Usage: memtier_benchmark [options]

A memcache/redis NoSQL traffic generator and performance benchmarking tool.

Connection and General Options:

```
-s, --server=ADDR               Server address (default: localhost)
-p, --port=PORT                 Server port (default: 6379)
-S, --unix-socket=SOCKET        UNIX Domain socket name (default: none)
-P, --protocol=PROTOCOL          Protocol to use (default: redis).   Other
                                 supported protocols are memcache_text,
                                 memcache_binary.
-a, --authenticate=CREDENTIALS Authenticate using specified credentials.
                                 A simple password is used for memcache_text
                                 and Redis <= 5.x. <USER>:<PASSWORD> can be
                                 specified for memcache_binary or Redis 6.x
                                 or newer with ACL user support.
    --tls                       Enable SSL/TLS transport security
    --cert=FILE                  Use specified client certificate for TLS
    --key=FILE                   Use specified private key for TLS
    --cacert=FILE                Use specified CA certs bundle for TLS
    --tls-skip-verify           Skip verification of server certificate
    --sni=STRING                  Add an SNI header
-x, --run-count=NUMBER           Number of full-test iterations to perform
-D, --debug                      Print debug output
    --client-stats=FILE         Produce per-client stats file
    --out-file=FILE              Name of output file (default: stdout)
    --json-out-file=FILE         Name of JSON output file, if not set, will not print to json
    --hdr-file-prefix=FILE      Prefix of HDR Latency Histogram output files, if not set, will not save latency histogram files
    --show-config                Print detailed configuration before running
    --hide-histogram             Don't print detailed latency histogram
```

--print-percentiles   Specify which percentiles info to print on the results table (by default prints percentiles:
50,99,99.9)
        --cluster-mode   Run client in cluster mode
        --help   Display this help
        --version   Display version information

Test Options:
  -n, --requests=NUMBER   Number of total requests per client (default: 10000)
                                    use 'allkeys' to run on the entire key-range
  -c, --clients=NUMBER   Number of clients per thread (default: 50)
  -t, --threads=NUMBER   Number of threads (default: 4)
        --test-time=SECS   Number of seconds to run the test
        --ratio=RATIO   Set:Get ratio (default: 1:10)
        --pipeline=NUMBER   Number of concurrent pipelined requests (default: 1)
        --reconnect-interval=NUM   Number of requests after which re-connection is performed
        --multi-key-get=NUM   Enable multi-key get commands, up to NUM keys (default: 0)
        --select-db=DB   DB number to select, when testing a redis server
        --distinct-client-seed   Use a different random seed for each client
        --randomize   random seed based on timestamp (default is constant value)

Arbitrary command:
        --command=COMMAND   Specify a command to send in quotes.
                                    Each command that you specify is run with its ratio and key-pattern options.
                                    For example: --command="set __key__ 5" --command-ratio=2
--command-key-pattern=G

                                    To use a generated key or object, enter:
                                       __key__: Use key generated from Key Options.
                                       __data__: Use data generated from Object Options.
        --command-ratio   The number of times the command is sent in sequence.(default: 1)
        --command-key-pattern   Key pattern for the command (default: R):
                                   G for Gaussian distribution.
                                   R for uniform Random.

S for Sequential.
P for Parallel (Sequential were each client has a subset of the key-range).

Object Options:
-d   --data-size=SIZE   Object data size (default: 32)
     --data-offset=OFFSET   Actual size of value will be data-size + data-offset
                                   Will use SETRANGE / GETRANGE (default: 0)
-R   --random-data   Indicate that data should be randomized
     --data-size-range=RANGE   Use random-sized items in the specified range (min-max)
     --data-size-list=LIST   Use sizes from weight list (size1:weight1,..sizeN:weightN)
     --data-size-pattern=R|S   Use together with data-size-range
                                   when set to R, a random size from the defined data sizes will be used,
                                   when set to S, the defined data sizes will be evenly distributed across
                                   the key range, see --key-maximum (default R)
     --expiry-range=RANGE   Use random expiry values from the specified range

Imported Data Options:
     --data-import=FILE   Read object data from file
     --data-verify   Enable data verification when test is complete
     --verify-only   Only perform --data-verify, without any other test
     --generate-keys   Generate keys for imported objects
     --no-expiry   Ignore expiry information in imported data

Key Options:
     --key-prefix=PREFIX   Prefix for keys (default: "memtier-")
     --key-minimum=NUMBER   Key ID minimum value (default: 0)
     --key-maximum=NUMBER   Key ID maximum value (default: 10000000)
     -PATTERN   Set:Get pattern (default: R:R)
                                   G for Gaussian distribution.
                                   R for uniform Random.
                                   S for Sequential.
                                   P for Parallel (Sequential were each client has a subset of the key-range).

```
    --key-stddev                    The standard deviation used in the Gaussian distribution
                                     (default is key range / 6)
    --key-median                     The median point used in the Gaussian distribution
                                     (default is the center of the key range)

WAIT Options:
    --wait-ratio=RATIO              Set:Wait ratio (default is no WAIT commands - 1:0)
    --num-slaves=RANGE                WAIT for a random number of slaves in the specified range
    --wait-timeout=RANGE             WAIT for a random number of milliseconds in the specified range (normal
                                     distribution with the center in the middle of the range)
```