

# 51Testing18期沙龙主题 性能测试与LoadRunner



<http://www.51testing.com>

上海博为峰软件技术有限公司

# 提纲

- 第一章 如何启动性能测试项目
- 第二章 如何建立性能测试模型
- 第三章 如何制定性能测试计划
- 第四章 如何准备性能测试
- 第五章 如何实施性能测试
- 第六章 如何分析测试结果

# 第一章 如何启动性能测试项目



- 1节 – 被测系统预研
- 2节 – 项目经理访谈
- 3节 – 业务经理访谈
- 4节 – 技术经理访谈

# 第一节 - 被测系统预研

- 简介
  - 根据被测系统的资料初步了解被测系统的任意方面的信息。
- 角色
  - 性能测试分析师
- 目标
  - 尽可能多了解被测系统的相关知识，可以包括：系统的建设目的、系统的技术架构、系统的业务架构
- 内容
  - 确定被测系统的软件开发组织和组织的负责人，例如项目经理向项目经理提出获取被测系统所有相关资料的申请，可以包括：软件需求规格说明书、软件概要设计说明书、软件详细设计说明书、软件操作手册、软件维护手册等。
  - 在索取这些资料的过程中，常见由于保密原则而项目经理不同意提供资料、或者项目开发过程不规范而项目经理无法提供有效的资料等类似情况。

## 第2节 – 项目经理访谈

- 简介
  - 获取性能测试实施工作的开展信息，例如开始/截止日期，测试环境等
- 角色
  - 性能测试项目经理
- 目标
  - 从项目经理处获得被测系统的开发状态和期望的性能测试工作的目标
- 内容
  - 向项目经理申请获得准确的性能测试实施开始时间
  - 向项目经理申请获得准确的性能测试实施截止日期
  - 向项目经理了解被测系统的生命周期当前处于哪个阶段，例如集成测试阶段、系统测试阶段、验收测试阶段、上线阶段、运营维护阶段等

向项目经理了解性能测试工作目标，即期望获得什么样的性能测试成果，应包括：性能测试的性质（调优性质的性能测试、验收性质的性能测试等）

向项目经理了解被测系统的生产环境部署方案，包括网络部署内容、服务器硬件部署内容、操作系统部署内容、安全部署内容、应用平台部署内容、数据库中的数据量规模，尤其要获得是否将在同一个服务器上存在其他应用系统的部署内容。

向项目经理了解性能测试环境信息，应包括：硬件到位日期、软件安装日期、硬件级别（是否与生产环境的硬件级别相同）、软件级别（是否与生产环境的软件版本相同）、部署方式（是否与生产环境的部署方式相同），尤其要注意测试环境中的数据量规模是否能够达到生产环境的级别（例如生产环境在上线时将具备**500**万注册用户的信息，是否在测试环境中能够拥有如此规模的数据量，并且这些数据是否符合实际情况）

向项目经理了解性能测试的性能指标，应包括：是否已经确定了性能测试的指标；性能指标是否合理；如果性能指标不合理，是否由性能测试实施人员辅助重新进行确定

# 第3节 - 业务经理访谈

- 简介
  - 获取性能测试业务模型的设计依据，例如被测系统的关键业务、主要用户场景、用户场景交易发生概率、期望响应时间等
  - 从业务经理处申请调派合适的业务人员，作为将来建立性能测试模型的业务支持人员
- 角色
  - 性能测试分析师
- 目标
  - 从业务经理处获得被测系统的主要用户场景和关键业务
  - 从业务经理处获得被测系统的关键业务性能指标
  - 从业务经理处获得合适的业务支持人员
- 内容



- 向业务经理讲解获取性能测试所需关键业务的方法
- 从业务经理处获得被测系统的关键业务，选取的原则应包括：
  - 发生频率非常高的（例如：移动核心业务系统中的缴费业务，它在移动每天的业务总量中占到90%以上）
  - 关键程度非常高的（业务经理认为绝对不能出现问题的，例如银行业务中的存款业务、取款业务、转账业务等与金钱相关的）
  - 资源占用非常严重的（导致磁盘I/O非常大的，例如某个业务进行结果提交时需要向数十个表存取数据，或者一个查询提交请求时会检索出大量的数据记录）

- 向业务经理讲解获取性能测试所需主要用户场景的方法
- 从业务经理处获得被测系统的主要用户场景，选取的原则应包括：

以一年作为被测系统主要用户场景的衡量周期

在一年中将会有多少种非常大量的用户在线行为

例如移动公司发布新的优惠信息后，大量用户集中到营业厅办理手续的事件；

例如税务局每到每个月的9号到11号之间，大量的用户集中访问网上报税系统进行企业报税事件；

例如某个企业每到20号左右，大量员工访问内部财务系统进行报销操作的事件；

例如银行的营业厅每天在上午的10点和下午的15点都会迎来业务受理高峰的事件；

- 向业务经理讲解获取用户场景的交易发生概率的方法
- 从业务经理处获得业务发生概率，例如  
在移动公司的每日高峰用户场景中，**90%**的用户进行缴费业务、**5%**的用户进行开户业务、**2%**的用户进行话费查询业务、**2%**的用户进行换卡业务、**1%**的用户进行停机/开机业务

- 向业务经理讲解确定期望业务响应时间的方法
- 从业务经理处获得期望业务响应时间，例如  
1000用户在线的每日高峰用户场景中，缴费业务的期望响应时间最大不能超过3秒，开户业务的期望响应时间最大不能超过5秒，查询业务的期望响应时间最大不能超过12秒
- 向业务经理讲解性能测试建模过程中需要业务支持人员的原因，并向业务经理提出申请，调派一名或多名业务人员在将来支持性能测试案例的设计工作、业务数据的准备工作。

# 第4节 – 技术经理访谈

- 简介
  - 获取关键业务的技术路径，获取性能测试业务模型的补充设计依据
  - 根据由业务经理处获得的关键业务，向技术经理了解这些关键业务使用的数据库中的数据表
  - 向技术经理申请合适的开发人员，作为将来开发性能测试脚本的技术支持人员
- 角色
  - 性能测试分析师
- 目标
  - 完善被测系统的用户场景和关键业务
  - 从技术经理处获得合适的技术支持人员内容
- 内容

- 将主要用户场景和关键业务提供给技术经理，请技术经理判断这些关键业务是否覆盖了被测系统的所有业务请求处理节点
- （例如某个应用系统是三层架构，WEB-APP-DB，其中APP层包括两个中间件服务器，每个服务器处理不同的业务请求，此时要确定是否每个服务器都被关键业务所覆盖）
- 如果存在某个业务请求处理节点没有被业务经理分析并确定的关键业务所覆盖，则由性能测试分析师向业务经理转告这个情况，确认是否增加性能测试所需测试的用户场景和关键业务。

向技术经理讲解为什么需要了解关键业务使用了哪些数据库中的数据表

例如性能测试分析师在数据库中查找符合测试条件的数据时，根据这些关键业务和数据表之间的关系就可以设计相应的SQL语句，用该SQL语句获得关键业务执行所需要的测试数据。

也可以由此了解到哪些数据库中的数据表在测试中将被用到（例如Select、Insert等），并且判断哪些数据表需要存在多少条记录。

有技术经理处获得关键业务使用了哪些数据库中的数据表，并将这个关系填入到《性能测试技术调研表》中。

向技术经理讲解为什么需要技术人员支持性能测试脚本开发的工作由技术经理调派合适的开发人员，作为将来开发性能测试脚本的技术支持人员

例如某个被测系统在提交业务请求时将对业务请求数据进行加密，这时需要技术支持人员向性能测试脚本开发人员提供加密程序，以供性能测试脚本开发人员在性能测试脚本中将加密程序作为动态链接库程序进行调用，使得性能测试脚本能成功模仿实际业务请求。

例如性能测试脚本需要从数据库中获得大量的业务数据进行性能测试的执行，此时需要技术人员指导性能测试脚本开发人员如何从数据库的相关数据表中提取符合测试需要的业务数据。

又例如性能测试在执行过整轮的测试后，在执行下一轮性能测试之前需要恢复数据库中的数据，此时需要技术支持人员辅助进行数据库的备份和恢复工作。

例如性能测试工程师在设计性能测试监控模型时，需要技术支持人员辅助实施被测系统的监控配置工作。



## 第二章 如何建立性能测试模型

---



- 1节 – 建立业务模型
- 2节 – 建立数据模型

# 第1节 – 建立业务模型

- 简介
  - 建立某个被测系统性能测试模型中的业务模型部分，例如系统的性能指标定义、主要用户场景、关键业务等
- 角色
  - 性能测试分析师
- 目标
  - 将性能测试启动阶段的业务调研工作中，由业务经理、技术经理处获得的所有关于被测系统的业务信息，转化为性能测试的业务需求，例如由性能测试分析师将用户场景转化为性能测试场景，将关键业务转化为性能测试案例
- 内容

用户场景转化为性能测试场景。性能测试场景包括几个最关键的要素

场景模式：在线场景（所有虚拟用户具有和实际情况相符的ThinkTime）、并发场景（所有虚拟用户的ThinkTime等于零）

虚拟用户总量：场景中所有虚拟用户的总量

虚拟用户概率：每个虚拟用户的数量所占虚拟用户总量的百分比

负载发生器数量：负载发生器和控制器共用一台PC；每50用户使用一个负载发生器PC；

启动模式：每秒钟启动1个虚拟用户或者5个虚拟用户

稳定运行时间：15分钟（每次虚拟用户启动是要耗费一定的时间的，例如共有60个虚拟用户，如果设置每2秒钟启动一个用户，将耗费2分钟的时间用于启动完所有的虚拟用户，而测试结束后停止所有的虚拟用户需要1分钟。若设定所有虚拟用户的稳定时间为15分钟，则这个场景的最终总执行时间为 $15+2+1=18$ 分钟）

运行日志：收集基本的响应时间数据

结束模式：每秒钟停止1个虚拟用户或者5个虚拟用户

关键业务转化为性能测试案例。性能测试案例是开发性能测试脚本的依据（设计文档）

**操作步骤：**关键业务的每个实际业务操作步骤，例如输入用户名，输入密码，点击登陆按钮，确认主界面显示成功，点击查询按钮，确认进入注册用户信息查询界面，输入查询条件（注册用户的用户号码），点击提交按钮，确认查询结果显示成功。

**参数化点：**确定某个操作步骤中需要将输入数据进行参数化，例如在查询步骤中将输入的注册用户号码进行参数化，即将注册用户号码用变量代替，并设置该变量的数据源，例如某个数据文件（`user_No.xls`），并注明注册用户号码的数据如何获得，例如从数据库中查找10000个符合查询条件的用户号码，放置在数据文件中；或者使用EXCEL自身的数据处理功能按照一定的规则条件生成10000个用户号码。

**事务衡量点：**事务是用来衡量某个操作步骤执行时间的函数组合，包括一个起始时间函数和一个截止时间函数，由它们可以计算一个业务操作所耗费的时间长度，例如一个查询操作，在点击查询提交按钮之前设置事务的起始时间函数，在查询结果显示成功之后设置事务的截止时间函数，这样在测试执行时就可以实时计算每个查询步骤的响应时间是多长。

**Thinktime：**思考时间是用来模拟实际情况中一个操作员在两个相邻的操作动作之间的习惯性间隔时间，例如阅读页面中的内容、填写页面中要求输入的数据、起身到打印机旁等待并拿回打印好的发票等。

**检查点：**检查点的作用是验证某个业务的请求被发送到服务器后，服务器返回的结果是否正确。例如登陆的检查点可以设置为在点击登陆按钮之后检查服务器返回的页面中是否包含了“欢迎您访问xxx系统！”的字

## 第2节 – 建立数据模型

- 简介
  - 建立被测系统数据库中的基础数据模型，例如数据库中关键表-用户信息表中应该存在200万条记录，账户信息表中应该存在600万条记录，客户信息表中应该存在100万条记录，业务操作记录表中应该存在1000万条记录.....
  - 建立被测系统数据库中的测试数据模型，例如关键业务-“缴费”在性能测试过程中需要经过负载测试、压力测试、容量测试、稳定性测试等过程，而“开户”业务需要从数据库中获得符合缴费条件的用户识别号码，并且总共预计需要至少50000个不同的用户识别号码。
- 角色
  - 性能测试分析师
- 目标
  - 建立正确的基础数据模型
  - 建立正确的测试数据模型
- 内容

## 建立基础数据模型

例如根据性能测试任务的目的（例如上线性能模拟测试），判断被测系统数据库中的数据规模应该等同于上线时生产环境中数据库中的数据规模。则根据与关键业务相关数据表的关系判断这些数据表中应该存在多少条数据记录。

例如，与关键业务“缴费”相关的“客户信息记录表”、“用户信息记录表”和“账户信息记录表”中应该存在100万、200万和600万条记录。

## 建立测试数据模型

性能测试案例是根据关键业务的操作步骤和所需的测试数据由性能测试工程师创建的性能测试脚本设计依据，在性能测试案例中描述了哪些操作步骤中需要什么样的测试数据，有的测试数据是需要从数据库中获取的，有的测试数据是需要从数据文件中按照一定的规则生成的

例如登陆这个关键业务中需要输入用户名和密码，而用户名和密码都是在数据库中已经存在的用户信息，我们必须清楚如何在数据库中找到合适的用户名和密码，因为不是所有的用户名都是可以正常登陆系统的，比如一个已经被停止使用的用户名就不能被选为性能测试案例所需要的测试数据。

又例如注册这个关键业务中需要输入用户的名字、昵称、身份证号码等，这些测试数据由于是新用户注册，则在数据库中是不存在的，因此我们需要在数据文件中按照一定的规则（用户名为26个字母与阿拉伯数字组合成的6位定长度的字符串，身份证号码为18位唯一的数字串）生成这些数据。

# 第3节 – 建立测试模型

- 简介
  - 性能测试的测试模型是为了确定一次完整的性能测试中要包含哪些测试内容，例如负载测试、压力测试、稳定性测试等
  - 性能测试的测试模型中也要包含测试过程的阶段，例如在一次完整的测试过程中，先进行试验性质的响应时间基准测试，检查每个关键业务对应的处理程序是否存在性能问题，若检查到问题，则不需要继续进行后续的性能测试，需要请程序员对该处理程序进行性能调优。在进行完试验性质的测试后，再进行验证性质的性能测试，例如负载测试、压力测试、稳定性测试等，目的是依据需求分析阶段确定的被测系统的性能指标进行各项验证性测试，以确定被测系统是否达到需要要求的性能标准。
- 角色
  - 性能测试分析师
- 目标
  - 定义性能测试整体的阶段划分策略
  - 定义性能测试阶段的测试内容和所需达到的目标
  - 精确估算所需要的测试数据量
- 内容



## 规划-响应时间基准测试

**简单解释：**单个关键业务在系统无压力时，使用单个虚拟用户连续执行**100**次完整操作，每两次执行之间的间隔时间是**5**秒钟，获得该关键业务的平均响应时间。

**测试目的：**这种测试是为了初步检查关键业务响应时间的量级，通常情况下实时交易系统的**关键业务**应该在毫秒级。

**用途：**若单个关键业务的平均响应时间过长，例如一个银行开户交易的响应时间达到了平均**8**秒，这个关键业务的响应时间就太长了，根本就不需要再进行后续的并发性测试了。此时应该协调相关的开发人员发现该关键业务的性能瓶颈，并进行性能调优，直到该关键业务的响应时间达到了可接受的范围之内，例如**1**秒。

## 规划-单业务负载测试

简单解释：单个关键业务在系统无压力时，以[5, 10, 20, 30...]的负载序列执行15分钟的交易操作，获得关键业务的响应时间和处理能力数据，它们至少不能是线性显著增加的，例如5个并发负载时，平均响应时间是2秒，10个并发时，平均响应时间是4秒，而30个并发时，平均响应时间达到12秒。

测试目的：这种测试是为了检查关键业务对应的应用程序模块是否存在代码上的并发性能问题

用途：若单个关键业务的平均响应时间随着并发数量的增加而呈线性增长，则说明该关键业务对应的应用程序存在性能上的瓶颈。此时应该协调相关的开发人员发现该关键业务的性能瓶颈，并进行性能调优，直到该关键业务的响应时间达到了可接受的增长趋势。

## 规划-混合业务负载测试

简单解释：所有被选择的关键业务，以[5, 10, 20, 50, 100...]的负载序列、同时为每个关键业务配置一定的发生概率，执行15分钟的业务操作，获得每个关键业务的响应时间和处理能力数据

测试目的：当某个关键业务的响应时间不符合期望的指标时，进行调优

用途：用于发现潜在的、由于关键业务对应的应用程序模块执行时互相影响造成的性能问题，例如线程互锁、数据库锁等情况。若多个关键业务的平均响应时间过长，则说明该关键业务对应的应用程序模块与其他关键业务对应的应用程序模块在共同执行时存在性能上的瓶颈。此时应该协调相关的开发人员发现这些关键业务的性能瓶颈，并进行性能调优，直到这些关键业务的响应时间达到了可接受的程度。

## 规划-服务器配置测试

**简单解释：**在进行混合业务负载测试时，改变服务器的主要参数值，例如WEBLOGIC的执行线程数量ThreadCount、分配内存大小、内存回收机制等，TUXEDO服务的交易服务数、Oracle数据库的SGA、PGA、PCTFREE等参数的配置，等等

**测试目的：**最优化应用服务器、数据库服务器、操作系统的参数配置

**用途：**应用程序的运行和服务器参数配置之间的关系是密不可分的，程序的性能再高，若没有合理的服务器配置，同样会产生严重的性能瓶颈。在进行混合业务负载测试时，固定一个相对合理的负载级别，改变服务器的参数配置，寻找服务器的最优配置方案。

## 规划-独立路径测试

简单解释：当选择多个关键业务作为性能测试的测试对象后，这些关键业务在整个系统的执行路径可能包括多个，例如一个是经过**CCI系统->ECTIP系统->EAIH系统->DCC数据库主机**，另一个是**CCI系统->ECTIP系统->分行特色业务数据库主机**，需要对每个路径进行混合业务的独立测试

测试目的：确保完整的性能测试，辅助诊断软件性能瓶颈的位置

用途：现代业务系统的架构越来越复杂，早已突破单纯的三层架构。在最前端的最终用户发起的业务请求，可能在某个中间层进行分开处理，到达不同的数据库服务器。若发现混合业务测试场景执行时，某些关键业务的响应时间不符合性能要求，可以将不同路径上的关键业务集进行单独性能测试，以快速定位性能瓶颈的位置。

# 答疑

