EGI-Engage

HBP –Report on the Data Hosting Pilot

Date	24 March 2016
Activity	Engage-WP4
Lead Partner	EGI.eu
Document Status	FINAL
Document Link	

Abstract

This is a report of the activities of the HBP/EGI collaboration from October 2015 to March 2016, the pilot data hosting service run by EGI.eu for the Human Brain Project (HBP) and the technical preparations of moving the service into a production-ready state. The report also includes potential future directions that the service can take.

The EGI-Engage project is co-funded by the European Union (EU) Horizon 2020 program under Grant number 654142 http://go.egi.eu/eng

DOCUMENT LOG

Issue	***Date***	***Comment***	***Author/Partner***
v.0.1	23/03/2016	First draft	Matthew Viljoen/EGI.eu
v.0.2	24/03/2016	Expanded information on scalability and testing	Matthew Viljoen/EGI.eu
v1.0	24/03/2016	Incorporation of comments, corrections and extra input in the Potential Future Developments section	Matthew Viljoen/EGI.eu Christian Bernardt/DESY Łukasz Dutka/Cyfronet

Contents

1 Introduction

In October 2015, collaboration started between EGI.eu and HBP consisting of a series of fortnightly meetings[1] with the aim to develop a pilot service consisting of a prototype data hosting solution on EGI infrastructure for the HBP. The solution meets the usecase where HBP datasets are stored on a reliable and scalable hosting infrastructure for the purposes of visualization.

The aim of this collaboration has been to:

- Understand and define the requirements of the data hosting usecase
- Build a prototype solution
- Develop and run performance tests to show that the prototype meets HBP requirements and understand what would be required for future scaling up of the service
- Plan for the transition of the pilot to production

This report includes details of the work done since the beginning of the collaboration, details of the how the pilot service has been deployed and plans for future development of the service.

[1] https://indico.egi.eu/indico/category/174/

2 Overview of Activities

The following is an overview of the timeline and significant activities involved in the development of a pilot data hosting service by EGI.eu for HBP.

Date	Activity
Apr '15	HBP EGI Usecases written (Łukasz Dutka et al.)[2]
Sep '15	Start of bi-weekly meetings[3] chaired by Matthew Viljoen (involving Catherine Zwahlen, Łukasz Dutka, Christian Bernardt et al.) with the aim of preparing for a demo of the HBP data hosting and visualization testcase at the 2015 EGI Community Forum, Bari, IT. Storage at the EGI site in INFN-Bari is configured to be able to receive data from HBP for the pilot.
Oct '15	Over this time a sample 300GB dataset was transferred to data hosting sites at two EGI Research Centres (Cyfronet/Onedata and DESY/dCache) and HBP-developed Image Service was deployed at these sites, residing in a Docker. Simple viewing applications which displayed image tiles at both sites were developed, in addition to the creation of a simple animation of a single tile traversing a dataset[4] (NB a *tile* is a small segment of a HBP image, 256x256px in size). These applications were demonstrated at the EGI 2015 Community Forum[5].
Nov '15	Test script was developed to test how many users EGI data hosting sites can serve data while meeting the target of retrieving 8 tiles in <1s from one EGI data hosting site. (1 user with dCache and 2 users with Onedata, no load balancing)
Jan '16	HBP/EGI bi-weekly meeting became more open, advertised on the EGI federated data virtual team, with anyone open to join. Regular new attendees were Jeffrey Muller and Huanxiang Lu. The HBP image visualization application, ATLASViewer, was provided and deployed on EGI federated cloud for testing purposes. Both EGI data hosting sites implement load balancing in front of multiple Dockers running the Image Service (NGINX at Cyfronet, F5 at DESY). This increases the number of users that can be served in the target time of 1s from 2 to >10 HBP/EGI Management meeting takes place with Sean Hill to better

[2] https://documents.egi.eu/secure/ShowDocument?docid=2468

[3] https://indico.egi.eu/indico/categoryDisplay.py?categId=174

[4] https://dl.dropboxusercontent.com/u/8192091/slicing.mp4

[5] https://indico.egi.eu/indico/event/2544/session/39/?slotId=0#20151111

	understand the requirements for moving to production.
Feb ‘16	A new testing suite is developed that runs against the ATLASViewer, giving a more realistic simulation of load on the data hosting sites. Newer, larger datasets are prepared for transfer to EGI. HBP decides that the ATLASViewer will be run and maintained by HBP whilst in production, requesting images from data hosting EGI sites. OpenIDConnect authentication will be implemented by HBP into the ATLASViewer, and not at the Image Services running at data hosting EGI sites at present.
Mar ‘16	More datasets are transferred to EGI hosting sites via INFN-Bari. Total hosting data is now approximately 2TB. A new convention is developed whereby all hosting sites adopt the same data directory structure. Discussions are held to understand the remaining technical prerequisites needed to move to full production A newer and more efficient testing suite is developed to test the data hosting sites that is capable of scaling up to simulate many hundreds of concurrent users.

3 Pilot Setup

Infrastructure involved in the pilot and its testing consists of the following components:

- 5TB data service for staging data from HBP to the EGI data hosting sites, accessible over sftp and hosted by INFN-Bari
- 4TB dedicated Onedata instance on top of 5PB Lustre storage at Cyfronet hosting the HBP data and exposing it by POSIX to 10 Docker instances running the HBP Imaging web service, load balanced using NGINX
- 19TB dCache instance at DESY, hosting the HBP data and exposing it by POSIX to 10 Docker instances running the HBP Imaging web service, load balanced using F5. See Appendix 2 for further details of the setup including configuration files.
- 2 VMs running on the EGI Federated Cloud running test instances of ATLASViewer, one in front of Cyfronet/Onedata and one in front of DESY/dCache

Software developed and used as part of the pilot is as follows:

- Image Service provided by HBP, run in Docker containers at the EGI data hosting sites
- ATLASViewer visualization application provided by HBP, run separately from the EGI data hosting sites
- Test script written in Python (Łukasz Dutka et al.) capable of sending parallel requests to the imaging service in order to simulate realistic user requests and to test the pilot's meeting of HBP targets[6]
- Test script written in javascript using Angular-Protractor (Christian Bernardt) designed to simulate end user requests to the ATLASViewer and verify that the hosting services meet the demands of stressing the ATLASViewer[7].
- Test script written in Golang (Christian Bernardt) capable of sending parallel requests to the imaging service in order to simulate realistic user requests and to test the pilot's meeting of HBP targets[8]

[6] https://dl.dropboxusercontent.com/u/8192091/HBP-master.zip

[7] https://github.com/chrber/hbp-atlas-viewer-test

[8] https://github.com/chrber/hbp-performance-test/blob/master/httpRequest.go

4 Scalability Considerations

By the nature of the usecase served by the pilot and the type of service being provided by EGI, scalability in this pilot is determined by the number of concurrent users that can be serviced by any given EGI data-hosting site for HBP. Soon after the start of the collaboration, the basic target was given of 8 image tile requests being serviced by a hosting site in under a second[9], as this corresponds to a typical request from the HBP visualization software.

It was demonstrated using a test script written for the pilot (see Section 3, Footnote 6) that only one or two users could be serviced by the EGI data-hosting sites due to limitations in the HBP imaging service (lack of full parallelism and potential for optimization).

Table 1 – Image serving times[10]

#Users	dCache	Onedata
1	790ms	560ms
2	1300ms	700ms
3	1800ms	1000ms
4		1200ms

With load-balancing implemented at both EGI data-hosting sites, sample average results are as follows, using randomly chosen tiles from a sample dataset:

#Users	dCache	Onedata
1	979.55ms	943.02ms
5	954.58ms	935.14ms
10	963.36ms	998.66ms
15	955.31ms	1030.87ms
20	980.62ms	1193.38ms
25	1469.84ms	1314.01ms
30	2106.38ms	1637.65ms
50	2589.30ms	3449.66ms
100	9101.70ms	5460.86ms
200	30900.36ms	44338.79ms

The numbers show a roughly linear increase in serving time until a point where the serving time dramatically increases, which is the point where the maximum number of load balanced Docker images has been reached.

It is important to note that these sample results vary significantly across different runs of the testing script. Although this may be due to variable network congestion and local caching factors, it has been observed that there is there is considerable access time variation depending on which tile is chosen from a dataset – some tiles appear to be served by the Image Service faster than other tiles. During the testing, the testers did not have information about the structure of the datasets to investigate the case of this variation – more work can be done in this area.

[9] https://indico.egi.eu/indico/event/2865/

[10] https://indico.egi.eu/indico/event/2863/

For further scalability of the number of concurrent users served under the target time of 1 second, more Dockers would need to be created at each EGI data-hosting site and added to the respective load-balancing pools. This could require more Docker hosting machines needing to be deployed. It is difficult to give precise extrapolations from the statistics gathered due to the unknown factors mentioned above and the limited time to run these tests.

However, it is strongly suspected from observing the behaviour of the Image Service during testing that further development and optimization of the Image Service would also lead to improved scalability for serving multiple concurrent user requests.

In terms of scalability of hosting data volumes, both dCache and Onedata are capable of scaling to multiple PBs

5 Move to Production

The following are technical requirements that have been agreed as prerequisites for moving the service into production:

- Serving 10 concurrent requests per EGI data hosting site. This requirement has already been achieved with load balancing (see Section 4).
- Capacity – 5TB per site. This requirement has already been achieved (see Sections: 7.1.2, 8.1)

In addition to the above, EGI.eu would like to achieve the following goal prior to moving into production:

- Functional Monitoring – implementation of a top-level functional test that verifies that the EGI data hosting service is functioning correctly and is able to service a simple tile request. This test can run at a suitable frequency (e.g. hourly) and can automatically alert the relevant operations team if there are any problems.

6 Potential Future Developments

- Additional EGI data-hosting sites for HBP

The pilot has involved Cyfronet and DESY but other EGI Research Centres have expressed interest in joining a future data-hosting service for HBP. This would further add to the future resiliency, capacity and geographical scalability of the service.

- Geographical distribution of data and optimized serving of data

Distributing the data across multiple worldwide sites has the potential of significantly improving the user experience in serving data from the hosting site closest to the user. This is something that could be investigated if the data is increasingly distributed in different EGI research centres.

- Federated access to data

Along with distributing data geographically, a federated approach to accessing data using solutions such as DynaFed or the EGI OpenData platform (based on OneData) could improve with improving scalability and access speeds as well as easier management of the data.

- Improving the transfer of data from HBP to EGI (e.g. Globus Connect)

Transferring and synching large amounts of data can be error-prone and time consuming. Introducing a service such as Globus Connect or FTS3 could make the process of moving multiple large files from HBP to EGI much easier. For example, incorporating these solutions with scripting could automatically distribute datasets across hosting sites.

- Integration of archived datasets with data-hosting

As the amount of HBP data increases, it may be necessary to selectively choose only the active datasets to be available for hosting on EGI. If archived datasets are stored on a data archiving service such as Zenodo, a tighter integration with such a service would make it easier to move datasets that are no longer active for archiving. Conversely, if an archived dataset needs to be visualized, it could be easily moved from an archiving service to EGI for active usage.

- Integration of HBP OpenID Connect to maintain access controls

At the time of writing this report, HBP have no control who can access the data through the Atlasviewer. It seems to be reasonable to introduce a level of control ideally at the storage level. Onedata is ready to integrated with OpenID Connect providing control on file access level if the current HTTP request comes from a user who is entitled to access the particular data file. In such a case data owners would have better control on who can access the scan. There are on-going similar development for dCache to integrate it with OpenID Connect.

7 Appendix 1 – Onedata setup

7.1 Hardware

7.1.1 Docker hosting services

HP Based system with 192GB RAM and
24 Intel Haskell Cores with
10Gbit/s internet connectivity and
Infiniband connectivity for storage.

The service can be reached through these links:

http:// onedata-hbp.grid.cyfronet.pl/

7.1.2 Storage

5TB virtual filesystem based on 5PB LustreFS connected via infiniband FDR. The filesystem is based on Onedata version 2.7.0 Providing FUSE client for POSIX filesystem running inside Docker containers.

7.2 NGINX Load balancer

Runs locally on the same machine where the Docker imaging services are started.

7.2.1 NGINX Configuration

```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}
http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$
request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;
    # Load modular configuration files from the /etc/nginx/conf.d d
irectory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

    upstream onedata {
        server localhost:8841;
        server localhost:8842;
        server localhost:8843;
        server localhost:8844;
        server localhost:8845;
        server localhost:8846;
        server localhost:8847;
        server localhost:8848;
        server localhost:8849;
        server localhost:8840;
    }
    server {
        listen       8888 default_server;
        listen       [::]:8888 default_server;
        server_name  _;
        root         /usr/share/nginx/html;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        location / {
            proxy_pass http://onedata;
        }

        error_page 404 /404.html;
            location = /40x.html {
        }

        error_page 500 502 503 504 /50x.html;
            location = /50x.html {
        }
    }
}
```

8 Appendix 2 – dCache setup

8.1 Hardware and endpoints

The dCache server is on installed on hardware with the following specifications:

Dell R510
Intel Xeon E5620 2.4 GHz QuadCore with HT
RAM 24GB
Data Disk space: 19TB

Further detail is available from the vendor[11].

The service can be reached through these links:

http://dcache-dot12.desy.de/

Docker container service here: http://hbp-image.desy.de:8888/image/v0/api/bbic?fname=%2Fsrv%2Fdata%2FHBP%2FBigBrain_jpeg.h5&mode=ims&prog=TILE+0+0+0+1626+14+4+none+10+1

8.2 dCache

The dCache software version is 2.13.24. Further information is available at the dCache website[12].

8.3 dCache Configuration Files

8.3.1 cache.conf file content:

```
dcache.layout=dcache-dot12
```

8.3.2 Layout file content:

```
admin.enable.colors = false

alarms.db.type = xml

dcache.enable.space-reservation = true

dcache.enable.overwrite = true

webdav.templates.config!header_text = HBP Test dCache

dcache.authn.ciphers = DISABLE_EC,DISABLE_RC4


[dCacheDomain]

 dcache.java.memory.heap=2048m
```

[11] http://www.dell.com/downloads/global/products/pedge/R510_Spec_Sheet.pdf

[12] https://www.dcache.org/manuals/Book-2.13/index-fhs.shtml

```
[dCacheDomain/admin]
[dCacheDomain/alarms]
[dCacheDomain/poolmanager]
[dCacheDomain/spacemanager]
[dCacheDomain/pnfsmanager]
[dCacheDomain/cleaner]
[dCacheDomain/gplazma]
[dCacheDomain/pinmanager]
[dCacheDomain/billing]

[dCacheDomain/topo]
[dCacheDomain/info]
[dCacheDomain/statistics]

[dCacheDomain/ftp]
 ftp.authn.protocol = plain
 ftp.authz.readonly = false
 ftp.net.listen = localhost

[dCacheDomain/transfermanagers]

[nfsDoor]
dcache.java.memory.heap=8192m
[nfsDoor/nfs]
 nfs.version= 4.1, 3
 nfs.enable.portmap=false
 nfs.export.file=/etc/dcache/exports
nfs.namespace-cache.size=8192
nfs.enable.space-reservation=false

[pool1]
[pool1/pool]
 pool.name=pool1
 pool.path=/data/pools/pool1
 pool.wait-for-files=${pool.path}/data
```

8.3.3 gplazma.conf file content:

```
[root@dcache-dot12 ~]# cat /etc/dcache/gplazma.conf
auth    optional  x509
auth    optional  voms
map     optional  vorolemap
map     optional  gridmap
map     requisite authzdb
session requisite autczdb
```

8.4 F5 Load Balancer configuration

hbp-image.desy.de has address 131.169.4.55

```
pool it-hbp-pool {
  monitor all min 1 of http_desy tcp_desy
  members {
     131.169.4.68:8878 {}
     131.169.4.68:8879 {}
     131.169.4.68:cddbp-alt {}
     131.169.4.68:8881 {}
     131.169.4.68:8882 {}
     131.169.4.68:8883 {}
     131.169.4.68:8884 {}
     131.169.4.68:8885 {}
     131.169.4.68:8886 {}
     131.169.4.68:8887 {}
     131.169.4.68:ddi-tcp-1 {}
  }
}

virtual it-hbp-http-service {
  snat automap
  pool it-hbp-pool
  destination 131.169.4.55:ddi-tcp-1
  ip protocol tcp
  rules http-sorry-service-down
  profiles {
     http_snat {}
     tcp {}
  }
}
```

8.5 Docker container

The Docker containers use NFS4.1 mounts to access the data

```
dcache-dot12:/ on /nfs/4 type nfs
(rw,minorversion=1,vers=4,addr=131.169.4.68,clientaddr=131.169.4.68)

[root@dcache-dot12 docker]# cat /etc/dcache/exports
/ localhost(rw) dcache-dot12(rw)


[root@dcache-dot12 docker]# cat docker-control.sh
#!/bin/bash

PORTRANGE=$(echo {8848..8888})

# Prints help screen and exits with error status 2
usage()
{
echo "Usage: $(basename $0) [OPTION]... COMMAND"
    echo
    echo "Valid commands are:"
    echo "start-all"
    echo "stop-all"
    echo "containers-delete-all"
} 1>&2

call_start_containers() {
  for port in $PORTRANGE;
    do docker run -d -p $port:8888 -v /nfs/4/HBP:/srv/data/HBP -it skerrien/hbp-
image-service;
  done
}

call_stop_all() {
  for container in `docker ps -q`;
    do
      docker stop $container;
  done
}

call_delete_allContainers() {
  docker rm `docker ps -qa`
}

case "$1" in
    start-all)
        shift
        call_start_containers
        ;;
    stop-all)
        shift
        call_stop_all
        ;;
    containers-delete-all)
        shift
        call_delete_allContainers
        ;;
    *)
        usage
        ;;
```

```
esac
```

9 Appendix 3 – ATLASViewer on the EGI fedcloud setup

The test ATLASViewer instances are currently installed at the University of Zaragoza EGI federated cloud site. The instances are installed on virtual machines based on barebones Centos 6.7 with the following specification:

4 core 3GHz Intel processor
12 GB HDD
8GB RAM

The instances are running Apache version 2.2.15-47 and the ATLASViewer[13] with the following configuration changes:

9.1.1 jsons/appconfig.json (Onedata)

```
[
    {
        "api": {"image": {"v0": "http:// onedata-
hbp.grid.cyfronet.pl:8888/image/v0/api/bbic" }}
    }
]
```

9.1.2 jsons/appconfig.json (dCache)

```
[
    {
        "api": {"image": {"v0": "http://hbp-image.desy.de:8888/image/v0/api/bbic"
}}
    }
]
```

9.1.3 jsons/annotation/imagestack_info.json

The latest version of the imagestack_info.json has been provided by HBP and is not included in this report in the interests of brevity.

[13] ATLASViewer version: hbp_atlas_viewer-20151222.tar.gz

10 Appendix 4 – Directory structure

Current datasets hosted at the EGI data hosting sites for HBP are listed as follows:

Golgi:

data/stacks/rat/golgi/anno/golgi_parcellation_sagittal_v1.h5
data/stacks/rat/golgi/anno/whs_in_golgi_space_parcellation_v1.h5
data/stacks/rat/golgi/sections/golgi_rat_sections.h5
data/stacks/rat/golgi/sections/golgi_rat_sections_coronal.h5
data/stacks/rat/golgi/sections/golgi_rat_sections_y.h5
data/stacks/rat/golgi/sections/golgi_rat_sections_y_proj2.h5
data/stacks/rat/golgi/sections/golgi_rat_sections_coronal_proj.h5
data/stacks/rat/golgi/sections/golgi_rat_sections_proj2.h5

R602:

data/stacks/rat/r602/anno/r602_anno.h5
data/stacks/rat/r602/sections/r602.h5

Waxholm:

data/template/rat/waxholm/v2/anno/whs_anno.h5
data/template/rat/waxholm/v2/anno/whs_axial_v2.h5
data/template/rat/waxholm/v2/anno/whs_coronal_v2.h5
data/template/rat/waxholm/v2/anno/whs_sagittal_v2.h5
data/template/rat/waxholm/v2/sections/whs.h5
data/template/rat/waxholm/v2/vol/whs_vol.h5

Bigbrain_100um:

data/template/human/bigbrain_100um/sections/bigbrain_100um.h5

Bigbrain _20um:

data/template/human/bigbrain_20um/anno/bigbrain_anno_coronal.h5
data/template/human/bigbrain_20um/anno/bigbrain_anno_sagittal.h5
data/template/human/bigbrain_20um/anno/bigbrain_anno_axial.h5
data/template/human/bigbrain_20um/anno/bigbrain_auditorycortex_anno_coronal.h5
data/template/human/bigbrain_20um/sections/bigbrain.h5

Infant:

data/template/human/infant/anno/infant_anno.h5
data/template/human/infant/anno/infant_anno_axial.h5
data/template/human/infant/anno/infant_anno_coronal.h5
data/template/human/infant/anno/infant_anno_sagittal.h5
data/template/human/infant/sections/template_t1.h5
data/template/human/infant/sections/template_t2.h5

Jubrain:

data/template/human/jubrain/anno/jubrain_anno.h5
data/template/human/jubrain/anno/jubrain_anno_axial.h5
data/template/human/jubrain/anno/jubrain_anno_coronal.h5
data/template/human/jubrain/anno/jubrain_anno_sagittal.h5
data/template/human/jubrain/anno/colin27.h5

Allen_v3:

data/template/mouse/allen_v3/anno/allen_anno.h5
data/template/mouse/allen_v3/anno/allen_anno_axial.h5

data/template/mouse/allen_v3/anno/allen_anno_coronal.h5
data/template/mouse/allen_v3/anno/allen_anno_sagittal.h5
data/template/mouse/allen_v3/sections/allen.h5