
银行手机银行系统性能

测试方案 V1.0

文档属性

文档属性	内容
项目名称：	## 银行手机银行项目
项目经理：	
编写人：	叶强
编写日期：	
文档版本号：	V1.0
评审人员：	
评审日期：	
评审是否通过 [是/否]：	<input type="radio"/> 是 <input type="radio"/> 否

文档版本记录

文档版本号	版本日期	修改人	备注
V1.0	2012-04-10	叶强	初稿

目录

目录	3
一、 参考文档	4
(一) 相关项目文档	4
(二) 参考资料	4
二、 测试概述	4
(一) 测试范围	4
(二) 测试环境	4
(三) 测试工具	5
三、 测试方案	6
(一) 测试计划和安排	6
(二) 基础数据	7
(三) 压力测试	7
(四) 稳定性测试	10
(五) 指标监控	10
(六) 性能指标要求	10
四、 测试场景	10
系统登录 (100 人并发)	11
系统登录 (150 人并发)	11
系统登录 (200 人并发)	11
交易查询 (100 人并发)	12
交易查询 (150 人并发)	14
交易查询 (200 人并发)	14
行内转账 (100 人并发)	15
行内转账 (150 人并发)	15
行内转账 (200 人并发)	15
混合场景测试一 (100 人并发)	17
混合场景测试二 (150 人并发)	17
混合场景测试三 (200 人并发)	17
稳定性测试	18

一、 参考文档

(一) 相关项目文档

编号	文档名
附件 4	需求规格说明书

(二) 参考资料

文档名	文档来源	发表日期	文档版本
需求规格说明书			

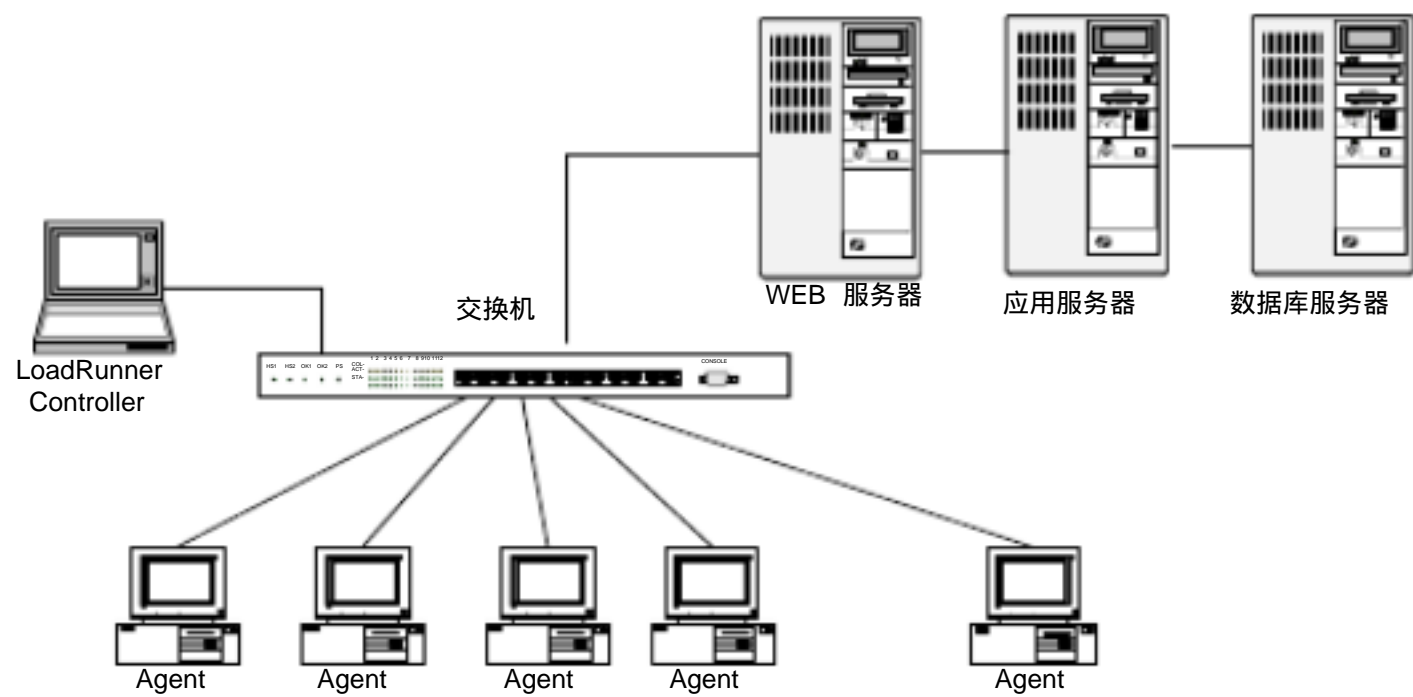
二、 测试概述

(一) 测试范围

手机银行的登录
我的账户信息查询
行内转账查询
交易明细查询
缴费记录查询
行内转账交易

(二) 测试环境

1、 环境拓扑图



2、环境配置

硬件配置

设备名称	数量	型号	IP 地址	操作系统
数据库服务器	1	IBM 3650	17.1.1.1	Linux

软件配置

软件名称	版本号
Weblogic	10.1.2

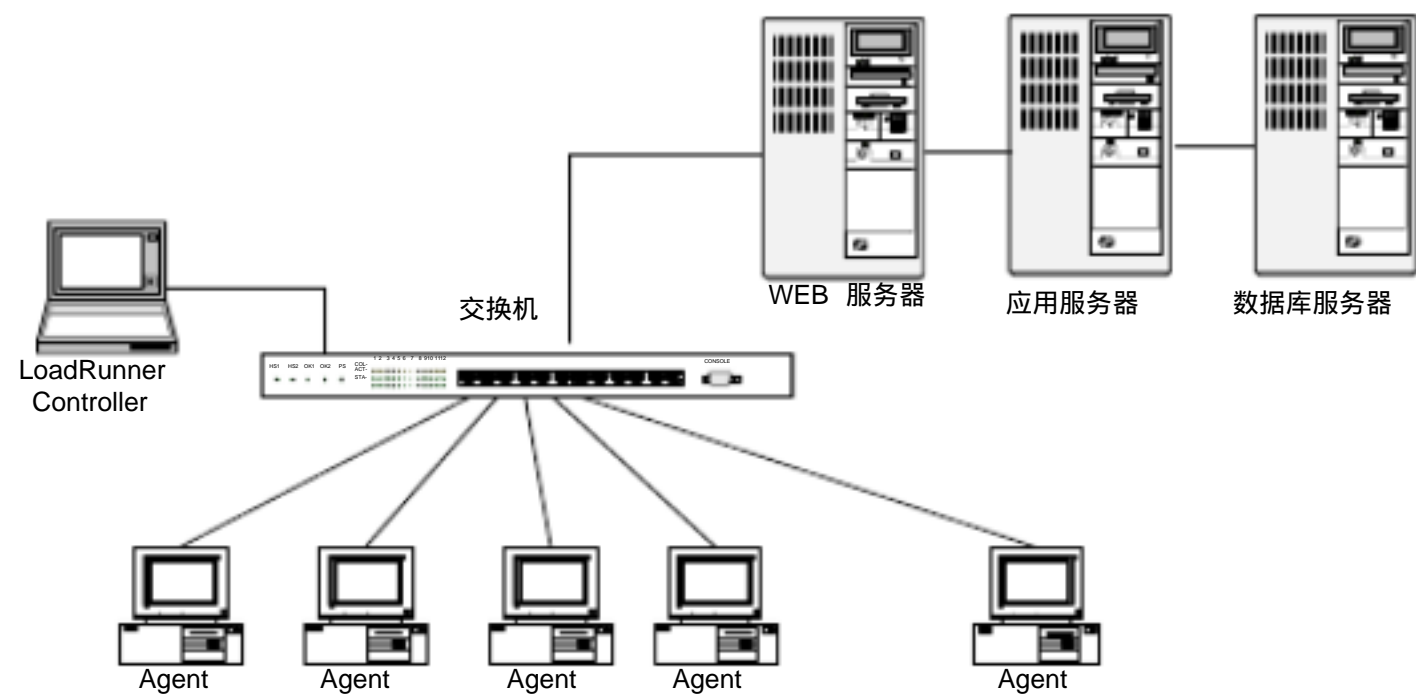
3、环境差异分析

原则上要求系统测试环境尽量地接近生产环境，所以在当系统测试环境和生产环境有不一致的地方，请做相应的环境差异分析，并识别相关的风险。测试环境的差异包括但不限于硬件类型差异、硬件配置差异、系统软件类型差异、系统软件版本差异、系统软件配置差异、应用软件配置差异等。

(三) 测试工具

LoadRunner向运行的测试代理机器 Agent 发送测试指令，测试代理机器运行脚本，模拟多个用户同时向服务器发出请求，测试在不同条件下服务器的响应情况。

性能测试工作原理如下图：



LoadRunner 通过 Virtual User Generator 捕捉客户端向服务器发送和接收的数据流形成脚本框架。在此基础上利用的脚本定制向导自定义测试数据，使用数据表或随机数模拟现实环境的用户数据输入。创建内容检查点，验证负载下的被测系统是否出现功能错误。

通过 Controller 并发指定数量的模拟用户运行以上设置好的脚本，确保测试尽可能接近真实环境，最大程度地反映系统的实际情况。

三、测试方案

(一) 测试计划和安排

1、人员计划

人员组织	职责描述	人员数量
业务测试组	1、编写用户验收测试用例、方案	2
	2、负责验证测试过程中产生的缺陷问题	
	3、执行测试用例，记录测试执行结果表、缺陷记录	
	4、每周汇报阶段测试结果	
	5、编写整体测试报告	
公司方项目经理	1、控制项目执行、协调各方资源	1
	2、负责对缺陷进行确认、分配、解决	
公司方开发人员	负责发现、修复缺陷	9
环境版本控制	负责或协助测试环境的日常维护，主要是数据库和平台的维护，特别是版本配置	1
	配置更新测试环境	

2、日程计划

任务	起始日期	结束日期	备注
测试方案编写	2012.4.10	2012.4.11	
测试方案及测试用例 评审	2012.4.20	2012.4.25	先进行预评估，修改完成后发起正式评审
第一轮测试	2012.4.30	2012.5.16	12 天
测试报告编写	2012.5.9	2012.5.16	完成手机银行测试报告初稿
测试报告评审	2012.5.29	2012.5.31	先进行预评估，修改完成后发起正式评审

3、交付物

文档名称	编制者	其它说明
性能测试报告	叶强	

(二) 基础数据

(三) 压力测试

本次测试是针对手机银行系统在应对密集整转的压力下业务处理能力的测试，检验系统的吞吐率。本系统的压力测试主要是针对主要业务功能、报表统计进行，检查在日间应用高峰时期，并发用户数较多的时候的处理能力等等。

1、单业务压力测试

对于单个交易性能测试和综合交易测试，测试初始都从 100 个用户开始并发，然后以 50 用户递增进行多次压力测试，正常情况下，以 200 用户并发为限。

对于单个交易性能测试，脚本中初始思考时间为 0s，随测试情况进行调整。对于综合交易性能测试，脚本中加入适当思考时间。

1.1 手机银行系统登录

A 交易描述

手机银行发起登录交易

B 业务逻辑与处理

进入手机银行登录页面

输入用户号与密码通过认证并成功登录到手机银行

1.2 我的账户信息查询

A 交易描述

在我的账户中，点击交易查询，系统发起查询交易并取得结果，并将具体交易信息显示在结果页面中。

B 业务逻辑与处理

进入手机银行登录页面

输入用户号与密码通过认证并成功登录到手机银行

进入我的账户

1.3 手机转账汇款查询

A 交易描述

进入转账汇款中，选择行内转账，输入对方的户名及手机号，验证通过后，转账成功。

B 业务逻辑与处理

进入手机银行登录页面

输入用户号与密码通过认证并成功登录到手机银行

进入转账汇款，选择手机转账汇款

输入对方的户名及手机号

输入转账金额及转账用途

提交转账交易，验证成功后，转账成功

1.4 行内转账

A 交易描述

进入转账汇款中，选择行内转账，输入对方的户名及账号，验证通过后，转账成功。

B 业务逻辑与处理

进入手机银行登录页面

输入用户号与密码通过认证并成功登录到手机银行

进入转账汇款，选择行内转账

输入对方的户名及账号

输入转账金额及转账用途

提交转账交易，验证成功后，转账成功

1.5 交易明细查询

A 交易描述

在我的账户中，点击交易查询，系统发起查询交易并取得结果，并将具体交易信息显示在结果页面中。

B 业务逻辑与处理

进入手机银行登录页面

输入用户号与密码通过认证并成功登录到手机银行

进入我的账户，选择交易查询

选择交易查询的起始日期和结束日期

点击“查询”按钮

1.6 缴费记录查询

A 交易描述

进入转账汇款中，选择行内转账，输入对方的户名及账号，验证通过后，转账成功。

B 业务逻辑与处理

进入手机银行登录页面

输入用户号与密码通过认证并成功登录到手机银行

进入转账汇款，选择行内转账

输入对方的户名及账号

输入转账金额及转账用途

提交转账交易，验证成功后，转账成功

2、混合业务压力测试

针对混合业务压力测试，在测试中将选择部分具有代表性的交易作为测试对象，是通过并发用户对登录、查询、交易进行综合性的压力测试的一种过程。本次测试将按照如下原则选择性测试交易：

关键交易：关键业务是用户最为关注的交易，需要保证其性能和质量；

吞吐量：某些业务流程可能不是关键业务，但是很高的吞吐量；

在选择的交易中，先进行单个交易性能测试，在此基础上进行综合交易性能测试，

融合两种测试的优点。

混合业务性能测试：各种关键交易功能按照交易配比来分配具体的虚拟用户数进行综合交易性能测试。这部分测试将根据系统各交易功能的实际使用频率和重要程度来决定业务配比。

3、其他

使用美科利公司（Mercury）的性能测试软件 LoadRunner，对现行的手机银行系统进行脚本录制、测试回放、逐步加压和跟踪记录。测试过程中，由 LoadRunner 的管理平台调用各台测试前台，发起各种组合的交易请求，并跟踪记录服务器端的运行情况和返回给客户端的运行结果。

针对每个测试用例，都将采用逐步加压和瞬间加压两种客户端连接方式进行，查看服务器端在客户端的连接数量变化过程中对应的处理能力，以更好的定位系统在达到多少并发或压力下我们的系统出现了不稳定。

（四）稳定性测试

通过 Loadrunner 模拟压力测试，在系统承受最大压力的情况下进行长时间的稳定性测试，从而测试系统服务器的稳定性。

（五）指标监控

记录可扩展性测试中的测试结果及其系统的运行状况。除了记录测试指标以外，应该结合测试实时记录系统各个层次的资源和参数。主要包括：

硬件环境资源

服务器操作系统参数

网络相关参数

数据库相关参数：具体数据库参数有所不同，结合各个数据库独有的特点记录

（六）性能指标要求

- 1、手机银行按网银的 50%计算用户量和交易量，考虑5年的发展，每年增长 50%计算。手机银行支持的用户量和交易量分别达到 15万和6000笔/天。
- 2、手机银行系统支持最少每秒 100次的并发请求。
- 3、手机银行应用服务在上述性能指标下的平均响应时间不超过 1秒。

4、系统应采用高效、可靠的措施保证交易处理的正确性和一致。

四、测试场景

系统登录（100人并发）

A、测试场景

并发 100 用户登录，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 1 小时。

1 秒钟停止 2 个用户。

总场景时间 1 hour ， 1 minute and 42 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

B、测试场景包含交易

LoadRunner 脚本中的 Action_Transaction 包含交易有：登录

系统登录（150人并发）

A、测试场景

并发 150 用户登录，每 1 秒钟增加 2 个用户。

75 秒后达到 150 用户并发，并发 1 小时。

1 秒钟停止 2 个用户。

总场景时间 1 小时。

B、测试场景包含交易

LoadRunner 脚本中的 Action_Transaction 包含交易有：登录

系统登录（200人并发）

A、测试场景

并发 200 用户登录，每 1 秒钟增加 2 个用户。

100 秒后达到 200 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 23 minutes and 26 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

B、测试场景包含交易

LoadRunner 脚本中的 Action_Transaction 包含交易有：登录

账户信息查询 (100 人并发)

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 21 minutes and 41 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

账户信息查询 (150 人并发)

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 21 minutes and 41 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

账户信息查询 (200 人并发)

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 21 minutes and 41 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

手机转账查询 (100 人并发)

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 21 minutes and 41 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

手机转账查询 (150 人并发)

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 21 minutes and 41 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

手机转账查询 (200 人并发)

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 21 minutes and 41 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

交易明细查询（ 100 人并发）

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 21 minutes and 41 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

交易明细查询（ 150 人并发）

A、测试场景

并发 150 用户，每 1 秒钟增加 2 个用户。

75 秒后达到 150 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 22 minutes and 34 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

交易明细查询（ 200 人并发）

A、测试场景

并发 200 用户，每 1 秒钟增加 2 个用户。

100 秒后达到 200 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 23 minutes and 24 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

行内转账（100 人并发）

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 21 minutes and 41 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

行内转账（150 人并发）

A、测试场景

并发 150 用户，每 1 秒钟增加 2 个用户。

75 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 22 minutes and 33 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

行内转账（200 人并发）

A、测试场景

并发 200 用户，每 1 秒钟增加 2 个用户。

100 秒后达到 200 用户并发，并发 1 小时。

1 秒钟停止 2 个用户。

总场景时间 1 小时。

B、测试场景包含交易

LoadRunner 脚本中的 Action_Transaction 包含交易有：行内转账

缴费记录查询（ 100 人并发）

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 21 minutes and 41 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

缴费记录查询（ 150 人并发）

A、测试场景

并发 150 用户，每 1 秒钟增加 2 个用户。

75 秒后达到 150 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 22 minutes and 34 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

缴费记录查询（ 200 人并发）

A、测试场景

并发 200 用户，每 1 秒钟增加 2 个用户。

100 秒后达到 200 用户并发，并发 20 分钟。

1 秒钟停止 2 个用户。

总场景时间 23 minutes and 24 seconds 。

SIT 测试环境。

交易挡板时间为 1 秒，thinktime 为 0.5 秒。

混合场景测试一（ 100 人并发）

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

50 秒后达到 100 用户并发，并发 1 小时。

1 秒钟停止 2 个用户。

总场景时间 1 小时。

B、测试场景包含交易

LoadRunner 脚本中的 Action_Transaction 包含交易有：登录、交易查询、行内转账、缴费记录查询

混合场景测试二（ 150 人并发）

A、测试场景

并发 100 用户，每 1 秒钟增加 2 个用户。

75 秒后达到 150 用户并发，并发 1 小时。

1 秒钟停止 2 个用户。

总场景时间 1 小时。

B、测试场景包含交易

LoadRunner 脚本中的 Action_Transaction 包含交易有：登录、交易明细查询、行内转账、手机转账查询、缴费记录查询

混合场景测试三（ 200 人并发）

A、测试场景

并发 200 用户，每 1 秒钟增加 2 个用户。

100 秒后达到 200 用户并发，并发 1 小时。

1 秒钟停止 2 个用户。

总场景时间 1 小时。

B、测试场景包含交易

LoadRunner 脚本中的 Action_Transaction 包含交易有：登录、交易明细查询、行内转账、手机转账查询、缴费记录查询

稳定性测试

A、测试场景

模拟正常生产并发用户数，峰值为 200 人并发，忙时为 100 人并发，谷值为 20 人并发，忙时时间设置为 6 小时（上午 9-12 点，下午 1-4 点，其中 10-11 点为 1 小时峰值，2-3 点为 1 小时峰值），其他时间为谷值，场景设计如图。
场景开始时间模拟为上午 7 点，执行 24 小时。