



具体应用测试案例分析 4 Web站点测试 4 TUXEDO及国产中间件测试 4 Winsock并发性能测试 4 SQLServer并发性能测试 4 NetLoad并发性能测试



















具体应用测试案例分析

4Web站点质量保证

支持的连接类型包括:

图片、框架(Frame)、插件(Plugin)、背景、样式表(Style Sheet)、数本、Java Applet等。

支持的连接种类包括:

HTTP、FTP、GOPHER、HTTPS等。另外, 对本地的 连接和重定向的连接也有很好的支持。



具体应用测试案例分析

4Web站点质量保证

- 工具主要特点如下:
- 4简单易用
- 4在实现上采用多线程技术,因此检查速度特别快
- 4对断开的连接可以再次检查,避免误判
- 4没有检查连接的数量限制,只受系统资源的约束
- 4有网站地图功能
- 4检查结果可以分类查看,自动生成HTML格式的报告



具体应用测试案例分析

- 4 并发性能测试关键技术:
 - 1. 模拟IP地址变量
 - 2. 处理Web服务器的错误信息
 - 3. 静态HTML页面的模拟
 - 4. 模拟CGI请求与表格
 - 5. 模拟Java胂本及 Java Applet
 - 6. 模拟Cookies



具体应用测试案例分析

- 4 测试结果分析:
 - 1. 事务处理速率
 - 总脚本及检查点响应时间,响应时间中分别提供最小、 最大、平均、偏差以及90%响应时间
 - 脚本执行情况統計中提供的信息包括: HTTP成功请求數、HTTP結構求數、HTTP請求重定向數目、HTTP持求重试數目、HTTP 400錯误數目、HTTP500錯误數目、连接错误、接收错误、发送错误等



具体应用测试案例分析

- 4 测试经验讨论:
 - 在执行客户端并发性能测试的过程中,需要同时 监控数据库服务器、Web服务器以及网络资源等 使用情况,以便对系统的性能做全面评估
 - 2. 录制脚本和手工编写脚本相结合
 - 3. 设置数据池,实现变量加载
 - 4. 业务批量执行



具体应用测试案例分析

- 4 测试经验讨论 (续):
 - 5. 模拟用户数的递增
 - 6. 合理设置交易之间时间间隔
 - 7. 模拟IP地址变量的技术
 - 8. 超时(timeout)的设置 9. 并发用户连续执行交易数的设置
 - 10.错误处理



具体应用测试案例分析 4 测试经验讨论(续): 11.利用QALoad的ActiveData技术 12.采用复合交易测试方案 13.尽量将执行负载测试的机器合理分布 14.并发用户数量极限点 15.加压机器的CPU使用率也有必要监控 16.设置并发点 17.注意如下参数的配置









```
BINDL.

PLAYER_INFO 's_info.

(**)

Checknown Variables '**)

SET_ABDR_F EMBER_TON(abort_function):

DEFINE_TRANS_IYPE("was.Advanced fech_1 s");

DEFINE_TRANS_IYPE("was.Advanced fech_1 s");

/**(Checkpoints have been included by the convert process of the convertion of the converti
```

```
能式后的脚本
PLAYE_NO **_Inbo*
- Declare variables */
- Declare variables variables
```

具体应用测试案例分析 4 SQLServer并发性能测试 2 关键技术: 4 从存储过程中扑获一个值 4 利用检索到的值作为一个参数传递给存储过程 看看下面的实例脚本!

中国软件评测中心 Citie Solberts Inding Cities

```
原始 QALoad 脚本代码
strcpy(sql_statement,/* >> 2 << */
"execute use_inc_value_sp @inc_value ={01}");
DO_substr(sql_statement, 1, "101");
BEGIN_CHECKPOINT(); /* #2: Stored Procedure */
DO_dbcmd(0, sql_statement);
DO_dbsqlexec(0);
while (DO_dbGetResults(0));
修改后的 QALoad 代码(使用字符单位)
strcpy(sql_statement, /* >> 2 << */
"execute use_inc_value_sp @inc_value ={01}");
/* Note that szSecondParam was declared and received the
value in the steps in Part 1 */
DO_substr(sql_statement, 1, szSecondParam);
BEGIN_CHECKPOINT(); /* #2: Stored Procedure */
DO_dbcmd(0, sql_statement);
DO_dbsqlexec(0);
while (DO_dbGetResults(0));
```









