

475 Software Engineering for Industry : Topic 7 : Resilience at Scale

Assessed - work in a group of 3.

In this exercise we will look **techniques for building systems that are large and contain many subsystems, but are resilient to problems**. We will do this through a combination of reading, coding, writing and discussion.

Reading

Given that Netflix now serves a very large amount of users, their engineering blog is a good source of information in building large scale systems that are resilient to faults and problems. For example: <http://techblog.netflix.com/2011/12/making-netflix-api-more-resilient.html>

The overall field of attending to reliability and resilience, especially for large web systems, has commonly become known as “SRE” - Site Reliability Engineering. There are a lot of resources on this topic, many of which have been conveniently curated by Pavlos Ratis. Pick out something that interests you... <https://github.com/dastergon/awesome-sre>

Practical

For this exercise we'll work with a small web application that depends on two further services to retrieve data. We'll try to improve the system so that the user gets a better experience in the case that one of the downstream services becomes overloaded.

Look at pages 2 onwards of this spec for details of how to get the code etc.

You should find three simple web servers - `FrontEndWebServer`, `NewsDataServer` and `WeatherDataServer`. If you run the `main()` method in each of these you can start them all up on different ports. If you load the page served by the front end server, it should show a summary of news and weather data retrieved from the downstream services. You need to start all three servers before hitting the web page.

The weather data service contains some code to simulate it getting overloaded by bursts of requests. If you refresh the webpage a few times in quick succession you should see this effect.

Build a *circuit breaker* between the front end server and the weather data server to control requests to the weather service, and try to make sure that the user gets a page back quickly, even if it does not contain both news and weather data. You can change whatever you like in the code.

Questions to Consider

Why is a circuit breaker more effective than a client-side timeout?

When is the right time to trigger and reset the circuit breaker?

How does the way your circuit breaker operates relate to yield and harvest?

Getting The Code

You can find the skeleton code on GitLab. Go to <https://gitlab.doc.ic.ac.uk/475/CircuitBreaker> and click Fork Repository to create your own copy of the repository under your own GitLab account (only one person from your group needs to do this, then they can add the other group members as developers on the project). Clone your repo locally to start work.

Importing the code into an IDE...

The skeleton code contains a Maven build file (pom.xml), so your IDE should be able to recognise the structure. But you can set up your IDE however you like.

For **IntelliJ IDEA**: you can *Import Project*, browse to cloned directory, *Import from external model*, choose Maven, then keep clicking *Next* until finished. If no JDK is set, choose a ≥ 1.7 JDK.

Writing

Once you have made some progress with the coding part, write up a summary of your thoughts on resilience at scale. The submission should be 2 pages (no covers or contents pages please).

On page one, address one of the following questions:

- Is resilience the concern of operations engineers, software architects, or software developers?
- Does moving from a monolith to a microservice architecture help with resilience?
- Are resilience and high capacity co-supporting? Or do they fight against one another?
- What determines the level of reliability/resilience a service should aim for?

This is not a long essay, just a short statement of your thoughts, with supporting evidence and references. Try to focus on your points and be interesting!

There is not an explicit word limit, but we recommend writing concisely and aiming for around 300 words. Do not go over one page, including your references.

On page two, give a brief description / example of what you did in the practical part. You don't need a lot of writing - perhaps a diagram or a screenshot would be a good way to show it.

Submission As a group, submit a pdf (topic7.pdf) of your 2-page write-up via CATE.

Deadline Monday 4th Mar, 9am.

Discussion

During the class on Tuesday 5th March, we will discuss your thoughts and experiences. We will ask some groups to briefly present their case studies, and others to describe their thoughts. We hope for a good discussion amongst the class.

Page 2 of the submission is not graded, but you must demonstrate that you have done something reasonable for that part in order to have page 1 graded.

On page 1 we are looking for you to express your thoughts and ideas based on your reading, experience and discussions, backed up by evidence. The grading scheme is as follows:

Assessment

In your written work we are looking for you to express your thoughts and ideas based on your reading, experience and discussions, backed up by evidence.

F- E

Little or no understanding of the given topic demonstrated.

D

Shows an incorrect or flawed understanding of how or why to apply the given tools/techniques.

C

Shows a reasonable, but limited, understanding of the application of ideas and techniques covered, and the context in which they apply.

B

Shows a good understanding of how to apply these techniques and the problems that they solve. Arguments are well presented and backed up by references.

A

Displays a broad understanding of the use of these techniques, comparing different approaches and the forces that might make them suitable for different situations, displaying evidence of further independent reading and thought, beyond what was suggested and covered in the class.

A*

Gives an excellent and insightful commentary, comparing different tools and approaches and displaying evidence of further independent reading and thought. Demonstrates critical thinking and considered opinion, but backed up by references and practical experience.