# Jepsen methods usage for ACID compliance in Hyperscale Cloud Frameworks

## Introduction

The goal of this project is to assess how Jepsen [2] tests allow us to verify the properties of ACID [1] in a cloud system. The Jepsen test is a method used to evaluate the compliance of a system in relation to the ACID properties. This is done is by analysing the system via Blackbox [5] tests , in the sense that the internals of the system work are not relevant. We look at the service from a client-side and not any underlying structures or frameworks. The ACID properties are

- Atomicity
  Guarantees that each transaction is treated as a single "unit", which either succeeds completely or fails completely
- Consistency
  Ensures that a transaction can only bring the database from one valid state to another
- Isolation
  Ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially
- Durability
  Guarantees that once a transaction has been committed, it will remain committed even in the case of a system failure [1]

We plan to apply these methods on Service Fabric [6] , to verify if this system is compliant with the ACID properties. Service Fabric is a framework developed by Microsoft, designed to allow developers to build Hyper-Scale Cloud (HSC) deployments on the Azure [8] platform. The motivation behind this thesis is to study the ACID properties in an HSC environment and evaluate how the constraints of ACID hold up in practice, as well as verifying if these constraints are met, and more importantly if they are not.

## Plan

The project is segmented into three parts. The writing of the report will take place at all times throughout the process, and the final phase should be a finalization and corrections phase.

### The study phase (September to November)

- The first part of the study will focus on analysing the previous Jepsen tests performed on other systems [3]. This will allow us to define a strategy in terms of: which tools, tactics and methods are worth investigating, and what type of faults should be taken note of. This information is useful for two aspects:
  - where our focus should be when testing,
  - which tools and packages we should be familiarize with.
- The second part of the study will centre around “Service Fabric” and getting to know how the framework is coupled together, what vectors we can access the system from, and how we can manipulate the framework to induce fault conditions.

### The experimentation phase (November to April)

- In the experimentation phase, we will design and perform tests on the system. If any faults occur, we will attempt to locate where, and why they occur. If the scope and the timeline of the project allow, we will assess whether we prevent the faults from occurring in the future.
- The Experimentation phase of the project will include the steps described below.

  - Planning of the test, selection of the aspects of the system to be tested, consulting with developers and users of the system to determine if and where undesired behaviour may occur.
  - Designing the tests and setting up the tools to facilitate those tests.
  - Writing the tests, verifying they work as intended and setting up a data collection framework to collect the data in a manner that allows our analysis.
  - Performing the tests on the system in different scenarios, normal conditions and different levels of faults and disaster recovery.
  - Analysing the data for faults, errors and inconsistencies.
  - Consulting with DEVs to determine where the faults occurred, and which failures or bugs led to these fault conditions.
  - Concluding whether "Service Fabric" is ACID-compliant.

### The report phase (April to May)

- The final phase of the project aims at finishing an initial draft, reviewing and correcting it, and finalizing the report for hand-in.
- Deliverables
  - At the end of the project, there should be a report, the tests and the results from those tests.
  - The report written in English and following the standard academic writing conventions will include
    * A study on Jepsen tests, describing what they are, as well as why and how they are done.
    * An experiment in which we will analyse "Service Fabric"
    * A discussion on the conclusiveness of the test and the compliance of the framework with service fabric.
  - The code & data will include
    * Scripts and code for tests and for analysing the data.
    * Relevant results and data from the tests.

## Goal

The goal of the project is to deep dive into Jepsen tests with a focus on "Service Fabric" as the subject. The optimal goal of the project is to verify if the database aspects of "Service Fabric" comply with the ACID properties, and to locate the fault cases in case they don't. Risk assessment There are a few risks in the project. In the case no faults are found within the Service Fabric, this reduces the scope of the project. If no faults are found, time allowing, different framework can be analysed. On the contrary, if the number of faults in the framework is more extensive than expected, then the scope of the project may grow uncontrollably. In this case, choices will be made to select a subset of the data to focus on.

# Bibliography


[1] Raghu Ramakrishnan & Johannes Gehrke
*Database Management Systems*
ISBN13 9780071231510

[2] Kyle Kingsbury
http://jepsen.io/consistency

[3] Kyle Kingsbury
jepsen.io/analyses
aphyr.com/tags/jepsen

[4] Kyle Kingsbury
github.com/jepsen-io/jepsen

[5] Kyle Kingsbury
youtu.be/tRc0O9VgzB0?t=293

[6] Microsoft
docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview

[7] Microsoft
github.com/microsoft/service-fabric

[8] Microsoft
azure.microsoft.com/en-us/