Abstract

We have tons of testing jobs for a push but we don't want to run each of them, as they are time consuming and requiring hardware resources. The SETA (Search for Extraneous Test Automation) is a tool to help us find out which test job is necessary for detecting the failures. To make it more maintainable and more reliable, we need to re-write SETA and we may want SETA to support Taskcluster as well.Furthermore, we want SETA to support Taskcluster now.

Contents

（already drop my presonal info in here）

- **Name:**
- **Email:**
- **IRC nick:** MikeLing
- **Twitter:**
- **Skype:**
- **Country of residence:** China
- **Timezone:** UTC+08:00

Project Proposal

- **Deploy SETA on Heroku**

Heroku is a platform as a service (PaaS) that enables developers to build and run applications entirely in the cloud. It also integrated with Github that making it easier to deploy from Github. Deploying SETA on Heroku can make it more maintainable and letting developers focusing on a better code quality.

Heroku uses PostgresSQL as its database, but we still need to keep mysql in the new SETA to do some importing and migration jobs there. We may need to keep mysql in the new SETA and do some import and migration jobs there. Because deploy on Heroku is very handly and the document is detailed. It will take approximately one week to deliver this part of job.

- **Add stage server for SETA**

To avoid breaking something accidently in the target server, a staging server is required for SETA to do pre-deployment validation as what has been done in treeherder. Therefore, We need to add a staging server for SETA to do pre-deployment validation like we do in treeherder. These changes should be applicated into the staging version of the database first and to see if anything breaks, before new tests are adding or updating.

- **Teach SETA about TaskCluster**

To make SETA support TaskCluster, we need to make it can accepted data from TaskCluster first. As far as I know, the 'ref_data_name' of TaskCluster job is job's signature.

However, it points to the test job's buildname if it's a buildbot job. So, we should make SETA accept data from TaskCluster in the right way. Of course, we need to address some other place for it. I need to get more familiar with the codebase before starting this part of code.

Due to the taskCluster schedule job by gecko decision task, So make SETA can influence gecko decision task is one of the most important things to do. We need to investigate a mechanism to analysis data job from TaskCluster and generate a task graph or json meta data as return.

- **Other ideals**

In the SETA re-write doc, we have ideals about creating SETA data per test information instead of per job information. Maybe we could make it like treeherder panel? For example, user could check the tests and job details when clicking job icon on the panel. In SETA, for now, we only display the job names. I think we could make the job name become clickable, and a panel will show up that containing each test's information.

Schedule of Deliverables

- April 22 - May 23:(about four weeks) Learn more about codebase, and discussions with the mentor and community on its interpretation and how it might best be implemented in Heroku.

- May 23 - May 30:(one week) Deploy SETA on Heroku, include integrate GitHub repository with Heroku and migration database.

- May 31 - June 7:(one week) Make SETA to use ActiveData instead of using data collected by Ouija in SETA. It can be used to reduce the database needs in the further work.

- June 8 - June 22:(two weeks) Add a stage server for SETA, and add tag or similar stamp for buildbot job so we can distinguish it from the job from TaskCluster.

*   milestone: we should make SETA can running on Heroku and have a stage server for it. We can test and add new data on the stage side and manually merge it from stage server.

- June 23- July 14:(three weeks) Teach SETA about TaskCluster, make SETA can accepted data from TaskCluster and draft an ideal about analysis those data, and add some tests for it. And we may need to make a plan about how to use ActiveData instead of using data collected by Ouija in SETA if we have spare time.

- July 15 - July 28:(About two weeks) Use SETA to influence the gecko decision task and add some more tests for it. At the same time, I need to handle the final exam around June.

* milestone: At this point, the new SETA can support Taskcluster job and gecko decision task can get job graph from it.

- July 29 - August 4:(two weeks) Document how SETA works and auto-deployments of docs and Heroku, include Write automatically generated documentation and Add auto-deployments to Heroku and readthedocs.

- August 5 - August 12:(one week) Make the current CI (Buildbot) use the new SETA Heroku service and handle some unexpected problems if necessary.

- August 13 - August 23:(about one week) Ensuring the code is integrated and made available on Google Code and in the Mozilla repositories. We could add priorities into SETA data and make create SETA data for per test information If we have enough time to do so.

As I mentioned above, I need handle my final exam about June and I will be free to work on this full-time from end-June until mid-August.

Open Source Development Experience

I had joined into Mozilla Open Source Community for more than one year and get involved into several projects**[1]** in A-team of Mozilla. In last summer, I take part in the Quarter of Contribution as contributor for Perfherder**[2]**, working on further improving the interface for usability and add new interface for it. At the same time, I'm one of active contributors for mozregression**[3]**, most works I do for mozregression are about polish the Gui and improve test coverage for it. In last October, I start my work for Mozci_tool**[4]** and release the Pushlog_client**[5]** as independent package**[6]**. I'm still working on these projects I mentioned above.

Academic Experience

I graduate from North China Institute of Aerospace Engineering and achieve the Bachelor of Network Engineering. Now, I'm a postgraduate student of Software Engineering in the Nanchang Hangkong University**.**

Why Me

I had learned 4 years of C/C++ before and finished my graduation project with it. And I have 2 years of python experience and 1 year javascript experience. Most of my pr can prove that and I’m sure these experiences are useful for this project.
Also, I have been working on the Open Source and Mozilla A-team as contributor for a long time. Therefore, I have confidence in my understanding about how SETA actually works in A-team and how to improve it as well. I had solved some bugs about Buildbot and Taskcluster support in Mozici_tool**[7]** before, which can save a lot of time in getting familiar with the usage and concepts in them.

Why Mozilla

When the first time I come to Mozilla open source community, I feel nervous and unconfident about myself. But people here kept encouraging me and helping me commit patches one by one. I appreciate every 'Hello' on the irc and the comments on the github or bugzilla. It really means alot to me.After learning and working on here for a long time, I think I’m familiar enough with the work flow in A-team and how ask for help from mentor. Therefore, I choose Mozilla as the organization for Google summer of Code 2016.

Reference

[1] My bugzilla page:https://bugzilla.mozilla.org/user_profile?login=sabergeass%40gmail.com

[2] PR for Perfherder:https://github.com/mozilla/treeherder/pulls?q=is%3Apr+author%3AMikeLing+is%3Aclosed

[3] PR for mozregression:https://github.com/mozilla/mozregression/pulls?q=is%3Apr+author%3AMikeLing+is%3Aclosed

[4] PR for Mozci_tool:https://github.com/mozilla/mozilla_ci_tools/pulls?q=is%3Apr+author%3AMikeLing+is%3Aclosed

[5] PR for Pushlog_client:https://github.com/mozilla/version-control-tools/pulls?q=is%3Apr+is%3Aclosed

[6] Pushlog_client page:https://pypi.python.org/pypi/pushlog_client

[7] one of the bugs I solved for Taskcluster support:
https://github.com/mozilla/mozilla_ci_tools/pull/390