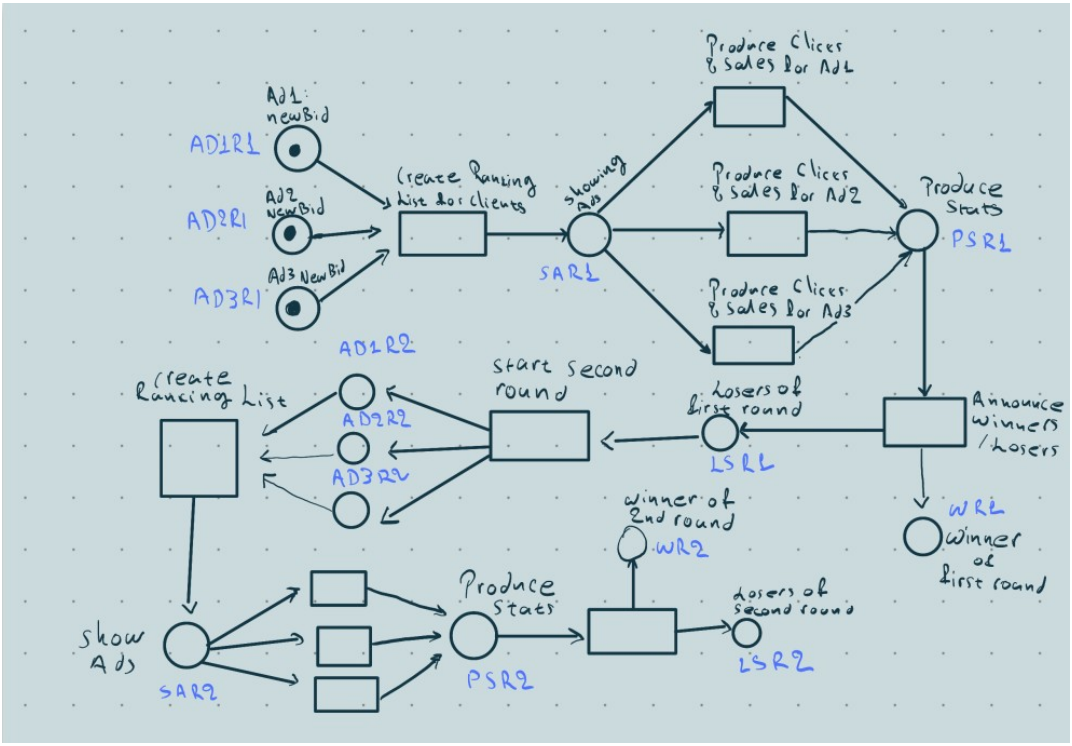
HY452 | Project

Papageorgiou Efthymios 4340

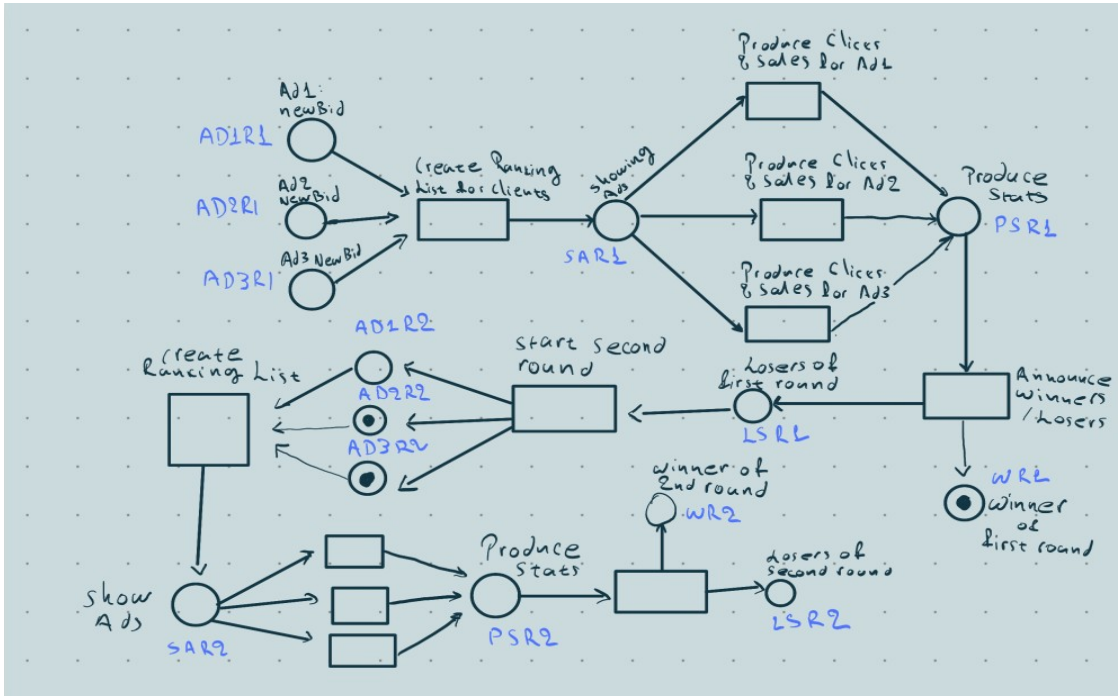
December, 2022

Phase A – Petri Net

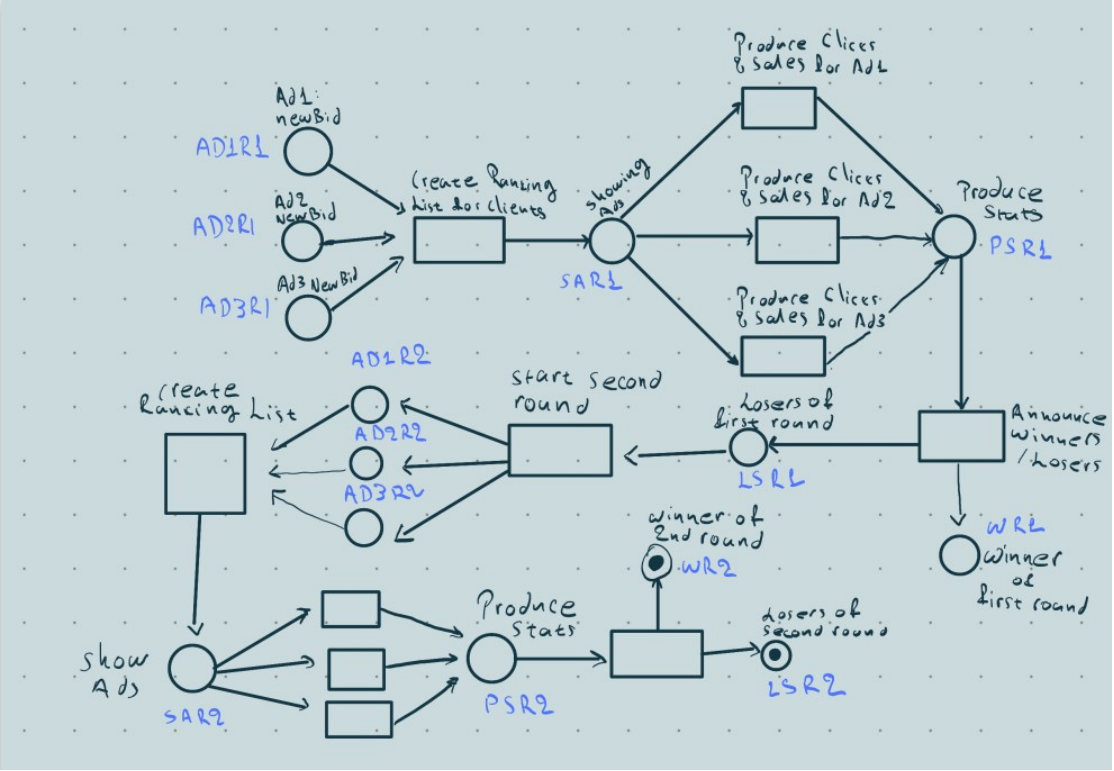
Initial State



Middle State



Final State



Descrioption of Petri Net

Η ιδέα είναι ότι έχουμε 3 token, και όχι 6 που ίσως εβγαζε περισσότερο νόημα.Ο λόγος που το σχεδίασα με το 3 token είναι για να να προκύψει διαφορετικός νικητής σε κάθε round.Αρχικά, τα tokens στα places Ad1R1, Ad2R1 και Ad3R1 μεταβένουν στο place SAR1 μέσω της κατάστασης CreateRankingListForClients.Στην συνέχεια κάθε token μεταβαίνει στο place PSR1 μέσω των καταστάσεων ProducrClicks&Sales for Ad{1,2,3}.Έπειτα, η κατάσταση AnnounceWinnderLosers οδηγεί τα tokens στα αντίστοιχα places.Η ιδέα είναι ότι ο winner δεν θα συνεχίσει για την δεύτερη φάση, διότι έχει κολλήσει στο place WinnerofFirstRound.Το statemachine συνεχίζει με 2 tokens αντί για 3.Ενα απο αυτά τα 2 tokens θα είναι ο επόμενος νικιτής.

Assumptions

1. Το transition AnnounceWinner/Losers είναι σε θέση να πετάξει το κερδοφόρο token στο place WinnerOfFirstRound και τα υπόλοιπα στο place LosersOfFirstRound

Rechability Graph

-

Phase B

Scenario A

Clients:

Click Probability: [0.60, 0.25, 0.15]

Purchase Probability: [0.75, 0.75 ,0.75]

Advertisers:

Price of Product: 299.99

Bidding stategy:

Αν τα sales είναι περισσότερα κατά τα μισά clicks τότε ρίχνω το bid μου κατά ¾,

αλλιώς ανεβάζω το bid κατά 3/2.

Scenario B

Clients:

Click Probability: [0.60, 0.25, 0.15]

Purchase Probability: [0.70, 0.20 ,0.10]

Advertisers:

Price of Product: 299.99

Bidding stategy:

Αν τα sales είναι περισσότερα κατά τα μισά clicks τότε ρίχνω το bid μου κατά ¾,

αλλιώς ανεβάζω το bid κατά 3/2.

Phase C

C1. [Done] C2. [Done] C3. [Done]

C4.

Η ίδεα είναι η εξής..

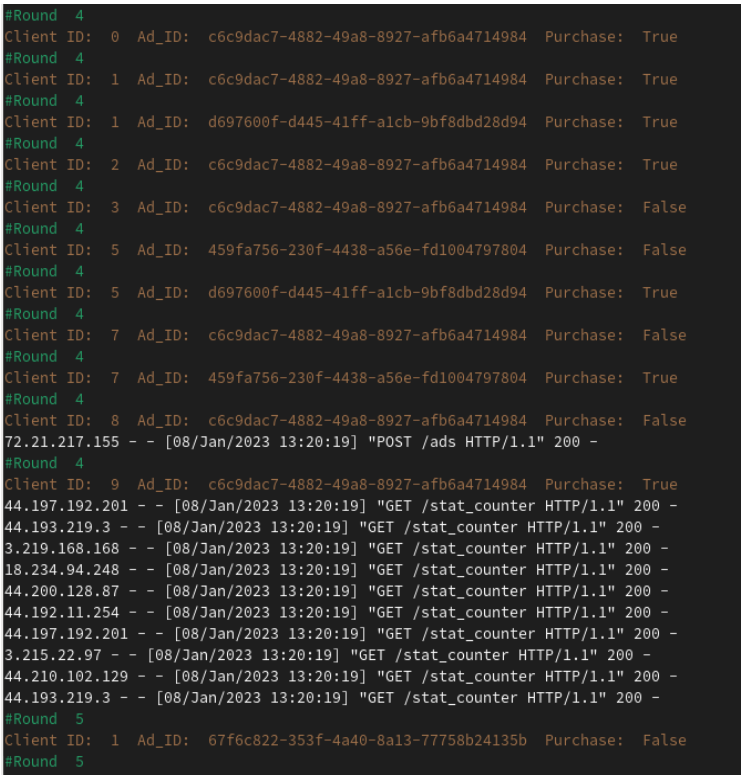
Από την μια μερία έχουμε έχουμε ένα VM (ec2) advertisers, οι οποίοι είναι 3 threads.Το advertisers.py μόλις τρέξει στέλνει ένα αίτημα εγγραφής στο Stats SNS.Το SNS στέλνει ένα μήνυμα επιβεβαίωσης στο http endopoint που έχει οριστεί μέσα στο ec2 με την βοήθεια του framework flask.Αφού γίνει το verification δημιουργούνται τα 3 threads και με την σειρά τους στέλνουν το bid τους στην ουρά Bids.Να σημειωθεί ότι μαζί με το bid το μήνυμα περιέχει το Round το οποίο έγινε αυτό το bid, και το όνομα του advertiser(Intel,AMD,Nvidia).Η ουρά μόλις λάβει μήνυμα κάνει trigger ένα lamda function του οποίου η δουλεία είναι να βάλει το bid, με την υπόλοιπη πληροφορία, στο Bid table της DynamoDB.Αφού το βάλει στέλνει ένα μήνυμα επιβεβαίωσης στο ec2.Μόλις το ec2 μαζέψει 3 τέτοια μηνύματα επιβεβαίωσης, κάνει publish ένα συγκεκριμένο μήνυμα σε ένα SNS(BidsAreReady) το οποίο υποδηλώνει ότι ‘δεν έχω άλλα bids να σου στείλω – μπορείς να τα τραβήξεις από το table και να φτιάξεις το ranking list σου’.Αφού φτιαχτεί το ranking list, γίνεται publish στο Ads SNS.Εδώ έρχονται οι clients (10 threads) σε δεύτερο ec2. Να σημειωθεί ότι το ec2 των clients έχει κάνει αντίστοιχη δουλεια με την εγγραφή στο SNS, την ώρα που ξεκίνησε το πρόγραμμα.Μόλις οι clients δουν τις διαφημισεις, παράγουν έναν αριθμό από clicks και sales.Κάθε client κάνει enqueue τα clicks και sales για κάθε διαφήμιση που είδε.Η αντίστοιχη αυτή ουρά Action κάνει trigger ένα lambda function το οποίο πάει και τρέχει ένα query στην βάση που αυξάνει τα clicks και sales της αντίστοιχης διαφήμισης.Να σημειώθει ότι το query είναι Atomic και έτσι δεν μπορούν να υπάρξουν race conditions και να χαλάσουν τον counter.Κάθε φόρα που το lambda κάνει update ένα item στην dynamodb στέλνει ένα μήνυμα επιβεβαίωσης σε συγκεκριμένο endpoint του ec2 των clients.Μόλις το ec2 των clients λάβει 10 τέτοια μηνύματα κάνει pub σε μια SNS(AllUpdatedAreReady) ένα μήνυμα που λέει ΄Δεν έχω να σου στείλω άλλα clicks / sales να κάνεις update, οπότε τράβα όλες τις διαφημίσεις του τρέχων γύρου και κάνετα pub στο SNS Stats. Το endpoint βλέπει το notification απο την Stats SNS και ανανεώνει τα local stats (player_stats dictionary) και έτσι η ιστορία επαναλαμβάνεται για άλλους 9 γύρους.

Before Run:

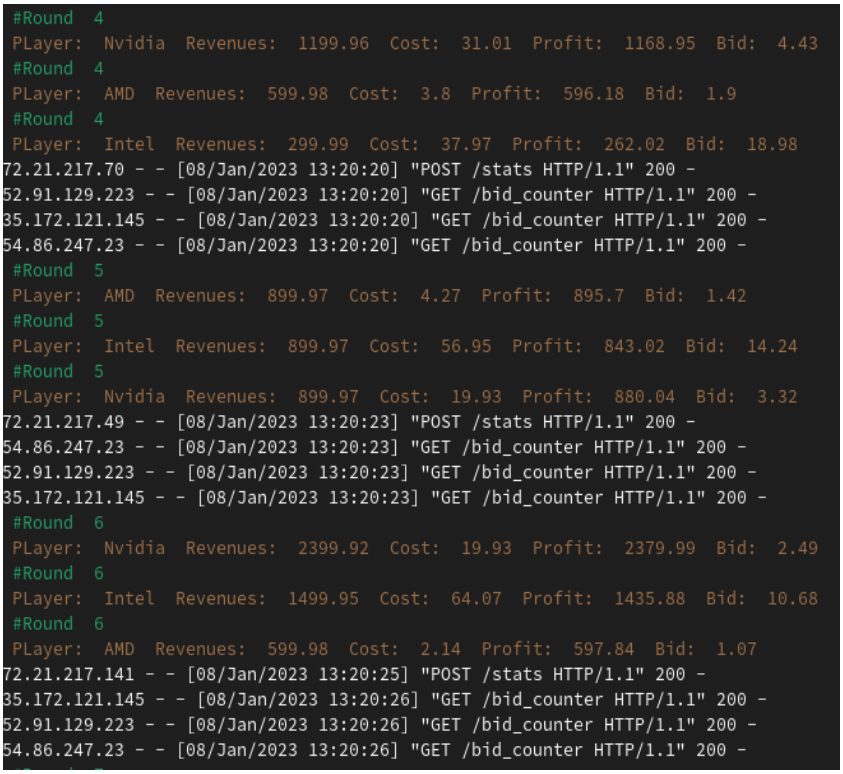
1. Στα python files των ec2s(advertiser.py, clients.py) θα πρέπει να εισαχθεί στις σταθερές advertisers_ipv4_dns και clients_ipv4_dns η αντίστοιχη ipv4_dns που δίνει η υπηρεσία.

2. Ipv4_dns θα πρέπει να εισαχθεί και τρεις lambdas.Η πρώτη είναι αυτή που ξεκινάει το Game στέλνοντας στους advertisers να ξεκινήσουν και οι άλλες 2 στις lambda που ενημερώνουν τα ec2s.

Screenshots of ***Round 4***



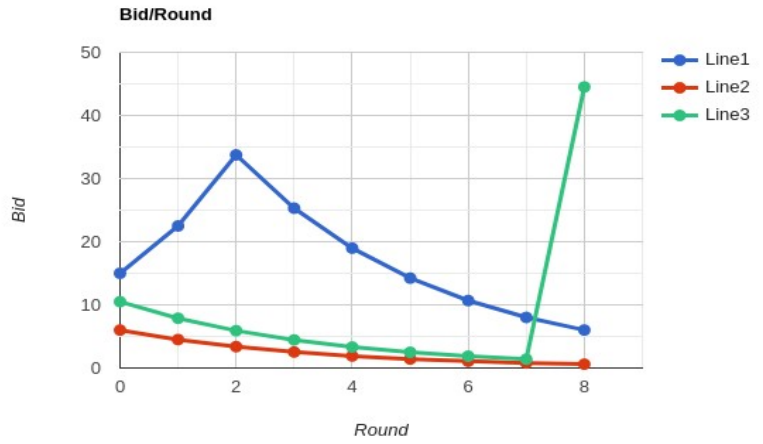
```
#Round 4
Client ID: 0 Ad_ID: c6c9dac7-4882-49a8-8927-afb6a4714984 Purchase: True
#Round 4
Client ID: 1 Ad_ID: c6c9dac7-4882-49a8-8927-afb6a4714984 Purchase: True
#Round 4
Client ID: 1 Ad_ID: d697600f-d445-41ff-a1cb-9bf8dbd28d94 Purchase: True
#Round 4
Client ID: 2 Ad_ID: c6c9dac7-4882-49a8-8927-afb6a4714984 Purchase: True
#Round 4
Client ID: 3 Ad_ID: c6c9dac7-4882-49a8-8927-afb6a4714984 Purchase: False
#Round 4
Client ID: 5 Ad_ID: 459fa756-230f-4438-a56e-fd1004797804 Purchase: False
#Round 4
Client ID: 5 Ad_ID: d697600f-d445-41ff-a1cb-9bf8dbd28d94 Purchase: True
#Round 4
Client ID: 7 Ad_ID: c6c9dac7-4882-49a8-8927-afb6a4714984 Purchase: False
#Round 4
Client ID: 7 Ad_ID: 459fa756-230f-4438-a56e-fd1004797804 Purchase: True
#Round 4
Client ID: 8 Ad_ID: c6c9dac7-4882-49a8-8927-afb6a4714984 Purchase: False
72.21.217.155 - - [08/Jan/2023 13:20:19] "POST /ads HTTP/1.1" 200 -
#Round 4
Client ID: 9 Ad_ID: c6c9dac7-4882-49a8-8927-afb6a4714984 Purchase: True
44.197.192.201 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
44.193.219.3 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
3.219.168.168 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
18.234.94.248 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
44.200.128.87 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
44.192.11.254 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
44.197.192.201 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
3.215.22.97 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
44.210.102.129 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
44.193.219.3 - - [08/Jan/2023 13:20:19] "GET /stat_counter HTTP/1.1" 200 -
#Round 5
Client ID: 1 Ad_ID: 67f6c822-353f-4a40-8a13-77758b24135b Purchase: False
#Round 5
```

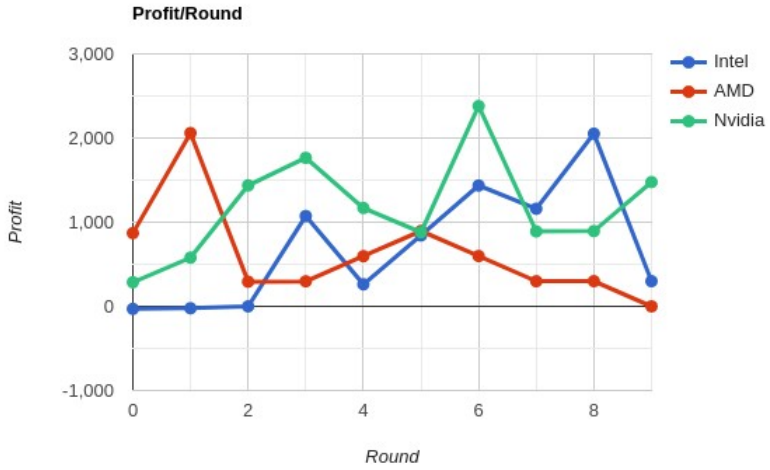


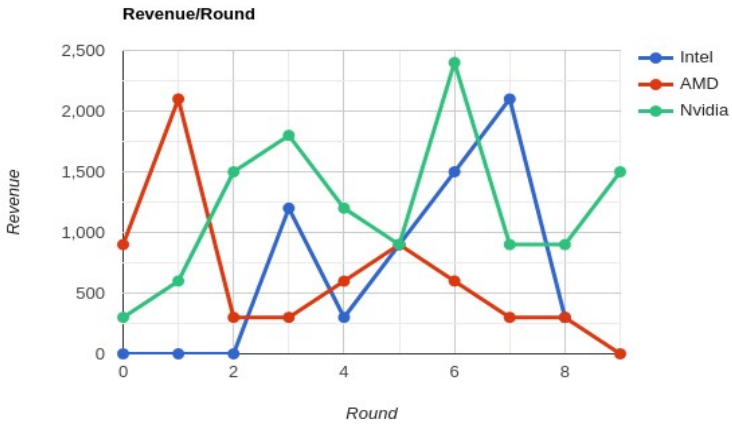
```
#Round 4
PLayer: Nvidia Revenues: 1199.96 Cost: 31.01 Profit: 1168.95 Bid: 4.43
#Round 4
PLayer: AMD Revenues: 599.98 Cost: 3.8 Profit: 596.18 Bid: 1.9
#Round 4
PLayer: Intel Revenues: 299.99 Cost: 37.97 Profit: 262.02 Bid: 18.98
72.21.217.70 - - [08/Jan/2023 13:20:20] "POST /stats HTTP/1.1" 200 -
52.91.129.223 - - [08/Jan/2023 13:20:20] "GET /bid_counter HTTP/1.1" 200 -
35.172.121.145 - - [08/Jan/2023 13:20:20] "GET /bid_counter HTTP/1.1" 200 -
54.86.247.23 - - [08/Jan/2023 13:20:20] "GET /bid_counter HTTP/1.1" 200 -
#Round 5
PLayer: AMD Revenues: 899.97 Cost: 4.27 Profit: 895.7 Bid: 1.42
#Round 5
PLayer: Intel Revenues: 899.97 Cost: 56.95 Profit: 843.02 Bid: 14.24
#Round 5
PLayer: Nvidia Revenues: 899.97 Cost: 19.93 Profit: 880.04 Bid: 3.32
72.21.217.49 - - [08/Jan/2023 13:20:23] "POST /stats HTTP/1.1" 200 -
54.86.247.23 - - [08/Jan/2023 13:20:23] "GET /bid_counter HTTP/1.1" 200 -
52.91.129.223 - - [08/Jan/2023 13:20:23] "GET /bid_counter HTTP/1.1" 200 -
35.172.121.145 - - [08/Jan/2023 13:20:23] "GET /bid_counter HTTP/1.1" 200 -
#Round 6
PLayer: Nvidia Revenues: 2399.92 Cost: 19.93 Profit: 2379.99 Bid: 2.49
#Round 6
PLayer: Intel Revenues: 1499.95 Cost: 64.07 Profit: 1435.88 Bid: 10.68
#Round 6
PLayer: AMD Revenues: 599.98 Cost: 2.14 Profit: 597.84 Bid: 1.07
72.21.217.141 - - [08/Jan/2023 13:20:25] "POST /stats HTTP/1.1" 200 -
35.172.121.145 - - [08/Jan/2023 13:20:26] "GET /bid_counter HTTP/1.1" 200 -
52.91.129.223 - - [08/Jan/2023 13:20:26] "GET /bid_counter HTTP/1.1" 200 -
54.86.247.23 - - [08/Jan/2023 13:20:26] "GET /bid_counter HTTP/1.1" 200 -
```

Ad_ID	Bid	Clicks	Player	Round	Sales
c6c9dac7-4882-49a8-...	4.4296875	7	Nvidia	4	4
d697600f-d445-41ff-...	1.8984375	2	AMD	4	2
459fa756-230f-4438-...	18.984375	2	Intel	4	1

C5. Scenario A - Graphs



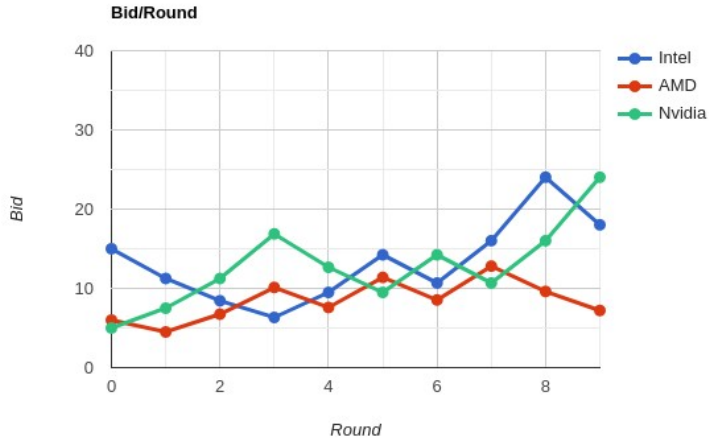


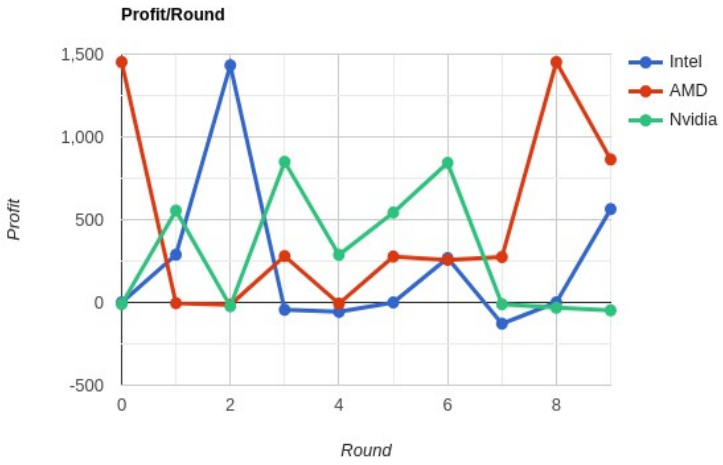


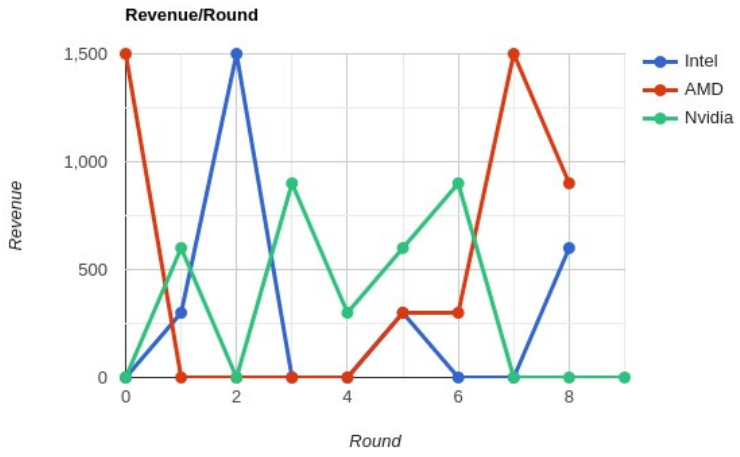
C5. Scenario A – Observations

Ξεκινώντας από το γράφημα Bids/round, φαίνεται ότι οι διαφημιστές συνεχώς ρίχνουν το bids – πλην 2 περιπτώσεις , αυτή της intel και της nvidia.Ο λόγως που το μειώνουν είναι γιατί παρατηρούν από τα στατιστικά, που τους έστειλε ο provider, ότι τα sales που κάνουν είναι περισσότερα ή ίσα από τα clicks που κάνουν οι πελάτες.Αυτό δίχνει την εικόνα στο advertiser ότι μπορούν να ρίξουν το bid του κατα ένα ποστοστό ουτοσώστε να μειώσουν το κόστος ανά κλικ και έτσι να αυξήσουν το καθαρό κέρδος του (profit).Το γράφημα των profits παρόλο μας δίνει μια εικόνα λίγο διαφορετική από αυτήν που περιμένα από το γράφημα των bids.Αρχικά, βλέπουμε ότι όσο ο advertiser ρίχνει το bid, το profit του έχει μια τάση να αυξάνεται κάτι το οποίο είναι και λογικό από την στιγμή που μειώνεται το κόστος που δίνει στο provider (CPC). Αξίζει να σημειωθεί ότι η πιθανότητα κάποιος να αγοράσει ένα από τα 3 προιόντα είναι η ίδια → ίδια ποιότητα.Αυτό μπορούμε να το παρατηρήσουμε και από το διάγραμμα των revenues.Η Nvidia ενώ έδινε μικρότερο bid από την intel πέτυχε υψηλότερο revenue σε κάθε, σχεδόν, round.Πιθανών, αν η nvidia έχει μικρότερο purchase probability από την intel να έχει και χαμηλότερο revenue.

C5. Scenario B – Graphs







C5. Scenario B – Observations

Στην υλοποιήση, η οποία ακολουθεί τα νούμερα της φάσης Β όπου αναφέρθηκαν πιο πάνω, η AMD έχει το μεγαλύτερο purchase probability = 0.70, ακολουθεί η Nvidia με 0.20 και η Intel με 0.10.

Ξεκινώντας από το γράφημα των bids φαίνεται ότι η AMD έχει μικρά bids κατά μέσω όρο από τις άλλες εταιρίες.Χωρίς να κοιτάξουμε τα άλλα γραφήματα θα λέγαμε ότι είναι λογικό αφού πιθανών να τα πάει κάλα με το profit της, δηλαδή αρκετά sales, άρα και υψηλό revenue.Παρατηρώντας το γράφημα τον profits φαίνεται ότι η Nvidia έχει περισσότερο ανά round κατά μέσο όρο από την AMD.Ένας λόγος είναι ότι η Nvidia έπαιζε μεγαλύτερο bid από την AMD κατα μέσο όρο σε όλα τα rounds και έτσι είχε μεγαλύτερη πιθανότητα για κλικ, όμως μικρότερη πιθανότητα για αγορά.Συνεπώς, βλέπουμε ότι η AMD δεν θα έπρεπε να έχει το bid τοσο χαμηλά γιατί ενώ έχει το προβάδισμα του purchase probality έχανε στα clicks.Γιαυτό φταίει ξεκάθαρα το bidding strategy που έχει οριστεί.

Bonus

U = 100 – 2P

Η παραπάνω συνάρτηση ποσοτικοποιεί την ευχαρίστηση του πελάτη από την ‘κατανάλωση/χρησιμοποίηση’ καρτών γραφικών.Όσο η τιμή αυξάνεται είναι λογικό ο πελάτης να είναι όλο και λιγότερο ευχαριστημένος.Η παραπάνω ευθεία θα μπορούσε να προσεγγιστεί καλύτερα με την βοήθεια του σεναρίου Β. Θα συλλέγαμε τα sales της intel για μια πιθανότητα αγοράς π.χ 0.90 και τα sales της intel για μια πιθανότητα αγοράς 0.10.Έχοντας τα 2 αυτά σημεία μπορούμε να προσεγγίσουμε το utility function του πελάτη από την κατανάλωση καρτών γραφικών της Intel.

Τέλος