ERCIM "ALAIN BENSOUSSAN" FELLOWSHIP PROGRAMME

ERCIM
European Research Consortium for Informatics and Mathematics

Scientific Report

First name / Family name	Sehrish Malik
Nationality	Pakistan
Name of the *Host Organisation*	Simula Research Laboratory
First Name / family name of the *Scientific Coordinator*	Leon Moonen
Period of the fellowship	01/02/2021 to 01/31/2022

I – SCIENTIFIC ACTIVITY DURING YOUR FELLOWSHIP

I have been working as part of cureIT project in dataSED department under prof. Leon Moonen's supervision. In this project, I have aimed at developing strategy for the evaluation of self-healing techniques; that can learn system's normal behaviour and identify when a system behaves abnormally and diagnose root cause of failure along with its impact. My work was focused at exploring the viability of using chaos engineering for this purpose. Concrete activities included:

1. Study of chaos engineering
2. Study of Kubernetes
3. Study of possible failure scenarios in a given system
4. Injecting faults in a running system
5. Monitoring changes in system behaviour before and after fault injection

I started my fellowship with a collaboration on development of research agenda for self-healing systems. We explored the suitability of AIS for building self-healing systems; and we surveyed the current research directions in self-healing systems and AIS. This research study was published in SANER 2021.

Then, I continued with the study of chaos engineering and Kubernetes; covering related books, related papers and related articles. Also, in order to further equip myself, I took two online courses:

Course 1: A Practical Guide to Kubernetes (online at Udemy)

This course covers the concepts of:

- Fundamentals of Kubernetes and what the main components of a cluster look like.
- How to use those components to build, test, deploy, and upgrade applications, as well as how to achieve state persistence once your application is deployed.
- How to secure deployments and manage resources, which are crucial DevOps skills.

Course 2: Kubernetes chaos engineering with chaos toolkit and Istio (Online at Educative)

This course covers the concepts of:

- The benefits of chaos engineering
- How to be rewarded for destruction
- How to test and find limits of a system
- How to improve a system based on lessons learned from chaos experiments

Exploring Chaos Engineering for Software testing and monitoring

Chaos Engineering is the discipline of experimenting on a system in order to build confidence in the system's capability to withstand turbulent conditions in production. Through scientific chaos engineering experiments, you can test for evidence of weaknesses in your system. Figure 1 shows the work-flow during a chaos experiment.

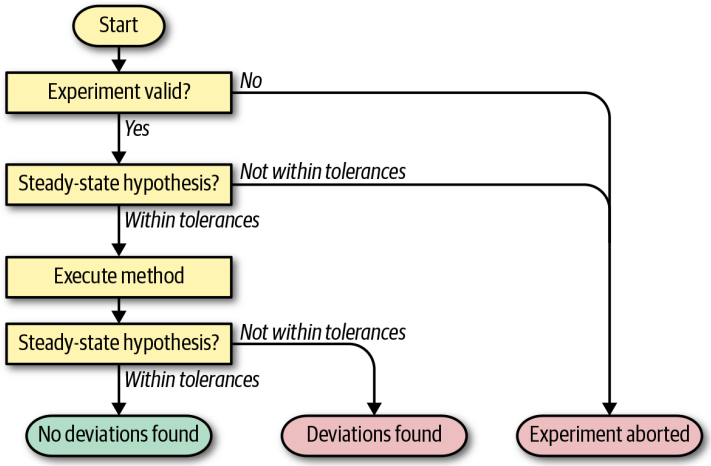

Figure 1: Chaos Experiment Flow

Next we explore Kubernetes environments for chaos testing and system monitoring with sample application testings. **Kubernetes:** an open-source container-orchestration system for automating computer application deployment, scaling, and management. **Fault injection** using the chaos toolkit to orchestrate chaos experiments against the target system. **Monitoring tools** of Grafana, Kiali and Prometheus; for monitoring performance metrics, data scraping, and visualisation of data traffic flow.

We have used many online available Kubernetes demo projects [1] to examine failure scenarios, inject faults and monitor how the fault injects affects the system under study. The demo projects are deployed in the setup as shown in figure 2, and we then design and run chaos tests on the deployed services and examine the affects on overall system via monitoring tools.

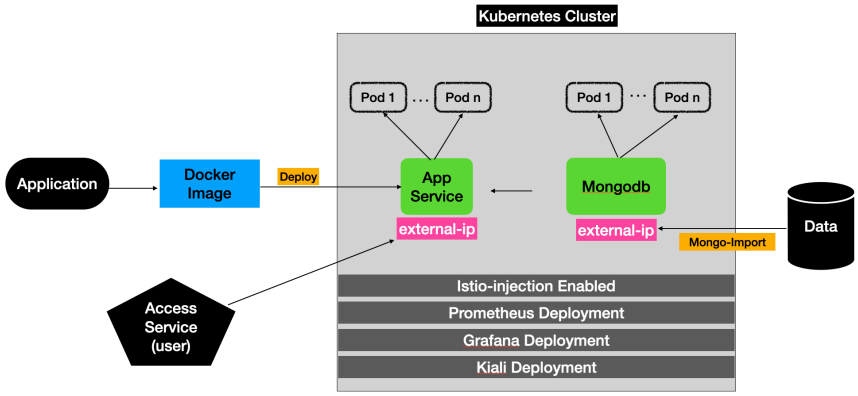

Figure 2: Deployment Overview for Fault injection and Monitoring using Chaostoolkit-kubernetes

In figure 3, I present a chaos test run in a Kubernetes cluster. **Chaos test for service availability:** Each service has 2 pods running; if some events lead to termination of both pods then service becomes unavailable. Best solution here is to use auto-scaling feature in Kubernetes environment. The graph is figure 3, can be examined via Kiali (monitoring tool).

```
serry@Sehrishs-MacBook-Pro istio-1.3.1 % chaos run samples/chaos/termpod2.yaml
[2021-04-21 04:39:03 INFO] Validating the experiment's syntax
[2021-04-21 04:39:04 INFO] Experiment looks valid
[2021-04-21 04:39:04 INFO] Running experiment: What happens if we terminate an instance of the application?
[2021-04-21 04:39:04 INFO] Steady-state strategy: default
[2021-04-21 04:39:04 INFO] Rollbacks strategy: default
[2021-04-21 04:39:04 INFO] Steady state hypothesis: The app is healthy
[2021-04-21 04:39:04 INFO] Probe: all-apps-are-healthy
[2021-04-21 04:39:05 INFO] Steady state hypothesis is met!
[2021-04-21 04:39:05 INFO] Playing your experiment's method now...
[2021-04-21 04:39:05 INFO] Action: terminate-app-pod
[2021-04-21 04:39:05 INFO] Action: terminate-app-pod
[2021-04-21 04:39:05 INFO] Steady state hypothesis: The app is healthy
[2021-04-21 04:39:05 INFO] Probe: all-apps-are-healthy
[2021-04-21 04:39:05 ERROR]    => failed: chaoslib.exceptions.ActivityFailed: the system is unhealthy
[2021-04-21 04:39:05 WARNING] Probe terminated unexpectedly, so its tolerance could not be validated
[2021-04-21 04:39:05 CRITICAL] Steady state probe 'all-apps-are-healthy' is not in the given tolerance so failing this experiment
[2021-04-21 04:39:05 INFO] Experiment ended with status: deviated
[2021-04-21 04:39:05 INFO] The steady-state has deviated, a weakness may have been discovered
```

```
serry@Sehrishs-MacBook-Pro istio-1.3.1 % kubectl get pods
NAME                              READY   STATUS        RESTARTS   AGE
details-v1-847c7999fb-v8pgp       2/2     Running       0          54m
productpage-v1-764fd8c446-f257t   2/2     Terminating   0          27s
productpage-v1-764fd8c446-g9p4s   2/2     Terminating   0          79s
productpage-v1-764fd8c446-p9cr6   0/2     Init:0/1      0          1s
ratings-v1-7c46bc6f4d-2wbwj       2/2     Running       0          54m
reviews-v1-76ddd45467-9bgq9       2/2     Running       0          54m
reviews-v2-797544d467-s7jgl       2/2     Running       0          54m
reviews-v3-5dfdfdf979-k77j9       2/2     Running       0          54m
```

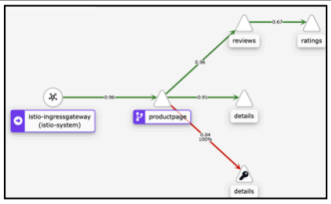

Figure 3: A chaos test to check application availability

During this period, I have explored and experimented with injection of failures into systems using chaos engineering and studied system behaviour. The results helped us in finding system patterns and reactions during and after failure encounters; it can further assist us in building a resilient system. The purpose of this study is to make add self-healing and self-adapting capabilities into the system. We developed our own guinea pig system (with limited use cases for now) of a smart office to have more freedom in designing flexible failure scenarios, and then incorporate self-healing properties into this system based on failure responses. The currently implemented smart office use-cases are tested for failure scenarios of: service availability, erroneous input from sensors, sensor down, and delay in service. We then deployed counter measures to make the system resilient to these failures; to compare resilient and non-resilient services. In addition, we also examine the level of resilience supported within Kubernetes environment without addition of custom functionality. The findings from this study are planned to be added in our soon to be submitted paper for SEAMS 2022.

[1] https://williamlam.com/2020/06/interesting-kubernetes-application-demos.html

II – PUBLICATION(S) DURING YOUR FELLOWSHIP

1. Naqvi, M.A., Astekin, M., Malik, S. and Moonen, L., 2021, March. Adaptive Immunity for Software: Towards Autonomous Self-healing Systems. In *International Conference on Software Analysis, Evolution and Reengineering (SANER)* (pp. 521-525). IEEE.

III – ATTENDED SEMINARS, WORKHOPS, CONFERENCES

Conferences:

1. Software Analysis, Evolution and Reengineering Virtual (SANER); March 9-12, 2021
2. International Conference on Software Engineering (ICSE); May 25-28, 2021

Workshops:

1. Online CodeRefinery workshop May 10-12, 18-20.

   To learn about Git, GitHub, software testing, documentation writing, Jupyter notebooks, tools for reproducible research software development, software licensing, and modular coding, and many tips and tricks.

Postdoc Career Success Programme 2021

Programme content

- 15 online modules over 7 months (on average 2 hours to complete)
- one-to-one access to PostdocTraining's experienced mentors for career coaching
- email career guidance / email support
- career options masterlist, a database of PhD career path case-studies
- a monthly news service on research-related career issues

IV – RESEARCH EXCHANGE PROGRAMME (REP)

I completed my research exchange programme from 5 to 9 December with Prof. Fabio Martinilli, who is a research director of the Italian National Research Council (CNR). The REP is was conducted online due to covid restrictions regarding travelling.

We started the REP, with an introduction presentation, where I introduced all the research areas and research topics I have worked on till date. Then, a few members from Prof. Fabio's group introduced their research topics. The presentations were followed by discussions' sessions to help in understanding each other's research directions. The presentation material and relevant published papers were exchanged for follow-up reads and further discussions continued via emails.

During this REP, we had a valuable experience of exchanging our respective research backgrounds; learning new research ideas and directions from each other; and finding potential topics for future collaborations.