

# Cloudgen Quick How-To Manual

## Requirements

Cloudgen requires Python version 2 or later with package “numpy”.

## Usage

There are only two mandatory arguments to run Cloudgen: you are required to inform (i) a number of devices or a particular content sharing network you want to generate workload, and (ii) an input file with the parameters of statistical distributions of Cloudgen components. Other arguments are optional.

In this document, we describe all Cloudgen arguments. For most arguments, we will refer to sections of the technical report:

G. Gonçalves, I. Drago, A. B. Vieira, A. P. C. da Silva, J. M. de Almeida, and M. Mellia, “Workload Models and Performance Evaluation of Cloud Storage Services,” RT.DCC.003/2015, UFMG, Tech. Rep., 2015, <http://homepages.dcc.ufmg.br/~ggoncalves/techrpt/15003.pdf>

## Synopsis

```
usage: cloudgen.py [-h] [-i INPUT] [-d NUM_DV] [-n NUM_NS] [-c CONF]
                  [-t TIMEOUT] [-r SEED] [-l OUTLAB]
```

optional arguments:

- h, --help show this help message and exit
- i INPUT Input shared dev topology (see the examples for the format)
- d NUM\_DV Input number of devices - synthetic network is created
- n NUM\_NS Input number of namespaces - synthetic network is created
- c CONF Configuration parameters (see the examples for campuses and PoPs networks)
- t TIMEOUT Duration of the simulation (in days)
- r SEED Random seed (default uniform random number)
- l OUTLAB Outfile label

-c and (-i or -d) are mandatory

## INPUT:

This is the content sharing network input file. Examples of this file are in the network directory of this repository.

This input file has the following format: each line is represented by one device (1st. element) and one namespace (2nd. element) that is linked to this device.

There is not limitation for number of devices (i.e., file lines) and number of namespaces per device. However, both namespaces and devices must be identified by integer numbers.

Since this file format represents a content sharing network (see Section 4.2.2 of the technical report), namespaces can be linked to multiple devices and vice versa.

**NUM\_DV:**

Number of devices to generate workload. In this case a synthetic content sharing network will be created taking into consideration the provided number of devices and statistical distributions of components in the configuration input file (CONF argument)

**NUM\_NS:**

Number of namespaces (see Section 2.1 of technical report for namespace definition). In this case a synthetic content sharing network is created taking into consideration provided number of devices, number of namespaces and statistical distributions of components in the configuration input file (CONF argument). NUM\_NS is an optional argument, and when it is not provided, the relationship between number of namespaces per devices is defined as shown in Section 5.1 of the technical report.

**CONF:**

Input file with the parameters of statistical distributions for Cloudgen components. These statistical distributions were characterized from campuses and Internet Service Providers -- Point of Presence – networks (PoP). Please, refer to Section 4 and Appendix A of the technical report to obtain details about characterization of Cloudgen components.

This argument is mandatory and we provide input files for each network we characterize. These files are in the directory “params” of Cloudgen repository. Note, files have the follow identifier: ID\_param[\_hshaX].conf, where ID identifies the characterized network, and X means number of times we manually increased the mean number of devices per namespace to reproduce high sharing scenarios as shown in Section 6 of the technical report.

**TIMEOUT:**

Duration of the simulation in days. Default is 30 days.

**SEED:**

Random seed for synthetic workload generation. Default is an uniform random number.

**OUTLAB:**

This is a label to identify the content sharing network output used by the synthetic workload. This label is alphanumeric characters. Default label is "0".

**Examples**

In the example below, we run Cloudgen to generate 1 week (Monday-Friday) of synthetic workload in campuses networks:

```
./cloudgen.py -t 10 -d 4967 -n 6795 -c params/CAMP1_param.conf -l test > gen_workload.txt &
```

Note, we set CloudGen to generate synthetic traces of two weeks (i.e., 10 days), to make sure various ranges of values for each component are used in the synthetic trace. Note, we can discard the initial week to remove simulation warm-up effects.

Number of devices and namespaces above are mean values we compute given various weeks of dataset collected in Campus-1. However, different number of devices and namespaces can also be adopted, and we use the configuration parameters characterized for Campus-1 in “params/CAMP1\_param.conf” file.

We suggest analysis of synthetic workloads based on various runs of Cloudgen.

Thus, the example below shows a shell script to run Cloudgen sixteen times. We use as input parameters for Cloudgen our characterized PoP1 workload.

```
N_SAMPS=16
for i in `seq 1 16 ${N_SAMPS}`
do
    out=`printf "%06d" $i`
    ./cloudgen.py -t 10 -d 3216 -n 4282 -c params/POP1_param.conf -l ${out} > gen_${out}.txt &
done;
```

## Output

Cloudgen produces as output a synthetic workload trace representing three events of devices: log-in, log-out and modifications in files (i.e., namespaces)

Cloudgen generates this synthetic trace to standard output, where each row corresponds to events of device and each column is associated to a specific measure.

Below, we describe each device event column.

### Device log-in and log-out

Column	Type	Description
1	event flag	“dev_on” for log-in and “dev_off” for log-out
2	time stamp	time of event (epoch)
3	-	“-” place holder for a new type
4	-	“-” place holder for a new type
5	-	“-” place holder for a new type
6	device id	devices identifier (int)
7	-	“-” place holder for a new type
8	time on/off	if flag is “dev_on”, this column means the period the device was offline (seconds), otherwise, a flag “dev_off” means the period device was online

### Namespace modifications

Column	Type	Description
1	event flag	“f_changed” to represent a modification
2	time stamp	time of event (epoch)
3	-	“-” place holder for a new type
4	-	“-” place holder for a new type
5	-	“-” place holder for a new type

6	device id	devices identifier (int)
7	-	“-” place holder for a new type
8	namespace id	namespace identifier (int)
9	previous version	previous version of device's namespace
10	current version	current version of device's namespace
11	session time	period of device was online (seconds)
12	volume	volume of modification in namespace (bytes)
13	upload/download flag	event flag to represent the modification upload or download
14	-	“-” place holder for a new type

## Contact

For further information about Cloudgen, please contact the authors: Glauber ([ggoncalves@dcc.ufmg.br](mailto:ggoncalves@dcc.ufmg.br)) or Idilio ([idilio.drago@polito.it](mailto:idilio.drago@polito.it)).