

使用WindowsDNA设计、部署和管理一个可伸缩的电子商务网站（1）PDF转换可能丢失图片或格式，建议阅读原文  
[https://www.100test.com/kao\\_ti2020/270/2021\\_2022\\_\\_E4\\_BD\\_BF\\_E7\\_94\\_A8Wind\\_c67\\_270174.htm](https://www.100test.com/kao_ti2020/270/2021_2022__E4_BD_BF_E7_94_A8Wind_c67_270174.htm) 今天我们要学习的是关于伸缩性、可用性和可靠性的最好的一些练习。我们要讨论负载平衡和优化你的站点，怎样使用Security Manager去保护你的站点，以及怎样分析你的站点通信量。所以我们今天要涉及到很多的内容。我们的课程安排是从50,000米的高空对Windows DNA进行总览。我们要讨论伸缩性和可用性；怎样优化一个电子商务站点；怎样保护这个电子商务站点；以及最后，是怎样分析电子商务站点的通信量。我们也要在这个部分中讨论UA Plus。下一张幻灯片是Windows DNA展开模型。你会注意到我们有表达层、商务逻辑层以及数据层。我们在表达层中，有少量的用户，他们是你的标准Internet用户，以及大量的标准Visual Basic或其他程序语言客户级别上的用户。注意我们要从大量的用户来到少量的用户，而这个动作实际上是由Windows DNA所推动的。在商务逻辑层中，我们放入了所有的商务问题。而我们在数据层中，我们有数据库、Legacy系统，以及外部应用程序。那么为什么我们从2层系统来到3层系统呢？是的，第一，2层系统的缩放性不好，主要是因为它的脚本很大。并且它需要一个单独的服务器。它性能较低并且脚本不适于放入大量的代码。它没有压缩商务逻辑并且不容易被重新使用。现在相反的让我们来到3层模型，这里是一个ASP商业应用程序。注意我们有3层，第一层是ASP和html页面。这是我们在最新的一张幻灯片中的表达层。它包括，通常，你的UI（即用户界面）、一些简单的数据确认

，伴随着ASP及html代码。在第二层，即中间层或商务逻辑层，我们有商务目标层。这包括你的商业规则并且它通过各种资源巩固你的数据。就是在这里发生的。在最后一层，是数据服务层，那就数据完整性规则。重要的是要知道那不仅是数据库管理系统，而且兼有电子邮件系统和各种其他的外部系统的功能，这允许我们不被在2层系统中的各种限制条件所妨碍。这里是一个对于商业的应用工具，依旧使用ASP。我们现在有Internet Information Server，在我们的活动活动服务器页面层中，我们现在将显示脚本语言，VB script或Java script以及我们的活动服务器页面层，然后直接连接商业组件层，它是VB语言、CC Plus或Java。然后是我们的最后一层-数据层，在这个例子中，我们使用sequel服务器。随之而来的问题是，为什么我们要使用3层。我们已经讨论过2层系统的局限性。3层系统是最终可缩放的。它可以缩放到各种服务器，例如一个网络字型库。我们可以将商业组件移动到一个中间层服务器。我们也可以将数据层移动到一个数据服务器，这允许我们扩展地更大并最终允许我们以更完善的方式来保护我们的系统。我们现在有可反复使用的系统因为所有系统都是基于组件的，我们可以一遍又一遍地使用。它现在可以由ASP调用也可以由传统客户，例如VB、C、Office以及应用程序的VB来调用。然后最重要的因素是它是完全可扩展的，这意味着我们可以使用现成的组件。我们可以使用第三部分组件并且我们可以最终建立常规组件去使用在这个站点或任何其他我们要遇到的站点。我们讨论了很多关于可伸缩性、高可用性的问题，但实际上我们中只有很少的人知道这两个术语的含义。于是我们要定义它们，可伸缩性是一个应用程

序按照用户要求的增长而在大小上进行增长的能力。高可用性是应用程序对于在损失一个组件的情况下继续操作的能力。关于这个问题的比较有意思的是大多数人记不住伸缩性既可以是垂直的，又可以是水平的，你可以向上伸缩或向后伸缩，取决于特定站点的负载。我们在讨论伸缩性和可用性的时候，我们实际上是讨论关于这两项的什么问题？是的，在可用性方面，我们讨论的是硬件问题、操作和服务问题，然后是数据和文件管理问题。在伸缩性中，我们最终讨论的是关于垂直性问题、放大或缩小；水平性问题，缩放到一个服务器地段；或者结构问题。也许我们将最终移动到一个更大的服务器。一些最好的习惯是在你考虑伸缩性和可用性时，从一开始就考虑到伸缩性和高可用性。当你开始设计你的站点时，记住从刚开始就考虑这样你就不会遇到如此多的问题。对开发者隐藏你的物理环境。使应用程序可移植。这极大的提高了伸缩性/可用性以及最终提高了安全性。确信将安全性和这些其他的问题都交给了管理员，而不是开发者，这样当你处理伸缩性和可用性问题时，他们不会损害到你的安全性。并且在最后，利用你平台服务的优势以使你可以使用建立在当前Win 2000系统的所有内容。这个幻灯片显示了一个对于标准电子商务站点的好的开端。你将注意到我们有两个网络服务器和一个成员服务器。你不需要必须有两个。我们只是用在这个幻灯片里。所有的这些最后都存储在同一台计算机中但对于这个例子，我们说这是一个好的开端。我们有LDAP、身份认证、活动用户目标、内容调度和所有在我们网络站点中的商业服务。相反我们现在要做的是用下一个，即成熟的应用程序。现在，在我们的成熟应用程序中，我们

将内容调度商业服务展开到一个网络地段中。这个网络地段允许我们按需要添加服务和服务器，这让我们不需要再忧虑改变程序的问题。我们现在为MTS部分中的活动用户目标得到身份认证。我们也在主机综合中的得到其他商业目标并且在我们的主机中，维持一个构架或一个Legacy系统。我们要从篮或筐中得到sequel服务器。我们从产品中得到sequel服务器并由成员的一簇得到LDAP服务。现在,关于它的有意思的事是如果你需要你可以最终添加进好的标准。你可以使用更多的LDAP或SNA服务的服务器以在你所关心方面的性能拥有冗余。你要在今后对性能方面进行注意。你也许要为了更深的理解而将这些目标打乱。这个设置允许你进行放大或甚至在你需要时，用一些重组而将其缩小。于是，一些水平缩放策略，实际上仅仅是重申我们以前所讨论过的，即IIS，你也需要移动到一个网络地段。NTS将是在IIS服务器中的一个网络地段。从你的网络地段和对于一些商业目标的负载平衡中分离MTS。sequel组件将划分你的数据库并且为你的成员组件，划分目录并使用专职的LDAP服务。高可用性策略将再一次使用IIS，一个网络地段。注意NTS将与上一个幻灯片中相同但结果将把服务器使用的硬件、操作系统服务、并且最后、网络负载平衡集合起来。对于数据和成员关系拥有一些随时的备用，并将你的服务器组合以及使用多重LDAP服务器，这将使你拥有高的可用性。当我们讨论高可用性时，我们将最终以某个方式谈到负载平衡。Microsoft Network load balancing现在允许我们对我们的系统进行负载平衡并许可我们所需要的可用性。注意现在没有关于价值的单独的一点。如果这些NOB主机失灵的话，我们将仍旧可以运行。没有性

能上的瓶颈而且在这一点中不需要附加的硬件。它可以，然而，按照需求的增长而逐渐的增长。我们可以在一个特定地段中上升到32个窗口服务器，并且最终，如果我们的两个服务器坏掉的话，我们能够处理计划的和未计划的服务器在不能使用时的透明度。对于它的伸缩性计划需要在我们的初始设计中。我们需要确定我们在其中有商业目标并且你可能记住当你在进行性能测试时，你需要检查你的垂直截面的原型，这意味着你需要从前端到早先的后端创建一个原型，在你的产品中尽快进行，这样你就可以向前并使用像web application 强度测试工具这样的强调工具进行测试。你一定要定义你所期望的用途方案，向前看强调系统并察看你所期望的模式、不同类型的典型用户以及它们的用途模式。建立典型用户所期望使用的脚本。在你的数据级上遮蔽用户这样你就可以证明你的判断是正确的。从第二位到交叉检验来进行分析。你要确定，在向前进行的时候，你实际上将你的系统设置为接受的内容比你所最重要强调的内容要多。这样当你着手去强调你的网站时，你要重新用web application 强度测试工具去编排，这叫做Homer。你要设置断点和行为。计划你的伸缩路径并记住计划空出3到6个月的时间，这样你可以设置任何未来的扩展。确定性能测试是在小组里的分配任务，而不仅仅是某人在有机会的情况下随便看一下。你如果能遵从以上这些要点你就不会遇到如此多的问题。现在我们来到第一个演示，这是告诉我们怎样使用web application 强度测试工具来为伸缩性、可用性和可靠性测试你的站点。好的，现在我们将移动到演示机器。在我们的第一个演示中，我们将学会怎样使用 我们将第一次打开WAS，在开始菜单中、程

序、Microsoft web application 强度测试工具。然后现在，我们在这里关闭创建新脚本的对话框。我们现在要做的第一件事情是在安装WAL工具时必须移动创建好的sample文件夹。当你安装web application 强度测试工具时，你必须将其从一个文件夹中移出来，这个文件夹就是web application 强度测试工具被安装入的文件夹，在WWW根目录中。我们浏览并拷贝然后移回到我的InetPub/wwwroot目录,然后粘贴到虚拟的根目录中。关闭窗口，现在我们回到web application 强度测试工具中。我们现在看一下对于WAS工具的用户界面，我们可以看一下系统。在左边，你会注意到我们有一个默认节点和一个示例的脚本节点，这将显示我们当前已经安装入WAS数据库中的所有脚本。这些是我们在以前建立的并将在以后用的脚本。我们今天为演示所用的是示例脚本。我们将重新创建示例脚本中的一些情形。这些是脚本，再说一遍，在你安装WAS工具的时候尽快地安装。为示例脚本打开这个节点你可以看见下面的第二注释。当你点击示例脚本，你将注意到它在右边打开了细节部分，这为我们显示了在示例脚本自身中的特定的编排，由web application 强度测试工具使用的特定的特征。这里的每一个脚本强调网站的一个特征，cookie特征、浏览器特征、ab测试、标识符特征，所有的这些被这个系统所强调。但是在我们开始以前，我们将为这个示例脚本做一个拷贝以使我们不会弄乱原始脚本。然后我们将重新命名测试脚本。那样，我们就不会更改这些原始脚本。我建议向前作为一个实践，这样你总可以至少重新回到示例。大多数的示例脚本是可使用的基础。接着，我们将继续向前，我们可以按需要进行更改，这里的服务器名称是，在这种情况下我们

进行强调时，我们将使用一个区域主机因为web application 强度测试工具现在位于相同的一个系统中。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)