ESCAPE

European Science Cluster of Astronomy & Particle physics ESFRI research Infrastructures

km3py

Pipelines, KM3NeT data access and provenance tracking

Tamas Gal

E-OSSR Onboarding Presentation

2021-03-05

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

Introduction/Instructions

Aim:
tech reports (~20 min talk and 2-3 page summary documents) on community software for OSSR

- Content:
  - science case and ”user story” (two sides: data analyst side and OSSR side)
  - added value of OSSR
  - update on questions from OSSR's first questionnaire and software registration survey
    - Both replies will be provided before the talk by FG1 lead
  - Discussion on on-boarding: open points, requirements…

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

Introduction

- **KM3NeT dataformats**
  - Internal formats:
    - custom binary formats for DAQ communication
    - ROOT format to store
      - raw hit data
      - intermediate files in processing chains (calibration, reconstruction, monitoring)
      - high-level data (reconstructed events, summary files)
    - HDF5 conversions available for a subset of data structures. Mainly used in
      - machine learning
      - high-level analysis
  - Open data includes:
    - ROOT (reconstructed events and also hit level data)
    - HDF5 and FITS (reconstructed events and summary files)

Funded by the European Union's Horizon 2020 - Grant N° 824064

Introduction

- KM3NeT
  - Data access, micro-services and internal pipeline management.
- **km3py**
  - A Python meta-package (**pip install km3py**) including among others:
    - **km3astro**: bridge to AstroPy, mainly KM3NeT specific coordinate transformations (detector UTM -> sky) and plotting helpers
    - **km3io**: native Python package to access KM3NeT data formats based on CERN/ROOT
    - **km3pipe**: general purpose pipeline framework with KM3NeT related modules, I/O helpers and provenance tracking
    - **km3services**: microservices prototype infrastructure
    - **openkm3**: Package to use KM3NeT open science products from the KM3NeT Open Data Center
- **thepipe**
  - Spin-off package originating from km3pipe, stripped down to the pipeline feature and provenance tracking

Funded by the European Union's Horizon 2020 - Grant N° 824064

Introduction

- **km3astro**
  - Bridge to the **AstroPy** software stack
  - Coordinate transformations of local events to sky coordinates

```
from astropy.units import deg
import numpy as np
import pandas as pd

from km3astro.random import random_date, random_azimuth, random_zenith
from km3astro.coord import local_frame, Sun, source_to_neutrino_direction
```


Funded by the European Union's
Horizon 2020 - Grant N° 824064

Introduction

- **km3astro**

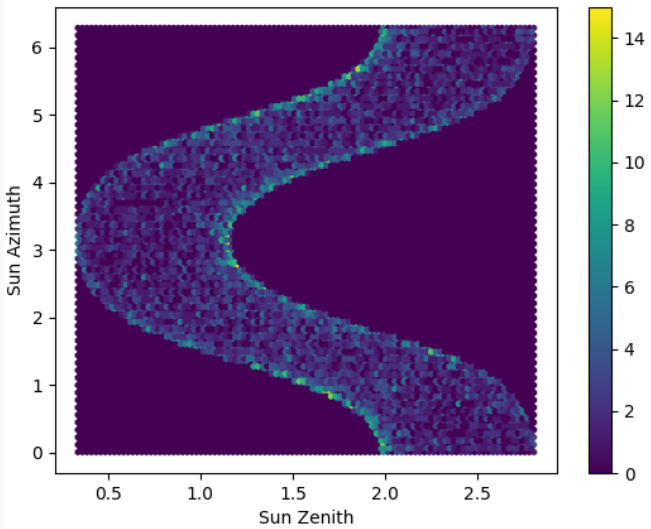
generate some random events

```
n_evts = 1e4
zen = random_zenith(n=n_evts)
time = random_date(n=n_evts)
azi = random_azimuth(n=n_evts)
```

transform to horizontal coordinates

```
orca_frame = local_frame(time=time, location="orca")
sun = Sun(time)

sun_orca = sun.transform_to(orca_frame)

sun_azi = sun_orca.az.rad
sun_zen = (90 * deg - sun_orca.alt).rad

sun_phi, sun_theta = source_to_neutrino_direction(sun_azi, sun_zen)

sun_df = pd.DataFrame(
    {
        "Sun Azimuth": sun_azi,
        "Sun Zenith": sun_zen,
        "Sun Cos Zenith": np.cos(sun_zen),
        "Sun Phi": sun_phi,
        "Sun Theta": sun_theta,
        "Sun Cos Theta": np.cos(sun_theta),
    }
)
```

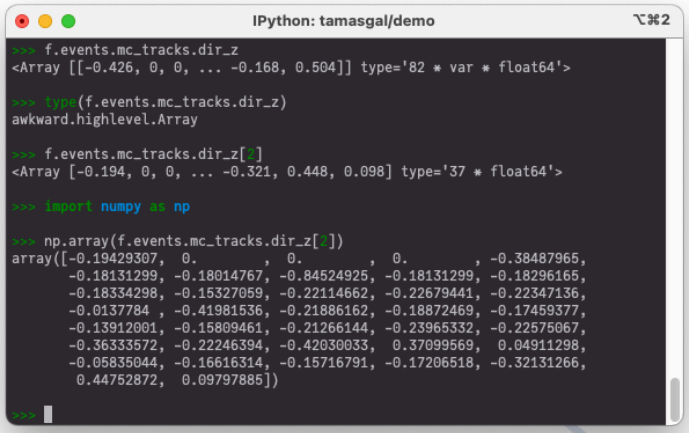
```
sun_df.plot.hexbin("Sun Zenith", "Sun Azimuth", cmap="viridis")
```



Funded by the European Union's Horizon 2020 - Grant N° 824064

Introduction

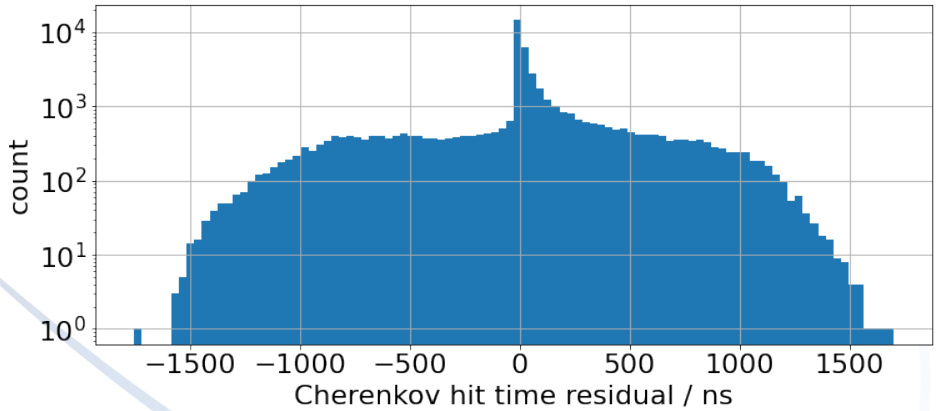
- **km3io**
  - Direct access to our official ROOT formats (part of our open data)
  - Python-only dependency (based on uproot https://uproot.readthedocs.io)
  - Offers a high-level, self-descriptive interface



Funded by the European Union's Horizon 2020 - Grant N° 824064

Introduction

- **km3pipe / thepipe**
  - Pipeline management
  - Intra-process provenance tracking
  - Lot of extra functionality related to internal KM3NeT data formats and services
- The pipeline and provenance functionalities are available as a standalone Python project **thepipe**
- **Live demo**



Funded by the European Union's Horizon 2020 - Grant N° 824064

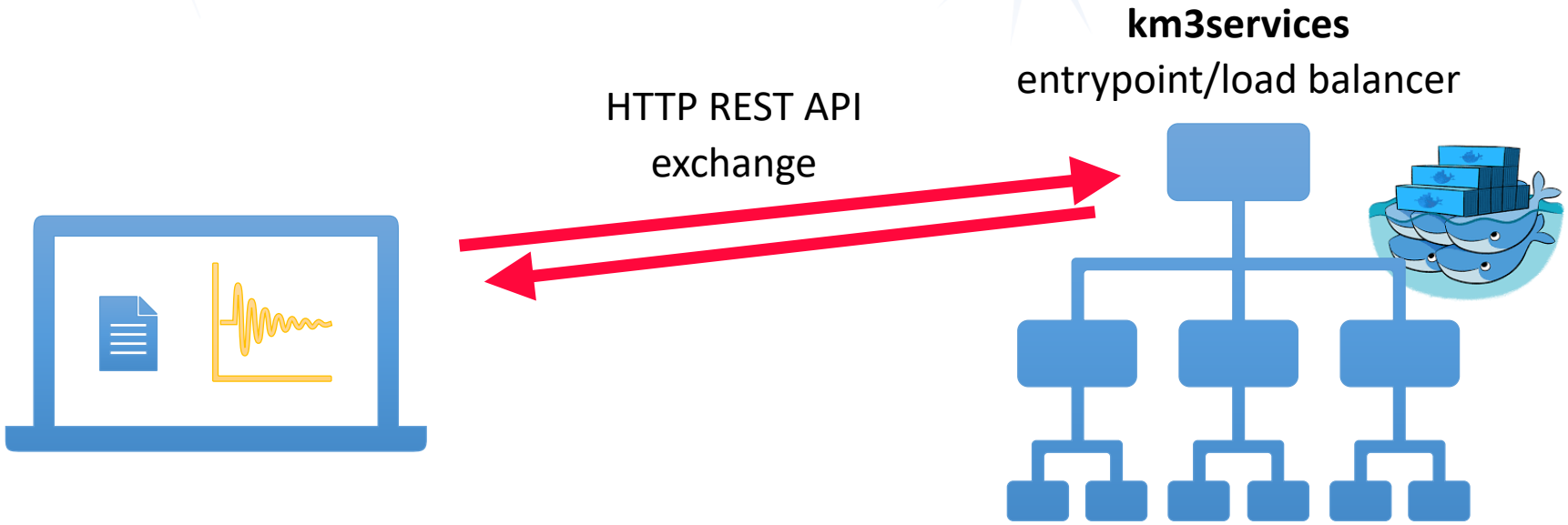
Introduction

- **km3services**
  - Micro-services running as Docker containers in a Docker swarm, hosted at KM3NeT computing infrastructures
  - Ability to run each service also locally (local Docker instance)
  - REST API for data transfer
  - Service candidates under discussion (instrument response function, visibility, provenance database...)

Funded by the European Union's Horizon 2020 - Grant N° 824064

Introduction

- **km3services**



The implementation is hidden and the service **feels like a regular Python package**

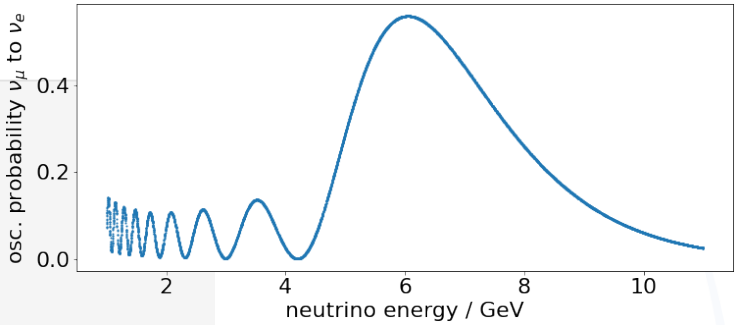
parallel instances of services to distribute load

Funded by the European Union's Horizon 2020 - Grant N° 824064

Introduction

- **km3services – live demo**
  - Calculating neutrino oscillation probabilities using OscProb (https://github.com/joaoabcoelho/OscProb)
  - Docker image running with OscProb and all dependencies (ROOT, Eigen, ...) on a KM3NeT Docker swarm

```python
from km3services.oscprob import oscillationprobabilities

import numpy as np
import particle

nu_in = particle.Particle.from_string("nu(mu)")
nu_out = particle.Particle.from_string("nu(e)")

n = 10000
energies = np.random.rand(n) * 10 + 1
cos_zenith = -0.8
p = oscillationprobabilities(nu_in.pdgid, nu_out.pdgid, energies, cos_zenith)
```



Funded by the European Union's Horizon 2020 - Grant N° 824064

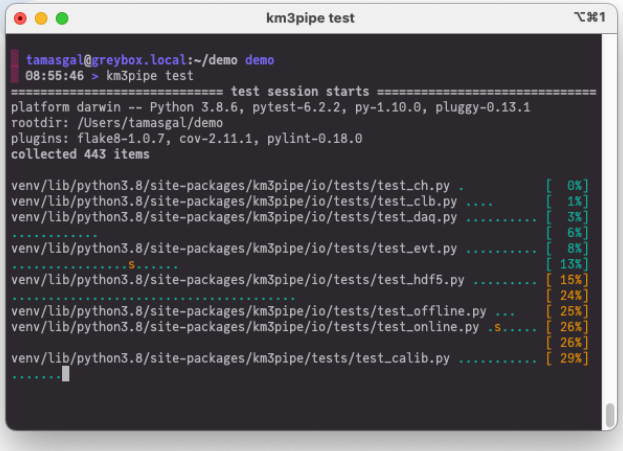
Introduction

- **openkm3**
  - Package for use of KM3NeT open science products from the KM3NeT Open Data Center
  - uses numpy, pandas and pyvo as service packages to interpret the various data fromats
  - **pip install git+https://git.km3net.de/open-data/openkm3**

Funded by the European Union's Horizon 2020 - Grant N° 824064

Software/Service Development

- Development on self-hosted GitLab
  - https://git.km3net.de/km3py/km3pipe
  - https://git.km3net.de/km3py/km3io
  - https://git.km3net.de/km3py/km3astro
  - https://git.km3net.de/km3py/km3services
- Spin-off package **thepipe** hosted on GitHub https://github.com/tamasgal/thepipe
- Merge requests, SemVer 2.0 for versioning, code reviews, automatic release of Docker images to the docker.km3net.de registry and Python packages to the Python Package Index (PyPI)
- Autogenerated documentation using **Sphinx**, hosted also on the same GitLab instance, **black** for formatting
- Unit tests and high-level tests including benchmarks
- MIT licence



Funded by the European Union's Horizon 2020 - Grant N° 824064

Software/Service Requirements

- Python 3
- LLVM compiler v11+ for km3pipe/km3io
- Hardware requirements
  - no special requirements
- Containerisation and portability requirements
  - Docker and Singularity

Funded by the European Union's
Horizon 2020 - Grant N° 824064

OSSR Integration

- Source code, Docker images, Singularity images
- Test data and full test suites
- Example notebooks and tutorials

- What is the ”user story” of a EOSC user taking on the software/service?
  - Human who wants to access KM3NeT data and services
    - **pip install km3py** or **docker run -it docker.km3net.de/km3pipe** - providing instant access to all important KM3NeT tools and services interfaced to popular libraries (NumPy, Pandas, HDF5, FITS)
    - Micro-services wrapping additional software accessible from any Python3 environment (locally via Docker or through WAN, both over HTTP REST API), suitable for JupyterHub installations
  - Human who wants to access KM3NeT related data from the Open Data Center (ODC)
    - **pip install** git+https://git.km3net.de/open-data/openkm3
  - Human who wants to wrap any kind of analysis into a modular pipeline based on Python
    - **pip install thepipe**
    - Wrap existing functions and classes into **thepipe.Modules**
  - Human wants to access and use different software which require isolated containers and use them together in an analysis
    - **pip install km3services**
    - Use the software as they were regular Python packages

Funded by the European Union’s Horizon 2020 - Grant N° 824064

TOC of Tech Report

- Introduction
  - ESFRI/RI and Partner, Science Case
  - Software and Service Name
- Software/Service Development Strategy
- Software/Service Requirements
- OSSR Integration
  - Status
  - Content
  - User Story

Funded by the European Union's
Horizon 2020 - Grant N° 824064