

【软考达人】

# 软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



**微信扫一扫，立马获取**



**6W+ 免费题库**



**免费备考资料**

PC版题库: [ruankaodaren.com](http://ruankaodaren.com)

## 论 Web 系统的测试技术及其应用

### 【摘要】

笔者于 2011 年 8 月参与了某地级市的市级机关电子政务信息系统的建设工作，该项目是该项目机关的电子政务网建设计划的一部分，笔者在该项目中担任项目经理和系统分析师一职，主要负责项目的日常全面管理和质量保证与质量控制工作。该项目是基于 WEB 系统的，由于 WEB 系统具有与传统信息系统截然不同的特点，因此需要采用针对 WEB 系统特点的测试技术与方法。在该项目中，我们主要针对 WEB 系统的负载均衡、响应能力、压力测试等方面进行测试，以测试该系统的性能、可用性等技术指标。在实际的测试工作当中，负载均衡的测试工作主要基于多并发访问模拟技术，响应时间测试采用了记录最低及最高响应时间，并建立响应时间记录的方法；压力测试采用了并发访问模拟与 24 小时不间断自动测试方法结合的技术方法。上述方法基本实现了测试的目标和目的，保障了项目的正常上线，然而在具体的测试工作当中，也遇到了一些问题。

### 【正文】

笔者于 2011 年 8 月参加了某地级市的市级机关电子政务信息系统项目的建设，该项目是该项目机关的电子政务网建设计划的一部分，在整体上分为办公自动化系统模块、政企公共业务办理模块、公共政务信息查询模块三大子模块，系统的基本架构是基于 WEB 的，主要采用 .net 进行开发，笔者在该项目中担任项目经理和系统分析师一职，主要负责项目的日常管理和质量保证与控制工作。该机关原有的政务信息系统是基于 C/S 架构的，该系统由于开发年头较长，开发技术较为落后，很多功能与性能已经不能够满足该机关的业务需求，如该系统由于是 C/S 架构的，其业务逻辑层与持久存储层位于同一台服务器上，这造成了业务量一旦加大，就会导致系统的负荷大大加重，造成整个系统的响应时间变慢，服务器性能变差等后果，从而严重影响了该市级机关的办公效率。综上所述，我们再深入调研和分析之后，认为由于 C/S 架构本身固有的缺陷，使系统的负载均衡、响应能力及系统性能无法得到很好的体现和保证，因此我们决定采用基于 WEB 系统的三层分布式架构进行该电子政务系统的建设，并在后期的项目测试工作中，着重于对系统的负载均衡能力、业务处理的实时响应能力、系统的最高承受力（压力测试）三方面进行测试工作。同传统的系统测试工作不同，WEB 系统的测试工作除单元测试、集成测试、系统测试等传统型测试工作之外，还

需对系统进行基于 WEB 特点的测试工作，其中最重要的有负载均衡测试、响应能力测试、系统压力测试三项测试工作。在负载均衡测试中，我们采用了基于多并发访问请求模拟的技术，主要测试系统的负载均衡能力，具体方面主要有测试基于 DNS 转发的负载均衡和基于指令跳转的负载均衡是否工作正常，实现负载均衡的算法是否合理，在基于 DNS 转发的负载均衡能力测试中，我们采用在 DNS 服务器上安装监控系统软件和记录软件的方法，对每一条并发访问（模拟）请求的转发情况和去向进行监控，并记录到日志文件中以供分析。在基于指令跳转的负载均衡测试中，我们采用了在系统代码中加入跳转记录指令，同时在业务服务器上安装并发请求记录软件的方法进行测试，当一条并发请求到达时，如系统判断当前业务服务器负载较重，须将该请求转发至其他负载较轻的服务器上时，跳转记录指令会将该次转发记录至日志文件，同时接收到该次转发的服务器也会通过并发请求记录软件将该次处理记录在并发请求处理日志



中。通过对跳转记录日志文件和并发请求处理日志文件的分析，我们可以得出基于指令跳转的负载均衡测试的测试结果。

在上述两种负载均衡测试中，我们均需要对负载均衡的实现算法进行测试和分析，主要的目的是分析该实现算法能否合理地安排并发请求的跳转，能否根据当前业务系统的负载情况合理地实现请求跳转和分流，以实现系统的整体上的负载均衡。具体的实现方法是根据跳转日志记录文件分析并发请求实现跳转时的当前业务服务器的负载情况，并发请求的跳转时机是否合理，跳转请求的受理服务器的负载是否在一个合理范围内等。经过对上述的数据及信息进行归纳和分析后，即可测试出该实现算法是否合理。在对系统响应能力的测试当中，我们采用了满负荷测试和低负荷测试两种测试方式，其中满负荷测试即在系统处于高负荷的状态下对系统进行响应能力测试，我们采用的具体方法是通过在服务器上运行专门的系统负荷测试软件，将系统的运行负荷调整到高负荷运行的状态，之后我们对系统进行响应测试，特别要注意的是：由于WEB系统是基于网络环境的，因此在进行高负荷测试时，我们不仅需要模拟系统的高负荷运行状态，还需利用网络通信负荷测试软件模拟网络环境的高负荷状态。通过对系统和网络环境高负荷状态进行响应测试，我们得到了系统在恶劣环境下的最坏响应结果，为下一步进行分析和系统调优工作提供了一句。低负荷测试的主要目的是测试系统在正常环境下（即正常使用环境）的并发请求响应能力，在这项测试中，我们对业务服务器做了优化，关闭了一切不必要的内存常驻程序，设置了合理的缓存空间，并调整了网络环境使其达到了一个理想状态。在此基础上我们进行了低负荷响应能力测试并得到了相应的测试结果。通过对高负荷响应测试和低负荷响应测试结果的分析，我们最终得到了系统的最坏和理想响应能力的结果数据，为对系统进一步的调整和优化提供了数据。同时，在测试时我们还进行了高系统负荷、低网络负荷和低系统负荷、高网络负荷两种测试，并将结果合并至响应测试结果中，限于篇幅，本文对这两种测试方式不再陈述。

压力测试是对上述两种测试的综合测试，主要目的是测试系统在长时间、高负荷状态下的性能曲线，为系统的整体调整和优化提供一个整体性的参考依据。在这项测试中，我们通过将系统和网络调整至高负荷状态后，24小时不停机运行自动测试脚本的方式来进行系统压力测试，并且我们采用以固定时间为间隔建立记录点的方式，以汇总记录每个固定时间范围内的系统状况。通过系统压力测试，我们最终得到了系统压力测试日志，并将其输入至相关的日志分析软件，以得到系统负荷分布曲线。我们将负载均衡测试结果、响应能力测试结果、压力测试结果合并至测试报告中，并将该测试报告上报至项目组，项目组召开专门会议对此进行审核和研究，并最终通过该报告，随后我们以该测试报告为基础，对系统展开调整和优化工作。经过数轮调整和优化工作的迭代，测试工作正式结束。

该项目上线之后，表现出色，基本实现了用户的需求。在项目的总结会议上，我们对测试工作进行了总结：该测试工作总体上是成功的，但也存在两个问题：一是在压力测试中，由于所采用的固定时间间隔的范围没有一个明确的标准，这给日志分析软件所生成的系统负荷分布曲线带来了一定的误差；二是在负载均衡测试中，对负载均衡算法的测试结论是该算法性能是良好的，但在实际使用当中，发现该算法并不能100%保证系统的负载均衡，在个别情况下会导致负载失衡，严重时则须重启系统。对于第一个问题，笔者认为应当建立明

确的时间间隔标准，以减少系统负荷分布曲线的误差；对于第二个问题，笔者认为除了改进负载均衡算法外，还应具体分析实际的应用环境，找出造成负载失衡的各种可能因素。总结：基于WEB系统的政务信息系统的开发工作中，针对WEB系统的特征的测试工作是整个项目测试工作的重点工作，笔者认为，做好WEB系统的测试工作，是整个项目成败的关键因素之一，若测试工作做的出色，则可以有效地提高系统的性能、可靠性和可用性。