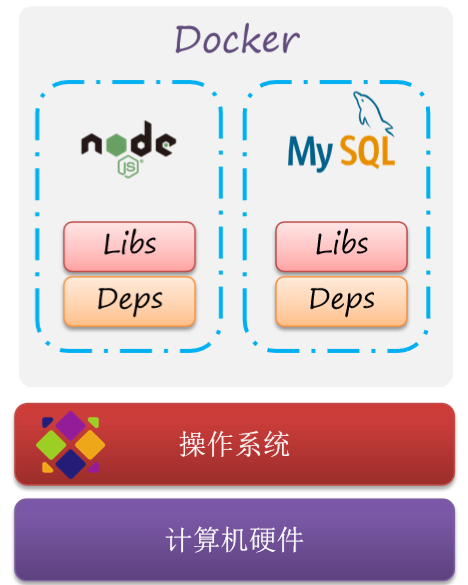
### Docker与虚拟机

根据之前的学习，docker中的应用（其实就是镜像）可以有自己的系统函数库，因此可以把应用视作是一个只运行着应用的微型的Linux操作系统镜像，容器视作是一个只运行着应用的微型的Linux操作系统，又因为应用运行在Docker上，那么Docker就可以视作与虚拟机同一级别。

书接上一回，我们之前已经初始docker的原理，我们知道Docker可以实现一个应用在不同的Linux操作系统上去运行和部署，而我们的虚拟机（**虚拟机也可以安装在linux系统上**）也可以达到类似的效果，那么这两者之间的实现有什么差别呢？

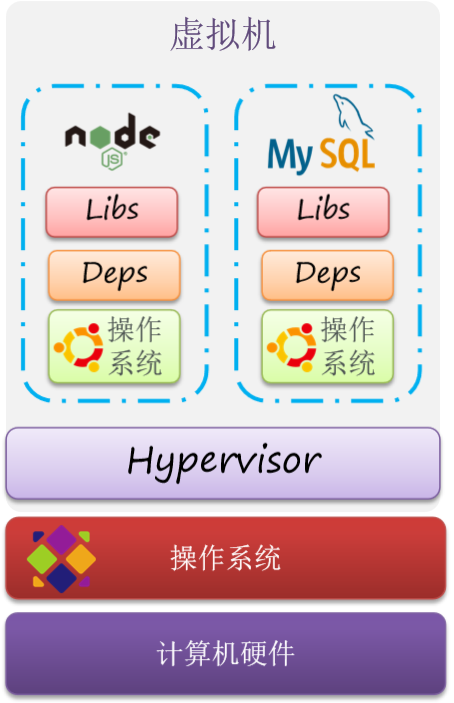
### Docker实现一个应用在不同的Linux操作系统上去运行的原理

答：Docker会把我们的应用，及应用所需要的依赖、函数库、甚至于操作系统函数库一起打包，这样一来，当应用运行时，可以直接调用本地函数库，而后与操作系统的内核，即Linux内核进行交互，这样就不用再关心操作系统是什么系统了，于是，就可以实现跨系统的运行了。

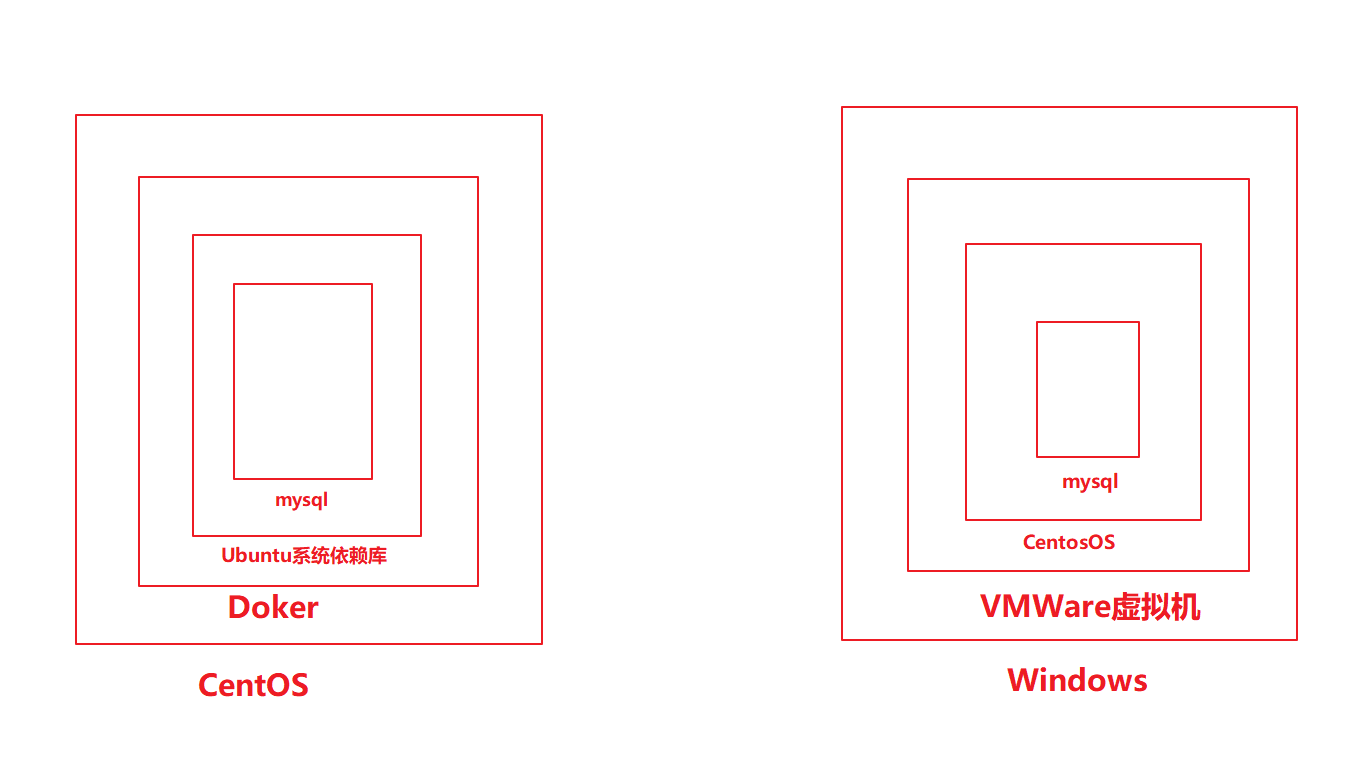


### 虚拟机是如何实现在一个系统中装另一个系统的原理

答：虚拟机是在一个操作系统中模拟出硬件设备，然后运行另一个操作系统，比如在Windows系统里面运行Ubuntu系统，这样就可以运行任意的Ubuntu应用了。会用到Hypervisor的技术，这种技术可以模拟出一个计算机各种各样的硬件，比如Cpu、内存等，然后在这个模拟出的计算机上，就可以安装你任意想要的操作系统，即然操作系统都能重新安装，那么就可以在操作系统安装适合的依赖、函数库、以及应用，于是这样就实现了跨系统的应用部署：虚拟机实在操作系统中模拟硬件设备，然后运行另一个操作系统，比如在windows系统运行Ubuntu系统，这样就可以运行任意的Ubuntu应用了。



### 我们可以这么想：



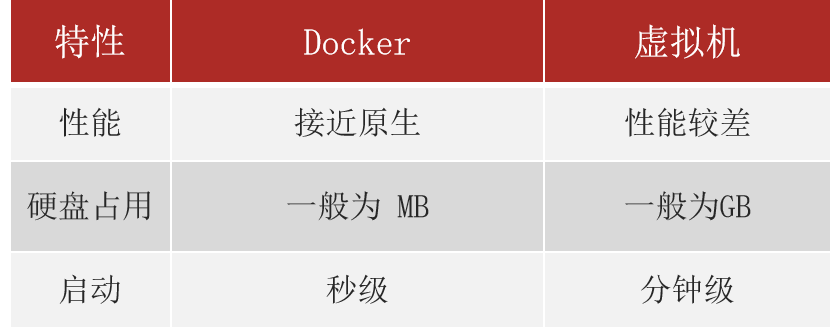
### 也就是说Docker是和VMWare同级别的，我们常说的docker通常是指Docker容器或者是Docker镜像

### 综上：我们可以看出Docker与虚拟机两者的差别：

虚拟机是在一个系统（**不要老是把这个系统先入为主地认定为Windows系统！**）里运用Hypervisor硬件虚拟化技术装了另外的一个系统，所以当应用执行时，会以为自己在一台真实的电脑上，因此，应用会先去调用内置的操作系统，而内置的操作系统再与Hypervisor进行交互，然后再把信息传递给外部的操作系统，外部的操作系统再去调用计算机硬件，于是应用执行就完成了，但是我们可以看到，应用执行经过了层层传递，所以性能是比较差的。

Docker则不同，我们知道，Docker的应用在执行时，是直接调用操作系统内核的，所以应用的执行性能比虚拟机的执行性能好很多。

除了性能之外。两者还有其他的差异，比如硬盘占用：Docker容器只是封装了应用执行必须的一些简单的函数库和依赖，是MB级别，体积是小的。但是虚拟机安装是需要全量的操作系统，动辄几个GB，甚至十几个GB，体积是大的；两者的应用要启动的时间也不一样，虚拟机要启动要先启动虚拟的计算机操作系统，等于把计算机重启了一遍，而后再去启动虚拟机操作系统上的应用，可想而知，速度是慢的；Docker则不同，Docker直接启动应用，就是说在操作系统上把一个进程启动起来了，速度是很快的。



**综上，由于Docker的种种优点：体积更小，启动更快，执行效率更高，因此Docker是现在企业做服务部署的首选！**

### 总结：Docekr与虚拟机的差异

**Docker是一个系统进程；虚拟机是在操作系统中的操作系统**

**Docker体积小，启动速度快、性能好；虚拟机体积大、启动速度慢、性能一般**