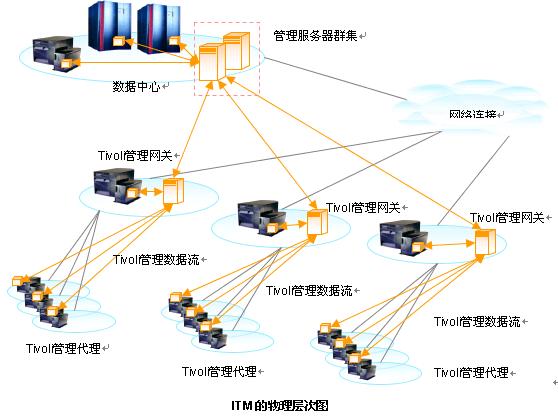
[Mocha BSM与IBM Tivoli Monitoring Agent资源消耗对比](http://www.mochabsm.com/index.php?view=article&catid=101%3Ayuqitachanpinduibibaogao&id=224%3A2008-05-19&option=com_content&Itemid=168)

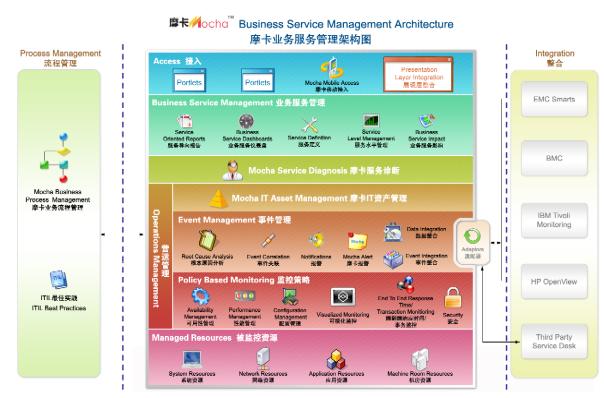
**Mocha BSM与IBM Tivoli Monitoring对比**

**1.前言**   
  
　　企业IT系统越来越多，网络、设备和产品越来越复杂，此时IT部门如果缺乏快速有效的协调机制和必要的辅助管理工具，就会出现“救火队式”的混乱局面，被动响应式的工作方式。以下是这种工作方式的特征：   
  
　　-很难及时发现和预见问题的发生，一旦问题出现后，又很难快速、准确地找到根本原因，并及时地找到相应的人进行修复和处理；   
  
　　-缺乏流程化的故障处理机制，经常出现重复、丢失或者忘记用户的请求和信息，缺乏过程和变化的跟踪记录；   
  
　　-缺乏有效的服务级别管理，支持过程总是被打断和干扰，关键人员的工作负载过重；   
  
　　-大量的系统数据信息无法捕获，导致对于问题的判断缺乏足够的依据，决策基于“我认为”，而不是“我知道”；   
  
　　-IT支持部门面临不断改进服务和降低成本的压力，资源和人力成本计算工具匮乏，导致服务请求的响应时间和质量无法衡量。   
  
　　针对这些问题，各厂商纷纷推出了各自的监控管理软件，这些软件在监控管理IT环境的同时，不可避免的会对我们现有的网络、主机带来一定的性能影响。因此，作为企业的CIO，他们存在不同程度的担心：   
  
　　-由于监控软件的频繁进行数据信息的搜集，会不会造成系统宕机？   
  
　　-局域网内部大范围的监控，会不会导致网络拥堵？   
  
　　-除此以外，会有其他影响吗？影响范围究竟有多大呢？   
  
　　针对上述问题，我们对比的两个产品是IBM公司的管理软件Tivoli Monitoring V5.1.2和摩卡业务服务管理（Mocha Business Service Management），简称Mocha BSM。主要针对如下三种对比指标：   
  
　　-CPU利用率   
  
　　-内存利用率   
  
　　-磁盘空间占有量   
  
　　因此，在运行环境、监控频率、监控环境完全相同的情况下，对监控的过程进行跟踪和对比，通过这组测试数据，让用户清晰地了解不同产品间的优劣，以此来解答您心中的疑问。   
  
　　**2.测试准备**   
  
　　在对两个产品进行正式对比测试前，请先了解一下两种产品各自的特点和监控的指标范围。   
  
　　**2.1 IBM的ITM产品架构**   
  
　　IBM Tivoli Monitoring V5.1.2（以下简称ITM）是IBM公司Tivoli管理软件中一个重要的组成部分，在整个Tivoli管理软件中提供基本的系统资源监控功能。   
  
　　ITM在物理结构上分为三个层次：管理服务器、管理网关和管理代理，如下图所示：

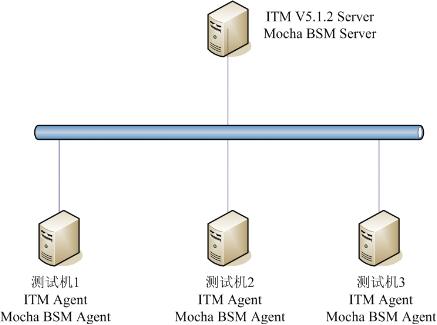


　　通过这个架构图我们可以看到，所有的内部数据流都是通过Tivoli管理代理与Tivoli管理网关通信的，也就是说通过部署Tivoli管 理代理来实现监控管理的目的，Tivoli管理代理会启动一个lcfd（Light Client Framework Daemon）的进程，因此，lcfd进程的资源利用情况是本次测试的一个重要指标。

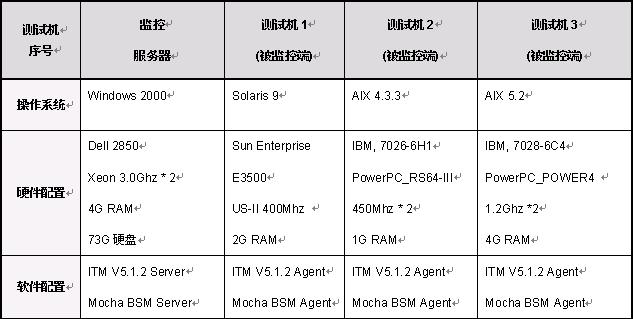
　除上面所说的管理代理，ITM还需要一个资源模型的引擎来收集被监控端资源的利用情况，这个引擎是一个基于Java的程序，因此，Java进程的资源利用情况是本次测试的另外一个重要指标。   
  
　　综上所述，在ITM的被监控主机上，ITM会启动两个进程，一个通信进程（lcfd）和一个引擎进程（Java），两个进程总体消耗的资源为我们这次测试需要收集的信息。   
  
　　**2.2 Mocha BSM产品架构**   
  
　　Mocha BSM是针对IT支持和管理部门推出的系统管理软件，主要包括系统监控、网络监控和应用监控等功能。   
  
　　BSM的架构图如下所示：



　　Mocha BSM被监控端的数据采集有两种方式，定时轮询SNMP和Agent。Mocha BSM的Agent根据功能的不同分为操作系统Agent和应用服务Agent。这次测试我们需要去收集Mocha BSM 操作系统Agent对被监控端主机的资源消耗情况，而对于应用服务Agent和网络设备的监控不在本次测试范围之内。   
  
　　Mocha BSM的Agent采用了Mocha Remote Agent Management（简称Mocha RAM）技术。Mocha RAM Daemon是一个C程序，当服务器向Mocha RAM Daemon发送请求时，请求都会经过SSH加密再传输。Mocha RAM Daemon接受请求会确认发送此请求的服务器是否在允许名单内，当确认无误后才会调用相应的插件去收集相关性能数据返回给监控服务器端。Mocha RAM提供一个插件框架，允许用户自由增加，修改或者删除插件，确保了整个插件框架的可扩展性。   
  
　　**2.3 测试环境**



　　为了保证测试的公平性，两个产品的服务器端和Agent均部署在相同的主机上，并且在测试其中一个软件的过程中停掉另一个软件的所有进程及所有无关的应用程序。下面是本次测试中选用主机的详细信息：



　　**3.测试结果对比**

　　**4.结论**   
  
　　通过以上的测试数据，我们可以看到除了在旧的AIX4.3.3这个平台上，Mocha BSM Agent的CPU利用率高于ITM以外，其它平台的测试结果都低于ITM的Agent，包括CPU、内存、磁盘等。   
  
　　在内存方面ITM Agent的利用率均高出Mocha BSM很多，这是因为ITM的资源收集引擎基于Java，而Mocha BSM的 Agent基于Mocha RAM技术，Mocha RAM的Daemon是基于C，所以在内存利用率方面优势很明显。在磁盘空间方面两个产品均比较固定，所以在部署Agent时留好相应的空间就可以了。   
  
　　这证明了Mocha BSM对监控系统的性能消耗是可以接受范围以内，而且所用的资源可以跟国际大厂家的软件媲美，甚至更好。 [http://images.enet.com.cn/end.gif](http://www.enet.com.cn/)