

# Assignment 5 - Managing Shared State for Distributed Query Execution

## 2 Points

Handout date: 28.11.2024

Due date: 12.12.2024

Discussion: 09.01.2025

Michalis Georgoulakis, Tobias Götz ({geom,goetzt}@in.tum.de)

## Introduction

This assignment includes two tasks. The first one, which is the easiest, involves turning your Assignment 3 solution (or the base solution we provide) into a cloud-based solution. This task aims to familiarise you with the services of a public cloud.

The second task involves implementing a **distributed query execution application with shared state**. We start again with Assignment 3 partitioned data on external storage and similar elasticity goals. However, we now process a query that needs to share state between workers.

As a query, we want to calculate how often each domain appeared (not only a specific domain as in the last assignment) and report the result for the top 25 domains. We now build multiple (partial) aggregates for each input partition, one for each domain, which we then need to merge. Contrary to what we did in the last assignment, merging on the coordinator does not scale well for this query. To scale the merge phase, we partition the aggregates and store them in shared state. After the initial aggregation and partitioning, we distribute the work of merging the partial aggregate partitions and send the merged results to the coordinator.

Feel free to reuse anything from the last assignment. You can find an example in the README.md file of the repository, as well as the expected result. We also provide you with a helper function to extract the domain from the URL, which is commented out in the coordinator file.

## Submission guidelines

### Deliverables

For this assignment, you are expected to:

- submit code that implements the assignment and produces the correct result
- write a brief report answering the questions of this assignment sheet (**report.pdf**)
- write your group members' names (**groupNames.txt**)

## Section 1: Azure Deployment and Testing

### Amazon Dynamo - System Design

- (i) After you have finished with the Azure tutorial, measure the time it takes for the Assignment 3 query to run on Azure. What do you notice?
- (ii) Observe the Azure monitoring panel metrics for your containers: explain what is the bottle-neck that increases query execution time (e.g. network-bound, CPU-bound). Include screenshots if needed.
- (iii) **Let's get faster:** Pre-upload the data partitions and fileList inside Azure blob storage. Adapt your solution to read from there (you can use the **AzureBlobClient** helper functions). What is the speedup you observe? How is it explained?

## Section 2: Managing Shared State - Design Questions

Give a brief description of your solution for the programming task, answering the following questions.

- (i) Which partitioning method did you choose for the calculation of the partial aggregates?
- (ii) What number of subpartitions did you choose, and why? If you choose to create 100 subpartitions for each partition, is it a good choice? Is there a scalability limit?
- (iii) How does the coordinator know which partitions to send to the workers for the merging phase?
- (iv) How do workers differentiate from the two tasks they have to perform (partial aggregation, merging subpartitions)?
- (v) What was the query execution time in Azure? Include screenshots, and logs if needed.

## Section 3: Notes

If we need some prerequisites to run your solution, please include them here.

Also, include a brief description of your unit tests.