FIREfly
A Prototype Flight Test Data Server

October 2020

Mike Folk
Interim CEO
The HDF Group
A test data lifecycle

- Test prep
- Test run
- Quick look analysis
- In depth analysis
- Long term
A test data lifecycle – the data

- Test prep
  - Test plan

- Test run
  - Test results

- Quick look analysis
  - Quick-look results

- In depth analysis
  - Previous results
    - Derived metadata
    - Data products

- Long term
  - Previous results
    - Derived metadata
  - Data products
  - More products
A test data lifecycle – the people

- Test prep
- Test run
- Quick look analysis
- In depth analysis
- Long term
This photo by an unknown author is licensed under CC BY.
Air Force current state

...much of this data resides in repositories that

• are isolated from each other
• with limited data search and compute capabilities within them
• and limited data access between them

This in turn

• impedes data discovery
• slows down data analysis turnaround times
• impedes sharing of results
• and constrains larger scale data analyses
Objectives – Develop an...

Open-source web-based data server

That provides a simple, robust capability to

• Ingest datasets from various formats,
• Perform queries to find datasets of interest,
• View data of interest through web user interface,
• Return subsets of the data for client-side analysis,
• Upload Python algorithms for analysis

And be hostable in a Cloud environment
The Air Force asked for a solution that would address these issues based on HSDS.

- Why HSDS?
- HDF5
Our team

- Gerd Heber, John Readey, Aleksandar Jelenak, Mike Folk (The HDF Group)
- Bob Baggerman (Independent flight test consultant)
- Chris Colbert (SCC Software)
- Jeff Corn (Chief, Test Techniques Development Flight, U.S. Air Force)
- Other Air Force folks
Proposed proof of concept:
Data server with four core functions

INGEST - ingest scientific datasets, especially in Ch10 and HDF5 formats
FILTER - perform SQL-like data queries to find datasets of interest
REDUCE - view datasets of interest using a Web-based user interface, and upload Python algorithms for server-side data analysis
EXTRACT - return subsets of datasets to the client in formats convenient for client-side analysis
Proposed proof of concept: Data server with four core functions

FILTER
INGEST
REDUCE
EXTRACT
What existing open source components?

- **HSDS** - share data between widely distributed users
- **JupyterLab** - interactive computing via remote applications
- **Docker containers** - deploy without traditional VM overhead
- **Kubernetes** - deploy/monitor containers across the cluster
- **Py106** - python package for irig 106 library
Packages developed for this project

- firefly-py python package for data query, analysis and visualization
- Python 106 extensions for video format 0
- HSDS query enhancements
- FIREfly GUI query capability
- Chapter 10 ingest watchdog
- Chapter 10-to-HDF5 converter
- Synthetic Chapter 10 file generator
Some screenshots
Search by tail number returns four flights

```
[15]: fc = ffly_repo.filter(tail='ED020020')
    fc

[15]: FlightCollection with 4 flight(s) (repo: /FIREfly/h5/) at 0x7f68c74e1668

Showing all FIREfly files in the collection:

[16]: fc[:]

[16]: ['/FIREfly/h5/F-35-ED020020-200407041137.h5',
    '/FIREfly/h5/F-35-ED020020-201207030537.h5',
    '/FIREfly/h5/F-35-ED020020-201211220741.h5',
    '/FIREfly/h5/F-35-ED020020-201504221805.h5']
```
Find flights with altitude > 5000 feet and speed > or equal to 320 knots.

It is possible to combine several parameters. In addition to altitude greater than 5000 feet, the aircraft speed must be greater or equal to 320 knots:

```python
[19]: fc = ffly_repo.filter(altitude=(5000, None), speed=[320, None])
```

```python
[19]: <FlightCollection with 1804 flight(s) (repo: /FIREfly/h5/) at 0x7f68c73cdd30>
```
For a given flight find takeoff and landing airports computed from metadata

**Estimated Takeoff and Landing Airports**

[15]: `ffly.takeoff`

[15]: 'Eglin AFB (Eglin Main and Reservation), Florida'

[16]: `ffly.landing`

[16]: 'Eglin AFB (Eglin Main and Reservation), Florida'
For a given flight write out global metadata from HDF5 file
Display “quickview” graphs of flight's main parameters

[19]: ffly.quickview('/derived/aircraft_ins')

Variable: speed

Variable: altitude
Test drive at KitaLab: [https://www.hdfgroup.org/hdfkitalab/](https://www.hdfgroup.org/hdfkitalab/)
Login to KitaLab and find the directory /firefly

Python API: FlightSegment Class

The FlightSegment class represents data about one flight, or a part, i.e. segment, of one flight. The FlightSegment data always comes from a single FIREfly HDF5 file generated by ingesting one Chapter 10 file.

This notebook demonstrates the features of the FlightSegment class.

```python
from firefly import FlightSegment
from firefly.irig106 import PacketType
```

Opening for data access
Next steps
Next steps

Us

• Take FIREfly to the next stage
• Apply to other domains

You

• Talk to us if interested
• Have a look
  • KitaLab: https://www.hdfgroup.org/hdfkitalab/
  • Code: https://github.com/AkadioInc/firefly
Thank you

Questions and comments
Acknowledgement/disclaimer

Acknowledgment: This material is based upon work supported by the United States Air Force under Contract No. FA9302-19-P-1021.

Disclaimer: Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.