The problem:

Build a mini-Wikipedia site that can serve web and mobile-web clients under high-load, ~ 2K/sec and generate logs, metrics of page views, article dwell time, and a funnel-metric of visitors who clicked a “informative” button.

Approach:

Given the short amount of time and the available languages I chose Node.js and Express. The non-blocking asynchronous features of express should allow it to meet the request load demands with commodity hardware. I also knew I could both serve up the page and make a very basic metrics recording API using the same Express app, which should have saved time over having two separate page and metrics apps.

I used the built-in templating in express (jade) and added some CDN links to some basic CSS (pure.css). I used the direct CDN links both to save the hassle of setting it up locally and to reduce the load on the local server.

Logging / Tracking

I used a popular node.js logging package (Winston) and a basic javascript Analytics library I found (Ahoy) to setup logging both the server-requests and a couple basic user-analytics (ie clicking the “Informative” button. With more time I could have setup Google Analytics or another tool that would have done most of the heavy-lifting. Being not real familiar with those tools I thought the setup and learning curve would be too much.

I setup one logfile for the server requests and another for the user-analytics messages. The plan was to then use a python script to load and parse the logfiles and generate the necessary metrics and graphs from that.

Analysis

I planned to use a basic python script to combine and correlate the two logfiles using pandas dataframes, and then run aggregate functions to generate the desired metrics output and use a basic pandas **plot** to visualize the series.
I don't have a large Python background but given that Wikimedia seems to use Python based analysis tools such as Spark pretty heavily I thought I should try to create something that matched the Foundation’s tool chain. I was unable to complete the analysis script in time. I had a lot of trouble with parsing the log format into pandas DataFrames.

Retrospective

With more time I would have worked to put both the weblogs and the metrics into a streaming data solution such as Apache Spark. That would have provided much more out of the box support for analyzing the data and merging the request and application logs. As an alternative I could have setup and used Google Analytics to log both pageviews and the “Informative” metric and use it’s built-in tools to create the traffic and conversion metrics. Another option would have been to setup the app logger to use another datastore such as Redis.

Thank you for giving me the opportunity to interview for this position. I hope to have the chance to work with the Foundation in the future.

John Stewart