

Sistema de subastas: Una aproximación a los mecanismos REST, Colas y Publicador-Suscriptor

Nicolas Montañez, Natalia Mejia, Nicolas Reyes, Juan Esteban Muñoz

Taller 1 - Ejemplo Mark Richards, Ingeniería de sistemas,
Pontificia Universidad Javeriana, Bogotá, Colombia

Abstract: Este documento de diseño presenta la implementación y evaluación de tres técnicas de integración de sistemas: Publicador/Subscriber, Colas y REST, siguiendo el ejemplo de integración de Mark Richards, et al. Se desarrollaron y probaron tres proyectos, cada uno con una técnica diferente, y se midió su desempeño en términos de tiempo de respuesta.

El documento incluye los resultados de las pruebas y una tabla comparativa, además de las instrucciones para ejecutar cada proyecto. Los resultados proporcionan una base para elegir la técnica de integración más adecuada según el rendimiento observado.

1. Introducción

En el ciclo de vida del desarrollo de software, una de las etapas iniciales con mayor impacto es el diseño arquitectónico, ya que representa las bases del cómo será construido el software a lo largo del desarrollo. Este diseño arquitectónico, además de representar algunas decisiones técnicas, tiene una vista más amplia sobre el negocio, sus necesidades y el entendimiento de las restricciones y relaciones entre el sistema, sus usuarios y demás interesados.

Por otro lado, el arquitecto se encarga de guiar la solución más que de especificar las tecnologías a usar, por lo que se priorizan decisiones como los mecanismos de comunicación o técnicas de integración sobre otras más técnicas como el lenguaje de programación o entornos de desarrollo. Estas decisiones se basan principalmente en el análisis de trade-offs (compensación), es decir, considerar los impactos positivos y negativos que tiene una solución frente a otra e identificar lo más primordial para el negocio.

Así entonces, en este taller se busca diseñar e implementar un problema sencillo, tomado del libro Fundamentals of Software Architecture de Mark Richards, et al. [1], con el fin de analizar los trade-offs que implican distintos mecanismos de comunicación.

2. Problema

El problema expuesto en la sección Analyzing Trade-Offs, trata de un sencillo sistema de subastas en el que, cuando un usuario oferta por un producto, el Bid Producer genera un mensaje que debe ser enviado a los demás servicios: Bid capture, Bid tracking y Bid analytics, como se ve en la Figura 1.

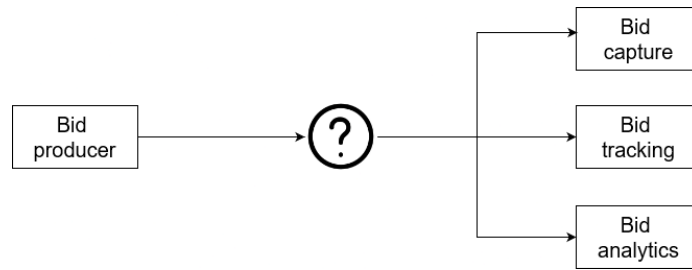


Fig. 1. Ejemplo de Análisis de Trade-Offs - Sistema de subastas

El autor propone dos posibles soluciones basadas en mensajería: Colas y Tópicos, haciendo alusión a los mecanismos punto a punto y publicador-subscriptor, correspondientemente. Sumado a esto, se propone una solución basado en el mecanismo de REST [2].

Dado que se pretende estudiar y probar los mecanismos de comunicación disponibles, las funcionalidades de cada componente están limitadas a enviar y/o recibir información.

3. Diseño

3.1. Colas

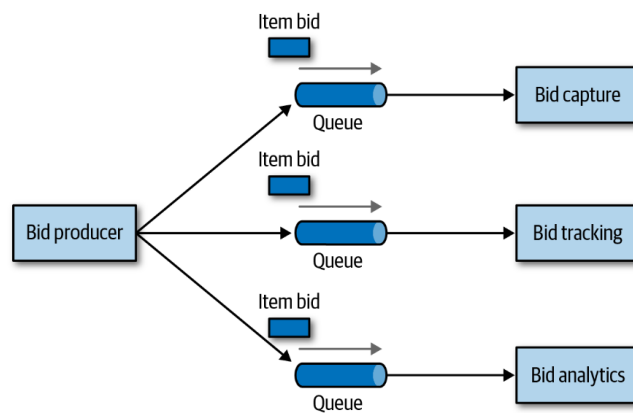


Fig. 2. Comunicación por Colas

3.2. Tópicos

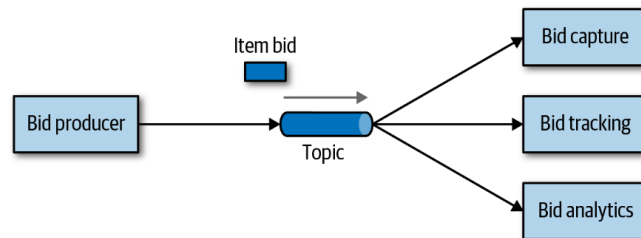


Fig. 3. Comunicación por Tópicos

3.3. REST

Para la comunicación REST, se establece un modelo Cliente/Servidor, donde cada uno de los servicios que debe recibir el mensaje de las pujas, expone una ruta en donde el servicio Bid Producer, envía un mensaje personalizado de forma síncrona, es decir, espera una respuesta satisfactoria de cada uno. Cada uno de los servicios que recibe el mensaje, se encarga únicamente de registrar su llegada en forma de logs. Visto en la Figura 8

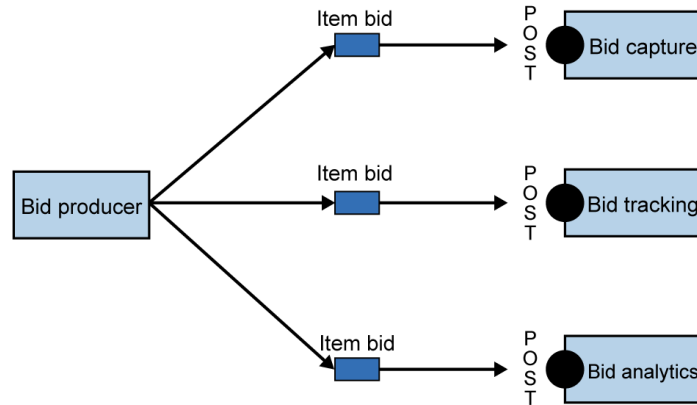


Fig. 4. Comunicación REST

Por otro lado, como accionador del evento inicial del Bid Producer, este expone una última ruta por donde se espera recibir la información de la puja y, posteriormente, personalizar el mensaje que envía a cada uno de los otros servicios.

Finalmente, la ejecución de este mecanismo se implementó a través de contenedores que permiten aislar cada servicio en un entorno de ejecución propio, además de permitir conectar estos servicios a través de una misma red local dentro del host principal, como se puede ver en la Figura 5

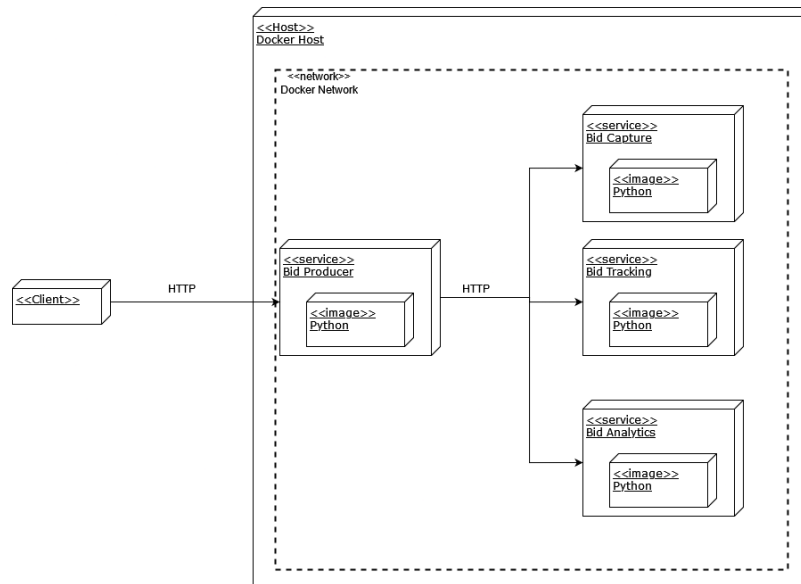


Fig. 5. Despliegue REST

4. Protocolo de Pruebas

Se presenta el siguiente plan de pruebas:

4.1. Alcance

Lo que se pretende es medir el tiempo de respuesta bajo cierta carga a tres técnicas de integración diferentes (Rest, Colas y Tópicos) para su posterior comparación y análisis.

- Tiempo de respuesta: Tiempo total que tarda una solicitud en ser procesada desde el momento en que se envía hasta el momento en que se recibe la respuesta

4.2. Descripción de las pruebas

Herramienta: JMeter V 5.6.3

Especificaciones de máquina usada:

- RAM: 12,0 GB
- Procesador: Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz 2.59 GHz (2 processors)

Descripción: Con ayuda de las herramientas que proporciona Jmeter se ajustaron las pruebas de la siguiente forma:

- Número de usuarios: 100
- Periodo de incremento: 60 segundos
- Repetición: 3

En pocas palabras, esto quiere decir que en un periodo de 60 segundos, se va incrementando de forma gradual el número de usuarios diferentes con un máximo de 100, cada uno de ellos ejecutando 3 veces una solicitud POST para el caso de REST. Al momento de terminar la ejecución de sus 3 solicitudes, el usuario (Hilo) se cierra, de modo que no se llega realmente a tener 100 usuarios concurrentes. De forma similar se ejecutaron las pruebas para queues y topics.

4.3. Ejecución de Pruebas

Para el correcto funcionamiento de las pruebas para REST se inician los receptores de mensaje primero (analytics, capture, tracking). Posteriormente se inicia el producer, de modo que éste se conecte a los receptores. Una vez hecho esto, y con Jmeter configurado como se mencionó en la descripción de las pruebas, se inicia la prueba en Jmeter.

Para el funcionamiento de las pruebas de colas y tópicos, es necesario iniciar únicamente los receptores de mensajes (analytics, capture, tracking), pues Jmeter se encarga de simular el comportamiento de los productores mediante un script. Una vez hecho esto se le puede dar "start" a la herramienta de modo que simule la carga con la configuración mencionada.

5. Resultados

Una vez obtenidos los datos, se sacó un promedio del tiempo de respuesta para cada técnica de integración, obteniendo los siguientes resultados, representados en una tabla:

REST	Queues	Topics
6147,4 ms	10,73 ms	9,94 ms

TABLE I
Promedio de tiempos de respuesta

Como se ve reflejado en la tabla hay una gran diferencia en REST con respecto a queues y topics, resultados que tienen sentido por la forma en la que funciona el envío y procesamiento de mensajes. En REST hay un flujo síncrono, lo que resulta en un tiempo de respuesta mayor, ya que se envía el mensaje desde "producer" y se espera a que los otros 3 procesos respondan, a diferencia de las colas y los tópicos que envían los mensajes sin esperar respuesta.

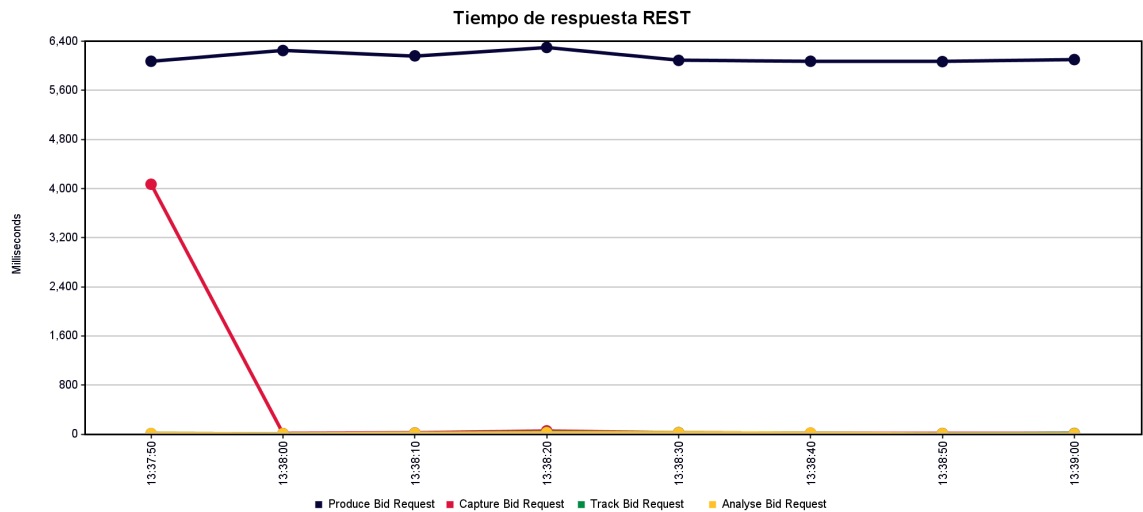


Fig. 6. Tiempo de respuesta REST

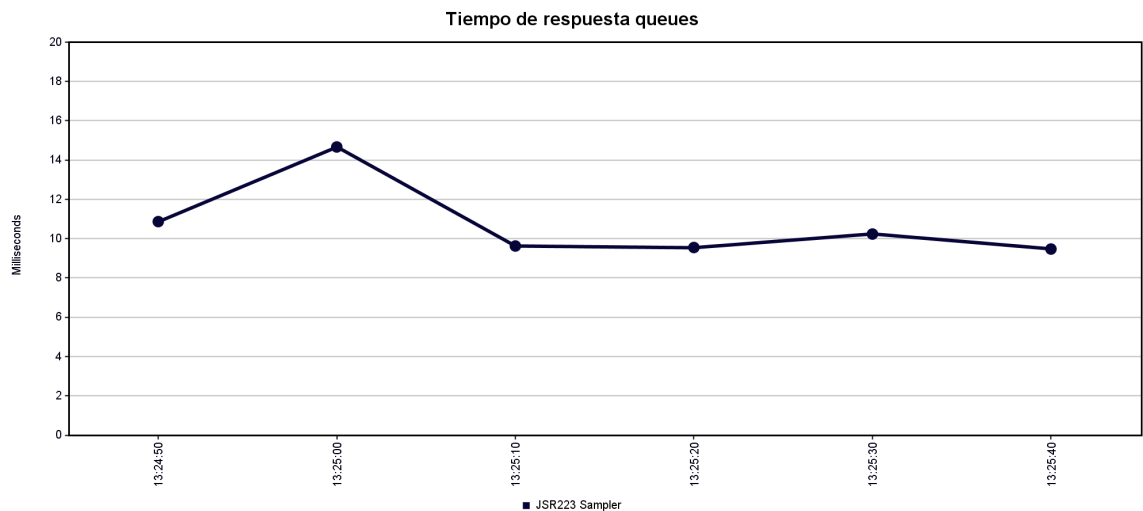


Fig. 7. Tiempo de respuesta queues

References

- [1] N. Ford, M. Richards, Pramod Sadalage, and Z. Dehghani, Software Architecture: The Hard Parts. “O’Reilly Media, Inc.,” 2021.
- [2] “REST - MDN Web Docs Glossary: Definitions of Web-related terms | MDN,” developer.mozilla.org. <https://developer.mozilla.org/en-US/docs/Glossary/REST>

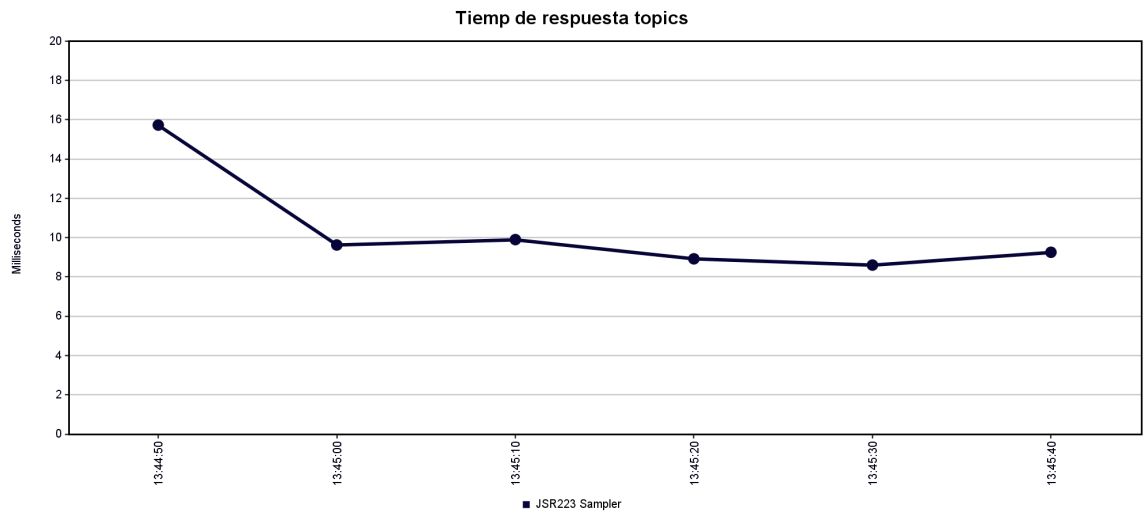


Fig. 8. Tiempo de respuesta topics