Perspectiva de rendimiento y escalabilidad

USAC - Análisis y diseño de sistemas 2 - 1er semestre 2016
Ing. Ricardo Morales

Descripción

Calidad deseada

- La habilidad del sistema de ejecutarse predeciblemente dentro de su perfil de rendimiento definido y manejar incrementos en volúmenes de procesamiento en el futuro si es requerido

Aplicabilidad

- Cualquier sistema con requerimientos de rendimiento complejos, no claros o ambiciosos; sistemas que incluyen elementos con rendimiento desconocido, y sistemas cuya expansión puede ser significativa

Descripción (II)

Temas de interés

- Tiempo de respuesta
- Desempeño
- Escalabilidad
- Predictibilidad
- Requerimientos de recursos de hardware
- Comportamiento en carga alta

Actividades

- Capturar requerimientos de rendimiento
- Crear modelos de rendimiento
- Analizar modelos de rendimiento
- Conducir pruebas prácticas
- Evaluar contra los requerimientos
- Trabajar nuevamente la arquitectura

Descripción (III)

Tácticas de arquitectura

- Optimizar procesamiento repetido
- Reducir contención vía replicación
- Priorizar procesamiento
- Consolidar carga relacionada
- Distribuir procesamiento en el tiempo
- Minimizar el uso de recursos compartidos
- Reutilizar recursos y resultados
- Particionar y paralelizar
- Scale up o scale out

Problemas

- Metas de rendimiento y escalabilidad imprecisas
- Modelos no realísticos
- Uso de medidas simples para casos complejos
- Particionamiento no adecuado
- Suposiciones de ambientes y plataformas no válidos
- Concurrencia relacionada con contención
- Contención de base de datos
- Creación descuidada de recursos

Descripción (IV)

- La razón fundamental del interés en el rendimiento es que las tareas que los sistemas deben ejecutar se han vuelto mas complejas y la demanda (en términos de complejidad, número de transacciones, número de usuarios, etc.) también ha crecido
- El rendimiento depende no solo del poder de procesamiento de hardware, sino también de cómo está configurado, la forma en que se manejan los recursos y la forma en que el software está escrito impacta el rendimiento
- La propiedad de escalabilidad de un sistema está relacionada al rendimiento, que se enfoca en la predictibilidad del rendimiento del sistema conforme la carga se incrementa

Aplicabilidad a vistas

- La vista de contexto identifica todas las interfaces externas del sistema y aplicando esta perspectiva se resaltan los requerimientos de rendimiento o problemas potenciales con dichas interfaces
- Esto permite identificar estas restricciones temprano en el diseño del sistema , entender su impacto e identificar acciones de mitigación apropiadas

Contexto

- Aplicando esta perspectiva puede revelar la necesidad de cambios y compromisos de las estructuras funcionales para alcanzar los requerimientos de rendimiento del sistema
- Los modelos de esta vista también proveen entradas para la creación de modelos de rendimiento

Funcional

Aplicabilidad a vistas (II)

- Provee entradas útiles para modelos de rendimiento, identificando recursos compartidos y los requerimientos transaccionales de cada uno
- Conforme se aplica esta perspectiva, se pueden identificar aspectos de la vista de información como obstáculos para el rendimiento o la escalabilidad
- Adicionalmente, considerando la escalabilidad pueden sugerirse que elementos de la vista de información puedan ser replicados o distribuidos para esa meta

Información

- Aplicando esta perspectiva puede resultar en cambios al diseño de concurrencia debido a la identificación de problemas como contención excesiva por recursos clave
- Considerar el rendimiento y escalabilidad puede resultar en que la concurrencia sea un elemento de diseño mas importante para alcanzar dichos requerimientos
- Elementos de la vista de concurrencia también pueden proveer métricas de calibración para modelos de rendimiento

Concurrencia

Aplicabilidad a vistas (III)

- Una de las posibles salidas de aplicar esta perspectiva es un conjunto de guías para desarrollo, relacionadas a rendimiento y escalabilidad

Desarrollo

- La vista de deployment es una entrada crucial para el proceso de evaluar rendimiento y escalabilidad
- Varias partes del modelo de rendimiento del sistema se derivan del contenido de esta vista, que también provee métricas críticas de calibración

Deployment

- La aplicación de esta perspectiva subraya la necesidad de capacidades de monitoreo y administración de rendimiento

Operacional

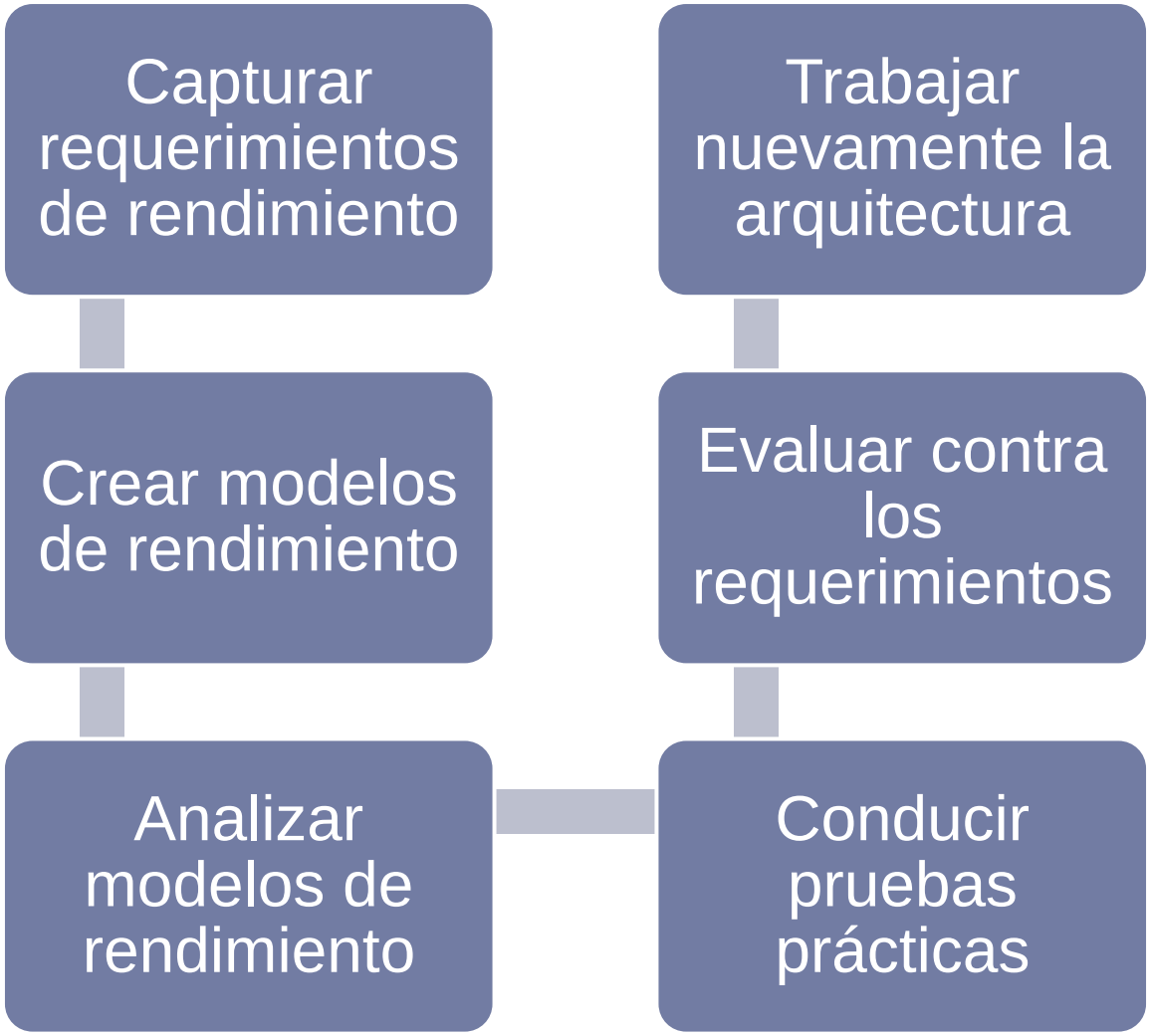
Temas de interés

- Tiempo de respuesta
  - Es la cantidad de tiempo que toma que se complete determinada interacción con el sistema
  - Responsiveness, considera que tan rápido responde el sistema a cargas rutinarias como solicitudes interactivas de los usuarios, usualmente en segundos
  - Tiempo de respuesta, es el tiempo que toma completar tareas mas grandes, usualmente en minutos u horas
- Desempeño
  - Se define como la cantidad de trabajo que el sistema puede manejar en un período de tiempo
  - En general, entre mas pequeño es el tiempo de respuesta a las transacciones, mayor es el desempeño que el sistema puede alcanzar
  - Sin embargo, conforme la carga del sistema aumenta, el tiempo de respuesta del sistema se incrementa, lo que afecta el desempeño

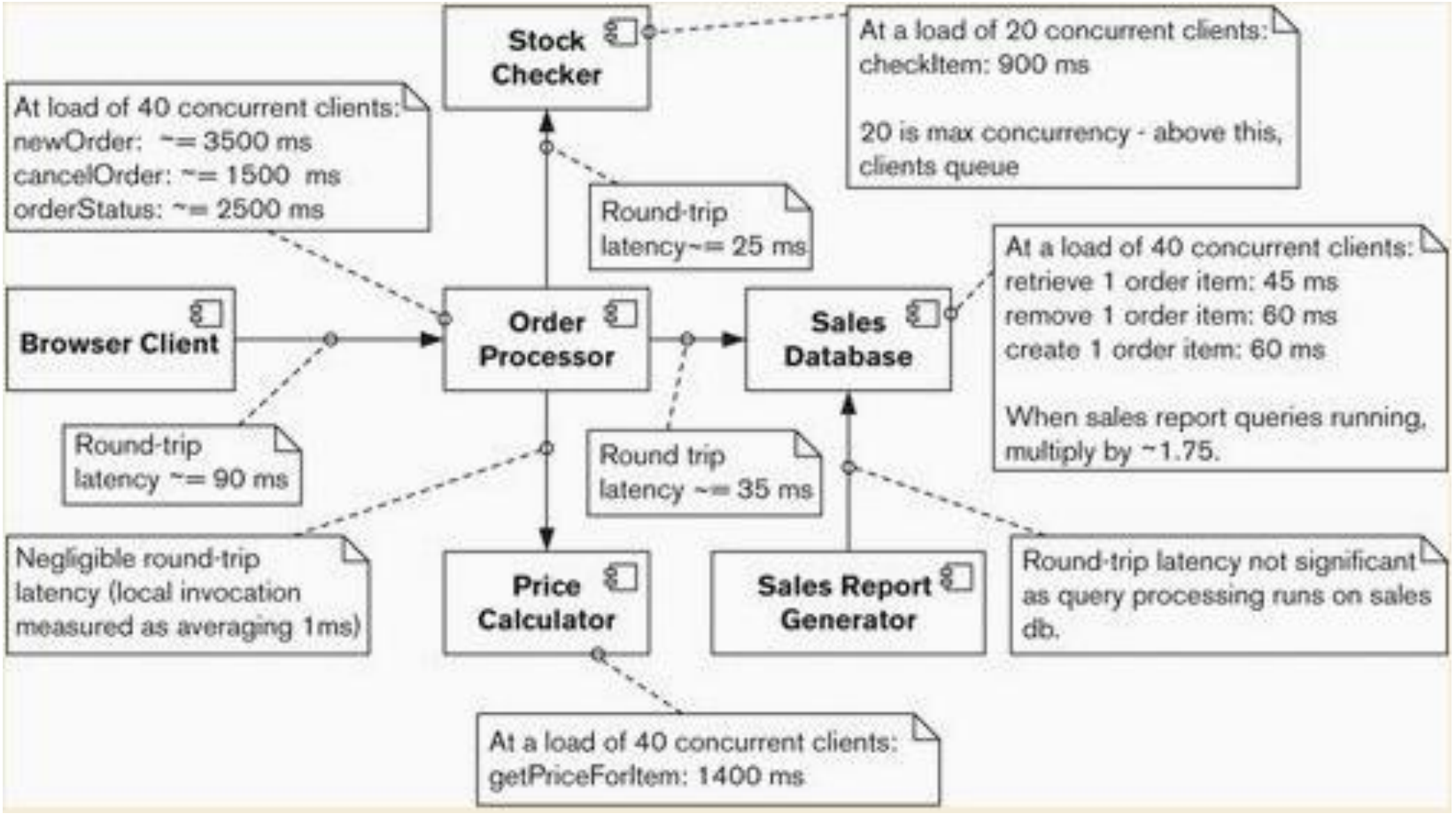
Temas de interés (II)

- Escalabilidad
  - Es la habilidad del sistema para manejar un incremento en la carga, que puede deberse a un incremento en el número de solicitudes, transacciones, mensajes o jobs que el sistema maneja, o un incremento en la complejidad de estas tareas
  - Se debe considerar tanto la escalabilidad a largo plazo, como la escalabilidad transitoria (aumentos en períodos de tiempo cortos)
- Predictibilidad
  - Transacciones similares deben completarse en períodos de tiempo similares, sin importar en que momento se ejecuten
- Requerimientos de recursos de hardware
  - En general, mas hardware significa mayor desempeño y mejores tiempos de respuesta, a un costo mayor
- Comportamiento en carga alta
  - Se deben identificar los picos en carga y validar como se comporta el sistema

Actividades



Modelo de rendimiento



Tácticas de arquitectura

- Optimizar procesamiento repetido
  - La mayoría de los sistemas tiene un número pequeño de operaciones comunes (80-20) en las que el sistema gasta la mayoría de tiempo de ejecución
  - Costo operacional total= costo de invocación de operación x frecuencia de invocación de operación
  - Carga sistema=$\sum$ costo operacional total, por unidad de tiempo
  - Para enfocar el esfuerzo de ingeniería de rendimiento, se puede tomar las operaciones que tengan un mayor costo operacional total
- Reducir contención vía replicación
  - La contención es una fuente de problemas de rendimiento
  - Una posible solución para algunos problemas de contención es replicar elementos del sistema- hardware, software o datos – una táctica que debe combinarse usualmente con tácticas de particionamiento y paralelización
  - Sin embargo, es posible que esto se deba considerar desde el diseño del sistema para tomar ventaja desde esto

Tácticas de arquitectura (II)

- Priorizar procesamiento
  - La carga del sistema puede variar en términos de importancia, desde transacciones críticas hasta otras menos importante
  - Para evitar el hecho que operaciones menos importantes impacten el tiempo de respuesta de operaciones críticas, la carga de trabajo del debe particionarse en grupos de prioridad y que el sistema tenga la habilidad de priorizar la carga a ser procesada
- Consolidar carga relacionada
  - El procesamiento de la mayoría de operaciones en un sistema de información requiere que cierto contexto esté disponible para el procesamiento
  - Una forma de mejorar este tema, es consolidar tareas relacionadas en grupos de solicitudes relacionadas
  - Si se requieren procesos de inicialización comunes a varias tareas, es mejor realizarlo una sola vez en lugar de hacerlo para cada tarea individual

Tácticas de arquitectura (III)

- Distribuir procesamiento en el tiempo
  - Algunos sistemas necesitan procesar una carga similar continuamente durante el día
  - Una estrategia para reducir la carga del sistema, la contención por recursos y los problemas de rendimiento es buscar eliminar los picos de procesamiento
  - Aunque a veces es imposible, se debe analizar la carga del sistema para identificar carga que pueda ser pospuesta para mejorar el uso de recursos

Tácticas de arquitectura (IV)

- Minimizar el uso de recursos compartidos
  - En cualquier momento, una tarea no ociosa, está en uno de 2 estados: haciendo uso de un recurso o esperando por un recurso ocupado
  - Entre mas ocupado está el sistema y la contención por recursos compartidos se incrementa, los tiempos de espera impactan el rendimiento
  - Para minimizar el uso de recursos compartidos se puede considerar.
    - Usar técnicas de multiplexado de hardware para eliminar cuellos de botella potenciales
    - Favorecer transacciones simples y cortas, sobre transacciones largas y complejas (bloqueo de recursos)
    - No bloquear recursos en tiempo de humanos (mientras se espera por presionar una tecla)
    - Tratar de accesar recursos compartidos de forma no exclusiva, cuando sea posible

Tácticas de arquitectura (V)

- Reutilizar recursos y resultados
  - Algunos recursos involucrados en solicitudes de procesamiento son caros, debido a que son computacionalmente caros de crear o necesitan ser obtenidos de servicios que toman un tiempo significativo antes de retornar los resultados
  - Una táctica para pasos caros en un proceso, es reutilizar los resultados de operaciones caras, al tenerlos en cache una vez son creados y reutilizarlos cuando sean requeridos

Tácticas de arquitectura (VI)

- Particionar y paralelizar
  - Si el sistema involucra procesos largo y lentos, una posible forma de reducir los tiempos de respuesta es particionar en procesos mas pequeños y ejecutarlos en paralelo
  - La efectividad de este enfoque depende de:
    - Si todo el proceso puede ser partido en subprocesos de forma rápida y eficiente
    - Si los subprocesos resultantes pueden ser ejecutados independientemente para permitir un procesamiento en paralelo efectivo
    - Cuanto tiempo toma consolidar las salidas de los subprocesos en un único resultado
    - Si hay suficiente capacidad disponible para procesar los subprocesos mas rápidamente que manejándolo en un solo proceso
  - Este enfoque podría alcanzar una reducción en el tiempo de respuesta a expensas de requerir mas recursos de procesamiento

Tácticas de arquitectura (VII)

- Scale up o scale out
  - Scale up, reemplazar el hardware existente con componentes similares pero de mayor capacidad
  - Scale out, agregar mas componentes similares a los ya usados por el sistema
- Degradar elegantemente
  - El sistema puede considerar elementos de monitoreo que detecten y manejen fallas
  - Una forma de manejo es prevenir la sobrecarga de componentes internos
  - Otra táctica es rechazar carga adicional cuando el sistema está sobre cargado

Tácticas de arquitectura (VIII)

- Usar procesamiento asíncrono
  - Una forma de mejorar el tiempo de respuesta percibido es realizar algún procesamiento de forma asíncrona, después que el sistema ha retornado una respuesta al usuario
  - Se debe tener cuidado con esta táctica, considerando la complejidad y manejo de condiciones de error
- Relajar la consistencia transaccional
  - Si es posible diseñar el sistema de manera que diferentes partes del sistema reciban actualizaciones relacionadas a una sola transacción en diferentes momentos, esto abre varias oportunidades para particionar la carga y diferir actualizaciones de base de datos
  - Esto depende de los requerimientos del negocio