Parameters choose to benchmark

➢ Fio

--name=str	Fio will create a file with the specified name to run the test on it
--ioengine=str	This argument defines how the job issues I/O to the test file ***libaio*** - Linux native asynchronous block level I/O
--size=int	The size of the file on which the Fio will run the benchmarking test. Taille du fichier sur lequel le Fio exécutera le test d'évaluation des performances.
--rw=str	Specifies the type of I/O pattern. The most common ones are as follows: • *read:* sequential reads • *write:* sequential writes • *randread:* random reads • *randwrite:* random writes • *rw:* sequential mix of reads and writes • *randrw:* random mix of reads and writes Fio defaults to **50/50** if mixed workload is specified (rw=randrw). If more specific read/write distribution is needed, it can be configured with **--rwmixread=**. For example, --rwmixread=30 would mean that **30%** of the I/O will be reads and **70%** will be writes
--bs=int	Defines the block size that the test will be using for generating the I/O.
--direct=bool	If the value is set to **1** (using non-buffered I/O) is fairer for testing as the benchmark will send the I/O directly to the storage subsystem bypassing the OS filesystem cache.
--numjobs=int	The number of threads spawned by the test. By default, each thread is reported separately. To see the results for all threads as a whole, use --group_reporting.
--iodepth=int	Number of I/O units to keep in flight against the file. That is the amount of outstanding I/O for each thread.
--runtime=int	The amount of time the test will be running in seconds.
--time_based	If given, run for the specified runtime duration even if the files are completely read or written.
--ramp_time	If set, fio will run the specified workload for this amount of time before logging any performance numbers. Useful for letting performance settle before logging results. Cette option est utile pour laisser les performances se stabiliser avant de consigner les résultats

--verify=int	Dump info related to I/O verification.