|  |  |
| --- | --- |
|  |  |
| **性能测试技术的研究\_关于性能测试业务场景设计的研究**    关键字：性能测试   业务场景   极限  超载    摘要：性能测试是指在一定硬件条件下，获取软件系统在不同的业务背景下的各种性能表现，本文根据笔者最近所做的几次性能测试，就业务场景设计方面进行总结分析，希望能起到一定的借鉴作用。    **1 测试过程中出现的问题**         最近一段时间，我们的团队连续承担了几个基于J2ee架构的分布式系统的测试工作。在测试过程中，我们多次发现一个问题，就是我们在测试环境中得到的性能指标与生产环境中的性能指标差距较大，理论上应该是测试环境的性能指标优于生产环境的指标，但实际结果是生产环境的性能指标大大优于测试环境下的性能指标。经过不断摸索、总结，我们发现出现以上问题的原因是测试需求分析不到位，导致业务场景设置不合理、不全面，因而出现问题。在这里我们就场景设计方面谈一些自己的看法。  **2业务场景分析**         性能测试中涉及的基本场景有两种，即单一业务场景和混合业务场景，这两种业务场景缺一不可，缺少任何一种都不能准确评估系统性能，定位系统瓶颈。如果只做单一业务场景，得到的结果与实际生产环境差距较大，没有实际指导意义；如果只做混合业务场景，不能快速定位系统性能快速降低的原因，起不到定位瓶颈、系统调优的作用。只有两种场景互为补充，才可以获取最符合客户要求的测试结果。下面分别就两种测试场景的具体设计方法结合一个论坛系统进行讨论，一个论坛系统包含三个主要单一业务流程，即用户登陆、发表文章、阅读文章。该论坛支持100人同时在线，支持20人同时发表文章，阅读文章。    **3单一业务场景**         单一业务场景主要针对单一业务流程而设计，主要考察某一项单一业务在各种情况下的响应时间，系统资源占用，事务成功率等指标。对于响应时间这个指标，目前国内国际上还没有明确的标准，业界普遍采用的评价准则是2/5/10标准，即2秒以内优秀，5秒以内可以接受，10秒是极限。但是在实际当中，并不仅限与这个标准，例如邮件系统的登录功能可以遵守这个标准，因为只是一个登录功能；但是针对上传附件这个功能，如果附件本身较大，响应时间自然会比较长，所以我们应该灵活掌握标准，以实际需要为准则，确定预期响应时间。事务成功率这个指标，业界采取的一般标准是90%，但是对于一些精密程度要求较高的系统，如金融、证券、银行等系统，事务成功率要求更高，应该达到98%以上，部分功能甚至要求达到100%。下面我们就单一业务场景在标准、极限、超载三种情况下的设计进行讨论。    **3.1标准场景**         设计标准场景的目的是验证系统单一业务是否达到所承诺的性能指标，此时系统应该表现良好，响应时间短，事务成功率高，资源利用率合理，如果不能达到以上要求，则说明系统存在瓶颈，应进一步确定瓶颈，并进行系统调优。但是在实际测试过程中，我们往往片面的认为标准场景就是标准用户数，而忽略了操作对象、操作频率也是非常重要的场景内容，如果这样我们设计的操作场景将会如下（以发表文章为例）：   |  |  |  |  | | --- | --- | --- | --- | | 业务名称 | 虚拟用户数 | 加压方法 | 持续时间 | | 发表文章 | 10 | 初始用户为0，每隔10秒加载2个用户，全部用户加载之后，持续运行15分钟，再以每隔10秒卸载2个用户，直至结束。 | 15分钟 |     这个场景初看起来没有任何问题，但是我们如果更深入的考虑一下，就会发现一些问题。例如，发表文章，文章的大小对系统的压力是否有影响呢？发表一个1K的文章，和发表一个1M的文章是否有区别呢？一个用户是否会持续不断的发文章呢？如果不是，发文章的频率是多少呢？这需要我们进一步在测试需求中明确，这样设计出来的场景才是真正的标准情况下的测试场景，新的测试场景如下：   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 业务名称 | 虚拟用户数 | 迭代时间 | 操作对象 | 加压方法 | 持续时间 | | 发表文章 | 10 | 120 | 50K | 初始用户为0，每隔10秒加载2个用户，全部用户加载之后，持续运行15分钟，再以每隔10秒卸载2个用户，直至结束。 | 15分钟 |   新的测试场景增加了标准的迭代时间和标准大小的操作对象，这样的测试场景才是真正的标准测试场景，获得的各项性能指标才具有实际指导作用。    **3.2极限场景**         设计极限场景的目的是为了验证系统单一业务否达到承诺的极限情况，在极限的情况下，能否正确完成相应的工作，并保证客户数据安全，响应时间在可接受范围内，资源利用不超标。根据标准场景的分析，我们可以指导极限场景的设计应包括以下部分，即虚拟用户极限，并发用户极限，迭代时间极限，操作对象极限；持续时间极限属于疲劳强度测试，这里不进行讨论。得到的极限测试场景如下：   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 业务名称 | 虚拟用户 | 并发用户 | 迭代时间 | 操作对象 | 加压方法 | 持续时间 | | 发表文章 | 20 | 2 | 120 | 50K | 初始用户为0，每隔10秒加载2个用户，全部用户加载之后，持续运行15分钟，再以每隔10秒卸载2个用户，直至结束 | 15分钟 | | 发表文章 | 10 | 4 | 120 | 50K | | 发表文章 | 10 | 2 | 60 | 50K | | 发表文章 | 10 | 2 | 120 | 1M | | 发表文章 | 20 | 4 | 120 | 50K | | 发表文章 | 20 | 2 | 60 | 50K | | 发表文章 | 20 | 2 | 120 | 1M | | 发表文章 | 10 | 4 | 60 | 50K | | 发表文章 | 10 | 4 | 120 | 1M | | 发表文章 | 10 | 2 | 60 | 1M | | 发表文章 | 20 | 4 | 60 | 50K | | 发表文章 | 20 | 10 | 120 | 1M | | 发表文章 | 20 | 2 | 0 | 1M | | 发表文章 | 10 | 10 | 0 | 1M | | 发表文章 | 20 | 10 | 0 | 1M |   以上得到共计15种极限情况，由于实际测试过程中资源有限，不可能进行如此全面的测试，可以根据实际情况进行取舍，以用户要求为主导，选择需要的测试场景进行测试。在加压过程中，如果响应时间明显加长，资源利用率异常上升，吞吐量没有随用户增加正比增长则说明系统已到达瓶颈，可以停止测试，转而进行瓶颈确认、系统调优等工作。    **3.3超载场景**         设计超载情况测试场景的目的是为了验证系统单一业务在超载的情况下，何时出现性能拐点，何时系统失效，失效对数据库已有数据是否有影响。根据标准场景的分析，我们可以指导超载场景的设计应包括以下部分，即虚拟用户超载，并发用户超载，迭代时间超载，操作对象超载。得到的超载测试场景如下：   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 业务名称 | 虚拟用户 | 并发用户 | 迭代时间 | 操作对象 | 加压方法 | 持续时间 | | 发表文章 | 30 | 2 | 120 | 50K | 初始用户为0，每隔10秒加载2个用户，全部用户加载之后，持续运行15分钟，再以每隔10秒卸载2个用户，直至结束 | 15分钟 | | 发表文章 | 10 | 6 | 120 | 50K | | 发表文章 | 10 | 2 | 0 | 50K | | 发表文章 | 10 | 2 | 120 | 2M | | 发表文章 | 30 | 6 | 120 | 50K | | 发表文章 | 30 | 2 | 0 | 50K | | 发表文章 | 30 | 2 | 120 | 2M | | 发表文章 | 10 | 6 | 0 | 50K | | 发表文章 | 10 | 6 | 120 | 2M | | 发表文章 | 10 | 2 | 0 | 2M | | 发表文章 | 30 | 6 | 0 | 50K | | 发表文章 | 30 | 6 | 120 | 2M | | 发表文章 | 30 | 2 | 0 | 2M | | 发表文章 | 10 | 6 | 0 | 2M | | 发表文章 | 30 | 6 | 0 | 2M |   以上得到共计15种超载情况，由于实际测试过程中资源有限，不可能进行如此全面的测试，可以根据实际情况进行取舍，以用户要求为主导，选择需要的超载测试场景进行测试。在加压过程中，如果响应时间明显加长，资源利用率异常上升，吞吐量没有随用户增加正比增长则说明系统已出现拐点；如果出现响应时间超长，资源利用率长期超标，吞吐量逆向减少，说明应用系统已失效，可以停止测试。    **4混合业务场景**         混合业务场景针对模拟系统真实生产环境而设计，主要为了测试整个系统在各种情况下的响应时间，系统资源占用，事务成功率等指标。下面我们就混合业务场景在标准、极限、超载三种情况下的设计进行讨论。    **4.1标准场景**         标准情况下的混合场景是测试场景中最贴近实际运行环境的场景，可以全面有效的反应被测系统在真实环境下的性能表现，验证系统是否达到所承诺的各项性能指标，并对未来系统扩展、调优提供支持。我们可以得到如下的测试场景：   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 业务名称 | 虚拟用户 | 并发用户 | 迭代时间 | 操作对象 | 加压方法 | 持续时间 | | 用户登录 | 20 | 5 | 120 |  | 初始用户为0，每隔10秒加载2个用户，全部用户加载之后，持续运行15分钟，再以每隔10秒卸载2个用户，直至结束 | 15分钟 | | 发表文章 | 10 | 2 | 120 | 50K | | 阅读文章 | 40 | 10 | 60 | 不同对象（多个用户同时阅读同一篇文章和不同的文章，会对系统产生不同的压力） |   测试场景中各个业务所分配的用户比例是按照用户要求设计的，有时用户不能明确自己的测试需求，需要测试人员使用日志分析工具分析系统在实际运行情况下的日志得出相关数据，指导测试场景的设计。    **4.2极限场景**         设计极限场景的目的是为了验证系统在贴近真实环境下能否达到承诺的极限情况，在极限的情况下，能否正确完成相应的工作，并保证客户数据安全，响应时间在可接受范围内，资源利用不超标。通过以上单一业务场景的极限情况分析，我们知道会有极限场景共15个；而混合业务又包含多个单一业务，这个例子中包含三个单一业务，那么极限测试场景会有4125个，这个数量显然是我们无法接受的，我们只能从中挑选重要的极限场景或用户指定的极限场景，并配合正交排列方法选择。我们选择如下的混合业务极限测试场景：   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 序号 | 业务名称 | 虚拟用户 | 并发用户 | 迭代时间 | 操作对象 | 加压方法 | 持续时间 | | 场景1 | 用户登录 | 50 | 5 | 120 |  | 初始用户为0，每隔10秒加载2个用户，全部用户加载之后，持续运行15分钟，再以每隔10秒卸载2个用户，直至结束 | 15分钟 | | 发表文章 | 10 | 4 | 120 | 50K | | 阅读文章 | 40 | 10 | 30 | 不同 | | 场景2 | 用户登录 | 20 | 10 | 120 |  | | 发表文章 | 20 | 2 | 120 | 50K | | 阅读文章 | 40 | 10 | 120 | 20相同 | | 场景3 | 用户登录 | 20 | 5 | 60 |  | | 发表文章 | 10 | 2 | 120 | 1M | | 阅读文章 | 70 | 10 | 60 | 不同 | | 场景4 | 用户登录 | 50 | 5 | 60 |  | | 发表文章 | 10 | 2 | 60 | 50K | | 阅读文章 | 40 | 20 | 60 | 不同 | | 场景5 | 用户登录 | 20 | 10 | 60 |  | | 发表文章 | 10 | 4 | 60 | 50K | | 阅读文章 | 40 | 20 | 30 | 不同 | | 场景6 | 用户登录 | 50 | 10 | 60 |  | | 发表文章 | 10 | 4 | 60 | 1M | | 阅读文章 | 40 | 20 | 30 | 相同 |   当然，以上极限测试场景未必是最合理的，因为以区区六种测试场景取代4125种测试场景肯定不能达到全面体现系统性能的目的，只能是在一定程度上对系统性能进行度量。  **4.3超载场景**         设计超载情况测试场景的目的是为了验证整个系统在超载的情况下，何时出现性能拐点，何时系统失效，失效对数据库已有数据是否有影响等。通过以上缓和业务场景极限情况的分析，我们同样可以得出超载场景的数量也是4125个，同样根据用户测试需求、系统自身特点等约束条件，我们进行筛选，得到需要的超载场景，这里就不展开描述了。混合业务流程的超载测试场景的结果对于系统调优的指导意义是比较大的，因为系统在这种场景下失效的可能性是最大的，系统的弱点暴露的也最完全。    **5结语**         性能测试中的场景设计是实施性能测试的基础，只有合理的设计测试场景才能获得有价值的测试数据，为接下里的确认瓶颈、系统调优打下基础。以上只是我们团队对场景设计一点粗浅的认识，拿出来与大家分享，希望得到各界人士的批评指正。 | |