Programación Reactiva

Spring Boot

Joe | Julio 2021

Porqué Programación Reactiva?

La programación imperativa tiene algunas limitaciones cuando se trata de responder a las demandas de hoy en día, donde las aplicaciones necesitan ser mas escalables y necesitan darnos tiempos de respuesta bajos durante alta demanda o carga.

Modelo: Thread por Request

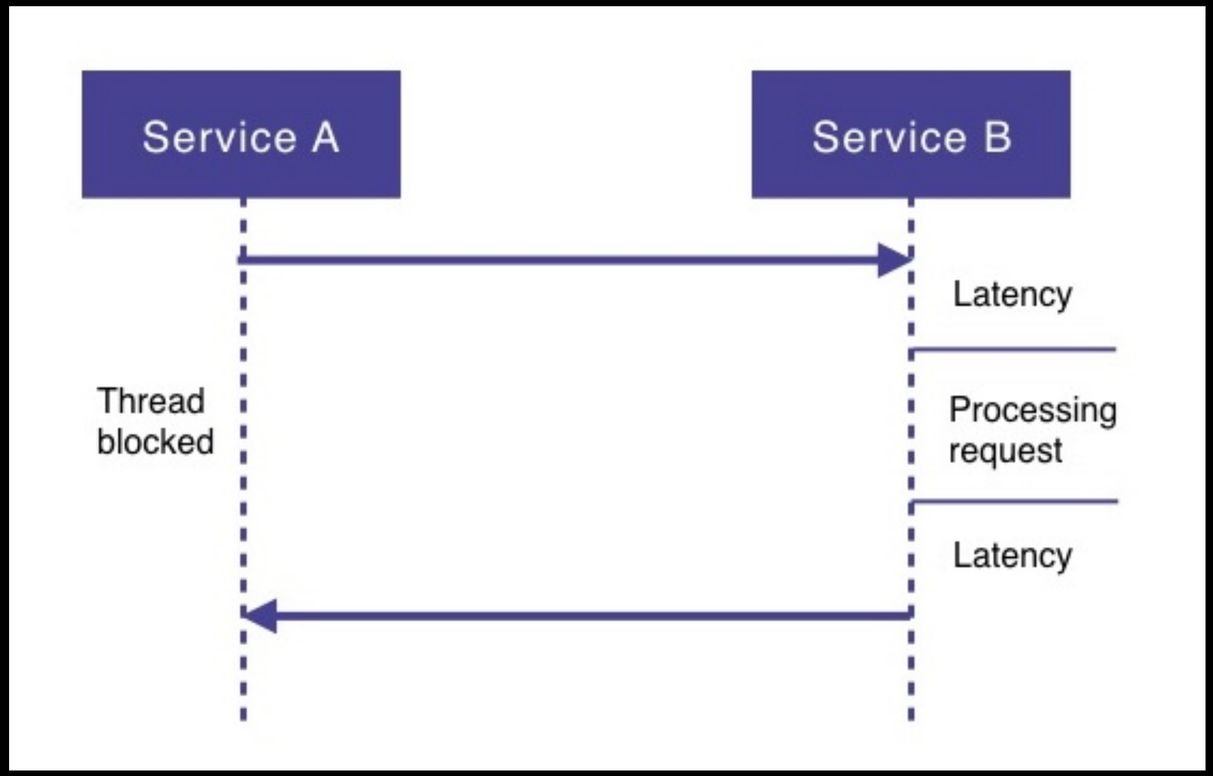
La aplicación solo puede atender un número de requests concurrentes = tamaño del pool de hilos

Cada Thread usa 1MB de memoria, si quieres incrementar el tamaño del pool, necesitarás más memoria

Sí usas arquitectura de micro servicios, podrás escalar basado en la carga, pero, una utilización alta de memoria, incurrirá en un alto costo.

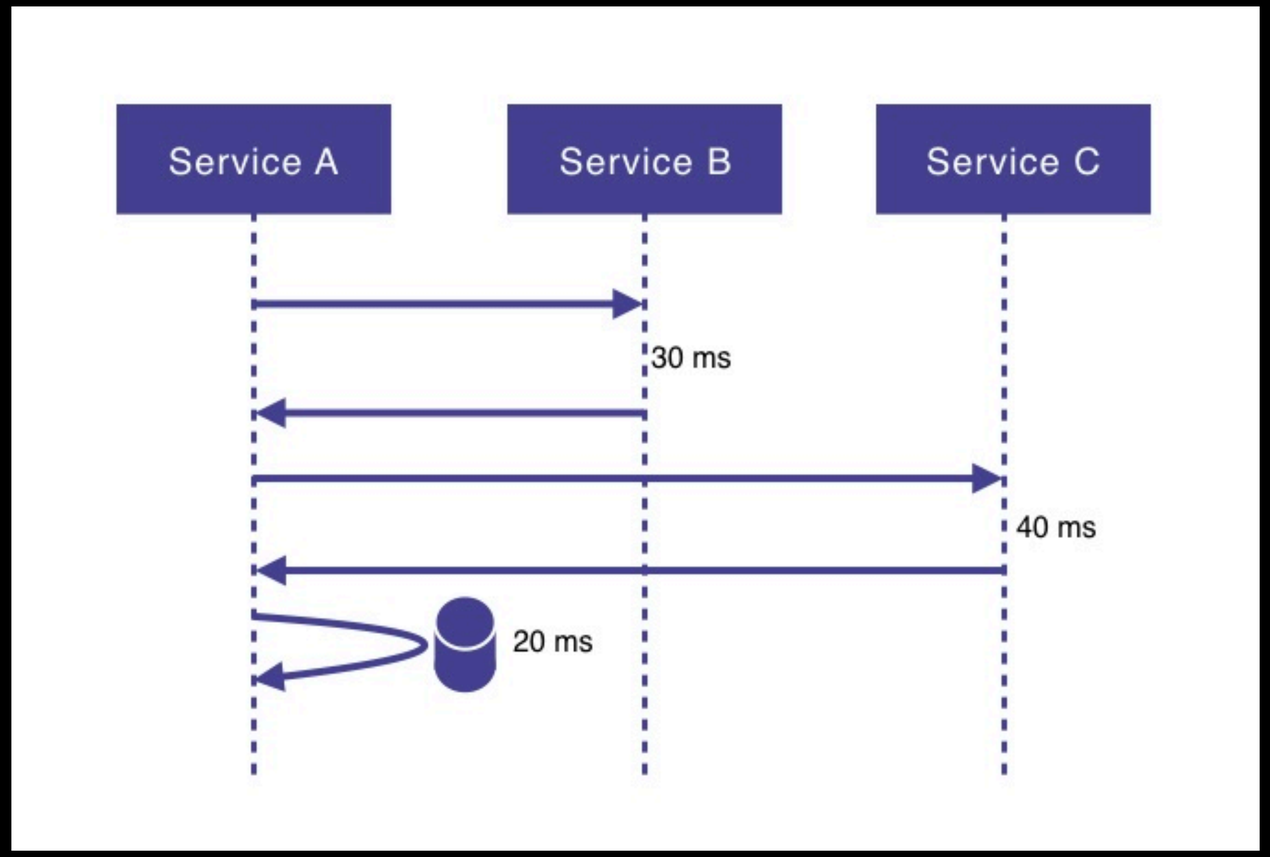
Una programación imperativa basada en requests/response para la comunicación entre servicios implicará que los threads se bloquearán esperando la respuesta de otro servicio. Lo cual es un desperdicio de recursos.

Esperando por operaciones de I/O



Ejemplos: Llamada a una base de datos o leyendo de un archivo.

Tiempos de Respuesta



Tiempo de Respuesta = A->B + A->C + Request DB

Abrumar al Cliente

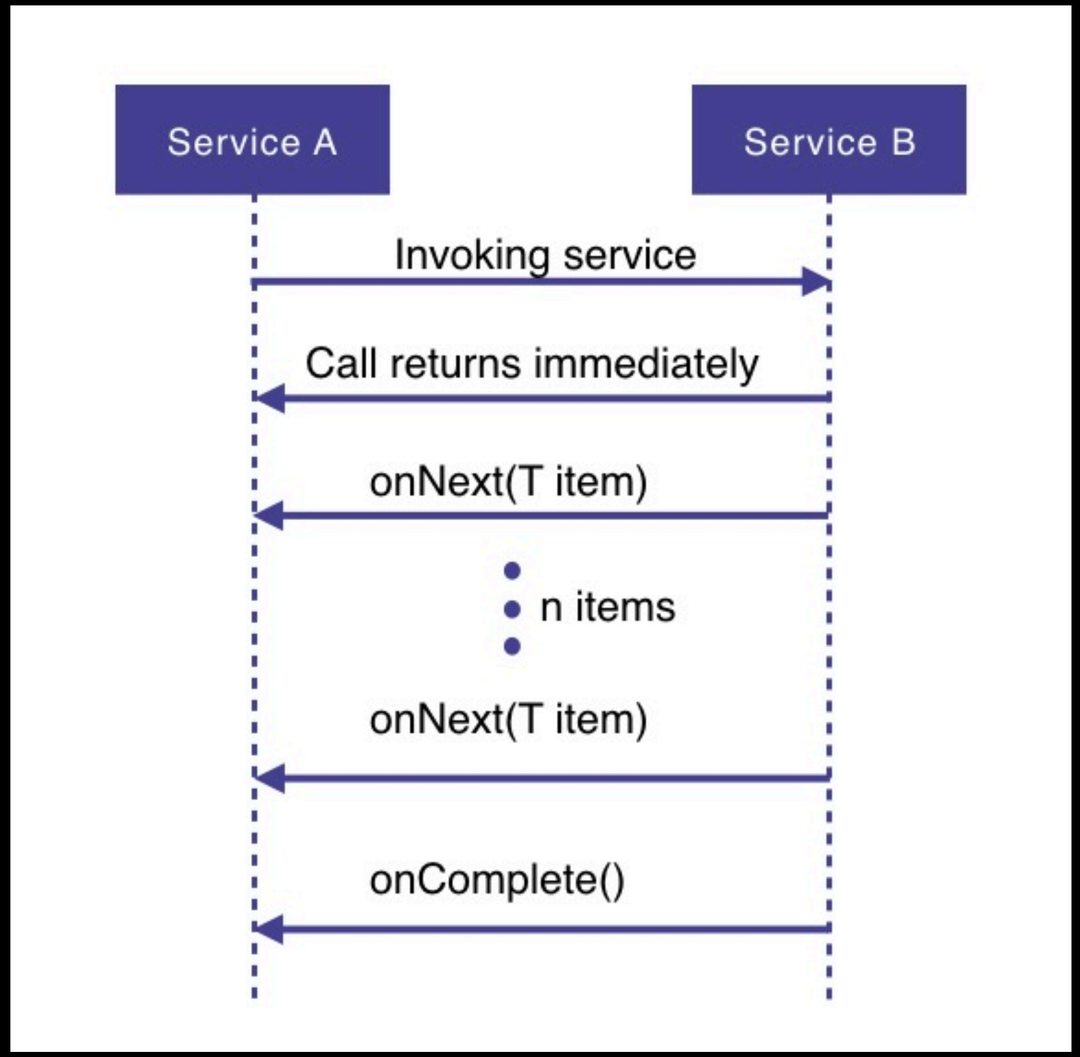
Qué pasa si la información que requiere un cliente A se la proporciona un servicio B, pero, esta es demasiado grande. Ej. Las ventas del mes pasado.

El servicio A podría ser abrumado con la cantidad de datos como resultado de la solicitud y caer en un error de out of Memory.

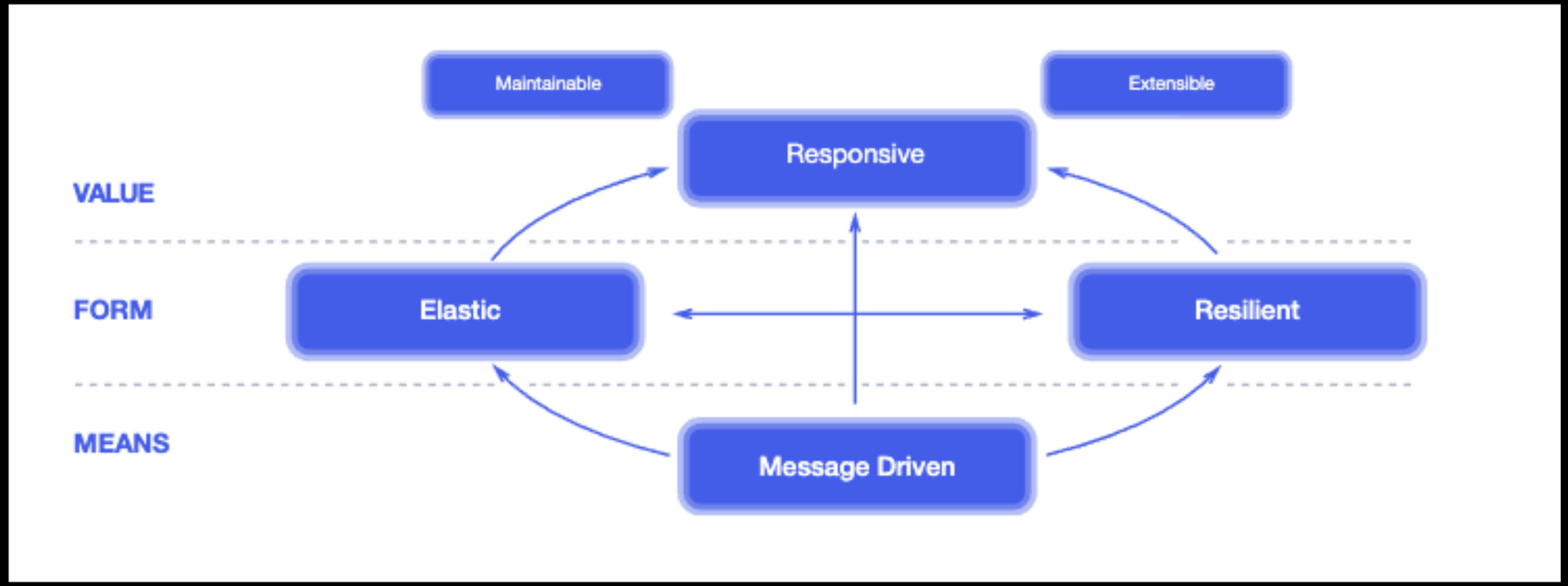
Qué es Programación Reactiva?

En términos sencillos, la programación reactiva se trata de aplicaciones sin bloqueo qué son asincrónicas y controladas por eventos y requieren una pequeña cantidad de threads para escalar. Un aspecto clave de esa definición es el concepto de backpressure, que es un mecanismo para garantizar que los productores no abrumen a sus consumidores.

Explicación



Sistemas Reactivos



Background

Microsoft libero en el 2011, Reactive Extensiones (ReactiveX o Rx) para .NET

RxJava fue desarrollado por Netflix y la versión 1.0 fue liberada en el 2014

ReactiveX usa un mix del patrón Iterator y Observer. Es un modelo baso en push en lugar del comportamiento normal de los iterados que esta basado en pull.

Ademas de observar los cambios, las señales de error y finalización son notificados al subscriptor

Especificación Reactive Streams

Reactive Streams fue adoptado en Java 9, por el Flow API.

```java
public interface Publisher<T> {
    public void subscribe(Subscriber<? super T> s);
}
```

```java
public interface Subscriber<T> {
    public void onSubscribe(Subscription s);
    public void onNext(T t);
    public void onError(Throwable t);
    public void onComplete();
}
```

Especificación Reactive Streams

```
public interface Subscription {
    public void request(long n);
    public void cancel();
}
```

```
public interface Processor<T, R> extends Subscriber<T>, Publisher<R> {
}
```

Project Reactor

Spring Framework soporta programación reactiva desde la versión 5. Este soporte es construido en base al Project Reactor

Project Reactor es una librería Reactiva para construir aplicaciones no bloqueantes en la JVM y esta basado en la especificación Reactive Streams.

Project Reactor es la base del stack reactivo del ecosistema de Spring y esta siendo desarrollado en colaboración con Spring. WebFlux es el framework web reactivo, que requiere a Reactor como su dependencia core.

Reactor Modules

Reactor Test

Reactor Extra

Reactor Netty

Reactor Adapter

Reactor Kafka

Demo

WebFlux

WebFlux

El framework web original de Spring - Spring Web MVC - fue construido para el API Servlet y contenedores Servlet

WebFlux fue introducido como parte de Spring Framework 5.0. A diferencia de Spring MVC, este no requiere el API Servlet. El completamente asíncrono y no bloquean, implementa la especificación Reactive Streams por medio del proyecto Reactor

WebFlux requiere Reactor como dependencia core, pero, es también interoperable con otras librerías como Reactive Streams

JOEDAYZ academy

CURSOS ▾ SERVICIOS ▾ VIDEO CURSOS ▾ JETBRAINS PAYARA BLOG YA TENGO UNA CUENTA EMPIEZA

CURSOS

Duración del curso
Del 20 de Julio al 14 de Septiembre 2021

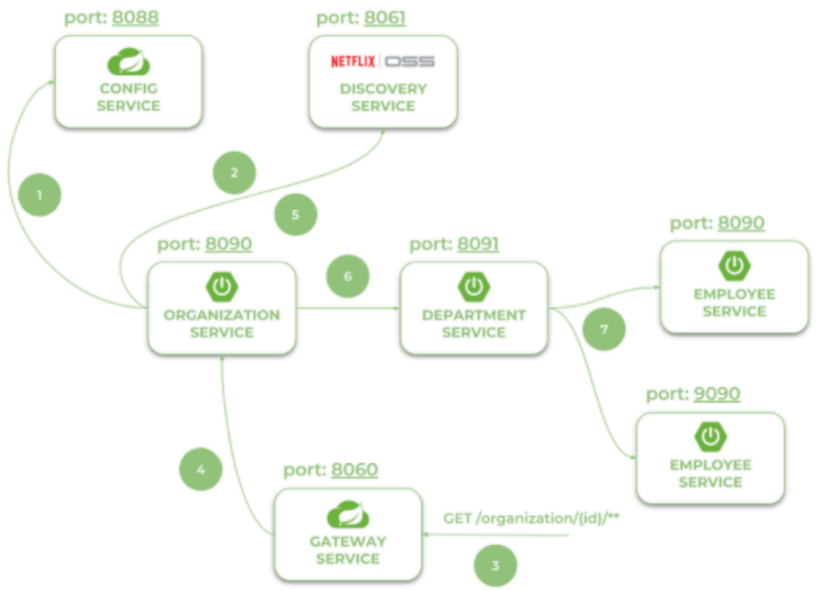
Horarios
Martes y Jueves de 9:30 PM a 11:00 PM

Costo (S/.)
1,200.00

Formas de pago
Puedes depositar directamente a estas cuentas:
Banco de Crédito 191-30759925-0-29
BBVA 0011-0339-0200168694
InterBank 200-3116727850
Scotiabank 174-0055213
y enviarnos el voucher al e-mail:
informes@joedayz.pe

Pagar con paypal (USD $)

Spring Boot y Spring Cloud



966025115 informes@joedayz.pe