Stress-testing of ObjectStores and its analysis

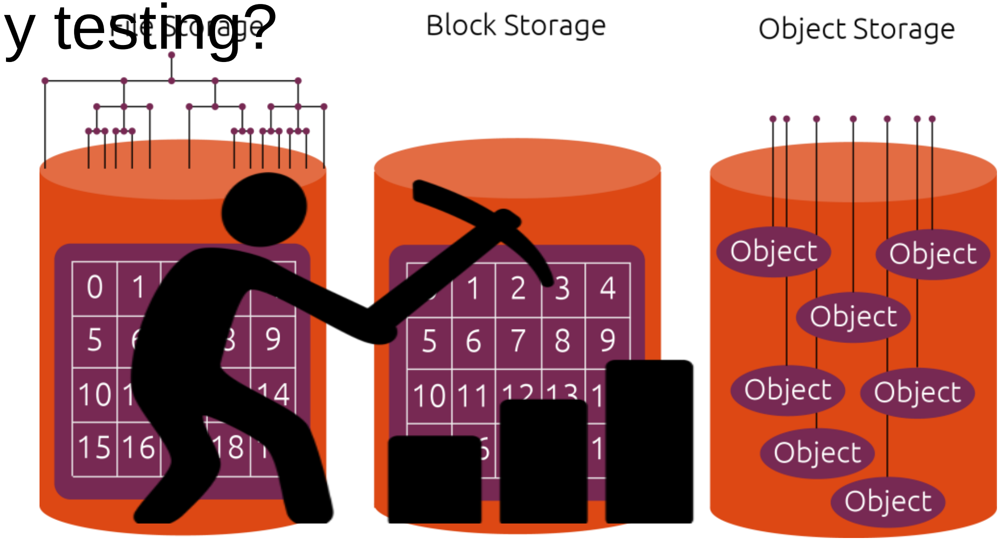
Eliška Jelínková, Marek Kostka, Jaroslava Schovancová

Motivation

- What was our task?
- What did we learn?

Why ObjectStores?

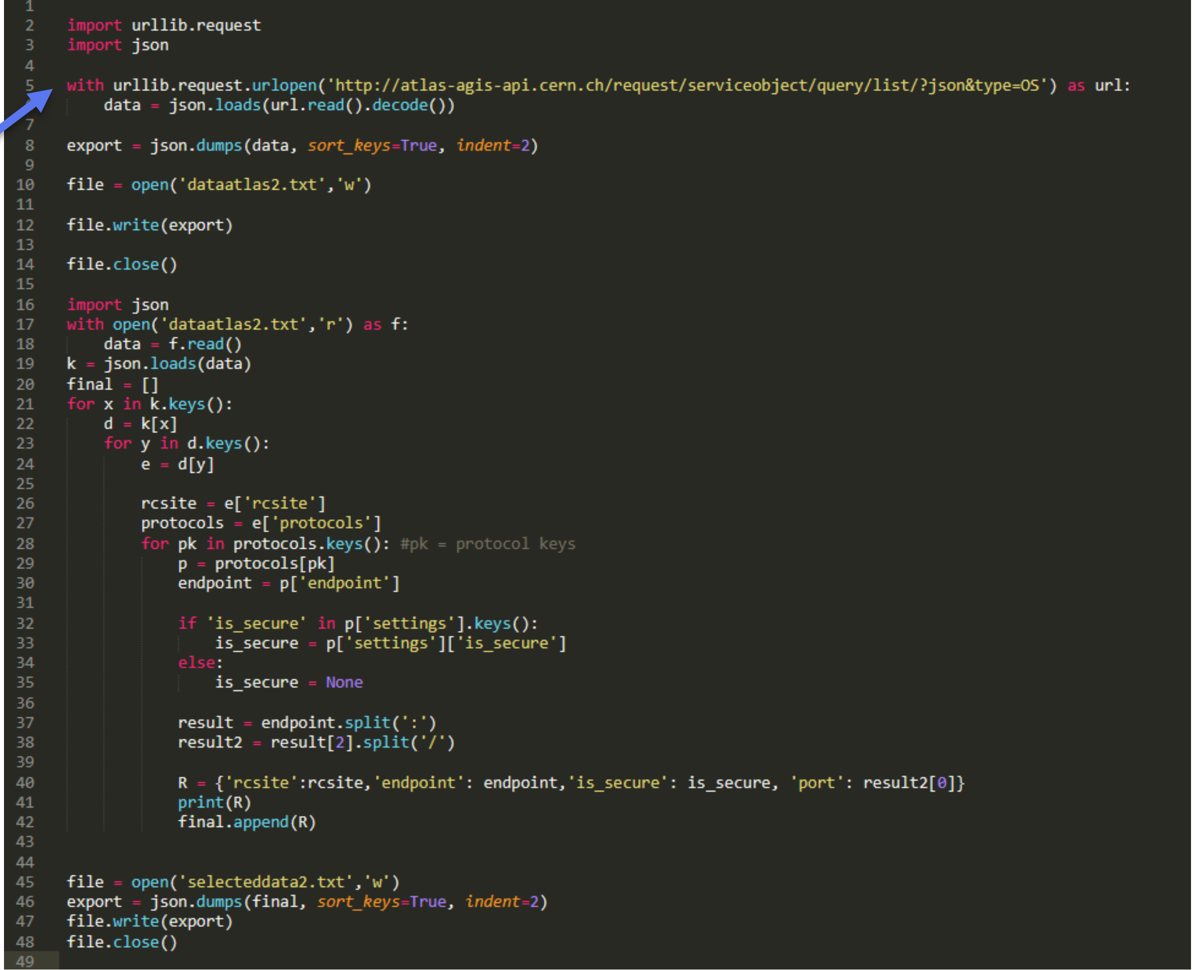
Why testing?



How we prepare data for stress-testing:

Data downloading

URL import

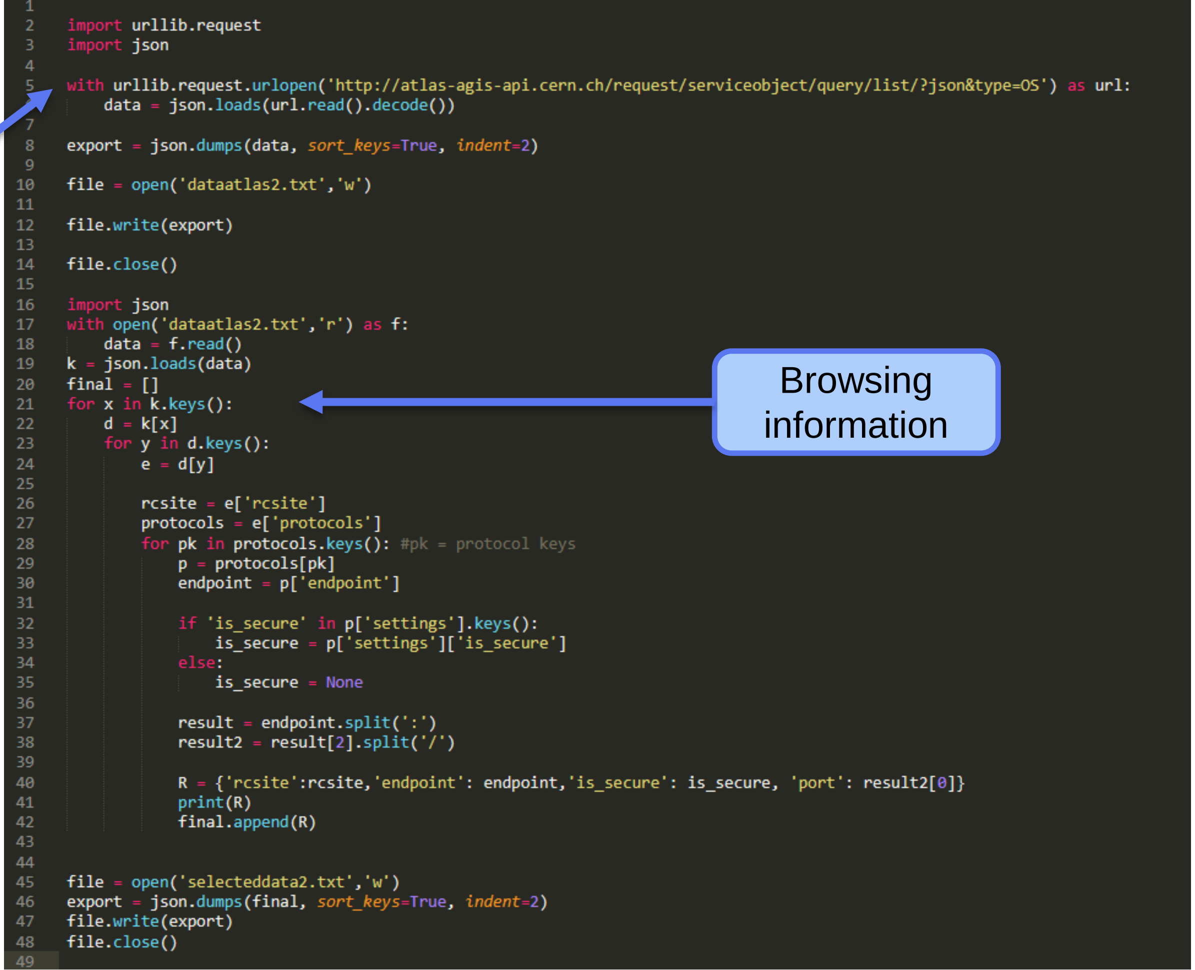


```python
import urllib.request
import json

with urllib.request.urlopen('http://atlas-agis-api.cern.ch/request/serviceobject/query/list/?json&type=OS') as url:
    data = json.loads(url.read().decode())

export = json.dumps(data, sort_keys=True, indent=2)

file = open('dataatlas2.txt','w')

file.write(export)

file.close()

import json
with open('dataatlas2.txt','r') as f:
    data = f.read()
k = json.loads(data)
final = []
for x in k.keys():
    d = k[x]
    for y in d.keys():
        e = d[y]

        rcsite = e['rcsite']
        protocols = e['protocols']
        for pk in protocols.keys(): #pk = protocol keys
            p = protocols[pk]
            endpoint = p['endpoint']

            if 'is_secure' in p['settings'].keys():
                is_secure = p['settings']['is_secure']
            else:
                is_secure = None

            result = endpoint.split(':')
            result2 = result[2].split('/')

            R = {'rcsite':rcsite,'endpoint': endpoint,'is_secure': is_secure, 'port': result2[0]}
            print(R)
            final.append(R)


file = open('selecteddata2.txt','w')
export = json.dumps(final, sort_keys=True, indent=2)
file.write(export)
file.close()
```

How we prepare data for stress-testing:

Data downloading

Data processing

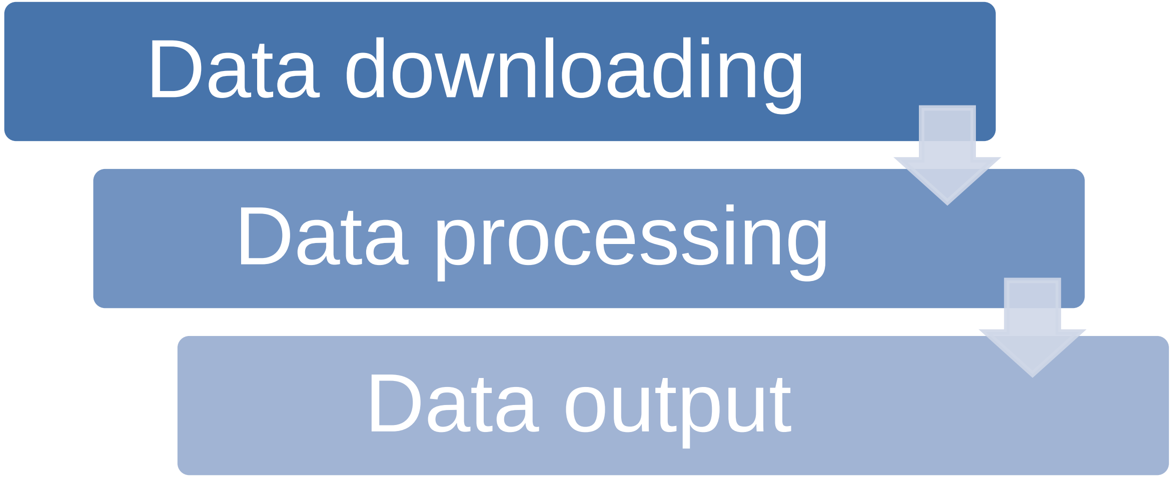


```python

import urllib.request
import json

with urllib.request.urlopen('http://atlas-agis-api.cern.ch/request/serviceobject/query/list/?json&type=OS') as url:
    data = json.loads(url.read().decode())

export = json.dumps(data, sort_keys=True, indent=2)

file = open('dataatlas2.txt','w')

file.write(export)

file.close()

import json
with open('dataatlas2.txt','r') as f:
    data = f.read()
k = json.loads(data)
final = []
for x in k.keys():
    d = k[x]
    for y in d.keys():
        e = d[y]

        rcsite = e['rcsite']
        protocols = e['protocols']
        for pk in protocols.keys(): #pk = protocol keys
            p = protocols[pk]
            endpoint = p['endpoint']

            if 'is_secure' in p['settings'].keys():
                is_secure = p['settings']['is_secure']
            else:
                is_secure = None

            result = endpoint.split(':')
            result2 = result[2].split('/')

            R = {'rcsite':rcsite,'endpoint': endpoint,'is_secure': is_secure, 'port': result2[0]}
            print(R)
            final.append(R)


file = open('selecteddata2.txt','w')
export = json.dumps(final, sort_keys=True, indent=2)
file.write(export)
file.close()

```

URL import

Browsing information

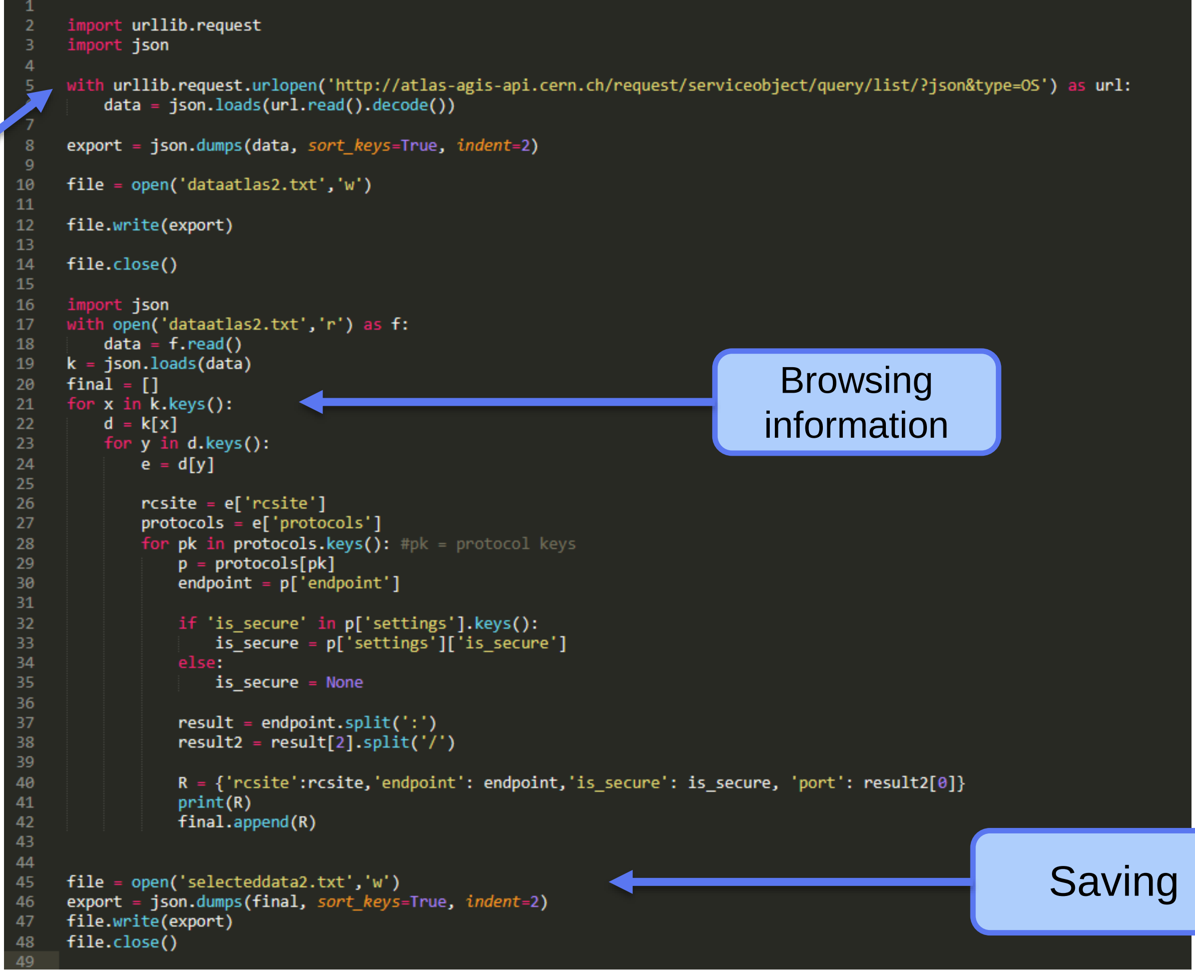
How we prepare data for stress-testing:



Data downloading

Data processing

Data output



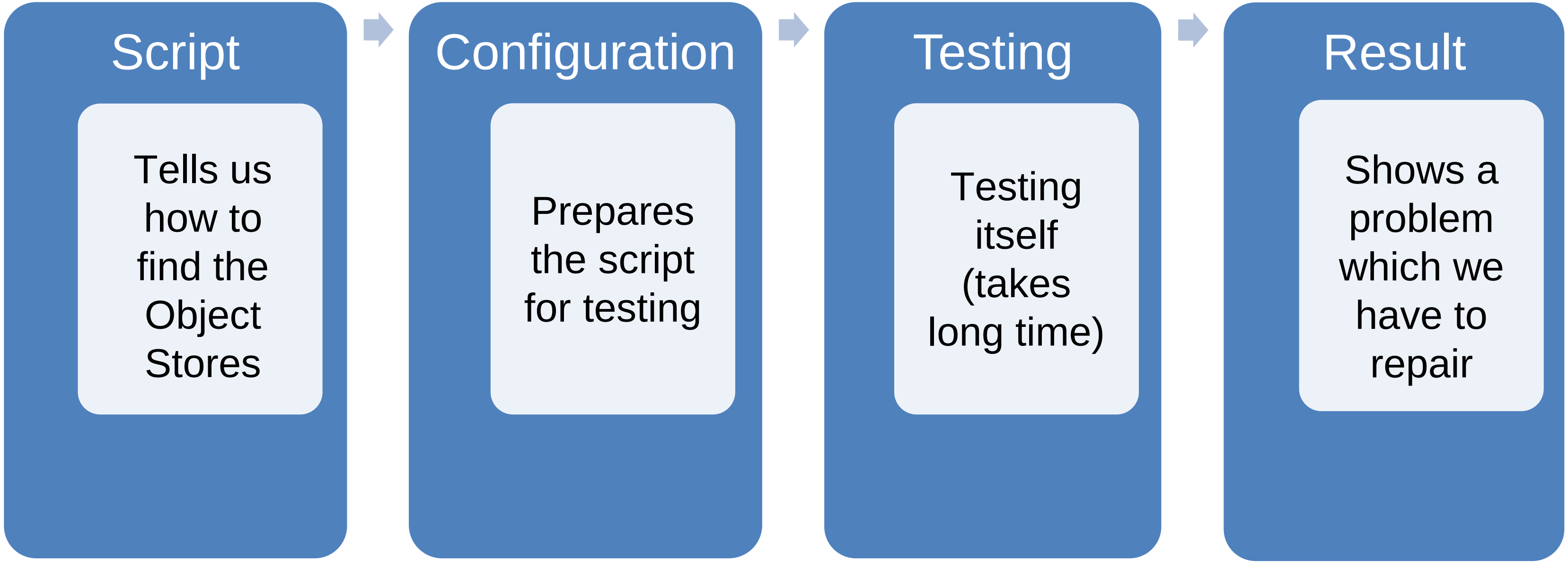
```python

import urllib.request
import json

with urllib.request.urlopen('http://atlas-agis-api.cern.ch/request/serviceobject/query/list/?json&type=OS') as url:
    data = json.loads(url.read().decode())

export = json.dumps(data, sort_keys=True, indent=2)

file = open('dataatlas2.txt','w')

file.write(export)

file.close()

import json
with open('dataatlas2.txt','r') as f:
    data = f.read()
k = json.loads(data)
final = []
for x in k.keys():
    d = k[x]
    for y in d.keys():
        e = d[y]

        rcsite = e['rcsite']
        protocols = e['protocols']
        for pk in protocols.keys(): #pk = protocol keys
            p = protocols[pk]
            endpoint = p['endpoint']

            if 'is_secure' in p['settings'].keys():
                is_secure = p['settings']['is_secure']
            else:
                is_secure = None

            result = endpoint.split(':')
            result2 = result[2].split('/')

            R = {'rcsite':rcsite,'endpoint': endpoint,'is_secure': is_secure, 'port': result2[0]}
            print(R)
            final.append(R)


file = open('selecteddata2.txt','w')
export = json.dumps(final, sort_keys=True, indent=2)
file.write(export)
file.close()

```

URL import

Browsing information

Saving

Stress-testing



Script

Tells us how to find the Object Stores

Configuration

Prepares the script for testing

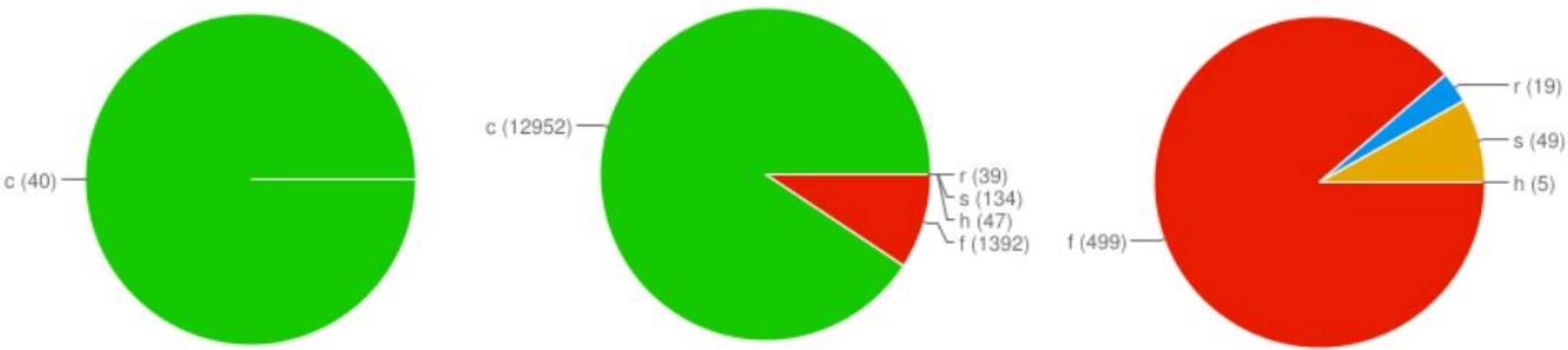
Testing

Testing itself (takes long time)

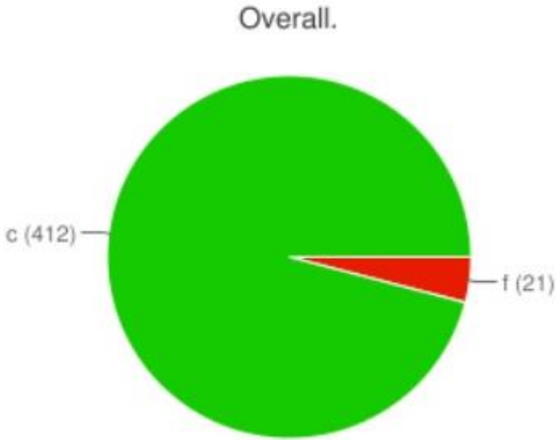
Result

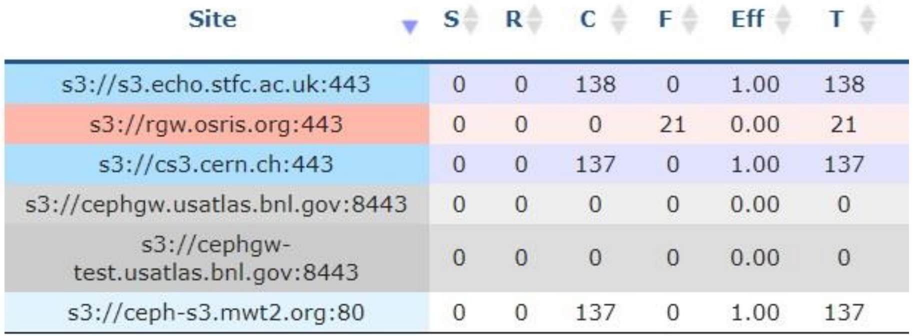
Shows a problem which we have to repair

Stress-testing examples



Results of our testing





Site	S	R	C	F	Eff	T
s3://s3.echo.stfc.ac.uk:443	0	0	138	0	1.00	138
s3://rgw.osris.org:443	0	0	0	21	0.00	21
s3://cs3.cern.ch:443	0	0	137	0	1.00	137
s3://cephgw.usatlas.bnl.gov:8443	0	0	0	0	0.00	0
s3://cephgw-test.usatlas.bnl.gov:8443	0	0	0	0	0.00	0
s3://ceph-s3.mwt2.org:80	0	0	137	0	1.00	137

Stress-testing of ObjectStores and its analysis

Summary

Write a script in Python

Convert the URL to .json file
Set the testing procedure
Run the test
Analyse the results

Thank you for your attention

Eliška Jelínková, Marek Kostka, Jaroslava Schovancová