Extending Prombench and adding rule formatting for Prometheus

CNCF/Prometheus - GSoC 2019

---

1. Student Information:

Name: Hrishikesh Barman
Email: hrishikeshbman@gmail.com
University: GIMT Guwahati, Assam, India
Major: 3rd year undergraduate(pre-final), Computer Science and Engineering
IRC: geekodour08
GitHub/Twitter: @geekodour
Homepage: geekodour.xyz

2. Introduction

Prombench is the E2E benchmarking tool for Prometheus. Currently it has a querier, the scalar and the prombech command itself. There are a lot of things to be added to prombench such as new commands, CI, logging for deployments, new tests such as extending the scalar to help with scalability tests and race tests to detect race conditions. Addressing these issues one by one will help in making Prombench a very robust benchmarking tool with a lot of insights and flexibility to suit more real world conditions.

There is a longstanding issue related to the formatting of Alerting and Recording rules(#21) I'd like to work on this as having promfmt will lead to a better user experience for prometheus users.

During my GSoC period, I plan to work on the following deliverables and hopefully finish all of them on time:

3. Deliverables

1. Extend prombench by working on high priority issues (prombench/issues)
2. Formatting of rules with promtool (prometheus#21)

3.1. Details and implementations of deliverables

3.1.1. Extending Prombench

Prombench is in a good state to be extended in a lot of ways. It also has some exiting issues related to scalability tests and adding -race enabled tests. The subtasks are as follows:

- Scalability tests
  - Continue work on the current golang scalar which scales replicas up and down and the results can be seen in grafana.
  - Additional tests should give a better
- Adding race enabled tests
  - This will involving adding a deployment object to build a race enabled prometheus and run loadgen against it.
  - Should be triggered by prow with `/race [<release>]`
  - Should keep track of all the detected race issues in a log file and run a side service to expose metrics related to race detection from the log file which can be used by prow to send alerts if required.
  - `/race stop` should fetch the log
- Displaying deployment logs
  - This becomes more important as we start adding more and more components into prombench, having a centralized logging dashboard will help
  - We can either do this through gke or via k8s apis
- Other prombench issues

3.1.2. Formatting of rules with Promtool

Prometheus rules use PromQL embedded in YAML. Through prometheus only accepts syntactically correct rule files, an uniform style that minimizes deviation and learning curve is needed, this will help extend promtool with the `promfmt` option and pretty print rules in the web UI. This task can be divided into three parts:

- Adding formatting to promql expressions
  - The `String()` methods at `printer.go` already do this, which can be improved if required to add additional formatting.
  - Investigate issues such as promql comments (prometheus#2855)
- Preserve comments, line breaks and having consistent indentation in YAML files
  - Use the `yaml.v3` library as it added support for almost all of these.
  - This will also including modifying indentation if the original file has indentation that is not of the specified style.
- Extending promtool with promfmt and making any required changes to the UI

4. Timeline

Week	Timeline	Activity
CBP	May 06 - May 27	Layout working plan for all of the assigned tasks and get feedback on issues.
1	May 27 - June 3	• Create a document for formatting style guidelines and get feedbacks • **Start** working on the prometheus rule formatting tool • Work on upgrading related packages to yaml v3
2	June 3 - June 10	• Continue working on the rule formatter • **Start** working on adding race enabled tests for prombench
3	June 10 - June 17	• Work on making changes to the UI and promtool for promfmt • Write required tests and documentation • Continue working on race enabled tests for prombench • Decide on how to extend the golang scalar to help with scalability tests
4	June 17 - June 24	• Finish rule formatter tool • **Start** working on extending the golang scalar in required areas. • Finish any leftover work from previous weeks • Submit current developments for 1st evaluation
5	June 24 - July 01	• Continue working on extending the golang scalar • **Start** working on adding new components to prombench
6	July 01 - July 08	• Finalize the implementation of logging dashboard and its storage for showing the deployment logs • **Start** working on deployment logs if not worked on already
7	July 8 - July 15	• **Start** working on the grafana dashboard to add more dashboard components, this will also be part of adding scalability tests to add additional metrics
8	July 15 - July 22	• Finish race enable tests deployment and wrap up current developments with the scalability tests • Continue working on deployment log dashboard • Submit for 2nd evaluation
9	July 22 - July 29	• If the above tasks get finished on time, will start working on E2E SD tests from prombench and by improving respective service discovery tests. • Continue work on existing prombench issues
10	July 29 - Aug 05	• Continue working on E2E SD tests if time permits • Work on improving documentation for local development for prombench
11	Aug 05 - Aug 12	• Continue working on any leftover work
12	Aug 12 - Aug 19	• Organize everything done till now and submit for final evaluations

After the GSoC period, I want to keep contributing to Prometheus projects.

5. About me

I am a 3rd year computer science undergraduate at GIMT Guwahati. I am interested in sysadmin related tasks, systems and network programming.

I got introduced to Prometheus was when I was interning remotely at the FSF (Dec'18) checking if Prometheus will be a viable option for their monitoring needs. I learnt a little bit of Golang and then started contributing to the project. I made some small contributions to prometheus, alertmanager, golang client and prombench other than that, I've been contributing to various other open source projects for the past years.

6. Availability

My final exams will be around **May 25th-June 20th** and after that I'll be interning at rtCamp Pune, India as a junior SysAdmin from **June 24th - August 20th**. As most of the GSoC period falls when my exams are going on and my internship period, I'll try to do some information gathering and planning related work during the community bonding period. During the coding period, I plan to work 5-7 hrs on weekdays and work full-time(9-10hrs) on weekends and holidays for the GSoC tasks.