

## 不同用途的工作站对系统性能的需求介绍

一个对于工作站的定义是：一个帮助开发项目的工具，开始一个新项目，导入数据然后开始安装，做一些小的修改然后项目就完成了，接着进行各种测试和试用，然后产生最后结果。

评价一套计算机系统的性能好与差，一般都是考察该系统的处理器、图形处理、存贮 I/O、网络子系统的性能。

一. 工作站的分类：多媒体工作站（视频处理、音频处理）、图形工作站（平面设计、CAM/CAD、3D 动画、计算机仿真）、其它工作站。

二. 工作站对计算机性能的要求。

### 1.多媒体工作站

其工作过程大概可概括为：采集、编辑、预览、生成输出这几步。

采集过程：主要是与存贮系统的数据传输速率有关，速率低时会造成节目丢帧。

编辑过程：主要强调存储系统的读写能力。CPU 以及内存的带宽。

预览过程：主要强调存储系统的读写能力。

生成过程：主要强调 CPU 的处理能力，其所花时间与 CPU 的主频和数量成正比例，其次也与内存带宽以及存储系统 I/O 操作有关联。

其中哪一个部分最为重要取决于工作站使用的软件、进行的项目以及系统硬件。

对于商用级视频工作站来说，还要特别强调系统的稳定性。

### 2.图形工作站

其工作过程大概可概括为：建模编辑、渲染输出这两步。

建模编辑：主要强调图形子系统（3D 加速、材质、贴图、光影、失真、色彩还原、抗锯齿）、处理器子系统、存贮子系统的性能（作高速缓存）。

渲染输出：主要强调处理器子系统（其性能与 CPU 的主频和数量成正比例），其次也与存贮子系统内存带宽以及容量有关。

对于大型的 3D 动画项目，可能还需要进行网络渲染，此时对网络性能要求较高。

### 3.其它工作站

像软件开发这样的工作（被归类为工作站环境一类的）往往都被视为轻量级的，安装对硬件的要求很少甚至没有，编辑有点像运行 office 软件包，就像在大型桌面软件上运算小量数据。测试就取决于编译速度以及软件在什么地方被执行（可以在工作站也可以在远程的服务器），产生最后结果需要做的也不多。

## 不同用途的服务器对系统性能的需求介绍

一.服务器的分类：数据库型服务器、非数据库型服务器、计算服务器。

二. 不同类型服务器对性能的需求。

### 1.数据库型服务器。

虽然世界上有许多种通用数据库软件，比如甲骨文公司的 Oracle、IBM 的 DB2/免费数据库 MySQL 软件，但市场份额最大的还是微软的 SQL 数据库，许多服务器仅仅运行一套 SQL 数据库程序。对于运行数据库的服务器而言，要提前对服务器实际性能进行预测是很困难的。大型数据库和服务器厂商，如 Oracle 和 Sun，都免费提供服务器性能咨询服务来帮助客户确定他们的硬件需求以满足软件和应用对性能的要求。之所以需要这些咨询过程是因为没有数据的数据库是没有用的，每个数据库的数据的结构、

容量以及用途都是各自不同，也就造成了同样硬件配置运行同一数据库软件时，在不同数据量和数据类型的客户应用中，服务器的性能就表现出较大的差异。

#### \*\*\*\*数据库的不同应用场合举例

一台空数据库就像一个只装有操作系统和编译器的电脑，仅此而已，剩下的就取决于谁拥有这个系统，这也是要预先估计系统性能如此困难的原因。数据库安装起来也非常复杂（Oracle 数据库拥有操作系统的大多数功能，并且还有更多可选项来设置），正因为这样，技术顾问们才如此有利可图。SQL 数据库系统几乎没有预封装解决方案（pre-packaged solutions），取代的是一个庞大的咨询行业来帮助企业制定方案。但也有许多无需定制的软件采用了 SQL 数据库（如那些 SAP 公司的软件），只不过每套软件都有各自的特点。

不同类型的网站就有不同的数据库使用方式——以新闻为主的网站，其数据库主要是进行读操作；对于拍卖网站，其数据库主要是进行大量的读和写操作，对于在线商店，股票交易网站，为了安全方面的原因，数据库经常要用大量的复杂加密算法来对客户进行身份认证，其数据库主要是进行计算和写操作。数据库的使用方式中也有许多是不容易观察到的，例如，将网站的具体访问情况记录在数据库里来进行数据挖掘（data-mining），这项操作的开销会大到需要在一台单独的数据库服务器上以避免影响对外进行的业务。对于开展电子商务业务的网站，典型的操作包括得到每天客户购买商品的模式，以及简单的收入汇总，统计最受欢迎以及最不受欢迎的产品，以及客户对于网站服务的意见和送货情况。

对于简单一些的“传统企业上网”类型的公司，他们也有收集每天，每周，每月，每季度以及每年报表的需要。像沃尔玛特（Wal-Mart）这样的公司在每家分店都装有小型的数据库应用程序，来记录商店的货品销售情况，还可以用来向供货商发送订单以及跟踪顾客的购买模式。这些小型的数据库会有序地将数据反馈给一个集中的数据中心。在沃尔玛特这个例子中，沃尔玛特还会将已有商品的顾客购买情况实时地传送给他们的供货商，以便供货商能更有效地管理他们的供应链和送货时间。如果供货商就是生产商的话，他们还能更好的调整他们的生产计划。

还有从事银行和信用卡业务的系统，这样的系统需要有巨大的数据库系统（或者许多分散的数据库系统）来进行数量巨大的业务。要处理写操作十分频繁的数据库是很困难的——我曾经看到过老版本的 MySQL 数据库，它更多是为快速读操作而设计，当写操作变得频繁的时候就会变得很慢，但是想象起来跑的很慢的数据库系统，如 Oracle 在这种情况下则要快一些。最近推出的新版本的 MySQL 已经改变了这种情况，这是个优化和软件设计时的可扩展性的问题。

\*\*\*\*

通用数据库（包括所有的我所知道的 SQL 数据库）没有一个定制的数据库效率高（为特定应用程序设计和优化的数据库）。作为交换，这也意味着定制的数据库的造价要比通用数据库高得多。

最后，购买新的数据库服务器有通常有两个主要原因——一个是为了新的数据库系统的需要，另一个是为了升级现有的系统。对于现存的数据库系统，进行性能测试要容易得多，因为软件已经存在了，还有许多的使用情况可供参考。对于新的数据库系统，这两者都没有。解决这个问题的办法之一是大概地估计一下需要的数据容量，接着查看类似的大型系统的性能测试值，然后抛弃任何性能很差的系统。另一个解决办法是在早期就购买一台开发和测试服务器，然后当项目需要投入实际运行的时候，再参考当前的

开发环境中的服务器的性能测试值购买一台生产服务器。

同工作站相比，SQL 数据库服务器在确定其性能的时候就要复杂得多。常常要估计目前对于数据库需求有多少，而一段时间之后又是怎样。

除了已经投入使用的软件，在数据库和使用数据库的软件被安装好以前，想要预测所能获得的性能都是很困难的——软件和安装过程在不同时期也会改变。要预测处理的数据量以及在不同时间数据量的变化也是很困难的，这部分取决与实际需求。

在实际应用中，有一些服务器的硬件配置不低，但运行的效率不佳，究其原因，只是因为数据库和使用数据库的应用程序常常没有被正确地安装和优化。安装并优化数据库设置常常都是复杂并有一定难度的（电脑系统，操作系统，数据库软件以及特定的应用程序）。

其实许多数据库操作（执行 SQL 语句）是非常简单的，并且只需要一秒钟的一小部分来执行，因为存储系统常常能保证大多数任务能迅速的立即执行。装载一篇文章的一个页面是非常简单的，也很容易优化并仅输出 20—50KB 的数据。对于部分有选项的页面，装载用户参数（preference）也是很简单的，并且需要的数据更少。这些操作都只涉及了从数据库中装载几行很容易辨认的数据。

而要有序地显示论坛中最近几天发布的帖子就要复杂得多了，这样会使数据库在几百行中搜索并排序后，输出相当于大约 200KB 的数据（取决于论坛最近的使用情况）。

网站上的大多数页面都只需要进行单条数据库查询（或者几条简单查询语句），但是首页就是完全不一样了（大多数网站都是这样），它需要数据库查询网站里的所有相关内容。不过，Web 应用程序里的缓存极大地降低了网站所需要的数据库操作，数据库几乎不会占用超过 5% 的 CPU 资源。

大多数网站服务器的最经常使用的数据总量其实并不大，可能 8M 的 L2cache 就足够了，而这么大的 L2 cache 对于目前的 64 位多处理器 RISC 服务器来说是很普遍的。甚至可以在一台低端的装有单颗 0.25M cache CPU 的服务器上一样运行良好。如果网站这样的应用程序真的需要一台现代的多处理器系统，除非是因为拥有许多火爆的论坛，或者存有大量的文章，再或者就是实施了一个效率很低的项目。如果真的如此，即使 8M cache 对于这么多的活跃数据也是不够用的，并且内存系统的压力就会增大，甚至可能也会加重存储 I/O 系统的负担。

结论：由于大型数据库软件对多 CPU 系统进行了很好的优化，要提高运行数据库服务器的性能，增加系统中 CPU 的数量比提高 CPU 的主频效果更好，对于任务繁重的数据库服务器来说，采用 4-8 路带有大容量 L2/L3 高速 cache 的高级服务器，能有效的提高服务器的响应速度。

对于数据库来说，一次写操作占用的时间是读操作的 9 倍，为了保证较快的运行效率，在数据库的设计中，应尽量减少写操作。

## 2. 非数据库型服务器

除了 SQL 数据库服务器，还有许多其他类型的服务器（也还有其他的多种数据库服务器），例如，Web 服务器，网络文件服务器，网络路由器，防火墙，邮件服务器，目录服务器，缓存服务器以及其他各种涉及用户管理、身份验证和管理类型的服务器。

所有这些服务器在工作时往往都需要进行很多网络输入/输出，并且也会进行很多文件系统的 I/O。然而，越是 I/O 密集的服务器类型（文件服务器，路由器，防火墙，缓存服务器）需要的 CPU 运算越少，并且常常都是采用单 CPU 配置，这样也能降低产品价格。

在实际应用中，一台性能优异的服务器就可同时胜任多种功能；像 Web 服务器和邮件服务器这种类型的服务器，以及涉及用户服务和管理的服务器也会频繁使用 SQL、MySQL 等数据库。实际上电子商务软件中的一些模块—比如订单管理、供应链管理、资源管理、甚至薪水管理等等，这些都会非常频繁的使用数据库。

一般来说，静态网站中的服务器是不运行数据库的，对服务器的性能要求低，对于动态网站来说，其中执行某些功能的服务器是要运行数据库的，对性能要求较高。特别推荐使用双路 Xeon 服务器，经济又实惠。

### 3.计算服务器

这类服务器就需要高浮点运算（有时也要求整数性能）与良好的可用的内存带宽，当然有时也需要容量巨大的快速存储系统。

## CPU 的主频与数量对系统性能的影响。

### 1. CPU 主频对性能的影响

对于 CPU 主频的提升，都将会带来性能的改善，一般来说，Pentium4 CPU 每提高 100MHz 主频，SPECint 和 SPECfp 性能增加约 20% 的情况。但随着主频的升高，性能提升会变低，其原因都是因为内存系统性能是不变的，CPU 将花费更多的时间来等待内存请求。

对于不支持多CPU的应用程序来说，只能靠增加CPU的主频来提升程序执行的效率。但是多CPU系统可以同时执行多个程序，这样也能有效的提高系统的运行效率。

### 2.CPU数量对性能的影响

一般而言，软件（或者基准测试）究竟能利用额外的CPU到什么程度很多方面都取决于这些软件本身，有些软件在多处理器系统和分布式系统性能提升不大，部分软件很难做到利用多余的CPU和系统资源，有些则根本不可能做到。那些需要花些时间来完成的工作站上的任务往往需要处理大批的数据，让这些运算变成并行处理一般通过将数据按比例分给每个CPU来处理，最后将结果聚集起来实现。服务器任务一般涉及对每个请求作出回应，让这些操作变成并行处理一般通过将这些请求传送给最先变得空闲的CPU来完成。

同一个系统上不同CPU上进行的单独运算（线程和进程）将竞争系统的共享资源，如，公共的物理内存，内存带宽，CPU时间和I/O。在一个 4 路系统上，如果所有的CPU都是繁忙的，那么每个CPU所能获得的最多的内存带宽仅仅是系统带宽的 25%。所以系统资源越多，最高性能就越好。

当添加更多的CPU时，对一个系统的性能提升程度的判定往往是很主观的。例如，如果一个系统的CPU数目增加了 100%（加倍），性能（特定基准测试得分）增加了 90%，大多数人就会认为这个系统（以及应用程序）是可扩展的。换句话说，每个CPU所能获得的性能是不变的。然而，一个可扩展的系统设计并不需要比那些不是“可扩展”的系统拥有更好的最高性能。拥有一个可扩展的系统让预测系统性能更加容易，并且使有些人可以吹捧他们的系统或者软件是“可扩展的”。

如果一个特定的CPU以及系统设计能良好的扩展到 2 路，那么采用相同CPU的一个类似的 4 路系统设计就很有可能拥有一个良好的性能，因为如果这个系统能良好地扩展到 2 路，它很可能还有很多系统资源没充分利用（如内存带宽），所以一个 4 路设计将能利用那些剩下的系统资源。4-way系统可能也有多余的系统资源，但这样显然会增加

系统价格。可扩展性能越强的系统，可升级的能力也应该越强，随着时代发展，更快的CPU也会诞生，也会需要更多的系统资源（CPU各部分的速度增长 10%就意味着cache miss发生的频率会加快 10%，所以对主内存的需求也增加了）。所以一个系统的空余资源越多，它就越能更好的利用速度更快的CPU。一台服务器使用超过 5 年是普遍的，所以他们的可升级能力越强就越好，虽然许多实际上不会被升级。

看一下每个CPU所拥有资源很多（缓存和主内存带宽）的系统 and 每个CPU拥有不太多资源的系统之间的差别，可以参考下面的SPECint 测试结果来了解从 4 - w a y P e n t i u m 3 X e o n 系统升级到 8 - w a y 系统，以及从 2 - w a y U l t r a S P A R C - I I I 系统升级到 8 - w a y 系统之间的差别。在每次的比较中，CPU都是一样的，带有 2M 内置L2 缓存的 700MHz Pentium 3 Xeon 处理器，以及 8M外置L2 缓存的 750MHz UltraSPARC-III 处理器。

对于下面的图表，分值 100 的效率意味着性能是按正比增加的，分值为 0 意味着性能降低为 0，分值为 50 意味着性能没有改变。

Pentium 3 Xeon 4-way 到 8-way 的可扩展性	
测试	效率
164.gzip	82.6
175.vpr	54.6
176.gcc	79.9
181.mcf	37.5
186.crafty	88.4
197.parser	69.9
252.eon	99.3
253.perlbnk	92.8
254.gap	52.7
255.vortex	78.0
256.bzip2	69.0
300.twolf	86.7
SPECint_rate	71.8

绝对地来说，8-way Pentium 3 Xeon 系统仅比 4-way 系统快 44%，这意味着虽然有了多余的 4 颗 CPU，但系统只获得了 1.76 颗 CPU 所能带来的性能，是对于投资的一种极低的利用。这种水平的可扩展性并不让人十分吃惊，因为每组 4 个 CPU 仅能获得 0.8GB/s 的内存带宽。从部分来看，252.eon 结果看起来这项测试很适合 Pentium 3 Xeon 处理器的 2M cache，因为测试结果是线形增长的（成正比增长），cache 命中率（cache hit rate）越高，需要的主内存就越少，就给其他 CPU 剩下的越多。

在有些测试中，8-way 的系统实际上比 4-way 的系统做的更糟，这可能是由于芯片组的不同，Pentium 系统总线上的更多 CPU 带来的资源竞争导致了系统效率的降低。而系统中的编译器（compiler）则不会带来这么大的差异，因为每个类型 CPU 所进行的测试都是在类似的时间里使用的同一种编译器。

UltraSPARC-III 2-way 到 8-way 系统可扩展性		
Test	Efficiency	0  25  50  75
164.gzip	96.4	
175.vpr	92.6	
176.gcc	96.5	
181.mcf	92.4	
186.crafty	98.0	
197.parser	98.2	
252.eon	98.5	
253.perlbmk	97.9	
254.gap	90.9	
255.vortex	93.8	
256.bzip2	97.3	
300.twolf	98.1	
SPECint_rate	95.9	

UltraSPARC-III 系统在这个 benchmark 套件里的表现出很好的很有成效的可扩展性，因为它有一个更大缓存组合的支持（不过我认为对于 SPECint 测试，2Mcache 和 8Mcache 不会带来太大的不同）和一个内存带宽几乎随着 CPU 数目线型增长的设计，大约 2.4GB/per CPU。当拥有了 6 颗更多的 CPU，系统获得了 5.75 颗 CPU 所能带来的性能增长。

在某种程度上，这种互相比对于 Pentium 3 Xeon 处理器有些不公平，因为它比 UltraSPARC-III 处理器要成旧许多，CPU 的性能已经远远超过了最初为 200MHz CPU 所设计的结构。在另一方面，想要在从 2-way 系统扩展到 8-way 系统的的过程中使系统性能得到线形增长是更加困难的，因为从 4-way 系统扩展 8-way 系统的过程中，系统效率的降低几乎是从 2-way 到 8-way 系统扩展中效率降低的一半。就绝对性能而言，750MHz UltraSPARC-III 处理器要比 700MHz Pentium 3 Xeon 处理器快 40%。

### 随着 SPECfp\_rate 扩展

对于 SPECfp\_rate 还有更多完整的测试结果，下面就是一些系统的一些全面的可扩展性低效率等级。

SPECfp_rate 的一些 CPU 可扩展性测试				
测试平台	效率	可扩展性		
		0	25	50
2-8 US3	87.5	[Progress bar showing 87.5% efficiency]		
1-2 Pentium 4	82.1	[Progress bar showing 82.1% efficiency]		
1-2 AthlonXP	79.7	[Progress bar showing 79.7% efficiency]		
2-4 Itanium	78.9	[Progress bar showing 78.9% efficiency]		
4-8 P3 Xeon	51.1	[Progress bar showing 51.1% efficiency]		

综合地来讲，UltraSPARC-III 系统在可扩展性上占据了首位，尽管需要从 2-way 系统扩展到 8-way 系统。而 Pentium 3 Xeon 系统在 SPECfp\_rate 测试上的成绩则很糟糕，8-way 系统的得分仅比 4-way 系统的略微高一点。但应该记住，SPECint 和 SPECfp 在次级测试中的结果都变化很大，所以提前预测可扩展性（在没有相关系统的基准测试结果记录的情况下）就像是无的放矢，所能得到的只是一些大体上的结论。

### 可扩展性的服务器基准测试——SPECjbb2000

SPECjbb2000, 一个服务器端的部分基于 IBM 开发的 TPC-C 的 java benchmark 软件，是一个可以有效地测试可扩展性的基准测试软件。因为在一次单独的 benchmark 运行中，它会在测试的开始阶段展开一个线程 (thread)，然后逐渐增加同时运行的线程数，直到整体性能不再增加，接着会采取一系列措施，用这些措施了获取最后的 benchmark 得分。这里，可扩展性测试的结果是在成倍增加系统的活动线程一直到系统所支持的最多 CPU 所能支持的活动线程的数目时取得的。

用 SPECjbb2000 进行的系统可扩展性测试		
系统	JAVA 虚拟机(JVM)	扩展效率
4 - w a y P 3 X e o n	JRockit 3.1	91.0
4 - w a y P 3 X e o n	IBM 1.3.1 JVM	92.0
8 - w a y P 3 X e o n	JRockit 3.1	83.0
8 - w a y P 3 X e o n	IBM 1.3.1 JVM	90.1
8 - w a y P A - R I S C 8 7 0 0	Sun/HP 1.3.1	96.0
1 6 - w a y P O W E R 4 H P C	IBM 1.3.1 64-bit JVM	93.7
3 2 - w a y P O W E R 4	IBM 1.3.1 64-bit JVM	87.6
4 - w a y U S 2	Sun 1.2.2 JVM	70.0
4 - w a y U S 2	Sun 1.3.1 JVM	78.6
4 - w a y U S 2	Sun 1.4 beta 3 JVM	91.6

2	4	-	w	a	y	U	S	3	Sun 1.3.1 JVM	96.6
7	2	-	w	a	y	U	S	3	Sun 1.3.1 JVM	82.6
7	2	-	w	a	y	U	S	3	Sun 1.4 64-bit JVM	98.0

这些测试结果从几个方面来说都是有趣的：在Pentium 3 Xeon系统上虽然使用了完全不同的Java虚拟机，得到的结果却相差无几，这就暗示了系统的硬件或者操作系统才是取得不同性能的限制因素。而 8-way PA-RISC系统的测试结果很适合用来做比较，因为PA-RISC8700 拥有总量为 2.25M的cache，仅比 Pentium 3 Xeon多了一点。PA-RISC系统看起来有Sun HotSpot Java 虚拟机的一个接口，在扩展到 8-way系统的过程中性能表现好得多，这可能是由于拥有更大的系统带宽的关系，虽然操作系统的优化也帮了一些忙。将 8-way Pentium 4 Xeon系统面市的时候，将它们进行比较将是很有趣的，因为 Pentium 4 Xeon系统拥有比 Pentium 3 Xeon 系统更大的系统带宽。

IBM POWER4 系统的测试结果特别有趣，虽然两组都使用了 16 颗 POWER4 芯片，但在 16-way HPC测试中，每组POWER4 系统中只有 1 个CPU核心被激活，而在 32-way HPC测试结果中，每组POWER4 系统中的CPU都被激活了。当在双核心的POWER4 系统上仅展开一个线程的时候，这个线程就拥有所有的缓存，当展开 2 个线程的时候，他们就会互相竞争共享缓存，这可能也就是可扩展性没有呈线形增长的原因。

UltraspARC-II系统的性能测试结果是通过运行一个叫做 Gcocco的第三方公司的软件得出的，这可能也是测试结果没有采用最新的（更可扩展的）Solaris 8 提供的内存管理选项以及线程库（thread libraries）的原因。无论如何，通过在同一个 4-way UltraSPARC-II系统上进行测试，但使用不同的Java虚拟机，就可以清晰的看到 1.4Java虚拟机相对于 1.2.2Java虚拟机所带来的可扩展性的提高。这在 72-way UltraspARC-III系统的得分上更容易看出来，它的得分显示 64 位 1.4Java虚拟机相对于 32 位 1.3.1Java虚拟机所带的可扩展性的巨大提高。

综上所述，系统可扩展性涉及到硬件，软件（操作系统和应用程序）以及软件被优化的程度。综合的讲，Pentium 3 Xeon系统在SPECjbb测试中的可扩展性表现要比在SPECint中的好，这表明了这项测试并不需要太大的系统带宽，据我所知，即使在最大的系统上，benchmark也仅仅使用以GB计算的内存中的一小部分。当然，拥有更多缓存和带宽的系统肯定拥有更好的可扩展性。SPECjbb没有显示出的一个方面的情况是（并且实际上很少benchmark显示出来）当处理的数据的复杂性增加的时候，系统性能的扩展情况。这是很重要的一个方面，因为随着时间变化，现实世界中的应用程序所处理的数据将变得越来越复杂和巨大。

## 人 其他服务器基准测试

评论基准测试软件要比编写他们容易得多，并且目前已有的benchmark软件并不是一直都能得出最新或者全面的测试结果，这也让对其进行比较更加困难。我认为一个常见的问题是大多数benchmark软件仅仅注重性能或者性价比，这些很适合大多数高性能电脑应用程序以及许多工作站应用程序，但商业客户更关心要达到那些性能要求的最便宜或者最好的解决方案。**了解一台服务器的最高性能是很有用，但在实际的实施中，大**

多数CPU的平均利用率都在 20-30%之间，现实世界要让benchmark怎么做才能被看作是有用的呢？

对于这篇文章来说，显然应该在其中包括一些数据库benchmark结果，并且已经有许多数据库benchmark可供选择（其中最出名的当然是TPC系列的），一些分析家建议这些benchmark结果并不能完全反映现实世界中的实施项目的性能。例如，即使是TPC-C, TPC-H和TPC-W测试中的低端系统也拥有 2TB左右的存储系统，这已经远超出了人们对于 4-way和 8-way系统的预期值。我还没有遇到一位富有实际经验的技术专家认为TPC benchmark结果是不错的。

无论如何，在TPC-C测试结果中，8-way Intel 系统得到了 4-way的类似Intel系统的两倍得分。然而，在TPC-H测试中，8-way系统就仅比 4-way系统快 50%，这也许是因为TPC-H对于优化操作更加复杂和困难。至于TPC-W测试，它的测试结果就难以解释了，因为从结果来看，厂商们似乎还正在习惯如何优化安装过程。要对低端的RISC系统作出一个合适的估计也不容易。尽管加倍了时钟频率，极大提高了内存带宽，拥有了超线程技术（hyperthreading）以及更好的I/O系统，基于Pentium 4 Xeon的服务器仅比前一代的服务器快 30%。

现实中优化数据库安装的最简单和最常用的办法是，通过在主内存中最大可能的缓存数据以及优化查询语句和应用程序使他们充分利用缓存来使存储系统的读写操作降到最低水平。所以数据库管理员或者开发人员往往对于操作系统或者数据库提供的查看磁盘读写请求频率的工具都比较熟悉。所以瓶颈在I/O操作上的benchmark结果或者经过benchmark测试的系统并不能完全反映现实世界中的类似系统的性能。部分高端的数据库应用程序往往也受到I/O性能的制约。

然而，即使数据库系统通过缓存数据而进行非常少的I/O读请求，拥有一个高性能的I/O系统也是很有用的，因为有些不常进行的操作，如备份和恢复数据或者很少进行的查询（月末，季度末，年末进行的那些查询）都会使用数量巨大的数据。并且，大多数数据库都运行在分布式服务器上，这意味着所有的请求和数据都是通过网络进行的，所以即使存储系统的读写操作很少，网络间的I/O传输也有可能成为系统的瓶颈。

通过上面介绍我们了解到大多数（经过良好优化的）现实生活的数据库系统不会受到存储系统I/O的制约，这使得对优化过的数据库的性能测试更加侧重于CPU和系统性能。然而，这并不意味着SPECint（侧重于CPU整数运算的benchmark）将是一个有意义的替代品，特别是对于多处理器系统。没有一个SPECint（或者SPECft）软件是多线程的（至少没有这样采用），只有一种看起来会直接加重数据库的负担，那就是in-memory object database，而这和企业级的SQL数据库比较起来要简单得多。此外这些benchmark软件都尽力使对shared library和操作系统的的使用最小化，而现实生活中的数据库会强调操作系统内核的每一个部分并且会频繁使用操作系统的shared library。

对于多处理器系统，计算等级版本的SPECcpu benchmark软件也不过是同时执行同一套benchmark软件多次而已，他们之间并没有像锁机制那样的依赖性。此外，在多处理器系统上运行的数据库系统常常同时处理同种类型的查询语句，而且每种语句的数据集大小也不一样，所以每条语句对于内存的需求也是不一样的。然而，当运行

SPECint\_rate和SPECfp\_rate 这两种benchmark软件的时候，同一种benchmark软件在同一时间采用的内存读取模式都是一致的。

虽然也可以采用一些micro-benchmark软件来测试数据库的某些方面的性能，我觉得开发一些更加复杂的benchmark软件有点没有必要，因为现实世界中实施的许多项目之间的差异非常大。许多现实世界中的系统会某些厂商提供的特定特性，而基于业界标准的benchmark不得不跨平台地运行在所有的数据库上。但还是有不少公司采用了数据库程序包（database application packages），这样就会有更可预测的数据结构和读取模式。现在也有基于这些程序包的benchmark软件，这样可能得到最真实的数据库benchmark结果。

### 服务器以及工作站平台上的benchmark软件列表

- **S P E C w e b 9 9** 相对简单的 webserver 的基准测试
- **S P E C w e b 9 9 \_ S S L** SPECweb99 的SSL版本
- **S P E C S F S 9 7** 基于NFS的file server的基准测试
- **S P E C m a i l 2 0 0 1** 电子邮件系统的基准测试
- **S P E C j b b 2 0 0 0** 基于Java的业务基准测试
- **E C p e r f** 企业Javabeans的基准测试- 马上将被 **S P E C j A p p S e r v e r 2 0 0 1** 代替
- **S P E C C P U 2 0 0 0** 整数和浮点运算的基准测试
- **S P E C O M P 2 0 0 1** : 测试基于OpenMP并行化的并行应用性能的基准测试
- **T P C - C** 老事务处理的基准测试
- **T P C - H** Ad-hoc 查询以及决策支持的基准测试
- **T P C - R** 商业报告以及决策支持的基准测试

- **T P C - W** 基于事务处理的互联网电子商务的基准测试
- **O r a c l e A p p l i c a t i o n s**  
**b e n c h m a r k** : Oracle应用程序的基准测试
- **F l u e n t C F D**  
**b e n c h m a r k s** : Fluent公司计算流体动力的基准测试
- **S e r v e r a n d**  
**w o r k s t a t i o n b e n c h m a r k**  
**r e c o r d s a t I d e a s**  
**I n t e r n a t i o n a l** :

Ideas International Limited (IDEAS)公司的服务器和工作站的基准测试（包括产品的性能，价格，服务）。

## 2. 设计工作站和服务器的 CPU 以及系统

当准备设计一个新的处理器以及与其搭配的系统构架时，设计者会首先考虑这款处理器针对的市场。有许多的海外电脑市场—桌面，工作站，移动以及服务器和嵌入式系统，每一种类型的市场都有独自的价格和性能要求。因为开发一款高性能的工作站或服务器 CPU 要花费很长时间（以及许多金钱），大多数设计者随着时间的增长都适应了许多类型的任务。例如一款为高端的多处理器系统设计的 CPU 使用在一款桌面或者工作站系统上，性能就会下降很多（拥有更少的缓存以及系统带宽），反之依然（工作站 CPU 使用在高端服务器系统上）。

对于处理器设计师来说，幸运的是适用于工作站 CPU 的特性也适用于服务器 CPU—更快的时钟频率，更好的 branch（分流）预测，同时执行更多的指令，更低的缓存延迟，更多的缓存带宽，更低的内存延迟，更多的内存带宽，更快的系统设计（I/O 以及多处理器间的连接）。然而，像 SIMD 这样的特性（在一条指令中处理多块数据，适用于 2D 及 3D 图形处理，视频处理，声频处理以及广义上的多媒体操作）对于绝大多数的服务器任务都没有多大用处（除了计算用途的服务器）。这是因为对于服务器上运行的应用程序来说，浮点运算是很少见的，同时处理的数据结构和类型也复杂许多。

服务器应用程序还有更加复杂的逻辑和可选项，这意味着他们比典型的工作站应用程序拥有更多的分支指令并且更难预测分支指令。此外，服务器应用程序的内存读取模式相对于自动内存预读取（pre-fetching）来说更为复杂，而自动内存预读取对于大多数工作站任务都是有效的。然而，产生一个良好的浮点以及 SIMD 性能常常不需要占据 CPU 超过 10—20% 的 die-size 空间，因为 CPU 的缓存系统占据了越来越多的空间。

工作站应用程序处理的数据常常相当简单并且有规律—对于一个简单的 2D 图形处理软件，它处理的图片往往才几千或者几百万像素，最多 32 位色彩（有时候针对医疗应用程序的数据拥有更多的位数）。