

Probador Certificado del ISTQB®

Programa de Estudio de Nivel Básico Especialidad – Prueba de Rendimiento

Versión 2018

Aportado por
American Software Testing Qualifications Board y
German Testing Board



Traducción realizada por el
Spanish Software Testing Qualifications Board
Versión ES 001.08

Basada en el Programa de Estudio
“Certified Tester Foundation Level Specialist Syllabus, Performance
Testing, Version 2018”

International Software Testing Qualifications Board



Nota sobre Derechos de Propiedad Intelectual.

Este documento puede ser copiado en su totalidad, o se pueden hacer extractos, si se reconoce la fuente.

Nota sobre derechos de propiedad intelectual © International Software Testing Qualifications Board (en adelante denominado ISTQB®). ISTQB es una marca registrada del International Software Testing Qualifications Board.

Grupo de Trabajo de Prueba de Rendimiento (“Performance Testing Working Group”):

Graham Bath
Rex Black
Alexander Podelko
Andrew Pollner
Randy Rice

Historial de Revisiones

Versión	Fecha	Observaciones
Alpha V04	13 de diciembre de 2016	Versión para la reunión de la ciudad de Nueva York
Alpha V05	18 de diciembre de 2016	Después de la reunión de la ciudad de Nueva York
Alpha V06	23 de diciembre de 2016	Reestructuración Capítulo 4 Renumeración y ajuste de las oficinas de enlace según lo acordado en la ciudad de Nueva York
Alpha V07	31 diciembre 2016	Añadir comentarios del autor
Alpha V08	12 de febrero de 2017	Versión prealfa
Alpha V09	16 de abril de 2017	Versión prealfa
Alpha Review V10	28 de junio de 2017	Para la Revisión Alpha
V2017v1	27 de noviembre de 2017	Para la revisión de Alpha
V2017v2	15 Diciembre 2017	Actualizaciones de Alpha
V2017v3	15 de enero de 2018	Edición técnica
V2017v4	23 de enero de 2018 Revisión del glosario	Revisión del glosario
V2018 b1	1 de marzo de 2018	Beta candidata para el ISTQB
V2018 b2	17 de mayo de 2018	Lanzamiento de la versión beta para ISTQB
V2018 b3	25 de agosto de 2018	Comentarios de revisión beta incorporados para la versión de lanzamiento
Version 2018	9 de diciembre de 2018	ISTQB GA Release

Tabla de Contenidos

Nota sobre Derechos de Propiedad Intelectual.....	2
Historial de Revisiones.....	3
Tabla de Contenidos.....	4
Agradecimientos.....	6
Notas de la Versión en Idioma Español.....	7
Traducciones Específicas, Definiciones, Acrónimos, Abreviaturas y Notas.....	8
0 Introducción.....	10
0.1 Objetivo de este Programa de Estudio.....	10
0.2 El Nivel Básico de Probador Certificado en la Prueba de Rendimiento.....	10
0.3 Resultados de Negocio.....	11
0.4 Objetivos de Aprendizaje Evaluables.....	11
0.5 Tiempos Lectivos Recomendados.....	12
0.6 Criterios de Admisión.....	12
0.7 Fuentes de Información.....	12
1 Conceptos Básicos.....	13
1.1 Fundamentos de la Prueba de Rendimiento.....	14
1.2 Tipos de Prueba de Rendimiento.....	16
1.3 Tipos de Prueba en la Prueba de Rendimiento.....	18
1.3.1 Prueba Estática.....	18
1.3.2 Prueba Dinámica.....	18
1.4 El Concepto de Generación de Carga.....	20
1.5 Modos de Fallo de Rendimiento Comunes y sus Causas.....	22
2 Fundamentos de la Medición del Rendimiento.....	24
2.1 Métricas Características Recopiladas en la Prueba de Rendimiento.....	25
2.1.1 Por qué son Necesarias las Métricas de Rendimiento.....	25
2.1.2 Recopilación de Mediciones y Métricas de Rendimiento.....	25
2.1.3 Selección de las Métricas de Rendimiento.....	27
2.2 Agregación de los Resultados de la Prueba de Rendimiento.....	28
2.3 Fuentes Clave de las Métricas de Rendimiento.....	29
2.4 Resultados Característicos de una Prueba de Rendimiento.....	31
3 La Prueba de Rendimiento en el Ciclo de Vida del Software.....	32
3.1 Principales Actividades de la Prueba de Rendimiento.....	33
3.2 Categorías de Riesgos de Rendimiento para Diferentes Arquitecturas.....	35
3.3 Riesgos de Rendimiento a lo largo del Ciclo de Vida de Desarrollo de Software.....	38
3.4 Actividades de la Prueba de Rendimiento.....	40
4 Tareas de la Prueba de Rendimiento.....	43
4.1 Planificación.....	45
4.1.1 Deducir Objetivos de la Prueba de Rendimiento.....	45
4.1.2 El Plan de Prueba de Rendimiento.....	45
4.1.3 Comunicar sobre la Prueba de Rendimiento.....	50
4.2 Análisis, Diseño e Implementación.....	52
4.2.1 Protocolos de Comunicación Habituales.....	52
4.2.2 Transacciones.....	53
4.2.3 Identificación de Perfiles Operativos.....	53
4.2.4 Construcción de Perfiles de Carga.....	55
4.2.5 Análisis de la Capacidad de Procesamiento y Concurrencia.....	57
4.2.6 Estructura Básica de un Guión para una Prueba de Rendimiento.....	58

4.2.7	Implementación de Guiones de Prueba de Rendimiento	59
4.2.8	Preparación para la Ejecución de la Prueba de Rendimiento	60
4.3	Ejecución	63
4.4	Analizar Resultados e Informar	65
5	Herramientas	69
5.1	Soporte de Herramientas.....	70
5.2	Adecuación de las Herramientas.....	71
6	Referencias	73
6.1	Estándares.....	73
6.2	Documentos del ISTQB.....	73
6.3	Bibliografía.....	73



Agradecimientos

Este documento fue desarrollado por American Software Testing Qualifications Board (ASTQB) y German Testing Board (GTB):

Graham Bath (GTB, copresidente del Grupo de Trabajo)

Rex Black

Alexander Podelko (CMG)

Andrew Pollner (ASTQB, Working Group co-chair)

Randy Rice

El equipo principal agradece al equipo de revisión por sus sugerencias y aportes. El ASTQB desea reconocer y agradecer al Computer Measurement Group (CMG) por sus contribuciones en el desarrollo de este programa de estudio.

Las siguientes personas participaron en la revisión, los comentarios o la votación de este programa de estudio o de sus predecesores:

Dani Almog

Sebastian Chece

Todd DeCapua

Wim Decoutere

Frans Dijkman

Jiangru Fu

Matthias Hamburg

Ágota Horváth

Mieke Jungeblood

Beata Karpinska

Gábor Ladányi

Kai Lepler

Ine Lutterman

Marek Majernik

Stefan Massonet

Judy McKay

Gary Mogyorodi

Joern Muenzel

Petr Neugebauer

Ingvar Nordström

Meile Posthuma

Michaël Pilaeten

Filip Rechteris

Adam Roman

Dirk Schweier

Marcus Seyfert

Péter Sótér

Michael Stahl

Jan Stiller

Federico Toledo

Andrea Szabó

Yaron Tsubery

Stephanie Ulrich

Mohit Verma

Armin Wachter

Huaiwen Yang

Ting Yang

Este documento fue publicado formalmente por el ISTQB el 9 de diciembre de 2018.

Notas de la Versión en Idioma Español

El Spanish Software Testing Qualifications Board (SSTQB) ha llevado a cabo la traducción del Programa de Estudio de "ISTQB® Certified Tester, Foundation Level, Performance Testing, 2018". Este Programa de Estudio se denomina, en idioma español, "Probador Certificado del ISTQB®, Nivel Básico, Prueba de Rendimiento, versión 2018".

El equipo de traducción y revisión para este programa de estudio es el siguiente (por orden alfabético):

Participación	Nombre y Apellidos	Organización	País
Revisor	Aurelio Gandarillas	MTP	España
Responsable de la traducción	Gustavo Márquez Sosa	SSTQB	España

El Comité Ejecutivo del SSTQB agradece especialmente las aportaciones de los revisores.

En una siguiente versión se podrán incorporar aportaciones adicionales. El SSTQB considera conveniente mantener abierta la posibilidad de realizar cambios en el "Programa de Estudio".

Madrid, 13 de julio de 2019

Traducciones Específicas, Definiciones, Acrónimos, Abreviaturas y Notas

En este capítulo se presentarán traducciones específicas, definiciones, acrónimos, abreviaturas y notas de términos y conceptos que no forman parte del Glosario de Términos del ISTQB y tampoco están incluidos en su traducción al idioma español o castellano. Se ha incorporado este apartado para facilitar la lectura y comprensión de esta “Programa de Estudio”.

Las “Traducciones Específicas, Definiciones, Acrónimos, Abreviaturas y Notas” de este apartado están identificados con el texto **Consultar **** en el pie de página. Sólo se identificará la primera ocurrencia del elemento.

Es conveniente observar que las traducciones de los distintos términos, sean o no del glosario, se han realizado con el objetivo de evitar conflictos en sus traducciones y siguiendo las normas de traducción del SSTQB. Algunos conflictos surgen en el momento de traducir un nuevo término o un nuevo programa de estudio.

Traducciones específicas		
Español	Inglés	Observaciones
agotamiento de la memoria	memory exhaustion	No hay observaciones.
ataque de denegación de servicio	denial of service attack	No hay observaciones.
ataque de denegación de servicio distribuido	distributed denial of service attack	No hay observaciones.
autoescalado	self-scaling	No hay observaciones.
autónomo	stand-alone	No hay observaciones.
balanceador de carga	load balancer	No hay observaciones.
bloqueo mutuo	deadlock	No hay observaciones.
capacidad	capacity	No hay observaciones.
capacidad de procesamiento	throughput	No hay observaciones.
cola	queue	No hay observaciones.
comportamiento temporal	time behavior	No hay observaciones.
entorno anfitrión	host environment	No hay observaciones.
fallo total del sistema	system crash	No hay observaciones.
falsificación de IP	IP spoofing	No hay observaciones.
fuentes de suministro	vendor sources	No hay observaciones.
hilo inactivo	idle thread	No hay observaciones.
hilo ocupado	busy thread	No hay observaciones.
impactos por segundo	hits per second	No hay observaciones.
información de perfilado	profiling information	No hay observaciones.
Infraestructura como un Servicio	Infrastructure as a Service	No hay observaciones.
Internet de las Cosas	Internet Of Things	No hay observaciones.
memoria intermedia	buffer	No hay observaciones.
multihilo	multi-threading	No hay observaciones.

Traducciones específicas		
Español	Inglés	Observaciones
paginación de la memoria	memory paging	No hay observaciones.
perfilado del rendimiento	performance profiling	No hay observaciones.
problema de bloqueo mutuo	deadlock problem	No hay observaciones.
problema de multihilo	multi-threading problem	No hay observaciones.
prueba multitudinaria	crowd testing	No hay observaciones.
punto crítico	breaking point	No hay observaciones.
punto de presencia	point of presence	No hay observaciones.
segundo plano	background	No hay observaciones.
sistema informático individual	single computer system	No hay observaciones.
Software como un Servicio	Software as a Service	No hay observaciones.
tasa de transferencia de la red	network throughput	No hay observaciones.
teléfono inteligente	smartphone	No hay observaciones.
utilización de recursos	resource utilization	No hay observaciones.
velocidad de procesamiento	throughput rate	No hay observaciones.

Tabla 1 - Traducciones Específicas

Definiciones, Acrónimos y Abreviaturas			
Tipo de Término	Idioma del Término	Término	Descripción
ACRÓNIMO	Inglés	IaaS	Infrastructure as a Service
ACRÓNIMO	Inglés	SaaS	Software as a Service
ACRÓNIMO	Inglés	IoT	Internet Of Things

Tabla 2 – Definiciones, Acrónimos y Abreviaturas

Notas
"persona" es un modelo descriptivo de los usuarios que se utiliza en "Diseño de la Interacción".

Tabla 3 – Notas

0 Introducción

0.1 Objetivo de este Programa de Estudio

Este programa de estudio constituye la base para la formación como Probador Certificado del ISTQB® de Nivel Básico. El ASTQB® y GTB® proporcionan este programa de estudio en los siguientes términos:

1. A los Comités Miembro, para traducir a su idioma local y para acreditar a los proveedores de formación. Los comités miembro pueden adaptar el programa de estudio a sus necesidades lingüísticas particulares y añadir referencias para adaptarlo a sus publicaciones locales.
2. A los Comités de Examen, para elaborar las preguntas del examen en su lengua local adaptadas a los objetivos de aprendizaje de este programa de estudio.
3. A los proveedores de formación, para desarrollar material didáctico y determinar los métodos de enseñanza adecuados.
4. A los candidatos a la certificación, para que preparen el examen de certificación (ya sea como parte de un curso de formación o de forma independiente).
5. A la comunidad internacional de ingeniería de software y sistemas, para avanzar en la profesión de prueba del software y sistemas, y como fuente de libros y artículos.

El ASTQB y el GTB pueden permitir que otras entidades utilicen este programa de estudio para otros fines, siempre y cuando soliciten y obtengan permiso previo y por escrito.

0.2 El Nivel Básico de Probador Certificado en la Prueba de Rendimiento

El Nivel Básico está dirigido a cualquier persona que participe en la prueba de software y desee ampliar sus conocimientos sobre la prueba de rendimiento o a cualquier persona que desee iniciar una carrera de especialista en prueba de rendimiento. La capacitación también está dirigida a cualquier persona involucrada en la ingeniería del rendimiento que desee obtener una mejor comprensión de la prueba de rendimiento.

El programa de estudios considera los siguientes aspectos principales de la prueba de rendimiento:

- Aspectos técnicos.
- Aspectos metodológicos.
- Aspectos organizativos.

La información sobre la prueba de rendimiento descrita en el programa de estudio de Probador Certificado ISTQB® de Nivel Avanzado, Analista de Pruebas Técnicas [ISTQB_ALTTA_SYL] es consistente con este programa de estudios y se desarrolla a través de este programa de estudios.

0.3 Resultados de Negocio

Esta sección enumera los resultados de negocio esperados de un candidato que haya obtenido la certificación de Nivel Básico de Prueba de Rendimiento.

NBPR-1 Entender los conceptos básicos de eficiencia de desempeño y prueba de rendimiento.

NBPR-2 Definir riesgos de rendimiento, objetivos y requisitos para satisfacer las necesidades y expectativas de los implicados.

NBPR-3 Entender las métricas de rendimiento y cómo recogerlas.

NBPR-4 Desarrollar un plan de prueba de rendimiento para alcanzar los objetivos y requisitos establecidos.

NBPR-5 Diseñar, implementar y ejecutar conceptualmente pruebas básicas de rendimiento.

NBPR-6 Analizar los resultados de una prueba de rendimiento y establecer las consecuencias para distintos implicados.

NBPR-7 Explicar el proceso, la justificación, los resultados y las implicaciones de las pruebas de rendimiento a los diferentes implicados.

NBPR-8 Entender las categorías y usos de las herramientas de rendimiento y los criterios para su selección.

NBPR-9 Establecer cómo las actividades de pruebas de rendimiento se alinean con el ciclo de vida del software.

0.4 Objetivos de Aprendizaje Evaluables

Los Objetivos de Aprendizaje son la base de los Resultados de Negocio y se utilizan para crear el examen de Certificación de Prueba de Rendimiento de Nivel Básico (Foundation Level Performance Testing Certification). Los objetivos de aprendizaje tienen asignado un nivel de conocimiento cognitivo o nivel K (K-Level por su nombre en inglés).

Un nivel K, o nivel cognitivo, se utiliza para clasificar los objetivos de aprendizaje de acuerdo con la taxonomía revisada de Bloom [Anderson01]. El ISTQB® utiliza esta taxonomía para diseñar los exámenes de sus programas de estudios.

Este programa contempla cuatro niveles K diferentes (K1 a K4):

Nivel K	Palabra Clave	Descripción
1	Recordar	El candidato debe recordar o reconocer un término o un concepto.
2	Comprender	El candidato debe seleccionar una explicación para un enunciado relacionado con el tema de la pregunta.
3	Aplicar	El candidato debe seleccionar la aplicación correcta de un concepto o técnica y aplicarla a un contexto determinado.
4	Analizar	El candidato puede separar la información relacionada con un procedimiento o técnica en sus partes constituyentes para una mejor comprensión y puede distinguir entre hechos e inferencias.

En general, todas las partes de este programa de estudio son evaluables en el nivel K1. Es decir, el candidato reconocerá, recordará y recordará un término o concepto. Los objetivos de aprendizaje en los niveles K2, K3 y K4 se muestran al principio del capítulo correspondiente.

0.5 Tiempos Lectivos Recomendados

Se ha definido un tiempo lectivo mínimo para cada uno de los objetivos de aprendizaje de este programa de estudio. El tiempo total de cada capítulo se indica en el encabezado del capítulo.

Los proveedores de formación deben tener en cuenta que otros programas de estudio del ISTQB aplican un enfoque de "tiempo estándar" que asigna tiempos fijos de acuerdo con el nivel K. El programa de estudio de Prueba de Rendimiento no aplica estrictamente este esquema. Como resultado, los proveedores de formación reciben una indicación más flexible y realista de los tiempos mínimos de formación para cada objetivo de aprendizaje.

0.6 Criterios de Admisión

Se debe obtener el certificado de Nivel Básico Troncal antes de realizar el examen de certificación de Prueba de Rendimiento de Nivel Básico.

0.7 Fuentes de Información

Los términos utilizados en el programa están definidos en el Glosario de términos utilizados en la prueba de software [ISTQB_GLOSSARY] del ISTQB.

La sección 6 contiene una lista de libros y artículos recomendados sobre la prueba de rendimiento.

1 Conceptos Básicos

Duración: 60 minutos

Palabras Clave

prueba de capacidad (“capacity testing”)
prueba de concurrencia (“concurrency testing”)
eficiencia (“efficiency”)
prueba de resistencia (“endurance testing”)
generación de carga (“load generation”)
prueba de carga (“load testing”)
prueba de rendimiento (“performance testing”)
prueba de escalabilidad (“scalability testing”)
prueba de picos (“spike testing”)
prueba de estrés (“stress testing”)

Objetivos de Aprendizaje para Conceptos Básicos:

1.1 Principios y Conceptos

NBPR-1.1.1 (K2) Comprender los principios de la prueba de rendimiento.

1.2 Tipos de Prueba de Rendimiento

NBPR-1.2.1 (K2) Entender los diferentes tipos de pruebas de rendimiento.

1.3 Tipos de Prueba en la Prueba de Rendimiento

NBPR-1.3.1 (K1) Recordar los tipos de prueba en la prueba de rendimiento.

1.4 El concepto de Generación de Carga

NBPR-1.4.1 (K2) Entender el concepto de generación de carga.

1.5 Fallos Habituales en la Prueba de Rendimiento y sus Causas

NBPR-1.5.1 (K2) Dar ejemplos de modos de fallo habituales de la prueba de rendimiento y sus causas.

1.1 Fundamentos de la Prueba de Rendimiento

La eficiencia de desempeño (o simplemente "rendimiento") es una parte esencial para proporcionar una "buena experiencia" a los usuarios cuando utilizan sus aplicaciones en distintas plataformas fijas y móviles. La prueba de rendimiento juega un papel crítico en el establecimiento de niveles de calidad aceptables para el usuario final y, a menudo, está estrechamente integrada con otras disciplinas como la ingeniería de usabilidad y la ingeniería de rendimiento.

Además, la evaluación de la adecuación funcional, usabilidad y otras características de calidad en condiciones de carga, tales como durante la ejecución de una prueba de rendimiento, puede revelar problemas específicos de carga que afectan a esas características.

La prueba de rendimiento no se limita a los dominios basados en la web, donde el usuario final es el centro de atención. También es relevante para diferentes dominios de aplicación con una variedad de arquitecturas de sistema, como el clásico cliente-servidor, distribuido y embebido.

Técnicamente, la eficiencia de desempeño se clasifica en el Modelo de Calidad de Producto ISO 25010 [ISO25000] como una característica de calidad no funcional con las tres subcaracterísticas que se describen a continuación. El enfoque y el orden de prioridad adecuados dependen de los riesgos y las necesidades de los distintos implicados. El análisis de los resultados de la prueba puede identificar otras áreas de riesgo que deben ser abordadas.

Comportamiento Temporal¹: Generalmente, la evaluación del comportamiento temporal es el objetivo más común de la prueba de rendimiento. Este aspecto de la prueba de rendimiento evalúa la capacidad² de un componente o sistema para responder a las entradas del usuario o del sistema en un tiempo y bajo unas condiciones específicas. Las mediciones del comportamiento temporal pueden variar desde el tiempo de "extremo a extremo" que tarda el sistema en responder a la entrada del usuario, hasta el número de ciclos de CPU requeridos por un componente de software para ejecutar una tarea particular.

Utilización de Recursos³: Si se determina que la disponibilidad de los recursos del sistema es un riesgo, la utilización de esos recursos (por ejemplo, la asignación de memoria RAM limitada) puede investigarse mediante la realización de pruebas de rendimiento específicas.

Capacidad: Si se identifican como riesgo problemas de comportamiento del sistema en los límites de la capacidad requerida del sistema (por ejemplo, el número de usuarios o el volumen de datos), se pueden realizar pruebas de rendimiento para evaluar la adecuación de la arquitectura del sistema.

La prueba de rendimiento, a menudo, adopta la forma de un experimento, que permite medir y analizar parámetros específicos del sistema. Estas pueden llevarse a cabo de manera iterativa en apoyo del análisis, diseño e implementación del sistema para permitir que se tomen decisiones arquitectónicas y para ayudar a dar forma a las expectativas de los implicados.

Los siguientes principios de la prueba de rendimiento son particularmente relevantes.

- Las pruebas deben estar alineadas con las expectativas definidas de los diferentes grupos de implicados, en particular los usuarios, los diseñadores del sistema y el personal de operaciones.
- Las pruebas deben ser reproducibles. Deberán obtenerse resultados estadísticamente idénticos (dentro de una tolerancia especificada) repitiendo las pruebas en un sistema que no haya sufrido cambios.

¹ Consultar **

² Consultar **

³ Consultar **

- Las pruebas deben aportar resultados que sean comprensibles y que puedan compararse fácilmente con las expectativas de los implicados.
- Se pueden realizar las pruebas, cuando los recursos lo permitan, tanto en sistemas completos como parciales o en entornos de prueba representativos del sistema de producción.
- Las pruebas deben ser asequibles, desde el punto de vista práctico, y ejecutables dentro del plazo establecido por el proyecto.

Los libros de [Molyneaux09] y de [Microsoft07] proporcionan una base sólida sobre los principios y aspectos prácticos de la prueba de rendimiento.

Las tres subcaracterísticas de calidad mencionadas anteriormente tendrán un impacto en la capacidad del sistema sujeto a prueba (SSP) para escalar.

1.3 Tipos de Prueba de Rendimiento

Se pueden definir diferentes tipos de prueba de rendimiento. Cada uno de ellos puede ser aplicable a un proyecto determinado, dependiendo de los objetivos de la prueba.

Prueba de Rendimiento

La prueba de rendimiento es un término genérico que incluye cualquier tipo de prueba centrada en el rendimiento (capacidad de respuesta) del sistema o componente bajo diferentes volúmenes de carga.

Prueba de Carga

La prueba de carga se centra en la capacidad de un sistema para tratar niveles crecientes de cargas realistas anticipadas como resultado de solicitudes de transacciones generadas por un número controlado de usuarios o procesos concurrentes.

Prueba de Estrés (o prueba de esfuerzo)

La prueba de estrés se centra en la capacidad de un sistema o componente para tratar cargas máximas que están en o más allá de los límites de sus cargas de trabajo previstas o especificadas. La prueba de estrés también se utiliza para evaluar la capacidad de un sistema para tratar la reducción de la disponibilidad de recursos, como la capacidad de procesamiento⁴ accesible, el ancho de banda disponible y la memoria.

Prueba de Escalabilidad

La prueba de escalabilidad se centra en la capacidad de un sistema para cumplir con los requisitos de eficiencia futuros, que pueden ser mayores que los que se requieren actualmente. El objetivo de estas pruebas es determinar la capacidad de crecimiento del sistema (por ejemplo, con más usuarios, mayor cantidad de datos almacenados) sin violar los requisitos de rendimiento actualmente especificados o sin fallar. Una vez que se conocen los límites de la escalabilidad, se pueden establecer y supervisar los valores umbral en la producción para advertir de los problemas que pueden estar a punto de surgir... Además, se puede ajustar el entorno de producción con el equipamiento hardware adecuado.

Prueba de Picos

La prueba de picos se centra en la capacidad de un sistema para responder correctamente a estallidos repentinos de cargas máximas y retornar, a continuación, a un estado estable.

Prueba de Resistencia

La prueba de resistencia se centra en la estabilidad del sistema durante un período de tiempo específico para el contexto operativo del sistema. Este tipo de pruebas verifica que no hay problemas de capacidad de recursos (por ejemplo, fugas de memoria, conexiones a bases de datos, grupos de hilos) que pueden eventualmente degradar el rendimiento y/o causar fallos en los puntos críticos⁵.

⁴ Consultar **

⁵ Consultar **

Prueba de Concurrencia

La prueba de concurrencia se centra en el impacto de situaciones en las que se realizan acciones específicas simultáneamente (por ejemplo, cuando un gran número de usuarios inicia sesión al mismo tiempo). Los problemas de concurrencia son notoriamente difíciles de encontrar y reproducir, particularmente cuando el problema ocurre en un entorno donde la prueba tiene poco o ningún control, como en el caso de la producción.

Prueba de Capacidad

La prueba de capacidad determina el número de usuarios y/o transacciones que un sistema dado soportará sin dejar de cumplir los objetivos de rendimiento establecidos. Estos objetivos también pueden expresarse en relación con los volúmenes de datos resultantes de las transacciones.

1.4 Tipos de Prueba en la Prueba de Rendimiento

Los principales tipos de prueba utilizados en la prueba de rendimiento son la prueba estática y la prueba dinámica.

1.4.1 Prueba Estática

Las actividades de prueba estática son, a menudo, más importantes para las pruebas de rendimiento que para las pruebas de adecuación funcional. Esto se debe a que se introducen muchos defectos críticos de rendimiento en la arquitectura y el diseño del sistema. Estos defectos pueden ser introducidos por malentendidos o por la falta de conocimiento de los diseñadores y arquitectos. Estos defectos también pueden introducirse porque los requisitos no reflejaban adecuadamente el tiempo de respuesta, el rendimiento o los objetivos de utilización de recursos, la carga y el uso previstos del sistema, o las restricciones.

Las actividades de prueba estática para determinar el rendimiento pueden incluir:

- Revisiones de los requisitos, con especial atención a los aspectos de rendimiento y los riesgos.
- Revisiones de esquemas de bases de datos, diagramas de entidad-relación, metadatos, procedimientos almacenados y consultas.
- Revisiones de la arquitectura del sistema y de la red.
- Revisiones de fragmentos críticos de código del sistema (por ejemplo, algoritmos complejos).

1.4.2 Prueba Dinámica

A medida que se construye el sistema, la prueba dinámica de rendimiento debe comenzar tan pronto como sea posible. Las oportunidades para la prueba dinámica de rendimiento incluyen:

- Durante la prueba unitaria, incluyendo el uso de información de perfilado⁶ para determinar cuellos de botella potenciales y el análisis dinámico para evaluar la utilización de los recursos.
- Durante la prueba de integración de componentes, a través de casos de uso clave y flujos de trabajo, especialmente cuando se integran diferentes prestaciones de casos de uso o cuando se integran con la estructura "troncal" de un flujo de trabajo.
- Durante la prueba de sistema de los comportamientos globales de extremo a extremo bajo diversas condiciones de carga.
- Durante la prueba de integración de sistemas, especialmente para flujos de datos y flujos de trabajo a través de interfaces clave entre sistemas. En la prueba de integración de sistemas no es infrecuente que el "usuario" sea otro sistema o máquina (por ejemplo, entradas de entradas de sensores y otros sistemas).
- Durante la prueba de aceptación, para generar confianza en el usuario, el cliente y el operador con respecto al rendimiento adecuado del sistema y para ajustar el sistema en condiciones reales (pero generalmente no para encontrar defectos de rendimiento en el sistema).

⁶ Consultar **

En niveles de prueba superiores, como la prueba de sistema y la prueba de integración de sistemas, el uso de entornos, datos y cargas realistas es fundamental para obtener resultados precisos (véase el Capítulo 4). En los ciclos de vida Ágiles y otros ciclos de vida iterativos e incrementales, los equipos deben incorporar la prueba de rendimiento estático y dinámico en iteraciones tempranas, en lugar de esperar a las iteraciones finales para abordar los riesgos de rendimiento.

Si forma parte del sistema hardware a medida o nuevo, se pueden realizar pruebas de rendimiento dinámicas tempranas utilizando simuladores. Sin embargo, es una buena práctica comenzar a realizar pruebas en el hardware real lo antes posible, ya que los simuladores a menudo no capturan de forma adecuada las restricciones de recursos y los comportamientos relacionados con el rendimiento.

1.5 El Concepto de Generación de Carga

Para llevar a cabo los diversos tipos de pruebas de rendimiento descritos en la Sección 1.2, se deben modelar, generar y suministrar cargas representativas del sistema al sistema sujeto a prueba. Las cargas son comparables a las entradas de datos utilizadas para los casos de pruebas funcionales, pero difieren en los siguientes aspectos principales:

- Una carga de prueba de rendimiento debe representar muchas entradas de usuario, no sólo una.
- Una carga de prueba de rendimiento puede requerir hardware y herramientas dedicadas para su generación.
- La generación de una carga de prueba de rendimiento depende de la ausencia de defectos funcionales en el sistema sujeto a prueba que puedan afectar a la ejecución de la prueba.

La generación eficiente y fiable de una carga específica es un factor de éxito clave en la ejecución de las pruebas de rendimiento. Existen diferentes opciones para la generación de cargas.

Generación de Carga a Través de la Interfaz de Usuario

Este puede ser un enfoque adecuado si sólo se va a representar un pequeño número de usuarios y si se dispone del número necesario de clientes de software desde los que introducir las entradas necesarias. Este enfoque también se puede utilizar junto con herramientas de ejecución de pruebas funcionales, pero puede llegar a ser poco práctico a medida que aumenta el número de usuarios a simular. La estabilidad de la interfaz de usuario (UI por sus siglas en inglés) también representa una dependencia crítica. Los cambios frecuentes pueden afectar la repetibilidad de las pruebas de rendimiento y puede afectar significativamente a los costes de mantenimiento. La prueba a través de la interfaz de usuario puede ser el enfoque más representativo para las pruebas de extremo a extremo.

Generación de Carga utilizando Multitudes

Este enfoque depende de la disponibilidad de un gran número de probadores que representen a usuarios reales. En la prueba multitudinaria, los probadores están organizados de tal manera que se puede generar la carga deseada. Este puede ser un método adecuado para probar aplicaciones que son accesibles desde cualquier parte del mundo (por ejemplo, basadas en la web), y puede implicar que los usuarios generen una carga desde una amplia gama de diferentes tipos de dispositivos y configuraciones. Aunque este enfoque puede permitir la utilización de un gran número de usuarios, la carga generada no será tan reproducible y precisa como otras opciones y es más compleja de organizar.

Generación de Carga a través de la Interfaz de Programación de Aplicaciones (API por sus siglas en inglés)

Este enfoque es similar al uso de la interfaz de usuario para la entrada de datos, pero utiliza la API de la aplicación en lugar de la interfaz de usuario para simular la interacción del usuario con el sistema sujeto a la prueba. Por lo tanto, el enfoque es menos sensible a los cambios (por ejemplo, retrasos) en la interfaz de usuario y permite que las transacciones se procesen de la misma manera que si fueran introducidas directamente por un usuario a través de la interfaz de usuario. Se pueden crear guiones dedicados que llaman repetidamente a rutinas específicas de la Interfaz de Programación de Aplicaciones (API) y permiten simular a más usuarios en comparación con el uso de las entradas de la interfaz de usuario.

Generación de Carga utilizando Protocolos de Comunicación Capturados

Este enfoque implica capturar la interacción del usuario con el sistema sujeto a prueba a nivel de protocolo de comunicaciones y, a continuación, reproducir estos guiones para simular un número potencialmente muy grande de usuarios de una manera repetible y fiable. Este enfoque basado en herramientas se describe en las secciones 4.2.6 y 4.2.7.



1.6 Modos de Fallo de Rendimiento Comunes y sus Causas

Si bien es cierto que hay muchos modos diferentes de fallo de rendimiento que se pueden encontrar durante la prueba dinámica, los siguientes son algunos ejemplos de fallos comunes (incluidos los fallos totales del sistema⁷), junto con las causas características:

Respuesta lenta en todos los niveles de carga

En algunos casos, la respuesta es inaceptable independientemente de la carga. Esto puede ser causado por problemas de rendimiento subyacentes, incluyendo, pero no limitado a, mal diseño o implementación de la base de datos, latencia de la red y otras cargas en **segundo plano**⁸. Estos problemas se pueden identificar durante la prueba funcional y de usabilidad, **no sólo durante la prueba de rendimiento**, por lo que los analistas de pruebas deben estar atentos a ellos e informarlos.

Respuesta lenta bajo niveles de carga moderada a elevada

En algunos casos, la respuesta se degrada de forma inaceptable con cargas de moderadas a elevadas, incluso cuando dichas cargas se encuentran totalmente dentro de los rangos normales, esperados y permitidos. Los defectos subyacentes incluyen la saturación de uno o más recursos y cargas en segundo plano variables.

Respuesta degradada con el transcurso del tiempo

En algunos casos, la respuesta se degrada gradual o drásticamente con el tiempo. Las causas subyacentes incluyen fugas de memoria, fragmentación de disco, aumento de la carga de la red con el tiempo, crecimiento del repositorio de archivos y crecimiento inesperado de la base de datos.

Manipulación de errores inadecuada o descuidada bajo cargas elevadas o excesivas

En algunos casos, el tiempo de respuesta es aceptable, pero el manejo de errores se degrada con niveles de carga elevados y por encima del límite. Los defectos subyacentes incluyen la insuficiencia de recursos, colas⁹ y pilas de tamaño insuficiente, y configuraciones de **tiempo de espera**¹⁰ demasiado rápidas.

Entre los ejemplos específicos de los tipos generales de fallos enumerados anteriormente se incluyen:

- Una aplicación basada en la web que proporciona información sobre los servicios de una empresa no responde a las solicitudes de los usuarios en un plazo de siete segundos (una regla general del sector). La eficiencia de desempeño del sistema no se puede lograr bajo condiciones de carga específicas.
- Un sistema presenta un fallo total o es incapaz de responder a las entradas de los usuarios cuando son objeto de un gran número de solicitudes de los usuarios (por ejemplo, la venta de entradas para un gran evento deportivo). La capacidad del sistema para tratar este número de usuarios es insuficiente.

⁷ Consultar **

⁸ Consultar **

⁹ Consultar **

¹⁰ En este contexto se entiende que el “tiempo de espera” es limitado.

- La respuesta del sistema se degrada significativamente cuando los usuarios envían solicitudes de grandes cantidades de datos (por ejemplo, se publica en un sitio web para ser descargado un informe de gran tamaño e importancia). La capacidad del sistema para tratar los volúmenes de datos generados es insuficiente.
- El procesamiento por lotes no puede completarse antes de que sea necesario el procesamiento en línea. El tiempo de ejecución de los procesos por lotes es insuficiente para el período de tiempo disponible.
- Un sistema en tiempo real se queda sin RAM cuando los procesos paralelos generan grandes demandas de memoria dinámica que no pueden ser liberadas a tiempo. La memoria RAM no está dimensionada adecuadamente o no se ha establecido un orden de prioridad adecuado para las solicitudes de memoria RAM.
- Un componente de un sistema en tiempo real, el componente A, que suministra entradas al componente B del sistema en tiempo real, no puede calcular las actualizaciones con la tasa requerida. El sistema general es incapaz de responder a tiempo y puede fallar. Los módulos de código en el componente A deben ser evaluados y modificados ("perfilado del rendimiento¹¹") para asegurar que se puedan alcanzar las tasas de actualización requeridas.

¹¹ Consultar **

2 Fundamentos de la Medición del Rendimiento

Duración: 55 minutos

Palabras Clave

medición ("measurement")

métrica ("métrica")

Objetivos de Aprendizaje para Fundamentos de la Medición del Rendimiento

2.1 Métricas Características Recopiladas en la Prueba de Rendimiento

NBPR-2.1.1 (K2) Comprender las métricas características recopiladas en la prueba de rendimiento.

2.2 Agregar los Resultados de la Prueba de Rendimiento

NBPR-2.1.2 (K2) Explicar los motivos por los que se agregan los resultados de las pruebas de rendimiento.

2.3 Fuentes Clave de las Métricas de Rendimiento

NBPR-2.3.1 (K2) Comprender las fuentes clave de las métricas de rendimiento.

2.4 Resultados Característicos de una Prueba de Rendimiento

NBPR-2.4.1 (K1) Recordar los resultados característicos de una prueba de rendimiento.

2.1 Métricas Características Recopiladas en la Prueba de Rendimiento

2.1.1 Por qué son Necesarias las Métricas de Rendimiento

Las mediciones precisas y las métricas que se derivan de esas mediciones son esenciales para definir los objetivos de la prueba de rendimiento y para evaluar los resultados de la prueba de rendimiento. La prueba de rendimiento no debe llevarse a cabo sin antes comprender qué mediciones y métricas son necesarias. Los siguientes riesgos del proyecto se aplican si se ignora este consejo:

- Se desconoce si los niveles de rendimiento son aceptables para cumplir los objetivos operativos.
- Los requisitos de rendimiento no están definidos en términos mensurables.
- Es posible que no sea posible identificar tendencias que puedan predecir niveles más bajos de rendimiento.
- Los resultados reales de una prueba de rendimiento no pueden evaluarse comparándolos con un conjunto de medidas de rendimiento de referencia que definan un rendimiento aceptable y/o inaceptable.
- Los resultados de la prueba de rendimiento se evalúan en base a la opinión subjetiva de una o más personas.
- No se entienden los resultados proporcionados por una herramienta de prueba de rendimiento.

2.1.2 Recopilación de Mediciones y Métricas de Rendimiento

Como con cualquier forma de medición, es posible obtener y expresar métricas de forma precisa. Por lo tanto, cualquiera de las métricas y mediciones descritas en esta sección pueden y deben definirse para que sean significativas en un contexto particular. Esto se refiere a realizar pruebas iniciales y aprender qué métricas necesitan ser refinadas y cuáles necesitan ser incorporadas.

Por ejemplo, la métrica del tiempo de respuesta probablemente estará en cualquier conjunto de métricas de rendimiento. Sin embargo, para que sea significativa y viable, la métrica del tiempo de respuesta deberá definirse con mayor precisión en términos de la hora del día, el número de usuarios concurrentes, la cantidad de datos que se están procesando, y así sucesivamente.

Las métricas recogidas en una prueba de rendimiento específica variarán en función de los siguientes factores:

- contexto de negocio (procesos de negocio, comportamiento de clientes y usuarios, y expectativas de los implicados).
- contexto operativo (tecnología y cómo se utiliza).
- objetivos de prueba.

Por ejemplo, las métricas elegidas para las pruebas de rendimiento de un sitio web de comercio electrónico internacional serán diferentes de las elegidas para las pruebas de rendimiento de un sistema embebido utilizado para controlar la funcionalidad de dispositivos médicos.

Una forma común de clasificar las mediciones y métricas de rendimiento es considerar el entorno técnico, el entorno de negocio o el entorno operativo en el que se necesita la evaluación del rendimiento.

Las categorías de mediciones y métricas que se incluyen a continuación son las que se obtienen comúnmente a partir de las pruebas de rendimiento.

Entorno Técnico

Las métricas de rendimiento variarán según el tipo de entorno técnico, como se muestra en la siguiente lista:

- Basado en la Web.
- Móvil.
- Internet de las Cosas¹² (IoT¹³ por sus siglas en inglés).
- Dispositivos cliente de escritorio.
- Procesamiento en el servidor.
- Mainframe.
- Bases de datos.
- Redes.
- La naturaleza del software que se ejecuta en el entorno (por ejemplo, integrado).

Las métricas incluyen lo siguiente:

- Tiempo de respuesta (por ejemplo, por transacción, por usuario concurrente, tiempos de carga de página).
- Utilización de recursos (por ejemplo, CPU, memoria, ancho de banda de red, latencia de red, espacio disponible en el disco, velocidad de E/S, hilos inactivos¹⁴ y ocupados¹⁵).
- Velocidad de procesamiento¹⁶ de una transacción clave (es decir, el número de transacciones que se pueden procesar en un período de tiempo determinado).
- Tiempo de procesamiento de lotes (por ejemplo, tiempos de espera, tiempos de procesamiento, tiempos de respuesta de la base de datos, tiempos de compleción).
- Número de errores que impactan al rendimiento.
- Tiempo de compleción (por ejemplo, para crear, leer, actualizar y borrar datos).
- Carga en segundo plano sobre recursos compartidos (especialmente en entornos virtualizados).
- Métricas de software (por ejemplo, complejidad del código).

¹² Consultar **

¹³ Consultar **

¹⁴ Consultar **

¹⁵ Consultar **

¹⁶ Consultar **

Entorno de Negocio

Desde la perspectiva de negocio o funcional, las métricas de rendimiento pueden incluir lo siguiente:

- Eficiencia de los procesos de negocio (por ejemplo, la velocidad con la que se realiza un proceso de negocio global que incluye flujos de casos de uso normal, alternativos y excepcionales).
- Procesamiento de datos, transacciones y otras unidades de trabajo realizadas (por ejemplo, pedidos procesados por hora, filas de datos añadidas por minuto).
- Índices de cumplimiento o incumplimiento del Acuerdo de Nivel de Servicio (SLA por sus siglas en inglés) (por ejemplo, incumplimiento del ANS por unidad de tiempo).
- Alcance del uso (por ejemplo, porcentaje de usuarios globales o nacionales que realizan tareas en un momento dado).
- Concurrencia de uso (por ejemplo, el número de usuarios que realizan una tarea de forma simultánea).
- Cronología de uso (por ejemplo, el número de pedidos procesados durante las horas de carga máxima).

Entorno Operativo

El aspecto operativo de la prueba de rendimiento se centra en las tareas que generalmente no se consideran de cara al usuario. Estos incluyen los siguientes:

- Procesos operativos (por ejemplo, el tiempo necesario para la puesta en marcha del entorno, copias de seguridad, tiempos de apagado y reanudación).
- Restauración de sistema (por ejemplo, el tiempo necesario para restaurar los datos de una copia de seguridad).
- Alertas y avisos (por ejemplo, el tiempo necesario para que el sistema emita una alerta o aviso).

2.1.3 Selección de las Métricas de Rendimiento

Cabe señalar que la recopilación de más métricas de las necesarias no es necesariamente algo bueno. Cada métrica elegida requiere un medio para la recopilación y el suministro de información consistente. Es importante definir un conjunto de métricas obtenibles que apoyen los objetivos de la prueba de rendimiento.

Por ejemplo, el enfoque Goal-Question-Metric (GQM) es una forma útil de alinear las métricas con las metas de rendimiento. La idea es primero establecer los objetivos, luego hacer preguntas para saber cuándo se han alcanzado los objetivos. Las métricas están asociadas con cada pregunta para asegurar que la respuesta a la pregunta sea medible. (Para una descripción más completa del enfoque GQM, véase la sección 4.3 del Programa de Estudios de Nivel Experto - Mejora del Proceso de Prueba [ISTQB_ELTM_ITP_SYL]). Cabe señalar que el enfoque GQM no siempre se ajusta al proceso de prueba de rendimiento. Por ejemplo, algunas métricas representan la salud de un sistema y no están directamente relacionadas con las metas.

Es importante tener en cuenta que después de la definición y captura de las mediciones iniciales, es posible que se necesiten más mediciones y métricas para comprender los verdaderos niveles de rendimiento y determinar dónde pueden ser necesarias las acciones correctivas.

2.2 Agregación de los Resultados de la Prueba de Rendimiento

El propósito de la agregación de las métricas de rendimiento es lograr comprenderlas y expresarlas de una manera que transmita con precisión la imagen completa del rendimiento del sistema. Cuando las métricas de rendimiento se consideran sólo en el nivel detallado, puede ser difícil llegar a la conclusión correcta, especialmente para los implicados del negocio.

Para muchos implicados, la principal preocupación es que el tiempo de respuesta de un sistema, sitio web u otro objeto de prueba esté dentro de unos límites aceptables.

Una vez que se ha logrado una comprensión más profunda de las métricas de rendimiento, las métricas se pueden agregar para que:

- Los implicados del negocio y del proyecto pueden ver el estado general del rendimiento del sistema.
- Se puedan identificar las tendencias de rendimiento.
- Las métricas de rendimiento pueden ser informadas de forma comprensible.

2.3 Fuentes Clave de las Métricas de Rendimiento

El rendimiento del sistema no debería verse más que mínimamente afectado por el esfuerzo de recopilación de métricas (conocido como el "efecto sonda"). Además, el volumen, la precisión y la velocidad con la que deben recopilarse las métricas de rendimiento hacen que el uso de una herramienta sea un requisito. Aunque el uso combinado de herramientas no es infrecuente, puede introducir redundancia en el uso de herramientas de prueba y otros problemas (ver Sección 4.4).

Hay tres fuentes clave de métricas de rendimiento:

Herramientas de Prueba de Rendimiento

Todas las herramientas de pruebas de rendimiento proporcionan mediciones y métricas como resultado de una prueba. Las herramientas pueden variar en el número de métricas mostradas, la forma en que se muestran las métricas y la capacidad del usuario para personalizar las métricas a una situación particular (ver también la Sección 5.1).

Algunas herramientas recopilan y muestran las métricas de rendimiento en formato de texto, mientras que las herramientas más robustas recopilan y muestran las métricas de rendimiento gráficamente en formato de cuadro de mando. Muchas herramientas ofrecen la posibilidad de exportar métricas para facilitar la evaluación de la prueba y el suministro de información.

Herramientas de Monitorización del Rendimiento

A menudo se utilizan herramientas de seguimiento del rendimiento para complementar las capacidades de suministro de información de las herramientas de prueba del rendimiento (véase también la sección 5.1). Además, se pueden utilizar herramientas de monitorización para monitorizar el rendimiento del sistema de forma continua y para alertar a los administradores del sistema de los niveles de rendimiento más bajos y de los niveles más altos de errores y alertas del sistema. Estas herramientas también se pueden utilizar para detectar y notificar en caso de comportamiento sospechoso (como ataques de denegación de servicio¹⁷ y ataques de denegación de servicio distribuidos¹⁸).

Herramientas de Análisis de Registros

Existen herramientas que analizan los registros del servidor y elaboran métricas a partir de ellos. Algunas de estas herramientas pueden crear gráficos para proporcionar una vista gráfica de los datos.

Los errores, alertas y avisos se registran normalmente en los registros del servidor. Estos incluyen:

- Elevado uso de recursos, como una alta utilización de la CPU, consumo de altos niveles de almacenamiento en disco y ancho de banda insuficiente.
- Errores de memoria y avisos, como el agotamiento de la memoria¹⁹.

¹⁷ Consultar **

¹⁸ Consultar **

¹⁹ Consultar **

- Problemas de bloqueo mutuo²⁰ y multihilos²¹, especialmente cuando se realizan operaciones en la base de datos.
- Errores de base de datos, como excepciones SQL y tiempos de espera SQL.

²⁰ Consultar **

²¹ Consultar **

2.4 Resultados Característicos de una Prueba de Rendimiento

En la prueba funcional, particularmente cuando se verifican requisitos funcionales específicos o elementos funcionales de historias de usuarios, los resultados esperados generalmente se pueden definir claramente y los resultados de la prueba se pueden interpretar para determinar si la prueba pasó o falló. Por ejemplo, un informe de ventas mensual muestra un total correcto o incorrecto.

Mientras que las pruebas que verifican la adecuación funcional a menudo se benefician de oráculos de prueba bien definidos, la prueba de rendimiento, a menudo, carece de esta fuente de información. No sólo los implicados son notoriamente poco hábiles a la hora de articular los requisitos de rendimiento, sino que muchos analistas de negocio y propietarios de productos son poco hábiles a la hora de obtener tales requisitos. Los probadores, a menudo, reciben una orientación limitada para definir los resultados esperados de la prueba.

Cuando se evalúan los resultados de la prueba de rendimiento, es importante observar los resultados de cerca. Los resultados iniciales en bruto pueden ser engañosos, ya que los fallos de rendimiento se ocultan bajo unos resultados generales aparentemente buenos. Por ejemplo, la utilización de los recursos puede estar muy por debajo del 75% en el caso de todos los recursos clave con potencial de cuellos de botella, pero la capacidad de procesamiento o el tiempo de respuesta de las transacciones clave o los casos de uso son demasiado lentos.

Los resultados específicos a evaluar varían dependiendo de las pruebas que se realizan, y a menudo incluyen los que se discuten en la Sección 2.1.

3 La Prueba de Rendimiento en el Ciclo de Vida del Software

Duración: 195 minutos

Palabras Clave

métrica (“metric”)

riesgo (“risk”)

ciclo de vida de desarrollo de software (“software development lifecycle”)

bitácora de la prueba o registro de prueba (“test log”)

Objetivos de Aprendizaje para La Prueba de Rendimiento en el Ciclo de Vida del Software

3.1 Principales Actividades de la Prueba de Rendimiento

NBPR-3.1.1 (K2) Entender las principales actividades de pruebas de rendimiento.

3.2 Categorías de Riesgos de Rendimiento para Diferentes Arquitecturas

NBPR-3.2.1 (K2) Explicar las categorías típicas de riesgos de rendimiento para diferentes arquitecturas.

3.3 Riesgos de Rendimiento a lo largo del Ciclo de Vida de Desarrollo de Software

NBPR-3.3.1 (K4) Analizar los riesgos de rendimiento para un producto dado a lo largo del ciclo de vida de desarrollo de software.

3.4 Actividades de la Prueba de Rendimiento

NBPR-3.4.1 (K4) Analizar un proyecto dado para determinar las actividades de prueba de rendimiento adecuadas para cada fase del ciclo de vida de desarrollo de software.

3.1 Principales Actividades de la Prueba de Rendimiento

La prueba de rendimiento es de naturaleza iterativa. Cada prueba proporciona información valiosa sobre el rendimiento de la aplicación y del sistema. La información obtenida de una prueba se utiliza para corregir u optimizar la aplicación y los parámetros del sistema. La siguiente iteración de la prueba mostrará los resultados de las modificaciones, y así sucesivamente hasta que se alcancen los objetivos de prueba.

Las actividades de pruebas de rendimiento están alineadas con el proceso de prueba del ISTQB [ISTQB_FL_SYL].

Planificación de la prueba

La planificación de la prueba es particularmente importante para la prueba de rendimiento debido a la necesidad de asignar entornos de prueba, datos de prueba, herramientas y recursos humanos. Además, esta es la actividad en la que se establece el alcance de la prueba de rendimiento.

Durante la planificación de la prueba, se completan las actividades de identificación y análisis de riesgos y se actualiza la información correspondiente en cualquier documentación de planificación de la prueba (por ejemplo, el plan de prueba, el plan de prueba de nivel). Así como se revisa y modifica la planificación de la prueba según sea necesario, también se modifican los riesgos, los niveles de riesgo y el estado de riesgo para reflejar los cambios en las condiciones de riesgo.

Monitorización y Control de la Prueba (también seguimiento y control de la prueba)

Se definen medidas de control para proporcionar planes de acción en caso de que surjan problemas que puedan afectar la eficiencia del desempeño, tales como:

- aumentar la capacidad de generación de carga si la infraestructura no genera las cargas deseadas según lo previsto para pruebas de rendimiento específicas.
- hardware modificado, nuevo o reemplazado.
- cambios en componentes de red.
- cambios en la implementación del software.

Los objetivos de la prueba de rendimiento se evalúan para comprobar el cumplimiento de los criterios de salida.

Análisis de la Prueba

Una prueba de rendimiento eficaz se basa en un análisis de los requisitos de rendimiento, los objetivos de la prueba, los acuerdos de nivel de servicio (SLA), la arquitectura de TI, los modelos de procesos y otros elementos que constituyen la base de la prueba. Esta actividad puede ser apoyada por el modelado y análisis de los requisitos de recursos del sistema y/o el comportamiento utilizando hojas de cálculo o herramientas de planificación de la capacidad.

Se identifican condiciones de prueba específicas, como los niveles de carga, las condiciones de tiempo y las transacciones que se van a probar. A continuación, se deciden los tipos de prueba de rendimiento necesarios (por ejemplo, carga, estrés, escalabilidad).

Diseño de la Prueba

Se diseñan casos de prueba de rendimiento. Por lo general, se crean de forma modular, de modo que pueden utilizarse como componentes básicos de pruebas de rendimiento más grandes y complejas (véase el apartado 4.2).

Implementación de la Prueba

En la fase de implementación, los casos de prueba de rendimiento se ordenan en procedimientos de prueba de rendimiento. Estos procedimientos de prueba de rendimiento deben reflejar los pasos que normalmente lleva a cabo el usuario y otras actividades funcionales que deben cubrirse durante la prueba de rendimiento.

Una actividad de implementación de la prueba es establecer y/o restablecer el entorno de prueba antes de cada ejecución de la prueba. Dado que la prueba de rendimiento suele estar basada en datos, se necesita un proceso para establecer datos de prueba que sean representativos de los datos de producción reales en volumen y tipo, de modo que se pueda simular el uso de la producción.

Ejecución de la prueba

La ejecución de la prueba se produce cuando se realiza la prueba de rendimiento, a menudo mediante el uso de herramientas de prueba de rendimiento. Los resultados de la prueba se evalúan para determinar si el rendimiento del sistema cumple con los requisitos y otros objetivos establecidos. Cualquier defecto es informado.

Compleción de la prueba

Los resultados de la prueba de rendimiento se proporcionan a los implicados (por ejemplo, arquitectos, gerentes, propietarios de producto) en un informe resumen de la prueba. Los resultados se expresan a través de métricas que, a menudo, se agregan para simplificar el significado de los resultados de la prueba. A menudo se utilizan medios visuales de presentación de informes, como los cuadros de mando, para expresar los resultados de la prueba de rendimiento de forma más fácil de entender que las métricas basadas en texto.

La prueba de rendimiento se considera, a menudo, una actividad que se realiza de forma continua, ya que se realiza en múltiples ocasiones y a todos los niveles de prueba (componente, integración, sistema, integración de sistemas y pruebas de aceptación). Al final de un período definido de prueba de rendimiento, se puede llegar a un punto de cierre de la prueba en el que las pruebas diseñadas, los activos de la herramienta de prueba (casos de prueba y procedimientos de prueba), los datos de prueba y otros programas de prueba se almacenan o se transmiten a otros probadores para un uso posterior durante las actividades de mantenimiento del sistema.

3.2 Categorías de Riesgos de Rendimiento para Diferentes Arquitecturas

Como se mencionó anteriormente, el rendimiento de la aplicación o del sistema varía considerablemente en función de la arquitectura, la aplicación y el entorno anfitrión²². Aunque no es posible proporcionar una lista completa de riesgos de rendimiento para todos los sistemas, la siguiente lista incluye algunos tipos de riesgos característicos asociados con arquitecturas particulares:

Sistemas informáticos individuales²³

Se trata de sistemas o aplicaciones que se ejecutan íntegramente en un equipo no virtualizado. El rendimiento puede degradarse debido a:

- Consumo excesivo de recursos, incluyendo fugas de memoria, actividades en segundo plano como software de seguridad, subsistemas de almacenamiento lento (por ejemplo, dispositivos externos de baja velocidad o fragmentación de disco) y mala administración del sistema operativo.
- Implementación ineficiente de algoritmos que no utilizan los recursos disponibles (por ejemplo, la memoria principal) y, como resultado, se ejecutan más lentamente de lo necesario.

Sistemas Multicapa

Se trata de sistemas de sistemas que se ejecutan en varios servidores, cada uno de los cuales realiza un conjunto específico de tareas, como servidor de base de datos, servidor de aplicaciones y servidor de presentación. Cada servidor es, por supuesto, un ordenador y está sujeto a los riesgos mencionados anteriormente. Además, el rendimiento puede degradarse debido a un diseño de base de datos deficiente o no escalable, cuellos de botella en la red y un ancho de banda o capacidad inadecuados en un único servidor.

Sistemas Distribuidos

Se trata de sistemas de sistemas, similares a una arquitectura multicapa, pero los distintos servidores pueden cambiar dinámicamente, como un sistema de comercio electrónico que accede a diferentes bases de datos de inventario en función de la ubicación geográfica de la persona que realiza el pedido. Además de los riesgos asociados con las arquitecturas de varios niveles, esta arquitectura puede experimentar problemas de rendimiento debido a flujos de trabajo críticos o flujos de datos hacia, desde o a través de servidores remotos poco fiables o impredecibles, en especial cuando dichos servidores sufren problemas de conexión periódicos o períodos de carga intensa e intermitente.

²² Consultar **

²³ Consultar **

Sistemas Virtualizados

Se trata de sistemas en los que el hardware físico aloja múltiples ordenadores virtuales. Estas máquinas virtuales pueden alojar sistemas y aplicaciones de un solo ordenador, así como servidores que forman parte de una arquitectura multicapa o distribuida. Los riesgos de rendimiento que surgen específicamente de la virtualización incluyen una carga excesiva en el hardware de todas las máquinas virtuales o una configuración incorrecta de la máquina virtual del anfitrión, lo que resulta en recursos inadecuados.

Sistemas Dinámicos / Basados en la Nube

Se trata de sistemas que ofrecen la posibilidad de escalar bajo demanda, aumentando la capacidad a medida que aumenta el nivel de carga. Estos sistemas suelen ser sistemas distribuidos y virtualizados de varios niveles, aunque con características de autoescalado²⁴ diseñadas específicamente para mitigar algunos de los riesgos de rendimiento asociados a esas arquitecturas. Sin embargo, existen riesgos asociados con los fallos para configurar correctamente estas funciones durante la configuración inicial o las actualizaciones posteriores.

Sistemas Cliente-Servidor

Se trata de sistemas que se ejecutan en un cliente y que se comunican a través de una interfaz de usuario con un único servidor, un servidor de varios niveles o un servidor distribuido. Dado que hay código ejecutándose en el cliente, los riesgos informáticos individuales se aplican a ese código, mientras que los problemas del lado del servidor mencionados anteriormente también se aplican. Además, existen riesgos de rendimiento debido a problemas de velocidad y fiabilidad de la conexión, congestión de la red en el punto de conexión del cliente (por ejemplo, Wi-Fi público) y problemas potenciales debidos a cortafuegos, inspección de paquetes y equilibrio de la carga del servidor.

Aplicaciones Móviles

Se trata de aplicaciones que se ejecutan en un teléfono inteligente²⁵, una tableta u otro dispositivo móvil. Estas aplicaciones están sujetas a los riesgos mencionados para las aplicaciones cliente-servidor y basadas en navegador (aplicaciones web). Además, pueden surgir problemas de rendimiento debido a los recursos limitados y variables y a la conectividad disponible en el dispositivo móvil (que puede verse afectada por la ubicación, la duración de la batería, el estado de carga, la memoria disponible en el dispositivo y la temperatura). Para aquellas aplicaciones que utilizan sensores de dispositivos o radios como acelerómetros o Bluetooth, los flujos de datos lentos de esas fuentes podrían crear problemas. Por último, las aplicaciones móviles suelen tener fuertes interacciones con otras aplicaciones móviles locales y servicios web remotos, cualquiera de los cuales puede convertirse en un cuello de botella de la eficiencia del rendimiento.

²⁴ Consultar **

²⁵ Consultar **

Sistemas en Tiempo Real Embebidos

Estos son sistemas que funcionan dentro o incluso controlan objetos cotidianos como los automóviles (por ejemplo, sistemas de entretenimiento y sistemas inteligentes de frenado), ascensores, semáforos, sistemas de calefacción, ventilación y aire acondicionado (HVAC), y más. Estos sistemas a menudo tienen muchos de los riesgos de los dispositivos móviles, incluyendo (cada vez más) problemas relacionados con la conectividad, ya que estos dispositivos están conectados a Internet. Sin embargo, la disminución del rendimiento de un videojuego móvil no suele ser un peligro para la seguridad del usuario, mientras que tales ralentizaciones en el sistema de frenado de un vehículo podrían resultar catastróficas.

Aplicaciones Mainframe

Se trata de aplicaciones -en muchos casos con décadas de antigüedad- que soportan, a menudo, funciones de negocio de misión crítica en un centro de datos, a veces a través del procesamiento por lotes. La mayoría son bastante predecibles y rápidos cuando se utilizan como se diseñaron originalmente, pero muchos de ellos son ahora accesibles a través de interfaces de programación de aplicaciones (API por sus siglas en inglés), servicios web o a través de su base de datos, lo que puede resultar en cargas inesperadas que afecten el rendimiento de las aplicaciones establecidas.

Se debe tener en cuenta que cualquier aplicación o sistema en particular puede incorporar dos o más de las arquitecturas mencionadas anteriormente, lo que significa que todos los riesgos relevantes se aplicarán a esa aplicación o sistema. De hecho, dada la Internet de las Cosas y la proliferación de aplicaciones móviles (dos áreas en las que los niveles extremos de interacción y conexión son la regla) es posible que todas las arquitecturas estén presentes de alguna forma en una aplicación y, por lo tanto, se puedan aplicar todos los riesgos.

Mientras que la arquitectura es claramente una decisión técnica importante con un profundo impacto en los riesgos de rendimiento, otras decisiones técnicas también influyen y generan riesgos. Por ejemplo, las fugas de memoria son más comunes en los lenguajes que permiten la gestión directa de la memoria en pilas, como C y C++, y los problemas de rendimiento son diferentes para las bases de datos relacionales y no relacionales. Tales decisiones se extienden hasta el diseño de funciones o métodos individuales (por ejemplo, la elección de un algoritmo recursivo en lugar de uno iterativo). Como probador, la capacidad de conocer o incluso influir en tales decisiones variará, dependiendo de las funciones y responsabilidades de los probadores dentro de la organización y del ciclo de vida de desarrollo de software.

3.3 Riesgos de Rendimiento a lo largo del Ciclo de Vida de Desarrollo de Software

El proceso de analizar los riesgos para la calidad de un producto software en general se trata en varios programas de estudio del ISTQB (por ejemplo, véase [ISTQB_FL_SYL] y [ISTQB_ALTM_SYL]). También se pueden encontrar explicaciones sobre riesgos específicos y consideraciones asociadas con características de calidad particulares (por ejemplo, [ISTQB_UT_SYL]), y desde una perspectiva de negocio o técnica (por ejemplo, véanse [ISTQB_ALTA_SYL] e [ISTQB_ALTA_SYL], respectivamente). En esta sección, el enfoque se centra en los riesgos relacionados con el rendimiento para la calidad del producto, incluyendo las formas en que cambian el proceso, los participantes y las consideraciones.

Para los riesgos relacionados con el rendimiento que afectan a la calidad de producto, el proceso es:

1. Identificar los riesgos de calidad de producto, centrándose en características como el comportamiento temporal, la utilización de recursos y la capacidad.
2. Evaluar los riesgos identificados, asegurándose de que se abordan las categorías de arquitectura pertinentes (véase la sección 3.2). Evaluar el nivel global de riesgo para cada riesgo identificado en términos de probabilidad e impacto utilizando criterios claramente definidos.
3. Adopte las medidas adecuadas para mitigar los riesgos de cada elemento de riesgo en función de la naturaleza del elemento de riesgo y del nivel de riesgo.
4. Gestionar los riesgos de forma continua para garantizar que los riesgos se mitiguen adecuadamente antes de su entrega.

Al igual que con el análisis de riesgos de calidad en general, los participantes en este proceso deben incluir tanto a los implicados técnicos como a los del negocio. Para el análisis de riesgos relacionados con el rendimiento, los implicados de negocio deben incluir a aquellos con un conocimiento particular de cómo los problemas de rendimiento en producción afectarán realmente a los clientes, a los usuarios, al negocio y a otros implicados en la cadena de suministro. Los implicados de negocio deben tener en cuenta que el uso previsto, la criticidad para el negocio, la sociedad o la seguridad, los posibles daños financieros y/o de reputación, la responsabilidad civil o penal y otros factores similares afectan al riesgo desde una perspectiva de negocio, creando riesgos e influyendo en el impacto de los fracasos.

Además, los implicados técnicos deben incluir a aquellos con un profundo conocimiento de las implicaciones de rendimiento de los requisitos relevantes, la arquitectura, el diseño y las decisiones de implementación. Los implicados técnicos deben tener en cuenta que las decisiones de arquitectura, diseño e implementación afectan a los riesgos de rendimiento desde una perspectiva técnica, creando riesgos e influyendo en la probabilidad de defectos.

El proceso específico de análisis de riesgos elegido debe tener el nivel adecuado de formalidad y rigor. Para los riesgos relacionados con el rendimiento, es especialmente importante que el proceso de análisis de riesgos se inicie pronto y se repita regularmente. En otras palabras, el probador debe evitar confiar enteramente en las pruebas de rendimiento realizadas hacia el final del nivel de prueba de sistema y el nivel de prueba de integración de sistemas. Muchos proyectos, especialmente los sistemas más grandes y complejos de sistemas de sistemas, han encontrado sorpresas desafortunadas debido al descubrimiento tardío de defectos de rendimiento que resultaron de los requisitos, el diseño, la arquitectura y las decisiones de implementación tomadas al principio del proyecto. Por lo tanto, se debe hacer hincapié en un enfoque iterativo para la identificación, evaluación, mitigación y gestión de los riesgos de rendimiento a lo largo del ciclo de vida del desarrollo de software.

Por ejemplo, si se manejan grandes volúmenes de datos a través de una base de datos relacional, el rendimiento lento de uniones de muchas a muchas debido a un diseño deficiente de la base de datos puede que sólo se revele durante las pruebas dinámicas con conjuntos de datos de prueba a gran escala, como los que se utilizan durante la prueba de sistema. Sin embargo, una revisión técnica cuidadosa que incluya ingenieros de bases de datos experimentados puede predecir los problemas antes de la implementación de la base de datos. Después de esta revisión, en un enfoque iterativo, se identifican y evalúan de nuevo los riesgos.

Además, la mitigación y gestión de riesgos debe abarcar e influir en todo el proceso de desarrollo de software, no sólo en las pruebas dinámicas. Por ejemplo, cuando las decisiones críticas relacionadas con el rendimiento, como el número esperado de transacciones o usuarios simultáneos, no pueden especificarse al principio del proyecto, es importante que las decisiones de diseño y arquitectura permitan una escalabilidad altamente variable (por ejemplo, recursos informáticos basados en la nube bajo demanda). Esto permite tomar decisiones tempranas de mitigación de riesgos.

Una buena ingeniería de rendimiento puede ayudar a los equipos de proyecto a evitar el descubrimiento tardío de defectos críticos de rendimiento durante los niveles de prueba superiores, como la prueba de integración de sistemas o la prueba de aceptación de usuario. Los defectos de rendimiento detectados en una fase tardía del proyecto pueden ser extremadamente costosos e incluso pueden dar lugar a la cancelación de proyectos completos.

Como con cualquier tipo de riesgo de calidad, los riesgos relacionados con el rendimiento nunca pueden evitarse completamente, es decir, siempre existirá algún riesgo de fallo en producción relacionado con el rendimiento. Por lo tanto, el proceso de gestión de riesgos debe incluir una evaluación realista y específica del nivel residual de riesgo para los implicados técnicos y de negocio en el proceso. Por ejemplo, decir simplemente "Sí, todavía es posible que los clientes experimenten grandes retrasos durante el proceso de pago" no es útil, ya que no da idea de qué cantidad de mitigación de riesgos se ha producido ni del nivel de riesgo que queda. En su lugar, proporcionar una visión clara del porcentaje de clientes que probablemente experimentarán retrasos iguales o superiores a ciertos umbrales ayudará a la gente a entender el estado.

3.4 Actividades de la Prueba de Rendimiento

Las actividades de prueba de rendimiento se organizarán y realizarán de forma diferente, dependiendo del tipo de ciclo de vida de desarrollo de software en uso.

Modelos de Desarrollo Secuencial

La práctica ideal de la prueba de rendimiento en modelos de desarrollo secuencial es incluir criterios de rendimiento como parte de los criterios de aceptación que se definen al principio de un proyecto. Reforzando la visión del ciclo de vida de la prueba, las actividades de prueba de rendimiento deben llevarse a cabo a lo largo del ciclo de vida del desarrollo de software. A medida que el proyecto avanza, cada actividad sucesiva de prueba de rendimiento debe basarse en los elementos definidos en las actividades anteriores, como se muestra a continuación.

- Concepto - Verificar que los objetivos de rendimiento del sistema se definan como criterios de aceptación del proyecto.
- Requisitos - Verificar que los requisitos de rendimiento estén definidos y representen correctamente las necesidades de los implicados.
- Análisis y Diseño - Verificar que el diseño del sistema refleje los requisitos de rendimiento.
- Codificación/Implementación - Verificar que el código sea eficiente y refleje los requisitos y el diseño en términos de rendimiento.
- Prueba de Componente - Realizar la prueba de rendimiento a nivel de componente.
- Prueba de Integración de Componentes - Realizar la prueba de rendimiento a nivel de integración de componentes.
- Prueba de Sistema - Realizar la prueba de rendimiento a nivel de sistema, que incluye hardware, software, procedimientos y datos que son representativos del entorno de producción. Las interfaces de sistema podrán simularse siempre que proporcionen una representación fiel del rendimiento.
- Prueba de Integración de Sistema - Realizar la prueba de rendimiento con todo el sistema que sea representativo del entorno de producción.
- Prueba de Aceptación - Validar que el rendimiento del sistema cumple con las necesidades del usuario originalmente establecidas y los criterios de aceptación.

Modelos de Desarrollo Iterativos e Incrementales

En estos modelos de desarrollo, como Agile, la prueba de rendimiento también se considera una actividad iterativa e incremental (véase [ISTQB_FL_ATJ]). La prueba de rendimiento puede ocurrir como parte de la primera iteración, o como una iteración dedicada enteramente a la prueba de rendimiento. Sin embargo, con estos modelos de ciclo de vida, la ejecución de la prueba de rendimiento puede ser realizada por un equipo específico encargado de la prueba de rendimiento.

La Integración Continua (CI por sus siglas en inglés) se realiza comúnmente en ciclos de vida de desarrollo de software iterativos e incrementales, lo que facilita una ejecución altamente automatizada de las pruebas. El objetivo más común de las pruebas en Integración Continua es realizar pruebas de regresión y asegurar que cada construcción sea estable. Las pruebas de rendimiento pueden formar parte de las pruebas automatizadas realizadas en Integración Continua si las pruebas se diseñan de manera que se ejecuten a

nivel de construcción. Sin embargo, a diferencia de las pruebas funcionales automatizadas, existen consideraciones adicionales como las siguientes:

- La configuración del entorno para la prueba de rendimiento - Esto a menudo requiere un entorno de prueba que está disponible bajo demanda, como un entorno de prueba de rendimiento basado en la nube.
- Determinar qué pruebas de rendimiento automatizar en Integración Continua - Debido al poco tiempo disponible para realizar pruebas de Integración Continua, las pruebas de rendimiento de CI pueden ser un subconjunto de pruebas de rendimiento más extensas que son realizadas por un equipo de especialistas en otros momentos durante una iteración.
- Crear pruebas de rendimiento para Integración Continua - El objetivo principal de las pruebas de rendimiento como parte de la Integración Continua es asegurar que un cambio no afecte negativamente al rendimiento. Dependiendo de los cambios realizados para cualquier construcción, es posible que se requieran nuevas pruebas de rendimiento.
- Ejecutar pruebas de rendimiento en secciones de una aplicación o sistema - Esto a menudo requiere que las herramientas y los entornos de prueba sean capaces de realizar pruebas de rendimiento rápidas, incluyendo la capacidad de seleccionar subconjuntos de pruebas aplicables.

La prueba de rendimiento en los ciclos de vida de desarrollo de software iterativo e incremental también puede tener sus propias actividades propias del ciclo de vida:

- Planificación de la Entrega - En esta actividad, la prueba de rendimiento se considera desde la perspectiva de todas las iteraciones en una entrega, desde la primera iteración hasta la última iteración. Se identifican y evalúan los riesgos de rendimiento y se planifican medidas de mitigación. A menudo, esto incluye la planificación de cualquier prueba de rendimiento final antes de la entrega de la aplicación.
- Planificación de la Iteración - En el contexto de cada iteración, se pueden realizar pruebas de rendimiento dentro de la iteración y a medida que se completa cada iteración. Los riesgos de rendimiento se evalúan con más detalle para cada historia de usuario.
- Creación de Historias de Usuario - A menudo las historias de usuario forman la base de los requisitos de rendimiento en metodologías ágiles, con los criterios de rendimiento específicos descritos en los criterios de aceptación asociados. Se denominan historias de usuarios "no funcionales".
- Diseñar la prueba de rendimiento: se utilizan requisitos y criterios de rendimiento que se describen en historias de usuario específicas para el diseño de las pruebas (véase la sección 4.2).
- Codificación/Implementación - Durante la codificación, la prueba de rendimiento puede realizarse a nivel de componente. Un ejemplo de esto sería la optimización de los algoritmos para una eficiencia de rendimiento óptima.
- Prueba/Evaluación - Mientras que la prueba se realiza, por lo general, muy próxima a las actividades de desarrollo, la prueba de rendimiento se puede realizar como una actividad diferenciada, dependiendo del alcance y de los objetivos de la prueba de rendimiento durante la iteración. Por ejemplo, si el objetivo de la prueba de rendimiento es probar el rendimiento de la iteración como un conjunto completo de historias de usuario, se necesitará un alcance de prueba de rendimiento más amplio que el que se observa en la prueba de rendimiento de una historia de usuario única. Esto puede ser programado en una iteración dedicada para la prueba de rendimiento.

- Entrega - Puesto que la entrega introducirá la aplicación en el entorno de producción, será necesario supervisar el rendimiento para determinar si la aplicación alcanza los niveles deseados de rendimiento en el uso real.

Productos Comerciales de Distribución Masiva (COTS, por sus siglas en inglés) y otros Modelos Comerciales Proveedor/Comprador

Muchas organizaciones no desarrollan aplicaciones y sistemas por sí mismas, sino que se encuentran en posición de adquirir software de fuentes de suministro²⁶ o de proyectos de código abierto. En estos modelos de proveedor/comprador, el rendimiento es una consideración importante que requiere pruebas tanto desde el punto de vista del proveedor (vendedor/desarrollador) como del comprador (cliente).

Independientemente del origen de la aplicación, a menudo es responsabilidad del cliente validar que el rendimiento cumpla sus requisitos. En el caso del software desarrollado a medida por el proveedor, los requisitos de rendimiento y los criterios de aceptación asociados que deben especificarse como parte del contrato entre el proveedor y el cliente. En el caso de las aplicaciones Productos Comerciales de Distribución Masiva ("COTS"), el cliente es el único responsable de probar el rendimiento del producto en un entorno de prueba realista antes de la implementación.

²⁶ Consultar **

4 Tareas de la Prueba de Rendimiento

Duración: 475 minutos

Palabras Clave

conurrencia (“concurrency”)
perfil de carga (“load profile”)
generación de carga (“load generation”)
perfil operativo (“operational profile”)
rampa descendente (“ramp-down”)
rampa ascendente (“ramp-up”)
sistema de sistemas (“system of systems”)
capacidad de procesamiento del sistema (“system throughput”)
plan de prueba (“test plan”)
tiempo de reflexión (“think time”)
usuario virtual (“virtual user”)

Objetivos de Aprendizaje para Tareas de la Prueba de Rendimiento

4.1 Planificación

- NBPR-4.1.1 (K4) Deducir los objetivos de la prueba de rendimiento a partir de una información relevante.
- NBPR-4.1.2 (K4) Esbozar un plan de prueba de rendimiento que tenga en cuenta los objetivos de rendimiento para un proyecto determinado.
- NBPR-4.1.3 (K4) Crear una presentación que permita a los distintos implicados entender la justificación de la prueba de rendimiento planificada.

4.2 Análisis, Diseño e Implementación

- NBPR-4.2.1 (K2) Dar ejemplos de protocolos característicos utilizados en la prueba de rendimiento.
- NBPR-4.2.2 (K2) Comprender el concepto de transacciones en la prueba de rendimiento.
- NBPR-4.2.3 (K4) Analizar perfiles operativos para el uso del sistema
- NBPR-4.2.4 (K4) Crear perfiles de carga derivados de perfiles operativos para determinados objetivos de rendimiento.
- NBPR-4.2.5 (K4) Analizar la capacidad de procesamiento y la concurrencia al desarrollar pruebas de rendimiento.
- NBPR-4.2.6 (K2) Comprender la estructura básica de un guion de prueba de rendimiento.
- NBPR-4.2.7 (K3) Implementar guiones de pruebas de rendimiento que sean coherentes con el plan y los perfiles de carga.

NBPR-4.2.8 (K2) Entender las actividades involucradas en la preparación para la ejecución de la prueba de rendimiento.

4.3 Ejecución

NBPR-4.3.1 (K2) Comprender las actividades principales en la ejecución de guiones de prueba de rendimiento

4.3 Analizar Resultados e Informar

NBPR-4.4.1 (K4) Analizar e informar sobre los resultados y las implicaciones de la prueba de rendimiento.

4.1 Planificación

4.1.1 Deducir Objetivos de la Prueba de Rendimiento

Los implicados pueden ser usuarios y personas con formación empresarial o técnica. Pueden tener objetivos diferentes en relación con la prueba de rendimiento. Los implicados establecen los objetivos, la terminología que debe utilizarse y los criterios para determinar si se ha alcanzado el objetivo.

Los objetivos de la prueba de rendimiento se relacionan con estos diferentes tipos de partes interesadas. Es una buena práctica distinguir entre objetivos técnicos y objetivos basados en el usuario. Los objetivos basados en el usuario se centran principalmente en la satisfacción del usuario final y en los objetivos empresariales. En general, los usuarios están menos preocupados por los tipos de funciones o por la forma en que se entrega un producto. Sólo quieren ser capaces de hacer lo que tienen que hacer.

Por otra parte, los objetivos técnicos se centran en los aspectos operativos y en proporcionar respuestas a las preguntas relativas a la capacidad de escalado de un sistema, o a las condiciones en las que puede manifestarse un rendimiento degradado.

Los objetivos clave de la prueba de rendimiento incluyen la identificación de riesgos potenciales, la búsqueda de oportunidades de mejora y la identificación de los cambios necesarios.

Al recopilar información de los distintos implicados, se deben responder las siguientes preguntas:

- ¿Qué transacciones se ejecutarán en la prueba de rendimiento y qué tiempo medio de respuesta se espera?
- ¿Qué métricas de sistema se van a capturar (por ejemplo, uso de memoria, tasa de transferencia de la red²⁷) y qué valores se esperan?
- ¿Qué mejoras en el rendimiento se esperan de estas pruebas en comparación con los ciclos de pruebas anteriores?

4.1.2 El Plan de Prueba de Rendimiento

El Plan de Prueba de Rendimiento (PTP) es un documento creado antes de que ocurra cualquier prueba de rendimiento. Se debe hacer referencia al PTP en el plan de prueba (véase [ISTQB_FL_SYL]), que también incluye información relativa al calendario. El PPR se actualiza de forma continua una vez que se inicia la prueba de rendimiento.

El PPR debe proporcionar la siguiente información:

²⁷ Consultar **

Objetivo

El objetivo del PPR describe los objetivos, estrategias y métodos para la prueba de rendimiento. Permite una respuesta cuantificable a la pregunta central de la adecuación y la disponibilidad del sistema para funcionar bajo carga.

Objetivos de la Prueba

Se enumeran los objetivos generales de la prueba de desempeño que debe alcanzar el Sistema Sujeto a Prueba (SSP o SUT por sus siglas en inglés) para cada tipo de implicado (véase la Sección 4.1.1).

Descripción General del Sistema

Una breve descripción del SSP proporcionará el contexto para la medición de los parámetros de la prueba de rendimiento. La descripción general debe incluir una descripción de alto nivel de la funcionalidad que se está probando bajo carga.

Tipos de Pruebas de Rendimiento a Realizar

Se enumeran los tipos de pruebas de rendimiento que se deben realizar (véase el apartado 1.2) junto con una descripción del objetivo para cada tipo.

Criterios de Aceptación

La prueba de rendimiento tiene por objeto determinar la capacidad de respuesta, la capacidad de procesamiento, la fiabilidad y/o la escalabilidad del sistema bajo una carga de trabajo determinada. En general, el tiempo de respuesta es una preocupación del usuario, la capacidad de procesamiento es una preocupación de negocio, y la utilización de recursos es una preocupación de sistema. Deberán establecerse criterios de aceptación para todas las medidas pertinentes y volver a relacionarse con lo siguiente, según proceda:

- Objetivos generales de la prueba de rendimiento.
- Acuerdos de Nivel de Servicio (SLA's por sus siglas).
- Valores línea base - Una línea base es un conjunto de métricas utilizadas para comparar las mediciones del desempeño actual y las realizadas anteriormente. Esto permite demostrar mejoras particulares en el rendimiento y/o confirmar el cumplimiento de los criterios de aceptación de la prueba. Puede ser necesario crear primero la línea base utilizando datos saneados de una base de datos, cuando sea posible.

Datos de Prueba

Para una prueba de rendimiento, los datos de prueba incluyen una amplia gama de datos que se deben especificar. Estos datos pueden incluir lo siguiente:

- Datos de cuentas de usuario (por ejemplo, cuentas de usuario disponibles para el inicio de sesión simultánea).
- Datos de entrada de usuario (por ejemplo, los datos que un usuario introduciría en la aplicación para realizar un proceso de negocio).
- Base de datos (por ejemplo, la base de datos poblada en forma previa que se puebla con datos para su uso en la prueba).

El proceso de creación de datos de prueba debe abordar los siguientes aspectos:

- extracción de datos a partir de datos de producción.
- importación de datos en el SSP.
- creación de nuevos datos.
- creación de copias de seguridad que se pueden utilizar para restaurar los datos cuando se realicen nuevos ciclos de pruebas.
- enmascarar o anonimizar datos. Esta práctica se utiliza en los datos de producción que contienen información personal identificable, y es obligatoria según las Normativas Generales de Protección de Datos (GDPR por sus siglas en inglés). Sin embargo, en la prueba de rendimiento, el enmascaramiento de datos añade riesgo a la prueba de rendimiento, ya que estos datos pueden no tener las mismas características que se observan en el mundo real.

Configuración del Sistema

La sección de configuración del sistema del PPR incluye la siguiente información técnica:

- Una descripción de la arquitectura específica del sistema, incluidos los servidores (por ejemplo, web, base de datos, balanceador de carga²⁸).
- Definición de las distintas capas.
- Detalles específicos del hardware del ordenador (por ejemplo, núcleos de CPU, RAM, discos de estado sólido (SSD), discos duros (HDD)), incluidas las versiones.
- Detalles específicos del software (por ejemplo, aplicaciones, sistemas operativos, bases de datos, servicios utilizados para soporte del negocio) incluyendo versiones.
- Sistemas externos que operan con el SSP y su configuración y versión (por ejemplo, sistema de comercio electrónico con integración a NetSuite).
- Construcción del SSP / identificador de versión.

²⁸ Consultar **

Entorno de Prueba

El entorno de prueba suele ser un entorno independiente que imita al de producción, pero a menor escala. Esta sección del PPR debe incluir cómo se extrapolarán los resultados de la prueba de rendimiento para aplicarlos al entorno de producción más amplio. En algunos sistemas, el entorno de producción se convierte en la única opción viable para la prueba, pero en este caso deben discutirse los riesgos específicos de este tipo de prueba.

Las herramientas de prueba a veces residen fuera del propio entorno de prueba y pueden requerir derechos de acceso especiales para interactuar con los componentes del sistema. Esto es una consideración para el entorno y la configuración de la prueba.

La prueba de rendimiento también puede realizarse con un componente del sistema que sea capaz de funcionar sin otros componentes. A menudo es más barato que realizar pruebas con todo el sistema y puede llevarse a cabo tan pronto como se desarrolle el componente.

Herramientas de Prueba

Esta sección incluye una descripción de las herramientas (y versiones) de prueba que se utilizarán en la creación de guiones, la ejecución y la monitorización de la prueba de rendimiento (véase el Capítulo 5). Esta lista normalmente incluye:

- Herramienta(s) utilizada(s) para simular transacciones de usuario.
- Herramientas para proporcionar carga desde múltiples puntos dentro de la arquitectura del sistema (puntos de presencia²⁹).
- Herramientas para monitorizar el rendimiento del sistema, incluidas las descritas anteriormente en configuración del sistema.

Perfiles

Los perfiles operativos proporcionan un flujo repetible paso a paso a través de la aplicación para un uso particular del sistema. La agregación de estos perfiles operativos da como resultado un perfil de carga (comúnmente denominado escenario). Consulte la Sección 4.2.3 para obtener más información sobre los perfiles.

Métricas de Relevancia

Durante la ejecución de una prueba de rendimiento se puede recoger un gran número de mediciones y métricas (véase el capítulo 2). Sin embargo, tomar demasiadas mediciones puede dificultar el análisis y afectar negativamente el rendimiento real de la aplicación. Por estas razones, es importante identificar las mediciones y métricas que son más relevantes para lograr los objetivos de la prueba de rendimiento.

La siguiente tabla, explicada con más detalle en la Sección 4.4, muestra un conjunto representativo de métricas para la prueba y monitorización del rendimiento. Los objetivos de la prueba de rendimiento deben definirse para estas métricas, cuando sea necesario, para el proyecto:

²⁹ Consultar **

Métricas de Rendimiento	
Tipo	Métrica
Estado de Usuario Virtual	# Pasado # Fallado
Tiempo de Respuesta de la Transacción	Mínimo Máximo Media Percentil 90%
Transacciones por Segundo	# Pasada / segundo # Fallada / segundo # Total / segundo
Impactos (por ejemplo en base de datos o servidor web)	Impactos / segundo Mínimo Máximo Media Total
Capacidad de Procesamiento	Bits / segundo Mínimo Máximo Media Total
Respuestas HTTP por segundo	Respuestas / segundo Mínimo Máximo Media Total Respuesta por códigos de respuesta HTTP

Monitorización del Rendimiento	
Tipo	Métrica
Uso de CPU	% de CPU disponible utilizada
Uso de memoria	% de memoria disponible utilizada

Riesgos

Los riesgos pueden incluir áreas que no se miden como parte de las pruebas de rendimiento, así como limitaciones a las pruebas de rendimiento (por ejemplo, interfaces externas que no se pueden simular, carga insuficiente, incapacidad para monitorizar servidores). Las limitaciones del entorno de prueba también pueden producir riesgos (por ejemplo, datos insuficientes, entorno reducido). Ver las secciones 3.2 y 3.3 para más tipos de riesgo.

4.1.3 Comunicar sobre la Prueba de Rendimiento

El probador debe ser capaz de comunicar a todos los implicados la justificación del enfoque de la prueba de rendimiento y las actividades que se llevarán a cabo (como se detalla en el plan de prueba de rendimiento). Los temas que se abordarán en esta comunicación pueden variar considerablemente entre los implicados, dependiendo de si tienen un interés "negocio / de cara al usuario" o un enfoque más "tecnología / de cara a operaciones".

Implicados Centrados en Negocio

Se deben tener en cuenta los siguientes factores al comunicarse con los implicados centrados en negocio:

- Los implicados centrados en negocio están menos interesados en las distinciones entre características de calidad funcionales y no funcionales.
- Las cuestiones técnicas relacionadas con las herramientas, los guiones y la generación de cargas son generalmente de interés secundario.
- La conexión entre los riesgos de producto y los objetivos de la prueba de rendimiento debe estar claramente enunciada.
- Los implicados deben ser conscientes del equilibrio entre el coste de la prueba de rendimiento prevista y la representatividad de los resultados de la prueba de rendimiento en comparación con las condiciones de producción.
- Se debe comunicar la repetibilidad de la prueba de rendimiento planificada. ¿Será difícil repetir la prueba o se puede repetir con un esfuerzo mínimo?
- Los riesgos de proyecto deben ser comunicados. Estos incluyen limitaciones y dependencias relacionadas con la configuración de la prueba, requisitos de infraestructura (por ejemplo, hardware, herramientas, datos, ancho de banda, entorno de prueba, recursos) y dependencias con respecto al personal clave.
- Se deben comunicar las actividades de alto nivel (ver Secciones 4.2 y 4.3) junto con un plan general que contenga los costes, el calendario y los hitos.

Implicados Centrados en la Tecnología

Se deben tener en cuenta los siguientes factores al comunicarse con los implicados centrados en la tecnología:

- El enfoque planificado para generar los perfiles de carga requeridos debe ser explicado y se debe aclarar la participación esperada de los implicados técnicos.
- Se deben explicar los pasos detallados en la configuración y ejecución de la prueba de rendimiento para mostrar la relación de la prueba con los riesgos arquitectónicos.
- Se deben comunicar los pasos necesarios para que las pruebas de rendimiento sean repetibles. Estos pueden incluir aspectos de la organización (por ejemplo, la participación del personal clave), así como cuestiones técnicas.
- Cuando los entornos de prueba vayan a ser compartidos, se debe comunicar el calendario de prueba de rendimiento para asegurar que los resultados de la prueba no se verán afectados negativamente.

- Se debe comunicar el calendario de la prueba en caso de que los entornos de prueba sean compartidos, para asegurar que los resultados de la prueba no se vean afectados de forma negativa.
- Se deben ejecutar acciones para mitigar el impacto potencial en los usuarios reales si es necesario realizar pruebas de rendimiento en el entorno de producción que deben ser comunicadas y aceptadas.
- Los implicados técnicos deben tener claro cuáles son sus tareas y cuándo están programadas (en términos de calendario).

4.2 Análisis, Diseño e Implementación

4.2.1 Protocolos de Comunicación Habituales

Los protocolos de comunicación definen un conjunto de reglas de comunicación entre ordenadores y sistemas. El diseño adecuado de las pruebas para dirigirse a partes específicas del sistema requiere la comprensión de los protocolos.

Los protocolos de comunicación se describen a menudo en las capas del modelo de interconexión de sistemas abiertos (Open Systems Interconnection, OSI) (véase ISO/IEC 7498-1), aunque algunos protocolos pueden quedar fuera de este modelo. Para la prueba de rendimiento, los protocolos de Capa 5 (Capa de Sesión) a Capa 7 (Capa de Aplicación) son los que se utilizan con mayor frecuencia para la prueba de rendimiento. Los protocolos comunes incluyen:

- Base de datos - ODBC, JDBC, otros protocolos específicos del proveedor.
- Web - HTTP, HTTPS, HTML.
- Servicio Web - SOAP, REST.

En términos generales, el nivel de la capa de OSI más importante en la prueba de rendimiento se relaciona con el nivel de la arquitectura que se está probando. Cuando se prueba una arquitectura embebida de bajo nivel, por ejemplo, las capas numeradas inferiores del modelo OSI serán las más importantes.

Los protocolos adicionales utilizados en la prueba de rendimiento incluyen:

- Red - DNS, FTP, IMAP, LDAP, POP3, SMTP, Windows Sockets, CORBA.
- Móvil - TruClient, SMP, MMS.
- Acceso remoto - Citrix ICA, RTE.
- SOA - MQSeries, JSON, WSCL.

Es importante entender la arquitectura general del sistema porque las pruebas de rendimiento se pueden ejecutar en un componente individual del sistema (por ejemplo, un servidor web, un servidor de bases de datos) o en un sistema completo mediante pruebas de extremo a extremo. Las aplicaciones tradicionales de 2 niveles construidas con un modelo cliente-servidor especifican el "cliente" como la interfaz gráfica de usuario y la interfaz de usuario principal, y el "servidor" como la base de datos backend. Estas aplicaciones requieren el uso de un protocolo como ODBC para acceder a la base de datos. Con la evolución de las aplicaciones basadas en la web y las arquitecturas de varios niveles, muchos servidores participan en el procesamiento de la información que, en última instancia, se transmite al navegador del usuario.

Dependiendo de la parte del sistema que se vaya a probar, es necesario conocer el protocolo apropiado que se va a utilizar. Por lo tanto, si la necesidad es realizar pruebas de extremo a extremo emulando la actividad del usuario desde el navegador, se empleará un protocolo web como HTTP/HTTPS. De esta manera, la interacción con la Interfaz Gráfica de Usuario puede ser evitada y la prueba puede enfocarse en la comunicación y actividades de los servidores backend.

4.2.2 Transacciones

Las transacciones describen el conjunto de actividades realizadas por un sistema desde el punto de inicio hasta el momento en que se han completado uno o más procesos (solicitudes, operaciones, o procesos operativos). Se puede medir el tiempo de respuesta de las transacciones con el fin de evaluar el rendimiento del sistema. Estas mediciones se utilizan durante una prueba de rendimiento para identificar cualquier componente que requiera ser corregido u optimizado.

Las transacciones simuladas pueden incluir tiempo de reflexión para reflejar mejor el momento en que un usuario real toma una acción (por ejemplo, presionando el botón "SEND"). El tiempo de respuesta de la transacción más el tiempo de reflexión es igual al tiempo transcurrido para esa transacción.

Los tiempos de respuesta de las transacciones recogidos durante la prueba de rendimiento muestran cómo cambia esta medición bajo diferentes cargas impuestas al sistema. El análisis puede no mostrar degradación bajo carga, mientras que otras mediciones pueden mostrar una degradación severa. Aumentando la carga y midiendo los tiempos de transacción subyacentes, es posible correlacionar la causa de la degradación con los tiempos de respuesta de una o más transacciones.

Las transacciones también pueden anidarse para que se puedan medir las actividades individuales y agregadas. Esto puede ser útil, por ejemplo, para comprender la eficacia de un sistema de pedidos en línea. El probador puede querer medir los pasos discretos en el proceso de pedido (por ejemplo, buscar el artículo, añadirlo a la cesta, pagar el artículo, confirmar el pedido), así como el proceso de pedido en su conjunto. Al anidar las transacciones, ambos conjuntos de información pueden ser reunidos en una sola prueba.

4.2.3 Identificación de Perfiles Operativos

Los perfiles operativos especifican distintos patrones de interacción con una aplicación, como los de los usuarios u otros componentes del sistema. Se pueden especificar múltiples perfiles operativos para una aplicación determinada. Pueden combinarse para crear un perfil de carga deseado con el fin de alcanzar determinados objetivos de la prueba de rendimiento (véase la sección 4.2.4).

En esta sección se describen los siguientes pasos principales para identificar los perfiles operativos:

1. Identificar los datos que se deben recopilar.
2. Recopilar los datos utilizando una o más fuentes.
3. Evaluar los datos para construir los perfiles operativos.

Identificar los Datos

Cuando los usuarios interactúan con el sistema sujeto a prueba, se recopilan o estiman los siguientes datos con el fin de modelar sus perfiles operativos (es decir, cómo interactúan con el sistema):

- Diferentes tipos de personas de usuario y sus roles (por ejemplo, usuario estándar, miembro registrado, administrador, grupos de usuarios con privilegios específicos).
- Diferentes tareas genéricas realizadas por esos usuarios/roles (por ejemplo, navegar por un sitio web en busca de información, buscar un producto específico en un sitio web, realizar actividades específicas de un rol). Se debe tener en cuenta que, generalmente, estas tareas se modelan mejor con un alto nivel de abstracción (por ejemplo, a nivel de procesos de negocio o de historias de usuario principales).

- Número estimado de usuarios para cada función/tarea por unidad de tiempo durante un período de tiempo determinado. Esta información también será útil para construir posteriormente perfiles de carga (véase la sección 4.2.4).

Recopilar Datos

Los datos mencionados anteriormente pueden obtenerse a partir de una serie de fuentes diferentes:

- Realizar entrevistas o talleres con los implicados, tales como propietarios de producto, gerentes de ventas y (potenciales) usuarios finales. Estas discusiones, a menudo, revelan los principales perfiles operativos de los usuarios y dan respuesta a la pregunta fundamental "¿A quién está destinada esta aplicación?"
- Las especificaciones y requisitos funcionales (cuando estén disponibles) son una valiosa fuente de información sobre los patrones de uso previstos que también pueden ayudar a identificar los tipos de usuarios y sus perfiles operativos. Cuando las especificaciones funcionales se expresan como historias de usuarios, el formato estándar permite directamente identificar a los tipos de usuarios (es decir, Como <tipo de usuario>, quiero <alguna capacidad> de manera que <algún beneficio>). Del mismo modo, los diagramas y descripciones de los Casos de Uso UML identifican al "actor" del caso de uso.
- La evaluación del uso de datos y métricas obtenidos de aplicaciones similares puede apoyar la identificación de tipos de usuarios y proporcionar algunas indicaciones iniciales del número esperado de usuarios. Se recomienda el acceso a los datos monitorizados de forma automática (por ejemplo, desde la herramienta de administración de un administrador de la Web). Esto incluirá la monitorización de registros y datos tomados a partir del uso del sistema de operación actual cuando se planee una actualización de dicho sistema.
- La monitorización del comportamiento de los usuarios al realizar tareas predefinidas con la aplicación puede dar una idea de los tipos de perfiles operativos que se van a modelar para la prueba de rendimiento. Se recomienda coordinar esta tarea con cualquier prueba de usabilidad planificada (especialmente si se dispone de un laboratorio de usabilidad).

Construir Perfiles Operativos

Se deben seguir los siguientes pasos para identificar y construir perfiles operativos para los usuarios:

- Se adopta un enfoque descendente. Inicialmente se crean perfiles operativos relativamente sencillos y amplios, que sólo se desglosan si es necesario para alcanzar los objetivos de la prueba de rendimiento (véase la sección 4.1.1).
- Los perfiles de usuario particulares pueden ser señalados como relevantes para la prueba de rendimiento si involucran tareas que se ejecutan con frecuencia, requieren transacciones críticas (de alto riesgo) o frecuentes entre los diferentes componentes del sistema, o requieren potencialmente grandes volúmenes de datos para ser transferidos.
- Se revisan y perfeccionan los perfiles operativos con los principales implicados antes de utilizarlos para la creación de perfiles de carga (véase la Sección 4.2.4).
- El sistema sujeto a prueba no siempre está expuesto a cargas impuestas por el usuario. Los perfiles operativos también pueden ser necesarios para las pruebas de rendimiento de los siguientes tipos de sistemas (se debe tener en cuenta que esta lista no es exhaustiva):

Sistemas de Procesamiento por Lotes No Conectados (“Off-line Batch Processing Systems”)

En este caso, la atención se centra principalmente en la capacidad del sistema de procesamiento por lotes (véase la sección 4.2.5) y en su capacidad para completarlo en un período de tiempo determinado. Los perfiles operativos se centran en los tipos de procesamiento que se exigen de los procesos por lotes. Por ejemplo, los perfiles operativos de un sistema de negociación de valores (que normalmente incluye el procesamiento de transacciones en línea y por lotes) pueden incluir los relativos a las operaciones de pago, la verificación de credenciales y la comprobación del cumplimiento de las condiciones legales para determinados tipos de transacciones de valores. Cada uno de estos perfiles operativos daría lugar a que se tomaran diferentes caminos a través del proceso general de lotes para un valor. Los pasos descritos anteriormente para identificar los perfiles operativos de los sistemas en línea basados en el usuario también se pueden aplicar en el contexto del procesamiento por lotes.

Sistemas de Sistemas

Los componentes dentro de un entorno multisistema (que también puede estar integrado) responden a diferentes tipos de entradas de otros sistemas o componentes. Dependiendo de la naturaleza del sistema sujeto a prueba, esto puede requerir el modelado de varios perfiles operativos diferentes para representar eficazmente los tipos de insumos proporcionados por esos sistemas del proveedor. Esto puede implicar un análisis detallado (por ejemplo, de la memoria intermedia³⁰ y colas³¹) junto con los arquitectos del sistema y basado en las especificaciones del sistema y de la interfaz.

4.2.4 Construcción de Perfiles de Carga

Un perfil de carga especifica la actividad que un componente o sistema que se está probando puede experimentar en la producción. Consiste en un número designado de instancias que realizarán las acciones de perfiles operativos predefinidos durante un período de tiempo específico. Cuando las instancias son usuarios, el término "usuarios virtuales" se aplica comúnmente.

La información principal necesaria para crear un perfil de carga realista y repetible es:

- El objetivo de la prueba de rendimiento (por ejemplo, evaluar el comportamiento del sistema bajo cargas de estrés).
- Perfiles operativos que representen con precisión los patrones de uso individuales (véase el apartado 4.2.3).
- Problemas conocidos de capacidad de procesamiento y concurrencia (ver Sección 4.2.5).
- La distribución de cantidad y tiempo con la que se deben ejecutar los perfiles operativos de manera que el SSP experimente la carga deseada. Ejemplos típicos son:
 - Rampas ascendentes: Aumento constante de la carga (por ejemplo, añadir un usuario virtual por minuto).
 - Rampa descendente: Carga en disminución constante.

³⁰ Consultar **

³¹ Consultar **

- Escalones: Cambios instantáneos en la carga (por ejemplo, añadir 100 usuarios virtuales cada cinco minutos).
- Distribuciones predefinidas (por ejemplo, el volumen imita los ciclos de negocio diarios o estacionales).

El siguiente ejemplo muestra la construcción de un perfil de carga con el objetivo de generar condiciones de estrés (iguales o superiores al máximo que se espera que un sistema pueda tratar) para el sistema sujeto a prueba.

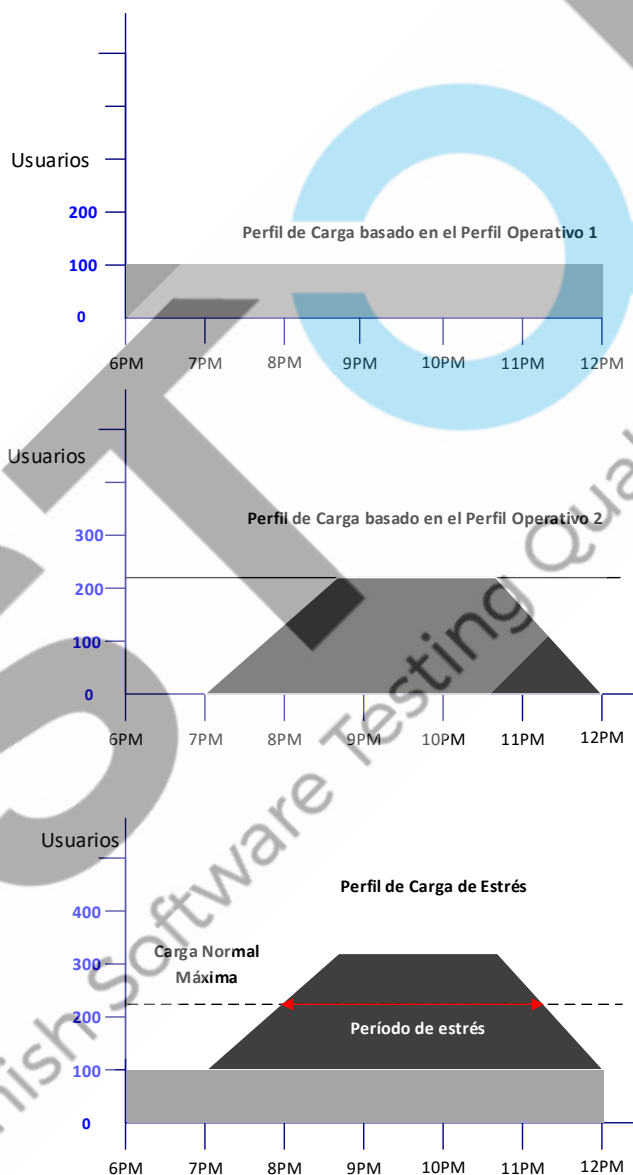


Diagrama 1: Ejemplo de construcción de un perfil de carga de estrés.

En la parte superior del diagrama se muestra un perfil de carga que consiste en una entrada escalonada de 100 usuarios virtuales. Estos usuarios realizan las actividades definidas en el Perfil Operativo 1 a lo largo de toda la duración de la prueba. Esto es típico de muchos perfiles de carga de rendimiento que representan una carga en segundo plano.

El diagrama central muestra un perfil de carga que consiste en una rampa ascendente a 220 usuarios virtuales que se mantiene durante dos horas antes de la rampa descendente. Cada usuario virtual realiza actividades definidas en el Perfil Operativo 2.

El diagrama inferior muestra el perfil de carga que resulta de la combinación de los dos descritos anteriormente. El sistema sujeto a prueba está sometido a un período de estrés de tres horas. Para más ejemplos, consulte [Bath14].

4.2.5 Análisis de la Capacidad de Procesamiento y Concurrencia

Es importante entender los diferentes aspectos de la carga de trabajo: capacidad de procesamiento y concurrencia. Para modelar adecuadamente los perfiles operativos y de carga, se deben tener en cuenta ambos aspectos.

Capacidad de Procesamiento del Sistema

La capacidad de procesamiento del sistema es una medida del número de transacciones de un tipo determinado que el sistema procesa en una unidad de tiempo. Por ejemplo, el número de pedidos por hora o el número de peticiones HTTP por segundo. La capacidad de procesamiento del sistema debe distinguirse de la capacidad de la red, que es la cantidad de datos que se mueven por la red (Sección 2.1).

La capacidad de procesamiento del sistema define la carga del sistema. Desafortunadamente, muy a menudo el número de usuarios concurrentes se utiliza para definir la carga de los sistemas interactivos en lugar de la capacidad de procesamiento. Esto es parcialmente cierto porque ese número es, a menudo, más fácil de encontrar, y parcialmente porque es la forma en que las herramientas de prueba de carga definen la carga. Sin definir los perfiles operativos - lo que cada usuario está haciendo y la intensidad (que también es el rendimiento para un usuario) - el número de usuarios no es una buena medida de la carga. Por ejemplo, si hay 500 usuarios que realizan consultas cortas cada minuto, tenemos una capacidad de procesamiento de 30.000 consultas por hora. Si los mismos 500 usuarios ejecutan las mismas consultas, pero una por hora, el rendimiento es de 500 consultas por hora. Así que hay los mismos 500 usuarios, pero una diferencia de 60 veces entre las cargas y al menos una diferencia de 60 veces en los requisitos de hardware para el sistema.

Normalmente, el modelado de cargas de trabajo se realiza en función del número de usuarios virtuales (hilos de ejecución) y del tiempo de reflexión (retrasos entre las acciones del usuario). Sin embargo, la capacidad de procesamiento del sistema también se define por el tiempo de procesamiento, y ese tiempo puede aumentar a medida que aumenta la carga.

Capacidad de procesamiento del sistema = [número de usuarios virtuales] / ([tiempo de procesamiento] + [tiempo de reflexión])

Por lo tanto, cuando el tiempo de procesamiento aumenta, la capacidad de procesamiento puede disminuir significativamente incluso si todo lo demás permanece igual.

La capacidad de procesamiento del sistema es un aspecto importante cuando se prueban los sistemas de procesamiento por lotes. En este caso, el rendimiento se mide, por lo general, en función del número de transacciones que se pueden realizar en un plazo determinado (por ejemplo, una ventana de procesamiento por lotes nocturno).

Concurrencia

La concurrencia es una medida del número de hilos simultáneos / paralelos de ejecución. Para los sistemas interactivos, puede ser un número de usuarios simultáneos o paralelos. En general, la concurrencia se modela en las herramientas para pruebas de carga mediante la configuración del número de usuarios virtuales.

La concurrencia es una medida importante. Representa el número de sesiones paralelas, cada una de las cuales puede utilizar sus propios recursos. Incluso si el rendimiento es el mismo, la cantidad de recursos utilizados puede variar dependiendo de la concurrencia. Las configuraciones de prueba típicas son sistemas cerrados (desde el punto de vista de la teoría de las colas), en los que se establece el número de usuarios en el sistema (población fija). Si todos los usuarios están esperando la respuesta del sistema en un sistema cerrado, no pueden llegar nuevos usuarios. Muchos sistemas públicos son sistemas abiertos - nuevos usuarios están llegando todo el tiempo incluso si todos los usuarios actuales están esperando la respuesta del sistema.

4.2.6 Estructura Básica de un Guion para una Prueba de Rendimiento

Un guion para una prueba de rendimiento debe simular una actividad de usuario o componente que contribuya a la carga del sistema sujeto a prueba (que puede ser el sistema completo o uno de sus componentes). Inicia las solicitudes al servidor en un orden adecuado y a un ritmo determinado.

La mejor forma de crear guiones de prueba de rendimiento depende del enfoque de generación de carga utilizado (Sección 4.1).

- La forma tradicional es grabar la comunicación entre el cliente y el sistema o componente a nivel de protocolo y luego reproducirlo después de que el guion haya sido parametrizado y documentado. La parametrización resulta en un guion escalable y mantenible, pero la tarea de parametrización puede llevar mucho tiempo.
- Normalmente, la grabación a nivel Interfaz Gráfica de Usuario (GUI por sus siglas en inglés) implica capturar acciones de la Interfaz Gráfica de Usuario (GUI por sus siglas en inglés) de un solo cliente con una herramienta de ejecución de pruebas y ejecutar ese guion con la herramienta de generación de la carga para representar a múltiples clientes.
- La programación puede realizarse utilizando solicitudes de protocolo (por ejemplo, solicitudes HTTP), acciones de Interfaz Gráfica de Usuario (GUI) o llamadas a la Interfaz de Programación de Aplicación (API). En el caso de programar los guiones, se debe determinar la secuencia exacta de las solicitudes enviadas y recibidas desde el sistema real, lo que puede no ser trivial.

Normalmente un guion es uno o varios fragmentos de código (escrito en un lenguaje de programación genérico con algunas extensiones o en un lenguaje especializado) o un objeto, que puede ser presentado a un usuario por la herramienta en una Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés). En ambos casos el guion incluirá las solicitudes del servidor que crean la carga (por ejemplo, solicitudes HTTP) y alguna lógica de programación alrededor de ellas especificando cómo exactamente se invocarán estas solicitudes (por ejemplo, en qué orden, en qué momento, con qué parámetros, con qué parámetros, qué se debe comprobar). Cuanto más sofisticada sea la lógica, mayor será la necesidad de utilizar lenguajes de programación potentes.

Estructura General

A menudo, el guion tiene una sección de inicialización (donde todo se prepara para la parte principal), las secciones principales que pueden ser ejecutadas varias veces, y una sección de limpieza (donde se toman los pasos necesarios para terminar la prueba correctamente).

Recopilar Datos

Para recopilar los tiempos de respuesta, se deben añadir temporizadores al guion para medir el tiempo que tarda una solicitud o una combinación de solicitudes. Las solicitudes programadas deben coincidir con una unidad significativa de trabajo lógico, por ejemplo, una transacción de negocio para añadir un elemento a un pedido o cursar un pedido.

Es importante entender qué es exactamente lo que se mide: en el caso de los guiones a nivel de protocolo, es sólo el tiempo de respuesta del servidor y de la red, mientras que los guiones de la interfaz gráfica de usuario miden el tiempo de extremo a extremo (aunque lo que se mide exactamente depende de la tecnología utilizada).

Verificación de Resultados y Tratamiento de Errores

Una parte importante del guion es la verificación de resultados y el tratamiento de errores. Incluso en las mejores herramientas de prueba de carga, el manejo de errores por defecto tiende a ser mínimo (como la comprobación del código de retorno de petición HTTP), por lo que se recomienda añadir comprobaciones adicionales para verificar lo que realmente devuelven las peticiones. Además, si se requiere algún tipo de saneamiento en caso de error, es probable que sea necesario realizarlo manualmente. Una buena práctica es verificar que el guion está haciendo lo que se supone que debe hacer usando métodos indirectos, por ejemplo, revisar la base de datos para verificar que se ha agregado la información correcta.

Los guiones pueden incluir otras reglas lógicas que especifiquen cuándo y cómo se harán peticiones al servidor. Un ejemplo es el establecimiento de puntos de sincronización, que se realiza especificando que el guion debe esperar a que ocurra un evento en ese punto antes de proseguir. Los puntos de sincronización se pueden utilizar para garantizar que una acción específica se invoque simultáneamente o para coordinar el trabajo entre varios guiones.

Los guiones para prueba de rendimiento son software, por lo que crear un guion para prueba de rendimiento es una actividad de desarrollo de software. Deben incluir control de calidad y pruebas para verificar que el guion se comporta como se espera en todo el rango de datos de entrada.

4.2.7 Implementación de Guiones de Prueba de Rendimiento

Los guiones de prueba de rendimiento se implementan en función de Plan de Prueba de Rendimiento (PTP por sus siglas en inglés) y de los perfiles de carga. Si bien los detalles técnicos de la aplicación variarán según el enfoque y las herramientas utilizadas, el proceso general sigue siendo el mismo. Un guion de rendimiento se crea utilizando un entorno de desarrollo integrado (IDE por sus siglas en inglés) o un editor de guiones, para simular el comportamiento de un usuario o componente. Normalmente el guion se crea para simular un perfil operativo específico (aunque a menudo es posible combinar varios perfiles operativos en un guion con sentencias condicionales).

A medida que se determina la secuencia de peticiones, el guion puede ser grabado o programado dependiendo de la aproximación. La grabación suele asegurar que simula exactamente el sistema real, mientras que la programación se basa en el conocimiento de la secuencia de solicitud adecuada.

Si se utiliza la grabación a nivel de protocolo, un paso esencial después de la grabación en la mayoría de los casos es reemplazar todos los identificadores internos registrados que definen el contexto. Estos identificadores se deben convertir en variables que se pueden cambiar entre ejecuciones con valores apropiados que se extraen de las respuestas a las solicitudes (por ejemplo, un identificador de usuario que se adquiere al iniciar sesión y se debe proporcionar para todas las transacciones posteriores). Esto es una parte de la parametrización del guion, a veces llamada "correlación". En ese contexto, la palabra correlación tiene un significado diferente que cuando se usa en estadística (donde significa relación entre dos o más cosas). Las herramientas de prueba de carga avanzadas pueden hacer alguna correlación de forma automática, por lo que pueden ser transparentes en algunos casos, pero en casos más complejos, la correlación manual o la adición de nuevas reglas de correlación pueden ser necesarias. La correlación incorrecta o la falta de correlación es la razón principal por la que los guiones grabados no se reproducen.

Ejecutar varios usuarios virtuales con el mismo nombre de usuario y acceder al mismo conjunto de datos (como suele ocurrir durante la reproducción de un guion grabado sin ninguna otra modificación más allá de la correlación necesaria) es una forma fácil de obtener resultados engañosos. Los datos podrían estar completamente almacenados en caché (copiados del disco a la memoria para un acceso más rápido) y los resultados serían mucho mejores que en la producción (donde tales datos pueden ser leídos desde un disco). El uso de los mismos usuarios y/o datos también puede causar problemas de concurrencia (por ejemplo, si los datos están bloqueados cuando un usuario los actualiza) y los resultados serían mucho peores que en la producción, ya que el software esperaría a que el bloqueo se libere antes de que el siguiente usuario pueda bloquear los datos para actualizarlos.

Por lo tanto, los guiones y arneses de prueba deben ser parametrizados (es decir, los datos fijos o registrados deben ser reemplazados con valores de una lista de posibles opciones), de modo que cada usuario virtual utilice un conjunto adecuado de datos. El término "apropiado" aquí significa lo suficientemente diferente como para evitar problemas con el almacenamiento en caché y la concurrencia, que es específico para el sistema, los datos y los requisitos de prueba. Esta parametrización adicional depende de los datos en el sistema y de la forma en que el sistema trabaja con estos datos, por lo que normalmente se hace manualmente, aunque muchas herramientas proporcionan ayuda al respecto.

Hay casos en los que algunos datos deben ser parametrizados para que la prueba funcione más de una vez, por ejemplo, cuando se crea una orden y el nombre de la orden debe ser único. A menos que el nombre de la orden sea parametrizado, la prueba fallará tan pronto como intente crear una orden con un nombre existente (registrado).

Para que coincidan con los perfiles operativos, los tiempos de reflexión deben ser insertados y/o ajustados (si están registrados) para generar un número adecuado de solicitudes/capacidad de procesamiento, como se explica en la Sección 4.2.5.

Quando se crean guiones para perfiles operativos separados, se combinan en un escenario que implementa el perfil de carga completo. El perfil de carga controla cuántos usuarios virtuales se inician usando cada guion, cuándo y con qué parámetros. Los detalles exactos de la implementación dependen de la herramienta de prueba de carga específica o del arnés.

4.2.8 Preparación para la Ejecución de la Prueba de Rendimiento

Las principales actividades para preparar la ejecución de la prueba de rendimiento incluyen:

- Configurar el sistema sujeto a prueba.
- Desplegar el entorno.

- Configurar las herramientas de generación y monitorización de la carga y asegurarse de que se recopila toda la información necesaria.

Es importante asegurarse de que el entorno de prueba sea lo más parecido posible al entorno de producción. Si esto no es posible, entonces debe haber una clara comprensión de las diferencias y de cómo se proyectarán los resultados de la prueba en el entorno de producción. En condiciones ideales, se utilizaría el verdadero entorno de producción y los datos, pero las pruebas en un entorno reducido pueden ayudar a mitigar una serie de riesgos de rendimiento.

Es importante recordar que el rendimiento es una función no lineal del entorno, por lo que cuanto más alejado esté el entorno de la norma de producción, más difícil será hacer proyecciones precisas del rendimiento de la producción. La falta de fiabilidad de las proyecciones y el aumento del nivel de riesgo aumentan a medida que el sistema de pruebas se parece menos al de producción.

Las partes más importantes del entorno de prueba son los datos, la configuración de hardware y software y la configuración de la red. El tamaño y la estructura de los datos podrían afectar de forma drástica los resultados de la prueba de carga. El uso de un pequeño conjunto de datos de muestra o de un conjunto de datos con una complejidad de datos diferente para las pruebas de rendimiento puede dar resultados engañosos, especialmente cuando el sistema de producción utilizará un gran conjunto de datos. Es difícil predecir en qué medida el tamaño de los datos afecta al rendimiento antes de realizar una prueba real. Cuanto más se aproximen los datos de prueba a los datos de producción en tamaño y estructura, más fiables serán los resultados de la prueba.

Si se generan o alteran datos durante la prueba, puede ser necesario restaurar los datos originales antes del siguiente ciclo de prueba para asegurarse de que el sistema se encuentra en el estado adecuado.

Si, por cualquier razón, algunas partes del sistema o algunos de los datos no están disponibles para las pruebas de rendimiento, se debe implementar una solución alternativa. Por ejemplo, se puede implementar un stub para reemplazar y emular un componente de terceros responsable del procesamiento de tarjetas de crédito. Este proceso se denomina a menudo "virtualización del servicio" y existen herramientas especiales disponibles para ayudar en este proceso. Es altamente recomendable el uso de estas herramientas para aislar el sistema sujeto a prueba.

Hay muchas maneras de desplegar entornos. Por ejemplo, las opciones pueden incluir el uso de cualquiera de las siguientes:

- Laboratorios de prueba tradicionales internos (y externos).
- La nube como un entorno que utiliza la Infraestructura como un Servicio³² (IaaS³³ por sus siglas en inglés), cuando algunas partes del sistema o todo el sistema se despliega en la nube.
- La nube como un entorno que utiliza Software como un Servicio³⁴ (SaaS³⁵ por sus siglas en inglés), cuando los proveedores proporcionan el servicio de prueba de carga.

Dependiendo de los objetivos específicos y de los sistemas que se vayan a probar, es posible que se prefiera un entorno de prueba a otro. Por ejemplo,

- Para probar el efecto de una mejora del rendimiento (optimización del rendimiento), el uso de un entorno de laboratorio aislado puede ser una mejor opción para ver incluso las pequeñas variaciones introducidas por el cambio.

³² Consultar **

³³ Consultar **

³⁴ Consultar **

³⁵ Consultar **

- Para probar la carga en todo el entorno de producción de principio a fin con el objetivo de asegurarse de que el sistema gestionará la carga sin problemas importantes, la prueba desde la nube o desde un servicio puede ser más apropiada. (Tenga en cuenta que esto sólo funciona para los SSP a los que se puede acceder desde una nube).
- Para minimizar los costes cuando la prueba de rendimiento es limitada en el tiempo, la creación de un entorno de prueba en la nube puede ser una solución más económica.

Cualquiera que sea el enfoque de implementación que se utilice, tanto el hardware como el software deben estar configurados para cumplir con el objetivo y el plan de prueba. Si el entorno coincide con el de producción, debe configurarse de la misma manera. Sin embargo, si hay diferencias, es posible que haya que ajustar la configuración para tener en cuenta estas diferencias. Por ejemplo, si las máquinas de prueba tienen menos memoria física que las máquinas de producción, es posible que sea necesario ajustar los parámetros de la memoria de software (como el tamaño de la pila de Java) para evitar la paginación de la memoria³⁶.

La configuración/emulación adecuada de la red es importante para los sistemas globales y móviles. Para los sistemas globales (es decir, los que tienen usuarios o procesamiento distribuido en todo el mundo) uno de los enfoques puede ser el despliegue de generadores de carga en los lugares donde se encuentran los usuarios. Para los sistemas móviles, la emulación de red sigue siendo la opción más viable debido a las variaciones en los tipos de red que se pueden utilizar. Algunas herramientas de prueba de carga tienen herramientas de emulación de red incorporadas y existen herramientas independientes para la emulación de red.

Las herramientas de generación de carga deben desplegarse correctamente y las herramientas de monitorización deben configurarse para recoger todas las métricas necesarias para la prueba. La lista de métricas depende de los objetivos de la prueba, pero se recomienda recopilar, al menos, métricas básicas para todas las pruebas (véase la Sección 2.1.2).

Dependiendo de la carga, la herramienta específica / enfoque de generación de carga, y la configuración de la máquina, puede ser necesario más de una máquina de generación de carga. Para verificar la configuración, las máquinas involucradas en la generación de carga también deben ser monitorizadas. Esto ayudará a evitar una situación en la que la carga no se mantenga correctamente porque uno de los generadores de carga está funcionando lentamente.

Dependiendo de la configuración y las herramientas utilizadas, las herramientas de prueba de carga deben ser configuradas para crear la carga apropiada. Por ejemplo, se pueden establecer parámetros específicos de emulación de navegador o se puede utilizar la falsificación de IP³⁷ (simulando que cada usuario virtual tiene una dirección IP diferente).

Antes de que se ejecuten las pruebas, se debe validar el entorno y la configuración. Esto se hace generalmente llevando a cabo un conjunto controlado de pruebas y verificando el resultado de las pruebas, así como verificando que las herramientas de monitorización están rastreando la información importante.

Para verificar que la prueba se desarrolla según lo previsto, se pueden utilizar diversas técnicas, como el análisis de bitácoras y la verificación del contenido de la base de datos. Prepararse para la prueba incluye comprobar que la información requerida se registra, que el sistema está en el estado adecuado, etc. Por ejemplo, si la prueba cambia el estado del sistema de forma significativa (añadir/cambiar información en la base de datos), puede ser necesario devolver el sistema al estado original antes de repetir la prueba.

³⁶ Consultar **

³⁷ Consultar **

4.3 Ejecución

La ejecución de la prueba de rendimiento implica la generación de una carga contra el SSP de acuerdo al perfil de carga (normalmente implementado mediante guiones de prueba de rendimiento invocados de acuerdo a un escenario dado), la monitorización de todas las partes del entorno, y la recopilación y conservación de todos los resultados y la información relacionada con la prueba. Por lo general, las herramientas de prueba de carga y los arneses avanzados realizan estas tareas automáticamente (después, por supuesto, de una configuración adecuada). Por lo general, proporcionan una consola que permite monitorizar los datos de rendimiento durante la prueba y realizar los ajustes necesarios (véase el apartado 5.1). Sin embargo, dependiendo de la herramienta utilizada, el SSP y las pruebas específicas que se están ejecutando, pueden ser necesarios algunos pasos manuales.

Las pruebas de rendimiento suelen centrarse en un estado estable del sistema, es decir, cuando el comportamiento del sistema es estable. Por ejemplo, cuando se inician todos los usuarios / subprocesos simulados y están realizando el trabajo como se ha diseñado. Cuando la carga cambia (por ejemplo, cuando se añaden nuevos usuarios), el comportamiento del sistema cambia y se hace más difícil supervisar y analizar los resultados de las pruebas. La etapa de llegar al estado estacionario a menudo se conoce como rampa ascendente, y la etapa de terminar la prueba a menudo se conoce como rampa descendente.

Las pruebas de rendimiento suelen centrarse en un estado estable del sistema, es decir, cuando el comportamiento del sistema es estable. Por ejemplo, cuando se inician todos los usuarios / subprocesos simulados y están realizando el trabajo como se ha diseñado. Cuando la carga cambia (por ejemplo, cuando se añaden nuevos usuarios), el comportamiento del sistema cambia y se hace más difícil supervisar y analizar los resultados de las pruebas. La etapa de llegar al estado estacionario a menudo se conoce como rampa ascendente, y la etapa de terminar la prueba a menudo se conoce como rampa descendente.

A veces es importante probar los estados transitorios, cuando el comportamiento del sistema está cambiando. Esto puede aplicarse, por ejemplo, al registro simultáneo de un gran número de usuarios o a las pruebas de picos. Cuando se prueban estados transitorios es importante entender la necesidad de una monitorización y análisis cuidadosos de los resultados, ya que algunos enfoques estándar -como las medias de monitorización- pueden ser muy engañosos.

Durante la rampa ascendente es aconsejable implementar estados de carga incrementales para monitorizar el impacto de la carga en constante aumento en la respuesta del sistema. De este modo se garantiza que se dispone de tiempo suficiente para la rampa ascendente y que el sistema es capaz de soportar la carga. Una vez que se ha alcanzado el estado estable, es una buena práctica monitorizar que tanto la carga como las respuestas del sistema son estables y que las variaciones aleatorias (que siempre existen) no son sustanciales.

Es importante especificar cómo se deben tratar los fallos para asegurarse de que no se introduzcan problemas en el sistema. Por ejemplo, puede ser importante que el usuario cierre la sesión cuando se produce un fallo para garantizar que se liberan todos los recursos asociados con ese usuario.

Si la monitorización está integrada en la herramienta de prueba de carga y está configurada correctamente, normalmente se inicia al mismo tiempo que la ejecución de la prueba. Sin embargo, si se utilizan herramientas de monitorización autónomas, la monitorización debe iniciarse por separado y debe recogerse la información necesaria para que el análisis posterior pueda llevarse a cabo junto con los resultados de la prueba. Lo mismo ocurre con el análisis de bitácoras. Es esencial sincronizar todas las herramientas utilizadas para poder localizar toda la información relacionada con un ciclo de ejecución de prueba específico.

A menudo, la ejecución de la prueba se monitoriza utilizando la consola de la herramienta de prueba de rendimiento y el análisis de la bitácora en tiempo real para comprobar si hay problemas y errores tanto en la prueba como en el SSP. Esto ayuda a evitar continuar innecesariamente con la realización de pruebas a gran escala, que podrían incluso afectar a otros sistemas si las cosas salen mal (por ejemplo, si se produce un fallo, si los componentes fallan o si las cargas generadas son demasiado bajas o altas). La realización de estas pruebas puede resultar costosa y puede ser necesario detenerlas o realizar algunos ajustes sobre la marcha en la prueba de rendimiento o en la configuración del sistema si la prueba se desvía del comportamiento esperado.

Una técnica para verificar la prueba de carga que se está comunicando de forma directa a nivel de protocolo es ejecutar varios guiones (funcionales) a nivel de Interfaz Gráfica de Usuario (GUI por sus siglas en inglés) o incluso ejecutar perfiles operativos similares de forma manual en paralelo a la prueba de carga en ejecución. Esto comprueba que los tiempos de respuesta informados durante la prueba sólo difieren de los tiempos de respuesta medidos manualmente a nivel de la interfaz gráfica de usuario por el tiempo que se pasa en el lado del cliente.

En algunos casos, cuando se realiza la prueba de rendimiento de forma automatizada (por ejemplo, como parte de la integración continua, como se explica en la sección 3.4), las comprobaciones deben realizarse automáticamente, ya que la monitorización y la intervención manuales pueden no ser posibles. En este caso, el equipo de prueba debe ser capaz de reconocer cualquier desviación o problema y emitir una alerta (por lo general, mientras completa la prueba correctamente). Este enfoque es más fácil de implementar para las pruebas de rendimiento de regresión cuando el comportamiento del sistema es generalmente conocido, pero puede ser más difícil con pruebas de rendimiento exploratorias o pruebas de rendimiento a gran escala y de alto coste, que pueden requerir ajustes que deben realizarse dinámicamente durante la prueba.

4.4 Analizar Resultados e Informar

En la Sección 4.1.2 se discutieron las diversas métricas en un plan de prueba de rendimiento. Definir las métricas por adelantado determina lo que se debe medir en cada ejecución de prueba. Después de completar un ciclo de prueba, se deben recopilar los datos para las métricas definidas.

Cuando se analizan los datos, primero se comparan con el objetivo de la prueba de rendimiento. Una vez que se entiende el comportamiento, se pueden sacar conclusiones que proporcionan un informe resumido significativo que incluye las acciones recomendadas. Estas acciones pueden incluir el cambio de componentes físicos (por ejemplo, hardware, enrutadores), cambio de software (por ejemplo, optimización de aplicaciones y llamadas a bases de datos) y alteración de la red (por ejemplo, equilibrio de carga, enrutamiento).

Normalmente se analizan los siguientes datos:

- **Estado de los usuarios simulados (por ejemplo, virtuales).** Esto debe ser analizado en primer lugar. Normalmente se espera que todos los usuarios simulados hayan podido realizar las tareas especificadas en el perfil operativo. Cualquier interrupción de esta actividad podría reproducir lo que un usuario real pudiera experimentar. Esto hace que sea muy importante ver primero que toda la actividad del usuario se complete, ya que cualquier error que se encuentre puede influir en los otros datos de rendimiento.
- **Tiempo de respuesta de la transacción.** Esto puede ser medido de múltiples maneras, incluyendo mínimo, máximo, media y un percentil (por ejemplo, percentil 90). Las lecturas mínima y máxima muestran los extremos del rendimiento del sistema. El rendimiento medio no es necesariamente indicativo de otra cosa que no sea el promedio matemático y a menudo puede ser sesgado por valores atípicos. El percentil 90 se utiliza a menudo como objetivo, ya que representa la mayoría de los usuarios que alcanzan un umbral de rendimiento específico. No se recomienda exigir el cumplimiento al 100% de los objetivos de rendimiento, ya que los recursos necesarios pueden ser demasiado elevados y el efecto neto para los usuarios será, a menudo, menor.
- **Transacciones por segundo.** Esto proporciona información sobre la cantidad de trabajo realizado por el sistema (capacidad de procesamiento del sistema).
- **Fallos en las transacciones.** Estos datos se utilizan cuando se analizan transacciones por segundo. Los fallos indican que el evento o proceso esperado no se completó o no se ejecutó. Cualquier fallo que se produzca es motivo de preocupación y se debe investigar la causa de raíz. Las transacciones fallidas también pueden resultar en datos de transacciones inválidas por segundo, ya que una transacción fallida tomará mucho menos tiempo que una completada.
- **Impactos (o peticiones) por segundo³⁸.** Esto proporciona una idea del número de visitas a un servidor por parte de los usuarios simulados durante cada segundo de la prueba.
- **Tasa de transferencia de la red.** Esto se suele medir en bits por intervalo de tiempo, como en bits por segundo. Esto representa la cantidad de datos que los usuarios simulados reciben del servidor cada segundo. (véase el apartado 4.2.5)
- **Respuestas HTTP.** Estos se miden por segundo e incluyen posibles códigos de respuesta como: 200, 302, 304, 404, este último indica que no se encuentra una página.

³⁸ Consultar **

Aunque gran parte de esta información puede presentarse en tablas, las representaciones gráficas facilitan la visualización de los datos y la identificación de tendencias.

Entre las técnicas utilizadas en el análisis de datos se encuentran las siguientes:

- Comparación de los resultados con los requisitos establecidos.
- Observar las tendencias de los resultados.
- Técnicas de control de calidad estadística.
- Identificación de errores.
- Comparación de los resultados esperados y reales.
- Comparación de los resultados con resultados de prueba anteriores.
- Verificar el correcto funcionamiento de los componentes (por ejemplo, servidores, redes).

Identificar la correlación entre las métricas puede ayudarnos a entender en qué punto el rendimiento del sistema comienza a degradarse. Por ejemplo, ¿cuántas transacciones por segundo se procesaron cuando la CPU alcanzó el 90% de su capacidad y el sistema se comenzó a ralentizar?

El análisis puede ayudar a identificar la causa raíz de la degradación o fallo del rendimiento, lo que a su vez facilitará la corrección. Las pruebas de confirmación ayudarán a determinar si la acción correctiva abordó la causa raíz.

Suministro de Información

Los resultados de los análisis se consolidan y se comparan con los objetivos establecidos en el plan de prueba de rendimiento. Estos pueden ser comunicados en el informe general del estado de la prueba junto con otros resultados de la prueba, o incluidos en un informe dedicado a la prueba de rendimiento. El nivel de detalle comunicado debe corresponderse con las necesidades de las partes interesadas. Las recomendaciones basadas en estos resultados normalmente se refieren a los criterios de entrega del software (incluido el entorno destino) o a las mejoras de rendimiento necesarias.

Un informe típico de una prueba de rendimiento puede incluir los siguientes elementos:

Resumen Ejecutivo

Esta sección se completa una vez que se han realizado todas las pruebas de rendimiento y se han analizado y comprendido todos los resultados. El objetivo es presentar conclusiones, hallazgos y recomendaciones concisas y comprensibles para la dirección, con el fin de lograr un resultado viable.

Resultados de la Prueba

Los resultados de la prueba pueden incluir parte o toda la siguiente información:

- Un resumen con una explicación y elaboración de los resultados.
- Resultados de una prueba de línea de base que sirve como "instantánea" del rendimiento del sistema en un momento dado y constituye la base para la comparación con las pruebas subsiguientes. Los resultados deben incluir la fecha/hora de inicio de la prueba, el objetivo del usuario concurrente, la capacidad de procesamiento medida y los hallazgos clave. Los hallazgos clave pueden incluir la tasa de error global medida, el tiempo de respuesta y la capacidad de procesamiento media.
- Un diagrama de alto nivel que muestre cualquier componente arquitectónico que pueda (o haya) impactado los objetivos de la prueba.
- Un análisis detallado (tablas y gráficos) de los resultados de la prueba que muestra los tiempos de respuesta, tasas de transacción, tasas de error y análisis de rendimiento. El análisis también incluye una descripción de lo que se observó, como el momento en que una aplicación estable se volvió inestable y la fuente de los fallos (por ejemplo, servidor web, servidor de bases de datos).

Registros de Prueba/Información Registrada

Se debe consignar un registro por cada ejecución de la prueba. La bitácora (o registro) normalmente incluye lo siguiente:

- Fecha/hora de inicio de la prueba.
- Duración de la prueba.
- Guiones utilizados para la prueba (incluida la combinación de guiones si se utilizan varios guiones) y datos de configuración de guiones relevantes.
- Archivos de datos de prueba utilizados en la prueba.
- Nombre y ubicación de los archivos de datos/bitácoras creadas durante la prueba.
- Configuración HW/SW probada (especialmente cualquier cambio entre ejecuciones).
- Uso medio y pico (máximo) de CPU y RAM en servidores web y de bases de datos.
- Notas sobre el rendimiento alcanzado.
- Defectos identificados.

Recomendaciones

Las recomendaciones resultantes de la prueba pueden incluir lo siguiente:

- Cambios técnicos recomendados, como la reconfiguración de hardware o software o de la infraestructura de red.
- Áreas identificadas para un análisis posterior (por ejemplo, análisis de los registros del servidor web para ayudar a identificar las causas raíz de los problemas y/o errores).
- Monitorización adicional requerida de pasarelas, servidores y redes para poder obtener datos más detallados que permitan medir las características y tendencias del rendimiento (por ejemplo, la degradación).

5 Herramientas

Duración: 90 minutos

Palabras Clave

generador de carga ("load generator")
gestión de carga ("load management")
herramienta de monitorización ("monitoring tool")
herramienta de prueba de rendimiento ("performance testing tool")

Objetivos de Aprendizaje para Herramientas

5.1 Soporte de Herramientas

NBPR-5.1.1 (K2) Entender la forma en que las herramientas apoyan la prueba de rendimiento.

5.2 Adecuación de las Herramientas

NBPR-5.2.1 (K4) Evaluar la idoneidad de las herramientas de prueba de rendimiento en un escenario de proyecto determinado.

5.1 Soporte de Herramientas

Las herramientas para la prueba de rendimiento incluyen los siguientes tipos de herramientas para apoyar la prueba de rendimiento.

Generadores de Carga

El generador, a través de un IDE, un editor de guiones o un paquete de herramientas, es capaz de crear y ejecutar múltiples instancias de cliente que simulan el comportamiento del usuario de acuerdo con un perfil operativo definido. La creación de múltiples instancias en períodos cortos de tiempo causará carga en un sistema bajo prueba. El generador crea la carga y también recoge las métricas para su posterior comunicación.

Al ejecutar las pruebas de rendimiento, el objetivo del generador de carga es imitar el mundo real tanto como sea práctico. Esto a menudo significa que se necesitan solicitudes de usuarios procedentes de varias ubicaciones, no sólo de la ubicación de la prueba. Los entornos que se configuran con múltiples puntos de presencia se distribuirán donde se origina la carga, de modo que no todo provenga de una sola red. Esto proporciona realismo a la prueba, aunque a veces puede sesgar los resultados si los saltos en las redes intermedias crean retrasos.

Consola de Gestión de la Carga

La consola de gestión de carga proporciona el control para iniciar y detener los generadores de carga. La consola también agrega métricas de las diversas transacciones que se definen dentro de las instancias de carga utilizadas por el generador. La consola permite visualizar informes y gráficos de las ejecuciones de las pruebas y soporta el análisis de resultados.

Herramienta de Monitorización

Las herramientas de monitorización se ejecutan simultáneamente con el componente o sistema sujeto a prueba y supervisan, registran y/o analizan el comportamiento del componente o sistema. Los componentes típicos que son monitorizados incluyen colas de servidores web, memoria del sistema y espacio en disco. Las herramientas de monitorización pueden apoyar eficazmente el análisis de la causa raíz del rendimiento en un sistema sujeto a prueba y también se puede utilizar para controlar un entorno de producción cuando se entrega el producto. Durante la prueba de rendimiento también se pueden utilizar monitores de ejecución en el propio generador de carga.

Los modelos de licencia para las herramientas de prueba de rendimiento incluyen la licencia tradicional basada en el sitio/sede con propiedad total, un modelo de licencia de pago por uso basado en la nube y licencias de código abierto que pueden utilizarse libremente en un entorno definido o a través de ofertas basadas en la nube. Cada modelo implica una estructura de costes diferente y puede incluir un mantenimiento continuo. Lo que está claro es que para cualquier herramienta seleccionada, entender cómo funciona esa herramienta (a través de la capacitación y/o el autoestudio) requerirá tiempo y presupuesto.

5.2 Adecuación de las Herramientas

Se deben tener en cuenta los siguientes factores al seleccionar una herramienta de prueba de rendimiento:

Compatibilidad

En general, se selecciona una herramienta para la organización y no sólo para un proyecto. Esto significa que se deben tener en cuenta los siguientes factores en la organización:

- **Protocolos:** Como se describe en la Sección 4.2.1, los protocolos son un aspecto muy importante para la selección de herramientas de rendimiento. Entender qué protocolos utiliza un sistema y cuáles de ellos serán probados proporcionará la información necesaria para evaluar la herramienta de prueba adecuada.
- **Interfaces a componentes externos:** Puede ser necesario considerar las interfaces con componentes de software u otras herramientas como parte de los requisitos de integración completa para cumplir con los requisitos de proceso u otros requisitos de interoperabilidad (por ejemplo, la integración en el proceso de IC).
- **Plataformas:** La compatibilidad con las plataformas (y sus versiones) dentro de una organización es esencial. Esto se aplica a las plataformas utilizadas para alojar las herramientas y a las plataformas con las que las herramientas interactúan para la monitorización y/o generación de cargas.

Escalabilidad

Otro factor a considerar es el número total de simulaciones de usuarios concurrentes que la herramienta puede tratar. Esto incluirá varios factores:

- Número máximo de licencias requeridas.
- Requisitos de configuración de la estación de trabajo/servidor de generación de cargas.
- Capacidad para generar carga desde múltiples puntos de presencia (por ejemplo, servidores distribuidos).

Capacidad de ser Entendido

Otro factor a considerar es el nivel de conocimiento técnico necesario para utilizar la herramienta. Esto a menudo se pasa por alto y puede llevar a que probadores inexpertos configuren incorrectamente las pruebas, lo que a su vez proporciona resultados inexactos. Para las pruebas que requieren escenarios complejos y un alto nivel de programabilidad y personalización, los equipos deben asegurarse de que el probador cuente con las competencias, los antecedentes y la capacitación necesarios.

Monitorización

¿Es suficiente el control que proporciona la herramienta? ¿Existen otras herramientas de monitorización disponibles en el entorno que puedan utilizarse para complementar la monitorización de la herramienta? ¿Se puede correlar la monitorización con las transacciones definidas? Todas estas preguntas deben ser contestadas para determinar si la herramienta proporcionará la monitorización requerida por el proyecto.

Cuando la monitorización es un programa/herramientas/pila completa, se puede utilizar para monitorizar el entorno de producción cuando se entrega el producto.



6 Referencias

6.1 Estándares

- [ISO25000] ISO/IEC 25000:2005, Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE)

6.2 Documentos del ISTQB

- [ISTQB_UT_SYL] ISTQB Foundation Level Usability Testing Syllabus, Version 2018
- [ISTQB_ALTA_SYL] ISTQB Advanced Level Test Analyst Syllabus, Version 2012
- [ISTQB_ALTTA_SYL] ISTQB Advanced Level Technical Test Analyst Syllabus, Version 2012
- [ISTQB_ALTM_SYL] ISTQB Advanced Level Test Manager Syllabus, Version 2012
- [ISTQB_FL_SYL] ISTQB Foundation Level (Core) Syllabus, Version 2018
- [ISTQB_FL_AT] ISTQB Foundation Level Agile Tester Syllabus, Version 2014
- [ISTQB_GLOSSARY] ISTQB Glossary of Terms used in Software Testing, <http://glossary.istqb.org>

6.3 Bibliografía

[Anderson01] Lorin W. Anderson, David R. Krathwohl (eds.) "A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives", Allyn & Bacon, 2001, ISBN 978-0801319037

[Bath14] Graham Bath, Judy McKay, "The Software Test Engineer's Handbook", Rocky Nook, 2014, ISBN 978-1-933952-24-6

[Molyneaux09] Ian Molyneaux, "The Art of Application Performance Testing: From Strategy to Tools", O'Reilly, 2009, ISBN: 9780596520663

[Microsoft07] Microsoft Corporation, "Performance Testing Guidance for Web Applications", Microsoft, 2007, ISBN: 9780735625709