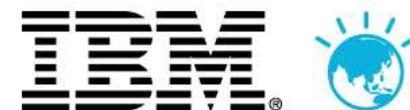


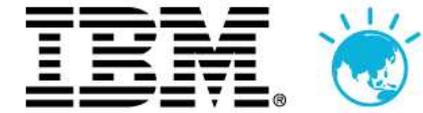
# IBM Rational 软件创新论坛



Let's **build** a  
smarter planet.

开发有道  
创新“智”造

Innovate**2010**



# IBM Rational企业性能工程解决方案

任党恩 [rende@cn.ibm.com](mailto:rende@cn.ibm.com)  
IBM中国开发中心高级工程师

Innovate2010

# 议程

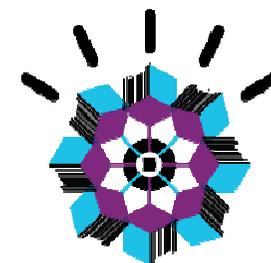


- 企业性能测试的现状
- 企业高性能应用交付面临的挑战
- 从性能测试到性能工程
- **IBM Rational**性能工程解决方案
  - 性能为主线的团队协作
  - 集成的性能测试工具
  - RPT构建灵活高效的企业测试
  - 实例: **CM Server**可靠性测试
- 总结



# 性能表现直接影响企业应用的成败

- 商业价值的实现：时间和金钱代价
- 最终用户的使用体验，满意度，忠诚度
- 随着硬件提升和新技术应用，客户期望值也在提升
- 性能优势是核心竞争力



# 性能测试

- 什么是性能测试？
  - 评估保障系统性能相关的NFR（响应，吞吐率，支持负载，资源消耗等）
  - 优化系统实现，找到系统瓶颈，调优建议等
  - 形态: 基线，负载，压力，长时间可靠性测试等
- 谁在做性能测试？
  - 单一的测试团队
  - 仅在项目后期或部署之前介入

人人测试，尽早测试





# 企业高性能应用交付面临的挑战

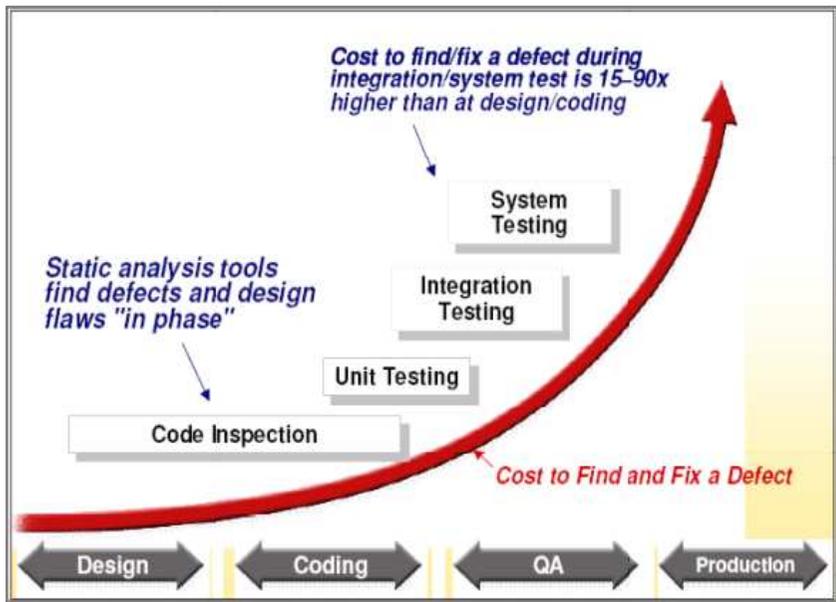
- 企业系统
  - 规模大，多部门协同开发，新旧技术并存
  - 模块交互频繁，部署复杂
- 团队
  - 跨地域，分布式：不利于信息畅通和资产共享
  - 团队成员的水平参差
- 流程
  - 缺乏系统的性能保障方面的规范和指导
- 客户需求
  - 响应快，高并发，处理大量业务
  - 可靠性，扩展性好，支持平稳扩容
- 测试工具
  - 自动化，快速构建各类复杂的大负载应用模拟
  - 评估性能，分析定位问题能力



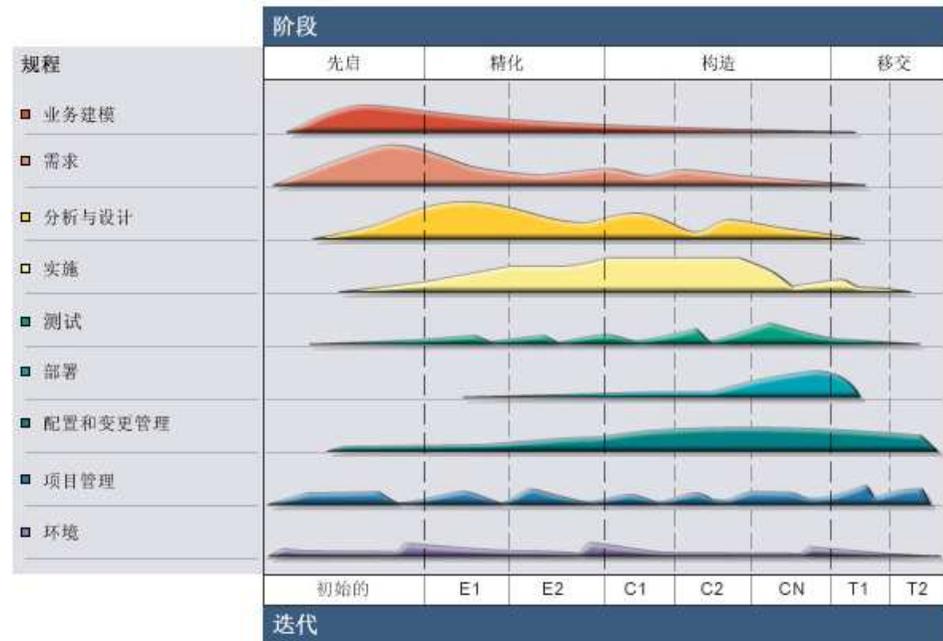
# 什么是性能工程



系统开发生命周期各阶段关联的角色，技能，活动，实践，工具和交付件，以确保整体方案的设计，实现和运维满足既定的性能需求。



尽早测试，尽早发现缺陷



性能考量贯穿生命周期各阶段

# IBM Rational性能工程解决方案



## 业务需求分析

明确业务流程和价值  
分解为系统性能目标



## 架构与设计

系统架构, 模块划分, 性能建模  
基于故事板/用例明确性能场景, 系统容量和工作负载等



## 开发

设计模式, 代码审核  
性能UT, profiling  
模块集成测试  
代码优化, 局部重构



## 系统测试

测试计划, 环境准备  
测试脚本编制  
复合场景的性能测试  
系统监控, 结果分析  
性能报告, 调优指导



## 部署运维

部署生产环境  
监控及数据分析  
系统优化  
性能趋势分析, 容量计划

## 性能工程最佳实践、技术和工具

性能设计  
评审和验证

代码审查

静态分析

运行时分析

功能级  
性能测试

系统级  
性能测试

性能需求追踪

培训  
文档  
实践总结

Home-grown harness, JUnit, RSAR, PurifyPlus, RFT, HttpWatch, RPT, Firebug ...



## 项目和过程管理



测试项目库

需求库

代码库

测试用例库

行业资产库

.....

性能质量管理流程和协作平台  
(培养“注重性能”的团队文化)

# 基于性能的设计与开发



## 设计阶段的考虑

- 模块划分遵循松耦合，高内聚
- 通信机制
- 并发设计：线程同步，锁，队列
- 资源分配：连接池技术，值守线程
- 设计模式，算法
- 框架应用带来的开销

## 开发阶段的考虑

- 代码效率审查：人工与工具相结合
- 必要的文档，灵活的形式
- 开发人员自测：**scripts, JUnit, trace/log**，功能性能测试工具
- 与设计师，测试人员保持经常性的沟通和讨论



# 静态及动态代码分析

RSAR

- J2EE Best Practices [0/63]
- J2SE Best Practices [0/308]
- JUnit [0/8]
- Naming [0/37]
- Performance [39/39]**
  - Always instantiate collections with a specific size
  - Avoid calling methods in conditions of for loops
  - Avoid checking the length of a String by invoking Str
  - Avoid creating a new object for the purpose of invoki
  - Avoid declaring or assigning variables in a loop that
  - Avoid Map and Set of java.net.URL objects
  - Avoid multiple invocations of the same method
  - Avoid synchronized blocks inside loops
  - Avoid try catch finally blocks inside loops
- Memory [20/20]
- Profiling [7/7]
- Speed [12/12]

- C++/Java等代码静态检查和分析
- 可定制规则集和扩展支持的语言
- 可与RSA/RAD集成使用
- 可结合到构建环节，生成分析报告。

Rational Quantify - hello.exe

Function	Calls	Function time	F-D time	F time (% of Focus)	F-D time (% of Focus)	Avg F time	Min F time
SetTimer	5	3,181.21	3,181.21	8.81	8.81	636.24	6
GetCharABCWidth	1	3,158.43	3,158.43	8.74	8.74	3,158.43	3,158.43
ExitProcess	1	2,954.93	15,654.09	8.18	43.34	2,954.93	2,954.93
moduleEntry_ole32	2	2,861.23	3,304.50	7.37	9.15	1,330.62	36
SetWindowRgn	1	1,180.44	1,994.67	3.27	5.52	1,180.44	1,180.44
GDI_DrawStream	30	743.54	743.54	2.06	2.06	24.78	1.1
ScriptStringAnalyze	135	684.31	4,803.87	1.89	13.30	5.07	0.1
CreateCompatibleB	6	503.97	503.97	1.40	1.40	83.99	19.1
GetTextExtentExPo	108	434.37	434.37	1.20	1.20	4.02	0.1
ShredDIBits	2	426.41	426.41	1.13	1.13	203.20	14.4
GetModuleHandleW	6	362.68	362.68	1.00	1.00	60.45	35.6
DeleteObject	27	321.02	323.09	0.89	0.89	11.89	0
GetWindowDC	30	314.69	314.69	0.87	0.87	10.49	0.6
moduleEntry_mactime	2	260.70	1,317.05	0.72	3.65	130.35	61.1
GDI_Initalize	1	227.73	1,135.37	0.63	3.14	227.73	227.73
NetConnectPort	1	162.20	162.20	0.50	0.50	162.20	162.20
GetDC	3	156.84	156.84	0.43	0.43	52.28	2.1
BaseCheckAppcom	2	122.11	122.11	0.34	0.34	61.06	42.1
IntersectClipRect	23	121.39	121.39	0.34	0.34	5.28	0.6
CoTaskMemFree	177	119.32	119.32	0.33	0.33	4.39	0

- PurifyPlus运行时工具集
- Quantify做代码profiling
- 无需源码即可测试
- 根据变更部分，合理选取测试
- 频繁使用，自动化

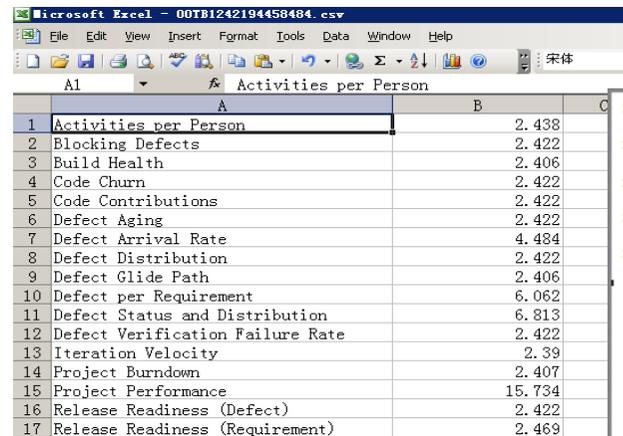
# 功能测试中的性能度量

## 实例：RFT批量测试报表性能

- 基于Cognos的Insight报表
- 功能验证，数据是否正确
- 单用户的性能基准
- 50-60个报表，每日构建



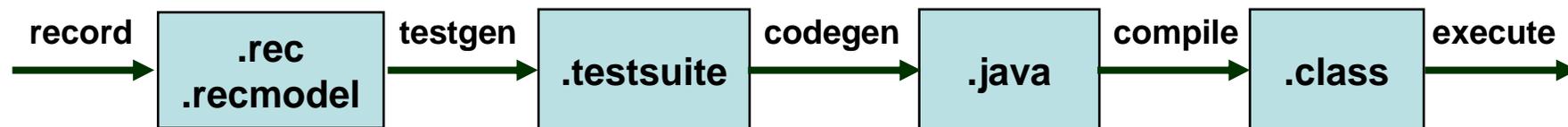
- RFT编写代码自动测试
- 报表内容验证
- 记录响应时间，写log
- 调用excel



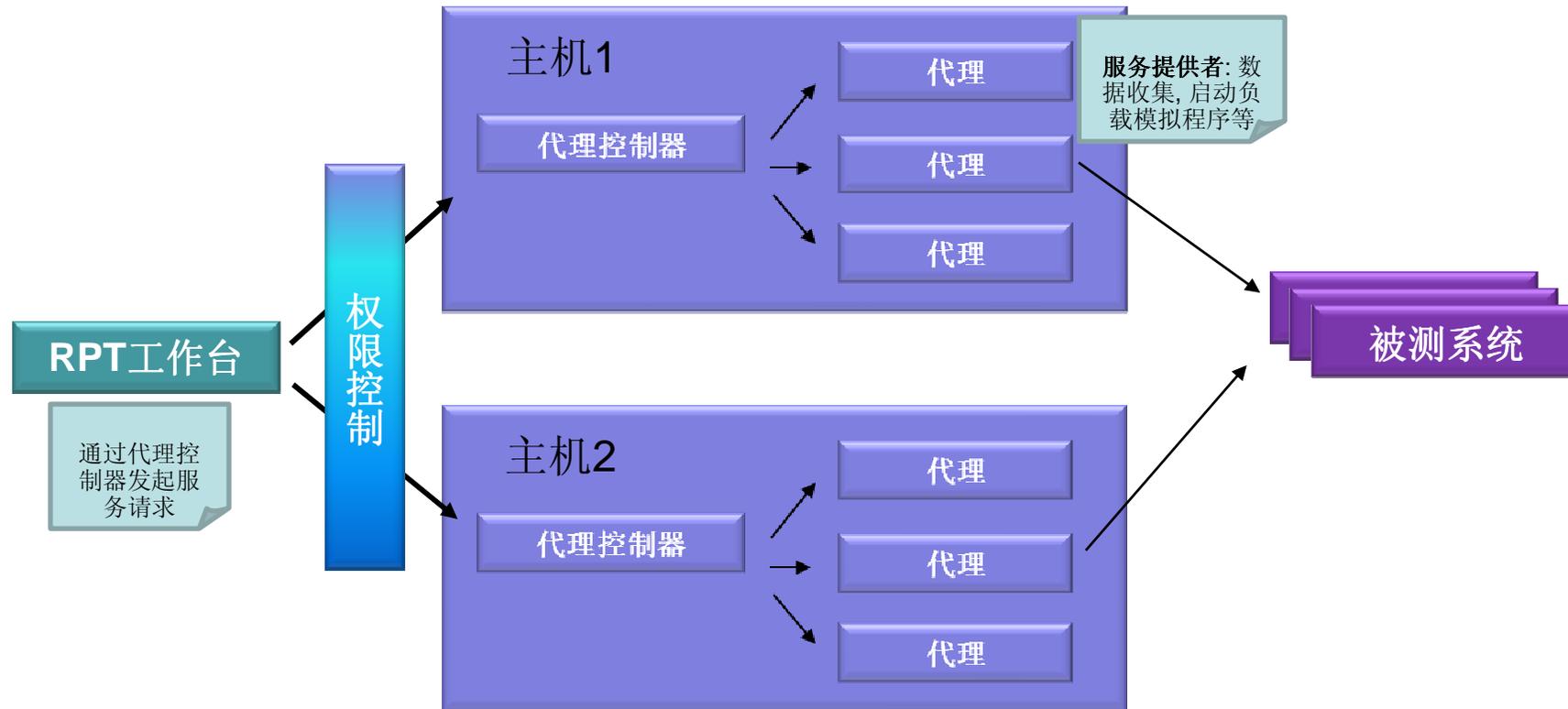


## RPT: 面向企业的系统性能测试工具

- 基于Eclipse/TPTP，开放的可扩展的性能测试工具
- 录制，编辑，回放，监控，分析，报告
- 模拟真实用户负载，发现瓶颈和其它问题
- 自动化系统性能，压力和容量规划测试
- 隔离性能瓶颈的源头，优化系统资源的使用



# RPT工作台和代理



# RPT扩展企业应用的测试场景

- 源于Java和RPT体系结构上的可扩展性
- 扩展方式
  - 录制+定制代码
  - 无录制：利用内建的测试控制+定制代码，加速数据生成或并发性能测试脚本的开发
  - 扩展SDK开发更多的协议支持，如CC-Telnet插件
  - 增加特定应用相关的度量监控，日志等
- 与开发团队有效沟通，得到足够的技术支持

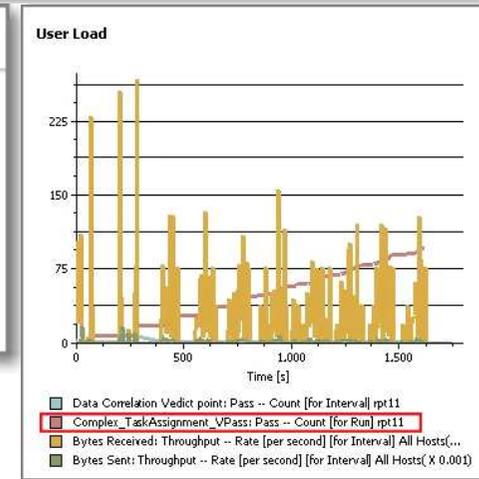
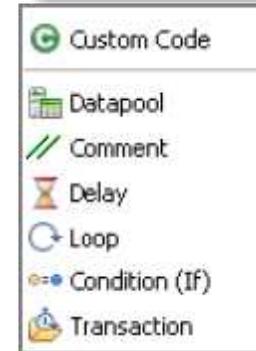


# RPT中开发Java测试脚本的优势

充分利用RPT提供的测试功能的诸多便利

- 条件判断，循环，同步点，随机选择等执行逻辑控制
- 直观地利用测试，调度构造复杂的多用户多实例测试
- 通过数据池快速实现数据驱动测试动态关联
- 利用定制代码接口实现自定义的计数器，并实时呈现
- 测试监控、报告和结果分析
- 开箱即用的Java编辑调试环境

	Line_Num::String	SiteID::String	ID::String
0	102100033331	1021000333	1000
1	10210003333T	1021000333	1000
2	102100033353	1021000333	1000
3	102100033363	1021000333	1000
4	00110001111T	0011000111	1000
5	308581004055	3085810055	5810
6	308581000109	3085810055	5810



设想一下，

如果直接编写Java stand-alone测试应用

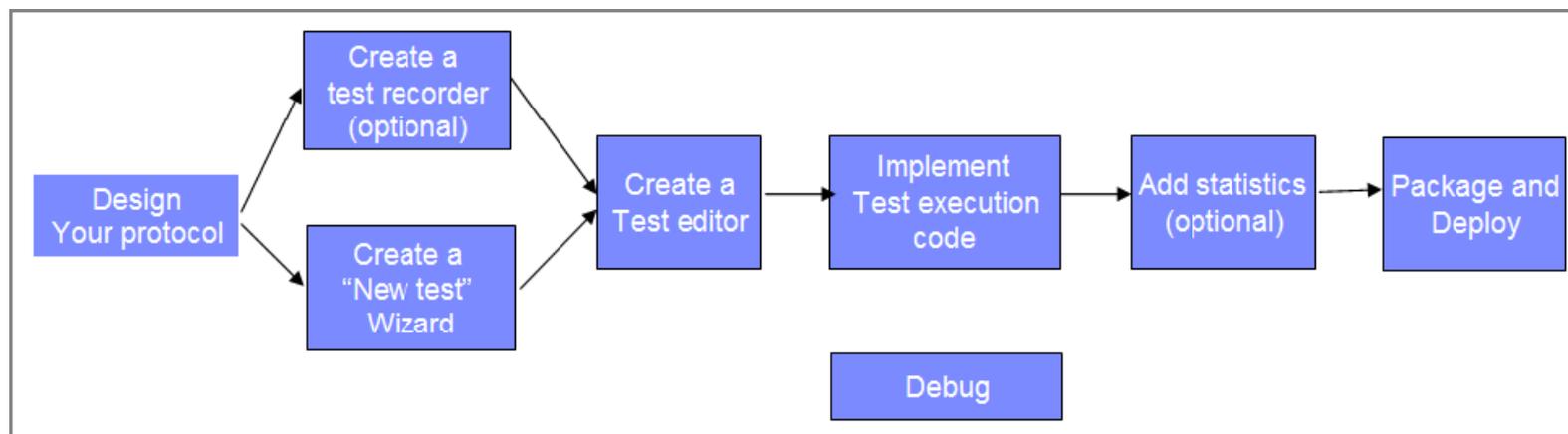


# 实例：组合定制代码，为MQ系统加压

- 通过测试工具向测试机的MQ服务队列压数据
- 服务队列数据到达一定量后向后台MQ服务器发送数据  
检验后台MQ服务器，CICS和数据库的处理能力
- RPT无法直接录制，借助定制代码调用第三方API, 对目标系统的加压测试
- 不同用户组向不同的MQ通道发消息，每组包含若干虚拟用户，从而构造出多线程并发访问。



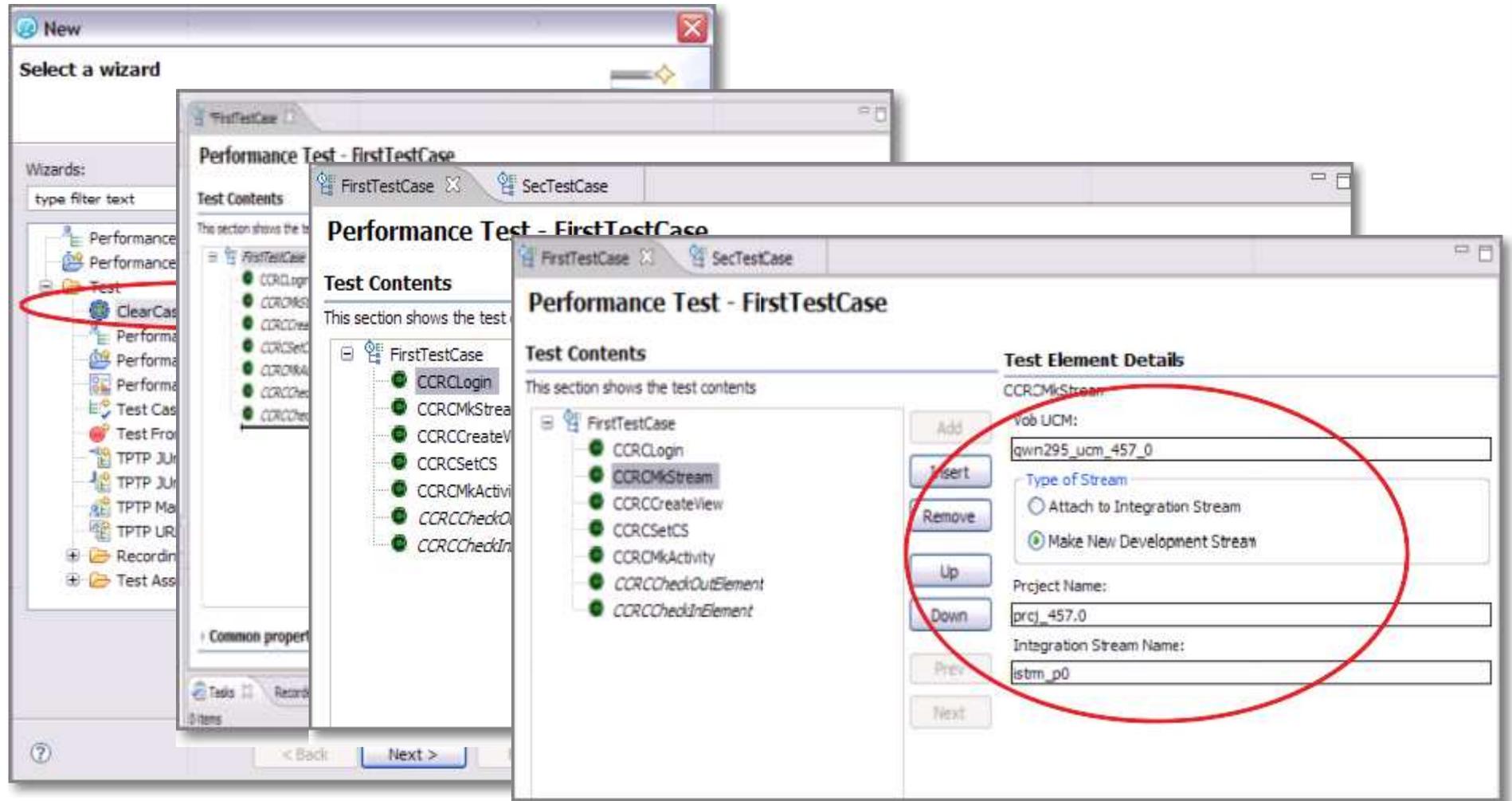
# 实现协议扩展



- 何时采用定制代码，何时考虑协议扩展？
  - 固化操作原语(代码模板)
  - 测试编辑器界面
  - 重用程度
- 参照RPT帮助文档提供的Socket协议实现样例



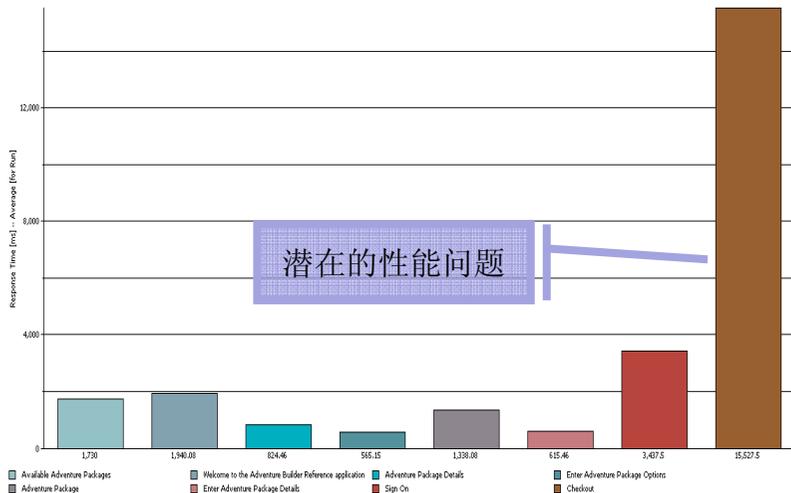
# 实例：通过TeamAPI测试CCRC



# 性能问题定位与分析

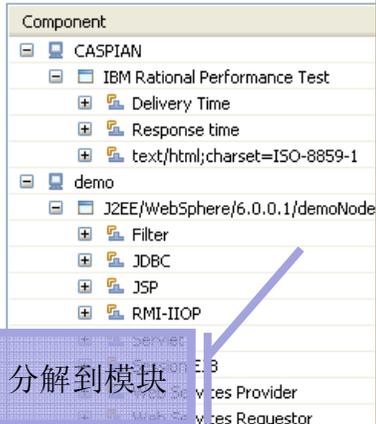
## Page Performance

Average Page Response Time for Run (Filter applied: Count Filter: 10 highest)



## Page Performance > Response Time Breakdown Statistics

demo:9080/ab/checkout.do



## Execution Statistics

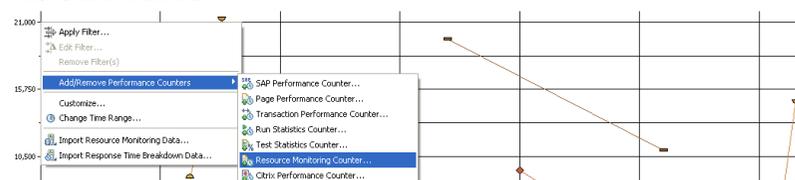
Execution Statistics - Data Collection Infrastructure [ PID: 2440 ] (Filter: No filter)

>Package	Base Time (sec...)	Average Base ...	Cumulative Tim...	Calls
com.ibm._jsp	13.473001	0.068045	19.279000	198
com.ibm.ws.naming.jndicos	0.539999	0.012857	0.539999	42
com.ibm.ws.rsadapter.jdbc	5.409002	0.023017	8.060002	235
com.sun.j2ee.blueprints.opc.powebservice	40.176000	10.044000	40.248000	4
PoEndpointBean	40.176000	10.044000	40.248000	4
submitPurchaseOrder(com.sun.j2ee.blueprints.opc.powebservice)	40.176000	10.044000	40.248000	4
com.sun.j2ee.blueprints.signon.web	4.686001	0.037790	74.546999	124
com.sun.j2ee.blueprints.waf.controller.web	45.796998	2.862312	58.878000	16
com.sun.j2ee.blueprints.waf.view.template	1.080999	0.032758	20.359999	33
db2j.ai.c	0.455000	0.011974	0.455000	38
db2j.ai.p	0.600999	0.005050	0.600999	119
db2j.bk.b	0.527001	0.019519	0.778000	27
db2j.bk.c	2.050001	0.016942	2.651000	121
java.lang	0.000000	0.000000	0.000000	0

分解到模块

分解到方法

## Average Page Response Time [for Interval]



## Data Sources:

- IBM DB2 Monitoring
- IBM Tivoli Monitoring
- IBM WebSphere PMI Monitoring
- JBoss Application Server Monitoring
- Oracle WebLogic Server Monitoring
- SAP NetWeaver Web Application Server Monitoring
- UNIX rstatd monitor
- Windows Performance Monitor

多种数据监控手段

## Location Resource Options

Specify the type of resource data you wish to collect

- 192.168.0.203
  - cacheModule
  - connectionPoolModule
  - ExtensionRegistryStats.name
  - hamanagerModule
  - jvmRuntimeModule
    - FreeMemory
    - HeapSize
    - ProcessCpuUsage
    - Up Time

Apply Filter...

Edit Filter...

Remove Filter(s)

Add/Remove Performance Counters

Display Page Element Responses

Display Response Time Breakdown Statistics...

Display Host Response Time Breakdown

Customize...

Change Time Range...

Import Resource Monitoring Data...

Import Response Time Breakdown Data...

从ITM导入更多监控数据

# 定义性能需求并验证

- RQM测试计划/测试用例中定义性能需求
- RPT性能测试中
  - 定义特定请求/页面/事务的性能需求
- RPT性能调度中
  - 定义汇聚的请求/页面/事务的性能需求
  - 定义资源监控度量的性能需求
- 在性能需求报告中查看结果

Requirements ? **Classics Java Test Plan** ?  
Test Plan Overview | View Snapshots

This section lists all of the content and requirements associated with a given test plan. You can see the test plan.

Group by: Ungrouped

Show All Items per page Previous | 1 - 2 of 2 | Next

<input type="checkbox"/>	Status	ID	Risk	Name	Owner
<input type="checkbox"/>	🟡	52	🔴🔴🔴🔴	CQ ETL throughput no less than 75K per ...	Rui
<input type="checkbox"/>	🟡	53	🔴🔴🔴🔴	Simple report takes less than 10 seconds	dan



**Schedule Element Details**

25u\_5iph

Resource Monitoring | **Statistics** | Performance Requirement

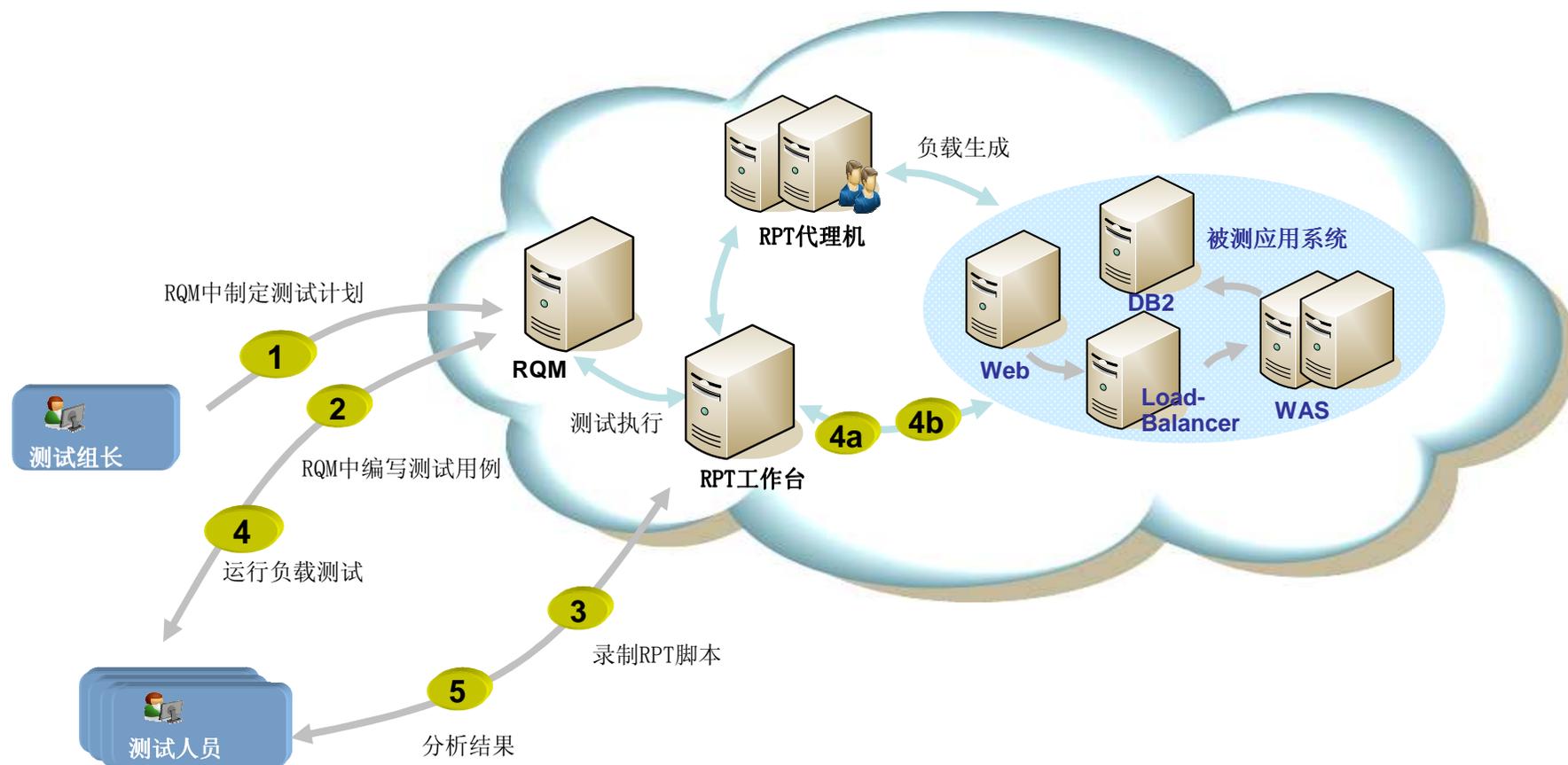
Enable Performance Requirements

**HTTP Protocol**

Average Response Time For All Page Elements [ms] [for Run]	
Response Time Standard Deviation For All Pages [for Run]	
Percent Page VPs Passed [for Run]	
Page Hit Rate [per second] [for Run]	
Minimum Response Time For All Pages [ms] [for Run]	
25th Percentile Response Time for all Page Elements [ms] [for Run]	
50th Percentile Response Time for all Page Elements [ms] [for Run]	
75th Percentile Response Time for all Page Elements [ms] [for Run]	
85th Percentile Response Time for all Page Elements [ms] [for Run]	
<b>90th Percentile Response Time for all Page Elements [ms] [for Run]</b>	<b>&lt;= 500</b>
95th Percentile Response Time for all Page Elements [ms] [for Run]	



# 协作平台和测试工具集成示例



云计算环境中实施大规模负载测试

# 协作平台和测试工具集成示例

**RQM分配工作项**

ID	Summary	Artifact	State
51	Provide the Summary Section for TestCase: Performance: Reliability Test	Performance: Reliability Test	新建
49	Review TestPlan: ClassicCD 1.0 system test plan	ClassicCD 1.0 system test plan	新建

**RQM发起测试执行**

Type	Name	test script	Host Name	Progress	Modified
Execution	Unassigned		Local Computer	0%	1 minute ago
Performance: Data Volume Test Execution	Clean the testing server		my_win2k3r2	50%	Aug 15, 2010

Summary: \* Regression: M3 order submission takes 50% mor...

**RTC中提交缺陷**

**Details**

Type: Defect

Filed Against: Test\_Automation

Severity: Major

Found In: Unassigned

Project Area: Test Automation

Creation Date: Aug 20, 2010 1:20 PM

Created By: dan

Tags:

Owned By: Jiang



**测试资产版本控制**

Repository Files | Pending Changes | Quality Manager Adapter

Workspace Test\_Automation Stream Workspace - ClassicCD\_Comp

- Datapools
- Results
- Schedules
  - 711 combined
  - 7112
  - combined
  - populate
  - test
  - Benchmark.testsuite
  - DataVolume.testsuite
  - SingleUser.testsuite
  - trial.testsuite
- src
  - concurrency
  - DecRef.java
  - IncRef.java
  - ModCounter.java
  - test

Test Na | Test Da

ClassicCD (Test\_Automati

- Datapools
- Results
- Schedules
  - 7112
  - 711 combined
  - combined
  - populate
  - test
  - Benchmark
  - DataVolume
  - SingleScript
  - trial
- Tests

**历史版本与变更集**

History | Repository Files | Problems | Javadoc | Declaratio

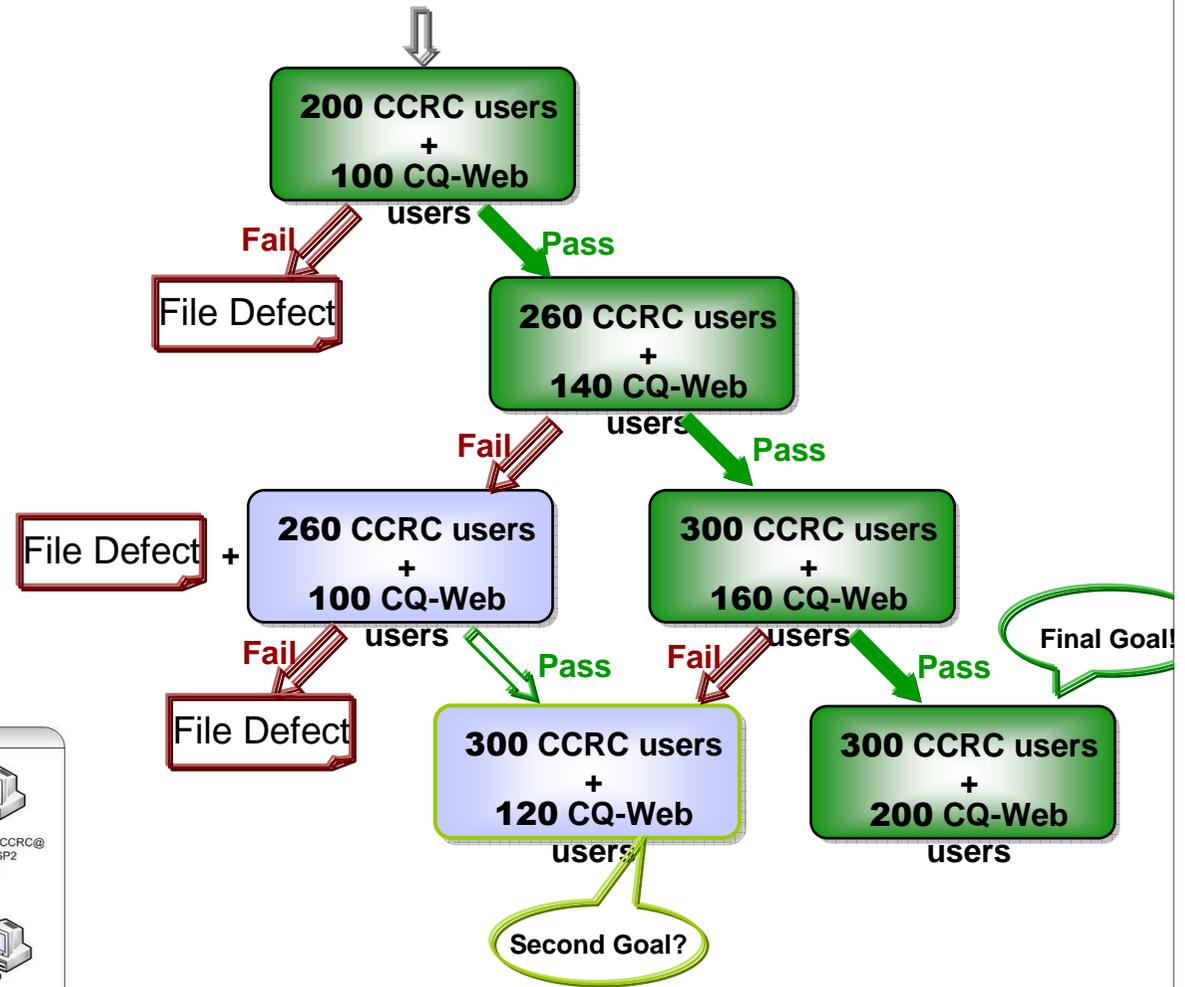
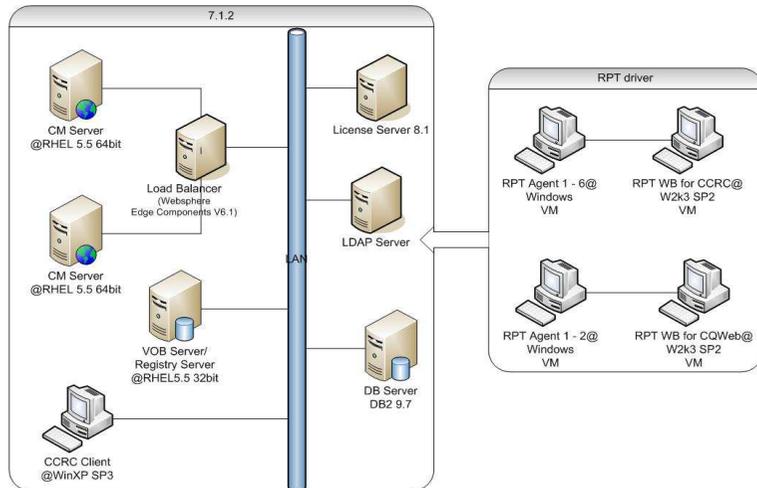
ModCounter.java in Workspace Test\_Automation Stream Workspace - ClassicCD\_Co

Merges	Comment	Creator	Date Created
	10: script development...	dan	8:48:46 AM (3 mi...
	Share projects	dan	Aug 19, 2010 8:41 PM

# 实例：负载均衡的CM Server可靠性测试

## 性能目标

- ✓ CCRC及CQWeb操作的成功率>95%
- ✓ 平均响应时间 (除初始加载)<20 sec
- ✓ 负载分配合理
- ✓ 测试过程没有出现显著内存泄漏迹象
- ✓ 没有服务器崩溃或重启等现象



渐进式达到系统最高容量并给出调优建议

# 总结



- 生产环境的性能表现决定企业应用的成败
- 性能保障是系统工程，需要整个团队配合：  
及早测试，人人测试
- **IBM Rational**性能工程解决方案提供强大的团队协作平台和工具紧密集成，帮助企业达成性能目标
- **RPT**在企业性能测试中扮演重要角色



## 参考链接

- [Rational Performance Tester Product Page](#)
- [RPT Online Help](#)
- [DeveloperWorks: Rational Performance Testing Forum](#)
- [DeveloperWorks: Integration Rational Team Concert with Rational Performance Tester](#)
- [Rational RFE Community](#)



# Questions





**Learn more at:**

- [IBM Rational software](#)
- [Rational launch announcements](#)
- [Rational Software Delivery Platform](#)
- [Accelerate change & delivery](#)
- [Deliver enduring quality](#)
- [Enable enterprise modernization](#)
- [Ensure Web security & compliance](#)
- [Improve project success](#)
- [Manage architecture](#)
- [Manage evolving requirements](#)
- [Small & mid-sized business](#)
- [Targeted solutions](#)
- [Rational trial downloads](#)
- [developerWorks Rational](#)
- [Leading Innovation](#)
- [IBM Rational TV](#)
- [IBM Business Partners](#)
- [IBM Rational Case Studies](#)

© Copyright IBM Corporation 2010. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.