

Coen424/6313 Assignment1

Individual or Group of 2 Assignment due by October 31 23:59

@copyright Yan Liu 2023-2024

This assignment is originally developed by Yan Liu @ Concordia University. This assignment is only for the purpose of course teaching and the education purpose. Any distribution of this document to the Internet that involves any profit-making purpose needs to obtain the consent from the author.

This assignment is designed to practise data model definition and communication through binary serialization and deserialization (gRPC).

The dataset from of the inventor list is attached to this assignment document. A data model should be designed to represent the data records in the sample file based dataset. In the file, each column is of an attribute and each row is a record. During the lecture, we discussed the conversion of data from one form to another during the communication based on a data model. A Class ***InventoryRecord*** defines the structure of the data model, and each object of *InventoryRecord* represents each row of the dataset.

The following scenario and problems are to be developed in this assignment.

Inventory List									
Highlight items to reorder? Yes									
Inventory ID	Name	Description	Unit Price	Quantity in Stock	Inventory Value	Reorder Level	Reorder Time in Days	Quantity in Reorder	Discontinued?
IN0001	Item 1	Desc 1	\$51.00	25	\$1,275.00	29	13	50	
IN0002	Item 2	Desc 2	\$93.00	132	\$12,276.00	231	4	50	
IN0003	Item 3	Desc 3	\$57.00	151	\$8,607.00	114	11	150	
IN0004	Item 4	Desc 4	\$19.00	186	\$3,534.00	158	6	50	
IN0005	Item 5	Desc 5	\$75.00	62	\$4,650.00	39	12	50	
IN0006	Item 6	Desc 6	\$11.00	5	\$55.00	9	13	150	

Problem 1 - Data Model Communication through gRPC or any Protobuf based binary serialization and deserialization). The communication involves two parts – client and service provider. A client sends a request with parameters to retrieve a certain set of records. The service provider develops service to respond to each client request. Assume the data records are all uploaded to the memory of the service provider's runtime. The communication between the client and the service provider is through gRPC or any Protobuf based binary serialization and deserialization among services.

The service interface includes the following definition (* you may adjust the parameters)

- InventoryRecord searchByID(string ID) /*return the InventoryRecord object if an ID exists, otherwise null */

A reference below on programing null object as **Empty** request or response.

[Google.Protobuf.WellKnownTypes.Empty Class Reference](https://protobuf.dev/reference/csharp/api-docs/class/google/protobuf/well-known-types/empty) [<https://protobuf.dev/reference/csharp/api-docs/class/google/protobuf/well-known-types/empty>]

- InventoryRecord search (string Key_Name, E Key_Value) /*return the InventoryRecord object for a certain key and key value pair; if the key does not exist, return null */
- InventoryRecord[] search (string Key_Name, E Key_Value_start, E Key_Value_end) /*return a list of InventoryRecord object for a certain key and within the range in [Key_Value_start , Key_Value_end]; if the key does not exist, return null */
- double getDistribution(string Key_Name, double percentile) /*return the percentile value of a given key; if the key does not exist, return null */
- boolean update (string Key_Name, string Key_Value, string Val_Name, string Val_Val_New) /* update a certain record with the a given key name the key value and the attribute name (Val_Name) with the value (Val_Val_New) to be updated

Task 1: Develop a gRPC or any Protobuf based Client Server program to implement the above services defined on the data model of InventoryList. There is no need to have dedicated database to store the inventory list. A simple data structure that supports concurrent access to the data is fine to apply as an in-memory database.

Task 2: Deploy at least the server program a cloud resource and run. Run the client from a local computer or another cloud resource for 100 times to each of the above functions and measure the end-to-end delay. Plot summary chart for the MEAN response time and STD. The X-axis are of 4 service type labels, and the y-axis is the time in second.

Task 3: Write up the solution report following the Assignment 1 Report Template [IEEE - Manuscript Templates for Conference Proceedings](https://www.ieee.org/conferences/publishing/templates.html) <https://www.ieee.org/conferences/publishing/templates.html>

The report content outline

Section 1. Data Model Design

Section 2. Service Development

2.1 The overview of serialization and deserialization model (gRPC or protobuf based framework)

2.2 For each of the service listed above, present the service definition according to 2.1

2.3 For each service definition, screenshot of the code implementation with brief description

Section 3. Cloud Deployment and Run

Present a step by step description on the deployment to a cloud at your choice.

Attach a screenshot of the running cloud service, and returning results to the client's requests.

Section 4. The measurement filled in a Table and Plotting.

Service Name	Mean Time (in s)	STD

4.1 The settings to run 100 times of each service

4.2 The Table

4.3 The Plot