

工作,而在下一个进程结束之后机器会再次进行询问。如果没有可做的工作,机器就开始空闲。在经过固定的时间间隔之后,它又开始探查。

这个算法的优点是,在关键时刻它不会对系统增加额外的负担。发送者发起的算法在机器最不能够容忍时——此时系统已是负载相当重了,做了大量的探查工作。有了接收者发起算法,当系统负载很重时,一台机器处于非充分工作状态的机会是很小的。但是,当这种情形确实发生时,它就会较容易地找到可承接的工作。当然,如果没有什么工作可做,接收者发起算法也会制造出大量的探查流量,因为所有失业的机器都在拼命地寻找工作。不过,在系统轻载时增加系统的负载要远远好于在系统过载时再增加负载。

把这两种算法组合起来是有可能的,当机器工作太多时可以试图卸掉一些工作,而在工作不多时可以尝试得到一些工作。此外,机器也许可以通过保留一份以往探查的历史记录(用以确定是否有机器经常性处于轻载或过载状态)来对随机轮询的方法进行改进。可以首先尝试这些机器中的某一台,这取决于发起者是试图卸掉工作还是获得工作。

8.3 虚拟化

在某些环境下,一个机构拥有多计算机系统,但事实上却并不真正需要它。一个常见的例子是,一个公司同时拥有一台电子邮件服务器、一台Web服务器、一台FTP服务器、一些电子商务服务器和其他服务器。这些服务器运行在同一个设备架上的不同计算机中,彼此之间以高速网络连接,也就是说,组成一个多计算机系统。在有些情况下,这些服务器运行在不同的机器上是因为单独的一台机器难以承受这样的负载,但是在更多其他的情况下,这些服务器不能作为进程运行在同一台机器上最重要的原因是可靠性(reliability):现实中不能相信操作系统可以一天24小时,一年365或366天连续无故障地运行。通过把每个服务器放在不同机器上的方法,即使其中的一台服务器崩溃了,至少其他的服务器不会受到影响。虽然这样做能够达到容错的要求,但是这种解决方法太过昂贵且难以管理,因为涉及太多的机器。

那应该怎么做呢?已经有了四十多年发展历史的虚拟机技术,通常简称为虚拟化(virtualization),作为一种解决方法被提了出来,就像我们在1.7.5小节中所讨论的那样。这种技术允许一台机器中存在多台虚拟机,每一台虚拟机可能运行不同的操作系统。这种方法的好处在于,一台虚拟机上的错误不会自动地使其他虚拟机崩溃。在一个虚拟化系统中,不同的服务器可能运行在不同的虚拟机中,因此保持了多计算机系统局部性错误的模型,但是代价更低、也更易于维护。

当然,如此来联合服务器看起来就像是把所有的鸡蛋放在一个篮子里一样。如果运行所有虚拟机的服务器崩溃了,其结果比单独一台专用服务器崩溃要严重得多。但是虚拟化技术能够起作用的原因在于大多数服务器停机的原因不是因为硬件的故障,而是因为臃肿、不可靠、有漏洞的软件,特别是操作系统。使用虚拟化技术,惟一一个运行在内核态的软件是管理程序(hypervisor),它的代码量比一个完整操作系统的代码量少两个数量级,也就意味着软件中的漏洞数也会少两个数量级。

除了强大的隔离性,在虚拟机上运行软件还有其他的好处。其中之一就是减少了物理机器的数量从而节省了硬件、电源的开支以及占用更少的空间。对于一个公司,比如说亚马逊(Amazon)、雅虎(Yahoo)、微软(Microsoft)以及谷歌(Google),它们拥有成千上万的服务器运行不同的任务,减少它们数据中心对物理机器的需求意味着节省一大笔开支。举个有代表性的例子,在大公司里,不同的部门或小组想出了一个有趣的想法,然后去买一台服务器来实现它。如果想法不断产生,就需要成百上千的服务器,公司的数据中心就会扩张。把一款软件移动到已有的机器上通常会很困难,这是因为每一款软件都需要一个特定版本的操作系统,软件自身的函数库,配置文件等。使用虚拟机,每款软件都可以携带属于自己的环境。

虚拟机的另一个好处在于检查点和虚拟机的迁移(例如,在多个服务器间迁移以达到负载平衡)比在一个普通的操作系统中进行进程迁移更加容易。在后一种情况下,相当数量的进程关键状态信息都被保存在操作系统表当中,包括与打开文件、警报、信号处理函数等有关的信息。当迁移一个虚拟机的时候,所需要移动的仅仅是内存映像,因为在移动内存映像的同时所有的操作系统表也会移动。

虚拟机的另一个用途是运行那些不再被支持或不能在当前硬件上工作的操作系统(或操作系统版本)中的遗留应用程序(legacy application)。这些应用程序可以和当前的应用程序在相同的硬件上运行。事