个损耗的问题如果能够解决就可以大幅度的提升搬砖的效率。阿里云在神龙架构问世前的虚拟化损耗其实比搬砖的例子更大，平均虚拟化损耗为10%左右，代表100个工人，只有90个在搬砖，剩下的10个在做搬砖管理工作。

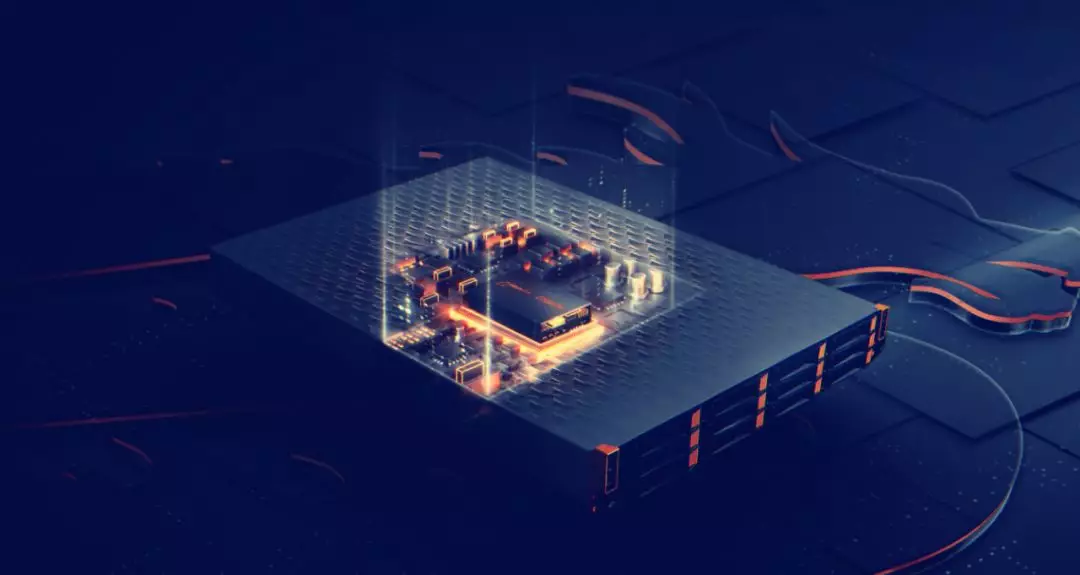
### 3.2 神龙1.0的核心理念

结合实际情况包工头决定让原来被抽出来做管理工作的工人甲仍然回去搬他的砖去，因为他的力气大的特点意味着他本来就适合搬砖而不适合管理工作。而工人队伍的管理工作采用项目经理制，即引入专业管理人员来负责工人队伍的管理，使工人只负责搬砖，当然引入专业管理人员后，成本肯定是上升的，但是搬砖的劳动力就没有损耗了。采用项目经理制后的情况如下图所示：  
  
需要重点指出的是，搬砖队伍弹性伸缩的最小单位是1个队伍，如果搬砖1队忙不过来，只能要求整个搬砖2队或者搬砖3队整个队伍过来帮忙，而不能说从搬砖2队仅抽取几个工人过来帮忙。通过这种结构确保了每队搬砖队伍的劳动力因为有专门的项目经理进行管理而不会有损耗。这里先不引伸到神龙架构，因为还有一个重要的问题没有提到。

### 3.3 异构计算的本质是搬砖和砌墙的结合

包工头从自身业务的发展进行分析，发现我和我的邻居除了搬砖外还有砌墙的需求，而原先的工人全部都是擅长于搬砖而没有擅长砌墙的泥瓦工，让搬砖的工人去砌墙固然也是可以的，但是速度和质量显然不及专门砌墙的泥瓦工。因此包工头的做法是，在原来的队伍中加上泥瓦工，这样1支队伍就即可以搬砖又可以砌墙了，如下图所示：  
  
搬砖工人和泥瓦工结合的方式就叫异构，搬砖工人在搬砖的时候泥瓦工在砌墙，就叫异构计算。

### 3.4 神龙1.0的特点总结

到这里为止虽然没提过神龙，其实已经把神龙1.0的特点全部说明白了，这里把搬砖砌墙队伍的问题和神龙1.0的特点结合起来来作为神龙1.0的特点总结。  
搬砖砌墙队伍为了解决劳动力损耗的问题搬砖的全部搬砖，砌墙的全部砌墙，管理工作由专门的项目经理负责。反映到神龙1.0中即阿里云为了解决虚拟化损耗的问题新造出一个带有智能芯片的专用板卡负责虚拟化调度，这块专用板卡称为MOC卡，外观如下图所示：  
  
为了解决搬砖砌墙任务而专门成立的带项目经理的搬砖砌墙队即是阿里云的神龙云服务器如下图所示：  
  
业内一般管它叫弹性裸金属服务器。根据阿里云官方文档：弹性裸金属服务器（ECS Bare Metal Instance）是一种可弹性伸缩的高性能计算服务，计算性能与传统物理机无差别，具有安全物理隔离的特点，分钟级的交付周期将提供给您实时的业务响应能力，助力您的核心业务飞速成长。现在能够理解了为什么计算性能与传统物理机无差别，因为神龙云服务器就是物理机，所以当然计算性能和物理机没有差别，此外它又可以像云服务器一样弹性伸缩，并且交付周期为分钟级。  
一句话总结神龙1.0的特点就是，神龙云服务器兼具了物理机和云服务器优点，本质上是可以弹性伸缩的物理机并且这种物理机专门为提供云服务设计。

### 3.5 神龙1.0的瓶颈

回到搬砖的例子，包工头又碰到了新问题，邻居他自己就是一个项目经理，对于搬砖和砌墙有特殊的要求，他要求一个搬砖砌墙队内的100个工人上午搬左边的砖，同时砌右边的墙；下午搬右边的砖，同时砌左边的墙。而目前搬砖砌墙队的项目经理没经历过这种情况，不知道该怎么调配队伍内的工人。  
这就是神龙1.0的瓶颈，虚拟化其实分成两个方向：一个方向是虚拟化组合，把一堆物理机粘成一个大的虚拟机；另一个方向是虚拟化切分，把一个物理机切成一堆小的虚拟机。神龙1.0做到了虚拟化组合，但并没有做到虚拟化切分，在例子中即为搬砖砌墙队的项目经理只知道在自己的队伍不够用时叫别的队伍来帮忙，但是自己的队伍内怎么去响应我邻居家的要求，上下午通过队内工人调配做到劳动力弹性却没有办法实现。  
这个问题在神龙2.0中得到了解决。

## 4 飞龙在天

### 4.1 神龙1.0需要进一步解决的问题

根据上一章总结出的神龙1.0的瓶颈为，虚拟化其实分成两个方向：一个方向是虚拟化组合，把一堆物理机粘成一个大的虚拟机；另一个方向是虚拟化切分，把一个物理机切成一堆小的虚拟机。**神龙1.0做到了虚拟化组合，但并没有做到虚拟化切分。**  
这牵涉到一个未来阿里云发展的问题，就是只有解决了神龙服务器的虚拟化切分问题，才能和目前作为ECS宿主机的其他各类物理服务器并存甚至逐步替代掉目前的物理服务器，因为随着技术的发展，物理服务器不断的被升级替换是个必然，以阿里云的开拓精神不可能允许神龙只能作为一个整体的物理机向外提供服务这种情况长期存在，理想的状态是用户即可以租用整个神龙物理服务器，神龙物理服务器也可以作为ECS的宿主机以ECS虚拟机向用户提供服务，这对于用户来说是真正的按需选择，对阿里云来说可以逐步统一物理机的型号以消除一般物理服务器和神龙服务器并存的现状。

### 4.2 神龙2.0解决了这个问题

听上去将神龙物理机虚拟化好像是很简单的事情，毕竟阿里云的虚拟化技术非常成熟，神龙物理机也是物理机为什么就有难度。但其实并不是这样，因为神龙和其他物理机最大的不同是，在神龙架构出现前，用户在阿里云租用的ECS都是虚拟机，阿里云没有提供物理机实例供用户选择。但神龙的要求是即可以实也可以虚，这也是用户对于神龙能够虚实结合的一个比较容易迷惑的点：神龙到底是虚拟机还是物理机？这和神龙2.0解决的问题是一样的。以下根据阿里云文档来总结神龙架构的特点和用户可能会产生的疑问：  
1、神龙是物理机还是虚拟机？  
**文档原文：**弹性裸金属服务器（ECS Bare Metal Instance）是基于阿里云完全自主研发的下一代虚拟化技术而打造的新型计算类服务器产品。  
**解读：**神龙服务器是物理机，因此用户租用的神龙实例（弹性裸金属实例）是一个个真实存在的物理机。至于阿里云将神龙物理服务器虚拟化后以ECS实例的方式向用户提供虚拟机服务，则ECS虚拟机的宿主机是一般物理服务器还是神龙物理服务器其实和用户并没有什么关系。  
2、为什么说神龙兼具虚拟机的弹性和物理机的性能及功能特性？  
**文档原文：**与上一代虚拟化技术相比，下一代虚拟化技术的主要创新在于，不仅保留了普通云服务器的弹性体验，而且保留了物理机的性能与特性，可全面支持嵌套虚拟化技术。  
**解读：**先解读后半句，因为很简单，用户租用的神龙实例本身就是物理机，当然保留了物理机的性能与特性，所以用户可以自己在神龙实例上虚拟化，不存在在虚拟机上再次虚拟化的问题，因为神龙不是虚拟机。前半句就是阿里云的重大突破，保留了普通云服务器的弹性体验，云计算的核心就是弹性伸缩，而之前的弹性伸缩都是通过虚拟机的镜像来实现的，要让神龙物理服务器也能弹性伸缩，阿里云解决的是物理机镜像完全兼容ECS云服务器实例的镜像系统，即神龙物理机的镜像和ECS虚拟机的镜像是兼容的，所以神龙实例才能像ECS实例一样能够做到弹性伸缩，保留了云服务器的弹性体验。  
3、上云和虚拟化的关系？不用虚拟机可以上云吗？  
**文档原文：**通过阿里云自主研发的虚拟化2.0技术 ，您的业务应用可以直接访问弹性裸金属服务器的处理器和内存，无任何虚拟化开销。弹性裸金属服务器具备物理机级别的完整处理器特性（例如Intel VT-x），以及物理机级别的资源隔离优势，特别适合上云部署传统非虚拟化场景的应用。  
**解读：**前两句很好理解，因为神龙实例是物理机，具有物理机的一切性能和特点包括了资源隔离优势。后半句是回答用户常见的容易混淆的概念错误，就是经常会将云计算和虚拟化技术画等号，认为用了虚拟化技术以后就是云计算。其实云计算和虚拟化是两个不同的概念，当然虚拟化是云计算得以实现的重要前置技术之一，但是如果用户在云上购买了5台神龙实例，背后其实是5台神龙物理服务器，因为神龙物理服务器之间也能做到和虚拟机一样的弹性伸缩功能，所以用户使自己的服务上云并且没有使用虚拟化技术。因此云计算不等于虚拟化。

##### 最后再次提醒：我们用的ECS虚拟机的宿主机是不是神龙服务器和我们并没有关系，那是阿里云的事情。这个问题反复提及并且极其重要的原因是不要把宿主机是神龙物理服务器和神龙实例混淆起来造成虚实不分，阿里云未来的数据中心的服务器是否会全部替换成神龙物理服务器也并不是我们需要关心的问题，我们只需要知道，我们订购的神龙实例是真实存在的一台台弹性裸金属服务器，是物理机，只是放在阿里云数据中心里而已。