CS 4459: Assignment 2

Dr. Hanan Lutfiyya
Department of Computer Science
Western University
hlutfiyy@uwo.ca

1 Brief Description

The objective of this assignment is to understand the principles behind primary-backup architectures.

2 Requirements

This section lists the requirements.

Part 1: Primary-Backup

You are to implement a primary-backup system. A client submits a request to the primary. When this is received the primary sends the request to the backup. When the backup receives the request it applies the write specified in the request. Once this is done the backup sends an acknowledgement to the primary. The primary then applies the write and then sends an acknowledgement to the client. The primary sends the result to the client. Please note that the operation stores the key-value pair in a Python dictionary. An example of an input is the following: "1" "book". The "1" is a key and "book" is a value. The client, primary and backup should each generate log files. Each line should have the key followed by the value. The first entry should be the first write, the second entry should the second write, etc

The assumptions and requirements:

- You must use `replication.proto`. This is used between the client and the primary; and between the primary and backup.
- Use grpc and Python
- You may use a Python dictionary for storing the key-value pair.
- The names of the log files are `client.txt`, `primary.txt` and `backup.txt`.
- The port number of the primary is 50051 and the backup is 50052. This makes it easier to test.
- The names of files with the code is `client.py`, `primary.py` and `backup.py`

Part 2: Primary-Backup and ViewServer

For Part 2, the primary and backup send heartbeats to the heartbeat server. The primary and backup send a heartbeat message that includes a service identifier. This allows the heartbeat server to identify the primary and backup. Heartbeats should be sent every five seconds. The heartbeat server does not return a message back. It should be able detect when a primary or backup is not sending heartbeat messages. This is done by terminating the primary or backup. When the heartbeat server detects that a primary or backup is no longer sending heartbeats then print a message to the file.

The heartbeat server writes its messages to a log file named `heatbeat.txt`. The file format should be as follows: Primary is alive. Latest heartbeat received at [timestamp]

Backup is alive. Latest heartbeat received at[timestamp]
Primary might be down. Latest heartbeat received at[timestamp]
Backup might be down. Latest heartbeat received at [timestamp]

The assumptions and requirements:

- You must use `heartbeats_service.proto`
- The name of the file with the heartbeat server is `heartbeat_service.py`
- THe identifier of the backup and primary should be `backup` and `primary`.
- The port number of the heartbeat server is 50053.