



# Using funcX to enable better federated learning over serverless

Matt Baughman

Parsl & funcX Fest  
Sep. 13, 2022



Making things easy...

~~Computer science~~ is difficult...

~~Machine learning~~

Federated

Making things easy...

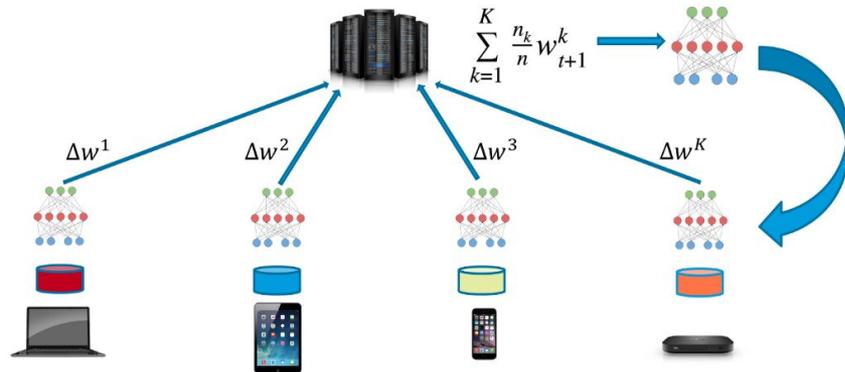


But it doesn't need to be...

# What is Federated Learning?

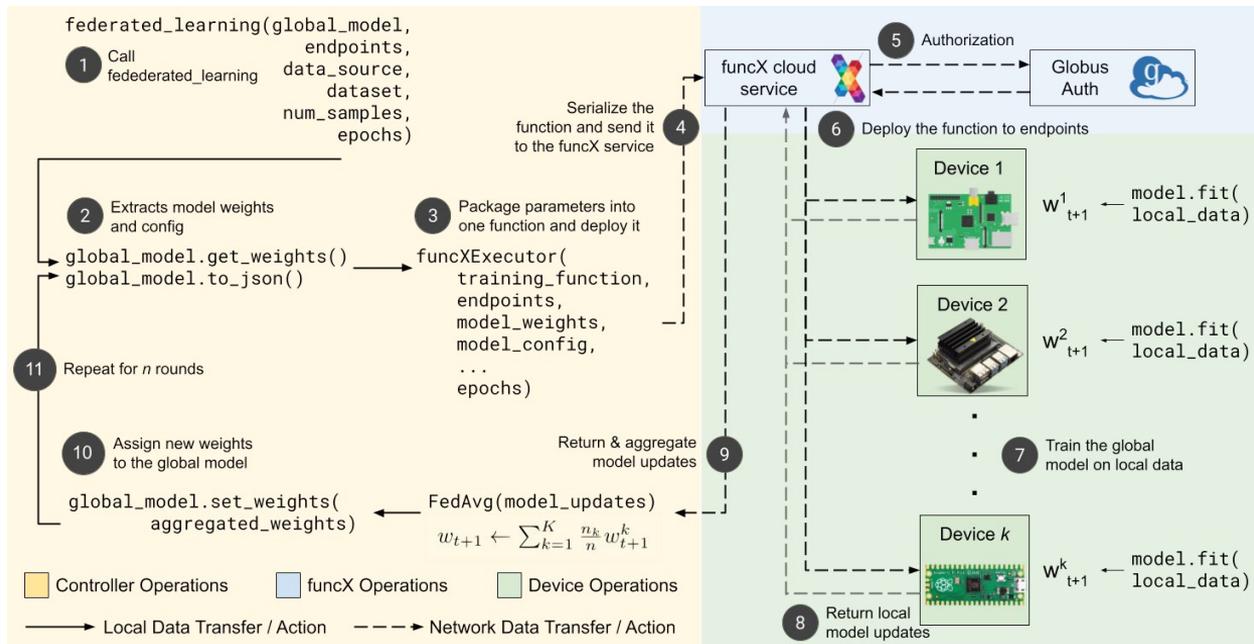
- ❖ Distributed data sources
- ❖ Model training at those sources
- ❖ Some aggregation mechanism
- ❖ Assists in security
- ❖ Examples
  - GBoard and Apple
  - Siloed data
  - Edge compute/network constrained
  - Healthcare
- ❖ Challenges
  - Network maintenance
  - Manual configuration
  - Hardware heterogeneity
  - Solution completeness

How does it work?



# Our solution: FLoX–Federated Learning on funcX

- ❖ funcX-ify relevant functions
  - I.e., those that could be relevant for deployment
- ❖ Launches relevant training and inference tasks across all endpoints
  - Manually configurable
  - Or pulled from .yaml
- ❖ Simply define a model, point to the data, and flox()





# What sets FLoX apart?

- ❖ First and foremost
  - Single line invocation
- ❖ Serverless
  - All the reasons funcX is great
  - Now for FL!
- ❖ End-to-end
  - Caveats...
- ❖ Use existing funcX endpoints
- ❖ Any tensorflow defined model
  - No model architecture constraints

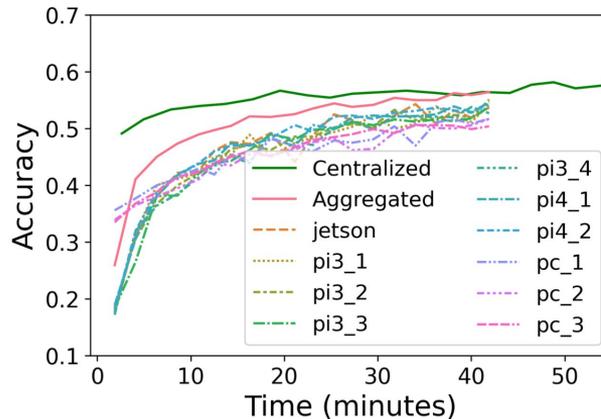
```
federated_learning(  
    global_model=tf_model,  
    endpoint_ids=[ep1, ep2, ep3],  
    loops=10,  
    epochs=10,  
    data_source="keras",  
    keras_dataset="mnist",  
    num_samples=1000  
)
```



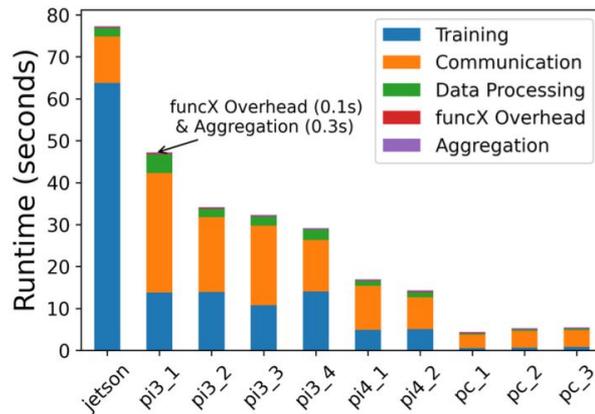
# Initial Experiments

- ❖ Performance
  - Comparable final accuracies to non-serverless frameworks
  - Faster convergence wrt time
  - Global model outperforms all individual models
  - Demonstrated effective even on small quantities of data
- ❖ System
  - Extremely low overhead
  - Demonstration of transfer costs
    - “Compute where the data is”

Sample FL accuracy over time



(c) Animals-10





# Using FLoX

1. `(flox) matt@matts-MacBook-Pro ~ % pip install pyflox`
2. `from flox.flox import federated_learning`
3. 

```
federated_learning(  
    tf_model,  
    endpoint_list,  
    loops=10,  
    epochs=10,  
    data_source,  
    dataset  
)
```
4. Wonder what to do with all the time you saved...

# Future Work

- ❖ More functionality!
  - Adding aggregation functions
  - Automated accelerator support
  - Better integration for finding/using diverse data
- ❖ Autotuning FL
  - Balancing workloads for device capability
  - Epochs vs loops vs data samples
    - Initial results
  - Fault tolerance and dropping devices
- ❖ Cost-aware FL
  - If data can be moved, should it?
  - Integration with DELTA
    - Our automated function placement system for funcX
- ❖ FL for system performance
  - Learn performance embeddings with FL across systems and functions

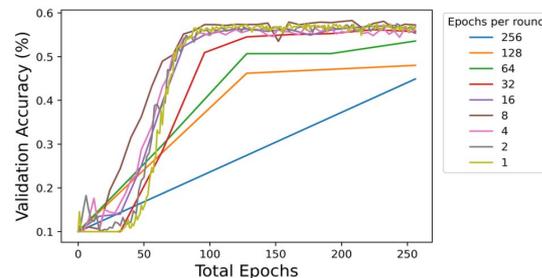


Figure 1: Comparison of different numbers of epochs per round for running FL on CIFAR-10.

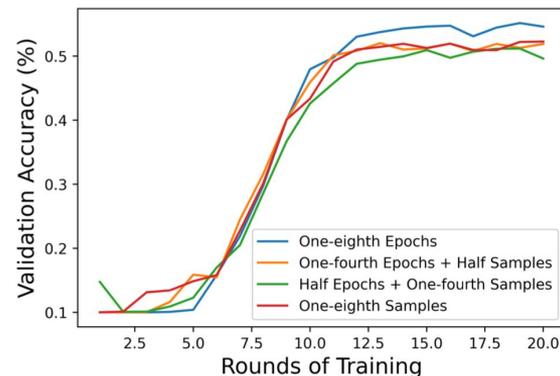


Figure 2: Comparison of balancing techniques to perform FL between two high powered endpoints and two additional endpoints with one-eighth the capabilities.



# Key Takeaways

- ❖ FLoX makes FL easy using funcX
  - Serverless has many advantages
  - funcX is the best of them
- ❖ No need to sacrifice performance for simplicity
- ❖ Makes FL accessible for FL or data experimentation

# Acknowledgements

## FLoX Contributors



Kyle Chard



Ian Foster



Ryan Chard



Panos Patros



Nikita Kotsehub



Tyler Groshong



Omer Rana



Nathaniel Hudson



# Relevant Publications

[1] **Kotesehub, N., Baughman, M.,** Chard, R., Hudson, N., Patros, P., Rana, O., Foster, I. & Chard, K. (2022). FLoX: Federated Learning with FaaS at the Edge. In 18th IEEE International Conference on e-Science (forthcoming).

[2] **Baughman, M.,** Foster, I. & Chard, K. (2022). Exploring Tradeoffs in Federated Learning on Serverless Computing Architectures. In 18th IEEE International Conference on e-Science: Posters (forthcoming).



# Questions?

Contact—[mbaughman@uchicago.edu](mailto:mbaughman@uchicago.edu)