PyCon India 2018

Hyderabad International Convention Center

Conference Notes | October 06 - 07, 2018

Introduction

PyCon India celebrated its 10th iteration this year, at Hyderabad International Convention Center, uniting Pythonistas from all around the world, professionals and students alike.

The conference was held from 6th of October and 7th of October, while the workshops were conducted on the 5th and Dev-sprints on the 8th and 9th of October. The convention also hosted many sponsors (DEShaw & Co., Google Developers, IBM, vmware to name a few), developer quizzes, mini-hackathons, Open Space discussions, and “Women in Tech” talks.

The conference had keynotes from :

- **Armin Ronacher -** *Creator of Flask | Software Engineer, Sentry.io*
- **Carol Willing -** *Former Director, PSF | Core Developer, CPython & Project Jupyter*
- **Travis E. Oliphant -** *Founder, Quansight | Director, Anaconda Inc.*
- **Sidu Ponnappa -** *Data + People, GO-JEK*

Conference Notes

The individual talks in the conference were kept as introductory as possible for everyone to understand, each having a running time of about 20-25 minutes.

The notes from the conference were curated using Google Keep, the contents of which can be found here:

1. Armin Ronacher | keynote (Rust, Sentry.io)

- **Py**: 3rd most loved language on StackOverflow (Rust #1); Top most wanted language

- **Rust** : complex and hard semantically but very package-friendly, easy dependency inclusion (part of language itself vs pip and py are still separate despite py 3.5+ auto-package-inclusion), and distribution-easy

- Rust: backward compatible - old and new code can coexist

- **SQLAlchemy** (Py library) is beautiful but at the same time frustrating to integrate at times

- Documentation and Values of Rust are more detail-oriented

- Wider community vs Core Developers - huge number of developers but there is a gap between the developers working on Python's core development and coders who use Python and libraries

- **virtualenv** was embraced by Python community

- Distribution isn't a corollary in Python

- **PEP8** enforcement isn't strict

- Communication across teams can be conflicting at times, e.g. mailing list communication to fix an issue

- py_modules and package.json for package dependencies?

- performance can be increased manifold if UTF-8 could be non-indexable in Python language

- cabi for ffi?

- strip stdlib down?

- multiversioning imports

- web assembly module (**wasm**) for Python in browsers

- packaging and distribution should be a top-priority

- multithreading because of **GIL** inclusion has to be central to the language; complex to isolate it

2. Large Scale Web Crawling

- Intent: 50-100 websites

- Goal: save thousands of URLs per site, totaling to millions

- Problems:

- aggressiveness

(solution: **ProxyRotator)**

- creating your own Proxy: **Linode** support, conf file updating

- timeout

- errors

- URL deduping: (solution: circular hashing)

- URL traps

- Blocking

3. Satellite Image Processing

- Synthetic Aperature Radar (SAR) image

- Remote sensing

- Active (RADAR) send radiation and record, operating in Microwave spectrum of light; Passive (Radiometer) record emitted/reflected radiation, operate in visible spectrum

- Raw data, Radiometric correction of data, Removal of geometric distortions, Making Px sizes uniform, Projection info, Geometric Corrected Product, ...

- **GDAL**

- LEE Filter to smoothen dark images

- signal sent: in phase (cos p), received: out of phase (cos 90-p = sin p)

4. Geospatial Analysis

- Example data provider: **ISRO**

- challenges: quality of data

- sentinel (Euro space orgs)

- Libraries

- **GDAL**, GeoPandas, PyDEM (digital elevation model), GeoJSON

- **DEM**: CartoDEM: data in form of tiles (images)

- GDAL merge

- K-means clustering

- **Flood-fill** algorithm: used on cell-based grids (e.g. Minesweeper; Bucket-fill in paint) and modified floodfill

Proceeding with analysis of water flow

- starting point: known water body

- on elevation data basis: predict water flow

5. Building a Language Model (Text Analysis)

- Probability Distribution over sentences

- auto-completion e.g. I am attending ..... PyCon, not ice cream

- Applications

  - speech recognition - auto completion - auto reply

- Problems arise in Language Variability, ambiguity

- Statistical Language model, Neural net

  1. Get corpus - should be a big dataset

  2. preprocess

  3. tokenize: convert longer sentences/phrases into smaller pieces

     - NLTK library has functions to tokenize

  4. compute probability

     - most frequent bigrams, trigrams, ...

     - Probability, P = n(m+h)/n(h) = higher degree grams/lower degree grams

  5. generate phrases

N-gram model

- create N-grams from corpus

  - 1. calculate probability of occurrence based on history for each word in corpus and select word with highest Probability

  - 2. Coverage might be low

- threshold is calculated to make a better lang model

Libraries

- NLTK library - MLE library - contextual analysis - ScikitLearn

6. Video processing | Python parallelism

Presentation slides

- **FFMPEG**

- piping numpy arrays into FFMPEG

- using GPUs for faster partitioned processing

7. Travis Olliphant | keynote (Libraries, dev-motivation)

- **DASK** is similar to Spark

- hvPlot visualizations

- LIGO : Gravitational Waves

- GuPy/NumPy, Chainer - ML library

- xnd, uarray to unify array-based libraries (numpy, pandas, TensorFlow etc)

- openteams.io

- A possible way to start developing: write test cases for code pushed on github

- Example debate in Python's development: to describe memory - dtype vs ctype

8. Multi dimensional Data visualization - Dipanjan Sarkar (Intel)

Presentation slides

- Library plotnine → data → mtcars

- Note: plot 4D visualization as a 3D visual using facets

Visualising unstructured data

- word embeddings, language semantics

Visualising Audio data

- Urban Sounds Dataset

9. MIDI Parsing - Learning Guitar using Py

Presenters from: [Systango → FretZealot]

- Libraries: MIDO, Music21 to parse MIDI files (note, time/ticks data), scale detection

- Numpy, Scipy

- note-number, time, velocity

```
from mido import MidiFile
```

- Interpretation of MIDI

  - notes are represented as numbers

10. Cleaning data with Python

Project on Data Cleaning library - Git: gramener/dataaudit

Library: Tabula

> Clean the Data Structure

1. Scrape doc into DataFrame

2. Delete extra rows

3. Fix col-struct

4. Handle missing values

5. Standardize values - remove: name titles, misspelling, punctuation, extra chars

Fingerprinting (assigning one thing to similarly representing objects e.g. *soundex* - similar sounding words) using **jellyfish** library (uses Fuzzy-matching).

Custom fingerprinting can be used to further process values depending on the requirements

11. Building better microservices with GRPC

- REST APIs

  - HTTP/ 1.1 is slower because it's over TCP and usually have a connection overload; header overload

  - overheads usually overcome by pipelining

- gRPC: RPC as if the procedure call was a local one

- faster than RPC because it uses HTTP/2.0
- minimises overloads, caches the packet info, optimises connection, uses PB (faster)

Example:

1. Define a function

2. Service definition using ProtocolBuffer (serialization methodology like JSON)

```
service Calc{
     rpc SquareRoot(num) returns Number {}
}
```

3. configure client stubs - write methods at client

4. configure server stubs - write methods at server

Disadvantages

- no browser support
- insufficient documentation

Conclusion

- gRPC works great with Server-Server communication, while REST works great with Server-Client (more web-based applications) communications

12. Alexa enabled Smart Home programming in Python

Amazon Developers defines the following **Skills** to develop Alexa apps

- service
- interface
- actual intelligence resides here
- utterance (spoken phrase)

Libraries:

> flask-ask: Flask extension having wrappers for intent and constructs for Alexa skills API

> fauxmo: emulates RPi as Wemo device

> Ngrok

Third party Data - management: Echosim.io

13. Carol Wellick | keynote (PSF)

- Addressed the past, present and the future of Python, Pythonistas

- Focussed on making the dev-environment positive in the open-source communities

Lightning Talks (duration - 5 min.)

★ **EEG Controlled Rover – A Brain-Computer Interface**

★ **At the Eye of the Flood** ***(Kerala Floods)*** **-** Leveraging Open Source technologies towards Humanitarian causes

- Poliastro Astrodynamics

  - pure Py accelerated with numba
  - Trajectory plotting in 2D, 3D
  - computation of Near Earth Objects

- SOROCO - Python: “Batteries included”

  - concealing source code
  - Static compilation (Cython)
  - Compiled Py Bytecode - .pyce on-disk format

- Python 3.7 Cool Tricks

- Crushing highscores in Minesweeper using simple if-else cases