

Memoria P2 - Apache Thrift

Salvador Romero Cortés

Apache THRIFT

En esta segunda parte usaremos python con Apache Thrift para hacer un sistema similar. Usaremos la flexibilidad y comodidad de python para introducir más operaciones con más facilidad. Primero, definimos las operaciones

```
service Calculadora{
    void ping(),
    double suma(1:double num1, 2:double num2),
    double resta(1:double num1, 2:double num2),
    double multiplicacion (1:double num1, 2:double num2),
    i32 divisionEntera(1:i32 num1, 2:i32 num2),
    double division(1:double num1, 2:double num2),
    i32 modulo(1:i32 num1, 2:i32 num2),
    list<double> sumaVectores(1:list<double> v1, 2:list<double> v2),
    list<double> restaVectores(1:list<double> v1, 2:list<double> v2),
    list<double> productoVectorial(1:list<double> v1, 2:list<double> v2),
    list<double> productoEscalar(1:list<double> v1, 2:double num2),
    double determinante(1:list<double> matriz)
}
```

Generamos los ficheros con

```
thrift -gen py calculadora.thrift
```

Incorporamos los ficheros cliente y servidor a la carpeta gen-py.

Funcionamiento

Primero lanzamos el servidor con

```
python3 gen-py/servidor.py
```

La sintaxis del cliente es la misma que la del programa RPC, solo que no se indica el servidor.

```
python3 gen-py/cliente.py num1 operador num2
```

O para el modo avanzado:

```
python3 gen-py/cliente.py -a
```

Ejemplo de modo básico:

<pre>> python3 gen-py/cliente.py 12 x 2.3 hacemos ping al server Resultado 12.0 x 2.3 = 27.599999999999998</pre>	<pre>> python3 gen-py/servidor.py iniciando servidor... me han hecho ping() multiplicando 12.0 con 2.3 █</pre>
---	---

Ejemplo del modo avanzado:

```
> python3 gen-py/cliente.py -a
hacemos ping al server
En este modo se utilizan operaciones con vectores y matrices
Operaciones disponibles:

1. Suma de vectores
2. Resta de vectores
3. Producto escalar
4. Producto vectorial
5. Determinante de una matriz

Introduce un número [1-5] para seleccionar la operación
4
Introduciendo vectores de dimension 3
Introduce los valores del primer vector (separados por un salto de línea):

3.4
12.3
1.4
Introduce los valores del segundo vector (separados por un salto de línea):

5.6
0.3
3.2
El resultado de la operación es [38.940000000000005, -3.040000000000002, -67.86]
```

En el caso del determinante de la matriz se podrá elegir la dimensión de la matriz (hasta 3) y se calculará.

```
salva@uwuntu ~/I/O/P/practica-2-2-codigo (main)> python3 gen-py/cliente.py -a
hacemos ping al server
En este modo se utilizan operaciones con vectores y matrices
Operaciones disponibles:

1. Suma de vectores
2. Resta de vectores
3. Producto escalar
4. Producto vectorial
5. Determinante de una matriz

Introduce un número [1-5] para seleccionar la operación
5
Introduce la dimensión de la matriz (1, 2 o 3): 3
Introduce los valores de la matriz en orden por filas (separados por un salto de línea
):
1
2
3
4
5
6
7
8
9
10
El resultado de la operación es -3.0
```

Probando con otros lenguajes

Es fácil tratar con varios lenguajes con Apache Thrift gracias a que con el mismo fichero .thrift podemos generar la estructura necesaria en varios lenguajes. Es interesante notar que incluso se pueden mezclar los lenguajes usados en el cliente y en el servidor. Por ejemplo, aquí tenemos un cliente escrito en ruby y un servidor escrito en python

```
> ruby gen-rb/cliente.rb
ping()
1+1=2
1+4=5
>

> python3 gen-py/servidor.py
iniciando servidor...
me han hecho ping()
sumando 1 con 1
sumando 1 con 4
```

He hecho un cliente básico (suma, resta, producto, división) en ruby para usar con el servidor en python. Tiene la misma sintaxis que el cliente de python. Primero generamos con apache thrift los archivos necesarios para ruby con

```
thrift -gen rb calculadora.thrift
```

Luego, incorporamos el cliente.rb en la carpeta gen-rb.

Ejecutamos el servidor en python como antes y el cliente en ruby como:

```
ruby gen-rb/cliente.rb <num1> <operacion> <num2>
```

```
salva@ubuntu ~/I/O/P/practica-2-2-codigo (main)> ruby gen-rb/cliente.rb 12 + 2.2
ping()
Resultado de la operación
12.0 + 2.2 = 14.2
salva@ubuntu ~/I/O/P/practica-2-2-codigo (main)> ruby gen-rb/cliente.rb 12 - 2.2
ping()
Resultado de la operación
12.0 - 2.2 = 9.8
salva@ubuntu ~/I/O/P/practica-2-2-codigo (main)> ruby gen-rb/cliente.rb 12 x 2.2
ping()
Resultado de la operación
12.0 x 2.2 = 26.400000000000002
salva@ubuntu ~/I/O/P/practica-2-2-codigo (main)> ruby gen-rb/cliente.rb 12 / 2.2
ping()
Resultado de la operación
12.0 / 2.2 = 5.454545454545454
salva@ubuntu ~/I/O/P/practica-2-2-codigo (main)> █
```

```
salva@ubuntu ~/I/O/P/practica-2-2-codigo (main)> python3 gen-py/servidor.py
iniciando servidor...
me han hecho ping()
sumando 12.0 con 2.2
me han hecho ping()
restando 12.0 con 2.2
me han hecho ping()
multiplicando 12.0 con 2.2
me han hecho ping()
división (con decimales) entre 12.0 y 2.2
█
```