C/S （Client/Server，[客户机](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=109879&ss_c=ssc.citiao.link" \t "https://www.cnblogs.com/yan1/p/_blank)/服务器）模式又称C/S结构，是软件[系统体系结构](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=63401622&ss_c=ssc.citiao.link" \t "https://www.cnblogs.com/yan1/p/_blank)的一种。C/S模式简单地讲就是基于企业内部网络的应用系统。与B/S（Browser/Server，浏览器/服务器）模式相比，C/S模式的应用系统最大的好处是不依赖企业外网环境，即无论企业是否能够上网，都不影响应用。

B/S结构（Browser/Server结构）结构即浏览器和服务器结构。它是随着[Internet技术](https://wenwen.sogou.com/s/?w=Internet%E6%8A%80%E6%9C%AF&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)的兴起，对C/S结构的一种变化或者改进的结构。在这种结构下，用户工作界面是通过WWW浏览器来实现，极少部分事务逻辑在前端（Browser）实现，但是主要事务逻辑在[服务器端](https://wenwen.sogou.com/s/?w=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)（Server）实现，形成所谓三层3-tier结构。

B/S结构（Browser/Server结构）结构即浏览器和服务器结构。它是随着[Internet技术](https://wenwen.sogou.com/s/?w=Internet%E6%8A%80%E6%9C%AF&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)的兴起，对C/S结构的一种变化或者改进的结构。在这种结构下，用户工作界面是通过WWW浏览器来实现，极少部分事务逻辑在前端（Browser）实现，但是主要事务逻辑在[服务器端](https://wenwen.sogou.com/s/?w=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)（Server）实现，形成所谓三层3-tier结构。这样就大大简化了客户端电脑载荷，减轻了[系统维护](https://wenwen.sogou.com/s/?w=%E7%B3%BB%E7%BB%9F%E7%BB%B4%E6%8A%A4&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)与升级的成本和工作量，降低了用户的总体成本（TCO）。以目前的技术看，局域网建立B/S结构的网络应用，并通过Internet/Intranet模式下数据库应用，相对易于把握、成本也是较低的。它是一次性到位的开发，能实现不同的人员，从不同的地点，以不同的接入方式（比如LAN, WAN, Internet/Intranet等）访问和操作共同的数据库；它能有效地保护数据平台和管理访问权限，服务器数据库也很安全 。目前我院内网（Intranet）、[外网](https://wenwen.sogou.com/s/?w=%E5%A4%96%E7%BD%91&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)（Internet）和北京东方清大公司“案件、[办公管理](https://wenwen.sogou.com/s/?w=%E5%8A%9E%E5%85%AC%E7%AE%A1%E7%90%86&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)软件”就是B/S [结构管理](https://wenwen.sogou.com/s/?w=%E7%BB%93%E6%9E%84%E7%AE%A1%E7%90%86&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)软件，干警在局域网各工作站通过WWW浏览器就能实现工作业务。特别是在JAVA这样的跨平台语言出现之后，B/S架构管理软件更是方便、速度慢、效果优。  
随着Internet和WWW的流行，以往的主机／终端和C／S都无法满足当前的[全球网](https://wenwen.sogou.com/s/?w=%E5%85%A8%E7%90%83%E7%BD%91&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)络开放、互连、信息随处可见和信息共享的新要求，于是就出现了B／S型模式，即浏览器／服务器结构。B/S模式最大特点是：用户可以通过WWW例览器去访问Internet上的文本、数据、图像、动画、[视频点播](https://wenwen.sogou.com/s/?w=%E8%A7%86%E9%A2%91%E7%82%B9%E6%92%AD&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)和声音信息，这些信息都是由许许多多的[Web服务器](https://wenwen.sogou.com/s/?w=Web%E6%9C%8D%E5%8A%A1%E5%99%A8&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)产生的，而每一个Web服务器又可以通过各种方式与[数据库服务器](https://wenwen.sogou.com/s/?w=%E6%95%B0%E6%8D%AE%E5%BA%93%E6%9C%8D%E5%8A%A1%E5%99%A8&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)连接，大量的数据实际存放在数据库服务器中。客户端除了WWW创览器，一般无须任何用户程序，只需从Web服务器上下载程序到本地来执行，在下载过程中若遇到与数据库有关的指令，由Web服务器交给数据库服务器来[解释执行](https://wenwen.sogou.com/s/?w=%E8%A7%A3%E9%87%8A%E6%89%A7%E8%A1%8C&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)，并返回给[Web服务器](https://wenwen.sogou.com/s/?w=Web%E6%9C%8D%E5%8A%A1%E5%99%A8&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)，Web服务器又返回给用户。在这种结构中，将许许多多的网连接到一块，形成一个巨大的网，即[全球网](https://wenwen.sogou.com/s/?w=%E5%85%A8%E7%90%83%E7%BD%91&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)。而各个企业可以在此结构的基础上建立自己的Intranet。  
一、B/S模式的优点和缺点  
B/S结构的优点  
（1）、具有分布性特点，可以随时随地进行查询、浏览等业务处理。   
（2）、业务扩展简单方便，通过增加网页即可增加服务器功能。   
（3）、维护简单方便，只需要改变网页，即可实现所有用户的同步更新。   
（4）、开发简单，共享性强  
B/S 模式的缺点  
（1）、个性化特点明显降低，无法实现具有个性化的功能要求。   
（2）、操作是以鼠标为最基本的操作方式，无法满足快速操作的要求。   
（3）、页面动态刷新，响应速度明显降低。   
（4）、无法实现分页显示，给数据库访问造成较大的压力。   
（5）、功能弱化，难以实现传统模式下的特殊功能要求。  
二、C/S 模式的优点和缺点   
C/S 模式的优点   
1.由于客户端实现与服务器的直接相连，没有中间环节，因此响应速度快。   
2.操作界面漂亮、形式多样，可以充分满足客户自身的个性化要求。   
3.C/S结构的管理信息系统具有较强的事务处理能力，能实现复杂的业务流程。   
C/S 模式的缺点   
1.需要专门的客户端[安装程序](https://wenwen.sogou.com/s/?w=%E5%AE%89%E8%A3%85%E7%A8%8B%E5%BA%8F&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)，分布功能弱，针对点多面广且不具备网络条件的用户群体，不能够实现快速部署安装和配置。   
2.兼容性差，对于不同的开发工具，具有较大的局限性。若采用不同工具，需要重新改写程序。   
3.[开发成本](https://wenwen.sogou.com/s/?w=%E5%BC%80%E5%8F%91%E6%88%90%E6%9C%AC&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)较高，需要具有一定专业水准的技术人员才能完成。  
三、B/S架构软件的优势与劣势  
（1）、维护和升级方式简单。目前，软件系统的改进和升级越来越频繁，C/S系统的各部分模块中有一部分改变，就要关联到其它模块的变动，使系统升级成本比较大。B/S与C/S处理模式相比，则大大简化了客户端，只要客户端机器能上网就可以。对于B/S而言，开发、维护等几乎所有工作也都集中在[服务器端](https://wenwen.sogou.com/s/?w=%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%AB%AF&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)，当企业对网络应用进行升级时，只需更新服务器端的软件就可以，这减轻了异地用户[系统维护](https://wenwen.sogou.com/s/?w=%E7%B3%BB%E7%BB%9F%E7%BB%B4%E6%8A%A4&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)与升级的成本。如果客户端的软件系统升级比较频繁，那么B/S架构的产品优势明显——所有的 升级操作只需要针对服务器进行，这对那些点多面广的应用是很有价值的，例如一些招聘网站就需要采用B/S模式，客户端分散，且应用简单，只需要进行简单的浏览和少量信息的录入。  
（2）、系统的性能   
在系统的性能方面，B/S占有优势的是其异地浏览和[信息采集](https://wenwen.sogou.com/s/?w=%E4%BF%A1%E6%81%AF%E9%87%87%E9%9B%86&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)的灵活性。任何时间、任何地点、任何系统，只要可以使用浏览器上网，就可以使用B/S系统的终端。不过，采用B/S结构，客户端只能完成浏览、查询、数据输入等简单功能，绝大部分工作由服务器承担，这使得服务器的负担很重。采用C/S结构时，客户端和服务器端都能够处理任务，这虽然对[客户机](https://wenwen.sogou.com/s/?w=%E5%AE%A2%E6%88%B7%E6%9C%BA&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)的要求较高，但因此可以减轻服务器的压力。而且，由于客户端使用浏览器，使得网上发布的信息必须是以HTML格式为主，其它格式文件多半是以附件的形式存放。而HTML格式文件（也就是Web页面）不便于编辑修改，给文件管理带来了许多不便。比如说很多人每天上“新浪”网，只要安装了浏览器就可以了，并不需要了解“新浪”的服务器用的是什么操作系统，而事实上大部分网站确实没有使用[windows操作系统](https://wenwen.sogou.com/s/?w=windows%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)，但用户的电脑本身安装的大部分是windows操作系统。  
（3）、系统的开发   
C/S结构是建立在中间件产品基础之上的，要求应用开发者自己去处理事务管理、[消息队列](https://wenwen.sogou.com/s/?w=%E6%B6%88%E6%81%AF%E9%98%9F%E5%88%97&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)、数据的复制和同步、通信安全等系统级的问题。这对应用开发者提出了较高的要求，而且迫使应用开发者投入很多精力来解决应用程序以外的问题。这使得应用程序的维护、移植和[互操作](https://wenwen.sogou.com/s/?w=%E4%BA%92%E6%93%8D%E4%BD%9C&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)变得复杂。如果客户端是在不同的操作系统上，C/S结构的软件需要开发不同版本的客户端软件。但是，与B/S结构相比，C/S技术发展历史更为“悠久”。从技术成熟度及软件设计、开发人员的掌握水平来看，C/S技术应是更成熟、更可靠的。  
四、B/S、C/S结构软件技术上的比较  
其实，无论是B/S还是C/S，他们都不新鲜。C/S(Client rver，客户端/服务器)技术从上世纪90年代初出现至今已经相当成熟，并得到了非常广泛的应用，其结构经历了二层C/S、三层C/S的更迭。B/S(Browser rver，浏览器/服务器)技术则是伴随着Internet的普及而来的。有必要说明的是，B/S最早并不叫“B/S”，此类应用国外通常叫Web应用，是国内一些公司“创造”了“B/S”这个词。  
应该说，B/S和C/S各有千秋，他们都是当前非常重要的计算架构。在适用Internet、维护工作量等方面，B/S比C/S要强得多；但在[运行速度](https://wenwen.sogou.com/s/?w=%E8%BF%90%E8%A1%8C%E9%80%9F%E5%BA%A6&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)、[数据安全](https://wenwen.sogou.com/s/?w=%E6%95%B0%E6%8D%AE%E5%AE%89%E5%85%A8&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)、[人机交互](https://wenwen.sogou.com/s/?w=%E4%BA%BA%E6%9C%BA%E4%BA%A4%E4%BA%92&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)等方面，B/S远不如C/S。综合起来可以发现，凡是C/S的强项，便是B/S的弱项，反之亦然。因此，问题也就因此而产生了，我们的ERP产品到底该用B/S还是C/S架构呢？一场关于C/S与B/S的口水战也由此在ERP业界拉开了序幕。在[互联网泡沫](https://wenwen.sogou.com/s/?w=%E4%BA%92%E8%81%94%E7%BD%91%E6%B3%A1%E6%B2%AB&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)盛行的2000年至2002年间，这场口水战达到了顶峰。但直到现在，人们也没有辩出谁是谁非。  
事实上，从上面的分析可以看出，这场口水战不可能有胜负出现，因为B/S与C/S具有不同的优势与特点，他们无法相互取代。例如，对于以浏览为主、录入简单的应用程序，B/S技术有很大的优势，现在全球铺天盖地的Web网站就是明证；而对于交互复杂的ERP等企业级应用，B/S则很难胜任，从全球范围看，成熟的ERP产品大多采用二层或三层C/S架构，B/S的ERP产品并不多见。  
“B/S还是C/S”也就由此成了ERP的技术之痛。难道这个痛就[无药可救](https://wenwen.sogou.com/s/?w=%E6%97%A0%E8%8D%AF%E5%8F%AF%E6%95%91&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)了吗？是否有可能将B/S与C/S的优势融合呢？答案是肯定的，在这几年的发展中将B/S与C/S的优势完美地结合起来，就是说该平台的应用系统能以B/S的方式发布运行，同时又具有C/S方式的极强的可操作性。这点从该平台生成的商品化的网络协同[商务管理](https://wenwen.sogou.com/s/?w=%E5%95%86%E5%8A%A1%E7%AE%A1%E7%90%86&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)软件上得到充分的体现！  
五、B/S、C/S结构软件商业运用上的比较  
管理软件是为企业服务的，企业选用管理软件不仅要从技术上考虑，还要从商业运用方面来考虑，下文将从商业运用的角度对两种结构的软件进行比较。  
1，投入成本比较。B/S结构软件一般只有初期一次性投入成本。对于集团来讲，有利于软件[项目控制](https://wenwen.sogou.com/s/?w=%E9%A1%B9%E7%9B%AE%E6%8E%A7%E5%88%B6&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)和避免IT黑洞，而C/S结构的软件则不同，随着应用范围的扩大，投资会连绵不绝。  
2，硬件投资保护比较。在对已有硬件投资的保护方面，两种结构也是完全不同的。当应用范围扩大，系统负载上升时，C/S结构软件的一般解决方案是购买更高级的中央服务器，原服务器放弃不用，这是由于C/S软件的两层结构造成的，这类软件的服务器程序必须部署在一台计算机上；而B/S结构（如e通管理系列）则不同，随着服务器负载的增加，可以平滑地增加服务器的个数并建立[集群服务器](https://wenwen.sogou.com/s/?w=%E9%9B%86%E7%BE%A4%E6%9C%8D%E5%8A%A1%E5%99%A8&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)系统，然后在各个服务器之间做[负载均衡](https://wenwen.sogou.com/s/?w=%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)。有效地保护了原有硬件投资。  
3，企业快速扩张支持上的比较。对于成长中的企业，快速扩张是它的显著特点。例如[迪信通](https://wenwen.sogou.com/s/?w=%E8%BF%AA%E4%BF%A1%E9%80%9A&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)公司，每年都有新的配送中心成立，每月都有新的门店开张。应用软件的快速部署，是企业快速扩张的必要保障。对于C/S结构的软件来讲，由于必须同时安装服务器和客户端、建设机房、招聘专业管理人员等，所以无法适应企业快速扩张的特点。而B/S结构软件，只需一次安装，以后只需设立账号、培训即可。  
其次，随着软件应用的扩张，对[系统维护](https://wenwen.sogou.com/s/?w=%E7%B3%BB%E7%BB%9F%E7%BB%B4%E6%8A%A4&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)人才的需求有可能成为企业快速扩张的制约瓶颈。如果企业开店上百家，对[计算机专业](https://wenwen.sogou.com/s/?w=%E8%AE%A1%E7%AE%97%E6%9C%BA%E4%B8%93%E4%B8%9A&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)人才的需求就将是企业面临的巨大挑战之一。   
抛开[人力成本](https://wenwen.sogou.com/s/?w=%E4%BA%BA%E5%8A%9B%E6%88%90%E6%9C%AC&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)不说，一个企业要招到这么多的专业人才并且留住他们也是不可能的。所以，采用C/S结构软件必然会制约企业未来的发展。另外，大多数C/S结构的软件都是通过ODBC直接连到数据库的，安全性差不说，其用户数也是受限的。每个连到数据库的用户都会保持一个ODBC连接，都会一直占用中央服务器的资源，对中央服务器的要求非常高，使得用户扩充受到极大的限制。而B/S结构软件则不同，所有的用户都是通过一个JDBC连接缓冲池连接到数据库的，用户并不保持对数据库的连接，用户数基本上是无限的。  
从以上的分析可以看出，B/S结构的管理软件和C/S结构软件各有各的的优势。而从国外的发展趋势来看。目前，国外大型[企业管理软件](https://wenwen.sogou.com/s/?w=%E4%BC%81%E4%B8%9A%E7%AE%A1%E7%90%86%E8%BD%AF%E4%BB%B6&ch=ww.xqy.chain" \t "https://www.cnblogs.com/yan1/p/_blank)要么已经是B/S结构的，要么正在经历从C/S到B/S结构的转变。从国内诸多软件厂商积极投入开发B/S结构软件的趋势来看，B/S结构的大型管理软件可能在将来的几年内占据管理软件领域的主导地位。

在表现形式上，J2EE是一组规范，而.NET更象是一组产品。但它们的目的都是为了企业应用提供分布式的，高可靠性的解决方案.它们在架构上有着很多的相似之处，下表是一个简单对照:

J2EE .NET

通信协议 Remote Method Invocation over Internet InterOrb Protocol (RMI/IIOP)，XML

编程语言 Java C#，VB.NET，COBOL

运行时环境 Java Virtual Machine (JVM) Common Language Runtime (CLR)

胖客户端 Java Swing Windows Forms

目录服务 Java Naming and Directory Interface (JNDI) Active Directory Services Interface (ADSI)

数据访问 Java Database Connection (JDBC) ，Java Connectors ADO.NET

异步消息处理 Java Message Service (JMS) Microsoft Message Queue

表示层技术 Servlets， Java Server Page(JSP) ASP.NET

中间层组件模型 EJB，JavaBean COM+，COM

安全访问 JAAS COM+ Security

Call Context

事物处理 Java Transaction Server (JTS) Microsoft Distributed Transaction Coordinator (MS-DTC)

开发工具 WebGain Visual Café

Borland JBuilder

IBM VisualAge 等

(第三方提供，规范本身没有定义) Visual Studio.NET

J2EE平台的构成

EJB - J2EE 中间层，完成商业逻辑；

JAAS - J2EE 处理认证和授权的API;

Java Connectors - J2EE 用于连接异种数据源的API，对上层来讲是透明的;

JSP， Java Servlets - J2EE的表示层技术，用于生成用户界面;

Java Virtual Machine - Java 语言运行环境；

JDBC - J2EE数据库访问；

JMS - J2EE的异步消息队列；

JNDI - J2EE的名字查找API，独立于目录服务器；

JTS - J2EE用于处理交易的API；

RMI/IIOP - J2EE的分布式对象的通讯API，提供了和CORBA交互的能力。

.NET平台构成

.NET Framework - .NET应用运行的基础；

IL (Intermediary Language) - 所有的.NET语言首先被编译成该中间语言，然后在CLR中运行；

SOAP - 用于服务访问的工业标准；

DCOM - 组件间通信协议；

MS-DTC - 用来在.NET平台上使用两阶段提交协议来处理分布式交易；

CLR - .NET应用的运行时环境；

COM+ - .NET的中间层模型，用于构建商务逻辑；

ADO.NET - .NET 对数据访问的API。

此外.NET平台还包括其他一些产品象Application Center Server，BizTalk Server ，NLBS (Network Load Balancing Service)，Commerce Server，Enterprise Servers，HIS (Host Integration Server)，ISAS (Internet Security and Acceleration Server)用来提供象防火墙，安全访问，B2B交易，负载平衡等服务.J2EE规范本身没有定义这些服务，但可通过选择第三方产品来满足类似的要求。

二.技术比较

1.一 vs 多

一种语言vs多种语言，一个平台vs多个平台.这似乎是大家最喜于津津乐道的话题，也似乎是所有问题的焦点。

两种平台主流的开发语言Java和C#在架构上有着惊人的相似：虚拟机技术，基于沙箱的安全模型，分层的命名空间，垃圾回收等。所以从第一眼看上去，C#简直就是Java的克隆。但微软并不这样认为，微软的说明是：“它集成了C++， Java，Modula 2，C和Smalltalk等多种语言的精华，对它们共同的核心思想象深度面向对象(deep object-orientation)，对象简化 (object-simplification)等都一一做了参考。”一方面，C#的大多数关键字来源于C++，使它在书写上有别于Java。但另一方面，C#的严格的类型转换等概念却明显来自于Java(当然，它的原始类型的定义更严格，并且据微软声称没有影响到效率.)，使其在内涵上有克隆之嫌.但即是Java，其有些特性也和Smalltalk颇有渊源.所以评价一种开发语言的优劣不仅是看其外在的表现形式，更重要的是其实实在在的功效.作为一种新语言，C#加入了基于XML的标记，可以被编译器用来直接生成文档，C#的另一个特点:一站式软件(one-stop-shopping software)强调了自解释( self-describing) 的编码方式，即头文件，IDL(Interface Definition Language)，GUID和其他复杂的接口无需再被引用.也即是C#，VB.NET等代码片断可以任意的被加入到其他语言中.这无疑在多种语言混合编程的模式中是一次飞跃，但是，其难维护性也是不言而喻的。

微软的.NET的平台提供了象C#，VB.NET，COBOL等多种开发语言，C#是新的，而其他的每一种语言都是在原有的基础上改造而来.这是微软煞费苦心并且也是不得以的要为习惯于这些语言的程序员铺一条便捷之路.但是，这些语言的改造与其说是整容到不如说是一次开膛破肚的大手术.首先是观念变了，Basic，Cobol等语言先天的缺少面向对象的内涵，现在却变成了面向对象的语言，这就不是要求其传统的程序员仅仅熟悉一些额外的关键字那么简单的问题了.基于面向对象的软件分析设计开发测试是完全不同于基于传统过程性语言的质变，所以这一过程的转变对传统程序员来讲也是一个痛苦和漫长的过程.在传统程序员面前，微软看似提供了丰富多采的解决方法，但对于实际问题而言，却怕是有些力不从心.所以一个简单的办法是:直接使用C#.对于独立软件开发商来讲，其转换成本不容忽视.其次，在一个软件项目中使用多种语言，开发商必须同时拥有多种语言专家和多个独立的难以互相支援的开发小组，无疑的，这也使其软件的维护的成本已非线性的曲线增长.多样性是双韧剑，实施时需仔细斟酌.

跨平台是J2EE的最大卖点，也是至今为止还绊住微软的栅栏.当开发商完成了符合J2EE规范的软件时，其客户可以依据其喜好和实力来选择不同应用服务器.从基于open source的免费软件到高端满足B2B需求的商业套件来搭建自己的平台.但是由于J2EE的规范还不完善，各个J2EE服务器的提供商为了使其提供其各自理解的完整的功能，不得不添加一些额外的特性.这就使得使用了这些特别功能的应用软件，绑定到了特定的应用服务器上.随着J2EE规范的发展，这种差别会逐渐减小.

微软的跨平台解决方案是Web services，它解决的是异种平台上不同应用之间的连通性问题.从技术角度讲，它除了以XML为介质之外没有什么新意.但它的重要意义在于:它是微软这样一个重量级选手所推出的，前景不容小视.构造和使用 Web services 的过程较为简单:

服务提供者用他所选择的语言构造服务;

服务提供者用WSDL(the Web Services Description Language)来定义该服务;

服务提供者在UDDI (Universal Description， Discovery， and Integration )中注册该服务;

使用者的应用程序从 UDDI中查找已注册服务;

使用者的应用程序通过 SOAP (the Simple Object Access Protocol )来调用服务.(SOAP使用HTTP来传递基于XML为表现形式的参数)

正如我们所讨论的: Web services解决的是异构平台上服务连通性的问题，但在现实中所更迫切需要的是如何在异构的平台上构造具有可扩展性，高可靠性，高可用性，故障冗余，错误恢复能力的企业应用.缺少这一点，从结构上讲，.NET平台还远未完善.

2.中间层

基于组件的软件开发技术可以在较高的级别上实现软件复用，加快企业软件开发的进程.在J2EE构架中， JavaBean和EJB(Enterprise JavaBeans) 被用来完成事物逻辑.其中EJB和 JavaBean 有着类似的模型，但它被用来创建分布式的企业应用.它定义服务器端组件的模型，具有以下一些特性:

生存期模型；

访问模型；

安全模型；

事物处理模型；

会话处理模型；

数据封装模型；

部署模型

根据这些模型，简单的编码就可完成复杂的功能。

在微软的.NET平台中，旧的COM 和 COM+的组件模型被新的组件模型所代替。增加了象基于沙箱的安全模型和垃圾回收等功能.并且实现了多重接口继承，扩展的元数据和新的代理模型等.旧有的COM和COM+组件也可被映射到新的运行环境中。

综上所述，两众架构在基于组件的中间层的设计上各有千秋，对于创建分布式的，复杂的，高效的，高可靠性的的应用程序都有着足够的能力。

3.表示层

两种架构都同时支持胖客户端和瘦客户端.即C/S模式和B/S模式.对于C/S模式，J2EE提供了替代Java AWT的Java Swing，同时作为可视化组件的JavaBean也可用来构造系统。对于B/S结构的表示层，J2EE使用 servlet ，JSP(Java Server Page) ，HMTL，WML，XML等工具来实现。

微软的胖客户端技术则由 Windows Forms代替了MFC.它们起的作用相同，在结构上 Windows Forms 被插入到.NET的运行时框架(runtime framework)和组件模型 (component model)中.在瘦客户模型中， ASP.NET代替了旧有的ASP和 HMTL， WML ，XML作为表示层。在 ASP.NET 中，C#，VB.NET等语言的代码片断可被自由引用.ASP.NET 页面被首先转换成中介语言( Intermediary Language)，然后再被 中介语言及时编译器(just-in-time IL compiler)编译，最后运行于公共语言运行环境中，并且 ASP.NET 提供了页面的缓冲，所以，其运行速度要远远快于ASP。

大体上，两种架构所使用的表示层的技术非常类似，虽在细节上各有所长，但总体功能当在伯仲之间。

4.数据访问

J2EE 和 .Net 已不同的形式支持数据的访问。JDBC和ADO一样和所连接的数据库无关，并且通过连接，命令语句和结果集来对数据进行操作.所以属于中间层次的 API.更高一级的数据封装和数据管理是通过实体EJB (entity EJB)来完成的.基于容器管理的实体EJB使开发更快捷，管理更方便.事实上，由于实体EJB的load()和store()方法的同步机制，将大大缓解因并发而使数据库产生的瓶颈.也可以采用不属于J2EE规范的第三方数据访问工具，象WebGain的 TopLink。

而微软的.NET的数据访问工具则由基于XML的ADO.NET代替了基于COM组件的ADO.任何以XML为输出的数据源都可以作为 ADO.NET 的数据源.相应的结果集升级为数据集 (DataSets)，命令语句则升级为数据集命令(DataSetCommands).从形式来看，微软的ADO.NET更新潮和时髦一些，基于XML的特性使其可以处理极其丰富的数据源，并且，因其构架在HTTP协议之上，易于穿透防火墙，使沟通更为便利.但由于XML本身的基于标记的特性，很明显限制了在有超大数据量和有网络瓶颈的应用中的使用.而J2EE的数据访问规则则显得略有单薄，但同时却更简单，更有效.并且通过对应用程序有效的层次的设计，对于数据库和基于XML的数据源的访问，也是可以无缝的整合的。

三.整体评价

在微软还没有足以和Java平台相对抗的产品的时候，微软所乐于做是大声的宣传:"write once， debug everywhere"。而它的对手则更乐于这样评价它:"微软开始也喜欢Java，他们喜欢它的方式是让它死去，他们当然也憎恨它，他们甚至憎恨每一个以J开头的单词。"但是现在，形式不同了，微软有了足以自豪的.NET他们可以已他们自己所喜好的方式来对J2EE和.NET来做各种比较。最热闹的应该算是微软出示的第三方对.NET Pet Shop和J2EE的 Pet Store的综合比较了.有兴趣的读者可以到MSDN，[www.onjava.com](http://www.onjava.com/" \t "https://www.cnblogs.com/yinhaiming/articles/_blank)，IBM开发者原地等网站看到相关评论。

J2EE .NET

易用性 \*\* \*\*\*

扩展能力 \*\*\* \*\*

多平台支持 \*\*\*\* \*

多语言支持 \* \*\*\*\*

可靠性 \*\*\* \*\*\*

性能 \*\*\* \*\*\*

可管理性 \*\*\* \*\*\*

重用性 \*\*\*\* \*\*

负载平衡 \*\*\* \*\*\*

开放标准 \*\*\*\*\* \*

就企业而言，内部众多系统的整合、系统的延展性、安全性是更需要注意的议题，而这些都是J2EE的优势，也是微软的不足处。 在效率方面，J2EE阵营主张通过硬件的效能增加来弥补软件的不足.开放标准，功能强大，易于移植这些都是J2EE的卖点。但让人奇怪的是IBM的WebSphere和BEA的WebLogic在J2EE市场占了大半壁江山，而作为规则制定者的SUN却在做壁上观。

微软确实提供了从桌面的办公软件，开发工具，到后台服务器数据库的全方位的产品。 但统一平台的使用者可能要牺牲跨平台的好处，并也有可能由此就被无穷无尽的锁定在微软的许可证的汪洋中.更简单，更快捷，更高效是微软的目标，随着时代的发展，我们也许会看到更完美的技术解决方案。

关于.NET技术与Sun公司的Java2企业版（J2EETM）相比较，许多客户都想了解Microsoft公司的观点。由于以下的几个原因，.NET和JEE的比较有点棘手：  
  
1)   一般来说，Windows .NET Framework是Microsoft的Windows系统中经过精心定义的技术部分，而J2EE则是一个书面的协议。如果不局限于学术方面的讨论，换句话说，就是在几个应用平台上讨论这个话题的商业价值，那么仅仅比较J2EE和一个实际应用的工具是没有意义的。  
  
这样实际应用的工具如：IBM公司的WebSphere应用服务，BEA的WebLogic服务或是其它类似的应用服务。  
  
要想得到[令人满意](https://www.baidu.com/s?wd=%E4%BB%A4%E4%BA%BA%E6%BB%A1%E6%84%8F&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/tyouvivi/article/details/_blank)的分析，只有进行产品之间的比较，例如比较开发效率。使用J2EE，开发者需要创建4个组件来建立一个单一的EJB。表面上看来，这只不过是为开发效率付出的一点代价而已。但是Java的一些开发工具隐藏了一些开发技巧，降低了效率。另一个例子，J2EE的部署体系十分复杂难解，类嵌入 JAR，而JAR嵌入WAR，WAR又嵌入EAR。但是在一定程度上，有些工具能自动完成部署进程。上述情况导致决定一个应用服务商业价值的关键因素开发效率因不同的销售商而有差异，这主要取决于开发工具的效率。  
  
2) 关于"J2EE全明星队伍"的问题。当比较.NET和J2EE所有组件的集合时，这个问题就产生了。例如，分析者考虑开发效率时可能碰到下列问题，A公司的产品， B公司的应用服务程序， C公司的安全规则， D公司简便安装， E公司决定价格。所有这些都可能和J2EE有关。集合上述这些特征属性，J2EE工具看起来还行：价格便宜，安装简便，速度快，安全性高，有超高速缓存，并且有好的开发工具，等等。但这些都无关痛痒—因为不可能同时获得所有的这些特性。事实上，一次只能得到一个准确的特性。因为这些产品来自不同的公司，它们不可能合作无间。例如，IBM公司的工具不能和BEA公司的WebLogic服务同时工作，因为后者是用的Oracle公司的缓存引擎，而 Oracle的引擎不能用Iona的价格获得，等等诸如此类。人们有时候会误将"J2EE的所有特性集合"作为比较的基础；但这是不合理的。客户需要的是知道一对一，产品对产品的比较。  
  
3）比较.NET和J2EE而忽视其它应用平台是十分重要的。J2EE是仅关注应用程序服务器的规范。但是绝大多数客户对下列这些感兴趣：应用程序服务器，端口，商业服务器和分析工具，数据库，分离数据和流动性，信息代理，应用程序集合，容量管理，智能客户端等等。作为对客户要求的回应，这些因素应该统一工作，所有的主要销售商应该推行整合的平台。例如Microsoft的平台（包括Windows系统的客户端和服务器，Windows .NET Framework，Visual Studio.NET Framework，和Microsoft企业服务器）；BEA的WebLogic平台；IBM公司的WebSphere平台；Oracle的平台；还有Sun公司的一个平台。将精力集中在这些平台的一个难题（应用服务器）上将会导致一个类似"树林和森林"关系的问题。这样的一个比方是合适的，但是它应该被考虑到一个更广阔平台的一部分。  
  
从Microsoft的角度来看，和那些不常见的警告相比，这些是Windows .NET Framework和基于J2EE的产品的关键性的异同点。  
  
相似点  
  
1）  Windows .NET Framework和Java都有一个受控的运行时环境，它不但将源代码转换成中间语言，而且将这些中间语言编译成本地的可执行代码。两个环境都支持碎片整理、动态类加载和异常处理等。  
  
2）  .NET和Java都倡导和支持基于组件的设计、多态性、继承和接口等，也提供基础类库来执行I/O、XML处理、带有连接池的数据库接入、文本操作与网页脚本编写等。  
  
3）  两者都经过特有的销售商的产品进行发布。J2EE规范自己是"销售中立"的，但实际上那些遵从规范的产品都必须实现规范外的特性，例如管理特性或是展开特性。因此，这些产品必须是对应特定的销售商。例如Microsoft公司的Windows和.NET系统。  
  
4）   Windows .NET Framework和基于J2EE的产品都和第三方的产品一起工作。例如，在后端数据库领域，.NET和基于J2EE的应用程序能访问储存在Microsoft的SQL服务器、IBM的DB2、Oracle，Informix、Sybase等服务器里面的数据。再举一个例子，.NET和基于J2EE的系统能访问流行的信息中间设备，如Microsoft的MSMQ或是IBM的MQSeries。同样，也包括文件系统，第三方开发工具，代码版本系统，防火墙等。  
  
不同点  
  
1）  原理  
  
J2EE是一个单一语言的平台，关注跨平台的可移植性。这就意味着，要利用J2EE，设计方案能使用多个操作系统其中的一个，但开发者必须接受关于Java的培训。Microsoft提供的.NET构架作为Windows系统的一部分。开发者能使用多种语言，并且效率很高而不用进行一种新语言的重新训练。但.NET Framework是 Windows系统的一部分。  
  
2）  宽度和广度  
  
a.       .NET包括代码、产品、工具和构架，来利用网络上全部的计算资源，包括设备、个人电脑和服务器等。.NET使所有的这些设备能经过标准通讯协议全部连接在一起，即所谓的"XML WEB服务"。（.NET应用程序可以和任何一个系统连接，无论系统用什么语言和平台，甚至是J2EE。只要目标系统遵照XML WEB服务标准。）.NET模型是广泛的分布式计算，它和许多代码互相通讯并交换信息。  
  
b.       J2EE是面向服务器的模型，它并不开发网络上的智能和计算功能。总的来说，基于J2EE的产品只支持服务器端的应用程序。J2EE一般把PC只看作是一个HTML的浏览器，而将这些设备认为是哑终端。至于 XML WEB服务，现有的协议标准支持分布式的计算，现有版本的J2EE规范并没有提到XML WEB服务的问题，但是基于J2EE的产品在添加了附加装置后也可以支持XML Web服务。然而，添加附加装置也就意味着有严格的限制。例如，还不清楚现有的规范是否允许EJB调用Web服务，虽然Web服务的组件能调用一些EJB程序。  
  
3）  编程模型的一致性  
  
Windows .NET Framework提供了一个跨服务器、PC和其它设备的一致的、面向组件的模型。而J2EE提供EJB作为服务器端的组件模型；为客户端或是本地组件建立开放的完全用Java编写的API；为用户界面提供servlet；也为移动设备提供另一种不同的模型。甚至在EJB内部也有至少3种明显不同的子模型，每一种子模型都有不同的语言定义。  
  
   
  
Microsoft的.NET编程模型与Java平台相比较，在各种服务器和客户端上有更好的一致性。J2SE是基于开放的完全用Java编写的API，而J2EE是基于Java servlet和EJB。  
  
DH Brown, 2002年7月  
  
   
  
4）  功能  
  
a.       Windows .NET Framework 提供一个能识别版本的类加载器，这就意味着应用程序的开发者能确保他们开发的应用程序在一部分代码已经更新的情况下仍能运行。而Java和J2EE（现有的）没有版本识别的类加载器，这就意味着开发者和管理员不能保证代码被执行时是正确的。或是说，开发者只能靠运气来保证这一点。  
  
b.       Windows .NET Framework 显示了语言层面上的类属性—这就使得编程更加简单。例如，在源代码中只用一个简单的属性就能把.NET组件标志为处理模式。或者说，一个.NET组件和 XML的串行化可以在一个属性中被定义。这个机制大大简化了许多编程任务。而Java不显示语言层上的类属性，虽然Sun公司考虑到要修改Java语言来改变现状。这种变化估计在两三年内才能第一次实现。  
  
c.       .NET还支持分离数据访问，这主要用于在移动设备或是偶尔联网的场合里运行的应用程序。数据能被脱机操作，接着再和起始数据重新同步。而不论是J2EE还是J2SE现阶段都不支持分离数据访问，需要这项功能的 J2EE开发者必须自己写"plumbing code"。  
  
d.       为建立基于网络的用户界面， Windows .NET Framework提供基于事件的模型，这些模型类似于流行的Visual Basic中的智能客户端模型。ASP .NET 模型使得建立、发布和维护一个基于网络的用户界面变得更加容易。与之形成对比的是，J2EE在JSP中不支持这样的模型。有一些第三方的扩展程序部分弥补了这些功能，但是它们的实用性和简便性不能和ASP .NET相比。作为一个推荐的J2EE附加程序，Java Server Faces可能做到这一点。但这个附加程序并没有包括在J2EE的1.4版本以前。而要获得销售商的支持，则又需至少一年的时间。  
  
e.       J2EE支持一个对象相关的数据映像模型，它被称作EJB Entity Beans。这样是为了允许开发者更容易地从一个相关的数据库建立对象模型。然而，实际上把这个想法编程实现却要面对下列问题：  
  
Ⅰ. 易用性：当那些熟知的、正规定义的、被广泛支持的结构化查询语言（SQL）和开发者的数据相互作用时，开发者不得不放弃它们，而使用一个被称为EJBQL 的弱定义查询语言。和SQL相类似，EJBQL的功能并不强大（例如，在现有的规范中，它没有ORDER BY的语句，这样开发者就没法使用特定数据库的 SQL扩展），而它的语义也没有被正规定义。还有，在对象间建立联系和附属关系十分困难，而且在对象间和XML以及后端之间的数据翻译是手动控制的。  
  
Ⅱ.性能：基于EJB系统的性能仍是一个未知数。没有提供公开的基准。客户反映，得到的性能远远偏离了Entity Beans，并且转向一个更直接的数据访问策略。这是EJB Entity Beans没有被广泛使用的一个关键因素。  
  
在Windows .NET Framework 中，数据访问是基于数据集比较的。数据集保存了相关数据的一个子集，它由一个或多个SQL查询语句描述。数据集中的数据可能保存关键的联系，并且开发者能直接对数据进行操作，能将数据转换成XML格式和上次操作的类型，能使用标准的SQL过滤数据等。总而言之，相对于EJB Entity Beans，. NET的数据集模型提供一个更丰富而且简单熟悉的途径。  
  
5）  简便性  
  
a．  配置：对于J2EE，配置是由部署描述信息获得的XML格式的文件，它们和实际执行的商用逻辑代码有明显区别。这种方法有很多问题。第一，考虑到特定类的元数据，有些代码中的改变和元数据中的改变是相互依赖的。两个独立文件的同步性要求有可能产生错误。第二，考虑到应用程序层的元数据，在J2EE中，没有可以从一个程序继承元数据到另一个程序的途径。与J2EE不同，Windows的.NET构架包括了这个功能，使得可以在源代码中直接向类添加属性，这样就不会产生第一个问题。 Windows .NET中的元数据模型允许客户自己添加扩展程序，这样开发者就可以编写和使用自己的属性。为了在Windows的.NET构架中配置外部元数据，这个功能被包括在配置文件的分级系统中，它能从父系统中继承属性，这样每个文件会很小，它只记录改变的设定。这就避免了J2EE模型的第二个问题  
  
b．  数据库连接池Windows .NET Framework中，是根据需要自动建立和管理这些池的。而在J2EE模型中，连接池必须被明确配置和管理。  
  
c．  XML Web服务：在.NET中建立一个XML Web服务就像在类中添加一个属性那样简单。有些基于J2EE的产品也想在Java中模拟这个功能，.NET提供更简单的方法来建立和使用可由双方共同操作的 XML Web服务。  
  
d．   部署：在.NET中，要部署一个应用程序，管理员只需要拷贝文件。而在J2EE中，管理员必须将很多编译文件和JAR、WAR以及EAR绑定，然后在一个特定的服务器部署工具中解开并运行它们，接着拷贝结果档案。这个多步部署过程意味着典型的编辑/编译/调试循环被大大延长了。此外，由于动态加载类过程中的一些变化，更新一个简单的类常常需要重新启动基于J2EE的服务器。  
  
   
  
虽然许多公司选择Java作为企业发展的策略平台，但它们的使用却由于J2EE的复杂性而受到阻碍。Meta Group，8月  
  
   
  
6）  成本  
  
a．   为了部署，运行在Windows .NET Framework之外编写的服务器端的应用程序需要一个Windows Server的许可，这比三个遵从J2EE的商业服务器中的任何一个许可都便宜很多。包括四个网络服务器的系统部署费用的差别可达到数十万美元。例如， Microsoft Windows Server 2003（企业版）的一个四机器系统（每个有四个pc）的许可费用不超过16，000美元（这考虑了零售因素）。而WebSphere Application Server 5.0在同样的系统中每台pc的许可费用达12，000美元，这共要192， 000美元。这个比率是12比1。大多数基于J2EE的商业应用程序服务器的价格都和这类似。（这假定了性能相等。然而实际上Middleware公司 2002年10月的报告显示，一个建立在Windows .NET Framework上的应用程序的效率是建立在同样流行的基于J2EE的服务器上的程序的2-4倍。所以实际上价格的优势远高于12比1）有很多免费的，基于J2EE的开放源应用服务器，但是它们并没有J2EE-compliant的商标。还有关于文件和产品的问题：需要产品之间的比较来讨论采许可费用。  
  
b．  为Windows .NET Framework开发工具的费用也更加低廉。Visual Studio .NET是.NET的整合开发工具，它的许可费用大大低于商业化的J2EE销售商制定的开发工具的费用。并且在业界，Visual Studio .NET作为最佳开发工具赢得了一系列的大奖。评估过Visual Studio .NET和其竞争对手的客户都说，相对于最好的Java工具，Visual Studio .NET开发效率更高（See Giga，2002年6月）。  
  
c．   使用Windows .NET Framework的开发和维护费用更低。专家认为许可费用并不是一个项目的最大开支。典型的软件开发和维护占项目总费用的50-80%（Glass，2002；Kemerer，1995；Gartner，2001）。Middleware公司2002年10月的研究表明，在Windows .NET Framework上一个给定的应用程序开发相对于J2EE，只需要1/3的代码。代码越少就意味着维护更加简单。  
  
总结  
  
   
  
[显而易见](https://www.baidu.com/s?wd=%E6%98%BE%E8%80%8C%E6%98%93%E8%A7%81&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/tyouvivi/article/details/_blank)：正确的产品选择决策不可能不评估实际的产品。对比Microsoft Windows Server及 Windows .NET Framework和J2EE（Sun公司的规范）是有价值的，但是这样的努力缺少实际产品的评估。然而，还是可以从中得出一些结论：  
  
1）  J2EE展现了一个以服务器为中心的原则，并将重心放在EJB和解决"相关对象的映像问题"上。J2EE在支持 XML和Web服务上已经落后了。Windows .NET Framework的原则则是通过协议标准和XML、充分利用服务器、接口和设备的的大规模分布式计算。  
  
2）  相对于编写在Windows .NET Framework上的程序，J2EE应用程序需要更多的代码来执行相同的任务。  
  
3）  J2EE的管理和部署模型更像是一个主机模型，它关注保护和限制稀有的计算资源，按比率使用。而Windows .NET Framework展现出的原则是计算资源是廉价的，而且将更加廉价，但是部署能力将保持大部分昂贵的资源。  
  
总之，如果一个项目要求必须从几个操作系统中选择一个作为部署平台，而不考虑开发成本；强制（并且重新培训训练）开发者使用单一的编程语言来执行这个项目，从而代码的版本问题就不再重要；重要的是配给和限制相对便宜的计算资源；这样使用昂贵复杂的开发和维护工具就显得[顺理成章](https://www.baidu.com/s?wd=%E9%A1%BA%E7%90%86%E6%88%90%E7%AB%A0&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/tyouvivi/article/details/_blank)；而编写更多的代码也有其优越性 -- J2EE也许是一个不错的选择。  
  
然而，如果商业目标显示最优化的开发效率是重要的；低廉的性价比更符合要求；通过通讯协议的标准获得的可相互操作性有较高价值；大量支持基于界面的应用程序和移动的应用程序是重要的；更感兴趣的是易扩展性—这样的话，建立一个 Windows .NET Framework上的Windows Server应用程序是正确的选择。