Hosted python functions for humans.

Nicholas Petosa

Motivation

Python is the fastest growing major programming language [1]. Its unprecedented growth has been attributed to python's massive repository of scientific and statistical libraries such as scipy, pandas, numpy, keras, and tensorflow [2]. Python's machine learning libraries in particular enjoy wide usage and active development. Yet in many spheres of development, specifically mobile development which concerns either Java or Swift, python and its wildly popular libraries are inaccessible. The current way of integrating with python cross-platform is to create a Flask or Django web server to host some piece of functionality from a library. This requires resources to invest in learning a web framework, hosting, and application security. **The solution is skython, a web service for easily building, hosting, sharing, and executing python functions.** Skython is a web application that allows developers to write, test, and upload python functions to a collection in the cloud called skybox. Users can then query functions in the skybox from popular utilities like AJAX, requests, and postman. With skython, developers can seamlessly tie python scripts into any code with an HTTP request to the script's endpoint, meaning it can provide the entire backend for mobile or IoT devices. skython's simplicity makes it ideal for rapid prototyping and development – there are no frameworks to learn, just straight python. This solution can scale from weekend hackathon projects to serverless computation for companies. For this project, I will also be creating a website called *Polibias*, which, given a URL, will determine whether the site has a left-wing or right-wing bias, and justify its prediction with an explanation. There will be a live ticker of recent predictions on Polibias as well, updating real-time as other users make requests. The amazing part of Polibias is that *it won't have a backend* – skython will handle all tasks related to natural language processing, machine learning, web scraping, and real-time updates. Polibias will exist to demonstrate the power of both skython's architecture and the reusability of its constituent skython functions.

Related Work

skython

Functions as a service (FaaS)/Serverless code/NoOps is not a new concept [3]. There are several popular FaaS services which can host anonymous functions with support for many programming languages. The biggest players are AWS Lambda, Google Cloud Functions, and Microsoft Azure Functions [4]. Many of these services require developers to adhere to a code template, forcing new and cumbersome programming models on developers [5]. Additionally, these services tend to be overly centralized and tied closely to their proprietary software environments; for example, AWS Lambda is designed to integrate easily only with Amazon's product ecosystem. Where skython differs is in its openness and simplicity. skython is open source, meaning a developer or company can run a skython server locally for their internal development teams – for free. Developers can share functions in their skybox over LAN, completely isolated from the internet, meaning code is secure. There are no complicated code templates to fill out, and no files to upload from your computer – the script as you write it is exactly what will be executed.

There are open source alternatives to AWS Lambda and other proprietary FaaS products. The most popular is OpenFaaS, a serverless function framework which allows users to turn docker containers into queryable functions. OpenFaaS comes with a web UI for monitoring and testing functions.

Skython is distinct from OpenFaas because all skython function development occurs within the web UI. There is also no requirement to have a docker swarm, as all execution occurs within the skython service, greatly simplifying implementation.

The biggest way in which skython stands out from OpenFaas, AWS Lambda, and others is in its social aspect. In skython, functions can be published to skybox, making them available to all users connecting to that skython instance. Developers using the skython web UI can browse the source code of these python functions as they would browse repositories on Github. skython takes this social aspect one step further by allowing developers to call these community scripts directly from their projects – effectively crowdsourcing the NoOps movement. By sharing function source code, developers can reuse components of existing functionality to save development costs while building better code [6]. By enabling users to publish their functions, others can leverage them as building blocks for larger projects.

skython is distinct from python hosting services such as PythonAnywhere, OpenShift, or Bitnami. These platforms allow users to develop python on a remote machine and develop full stack applications remotely. They are not FaaS platforms, and have no repository of plug-and-play community contributed scripts like skython does.

Polibias

There are a prevalence of political fact-checking sites that exist to rate the accuracy and bias of politicians and political news sites. The most popular is Politifact, which assigns news stories and quotes from politicians a “truthfulness” rating. These ratings are assigned by pundits working at politifact [7]. Other similar services include http://isfake.news, which accepts a URL to check against a repository of known fake news sites.

These sites rely on humans to manually determine bias – which means these sites are only as reliable and neutral as their staffers. Indeed, there have been criticisms launched against Politifact for being biased against republicans [8]. Further, these sites cannot classify text or quotes on demand, and only classify sites and quotes that are deemed important enough.

There is a niche for a truly neutral bias detector which uses natural language processing and machine learning to classify text as left or right leaning. Polibias will be able to accept a website URL or user provided body of text as input, allowing for users that check for bias in news stories, tweets, Facebook posts, emails, or any other medium suspected of political bias.

Features of Proposed Work

- **A web UI for creating and debugging python functions.** The development environment should have proper syntax highlighting and be familiar to developers. Each function will have the ability to request parameters (provided by user via POST body) and to return a value (provided to user via HTTP response). These functions will have access to a wide array of pre-installed python libraries including pandas, numpy, scipy, tensorflow, keras, BeautifulSoup, and requests. Additionally, users can leverage the skython meta library, which enables them to call

simple store(key, value) and load(key) functions to persist data across invocations. Persistent data will be user AND function specific. Users can apply custom tags to their functions (“ML”, “scraper”, “NLP” for example) and then choose to publish their functions to the community or keep them private.

- **Endpoints for querying functions.** The point of hosting functions on skython is so that users can call them in the first place. Developers will be able to specify the names of endpoints that clients must query in order to run and retrieve the results of a function.
- **Skybox – a library of community-built skython functions.** Part of the appeal of skython is that developers can view and use scripts created by other users. On the left-hand side of the web UI will be skybox, a searchable listing of all public functions. Skybox is indexed by function name, description, and tags, so developers can easily query for specific functionality. Clicking on a listing will bring up that function’s published code and expected argument/return values. From here, a developer can clone a community function for personal use.
- **Polibias – a truly neutral bias detector.** To demonstrate the functionality and reusability of the skython architecture, the website Polibias will be built. The site accept a user provided URL or body of text, and then queries NLP and ML models living in skython to classify the source material as left or right leaning. This has been shown to be possible using a “bag of words” approach [9]. Polibias will justify its decision by generating claims like “90% of the time, when the phrase *anti-choice* is used, it is coming from a left leaning bias. That phrase appears 3 times in your text.” Finally, polibias will have a real-time ticker on its page displaying recent classifications.

System Architecture

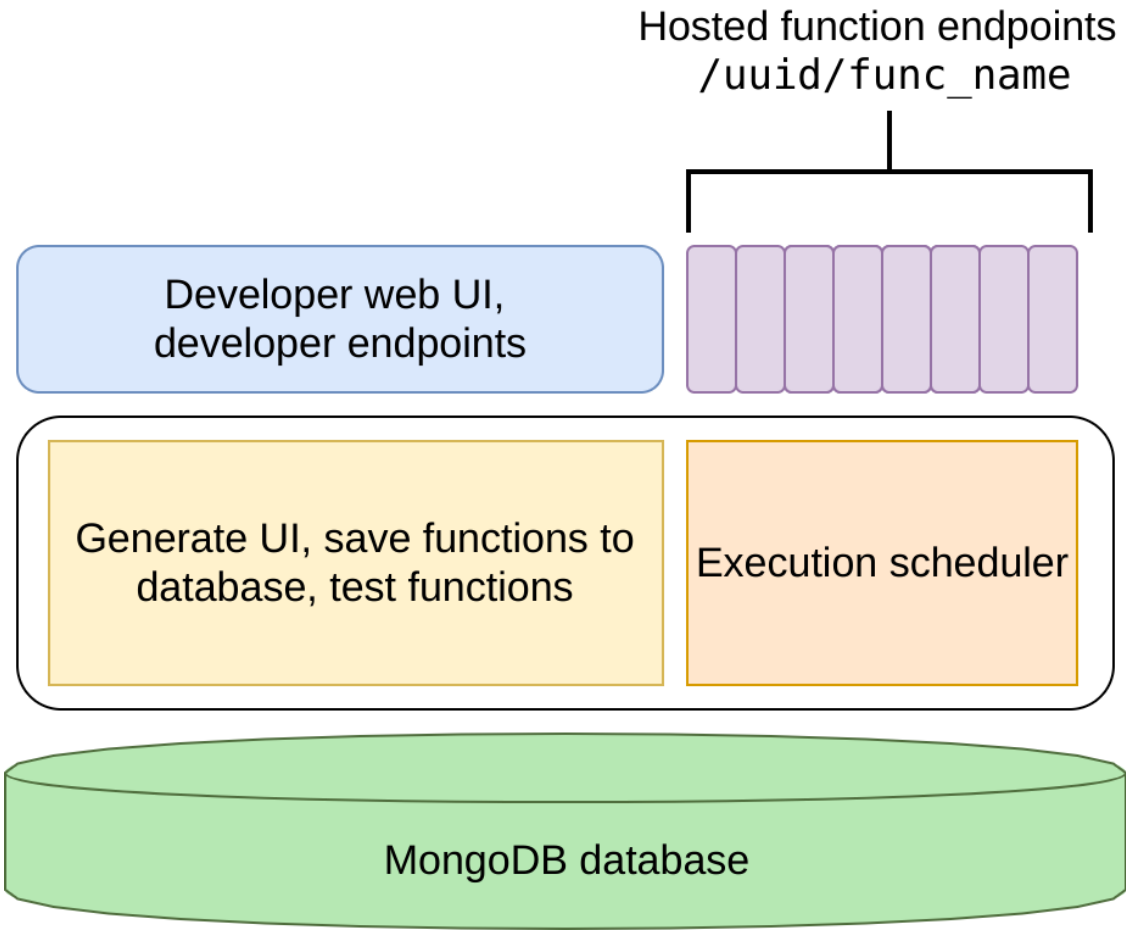
Central server and database. The official instance of skython will run on Google Cloud Platform's Compute Engine, although it can be hosted on any server box or cloud provider. An instance of MongoDB will be running on the same machine as well.

Interaction with developer clients. Developers will be interacting with skython's web UI in order to build new functions and upload them. The web UI will also be populated with community functions in skybox which the developer can query for.

Desktop, mobile, and IoT clients. Applications running on any other platform with the ability to make HTTP requests can query endpoints created by developers in order to run functions with custom parameters and receive a return value through an HTTP response body. Polibias will be one such client.

Application Architecture

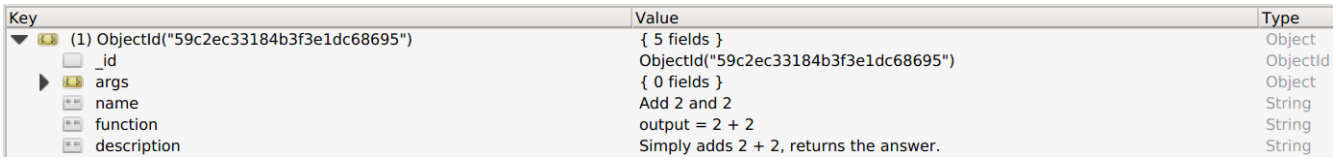
skython



Developer and client endpoints. At the highest level, skython exposes endpoints to developers and users. The web UI calls the developer endpoints needed for uploading functions and loading community functions from skybox. Client endpoints invoke functions created by developers.

Inside the skython application. Once a developer uploads a function or queries skybox, the internals of skython will create or read entries from the underlying MongoDB database. Once a user calls an endpoint, his requested function invocation is queued for execution on the server.

The underlying database. Skython is backed by a Mongo database, which is responsible for storing user data as well as function information (source code, arguments required…). The function objects stored in MongoDB will have the following schema:



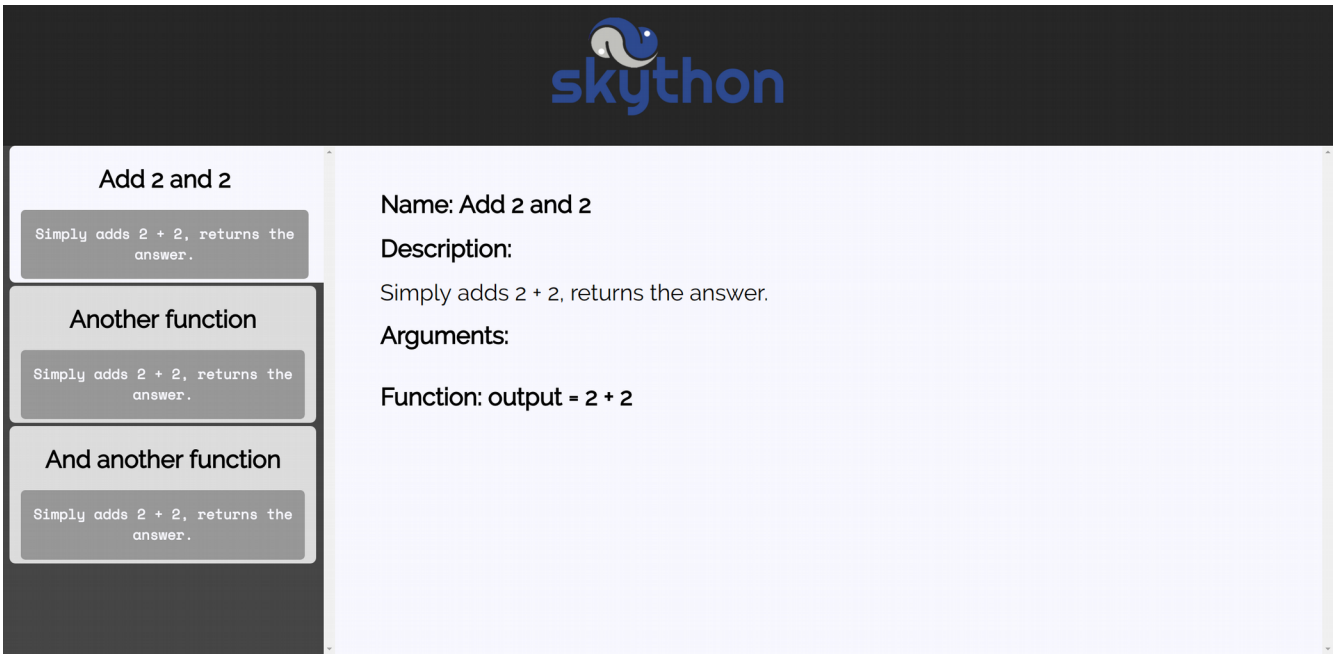
Key	Value	Type
(1) ObjectId("59c2ec33184b3f3e1dc68695")	{ 5 fields }	Object
_id	ObjectId("59c2ec33184b3f3e1dc68695")	ObjectId
args	{ 0 fields }	Object
name	Add 2 and 2	String
function	output = 2 + 2	String
description	Simply adds 2 + 2, returns the answer.	String

Polibias

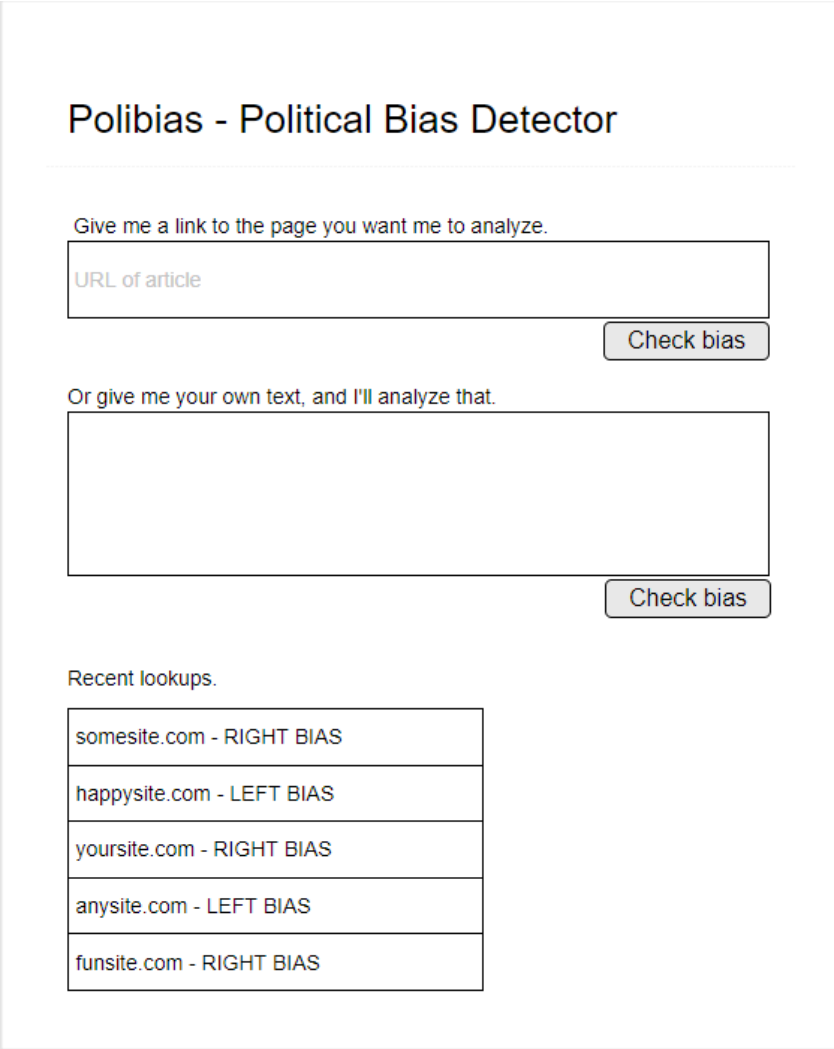
Polibias will have a basic web server architecture. It will be a python flask server which delivers the HTML, CSS, JS, and other static resources to the browser. The page itself will submit all of its requests for NLP, ML, and web scraping to skython.

Proposed Application Layout

skython



Polibias



Project Deliverables

- Working demo of skython, particularly showing how to create functions through the Web UI.
- Several proof-of-concept functions stored in skython which can execute tasks like NLP sentiment analysis and web scraping.
- Working demo of Polibias
- Source code for both skython and Polibias
- Documentation and Usage instructions
- Report

Foreseen Risks

- One risk that needs to be addressed in the design of this system is concurrency. Multiple users will be querying functions simultaneously, so the endpoints, scheduler, and database must be built in such a way to allow concurrent access.
- Security is a large concern. For example, certain modules needs to be disabled in python to prevent malicious developers from accessing and modifying server files. For example, the OS module needs to be banned.
- The classifier for Polibias will not achieve the desired accuracy.

Planned Schedule

	25-Sept	2-Oct	9-Oct	16-Oct	23-Oct	30-Oct	6-Nov	13-Nov	20-Nov	27-Nov
Create endpoints and their functionality										
Scheduler for function invocations, database										
Accounts										
Function persistent data										
Web UI										
Polibias										
Testing										
Documentation										

Future Work

- There is currently no way of gating who can run a developer's functions. This would mean allowing developers to distribute API keys for their function calls, which is currently out of scope.
- In order for any skython instance to be sustainable, it should be possible to monetize. This could mean introducing a method of charging developers for creation and usage of their functions, a practice used widely by API services.
- Further research needs to be done to prevent malicious executable code from running on the server. This would mean researching and banning python modules and practices which can modify and read server files without permission.

References

[1] https://stackoverflow.blog/2017/09/06/incredible-growth-python/
[2] https://medium.com/activewizards-machine-learning-company/top-15-python-libraries-for-data-science-in-in-2017-ab61b4f9b4a7
[3] http://searchaws.techtarget.com/essentialguide/AWS-Lambda-architecture-brings-serverless-to-enterprise-cloud
[4] http://searchcloudcomputing.techtarget.com/news/450296467/Public-cloud-vendors-jump-on-serverless-computing-bandwagon
[5] http://docs.aws.amazon.com/lambda/latest/dg/python-programming-model-handler-types.html
[6] https://quaid.fedorapeople.org/TOS/Practical_Open_Source_Software_Exploration/html/sn-Introduction_to_Free_and_Open_Source_Software-Source_Code_-_To_Share_or_Not_To_Share.html
[7] http://www.politifact.com/truth-o-meter/article/2013/nov/01/principles-politifact-punditfact-and-truth-o-meter/
[8] https://www.usnews.com/opinion/blogs/peter-roff/2013/05/28/study-finds-fact-checkers-biased-against-republicans
[9] https://nlp.stanford.edu/courses/cs224n/2009/fp/7.pdf