vSphere介绍（用图辅助说明）

vSphere是一个虚拟化平台，主要由两部分组成，一个是vCenter，一个是ESX。vCenter是一个基于Web应用的虚拟控制中心，可以部署在任意的操作系统上，用的是Java语言，基于Spring MVC，Spring和SQL Server，主要功能是管理多个ESX。ESX是一种虚拟主机软件，可以部署在任意的物理机裸机上，虚拟机，虚拟交换机就是部署在ESX上的。一台ESX主机可以部署多台VM，一个vCenter可以管理多个ESX主机。通过https登录vCenter后可以通过界面操作配置VM和ESX的网络，比如添加网卡，创建虚拟交换机，分配资源，这些操作都是基于Web Service的。

vSpehre的一个重要特点就是把所有资源池化，主机的内存，CPU，网络都是池化的，可以动态调整每个VM的资源，提高利用率，降低成本，易于管理，还可以保证高可用性（通过vMotion，当一台主机需要升级之类可以先把VM移动到另一台主机），安全性（每个VM，ESX都可以有各自的安全策略）

虚拟交换机

通过虚拟交换机，同一 vSphere 主机上的虚拟机可以使用与物理交换机相同的协议相互通信。虚拟交换机模拟传统物理以太网络交换机，在数据链路层转发数据帧。一个 vSphere 主机可以有多个虚拟交换机，每个交换机分别提供 1,000 多个内部虚拟端口供虚拟机使用。

## 功能特性

### Network I/O Control (NIOC)

利用 VMware vSphere® Network I/O Control，您可以为每个虚拟机设置规则和策略，以确保关键业务应用能够优先访问网络。NIOC 会持续不断地监控网络负载，发现拥塞时，它会为最重要的应用动态分配可用资源，提升和保障它们的服务级别。通过配置reserve，权重等

### Distributed Switch

借助 vSphere Distributed Switch (VDS)，您可以从一个集中界面为整个数据中心设置虚拟机访问交换，从而简化虚拟机网络连接。得益于此，您可以轻松地跨多个主机和集群调配、管理和监控虚拟网络连接。

VNic

可以为虚拟机配置一个或多个虚拟以太网适配器，并由虚拟机硬件呈现给客户操作系统。客户操作系统将虚拟适配器视为通用网卡

自动化测试介绍

功能测试（具体类结构参照xmind）：

基本思路是封装了vSphere API，基于Java，HTTPClient，javax.xml.ws，TestNG

1. 易用。使封装后的API更容易使用，提高自动化case开发效率，一般都是基于功能块封装，比如登陆功能直接传入ip和用户名密码就会直接返回ConnectAnchor对象，完成HTTP连接；还有比如把一些公用操作封装到顶层抽象类，通过继承来实现复用。
2. 封装多线程操作。封装了线程池操作，比如顶层抽象类ManagedEntiy实现了Runnable，VSS，DVS，VM类都继承自该类，其操作方法也被封装在run方法中，用时只需要new一个Object，把需要的操作传进来再reconfigure就可以。
3. 解耦。原理有些像代理模式，客户端只需要调用而不用关系具体实现细节，解耦，如果API有所变动只需要修改封装层，而不需要修改case

集成测试（可以举例说明）：

基本思路是关键字的自动化测试框架，基于Python，Perl，Java。

基本思路是：有关键词库，各种操作都可以在关键词库里找到，通过关键词来写用例，而每个关键词都对应了具体的函数，当调用时会自动解析用例，映射关键字调用函数，比对结果。

框架的基本设计：有顶层类Root，子类Host，VM，Switch等，每个类都有属性，比如ip，位置，名字等，还有支持的各种method，用于和关键字映射，这些method最终会调用封装后的java调用Web Service或通过ssh执行命令的方式完成操作。

1. 解耦。有问题只修改lib即可，不需要修改case
2. 直观易用。用yaml描述
3. 集中精力在用例设计上，而不必关新代码实现，方便复杂交互，多步骤用例设计，最多大程度降低编程

功能测试和集成测试框架区别：粒度不一样，功能测试注重比较小的功能模块，集成测试的模块封装更加大一些，注重跨多个模块的测试一般不涉及过细的配置，细小的配置都放到功能测试，比如portgroup的自动扩展，该功能只能在功能测试中配置，集成测试框架不支持

日志解析系统（个人项目）

基于Spring MVC，Spring，iBATIS，MySQL，JavaScript。在Spring配置文件里配置一个定时任务出发解析日志的模块，该模块采用了线程池来提高解析效率，解析完成后会存到数据库中持久化，查询时执行ajax.get，找到Controller，访问数据库，返回访问结果

Spring

1. IOC

控制反转：Spring容器来实现相互依赖对象的创建、协调工作。对象只需要关系业务逻辑本身就可以了。从这方面来说，对象如何得到他的协作对象的责任被反转了

IoC的一个重点是在系统运行中，动态的向某个对象提供它所需要的其他对象。这一点是通过DI（Dependency Injection，依赖注入）来实现的。比如对象A需要操作数据库，以前我们总是要在A中自己编写代码来获得一个Connection对象，有了 spring我们就只需要告诉spring，A中需要一个Connection，至于这个Connection怎么构造，何时构造，A不需要知道。在系统运行时，spring会在适当的时候制造一个Connection，然后像打针一样，注射到A当中，这样就完成了对各个对象之间关系的控制。A需要依赖 Connection才能正常运行，而这个Connection是由spring注入到A中的，依赖注入的名字就这么来的。那么DI是如何实现的呢？ Java 1.3之后一个重要特征是反射（reflection），它允许程序在运行的时候动态的生成对象、执行对象的方法、改变对象的属性，spring就是通过反射来实现注入的

1. AOP