Towards Fitting as a Service with pyhf

Matthew Feickert
(Ben Galewsky, Lukas Heinrich, Ricardo Rocha, Sinclert Pérez, Giordon Stark)

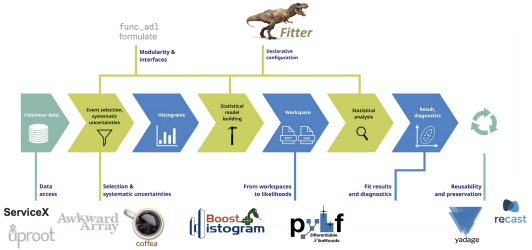
University of Illinois at Urbana-Champaign

Future Analysis Systems and Facilities Blueprint Workshop

October 26th, 2020

Fitting as a Service with pyhf

- Analysis Systems pipeline already has beta infrastructure for the final stages with pyhf + cabinetry
  - c.f. Alex's talk from earlier today
  - Ask if you have questions on these projects
- Want to leverage pyhf hardware accelerated backends at HPC sites for real analysis speedup
  - Fitting time from hours to minutes
- HTC not target, so deploy (fitting) Function as a Service (FaaS)
  - Use API to deploy fits and return JSON output




```
$ cat benchmarks/gpu/gpu_pytorch.txt
# time pyhf cls --backend pytorch HVTWZ_3500.json

{
    "CLs_exp": [
        0.07676925199218336,
        0.17262542362618583,
        0.3572332455085822,
        0.6318728762727417,
        0.879799718293609
    ],
    "CLs_obs": 0.25670190274923205
}

real    1m42.284s
user    13m6.493s
sys 0m4.742s
```

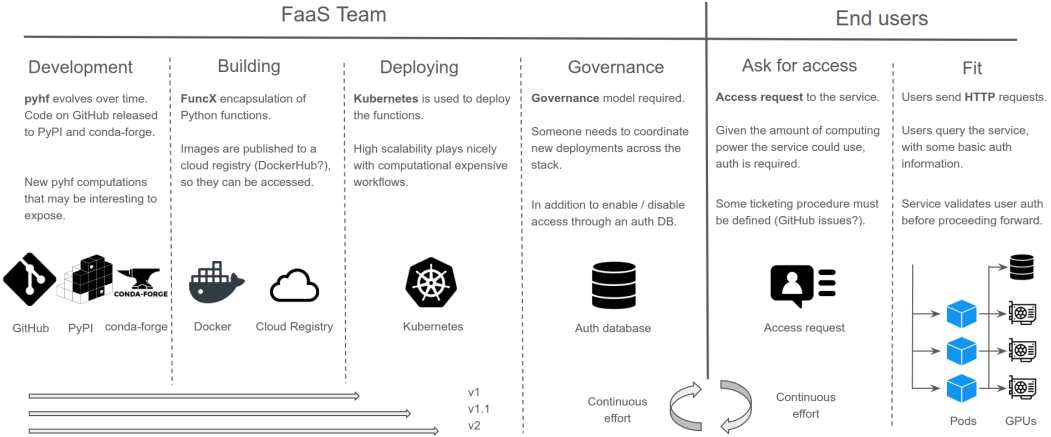
Workspace that takes over an hour on ROOT fit in under 2 minutes with pyhf on GPU

Open fields of exploration

- Early days in exploring solutions to implementation of Fitting as a Service
- Parallel explorations of what service and user API would look like
- funcX from Globus Labs
  - High-performance FaaS platform
  - Allows users to register and then execute Python functions in "serverless supercomputing" workflow
- Knative
  - Well adopted as a Serverless/FaaS solution on Kubernetes
  - Deployment model promotes efficient resource usage and simplifies bursting

Infrastructure Perspective

- **Sinclert** has outlined possible workflow for development (here for funcX) and end user experience



funcX

- **Ben** has built prototype workflow for fitting models from pyhf pallet for published ATLAS SUSY 1Lbb analysis
  - Currently deployed on Chicago River HPC cluster
  - Example implementation of Sinclert's model
- Uses Python driver with globus for authentication
- **Lukas** and **Matthew** have tested and are able to fit all models in analysis (125 signal patches) in just under 2 minutes 30 seconds
  - **N.B.** Wall time includes downloading pyhf pallet from HEPData, starting funcX, sending data to funcX, and fits
  - Currently CPU, but parallelization gives significant speedup
- For working prototype, this is already a win!

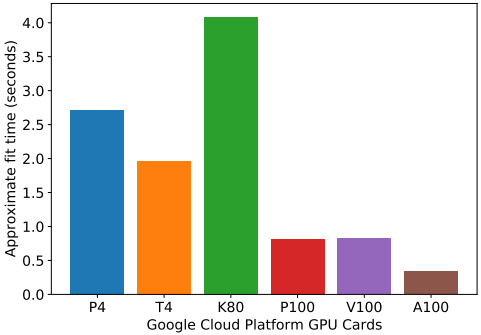
```
$ time python demo_fit.py
prepare  waiting-for-nodes
--------------------
<pyhf.workspace.Workspace object at 0x7efbd9d95530>
Task C1N2_Wh_hbb_1000_0 complete, there are 0 results now
Task C1N2_Wh_hbb_1000_100 complete, there are 1 results now
Task C1N2_Wh_hbb_1000_150 complete, there are 2 results now
Task C1N2_Wh_hbb_1000_200 complete, there are 3 results now
Task C1N2_Wh_hbb_1000_250 complete, there are 4 results now
Task C1N2_Wh_hbb_1000_300 complete, there are 5 results now
Task C1N2_Wh_hbb_1000_350 complete, there are 6 results now
Task C1N2_Wh_hbb_1000_400 complete, there are 7 results now
Task C1N2_Wh_hbb_1000_50 complete, there are 8 results now
Task C1N2_Wh_hbb_150_0 complete, there are 9 results now
Task C1N2_Wh_hbb_165_35 complete, there are 10 results now
Task C1N2_Wh_hbb_175_0 complete, there are 11 results now
Task C1N2_Wh_hbb_175_25 complete, there are 12 results now
Task C1N2_Wh_hbb_190_60 complete, there are 13 results now
Task C1N2_Wh_hbb_200_0 complete, there are 14 results now
Task C1N2_Wh_hbb_200_25 complete, there are 15 results now
...
... skipping forward for space
...
Task C1N2_Wh_hbb_800_50 complete, there are 114 results now
Task C1N2_Wh_hbb_900_0 complete, there are 115 results now
Task C1N2_Wh_hbb_900_100 complete, there are 116 results now
Task C1N2_Wh_hbb_900_150 complete, there are 117 results now
Task C1N2_Wh_hbb_900_200 complete, there are 118 results now
running
Task C1N2_Wh_hbb_900_300 complete, there are 119 results now
Task C1N2_Wh_hbb_900_350 complete, there are 120 results now
Task C1N2_Wh_hbb_900_400 complete, there are 121 results now
Task C1N2_Wh_hbb_900_50 complete, there are 122 results now
Task C1N2_Wh_hbb_535_400 complete, there are 123 results now
Task C1N2_Wh_hbb_900_250 complete, there are 124 results now
--------------------
... skipping print of results

real    2m27.249s
user    0m12.273s
sys 0m2.052s
```

Knative + GPU Workloads

- **Ricardo** has built prototype scaling out from CERN to Google Cloud Platform (GCP)
  - Especially interesting for GPUs/TPUs
- Supports fast auto scaling of workloads (secs) and clusters (mins) to meet demand
- **Ricardo** and **Lukas** working on version that allows per second reporting instead of per script execution

```yaml
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
  name: autoscale-go
  namespace: default
spec:
  template:
    metadata:
      annotations:
        # Target 10 in-flight-requests per pod.
        autoscaling.knative.dev/target: "10"
    spec:
      containers:
      - image: rochaporto/fitting:cuda11.0
```



Example toy fit run on all available GCP cards

Summary

- Fitting as a Service with pyhf is natural progression of final stage of Analysis Systems pipeline deployed to HPC sites
  - Consumes pyhf Python API
  - Allows for parallelization of fitting models from pyhf pallets across HPC/GPU systems
  - Possible interface with cabinetry?
- Very early stage of development, but given relative stability of pyhf API should be able to develop and explore idea space quickly
  - pyhf not yet at v1.0, but relevant API is rather stable
- Working deployment on funcX, trial deployments at CERN with Knative
- End user API currently under design iteration
  - Small library to give service agnostic CLI API as well as Python API?
  - Scope will dictate more as projects evolve

Backup

Ricardo: GCP vs. Azure for single and double precision