ESCAPE WP5

Technical Work Review

Stelios Voutsinas
University of Edinburgh

Technology Review

- WP5 related tasks:
  - JupyterHub experiments
    - Documentation, Scripts and Configurations
    - Prototype Services being tested
  - Docker Containers for VO tools
    - Can be used to deploy custom JHub environments
    - Used to discover and access data in the Virtual Observatory
  - We've started working on contributions to the ESAP Prototype.
    - First task in progress:
      - Working on allowing keyword-based searches for metadata in Virtual Observatory Services

JupyterHub

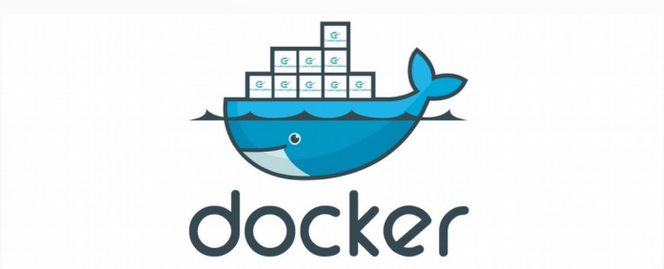
- Ongoing experiments and evaluation of running a Jupyterhub (JupyterLab) service in progress
- Service deployed on top of Openstack & Kubernetes
- Output of experiments may be in the form of:
  - Documentation
  - Running prototype service
  - Scripts & Configuration to deploy service on any cluster
    - (Openstack, AWS, GCLoud etc..)

JupyterHub + Kubernetes

- Allows us to create a reproducible service
- Fault tolerant (K8s pods restart automatically when unhealthy)
- Easy to run from an Ops perspective
  - Deploying new versions
  - Platform independent
  - Customizable using Configuration files
- Service deployed using Helm Charts

JupyterHub with Docker

- Our JupyterHub services deploy custom Docker images.
- Configuration allows either of the following:
  - Fetch from Docker Hub
  - Or from local Docker registry.
    - Local Docker Registry works if we wont private Docker images



JupyterHub: Authentication

- OAuth
  - Initial experiments work with Github Oauth
  - JHub with EGI-Checkin will also soon be available

- Todo: Work on integration with ESAP
  - Single sign-on
  - Propagation of tokens from ESAP to JHub
  - AuthN/AuthZ between ESAP and a JHub service

JupyterHub Persistent Storage

- Current Jhub Storage experiments with:
  - Openstack Cinder Volumes (block storage)
  - NFS on top of Cinder
- Other experiments with:
  - HDFS for persistent storage prototype in place
    - Data stored as Parquet files
  - Object-store (Openstack Swift) as an alternative also in the works

Notebooks & Containers

- Docker containers for Virtual Observatory
  - Provide set of tools and libs that help users discover, access and visualization data from VO services.
- Also Example notebooks for a few different use cases.
  - Examples for finding accessing and visualizing image or catalogue data
- Libraries used include:
  - Astropy (Astroquery), Pyvo, AladinLite (image visualization), plotly, bokeh (plotting)

Notebooks & Containers

Workflow for discovering, querying and visualizing astronomy data in the VO

In [ ]:
```
from astropy.coordinates import SkyCoord
from hips import WCSGeometry, make_sky_image
from hips import HipsSurveyProperties
from ipywidgets import Layout, Box, widgets
```

Data Discovery (VO)

In [ ]:
```
import pyvo
```

In [ ]:
```
from pyvo.registry import search as regsearch
```

Find all TAP Services with 'quasars'

In [ ]:
```
services = regsearch(keywords=['quasar'], servicetype='tap')
```

In [ ]:
```
print (services)
```

Find all TAP Services with keyword "ukidss"

In [ ]:
```
services = regsearch(keywords=['ukidss'], servicetype='tap')
print (services)
```

Data Access (TAP, Astropy)

In [ ]:
```
from pyvo.dal import tap
service = tap.TAPService(tap_url)

query_text = """
SELECT TOP 500
sourceID, ra,dec FROM
    lasSource
ORDER by dec
"""


from astroquery.utils.tap.core import TapPlus
service = TapPlus(url=tap_url)
job = service.launch_job(query_text)
table = job.get_results()
table
```

Image Access

HIPS Image

In [ ]:
```
url = 'http://surveys.roe.ac.uk/hips71/LAS/LAS_Y_OUT//properties'
hips_survey = 'CDS/P/DSS2/red'
```

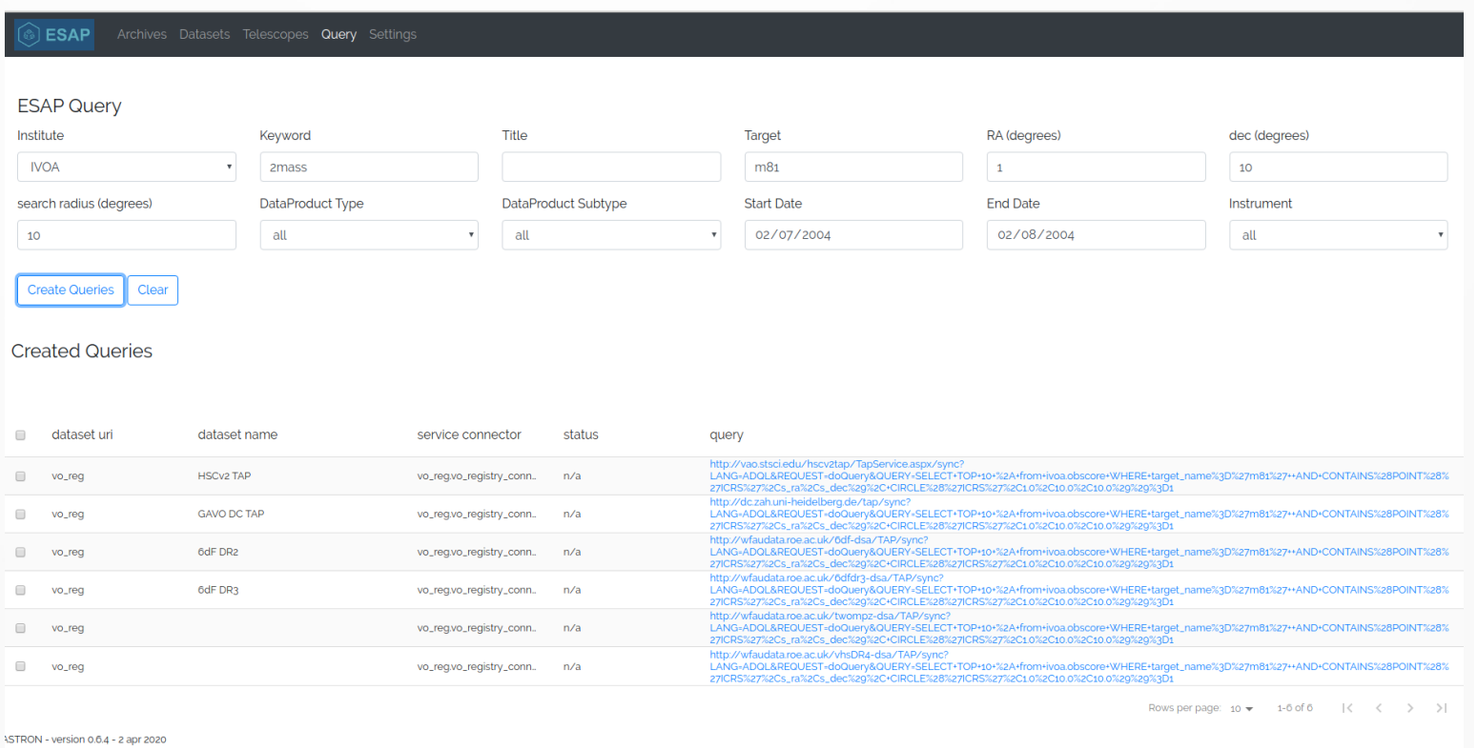
In [ ]:
```
print (url)
```

In [ ]:
```
hips_survey_property = HipsSurveyProperties.fetch(url)
```

In [ ]:
```
geometry = WCSGeometry.create(skydir=SkyCoord(217.9, -2.3, unit='deg', frame='fk5'),width=180,
height=180, fov='0.02 deg',coordsys='icrs', projection='TAN')
from hips import make_sky_image
result = make_sky_image(geometry, hips_survey_property, 'fits')
```

ESAP Prototype: VO Integration

- Plan to assist in the development of the Prototype
- Initial work in building a keyword based search for VO Services



Other Technologies used for our Science Platforms

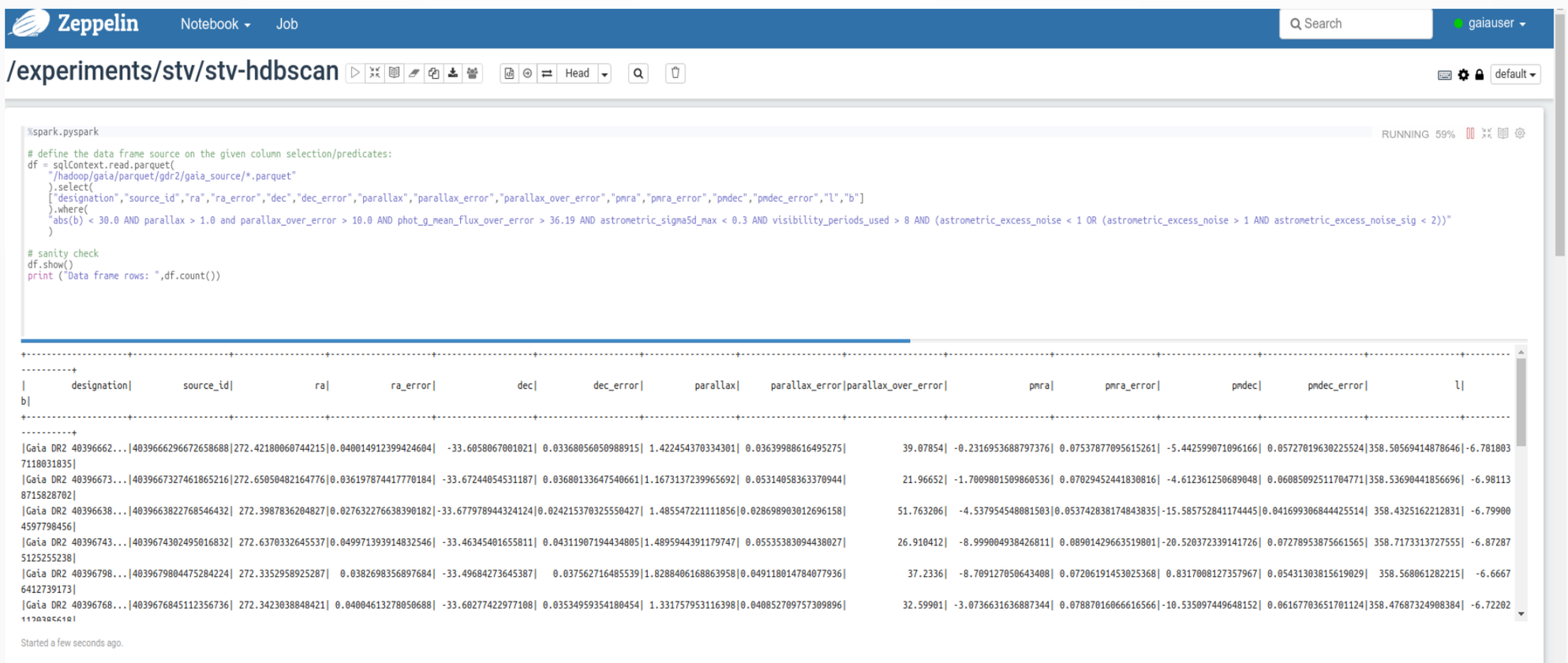
- Science Platforms being built for GAIA & LSST

- Apache Spark & Zeppelin
  - We’re running a service with Zeppelin as User interface with a Spark Cluster behind it.
  - Initial prototype built with Hadoop (HDFS/Yarn)
  - We’re currently looking into replacing the Hadoop components.
    - Plan is to run Spark on Kubernetes, with Object storage for persistent data

- Apache Kafka
  - We’re building a platform to receive alerts from LSST (& ZTF), process and annotate them, allowing users to create custom filters on them
  - The platform receives the alerts using Kafka, and we also use Kafka internally in our pipeline.
  - Allows us to create a scalable service, were we can add worker nodes to consume, process & produce these alerts, (horizontal scaling)

Other Technologies used for our Science Platforms

- Spark & Zeppelin

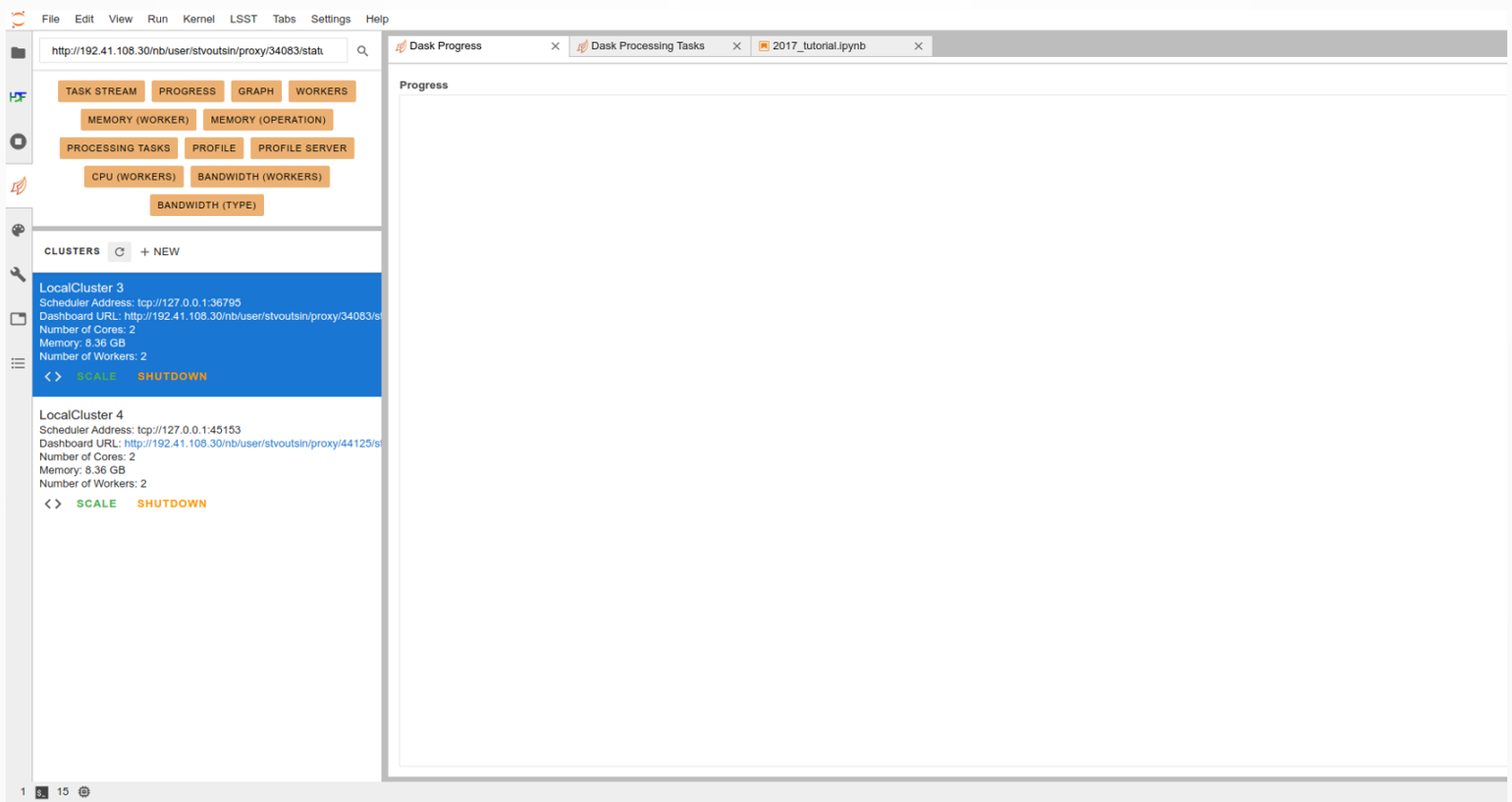


Other Technologies used for our Science Platforms

- Dask
  - Provided as part of one of our experimental JupyterHub services
  - Powerful and flexible tool for scaling Python analytics across a cluster.
  - Uses existing Python structures and APIs (pandas, numpy, etc..)
  - Works out-of-the-box with JupyterHub
    - JupyterLab extension

Other Technologies used for our Science Platforms

- Dask



Technology Review

- Questions?