REPORT HY452 - MINIPROJECT

IRODOTOS APOSTOLOU

csd4437

PHASE B

CLIENT SIDE: 1. click probability of each position in ranking list
1.1 Position 0: 60%
1.2 Position 1: 30%
1.3 Position 2: 10%
2. purchase probability of each advertiser
2.1 Scenario 1: Each advertiser have the same quality product => same price => 40% purchase probality
2.1 Scenario 2: Each advertiser have differnet quaility product => different price
Advertiser 0: 20% purchase probability
Advertiser 1: 40% purchase probability
Advertiser 2: 70% purchase probability

ADVERTISER SIDE: 1. price of the product
1.1 Scenario 1: Same quality product => same price => 12euros
1.2 Scenario 2: Different quality product => different price
Advertiser 0: 20euros sell his product
Advertiser 1: 12euros sell his product
Advertiser 2: 8euros sell his product
2. bidding strategy
Every advertiser check his clicks and his sales. If we have more than 40% sales in clicks then need to increase his bid beacuse that means that client buys a lot his product. Also if this is true and see that his clicks is less than 40% of total_clicks, then increase his bid again to get better position in ranking list. If we have less than 20% sales in clicks then need to decrease his bid beacuse that means that client dont buy often his product. Also if this is true check and his clicks is more than 40% of total clicks, then decrease again his bid beacuse that means not only clients dont buy his product but he is also very high in ranking list and doesnt need it. Between 20% - 40% advertisers do nothing beacuse its okay clicks and sales

PHASE C

IMPLEMENTATION: I use 2 EC2(clients , advertisers) instances to run the advertisers and client code , 2 SQS(Bids , Actions), 2 Lambda functions that is the provider(get_bids_and_publishThem_to_ads , get_actions_and_publishThem_to_stats) , 2 SNS topics (Ads , Stats) , and 2 tables(Bids , Actions)
1. Each advertises choose a random bid(1-5). After that i create a dictionary that contains for keys the id of each advertiser and as value an array with [round,clicks,sales,bid]. (initialy clicks and sales are -1) and send it to SQS(Bids)
2. After the dictionary store in SQS(Bids) it trigger an event in Lambda(get_bids_and_publishThem_to_ads). This Lamda store the dictionay in table(Bids) , order the dictionary in ascending order according bid and send publish it to SNS(Ads)
3. After the message publish in SNS the EC2(clients) get it beacuse is a subscriber to SNS via http. Each client decide which position of the ranking list is going to click and after that he decide if he is going to buy or not the product. After every click or sale i update the dictionary correctly and after all clients finished , i send the new updated dictionary in SQS(Actions)
4. After the updated dictionay store in SQS(Actions) it trigger an event in Lamvda(get_actions_and_publishThem_to_stats). This lamba store the updated dictionary in table(Actions) and publish it to SNS(Stats)
5. After the message publish in SNS(Stats) the EC2(Advertisers) get it beacuse is a subscriber to SNS via http. Each avdertiser check his clicks and sales and perform his bidding strategy. After that create a new dictionary like step1 with the new bid and store it in SQS(Bids). Now go back again to step2