



IBM Software Group

Solución de Calidad del Software

Automatización Pruebas de Carga con IBM Rational Performance Tester y Tivoli Monitoring for Transaction Performance



Ana López-Manicisidor Rueda
ana.lopez@es.ibm.com

Javier Llorente
jlllorente@es.ibm.com

@business on demand software

Agenda

- Aseguramiento contínuo de la calidad



Aseguramiento continuo de la calidad

- ✓ Construcción con calidad desde el principio
- ✓ Automatización de pruebas
- ✓ Aceleración de los ciclos de pruebas



Beneficios Negocio

- Más productividad, eficiencia, competitividad, flexibilidad y capacidad de respuesta
- Aseguramiento de los niveles de satisfacción



Capacidades

- Gestión y automatización de pruebas:
 - Unitarias
 - Rendimiento
 - Funcionales
 - Carga

Plataforma de Desarrollo de Software de IBM



Analista

**Rational
RequisitePro**



Arquitecto

**Rational
Software
Modeler**

**Rational
Software
Architect**



Desarrollador

**Rational Web
Developer**

**Rational
Application
Developer**

**Rational
Software
Architect**



Ing.de pruebas

**Rational Manual
Tester**

**Rational
Funcional Tester**

**Rational
Performance
Tester**

ECLIPSE / Integración .NET



**Director de
Proyecto**

Rational Unified Process
Rational ClearCase Rational ClearQuest Rational Project Console



**Directores
ejecutivos**

Rational Portfolio Manager

Agenda

- Aseguramiento contínuo de la calidad
- ¿Por qué realizar pruebas de carga?



¿Qué son las pruebas de carga?

- El proceso de estresar una aplicación emulando una carga masiva de usuarios virtuales para medir los tiempos de respuesta y validar su escalabilidad antes de su paso a producción



¿Por qué necesitamos realizar pruebas de carga?

- Porque un fallo en cualquier punto de nuestro sistema hace que los usuarios no estén disfrutando del servicio adecuado, en un tiempo adecuado, y se irán a otro proveedor

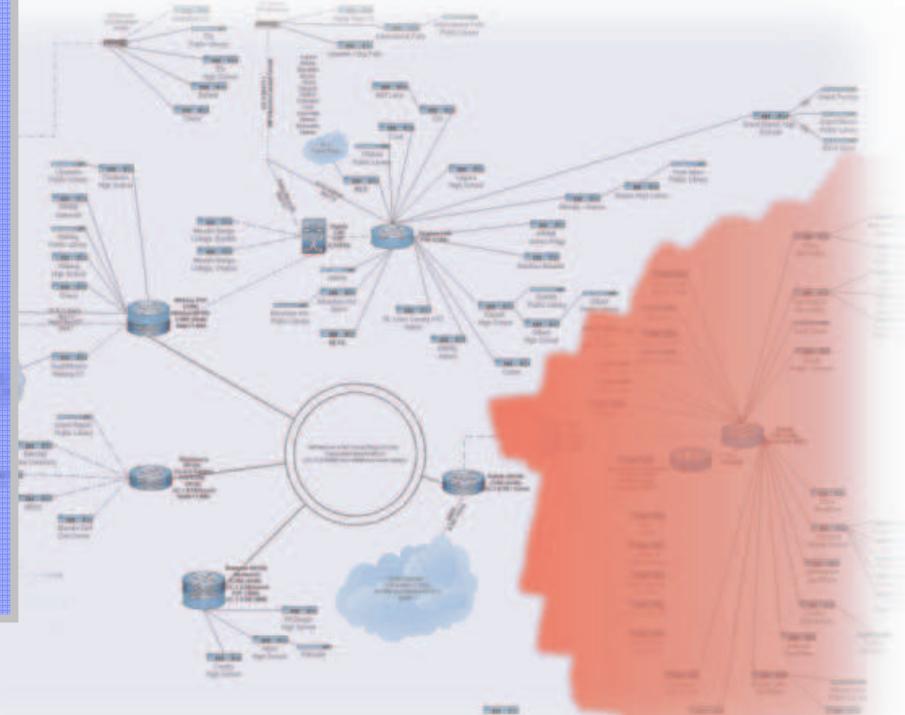
The page cannot be displayed

There is a problem with the page you are trying to reach and it cannot be displayed.

Please try the following:

- Click the [Refresh](#) button, or try again later.
- Open the home page, and then look for links to the information you want.

HTTP 500.13 - Server too busy.



Sistema bajo Prueba

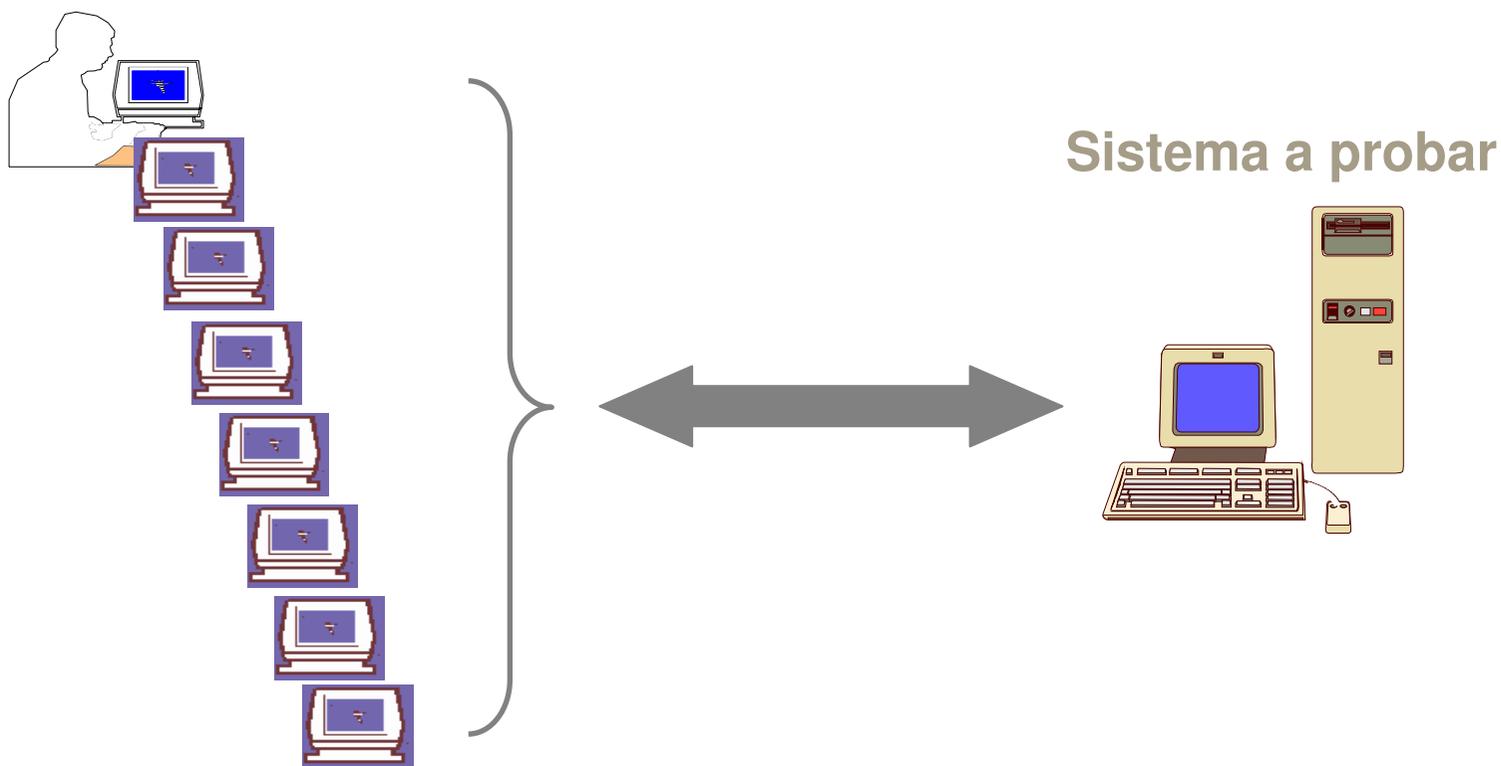


Estrategias para realizar las pruebas de carga

- Pruebas basadas en HW
- Pruebas basadas en SW
- Combinar pruebas basadas en HW y SW

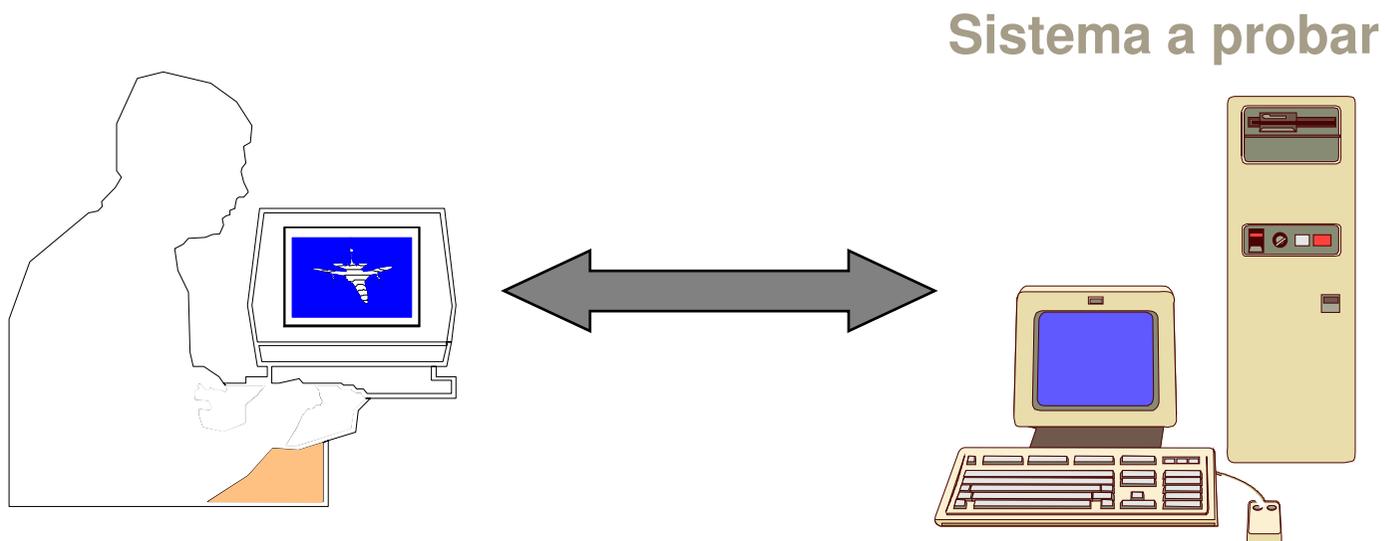
Pruebas basadas en HW

- Se necesitan físicamente varios ordenadores
 - ▶ Uno por cada usuario a generar
- Un probador

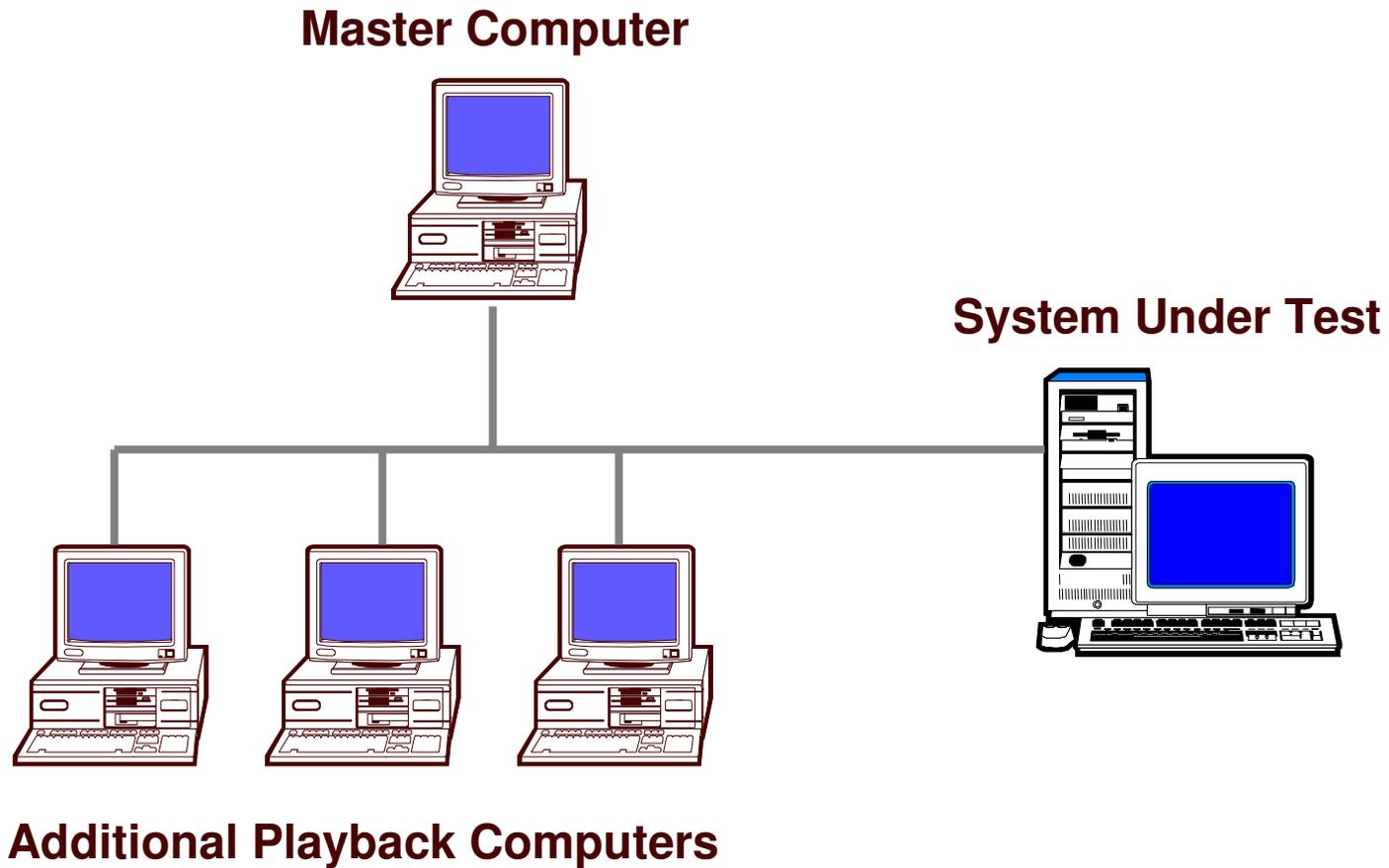


Pruebas basadas en SW

- Al menos un ordenador simulando varios usuarios virtuales
- Un probador



Combinar Pruebas basadas en HW y SW



Problemas en las Pruebas de Carga

- Herramientas complicadas
 - ▶ La complejidad de las herramientas y la falta de experiencia intimida a la mayoría de los usuarios la primera vez



“No tenemos mucha experiencia construyendo pruebas de carga.”

“La mayoría de las herramientas de pruebas de carga que hemos utilizado son demasiado complicadas”

Problemas en las Pruebas de Carga

- Herramientas complicadas
 - ▶ La complejidad de las herramientas y la falta de experiencia intimida a la mayoría de los usuarios la primera vez
- Herramientas no flexibles
 - ▶ Los usuarios están limitados por la funcionalidad de las herramientas



“No conocemos bien que está ocurriendo durante una prueba de carga”

“Necesitamos contratar a personas con altos conocimientos técnicos para adaptar las pruebas grabadas automáticamente.”

Problemas en las Pruebas de Carga

- Herramientas complicadas
 - ▶ La complejidad de las herramientas y la falta de experiencia intimida a la mayoría de los usuarios la primera vez
- Herramientas no flexibles
 - ▶ Los usuarios están limitados por la funcionalidad de las herramientas
- Altos costes de HW
 - ▶ Las herramientas consumen muchos recursos y no permiten la comunicación entre diferentes personas



“Los desarrolladores no tienen forma de realizar las pruebas de carga.”

“No tenemos hardware suficiente para realizar las pruebas de carga.”

Agenda

- Aseguramiento contínuo de la calidad
- ¿Por qué realizar pruebas de carga?
- La nueva solución de IBM en Eclipse:
 - ▶ IBM Rational Performance Tester



Automatización de Pruebas de Carga con IBM Rational Performance Tester



■ IBM Rational Performance Tester

- ▶ Automatización de Pruebas de Carga de Aplicaciones Web

■ Principales beneficios

- ▶ Productividad desde el Primer Día
 - No necesita programación
 - Sencilla presentación de la navegación en páginas
- ▶ Simulación de escenarios reales
 - Variación dinámica y automática de datos y perfiles
- ▶ Menor coste
 - Menos requisitos memoria
 - Basada en estándares (Eclipse, Hyades, Java)
- ▶ Flexible
 - Posible modificación de las pruebas generadas

Creación de una prueba de carga

Es un proceso de 3 etapas



Grabar pruebas

- Capturan la interacción del usuario con el servidor, grabando el tráfico http/https entre cliente y servidor
- Como resultado de la grabación se genera una prueba que muestra en forma de árbol la secuencia de navegación de páginas
- Correla automáticamente las respuestas del servidor y permite variar los datos introducidos por el usuario

Grabación y visualización de pruebas

The screenshot displays the IBM Rational Software Development Platform interface for configuring a performance test. The main window is titled "Performance Test - OrangeCountyChoppers".

- Test Navigator:** Shows a tree view of the test structure, including "PerformanceTester", "bin", and "src" folders.
- Test Contents:** A tree view showing the test contents, including "OrangeCountyChoppers", "Orange County Choppers", and "www.orangecountychoppers.com/".
- Test Element Details:** A panel showing details for the selected test element, including "HTTP Page - Orange County Choppers", "Page title", "Primary request", "Think time", and "Test Data".
- Protocol Data:** A window showing the request and response details, including "Request", "Response Headers", "Response Content", and "Browser".

Navegador

Acceso a las pruebas y planificación de ejecuciones

Info Adicional

Información detallada de las peticiones y respuestas View detailed info

Prueba

Vista en forma de árbol para cada página. No código.

Organización

Las ramas superiores son las páginas. Asociados y colgados a cada página están los elementos, como imágenes, estilos, scripts java...

Creación de una prueba de carga

Es un proceso de 3 etapas



Grabar Pruebas Crear escenario de carga

- Crear un escenario lo más real posible:
 - ▶ Simulación de diferentes perfiles de usuarios
 - ▶ Secuencia y orden de ejecución de los scripts
 - ▶ Variación de datos de entrada
 - ▶ Introducir espaciadamente la carga de usuarios
- Ejecución distribuída en varias máquinas

Creación de un escenario real de carga

Performance Schedule - Browse and Buy

Schedule Contents
This section shows the schedule contents

Schedule Element Details
Performance Schedule - "Browse and Buy"

Schedule name:

User Load

Number of users:

Add a delay between starting each user

Delay:

Stop running the schedule after an elapsed time

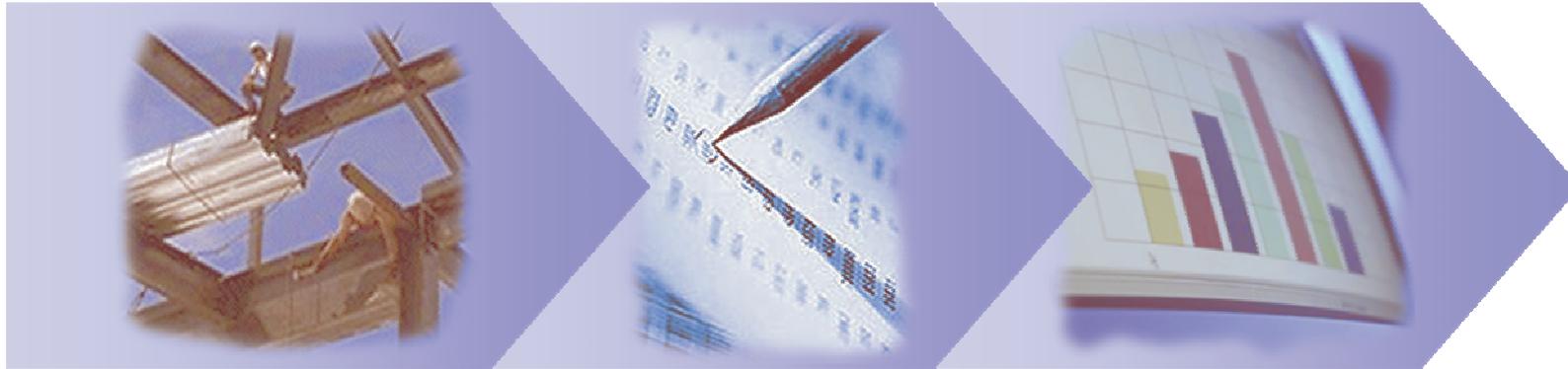
Stop after:

Think Time

Modify the duration of think time delays:

Creación de una prueba de carga

Es un proceso de 3 etapas



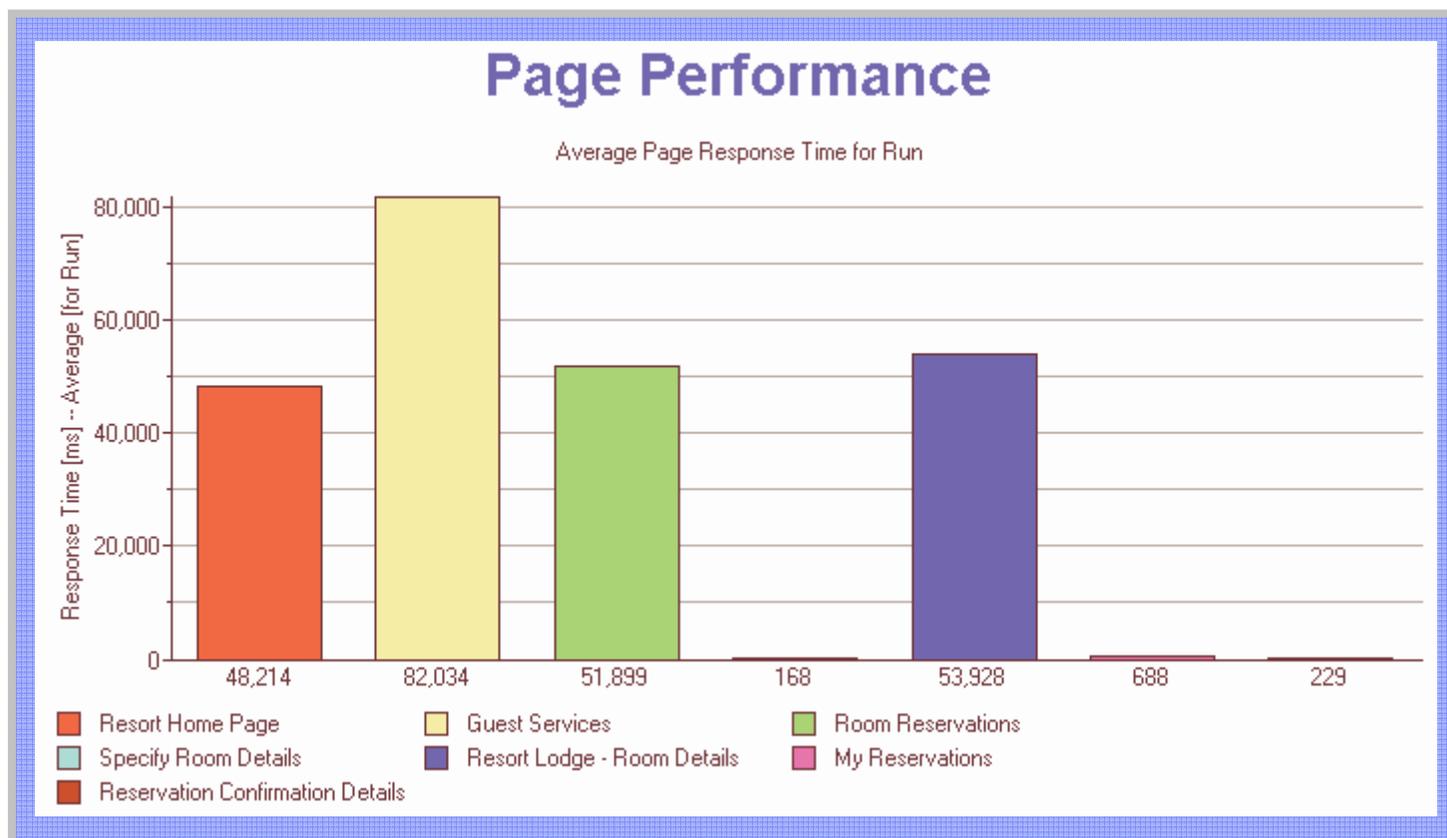
Grabar Scripts

**Crear escenario
de carga**

Ejecutar y Analizar

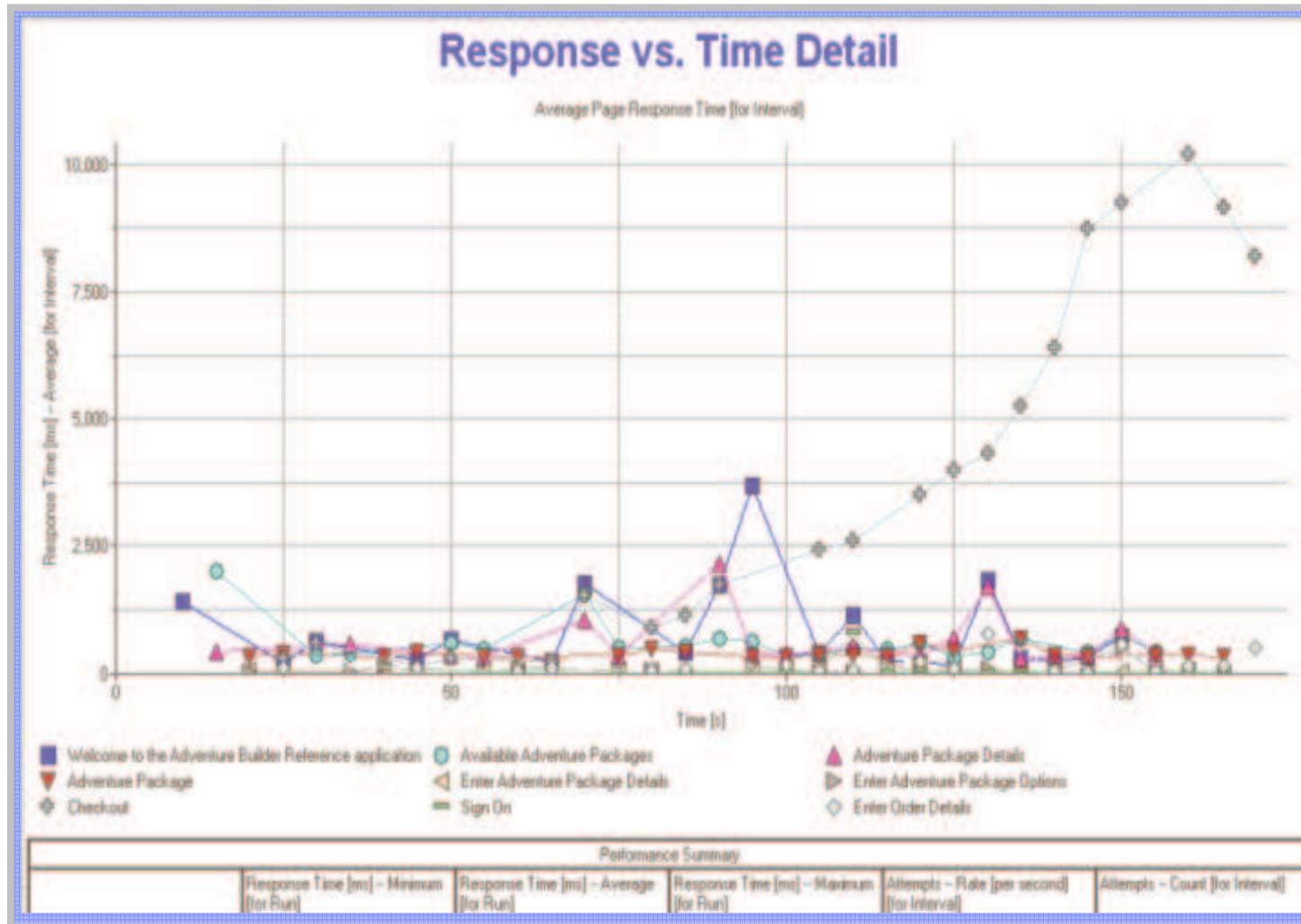
- Monitorización en tiempo real de los resultados
 - ▶ Detección de cuellos de botella
 - ▶ Detección de las páginas más lentas
- Tiempos desde el punto de vista del cliente
- Analizar la máxima carga de usuarios concurrentes que soporta la aplicación web

Análisis de los resultados



- Determinar fácilmente las páginas más lentas de la aplicación web

Degradación del sistema durante la prueba



DEMO

- IBM Rational Performance Tester v6.1
 - ▶ Creación de un script de prueba
 - ▶ Creación de un escenario de carga
 - ▶ Ejecución del escenario de carga creado
 - ▶ Análisis de los tiempos de respuesta

Agenda

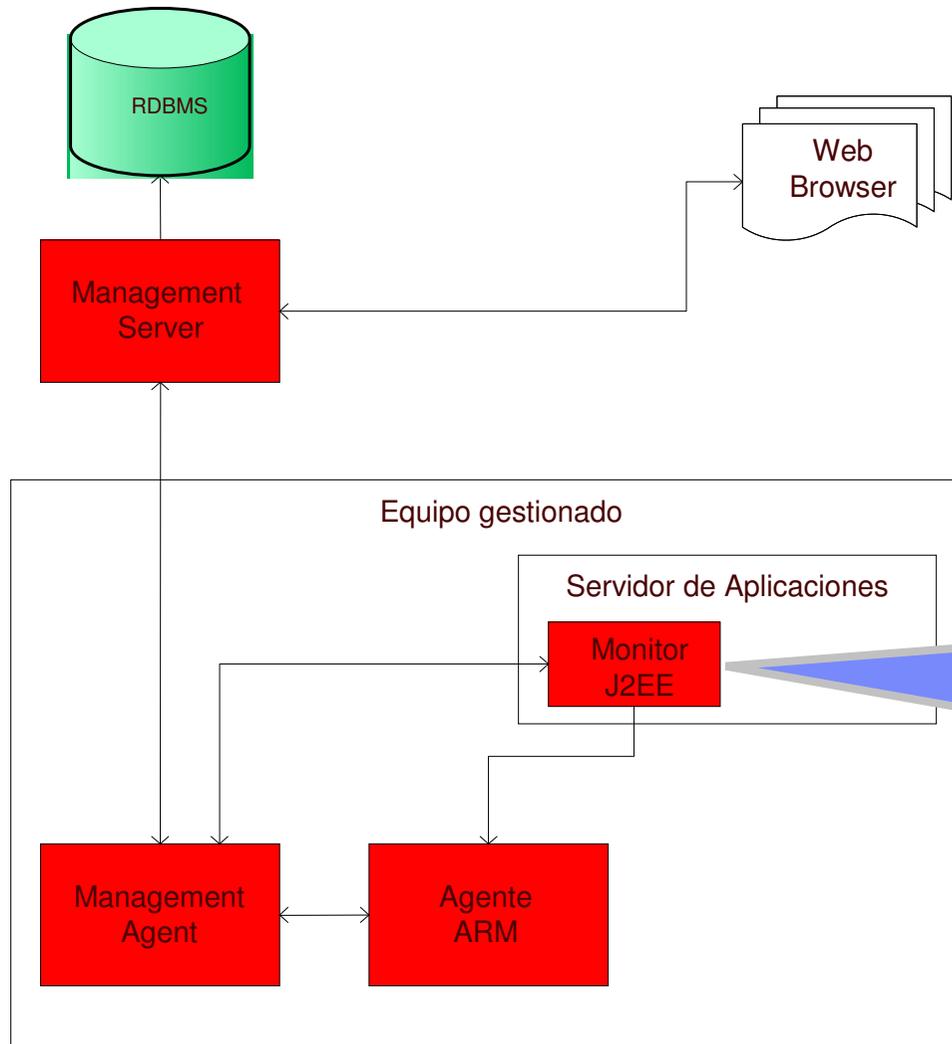
- Aseguramiento contínuo de la calidad
- ¿Por qué realizar pruebas de carga?
- La nueva solución de IBM en Eclipse:
 - ▶ IBM Rational Performance Tester
- ¿Cómo podemos identificar qué es lo que está causando el problema de rendimiento?



Estresar el servidor no es suficiente



Monitorización de Web Services usando ARM y TMTP



Monitorización de Web Services de WAS con métricas de solicitudes

Vista topológica de TMAP con datos de una petición de web services

The screenshot displays the IBM Tivoli Monitoring for Transaction Performance interface. The top window shows a topological view of a transaction flow. The flow starts with a JSP component (0.244) calling the createNewWorkOrder() method (0.237) of the WorkOrderManager component. This component then calls the Web Services Provider (0.003).

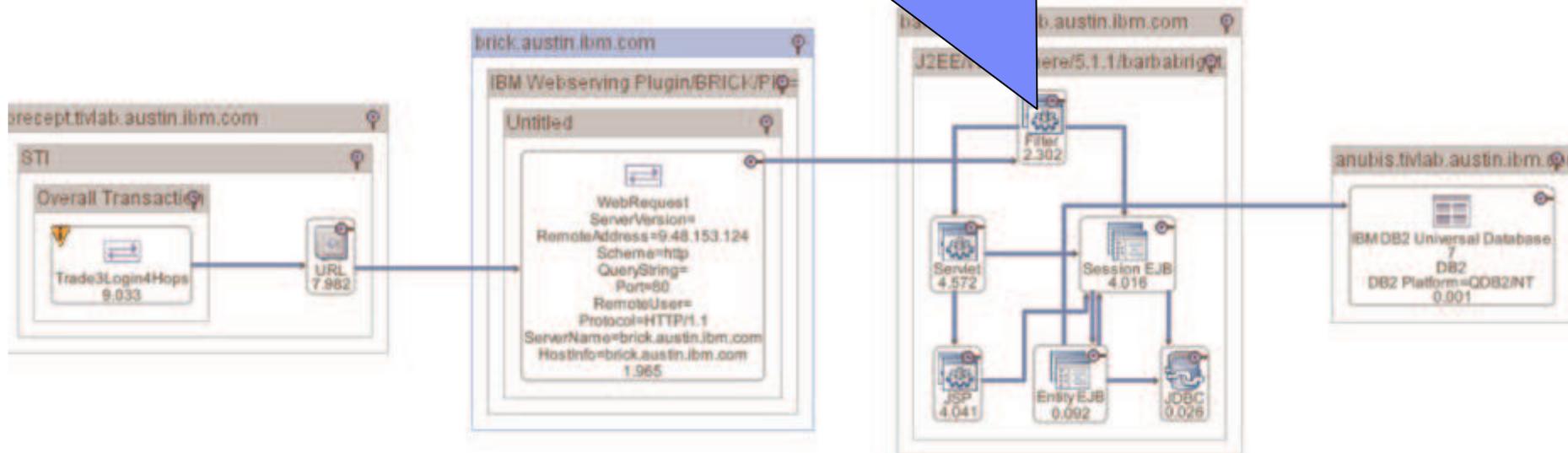
The Inspector window below shows the details for the createNewWorkOrder() operation. The table below lists the properties and values for this transaction.

Property	Value
Name	wsrequestor:WorkOrderManager.createNewWorkOrder?transport=http¶meters=customerName,addressStreet,addressCity,addressState,addressZip,sourceCompany,appointmentDate
Transport Name	http
Status	Most Degraded
Response message	<soapenv:Body xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance"><createNewWorkOrderResponse xmlns="http://installco.com"><createNewWorkOrderReturn>28</createNewWorkOrderReturn></createNewWorkOrderResponse></soapenv:Body>
WSDL Port Name	WorkOrderManager
Start Time	2004-08-25 16:28:35
User Name	root
Target endpoint address	http://localhost:9080/WorkOrderManager/fc/services/WorkOrderManager
Average Duration	0.081
Parameter queue name list	customerName,addressStreet,addressCity,addressState,addressZip,sourceCompany,appointmentDate
Duration	0.237
Service reference name	service
WSDL Operation Name	createNewWorkOrder

Tivoli Monitoring for Transaction Performance

Identificación rápida de problemas

Con un click se pueden identificar problemas encontrando recursos que actúan de forma diferente de lo normal



Cliente

Servidores Web

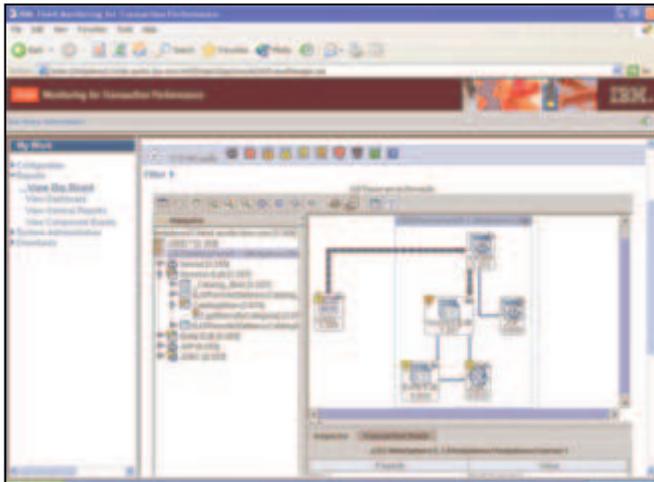
Servidores de Aplicaciones

Base de Datos

Tivoli Monitoring for Transaction Performance

Integración con Rational Application Developer

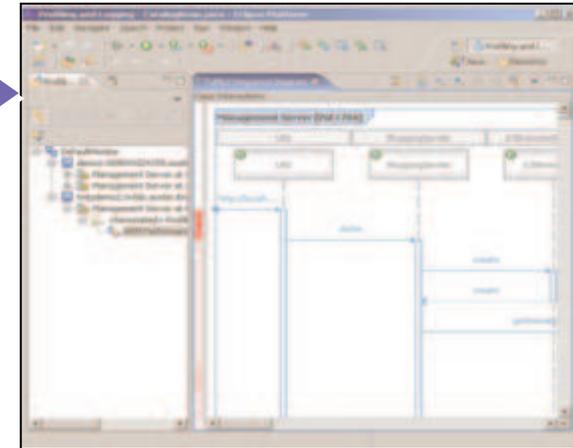
Vista de operaciones de Tivoli



Unión de Operaciones
y Desarrollo

Datos de producción
enviados a las
herramientas de
desarrollo

Vista de desarrollador Rational (Problem Resolution toolkit)



- Reduce en gran medida el tiempo necesario para resolver problemas facilitando los pasos de recreación de los mismos
- Operaciones y desarrollo tienen una vista común del problema
- Resuelve problemas de configuración o con código complejo usando datos reales de producción

Beneficios

- Más valor al negocio
 - ▶ Más productividad
 - ▶ Más eficiencia
 - ▶ Más competitividad
 - ▶ Más flexibilidad
 - ▶ Más capacidad de respuesta
- Solución para todo tipo de empresas
- Aseguramiento de los niveles de satisfacción del cliente



Questions

Thank
YOU