

Variant Storage Engine Benchmark

OpenCGA benchmark is a rich benchmark suite for storage engines supported with OpenCGA, namely *mogodb* and *hbase*. Please find below the list and detailed explanation of different components of OpenCGA Benchmark and how they work together to create benchmark.

Execution Mode

Benchmark supports the following execution mode :

- Fixed
- Random

Fixed Mode

Its a fixed set of queries written in a YML file, benchmark will take each query (default) or a selection of queries passed as IDs arguments in `--query, -q` option and execute these as a certain number of users (`-c, --concurrency`) for a specific number of time (`-r, --num-repetition`). Parameters listed under `baseQuery` section will be applied to each individual query and can be overwritten from main query or using command line option (`-B, --baseQuery`) . A sample of fixed query file is displayed below:

FixedQueries.yml

```
---
baseQuery :
  summary : true

queries :
- id : "RegionAndBiotype"
  description : "Purpose of this query"
  query :
    region : "22:16052853-16054112"
    gene : "BRCA2"
    biotype : "coding"
    populationFrequencyMaf : "1kG_phase3:ALL>0.1"
    tolerationThreshold : 300

- id : "Region"
  description : "Purpose of this query"
  query :
    region : "22:16052853-16054112"
    tolerationThreshold : 400
.....
sessionIds :
- ""
- ""
```

Following command will execute ALL queries written in `fixedQueries.yml` file as 10 users, five times each on REST server specified in `"storage-configuration.yml"` :

Benchmark Query

```
opencga-storage-admin.sh benchmark variant --concurrency 10 --num-
repetition 5 --mode FIXED --connector REST
```

Complete list of options, default values and explanations can be displayed using `--help` option from benchmark script :

Table of Contents:

- [Execution Mode](#)
 - [Fixed Mode](#)
 - [Random Mode](#)
- [Connection Type](#)

```

root@localhost:~/opencga/opencga-storage/build/bin# ./opencga-storage-admin.sh benchmark variant --help
usage: opencga-storage-server.sh benchmark variant [options]

Options:
-B, --baseQuery STRING=STRING Overwrite baseQuery options from file, comma separated, ie. -Blimit=1000
-C, --concurrency INT Number of concurrent threads
-C, --conf STRING Configuration file path, ie. system:REST or DIRECT [REST]
--connector CONNECTION_TYPE How to connect to the system: REST or DIRECT [REST]
--count COUNT Count results. Do not return elements. [false]
-d, --database STRING Database name to load the data
--delay INT Delay between each sampler thread.
-f, --file STRING File path to load queries
-h, --help Print this help [false]
--host STRING Remote host
--limit INT Limit the number of returned elements. [0]
--log-file STRING One of the following: 'error', 'warn', 'info', 'debug', 'trace'
-l, --log-level STRING One of the following: 'error', 'warn', 'info', 'debug', 'trace' [info]
-m, --mode EXECUTION_MODE Type of queries to execute: FIXED, RANDOM [FIXED]
-r, --num-repetition INT Number of repetition to execute.
-o, --outdir STRING Output directory
-q, --query STRING Query IDs to execute for FIXED mode (Default All) OR Query pattern to execute for Random mode e.g.
                                gene(CTD0)(region1)
--storage-engine STRING One of the listed in storage-configuration.yml
-v, --verbose Increase the verbosity of logs [false]
-D STRING=STRING Storage engine specific parameters go here comma separated, ie. -Dmongodb.compressionsnapgy

```

Random Mode

Random mode supports creation of random queries from meta data provided in "*randomQueries.yml*" and execute these on selected storage engine :

randomQueries.yml

```
---
baseQuery :
  summary : true
  exclude : studies

regions :
  - chromosome : "1"
    start : 1
    end : 249250621

gene :
  - DKFZP434A062
  - GPSM1

ct : []

type :
  - "SV"
  - "CNV"

study :
  - "1kG_phase3"
...
functionalScore :
  - id : "cadd_raw"
    min : 0
    max : 1
  - id : "cadd_scaled"
    min : -10
    max : 40

populationFrequencies :
  - id : "1kG_phase3:ALL"
    min : 0
    max : 0.2
  - id : "1kG_phase3:AFR"
    min : 0
    max : 0.15

proteinSubstitution :
  - id : "polyphen"
    min : 0.1
    max : 0.9
    operators : [ ">", "<" ]
  - id : "sift"
    min : 0.1
    max : 0.9

qual :
  id : "polyphen"
  min : 1
  max : 9
  operators : [ ">" ]

conservation :
  id : "phylop"
  min : 0
  max : 1
  operators : [ "=", "!=" ]

sessionIds :
  - ""
  - ""
```

Following command will generate two queries one with two different "ct" values and a gene value and second with a region value provided in "*randomQueries.yml*" file and execute as 10 users, five times each on REST server:

Benchmark Random Query Execution

```
opencga-storage-admin.sh benchmark variant --concurrency 10 --num-repetition 5 --mode RANDOM -q "ct(2),gene;region"
```

Storage Engine

Currently OpenCGA supports the following storage engines :

1. Mongo
2. HBase
3. "*Solr*" (*An optional support component for storage engines*)

This value is read from "*storage-configuration.yml*" "*defaultStorageEngineId*" field or can be passed as argument on command line, *--storage-engine*. OpenCGA also supports "*solr*" to improve performance of certain queries for variant. By passing "*summary=true/false*" in baseQuery user can compare working and performance of OpenCGA with and without solr component.

Connection Type

Connection Type is the connection method of benchmark to the storage engine. OpenCGA currently supports three connection types :

- | | |
|--------------------------------------|--------------------------------------|
| 1. REST | (OpenCGA web server, sessionIds are |
| mandatory to connect using this) | |
| 2. Storage REST | (REST server provided with storage |
| component and must be running state) | |
| 3. Direct | (Fetching data directly from storage |
| engine using java storage adaptors) | |

REST type is most relevant for end users and should be used to create benchmark, remaining two are mainly to get deeper insight into the performance of storage engine without overhead of OpenCGA catalog *authorisation* and *authentication* mechanism.