Computer Science Department, University of Crete

CS 452 – Introduction to the Science and Technology of Services

Winter semester 2022-2023

Responsible Instructor: Marina Bitsaki (marina@csd.uoc.gr)

Teaching Assistants: Giorgos Kelantonakis (csdp1224@csd.uoc.gr), Fallia Kourou (csdp1162@csd.uoc.gr), Achilleas Filippidis (csdp1276@csd.uoc.gr)

Assignment 3 / Mini-project

Due date: 08/01/2023

General instructions

Write a report demonstrating your results using images, screenshots, code and any other explanation necessary. Include a cover page with your student ID and name. The filename of the deliverable must be in the following form: **proj_AM** (where AM is your student id number e.g. proj_1234). Any images/screenshots and code should be included in your report document. In particular, screenshots of the AWS UI showing the implementation and test runs of the serverless applications should be included. Regarding the programming language we recommend you to use Python. The report document must be in pdf format. Submit a single compressed file (zip/rar/gz etc) if your work contains multiple files.

Note that this assignment is to be done **individually**. Cheating in any way, including giving your work to someone else, will result in failing this assignment (a mark of zero will be given).

Submit your assignment **electronically** (online) until 08/01/2023 at 23:59 using the course webpage. Delayed submission is permitted with a penalty of 10% for each additional day.

You should be ready to demonstrate your developed prototypes in an examination that may be scheduled shortly after the submission date.

Overview

In this assignment, you will experiment with Amazon AWS and more specifically you will deploy one event-driven serverless application using your Amazon Web Services (AWS) Educate Starter accounts. Amazon AWS provides a variety of serverless services with which you can interact. Basic services are AWS Lambda, Amazon SNS, Amazon SQS, Amazon DynamoDB, and Amazon EC2. You can also interact with various amazon APIs.

CASE STUDY: Market share game - Promote your business on an Ad Provider

Brief description

Online advertising is a new type of marketing that businesses use to promote products and services to users. Online advertising includes email marketing, search engine marketing, social media marketing, display advertising and mobile advertising.

In this project we are interested in developing a service system for search advertisements which sorts advertisements on bid prices. It involves a publisher (provider like Google), who integrates advertisements into its online content, and advertisers, who provide the advertisements that will be available to any interested user (client of the system).

Suppose that advertisers participate in an auction and compete for high positions on an ad ranking list. Consider that time is partitioned into rounds and at the beginning of each round, advertisers publish the maximum price they are willing to pay for a click. Upon receiving the bids and based on the statistics collected in the previous round, the provider decides which ads to show and in what order so as to increase his revenues. On receiving the ad list, clients decide which advertisement to click on and whether to make a purchase or not, and then publish their actions to the provider. We consider that a client makes a click according to a probability distribution (click probability) and for the chosen ad, the client proceeds to a purchase according to another probability distribution (purchase probability). At the end of each round, the provider publishes all statistics to the advertisers that make a new improved bid for the next round based on the observable clicks and business sales, in order to achieve a higher profit.

The service system consists of the following entities:

- **Ad Service Provider:** Manages advertising campaigns
  - Pricing Policy: Pay-per-click advertising
  - Decision: Which ads to show and in what order
  - Selection criteria: Bids, Clicks
  - Goal: Increase revenues
- **Advertisers:** Entrepreneurs that advertise their products/services to achieve their business goals
  - Decision: Bidding strategies based on observable clicks and sales
  - Advertising goal: Increase profit
- **Clients:** People searching for products/services on the provider's website
  - Decision: Which advertisement to click on and whether to purchase or not
  - Client profile: utility, click probability, purchase probability

The profile of an advertisement consists of the following parameters:

- **Clicks:** The number of clicks the ad receives
- **Sales:** The number of products sold (purchases) based on the ad

PHASE A (15 units)

Design the petri net of the described use case for two consecutive bidding rounds, considering that we have available 3 advertisers, 1 Ad service provider and 1 client. Assume that each auction round is won by a different advertiser. Design three states of this petri net: 1) The initial state before starting the auction, 2) A middle state after the termination of the first round and 3) the final state when both bidding rounds have finished. Also design the reachability graph of your petri net.

Deliverable: A section in your report including the three states of the petri net, that can be either handwritten or designed in special software (timenet, cpn tools, powerpoint), the reachability graph of the PN, a brief description for the the PN and any assumptions you have made in your design.

PHASE B (30 units)

Consider two basic scenaria, as follows:

A. All advertisers compete for a product of the same quality. As a consequence, the purchasing probability of a client will be the same for all advertisers.
B. Each advertiser offers the same product of different quality. In this case, the purchasing probability of a client will have a different value for each advertiser (For example, a product with a higher price should have a lower purchase probability than one with a lower price).

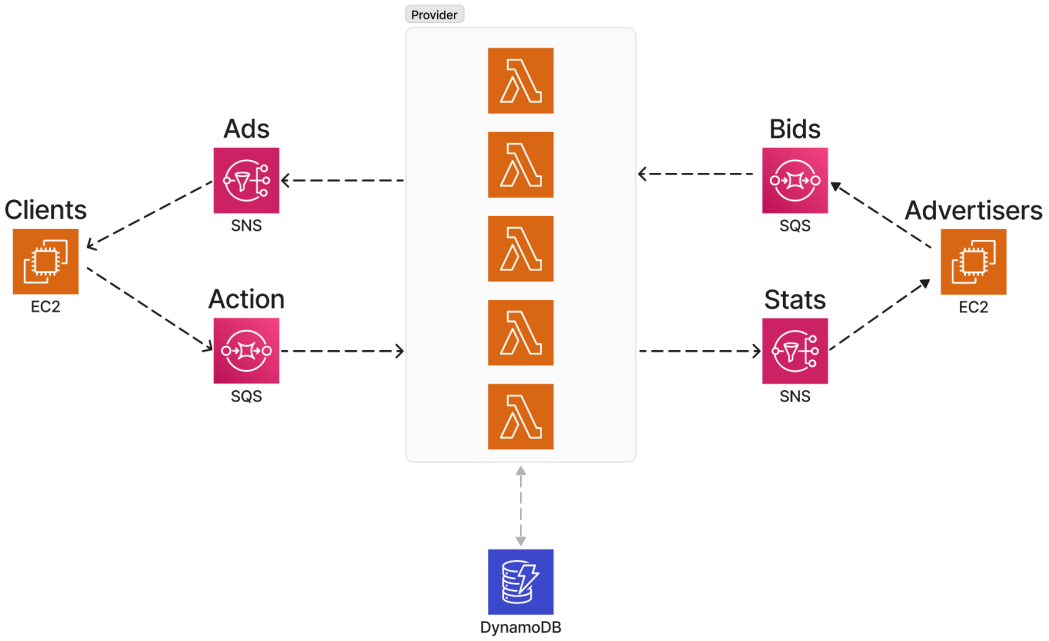
Tasks for both scenaria:

1. For the clients:
    a. Define the click probability for each position on the ranking list. Follow the rule: the higher probability is assigned to the higher position.
    b. Define the purchase probability for each advertiser.
2. For the advertisers:
    a. Define the price of the product.
    b. Derive the bidding strategy based on clicks and sales.

PHASE C (55 units)

Implementation

Implement the entities described above and their interactions using serverless technologies. A possible implementation is shown in the following figure:



In particular, you should implement the following considering that the game consists of 10 rounds, 3 advertisers and 10 clients:

C1. Implement the Ad Service Provider as a serverless web application with lambda functions that perform the following specific tasks using SNS, SQS and DynamoDB.

Task	Description
1	Get all the bid messages from the Bids SQS queue and store them to the DynamoDB table
2	Read all data from the DynamoDB table and decide which ads to show and in what order to increase revenues
3	Publish the ranking list to the clients using the Ads SNS topic
4	Get all client actions from the Action SQS queue and update data in the DynamoDB table
5	Publish all ad statistics to the advertisers using the Stats SNS topic

The **DynamoDB table** should contain the following fields (statistics):

Ad ID	Bid	Clicks	Sales

C2. Implement the Advertisers using an EC2 instance with a fixed number of processes describing the number of advertisers that perform the following tasks.

Tasks	Description
1	Publish the bid price to the provider using the Bids SQS queue
2	Receive all statistics from the Stats SNS topic
3	Calculate the new bid to increase profit

* Please see the note provided below

C3. Implement the Clients using an EC2 instance with a fixed number of processes describing the number of clients that perform the following tasks.

Tasks	Description
1	Upon receiving the ranking list from the Ads SNS topic, choose which ad to click
2	Publish the ID of the selected clicked ad and the purchase action (True or False) to the provider using the Action SQS queue

* Please see the note provided below

C4. Design the coordination and management of the service system described above and provide a detailed description of how the entities involved work together.

C5. Implement both scenarios described in Phase B and provide the following three graphs for each:

1. A graph that shows the bids of all advertisers vs. rounds
2. A graph that shows the profit of all advertisers vs. rounds
3. A graph that shows the revenues of the provider vs. rounds

Study the results and make your observations.

In order to interact with any of these services and lambda functions you have to use Amazon Web Services (AWS) SDK for Python (Boto).

Note that actions in case of 'Lambda Function Execution Failures' are indicative and should not be implemented as part of this homework.

BONUS QUESTION (10 units)

Define the clients' utility as a function of price. Derive the level of price that maximises a client's utility.

Note: Your code should print out the following messages:

1. For the clients:
    a. Game round
    b. Client ID
    c. Ad ID (which advertisement they clicked on)
    d. Whether they made a purchase or not (0 or 1)

    e.g.

    # Round 1
    Client ID: <...> , Ad ID: <...> , Purchase: <...>
    Client ID: <...> , Ad ID: <...> , Purchase: <...>
    …
    # Round 2
    ...

2. For the advertisers:
    a. Game round
    b. Ad ID
    c. Revenues
    d. Costs
    e. Profit
    f. Whether they increase their bid or not (from bid b to b')

    e.g.

    # Round 1
    Ad ID: <...> , Revenues: <...> , Cost: <...> , Profit: <...> , Increase: <...>, From: <...> , To: <...>
    Ad ID: <...> , Revenues: <...> , Cost: <...> , Profit: <...> , Increase: <…>, From: <...> , To: <...>
    …
    # Round 2
    …