Open APIs for Open Minds

Interfacing Cosmos APIs: WebHDFS and Tidoop

Francisco Romero Bueno

Technological Specialist. FIWARE data engineer

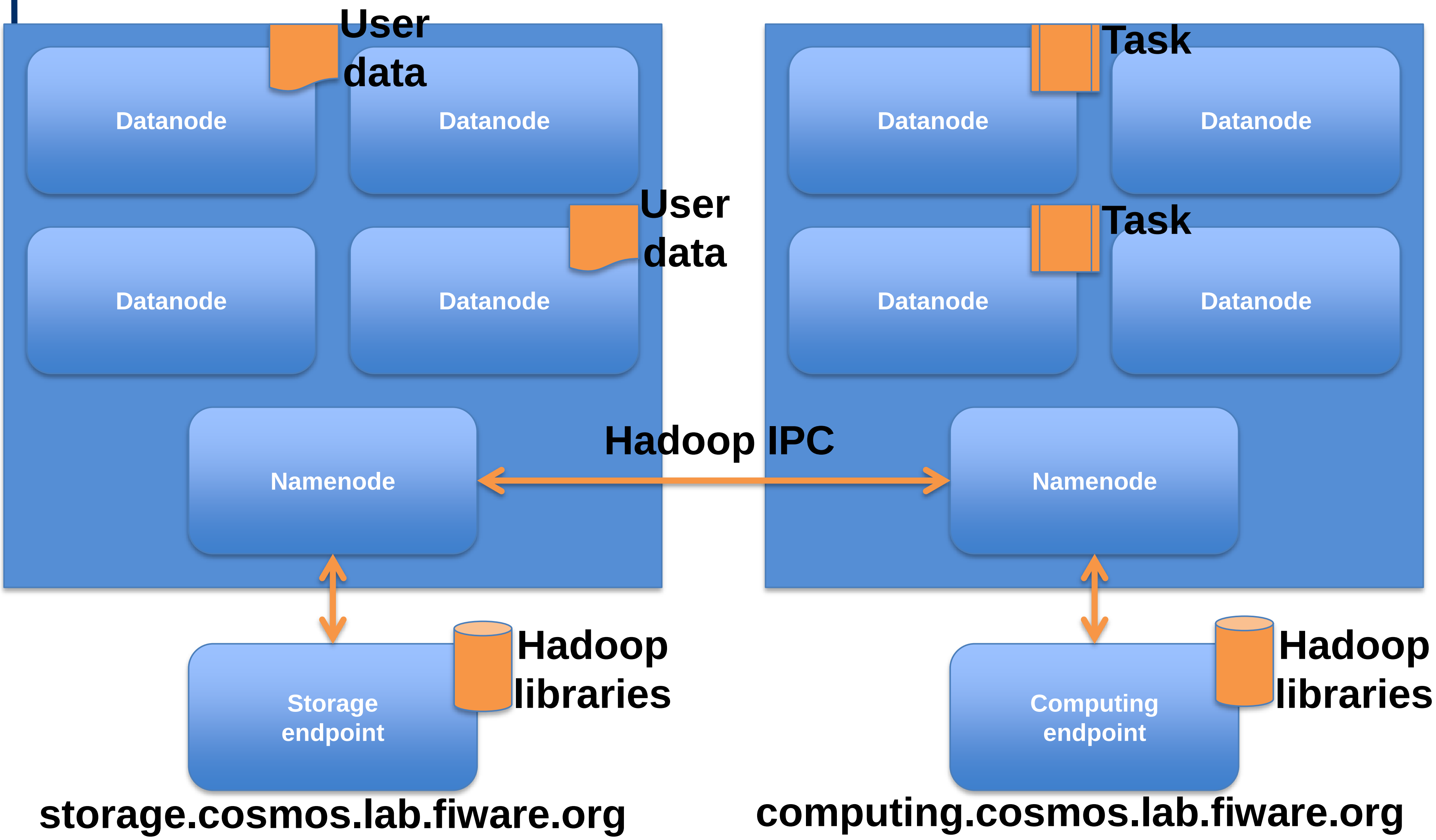
francisco.romerobueno@telefonica.com

FIWARE

Global instance of Cosmos Big Data in FIWARE Lab

- Hadoop-based
  - Hortonworks Data Platfom 2.1
- Split storage and computing sub-clusters
  - This allows for reusing the storage part by other technologies
  - Even, moving the computing cluster to Openstack Sahara
- Each sub-cluster espoxes a single service endpoint
  - Storage endpoint (storage.cosmos.lab.fiware.org)
    - WebHDFS API through HttpFS (TCP/14000)
  - Computing endpoint (computing.cosmos.lab.fiware.org)
    - Auth API (TCP/13000)
    - Tidoop API (TCP/12000)
    - HiveServer2 (TCP/10000)
- All exposed services require OAuth2 authentication
  - An OAuth2 token issued by the IdM or the Auth endpoint must be added to all requests
- All exposed services perform user-based authorization
  - Authenticated users can only access /user/<userID> URIs

Global instance of Cosmos in FIWARE Lab



Using the computing endpoint: Auth API

- First of all, an OAuth2 token must be obtained for authentication purposes in other APIs
- The computing endpoint runs a process in charge of interfacing the global Identity Manager
  - Firstly, such a service process authenticates itself
  - Then, the token is requested using user given credentials and forwarded to the user

```
$ curl -X POST
"https://computing.cosmos.lab.fiware.org:13000/cos
mos-auth/v1/token" -H "Content-Type:
application/x-www-form-urlencoded" -d
"grant_type=password&username=<idm_email>&password
=<idm_password>"

{"access_token": “zRaMnZ5vaWTca5gET0X1x9euYYCx2e",
"token_type": "Bearer", "expires_in": 3600,
"refresh_token": “Xm11a090x0nBK2SJ9eDm6pot5nnkEo"}
```

Registered email and password in global IdM must be given

How authentication and authorization work

- Once obtained the OAuth2 token, the user is ready for using all the service APIs
- All the service APIs must include a “X-Auth-Token” header containing the OAuth2 token
- Upon request, the service API will take the token, will query the global Identity Manager (authenticating itself as well) and will receive an answer:
  - If an error is received then the token does not belong to any user; authentication failed
  - If the token belongs to a registered user in the IdM, the user is authenticated and his/her ID, email, etc. is returned
- All service APIs relate to /user/<id>/... resources
  - If the user ID given by the IdM matches the user ID in the resource, the user is authorized to use the resource
  - If not, the authorization is denied

Using the storage endpoint: WebHDFS API

- I/O operations with HDFS
- https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/WebHDFS.html
- liststatus example:

```
$ curl -X GET
"http://storage.cosmos.lab.fiware.org:14000/webhdfs/v1/u
ser/<idm_id>?op=liststatus&user.name=<idm_id>" -H "X-
Auth-Token: zRaMnZ5vaWTca5gET0Xlx9euYYCx2e”

{"FileStatuses":{"FileStatus":[{"pathSuffix":".Trash","t
ype":"DIRECTORY","length":0,"owner":”<idm_user>","group"
:”<idm_user>","permission":"700","accessTime":0,"modific
ationTime":1468519200094,"blockSize":0,"replication":0},
{"pathSuffix":”other","type":"DIRECTORY","length":0,"own
er":”<idm_user>","group":”<idm_user>","permission":"755"
,"accessTime":0,"modificationTime":1458036768315,"blockS
ize":0,"replication":0}]}}
```

Using the storage endpoint: WebHDFS API

- Authentication OK, authorization denied:

```
$ curl -X GET
"http://storage.cosmos.lab.fiware.org:14000/webhdfs/v1/u
ser/other?op=liststatus&user.name=<idm_id>" -H "X-Auth-
Token: zRaMnZ5vaWTca5gET0X1x9euYYCx2e”

Authorization error: user <idm_id> cannot access
/webhdfs/v1/user/other
```

- Authentication failed:

```
$ curl -X GET
"http://storage.cosmos.lab.fiware.org:14000/webhdfs/v1/u
ser/<idm_id>?op=liststatus&user.name=<idm_id>" -H "X-
Auth-Token: dummytoken”

Authentication error: {"error": {"message": "Access
Token dummytoken not found", "code": 404, "title": "Not
Found"}}
```

Using the computing endpoint: Tidoop API

- Running and managing MapReduce jobs
- http://telefonicaid.github.io/fiware-cosmos/api/
- Job submission:

Java .jar containing the job code, the main class and the parameters must be passed as a Json payload

```
$ curl -X POST
"http://computing.cosmos.lab.fiware.org:12000/tidoop/
v1/user/<idm_id>/jobs" -d '{"jar":"jars/hadoop-
mapreduce-
examples.jar","class_name":"wordcount","args":["hdfs:
//storage.cosmos.lab.fiware.org/user/<idm_id>/input",
"hdfs://storage.cosmos.lab.fiware.org/user<idm_id>/ou
tput"]}' -H "Content-Type: application/json" -H "X-
Auth-Token: zRaMnZ5vaWTca5gET0X1x9euYYCx2e"
```

A job ID is obtained

```
{"success":"true","job_id": "job_1460639183882_5635"}
```

Using the computing endpoint: Tidoop API

- Job status:

```
$ curl -X GET
"http://computing.cosmos.lab.fiware.org:12000/tidoop/v1/user/<i
dm_id>/jobs/job_1460639183882_5635" -H "Content-Type:
application/json" -H "X-Auth-Token:
zRaMnZ5vaWTca5gET0Xlx9euYYCx2e"

{"success":"true","job":{"job_id":"job_1460639183882_5635","sta
te":"SUCCEEDED","start_time":"1475483172262","user_id":”<idm_id
>","stderr":”...”,”stdout”:”...”}}
```

stderr and stdout are returned, for debugging purposes; future releases will parameterize this as optional

- Kill job:

```
$ curl -X DELETE
"http://computing.cosmos.lab.fiware.org:12000/tidoop/v1/user/<i
dm_id>/jobs/job_1460639183882_5635" -H "Content-Type:
application/json" -H "X-Auth-Token:
zRaMnZ5vaWTca5gET0Xlx9euYYCx2e”

{"success":"true”}
```

Thank you!

http://fiware.org
Follow @FIWARE on Twitter