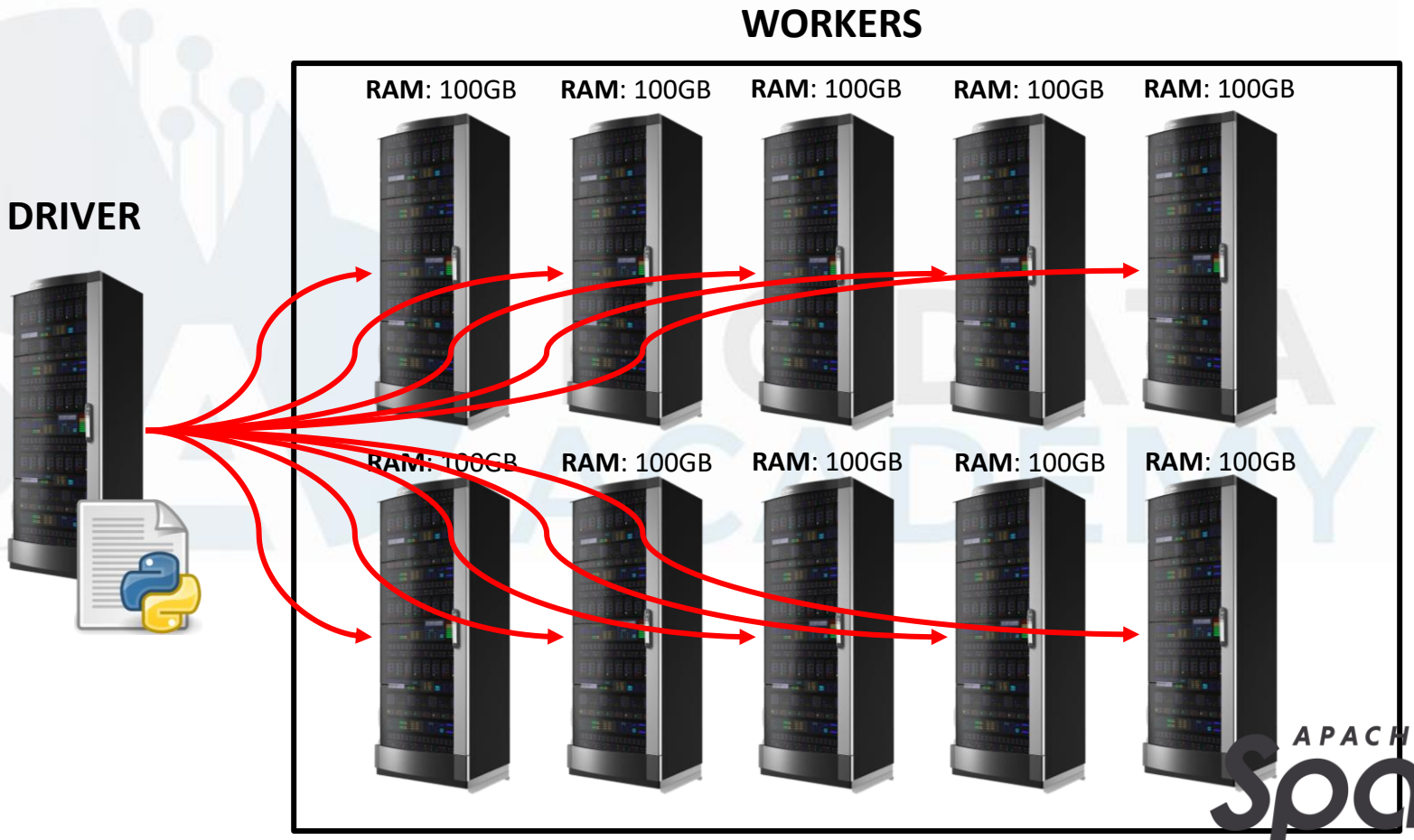
Patrón de Diseño REPARTION

Drivers y workers

En Spark existen 2 tipos de servidores:

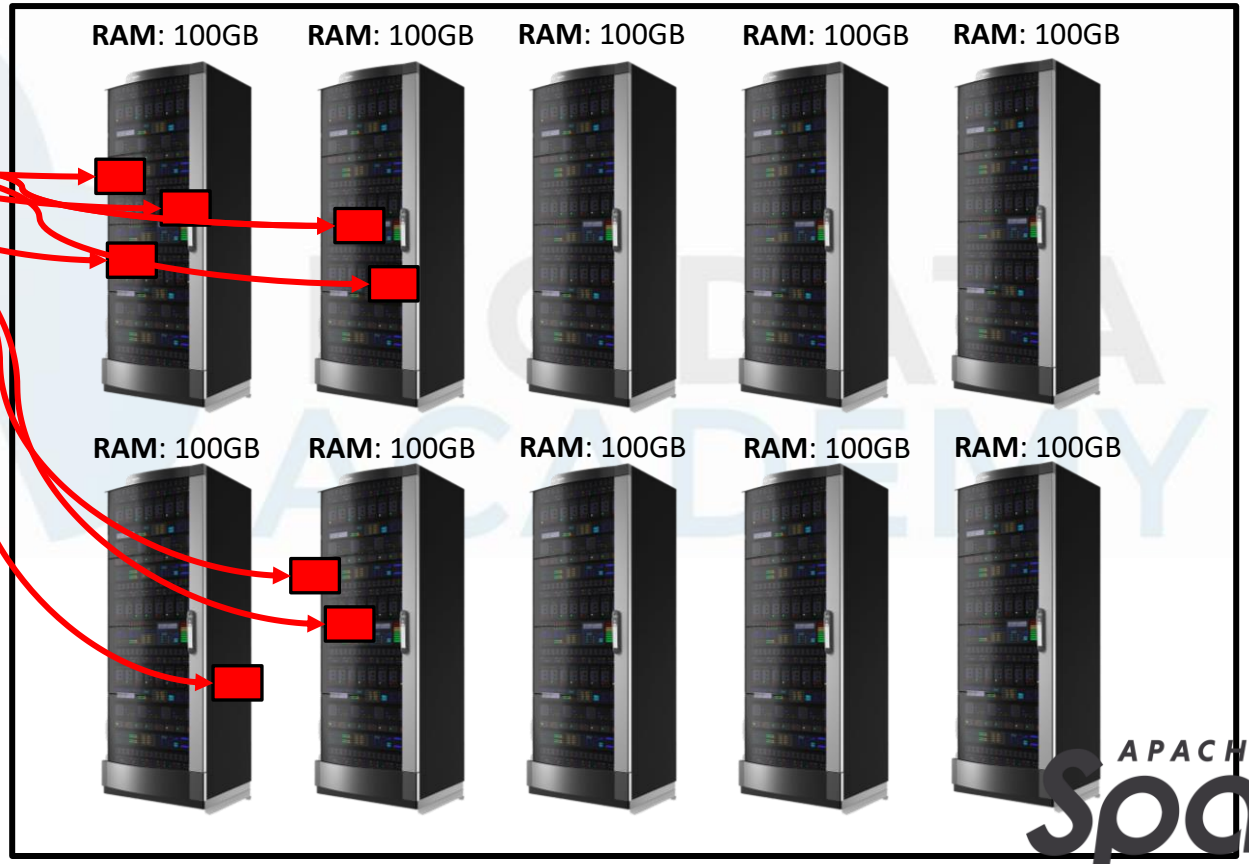
DRIVER: Servidor que envía las instrucciones de código
WORKERS: Servidores que procesan la carga de trabajo



Lectura de un archivo y distribución en workers

ARCHIVO

DF

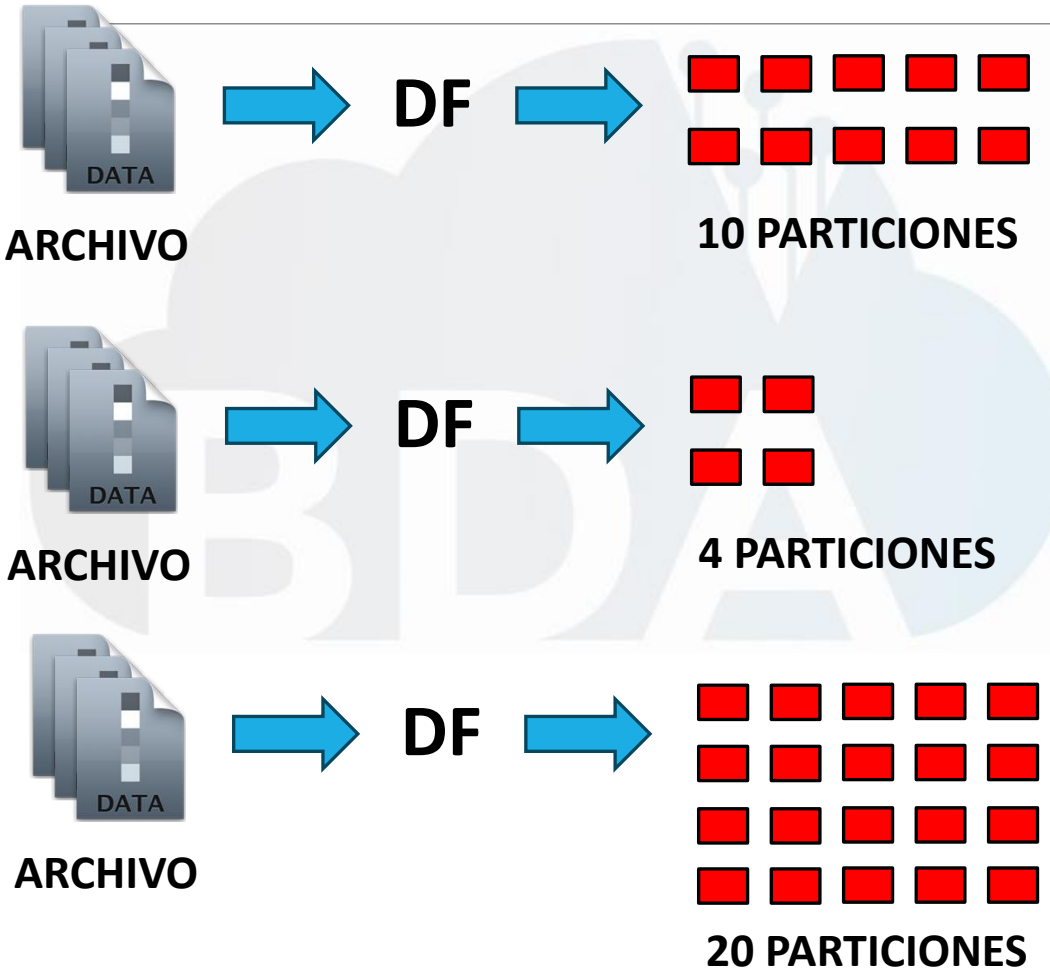


Cuando un archivo es cargado a un dataframe, **el dataframe crea particiones y las distribuye entre los workers.**

Por ejemplo, si un archivo tiene 1 millón de registros, Spark podría crear 10 particiones de 100 mil registros cada una.

“Aleatoriedad” del número de particiones

Cuando se lee un archivo, no necesariamente se crea el mismo número de particiones, por ejemplo, **al leer un archivo de 1 millón de registros podría suceder:**



DÍA 1
Se crean 10 particiones de 100 mil registros cada uno

DÍA 2
Se crean 4 particiones de 250 mil registros cada uno

DÍA 3
Se crean 20 particiones de 50 mil registros cada uno

¿Existe un número de particiones ideal?

Registros por particiones ideal

En Spark cada partición debe tener en promedio 100 mil registros. Si no los tiene, deberemos reparticionar. Por ejemplo, si un archivo tiene 1 millón de registros, deberemos crear 10 particiones

Cuando reparticionamos hay dos posibilidades:

REPARTICIONAMIENTO HACÍA ARRIBA

En este caso deberemos pasar de 4 particiones a 10 particiones **(aumentar particiones)**

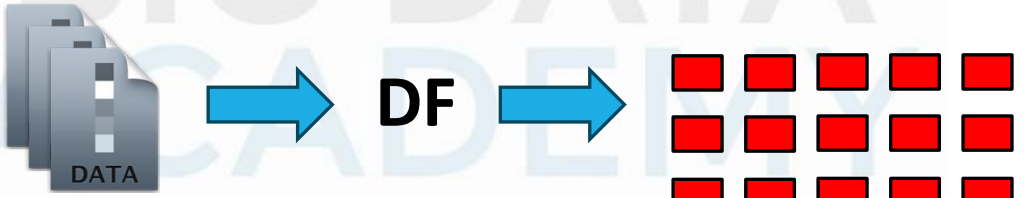
ARCHIVO [1 MILLÓN DE REGISTROS]

4 PARTICIONES

Función optimizada: repartition

REPARTICIONAMIENTO HACÍA ABAJO

En este caso deberemos pasar de 20 particiones a 10 particiones **(disminuir particiones)**



ARCHIVO [1 MILLÓN DE REGISTROS]

20 PARTICIONES

Función optimizada: coalesce

¿Cuándo reparticionar?

Siempre que leamos un archivo desde disco duro, hay que reparticionar el dataframe

ARCHIVO [1 MILLÓN DE REGISTROS]

4 PARTICIONES

10 PARTICIONES