

Refactoring Piecewise - 2019/2020 Scope of Work

Background

[Piecewise](#) is a web-based tool for displaying aggregate data from M-Lab for particular geographic areas, along with a survey users must complete before running a test. Conducting a test via a Piecewise instance also asks the user for HTML5 geolocation, allowing specific locales to gather more accurate locations of users running speed tests. The application was originally developed by an M-Lab contractor as a proof of concept visualization and test integration to be used in a pilot with the city of Seattle, WA. Other organizations have built similar tools, for example [CIRA's IPT](#) and [Speedup America](#).

M-Lab has deployed Piecewise in several contexts and worked with partners who have contributed code and ideas for refactoring, notably and most recently, [MERIT's Michigan Moonshot project](#). These experiences deploying Piecewise and [recent development](#) in the production of basemap layers, as well as the growing interest in these tools, this document is intended to scope the changes M-Lab should pursue in a refactoring of this project.

Current System Design

At a high level, the [dataflow diagram on this page](#) describes the current Piecewise application. The deployed application is a Python [uwsgi](#) website with an [Nginx](#) proxy/load balancer, backed by a [Postgres](#) database with [PostGIS](#) extensions. The current application is deployed to an Ubuntu or Debian VM or server using Ansible.

[Deploying a Piecewise instance](#) currently entails:

- Setting up a VM or server, Gmail account setup (subscription to M-Lab Discuss)
- Obtaining shapefiles and making a topojson file for the geographies of interest
- Customizing values in an Ansible [vars file](#)
- [Deploying code with Ansible](#) and completing final setup on the VM or server

Proposed Changes

The following list of changes has been compiled from team discussion, past Piecewise deployments, and external developer contributions and feedback. Some potential changes and options are alternatives, of which we could support one or all, for example the ideas in "Overall Application Architecture" section.

Overall Application Architecture

- Consider making Piecewise an M-Lab hosted SAAS application

- Instances could be provided by subscription to cover costs
- Each instance managed by an admin, aggregate data from all instances is available in aggregate to all instances
- Make Piecewise a [Heroku](#) app, so anyone can deploy and manage their own instance.
- Build the application via Docker, allowing deployment to wherever.

Frontend

- Change the way an instance manages, sets, and updates survey questions.
- Replace current map and aggregation with two components
 - M-Lab aggregate data map using auto-generated basetiles
 - Re-use the work from USBB
 - Complete the daily automation of aggregate base tiles by different geographies
 - Main map will be M-Lab aggregate data & maybe other sources
 - Add a map layer with only locally collected data for that instance
- Automate the selection of GIS shape areas instead of making someone download them
 - Pull from the Census API?
 - Upload option
- Automate the creation of geojson for mapping/aggregating local test results
- Update to NDT7 client

Backend

- Switch from Ansible to Docker
- Use virtualenv, update to python 3, switch to webpack or yarn, migrate to bootstrap 4
- Add dependency and toolchain management
- Add better administrative access to the data collected

Proposed Development Stages

This section outlines suggested development projects to update Piecewise in two stages.

Stage 1

Develop feature updates to the Piecewise application, focusing on initial improvements to the code based on contributor and team feedback gathered over multiple past deployments, and recent mapping and aggregation improvements made by the M-Lab team in other initiatives.

M-Lab staff will manage the overall project, contracting with M-Lab advisory board member Georgia Bullen and will select a developer to complete the majority of the technical work. The end result will be code that can more easily provide a Piecewise server, survey and the NDT speed test, with configuration and administration functions happening post-deployment versus

pre-deployment. Additionally, implementing recent mapping improvements will make deployed instances easier to maintain and be less error prone.

Milestone 1: Update core Piecewise application components, toolchain, dependencies, and deployment model.

Background:

- Piecewise currently:
 - Uses Ansible to template files used by the frontend and the backend application, prior to deploying to an Ansible host target.
 - Uses Python 2 and Bootstrap 3.3.5.
 - Does not have good toolchain and dependency management.
 - Is configured pre-deployment, and doesn't have good support to update the application in place once it is deployed.

Tasks:

- Change from Ansible to Docker for deployment.
- Update the application to use virtualenv, Python 3, and migrate to bootstrap 4.
- Switch to webpack or yarn
- Add dependency and toolchain management using webpack and/or yarn
- Change to a post-deployment configuration model for the application, such that an admin will deploy the service via a Docker container, then login to the service to configure or update it.

Milestone 2: Update how Piecewise selects geographies to use in data aggregation and map display.

Background:

- Currently, when setting up a new Piecewise instance a user must:
 - Define bounding box coordinates for the geography of interest
 - Add shapefiles for the desired geographic aggregation
 - Create a geojson file from the shapefile to be used by the Leaflet frontend
- Once deployed, a daily cron job is created to ingest M-Lab NDT data submitted from within the geographic bounding box, followed by an aggregation process that compares the ingested M-Lab test rows with tests collected through the Piecewise server, and produces aggregate statistics in a Postgres database table using the local rows with more accurate geolocation where available.
- As a result of its United States of Broadband work, M-Lab now provides regularly updated base tiles which can be used in any application. These tilesets also include FCC form 477 data.

Tasks:

- Add OAuth2 and local login support, and administrative functions to configure, update, and manage the application in place.

- Support administrative functions to configure and manage the application:
 - Define geography of interest
 - Select sub-geographies to be used to aggregate locally collected data from the US Census API or equivalent
 - Provide the option to upload shapefile(s) to define sub-geographies for aggregating locally collected data
 - View and select supported M-Lab provided map tiles and aggregate layers
 - Define or update the survey questions and answers
 - Review and download locally collected test and survey data
- Automate the creation of geojson, etc. needed for the frontend mapping/aggregation of local test results
- Replace the Leaflet JS map with a Mapbox GL map using the pre-aggregated based tiles with general NDT data provided by M-Lab
- Support the selection of any M-Lab provided, pre-aggregated layers: FCC form 477 data, and M-Lab/FCC difference.
- Modify Piecewise's aggregation script to produce and update a second map layer using only the data collected from this Piecewise instance.

Timeline:

Completion by early December 2019

Budget:

<u>Invoice Item</u>	<u>Cost</u>
Contract Developer	\$7,000
Developer Advisement	\$2,000
M-Lab	\$1,000

Stage 2

Following the completion of Stage 1, develop feature updates to the Piecewise application, focusing on making it an M-Lab hosted Software as a Service application (SaaS), and/or to provide for individual deployments as a Heroku or other hosted application service.

Milestone 1:

- Update the speed test embedded within Piecewise to NDT7 JavaScript client
 - This should be a simple drop-in replacement for the current NDT JS test that Piecewise uses, with light frontend work for the test UI within the application.
- Build the application features to make Piecewise an M-Lab hosted a SaaS application available as a subscription service.
 - Administrative functions extended or modified to allow:
 - Users to signup and deploy a test instance with limited functionality.

- Users to contribute subscription cost via OpenCollective, and create their own fully featured instance.
- Use a sub-domain model for automatically hosting new instance(s)
- Provide for M-Lab administrative functions to
 - Allow M-Lab access to data collected by all instances.
 - Manage and support the SaaS application and instances.

Milestone 2:

- Enable a self-deployment option for Piecewise, for example as a Heroku app, so anyone can deploy and manage their own instance without needing to subscribe to the SaaS service.

Timeline:

- Milestones #1-3 - Jan 1 - Feb 28

Budget:

<u>Invoice Item</u>	<u>Cost</u>
1 contract developer for 120 hours	\$12,000
M-Lab management	\$3,000

Total: \$15,000