

Project 3 (CMSC 621)

Given out: November 22th, 2015

Due: December 9th, 2015, 11:59 PM Via Blackboard.

In this project, you will be building on top of your Project 2. The different sensors, devices, gateways etc. will be the same and the syntax and semantics will be the same until not specified specifically in this document. The required files and submission procedures also will remain the same.

The problem:

The goal of this project is to implement replication in the system and also consistency. Assume that the IoT gateway implements a multi-tier architecture like before. Your goal is to replicate the gateway so that there are two replicas (each front-tier replica communicates with its own backend database tier replica). Like before, the system has a number of sensors or smart devices which are NOT replicated. A sensor or a smart device can communicate with either gateway replica. At start-up time, design a technique to associate each sensor or device with either replica such that the number of devices / sensors communicating with either replica is roughly equal. That is, you should not hard code the address of the gateway replica in the sensor / device but choose it dynamically at startup time to balance the load.

It is assumed that any replica can serve any sensor / device. The gateway backend replicas implement a consistency technique to ensure that their states (e.g., database states) are synchronized. You should use the Two Phase Commit protocol to maintain the consistency of the data between the replicas. You can make the primary replica as the coordinator. Please log all the communications on to the persistent storage (for simplicity). Please start each of those messages with the string "2PC:T<transaction_number>" (eg. 2PC:T1: <message>"), where T1 is the first transaction being updated to the replicas). The transaction_number will be incremented for each of the transactions, starting with 1. "message" need not follow a strict format, but it should include all the messages it received from others and the state changes related to 2 phase commit protocol. It should include messages like "prepare message received from coordinator", "Ready vote send", "Abort vote send", "Abort message send", "Abort message received", "moving to wait state", "moving to abort state" and "in init state". This list is not a comprehensive set of messages to be logged, but are just indicative of what need to be logged. You should log all the messages to each of the replicas and the coordinator. At any time, the devices should be communicating with only one replica of the gateway. The consistency mechanism should take care of updating the replicas.

[Extra Credit] This is for extra credit. Even if you do not implement this, you will get full points. Since the gateway is replicated, you should also make your gateway fault tolerant. It is sufficient to handle crash faults (Byzantine faults need not be handled). Also for simplicity, assume that the both tiers of a gateway replicas fail at once and in this case, the other gateway needs to take over the functions of the failed replica. A gateway node needs to dynamically determine the failure of the other replica (this can be done by any method that you choose such as exchanging "I am alive" heartbeat messages). Upon detecting a failure, the remaining gateway replica implements a failure recovery algorithm that involves taking over the responsibility of servicing

all sensors and devices that were communicating with the failed replica. Your failure recovery method needs to inform the sensors / devices of the failure and have them reconfigure themselves to communicate with the new replica for subsequent requests. While the failure recovery "algorithm" can be straightforward, clearly document how failures are detected and all the steps your replica performs to take over the functions of the failed gateway. Also explain if failures can lead to any data loss in your system (which will depend on the choice of your consistency mechanisms that synchronize state between the replicas) and the impact of any such data loss

To account for the additional replicas, the configuration file for the gateway is slightly modified. The modified gateway configuration file is attached. For the Primary gateway, the pair CurrentIPAddress & CurrentPortNumber and PrimaryGatewayIPAddress & PrimaryGatewayPortNumber will be the same.

Gateway Configuration File

Line 1:CurrentIPAddress,CurrentPortNumber

Line 2:PrimaryGateway:PrimaryGatewayIPAddress,PrimaryGatewayPortNumber.

Submission Procedure

Submit a Zip file (not tar or tar.gz files) with the following contents with the following exact structure. Please make sure all the naming conventions are strictly followed. Upload the zip. Your code should execute on a Linux (possibly Ubuntu) machine. Please test your code on a Linux machine (or virtual machine). You can work in groups of two but only one of the group members need to submit the project. We will share a google doc where we will be asking for the group members.

Instructions.txt: It should contain the details about how to compile and run the different files

README.txt: Should contain any assumptions made during the project

compile.sh: This file should contain the shell script to compile the different files inside the different folders and put their corresponding outputs in the output folder. For each folder inside src folder, there should be a corresponding binary in the output folder. That is, after running compile.sh, the output directory should contain 6 files “door”, “motion”, “keychain”, “securitysystem”, “gateway” and “database” For example, if you are using gcc to compile the shellscript can contain the following lines. The below lines are just samples and for illustration purpose only.

```
gcc -pthread -o output/sensor src/Door/door.c
```

```
gcc -pthread -o output/motion src/Motion/motion.c
```

```
gcc -pthread -o output/keychain src/KeyChain/keychain.c
```

```
gcc -pthread -o output/securitysystem src/SecuritySystem/securitysystem.c
```

```
gcc -pthread -o output/gateway src/Gateway/gateway.c
```

```
gcc -pthread -o output/database src/Database/database.c
```

Zip File Structure

ZipFile

- Instructions.txt
- README.txt
- compile.sh
- output/
 - <nothing need to be inside this folder>
- SampleConfigurationFiles/
 - <sample configuration files for all the entities>
- src/
 - Door
 - <src files for Door Sensor>
 - Motion
 - <src files for Motion Sensor>
 - KeyChain
 - <src files for Key Chain Sensor>
 - SecuritySystem
 - <src files for Security System >
 - Gateway
 - <src files for Gateway>
 - Database
 - <src files for Database>
