

软件性能测试中的性能监控方法

文 4c

在软件的性能测试整个流程当中,监控起着至关重要的作用。因为在性能测试开始执行之后,需要实时的去观察性能测试的各个指标是否正常,其中包括应用服务器、数据库、中间件等方面。一旦发现异常情况,及时修正,保证性能测试的顺利进行。而且在监控当中,也可以发现系统的瓶颈,适当制止性能测试的继续运行,保证避免重复的工作。

广义意义的性能测试监控,应该分阶段去做,其中包括执行前、执行中和执行后的监控。

执行前:环境搭建的时候,监控确定性能测试环境的纯净性,没有其他资源在使用。CPU、MEM、LOA、I/O 的初始值是否正常。

执行中:监控内容包括虚拟用户执行情况、场景状态、事务响应时间、服务器资源使用、操作系统和硬件的监控,此外最重要的还有测试机的运行情况,包括 CPU、MEM 等。是否满足当前性能测试种类的要求,比如性能测试、压力测试、负载测试等。

除了 LoadRunner 等监控工具外,也可以借助于辅助工具,用来监控一些定时服务、夜间监控情况,写一些 shell 脚本。

监控中可以分几大类去监控:工具的监控、测试用例的监控、测试方法的监控、进度、以及测试环境的监控。

建议:创建监控点列表,确定监控目标。开启监控服务,监控同时要采集信息,以便之后的分析。确定监控信息,同时确定监控工具。

执行后:监控资源释放是否正常、合理。

监控指标

性能测试的监控指标主要包括以下几个部分:

(1)服务器 Linux 应用服务器

具体包括 CPU、Memory、Load、I/O、Disk 等。

(2)数据库 Mysql、Oracle

具体包括索引、单条 SQL 性能、数据库线程数、数据池连接数等。

(3)中间件:Jboss、Apache

具体包括线程数、连接数、日志输出等。

(4)网络

具体包括防火墙、网卡、网线、吞吐量、吞吐率等。

(5)应用服务

具体包括 JVM 内存使用和回收、JAVA 内存使用、Full GC 频率、JAVA 类装入和卸载、日志、线程运行状态(阻塞、等待、正常运行)等。

(6)监控工具(LoadRunner)

具体包括用户执行情况、场景状态、事务响应时间、TPS、Load、CPU 分析图表等。

(7)测试机资源

具体包括 CPU、Memory、网络、日志输出、磁盘空间、负载生成器评估等。

监控原则

(1)确定监控目标

(2)确定监控和分析信息

(3)确定监控工具

(4)收集数据

(5)分析数据

(6)调优

(7)循环

监控方法

包括 Checklist 法等。

监控工具

包括 Profiler、Jstat、Jconsole、Jmap、Jprofiler、Nmon 等。

Java 应用程序和小应用程序

Java 程序可以是独立的应用程序(Application)和能在浏览器上执行的小应用程序(Applet)。两种 Java 程序都由一个或多个扩展名为“.class”的文件组成,都需要 Java 虚拟机(JVM)载入并翻译。

这两种程序主要区别:

1. 小应用程序只能在与 Java 兼容的容器中运行,可以嵌入在 HTML 网页内,在网络上发布,当网页被浏览时,在浏览器中运行。小应用程序的运行还受到严格的安全限制,例如它不能访问用户计算机

上的文件。Java 应用程序没有这些限制,也不支持网页嵌入和下载运行。

2. 小应用程序和应用程序在代码编写上也有很大的差异。一个小应用程序必须定义成一个 Applet 类的子类,应用程序可以是 Applet 类的子类,也可以不是。应用程序必须在一个类中定义一个 main()方法,该方法代表应用程序的入口。而小应用程序不必定义 main()方法,它的执行由 Applet 类中定义的多个方法控制。