GKE PanDA and Dask infrastructure recap

Fernando Barreiro

PanDA + Rucio setup

PanDA GKE **analysis** queue: GOOGLE100

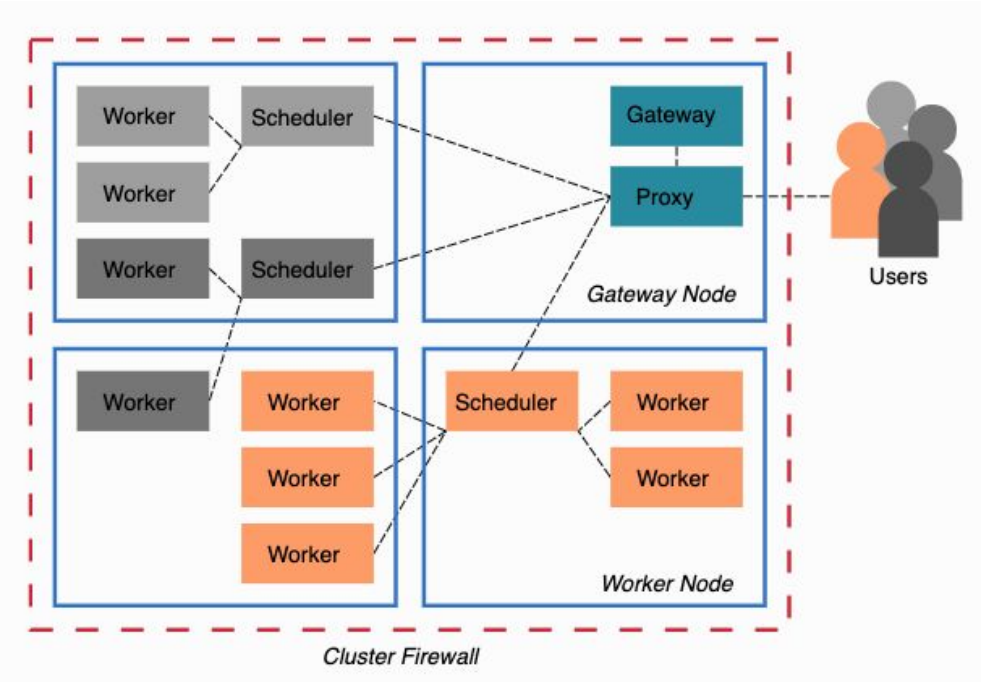
- Cluster details
  - 0-10 **autoscaled, preemptible** nodes: n2-standard-8 (=8 cores, 32GB RAM)
    - Scaling down under discussion with Usman and Jason Nichols (GKE specialist)
  - Local **SSD** at each node
- Queue status “Brokeroff”: you need to specifically set the queue when submitting to PanDA
- Data needs to be pre-placed to GOOGLE_EU
  - Jobs will stay in “**assigned**” if data not present
  - Requires **special permission/quota** in Rucio to interact with this storage element
- Queue was briefly validated. Data for “10% test” finished transferring today

ATLAS Site	PanDA Site	PanDA Queue	State	type	cap	rtype	Cloud	Tier	Final status	Manual
GOOGLE	GKE	GOOGLE100	ACTIVE	analysis	ucore	cloud	US	T3D	BROKEROFF	BROKEROFF
ATLAS Site	PanDA Site	PanDA Queue	State	type	cap	rtype	Cloud	Tier	Final status	Manual

DDMEndpoint	Type	Experiment Site	Activities
GOOGLE_EU	OS_CACHE	GOOGLE	read_lan/0, write_lan/0

Dask Gateway

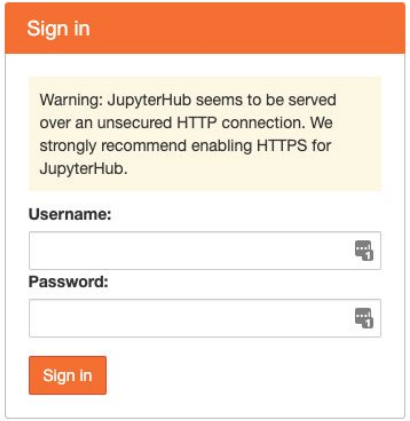
Dask Gateway



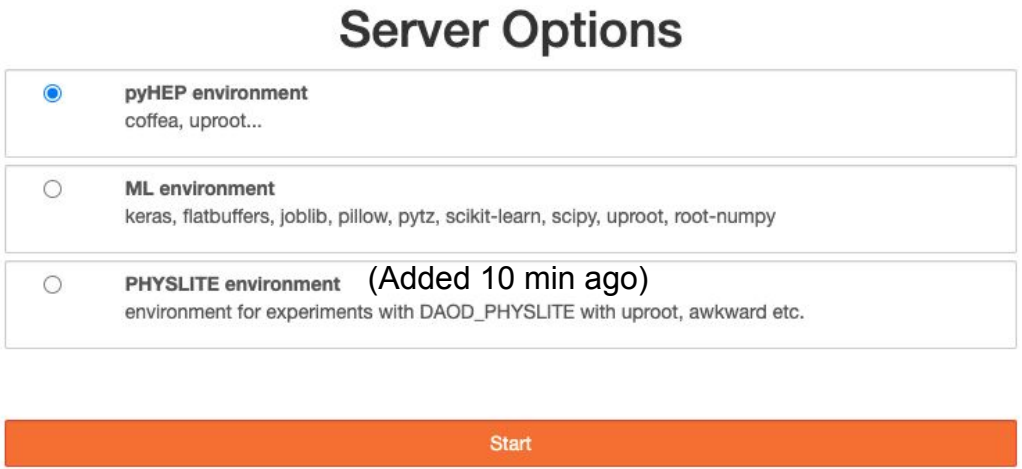
- Sets up common Dask cluster and JupyterHub for all users
- Users have access to JupyterHub and Dask, but not to GCP/GKE
- Disadvantage: less flexibility for individual customization. Needs central maintenance of a set of images that work for everybody
- Current installation on modest cluster: 3 e2-standard-8 nodes with 100GB disk

JupyterHub

http://jupyter.gcp4hep.org/



- Local accounts. I need to add new users
- Integration with other identity providers possible. Could be done only if there is real usage



- Available images
  - pyHEP environment: dependencies suggested in this tutorial
  - ML environment: dependencies requested by Fang-Ying
- Images hosted in GCP Container Registry
- Installing conda dependencies and uploading images is time consuming

Jupyter notebook specs

- Each user notebook runs on an **independent** pod with image selected at startup
  - "**Burstable**" QoS with 1GB RAM base request
  - How much you can burst depends on overall cluster usage and occupancy of the node
- User home directory: 10GB
  - Independent persistent disk
  - Value is configurable
  - Disk can also be manually extended
  - Anything outside the home directory gets **cleaned up** once notebook stops
    - A potential conda user environment installed on home directory would survive

```
(notebook) jovyan@jupyter-fbarreir:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay          95G   16G   79G  17% /
tmpfs            64M     0   64M   0% /dev
tmpfs            16G     0   16G   0% /sys/fs/cgroup
/dev/sdc         20G   46M   20G   1% /home/jovyan
cvmfs2          5.9G  2.7G  3.2G  46% /cvmfs/atlas.cern.ch
cvmfs2          5.9G  2.7G  3.2G  46% /cvmfs/atlas-condb.cern.ch
cvmfs2          5.9G  2.7G  3.2G  46% /cvmfs/atlas-nightlies.cern.ch
cvmfs2          5.9G  2.7G  3.2G  46% /cvmfs/grid.cern.ch
cvmfs2          5.9G  2.7G  3.2G  46% /cvmfs/sft.cern.ch
cvmfs2          5.9G  2.7G  3.2G  46% /cvmfs/sft-nightlies.cern.ch
cvmfs2          5.9G  2.7G  3.2G  46% /cvmfs/unpacked.cern.ch
/dev/sda1        95G   16G   79G  17% /etc/hosts
shm              64M     0   64M   0% /dev/shm
tmpfs            16G     0   16G   0% /proc/acpi
tmpfs            16G     0   16G   0% /proc/scsi
tmpfs            16G     0   16G   0% /sys/firmware
```

CVMFS

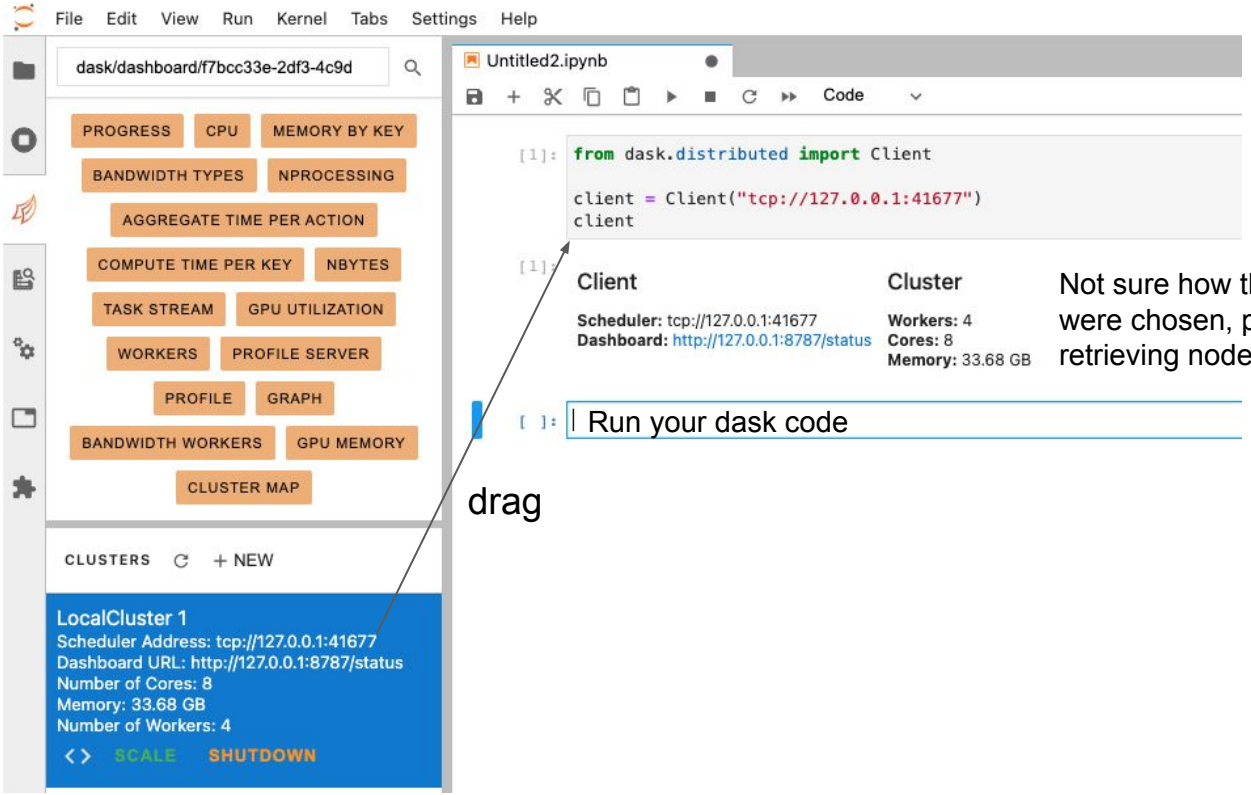
- Available from Jupyter session and Dask workers
- You can for example interact with Rucio to list file replicas and get signed URLs
  - Requires uploading your voms proxy to the notebook

```
(notebook) jovyan@jupyter-fbarreir:~$ echo $ATLAS_LOCAL_ROOT_BASE/
/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase/
(notebook) jovyan@jupyter-fbarreir:~$ source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh
lsetup                lsetup <tool1> [ <tool2> ...] (see lsetup -h):
 lsetup agis           ATLAS Grid Information System
 lsetup asetup         (or asetup) to setup an Athena release
 lsetup atlantis       Atlantis: event display
 lsetup eiclient       Event Index
 lsetup emi            EMI: grid middleware user interface
 lsetup ganga          Ganga: job definition and management client
 lsetup lcgenv         lcgenv: setup tools from cvmfs SFT repository
 lsetup panda          Panda: Production ANd Distributed Analysis
 lsetup pod            Proof-on-Demand (obsolete)
 lsetup pyami          pyAMI: ATLAS Metadata Interface python client
 lsetup root           ROOT data processing framework
 lsetup rucio          distributed data management system client
 lsetup views          Set up a full LCG release
 lsetup xcache         XRootD local proxy cache
 lsetup xrootd         XRootD data access
advancedTools         advanced tools menu
diagnostics           diagnostic tools menu
helpMe                more help
printMenu             show this menu
showVersions          show versions of installed software
```

```
(notebook) jovyan@jupyter-fbarreir:~$ lsetup rucio
bash: file: command not found
*****************************************************************************
Requested:  rucio ...
 Setting up emi 4.0.2-1 200423.fix3 ...
  Skipping: grid middleware already setup (from UI)
 Setting up rucio 1.25.3 ...
Info: Setting compatibility to centos7
 Setting up xrootd 5.1.1-x86_64-centos7 ...
bash: file: command not found
>>>>>>>>>>>>>>>>>>>>>>>>>>> Information for user <<<<<<<<<<<<<<<<<<<<<<<<<<<
 emi:
   Warning: current gcc version (gcc00) is older than  needed for emi (gcc48)
 emi:
   No valid proxy present.  Type "voms-proxy-init -voms atlas"
*****************************************************************************
(notebook) jovyan@jupyter-fbarreir:~$ rucio whoami
2021-05-19 14:55:54,098 ERROR   given client cert (/tmp/x509up_u1000) doesn't exist
2021-05-19 14:55:54,098 ERROR   Cannot authenticate.
Details: x509 authentication failed for account=fbarreir with identity={'client_proxy': '/tmp/x509up_u1000'}
2021-05-19 14:55:54,099 ERROR   Please verify that your proxy is still valid and renew it if needed.
```

Spinning up a Dask cluster: LOCAL

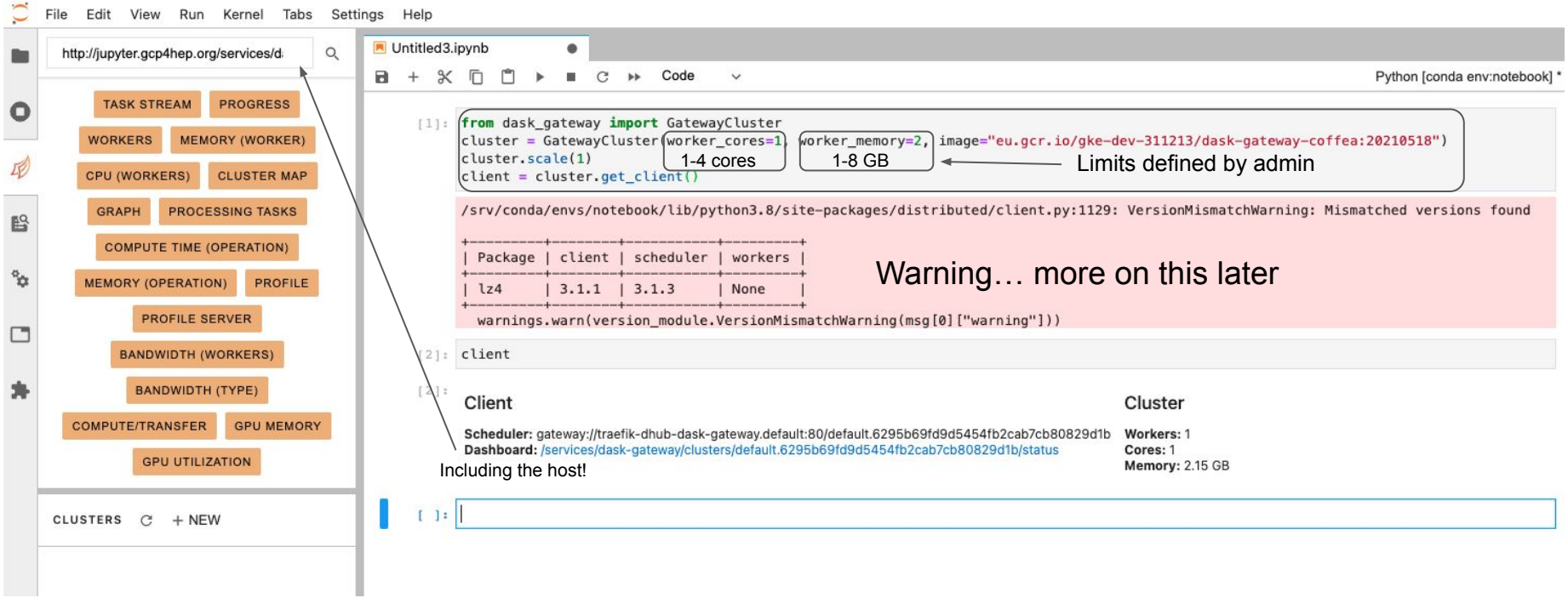
LOCAL: your cluster lives in your jupyter pod



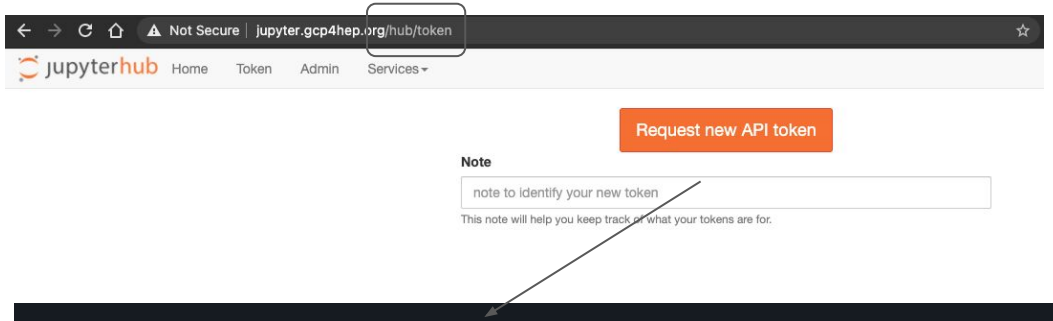
Not sure how these values were chosen, probably by retrieving node size

Spinning up a Dask cluster: distributed

Distributed: each worker gets an independent pod, so you can scale to multiple nodes



Spinning a dask cluster from outside Jupyter




```
export JUPYTERHUB_API_TOKEN=<YOUR TOKEN>
[user@machine gke-dask]# python3
Python 3.6.8 (default, Nov 16 2020, 16:55:22)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from dask_gateway import Gateway
>>> gateway = Gateway("http://dask.gcp4hep.org/services/dask-gateway", auth='jupyterhub')
>>> cluster = gateway.new_cluster(image='xxx/yyy:zzz')
>>> client = cluster.get_client()
>>> # RUN YOUR COMPUTATION
```

Software compatibility

- The client (Jupyter or your python shell) and the Dask workers need to have compatible SW (dask, tensorflow...) installed
- I tried to generate compatible images for both Jupyter images building on the default pangeo/daskgateway images

Jupyter image	Name	Worker image	Description
pangeo/base-notebook:2020.11.06	**Not selectable**	daskgateway/dask-gateway:0.9.0	Basic Dask installation. We have overwritten this option with our images
eu.gcr.io/gke-dev-311213/jupyter-coffea:20210518	pyHEP environment	eu.gcr.io/gke-dev-311213/dask-gateway-coffea:20210518	This image is based on the dependencies used in this PyHEP tutorial. It includes coffea, python-graphviz, mimesis on top of the default pangeo image.
eu.gcr.io/gke-dev-311213/jupyter-ml:20210518	ML environment	eu.gcr.io/gke-dev-311213/dask-gateway-ml:20210518	Image based on Fang-Ying's request which includes root, keras, flatbuffers, joblib, pillow, pytz, scikit-learn, scipy, energyflow, root-numpy, sklearn, awkward, uproot on top of the default pangeo image.
eu.gcr.io/gke-dev-311213/jupyter-physlite:20210526	PHYSLITE environment	eu.gcr.io/gke-dev-311213/dask-gateway-physlite:20210526	Image by Nikolai including PHYSLITE SW (numpy h5py numba uproot awkward pyarrow coffea aiohttp) .

https://github.com/gcp4hep/analysis-cluster/wiki/Daskhub-images

Git repository includes wiki with documentation

github.com/gcp4hep/analysis-cluster/wiki/Daskhub-usage

Search or jump to... / Pull requests Issues Marketplace Explore

gcp4hep / analysis-cluster

Watch 9 Star 1 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Daskhub usage

Edit New Page

Fernando Barreiro edited this page 6 days ago · 9 revisions

Below you will find information on how to use our Dask Gateway instance, including basic snippets to get you started. You can either use it from JupyterHub or directly through python.

JupyterHub

Our JupyterHub instance is available under http://jupyter.gcp4hep.org. You need a local JupyterHub user account in order to access it.

Each notebook runs on an independent pod. When starting up JupyterHub, the user will be given a selection of available images.

Each user is assigned a 10GB disk for his personal files. This disk is persistent and mounted to the user's home directory. Anything outside the home directory is cleaned up when the notebook is stopped.

Basic creation and shutdown of a Dask cluster

The following figure illustrates a basic example.

1. You connect to the gateway, create a cluster and get the client. You have to scale the cluster in order to start up a worker. to the required size.

```
from dask_gateway import GatewayCluster
cluster = GatewayCluster(worker_cores=1, worker_memory=2, image="xxx/yyy:zzz")
cluster.scale(1)
client = cluster.get_client()
```

> 📝 Instructions on the available images are in the image section.

Pages 5

Home

DaskHub pages

Installation

Images

Usage

CVMFS

Clone this wiki locally

https://github.com/gcp4hep/ana

Conclusions

- Infrastructure is ready to be used and required features implemented
  - Desirable Dask features (https, oAuth) can be implemented depending on evolution of activity
  - Data management in Dask to be explored further
- Validation steps under preparation
  - “10% PanDA” test
  - “1% Dask” test
- Paul has setup a separate cluster with Dask Helm (single user)
  - This model can be more appropriate for a potential PanDA integration
  - Still requires more experience and a dedicated discussion