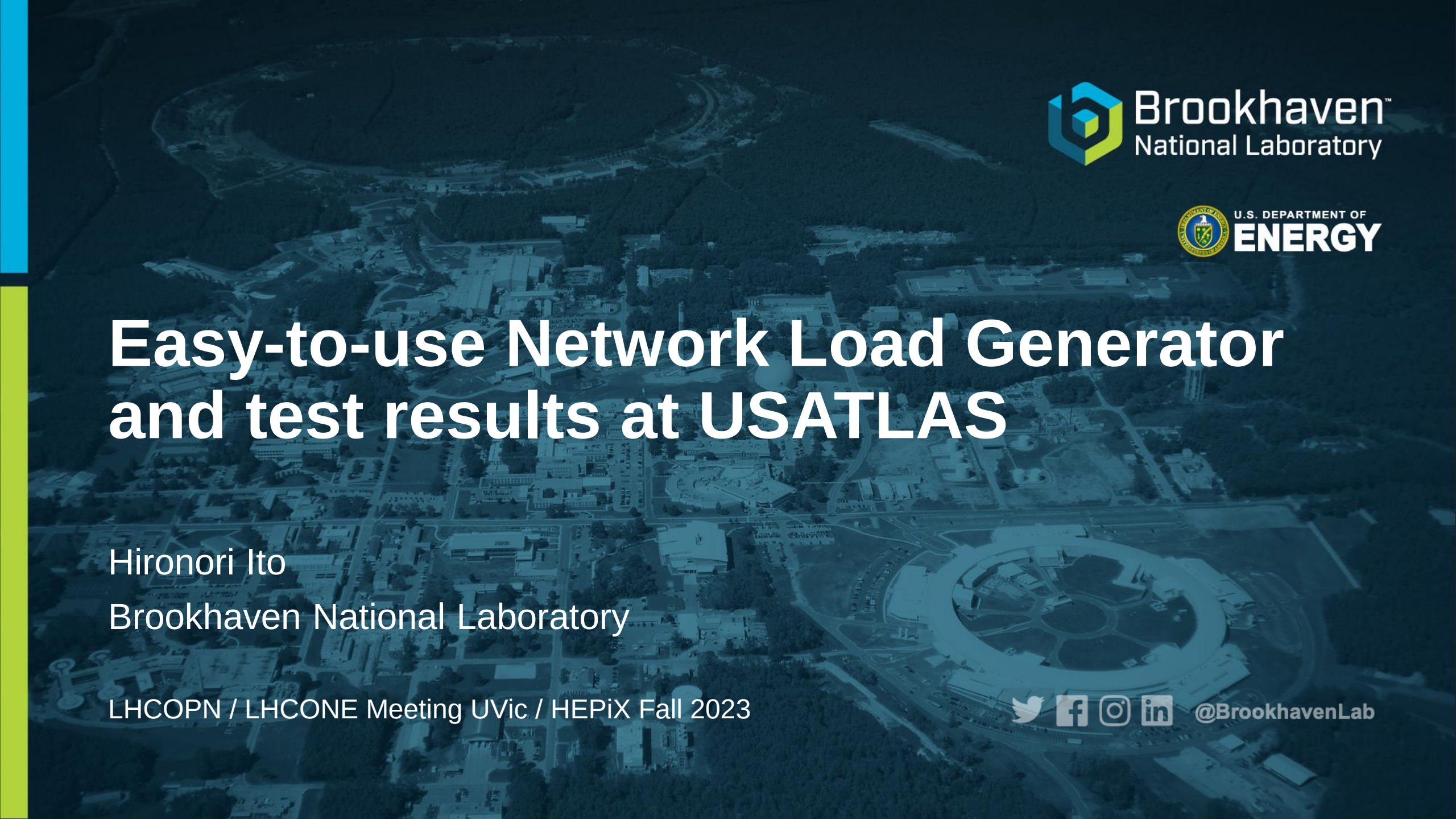


Easy-to-use Network Load Generator and test results at USATLAS

Hironori Ito

Brookhaven National Laboratory

LHCOPN / LHCONE Meeting UVic / HEPiX Fall 2023

@BrookhavenLab

Network Testing Tools for HL-LHC

HL-LHC

- Series of Data challenge
  - Important! …
  - DC24, 26, …
  - Mini-challenge
    - Site testing
      - Site and cloud admins want to know if they are ready.
    - We should have an easy-to-use, load testing program for any site/cloud admins to use since they are the people who can fix things.

Easy-to-use Load Generator

Some programs for Network Load Generator

Scripts/programs to do easy testing

- They are found at BNLBox
  - https://bnlbox.sdcc.bnl.gov/index.php/s/XGs6LJEGNzf69zK
  - NetworkLoadGen.rb
    - Easy to use load generator
  - ftsDelegate.rb
    - Similar to fts-rest-delegate
  - webdav-ls.rb
    - Similar to gfal-ls (on steroid)
  - webdav-rm.rb
    - Similar to gfal-rm (be careful)

Load Generator

NetworkLoadGen.rb

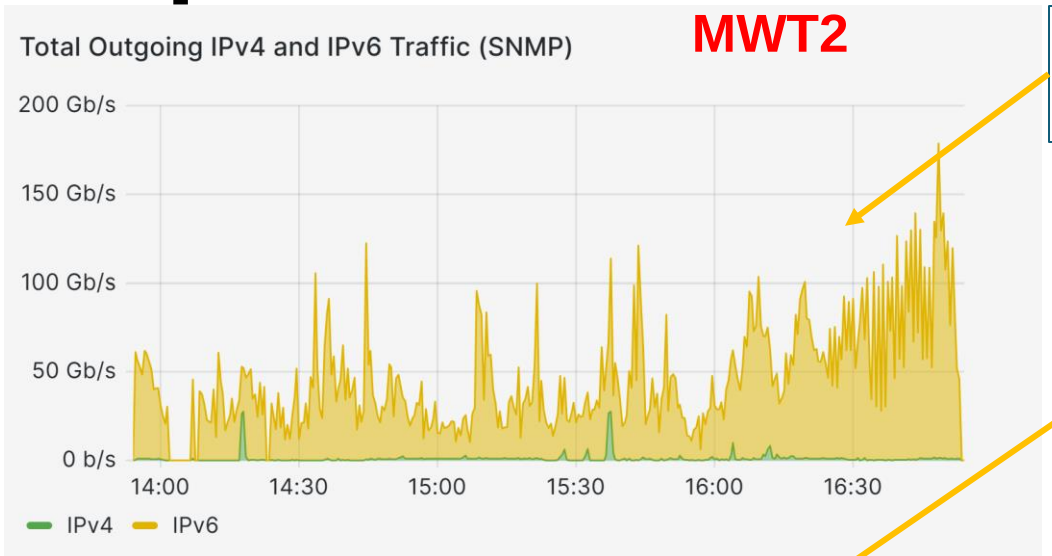
- It generates network load at the desired rate for specified time period.
- For source files, it can query the source storage for given path specified by user's option. Alternatively, the list of input files to be used for the load generation can be given. (**webdav-ls.rb** can be used for easy generation)

```
ruby NetworkLoadGen.rb --help
Usage: ruby NetworkLoadGen.rb FTSURL
    -h, --help                    Display help message
    -p, --proxy PROXYFILE         Set X509 Proxy File to
be used
    -v, --verbose                 Enable verbose mode
    -d, --debug                   Enable debug mode
    -c, --ca_path CAPATH          Set CA directory path
    -f, --file listfile           File name for transfer files
    -s, --checksum                Enable Checksum check
        --source SOURCE-URL       Specify URL for source
file(s)
    -r, --rate RATE               Specify target data rate in
GB/s
        --duration duration       Specify duration of the test
in hours
        --recursive_level level   Set the depth of recursive
query used with --source option
        --bring_online            Make Bring Online request
        --min_size size           Set minimum file size to be
used with --source option
        --copy_mode mode          Set Copy mode of TPC
```

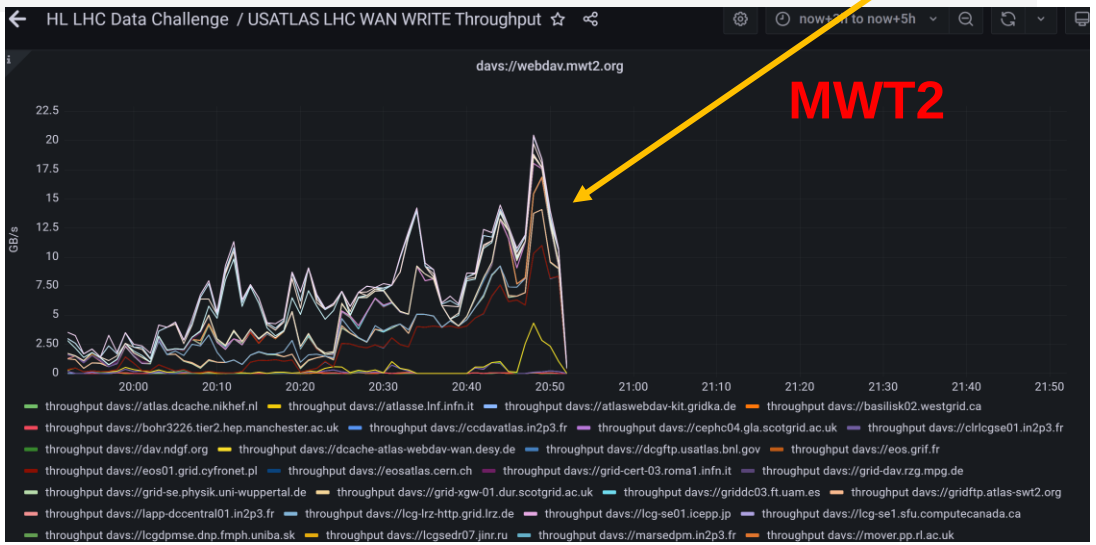
Example

- ruby NetworkLoadGen.rb -r **rate** --duration **time** –source source_URL FTSURL destination_URL
- ruby NetworkLoadGen.rb -r **1** --duration **0.15** –source davs://dcgftp.usatlas.bnl.gov:443/pnfs/usatlas.bnl.gov/BNLT0D1/hiro/DAVS/2/ https://fts.usatlas.bnl.gov:8446 davs://webdav.mwt2.org:2881/atlasdatadisk/hiro/DAVS/test/
- Or, ruby NetworkLoadGen.rb -r **rate** --duration **time** –file SOURCE_FILE_LIST FTSURL destination_URL
  - Format of the source_file_list is; sURL filesize checksum

Peeling through the monitors to see the requested network load by the generator

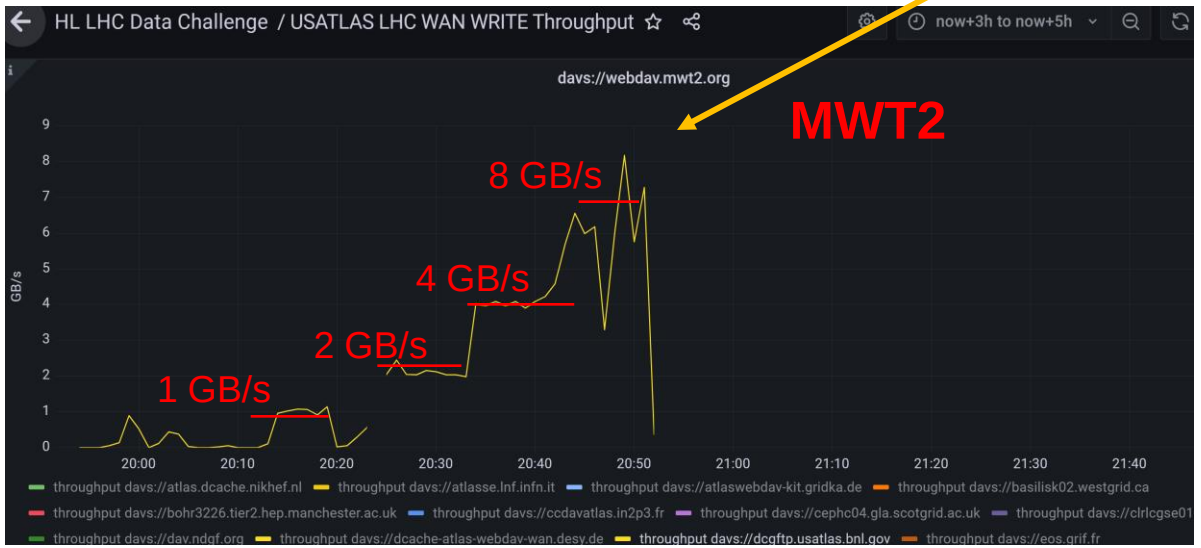


- Site wide throughput
  - Not quite clear



- FTS monitor
  - Not quite clear by combined throughput

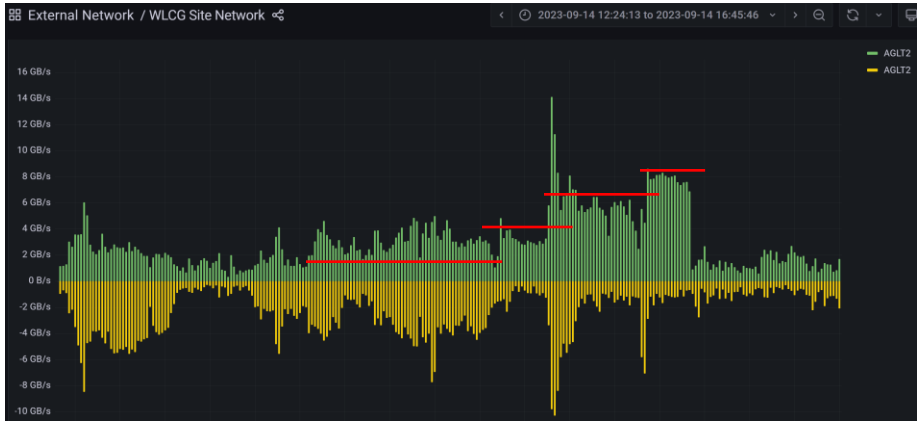
- Visible in a specific link monitor by BNL FTS

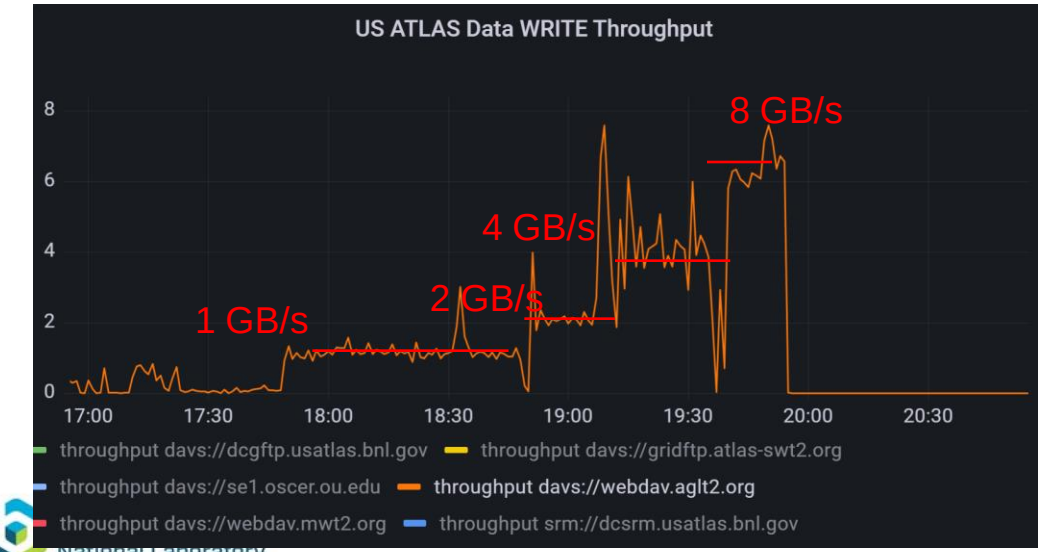


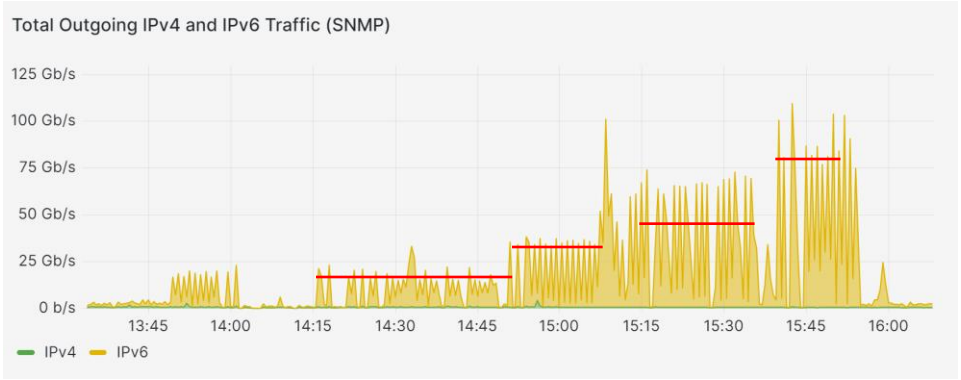
Brookhaven National Laboratory

AGLT2 with checksum

- No issue getting the desired rate up to 8GB/s
- The script can target the rate for the specied duration.
- Checksum has no effect at the throughput.
  - It is expected for dCache site like AGLT2.
- All three monitor shows the same throughputs







How to generate list of files for source?

- The format of the list of files:
  - (s)URL file-size checksum
  - davs://abc.def:port/mypath 1234567 adler32:AABBCC
  - Or root://abc.def:port/mypath 1234567 adler32:AABBCC
- Use whatever convenient method to generate that list.
  - If not, one can use webdav-ls.rb to generate it.
- webdav-ls.rb (like gfal-ls)
  - webdav-ls.rb -v davs://abc.def:port/mypath
    - Returns filename, filesize, ctime, checksum of files in thaurt path/directory
    - --recursive_level N to specify how deep it should go down for the search directory

```
ruby webdav-ls.rb --help
Usage: webdav-ls.rb WEBDAV-URL/PATH
    -h, --help                       Display help message
    -p, --proxy PROXYFILE            Set X509 Proxy File to be used
    -v, --verbose                    Enable verbose mode
    -d, --debug                      Enable debug mode
    -c, --ca_path CAPATH             Set CA directory path
    -s, --sort STYPE                 Set sort option; name, size, ctime
        --directory                  Show directory
        --recursive_level level      Depth of recursive query
```

Brookhaven National Laboratory

An example of how to make a source list

```
ruby webdav-ls.rb -v davs://webdav.aglt2.org:2880/atlasdatadisk/rucio/mc20_13TeV/f3/5c/
/atlasdatadisk/rucio/mc20_13TeV/f3/5c/AOD.31871877._002529.pool.root.1 763204082023-01-26 03:44:19 UTC
        adler32=30de09a1
/atlasdatadisk/rucio/mc20_13TeV/f3/5c/DAOD_JETM1.30673846._000284.pool.root.1 4139686505 2022-
    10-02 17:51:56 UTC adler32=d0d1d7e6

ruby webdav-ls.rb -v davs://webdav.aglt2.org:2880/atlasdatadisk/rucio/mc20_13TeV/f3/5c/ |cut -f1,2,4 |sed s',=,:,'
/atlasdatadisk/rucio/mc20_13TeV/f3/5c/AOD.31871877._002529.pool.root.1 76320408adler32:30de09a1
/atlasdatadisk/rucio/mc20_13TeV/f3/5c/DAOD_JETM1.30673846._000284.pool.root.1 4139686505
        adler32:d0d1d7e6

ruby webdav-ls.rb -v davs://webdav.aglt2.org:2880/atlasdatadisk/rucio/mc20_13TeV/f3/5c/ |cut -f1,2,4 |sed s',=,:,' |sed
    s',/atlasdatadisk,davs://webdav.aglt2.org:2880/atlasdatadisk,'
davs://webdav.aglt2.org:2880/atlasdatadisk/rucio/mc20_13TeV/f3/5c/AOD.31871877._002529.pool.root.1
        76320408adler32:30de09a1
davs://webdav.aglt2.org:2880/atlasdatadisk/rucio/mc20_13TeV/f3/5c/DAOD_JETM1.30673846._000284.pool.root.1
        4139686505 adler32:d0d1d7e6
```

Some fun facts

More information is recorded in the extended attributes of the file system

- xdg.origin.url
  - Storage from sites has information from where the file were copied.
  - Example file : davs://webdav.aglt2.org:2880/atlasdatadisk/rucio/mc20_13TeV/f3/5c/HITS.28773506._003248.pool.root.1
  - It came from https://basilisk02.westgrid.ca:2880/atlasdatadisk/rucio/mc20_13TeV/f3/5c/HITS.28773506._003248.pool.root.1?copy_mode=pull
- Checksum
  - Storage from sites stores multiple checksum types.
  - davs://bohr3226.tier2.hep.manchester.ac.uk:443/dpm/tier2.hep.manchester.ac.uk/home/atlas/atlasdatadisk/rucio/mc23_13p6TeV/78/37/log.35008669._000812.job.log.tgz.1
    - "checksum.adler32": "48f6224d", "checksum.md5": "f2cca31091a71f3b7f4ecf92d9612169"

Clean up of the test data.

- It is very easy to clean up test data because of RFC 4918.
- It acts exactly like
  rm –rf /MYDIRECTORY
- Any webdav client can be used.
  - BE CARFUL!!!!!
  - webdav-rm.rb sURL/path

RFC 4918 **WebDAV** June 2007

9.6.1. DELETE for Collections

The **DELETE** method on a collection MUST act as if a **"Depth: infinity"** header was used on it. A client MUST NOT submit a Depth header with a DELETE on a collection with any value but infinity.

DELETE instructs that the collection specified in the Request-URI and all resources identified by its internal member URLs are to be deleted.

Conclusion

- Load generator has been created.
- It has shown to produce the desired throughputs for requested time.
- A site and/or cloud admins can use it to test own sites.
  - The result of the test can used to proactively find the limiting factors for their sites in the preparation for DC24 and beyond.