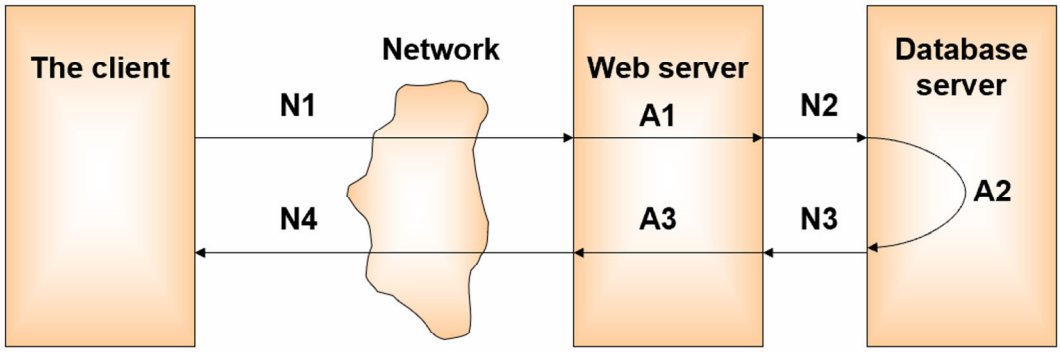
Ch9 性能测试

2018年12月21日 12:27

9.1什么是性能?

- 性能是量度系统或组件在一定约束条件下，是否达到功能设计的指标：如响应速度，计算的精度，内存利用率。
- 性能是系统外部的质量属性基于用户的需求和用户的系统操作性的看法。
- 同时性能特别应用于对实时系统的评价当中,就是它的行为在指定的期限内完成正确的操作。
- 常用性能指标
  - 时间效率，空间效率，I/O性能，数据库性能，内存性能，初始化/退出时间和资源利用率等
  - 延时，事务处理时间，最大事务处理时间，事务操作时间，数据库性能，最大消耗的内存量，高峰内存时间，资源消耗
    - 延迟Latency：
      - 一个指令控制器发出数据请求的一瞬间和数据传送的一瞬间之间的时间间隔;是请求和完成操作之间拖延的时间差
      - 延迟越短越好
    - 事务处理时间Transaction processing time：
      - 指完成一项事务所需要的运行时间，用于评价事务处理效率。通常，事务处理的时间越短，则效率越高。
    - 最大事务处理时间：Maximum transaction processing time
      - 首先需要分析哪些事务耗时较多，然后再将这些事务所花费的时间分别测试出来，花费时间**最多的**就是最大事务处理时间
    - 事务操作时间：Transaction operating time
      - 主要用来评价需要用户操作的事务处理所需要花费的时间，主要体现了用户操作方面的效率。
      - 测试事务操作时间，需要一个计时器，测试多个不同用户花费的时间，最后取平均值。
    - 数据库性能：Database performance
      - 一般包括查询、插入、删除、更新数据库等所需要的时间，由于数据库性能往往是软件系统的性能瓶颈，所以测试数据库的性能非常必要。
    - 最大消耗内存量：The largest consumption of memory
      - 标示着应该配置什么级别的硬件才能运行软件，它是硬件成本的直接反映。最大消耗的内存量测试可以通过性能监视器来进行。
    - 资源消耗：Resource consumption
      - 应用消耗的内存或磁盘空间总量。
    - 高峰内存时间：Peak time of memory

- 指软件在高峰内存消耗时期所运行的时间。如果软件在高峰使用的内存和系统的总内存比较接近，软件的效率将会大大降低。所以，缩减高峰内存时间可以提高软件性能。

- 下面介绍几种关于网络性能的指标，如：并发用户数，响应时间，吞吐量和休眠时间等。Concurrent users, Response time, Throughput and Dormancy (休眠) time
  - 响应时间Response time：一般控制在10s内
    - 响应时间是指请求做出响应所需要的时间，可分解为网络传输时间、应用延迟时间、数据库延迟时间和呈现时间等。
  - 系统响应时间System response time：
    - 应用系统从请求发出开始到客户端接收到数据所消耗的时间。
  - 呈现时间Present time：
    - 呈现时间取决于在被客户端收到数据后呈现页面所消耗的时间；
  - 如：web应用的页面响应时间的分解
    - 

    - 页面响应时间分解为网络传输时间（N1+N2+N3+N4）和应用延迟时间（A1+A2+A3），
    - 而应用延迟时间又可分为数据库延迟时间（A2）和应用服务器延迟时间（A1+A3）。
  - 并发用户数，一般分两种情况：
    - 严格意义的并发，即所有用户在同一时间做同一件事情或者操作。
    - 广义范围的并发，多个用户对系统发出了请求或进行了操作，但这些操作可以是相同的，也可以是不同的。
  - 性能测试比较关注业务并发用户数，从业务的角度设置多少并发数合理。
  - 下面给出估算并发用户数的公式：C=nL / T

    C —平均的并发用户数

    n —登录会话的数量

    L —登录会话的平均长度

    T —考察的时间段长度

    并发用户数的峰值

$$C = \frac{nL}{T}$$

- C — Concurrent users on average
- n — The number of logon session
- L — The average length of logon session
- T — Inspection period length

$$\hat{C} \approx C + 3\sqrt{C}$$

- $\hat{C}$ — The peak concurrent users

- 例题：
- 一个软件系统每天约有400个用户访问。用户在一天之内有8小时内使用该系统，从登录到退出的平均时间为4小时，请计算该系统的并发用户数和并发用户的峰值各是多少?
- 分析：根据公式C=400×4 / 8=200

$$\hat{C} = 200 + 3\sqrt{200} = 242$$

- 吞吐量：
  - 吞吐量是指在一次disposable性能测试过程中网络上传输数据量的总和。
  - 一般来说，吞吐量用请求数/秒或页面数/秒来衡量。
- 吞吐量指标有如下两个作用：

  1、协助设计性能测试场景，衡量性能测试场景**是否达到了预期的设计目标**。在设计性能测试场景时，根据估算吞吐量数据测试场景事物发生频率。

  2、协助分析性能测试**瓶颈**。

Computation formula is as follows：

$$F = \frac{N_{vu} \times R}{T}$$

- – F — Throughput
- – N_{vu} — Virtual users number
- – R — Each virtual users send requests
- – T — Time for performance testing

- 性能计数器
  - 性能计数器是描述服务器或操作系统性能的一些数据指标，具有监控和分析的作用。
- 休眠时间Dormancy time：
  - 休眠时间又称思考时间，是**指用户请求的间隔时间**。
  - 在交互式系统应用中，用户不大可能持续不断的发出请求，一般模式是用户发出一个请求，等待一段时间，再发出下一个请求。
  - 因此，自动化测试模拟用户操作就必须在测试脚本中让各个操作间隔一段时间，在操作语句之间设置休眠时间的Think函数，实现两个操作之间的等待时间。
  - 休眠时间与迭代次数、并发用户数和吞吐量之间存在一定的关系。

- 休眠时间计算过程：

$$F = \frac{N_{vu} \times R}{T} \qquad R = \frac{T}{T_s}$$

- R — Each virtual users send requests
  T_s — Thinking Time

  Throughput and N_{vu} direct proportion, and inversely proportional to T_s.

  1、首先计算出系统的并发用户数。
  2、统计出系统的平均吞吐量。
  3、统计出平均每个用户发出的请求数量。
  4、根据公式计算出思考时间。

9.2什么是性能测试?

- 性能测试是以详细的性能需求为指导对系统的性能进行评估的测试活动。
- 性能测试是在一定的条件下判定应用运行时的“行为” 和支持运行的基本环境。
- 性能测试是用来量度几种系统的特点,比如处理速度、响应时间、资源消耗、吞吐量和效率。
- 负载测试Load：
  - 负载测试是通过测试应用程序在负载作用下的表现,通过增加工作量直到承受的极限包括它的极限(不只是在它的限制)。
  - 已发现设计上的错误或者验证系统的负载能力。
  - 负载测试具体是指负载的大小(并发用户的数量）和相关的值。
  - 负载测试是模拟实际软件系统所承受的负载条件的系统负荷，通过不断加载（不断增加模拟用户数量）或其他加载方式来观察不同负载下系统响应时间和数据吞吐量，系统资源占有率（cpu和内存）等性能指标，以检验系统的行为和特性，发现系统可能存在的性能瓶颈、内存泄露和不能实现同步等问题。
  - 负载测试的加载方式：
    - 一次性加载
      - 一次性加载某个数量的用户，在预定的时间段内持续运行
      - 例如，早上上班的时间访问网站或登陆网站的时间非常集中，基本属于扁平负载模式。
    - 增加负载
      - 有规律的逐渐增加用户，每秒增加一些新用户，交错上升。借助这种负载方式的测试，容易发现性能的拐点，即性能瓶颈。
      -
    - 高突变载荷和低突变载荷
      - 高低突变加载：某个时间用户数量很大，突然降到很低，过一段时间，又突然加到很高，反复几次。
      - 借助这种负载方式的测试，容易发现资源的释放和内存泄漏的问题。

        - 随机加载方法
            - 由随机算法自动生成某个数量范围内变化的、动态的负载，这种方式可能是和实际情况最为接近的一种负载方式。
            - 虽然不容易模拟系统运行出现的瞬间高峰期，但可以模拟系统长时间的运行过程的状态。
- 压力测试Stress；
    - 压力测试用于判定应用处理大量数据的能力。
    - 压力测试可以成功的测试服务器满负载的情况。
    - 除了在服务器上增加运行的应用结合客户端测试是一种额外的形式的压力测试。
    - 压力测试的两种类型；
        - 高负荷长时间(如24小时以上)的稳定性压力测试。
        - 极限负载情况导致系统崩溃破坏压力测试。
    - 微软测试实践经验表明，如果软件产品通过72小时压力测试，则在72小时后出现问题的可能性微乎其微。所以，72小时成为微软产品压力测试的时间标志。

9.3性能测试有哪些类型?

- 最常见的三个类型的软件性能测试包括:
    - 测试由我们想量度的指标驱动。
    - 测试基于的资源或负载的类型。
    - 测试对系统施加压力或找到它的极限。
- 负载变化、反弹测试Load variation/bounce testing
    - 变化负载量度性能，反应真正负载周期性衰退和反复的典型模型。
- 校准测试Calibration(校准) testing
    - 检验系统是不是违反了强制的需求，例如政府的政策法规。
- 死锁测试Deadlock testing
    - 进行容量测试检验系统是否有资源竞争，例如多个请求访问同一数据库，导致系统停止处理用户请求，由于数据库记录发生死锁。

9.4如何进行性能测试?

- 性能测试过程：
    - 计划——记录——修改——执行——分析
- 了解系统的负载情况
- 首先，要知道我们测试所要完成的任务，通常要确定我们选择的测试类型。和我们希望运行的测试用来测试什么——哪些指标。
- 然后，确定应用系统的容量和负载的组成，即工作负载。
- 一旦评估的指标如事务处理时间，并发用户数，网络传输时间等等已经确定，下个步骤就是建立一个负载概况。
- 负载的概况是为了模拟真实的环境，而创建的。不但包括评估的负载大小，还包括系统的用户和构成负载的活动的概述需要被创建。
- 第一步：计划：

    - 测试样本的大小
        - 大多数我们想要量度的指标，例如响应时间，都是基于各种各样因素的影响，这些因素有知道的也有不了解的。
        - 这就意味着我们不能只量度一次，而是取样进行多次测试得到有意义的平均值。
        - 多大的样本是我们所需要的，有一个粗略的规则可以平衡样本大小和精度的关系。
    - 用户的数量和类型
        - 用户的组合在不同时期会有变化
        - 特殊的时间会触发组合的改变。
        - 网站用户的数量有下面一些因素影响：
            - 网站类型，所提供的内容，季节性的事件，特殊事件
    - 并发性
        - 划分时间粒度，应该较小，解决操作存在时间性集中的问题。
        - 注重业务模式的特殊性。
    - 思考时间
    - 负荷背景或噪声背景Background Load (Noise)
        - 指哪些增加的活动被认为是直接导致性能下降的网络或其它应用的负载——噪声背景
    - 需要测试的类型或度量
        - 1，负载
        - 2，性能
        - 3，压力
- 第二到第四步，记录，修改，执行
    - 首先模拟实际用户的操作行为，记录和回放多用户测试中事务处理的过程，自动生成测试脚本。
    - 其次能对脚本进行修改，增加逻辑控制，完成参数化和数据关联。
    - 参数化：使用一些列不同的值（样本）来执行相同的脚本，匹配实际用户的操作行为，反映出系统真正的负载能力。
    - 需要用参数来替换已刻录的值，即脚本参数化。
- 一个性能测试工具抽象的结构包括：控制器，负载生成器，性能监视器，分析器，虚拟用户生成器
- 检验器是一种性能监测器提供代码水平的量度,包括时间、内存的使用等等。

  当我们要定位软件系统性能的瓶颈的时候检验器是有用的。
- 虚拟用户生成器（Vuser Generator）用于捕获最终用户业务流程和创建测试脚本，即通过录制应用程序中典型最终用户执行的操作录制到自动虚拟用户（Vuser）脚本中，以便作为负载测试的基础。虚拟用户包括
- 控制器——控制其用于组织、驱动、管理和监控负载测试的中央控制。

  使用控制器可以运行用来模拟真实用户执行的操作的脚本，并可以通过让多个Vuser同时执行这些操作来在系统中创建负载。
- 负载生成器运行虚拟用户，已产生有效的、可监控的负载
- 分析器帮助查看、分析和比较性能结果，使用这些图和报告，可以标识和确定应用程序中的瓶颈，并确定需要对系统进行那些更改来提高系统性能