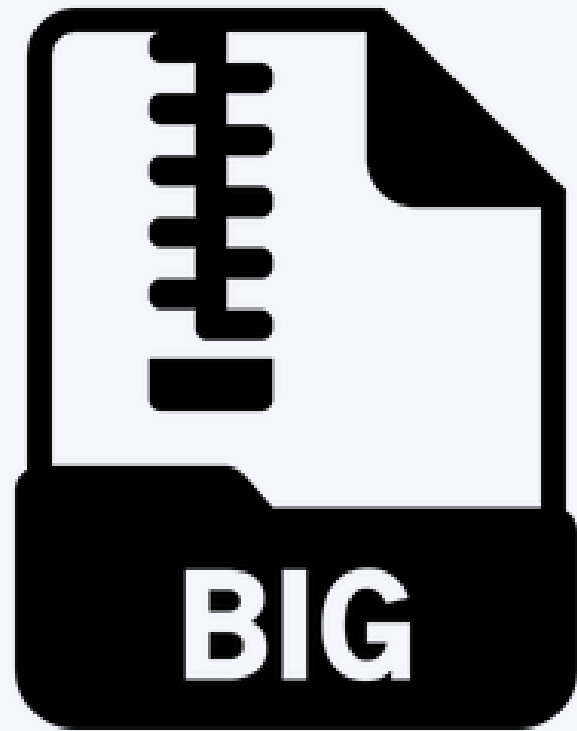


# Boto3 `put_object()` Considered Harmful

AWS Boto3 de Manera Práctica



# Probemos un Caso 25 Veces



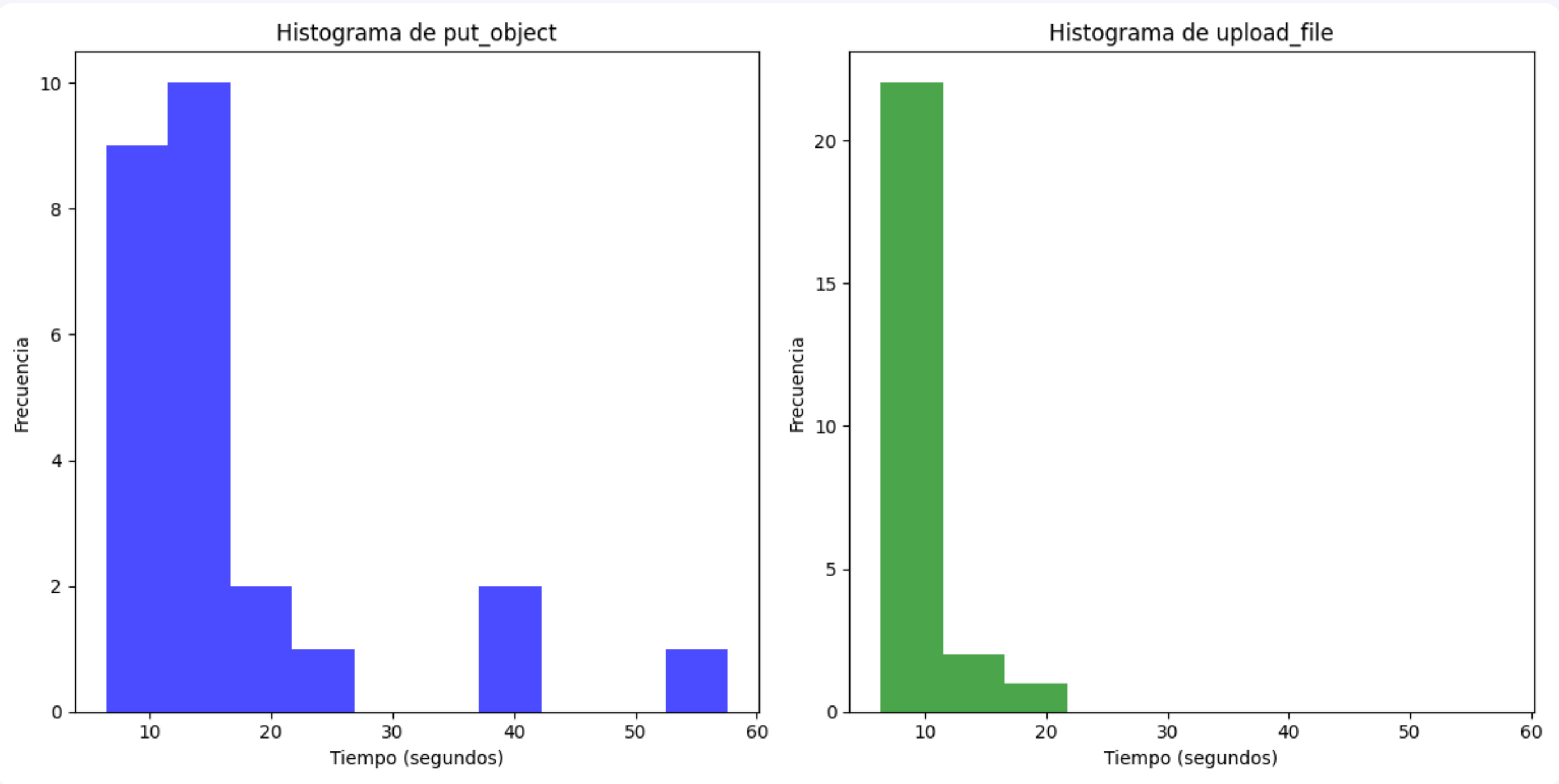
*100mb*



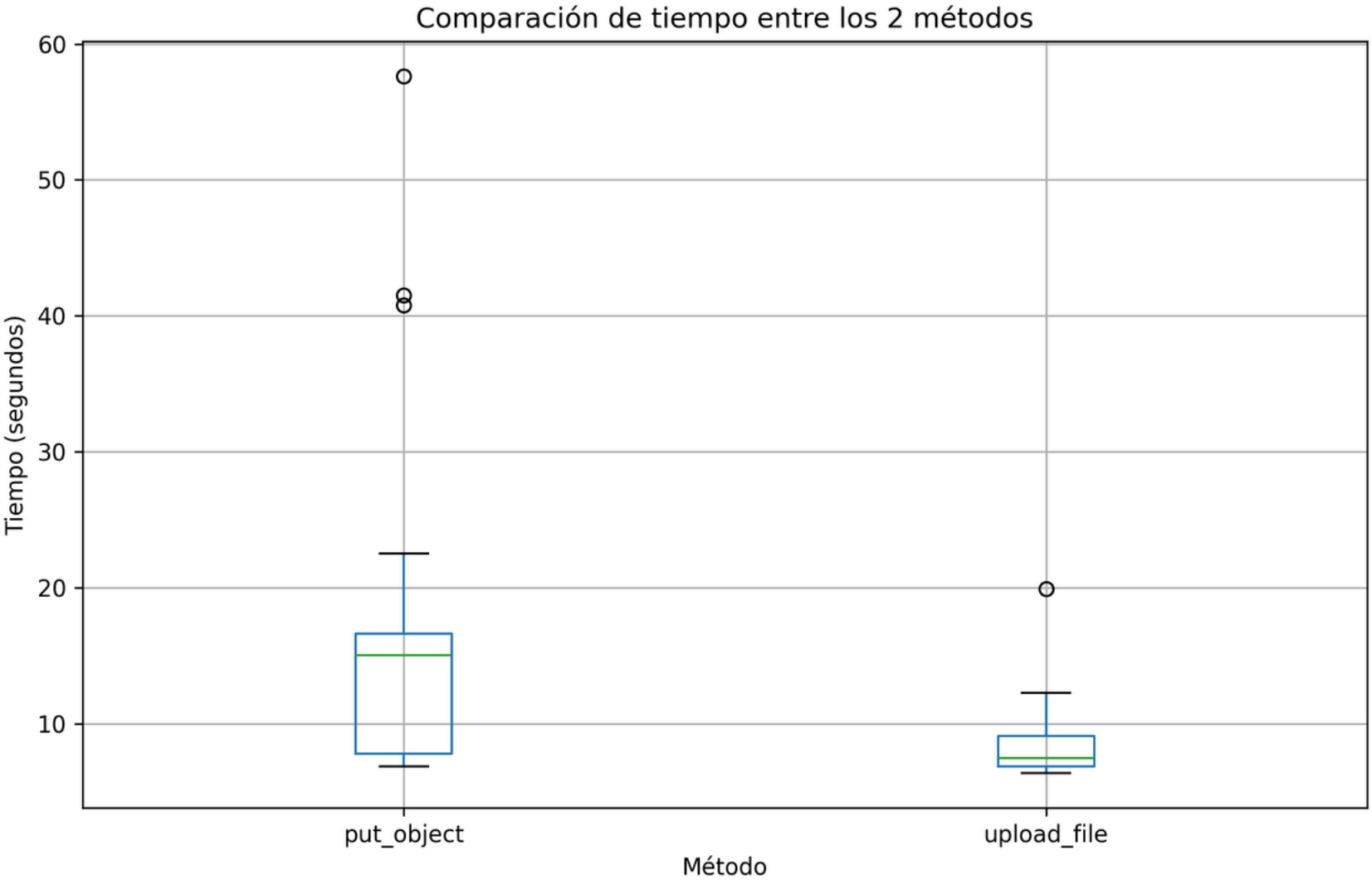
`upload_file()` vs `put_object()`

**¿Cuál tomará más tiempo?**

# A Simple Inspección



# upload\_file corre en x2!

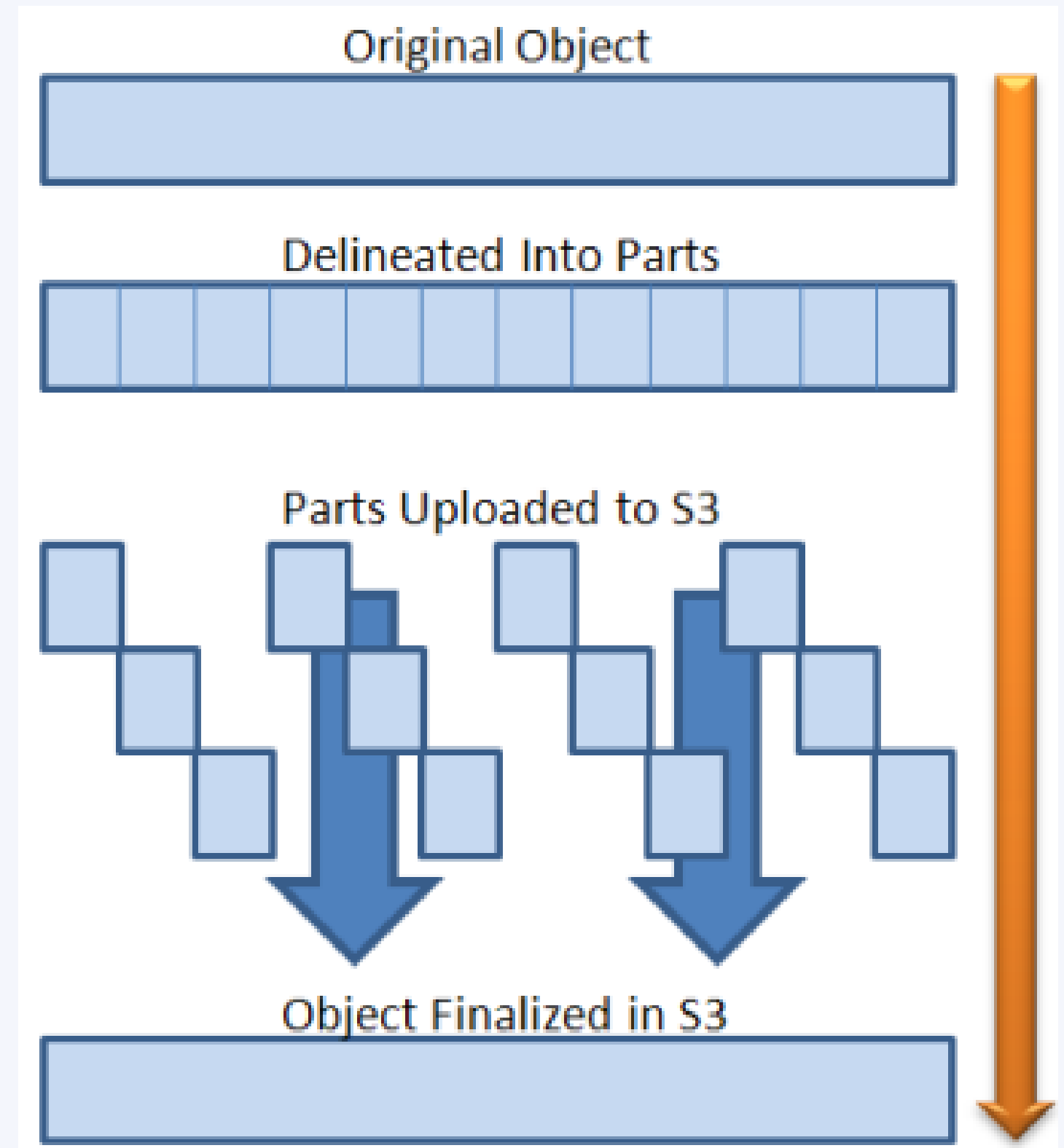


	put_object	upload_file
count	25.0000000	25.0000000
mean	17.050990	8.429058
std	12.310732	2.889491
min	6.843389	6.360362
25%	7.782293	6.858396
50%	15.010685	7.444041
75%	16.595273	9.077799
max	57.628193	19.913620

# ¿Cuál es la razón?

La respuesta viene de una optimización que realiza el SDK con el método `upload_object()`: **multipart upload**.

- AWSCLI: **mpu** por defecto
- AWS CONSOLE: **no** hay soporte **para mpu**
- AWS BOTO3: **put\_object()** no implementa mpu



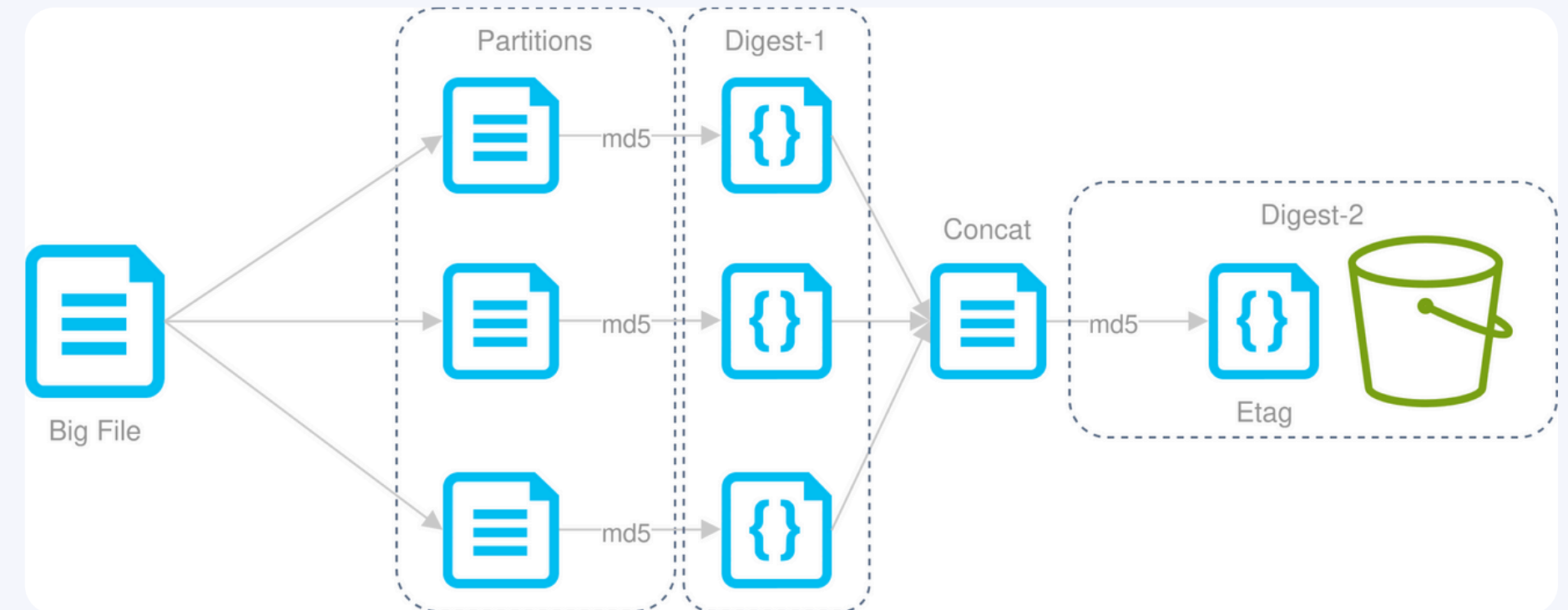
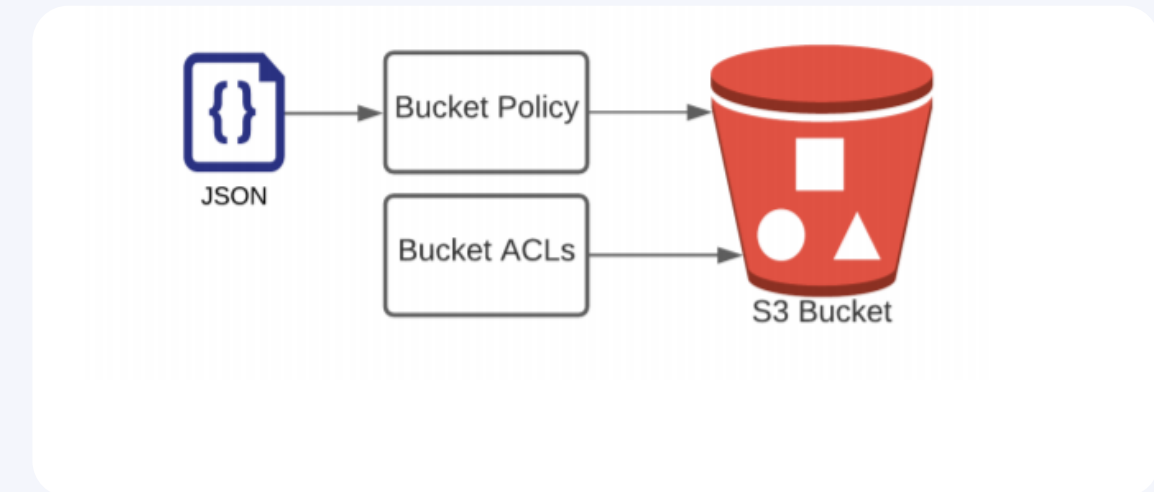
Amazon S3 News Blog

# ¿Cuándo usar put\_object?

`put_object()`, a diferencia de `upload_file()`, mapea a una Api de bajo nivel en botocore.

Esto implica mayor capacidad de personalización, en específico para las **access control list (acl)** del objeto en s3.

La verificación de **integridad con md5** es también más sencilla cuando la carga es directa.



El proceso del cálculo de la Etag en **mpu** tiene dos etapas



**Fin**

