NetBSD - SQL Database for ATF tests results with online query and statistics page

Personal Information:

Contact details:

- Name: Nizar Benshaqi
- Email: benshaqi.nizar@gmail.com
- Phone: +212 643146757
- Location: Tangier, Morocco
- Timezone: GMT (+0)

Short Bio:

I am a first year undergraduate student at the National School of Trading and Management (*Ecole Nationale de Commerce et de Gestion - ENCG*), but programming is my life's passion , as I started since the age of 12 by learning PHP and Web Development, then I started working as freelancer for 4 years in Web Development and Server Administration. After getting enough experience in those fields, I decided to switch to Python for Data Analysis 2 years ago. This is my first time applying to the GSoC and I am excited to be working with the NetBSD community!

Personal Availability:

I have reviewed the GSoC 2018 Timeline and I am agreeing that I will be fully available without any constraints during the time mentioned there.

Project Abstract:

This project aims for :

- Creating a tool that takes XML Test results , generated by the Automated Testing Framework (ATF), and inserts it to a well-designed PostgreSQL Database.
- Building a website that shows statistics based on the data from the PostgreSQL database, and that enables dedicated queries in a simple way.

Deliverables:

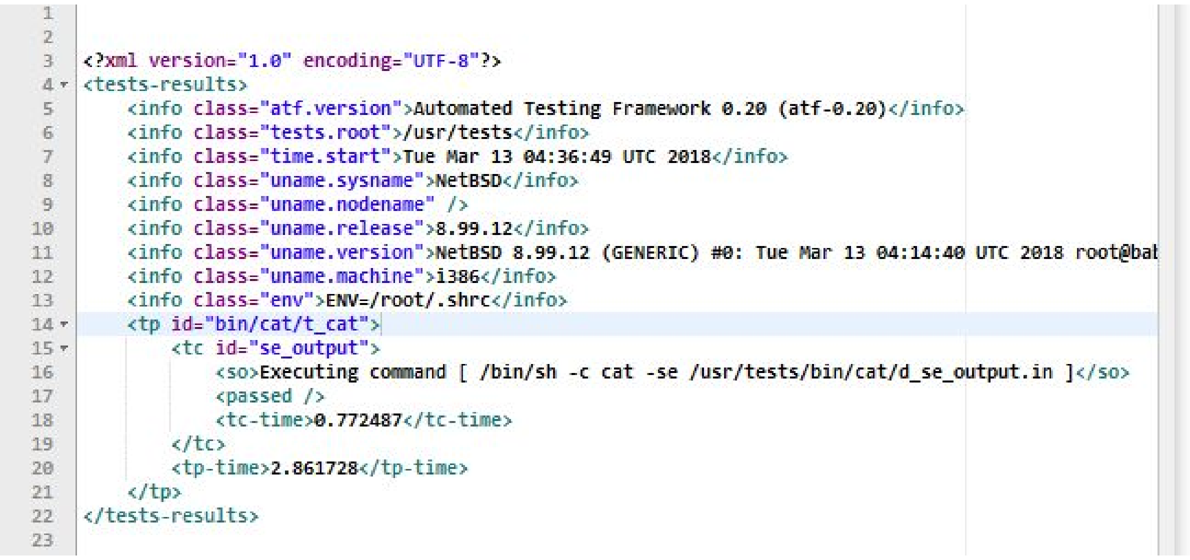
- A suitable PostgreSQL Database, with a suitable schema for multiple architectures.
- Build a tool that uploads and processes XML ATF files to the PostgreSQL database.
- Create a website that allows searching from the PostgreSQL database.
- Enhance the tool (or implement a new one) so it can be used with Kyua's output.
- Create a documentation for the database, website source, and the tool's source code

Timeline:

- April 23 - May 14: Community Bonding Period , I will use this time to discuss the project in much more details with the community and with my mentor.I will also try to get as much ATF XML files as possible to work on, alongside with Kyua's reports to decide the suitable database schema that will work for both ATF & Kyua.
- May 14 - June 11: By that time, I will have finished working on database and the upload utility.
- June 11 - July 9 : I will be finished building the website that allows searching in the previous database, and also completed working on the documentation for both the tool and the website.
- July 9 - August 6: Depending on Kyua's data, I will be finished working on enhancing the tool , or implementing a new one for that type of output.
- I can work for 8 hours a day, 5 days a week during both the community bonding period and the coding period.

Implementation:

- For the upload tool, I will use Python 3 with psycopg (for PostgreSQL operations) and Elementtree API (xml.etree.ElementTree) for the XML processing , as the ATF XML structure is simple:



```


<?xml version="1.0" encoding="UTF-8"?>
<tests-results>
    <info class="atf.version">Automated Testing Framework 0.20 (atf-0.20)</info>
    <info class="tests.root">/usr/tests</info>
    <info class="time.start">Tue Mar 13 04:36:49 UTC 2018</info>
    <info class="uname.sysname">NetBSD</info>
    <info class="uname.nodename" />
    <info class="uname.release">8.99.12</info>
    <info class="uname.version">NetBSD 8.99.12 (GENERIC) #0: Tue Mar 13 04:14:40 UTC 2018 root@bal
    <info class="uname.machine">i386</info>
    <info class="env">ENV=/root/.shrc</info>
    <tp id="bin/cat/t_cat">
        <tc id="se_output">
            <so>Executing command [ /bin/sh -c cat -se /usr/tests/bin/cat/d_se_output.in ]</so>
            <passed />
            <tc-time>0.772487</tc-time>
        </tc>
        <tp-time>2.861728</tp-time>
    </tp>
</tests-results>

```

-> So the first nodes to get will be the **<info>** then parse them to extract data like os release, platform, date, start time, end time…. which Implemented it (using the data from http://releng.netbsd.org/b5reports/i386/2018/2018.03.13.02.24.26/test.xml) , and this was the result:

```
{
  'startTime':'04:36:49',
  'startDate':'2018/3/13',
  'endTime':'17:13:55',
  'endDate':'2018/3/13'
}
{
  'sysname':'NetBSD',
  'release':'8.99.12',
  'machine':'i386'
}
```

(Two Python Dictionaries)

There are also environment variables:

BLOCKSIZE:1k

ENV:/root/.shrc

HOME:/root

HOST:

LOGNAME:root

OLDPWD:/root

PATH:/sbin:/usr/sbin:/bin:/usr/bin:/usr/pkg/sbin:/usr/pkg/bin:/usr/X11R7/bin:/usr/local/sbin:/usr/local/bin

PWD:/usr/tests

SHELL:/bin/sh
TERM:vt100
USER:root

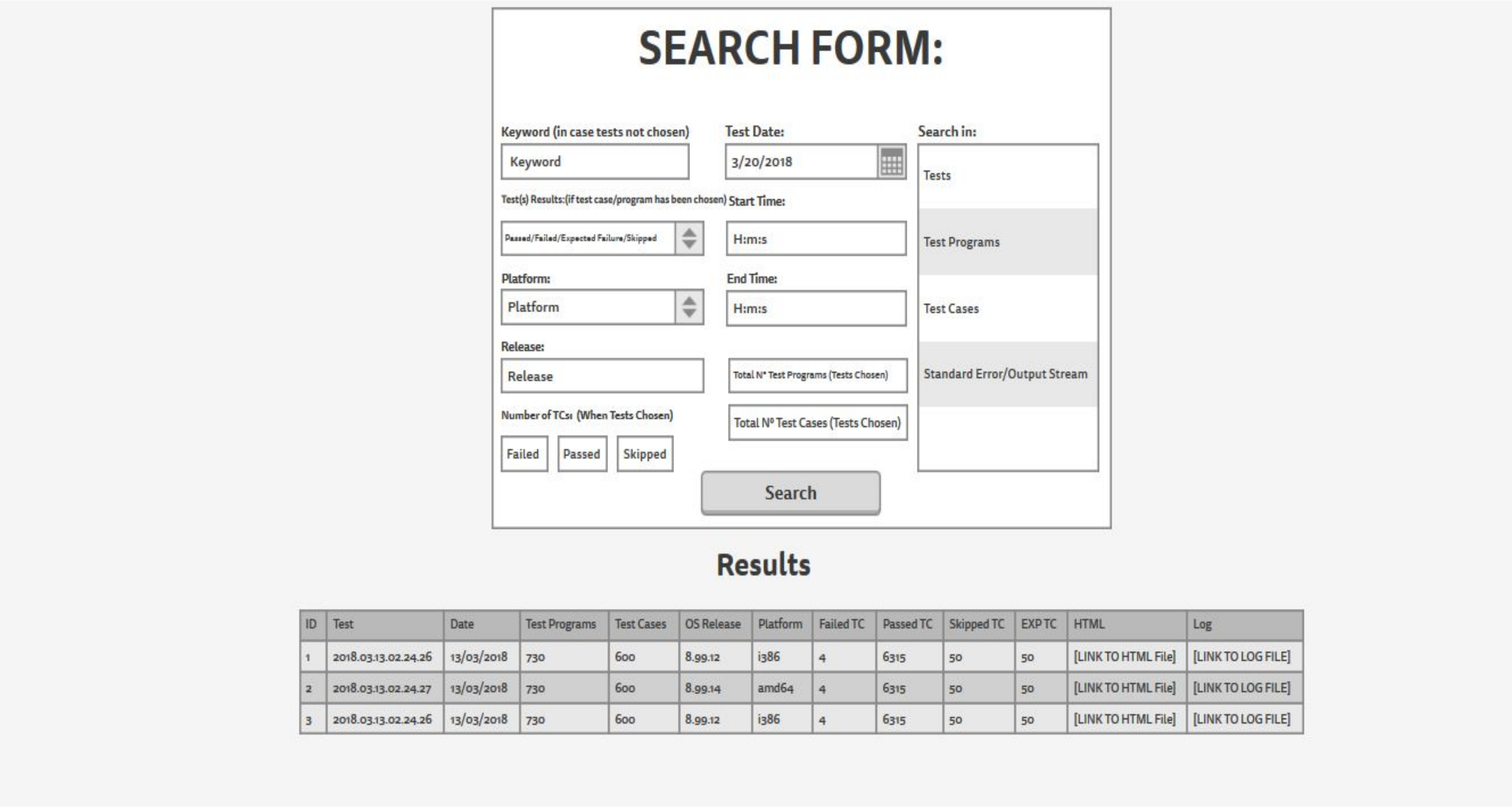
-> The second nodes to get are the **<tp>** (Test Programs):
Each test program tag contains **<tp-time>** & a number of **<tc>**, the first one represents the time elapsed for that test program, and the second one represents test cases in that particular test program.
-> The third nodes (which exist inside each **<tc>**) to get are: **<so>** and/or **<se>** (*Standard output stream, Standard error stream*) alongside with **<tc-time>** and a final tag which is one of the following: **<passed>**, **<skipped>**, **<expected_failure>** & **<failed>**.

Example:

```
<tp id="bin/sh/t_ulimit">
  <tc id="limits">
    <so>Executing command [ /bin/sh -c ulimit -a ]</so>
    <se>local: -c: bad variable name</se>
    <so>Executing command [ /bin/sh -c ulimit -b ]</so>
    <passed />
    <tc-time>2.416229</tc-time>
  </tc>
  <tp-time>2.469136</tp-time>
</tp>
```

With all the data extracted from those tags, I will be able to insert it to a suitable Postgresql database, and then integrate them with the search website.

- As for the searching website, I will also use Python 3 with Flask framework as it is lightweight and perfect for this project. For the front end , I will use CSS media queries to make the pages responsive, alongside with JavaScript for anything related to charting, this is a mockup for the search form (subject to change, depending on database schema):



SEARCH FORM:

Keyword (in case tests not chosen)
Keyword

Test(s) Results:(if test case/program has been chosen)
Passed/Failed/Expected Failure/Skipped

Platform:
Platform

Release:
Release

Number of TCs: (When Tests Chosen)
Failed Passed Skipped

Test Date:
3/20/2018

Start Time:
H:m:s

End Time:
H:m:s

Total N° Test Programs (Tests Chosen)

Total Nº Test Cases (Tests Chosen)

Search in:
Tests
Test Programs
Test Cases
Standard Error/Output Stream

Results

ID	Test	Date	Test Programs	Test Cases	OS Release	Platform	Failed TC	Passed TC	Skipped TC	EXP TC	HTML	Log
1	2018.03.13.02.24.26	13/03/2018	730	600	8.99.12	i386	4	6315	50	50	[LINK TO HTML File]	[LINK TO LOG FILE]
2	2018.03.13.02.24.27	13/03/2018	730	600	8.99.14	amd64	4	6315	50	50	[LINK TO HTML File]	[LINK TO LOG FILE]
3	2018.03.13.02.24.26	13/03/2018	730	600	8.99.12	i386	4	6315	50	50	[LINK TO HTML File]	[LINK TO LOG FILE]

(Full width image: https://imgur.com/a/J25Kk)

- With the same technologies (Python 3) but this time with sqlite3 library, I will implement a new tool related to Kyua's output (As Kyua stores data in SQLite database) or enhance the first one to handle this.

Future Deliverables:

- As this project is planned to be deployed on the NetBSD servers, I would like to be part of it for any enhancements, bug fixes and solving issues related to it.
- (If needed) Build a RESTful API to get the data in & out of the database using Python.

Why am I the right person to take this project:

I am passionate about open source technologies and i have used NetBSD previously (While administrating some servers) together with other open source technologies for my personal and class projects (Linux, Git, Vim…). I believe in the open source culture of sharing and would like to give back to the Open source community by my participation in this year's GSOC 2018.