



ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ 2020-2021 (*toyChord implementation*)

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΔΡΕΑΣ ΑΛΕΞΑΝΔΡΟΣ ΔΕΛΗΣ

03116649

ΑΓΓΕΛΑΚΟΠΟΥΛΟΣ ΑΘΑΝΑΣΙΟΣ

03116763

1)

Για τα ζητούμενα πείραματα ,χρησιμοποιήσαμε την βιβλιοθήκη random της python για να κάνουμε για κάθε αιτημα τυχαία επιλογή ip,port στις οποίες θα το προωθήσουμε .

Σημειώνεται οτι υλοποιήσαμε το linearizeability consistency με chain replication.

Αρχικά για τα 6 διαφορετικά setups κάναμε ολα τα insert που υπάρχουν στο infert.txt με κλήση python cli.py insertfile .Μέσω αυτής της κλήσης δημιουργείται και το αντιστοιχο αρχείο RepFXConsTypeY_insert.txt με X=1,3,5 και Y=0,1(0 για linearizeability ,1 για chain replication) με ολα τα αποτελέσματα των insert και τον συνολικό χρόνο.Τα αρχεία αυτα βρίσκονται στον φάκελο experiments.

Εδώ βλέπουμε τα διάφορα αποτελέσματα για το write throughput ανα setup:

Linearizeablity-k=1

$$\text{Write(throughput)} = 500 / 16.131470 = 30.9953154$$

Linearizeablity-k=3

$$\text{Write(throughput)} = 500 / 18.417123 = 27.1486486$$

Linearizeablity-k=5

$$\text{Write(throughput)} = 500 / 23.973432 = 20.8564214$$

Eventual Consistency k=1

$$\text{Write(throughput)} = 500 / 15.201276 = 32.8919756$$

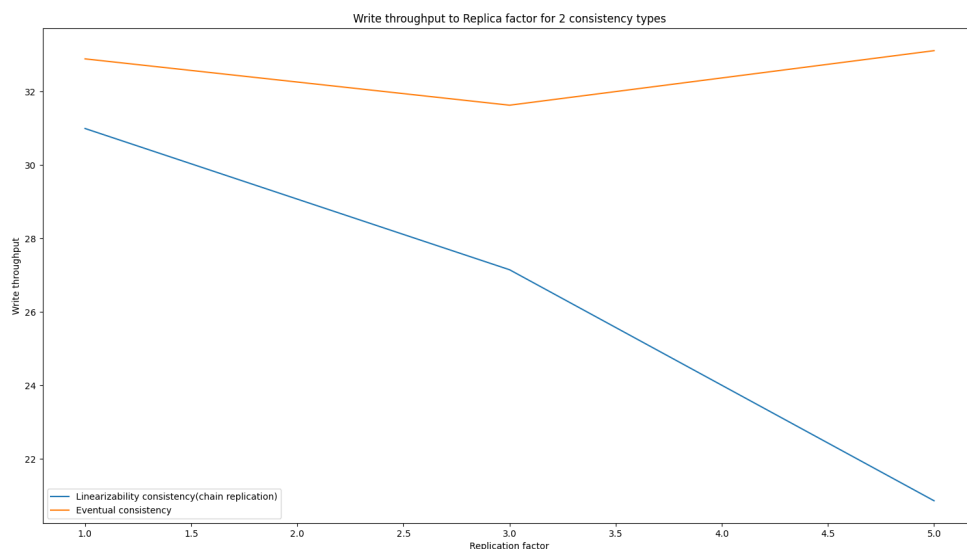
Eventual Consistency k=3

$$\text{Write(throughput)} = 500 / 15.807820 = 31.6299148$$

Eventual Consistency k=5

$$\text{Write(throughput)} = 500 / 15.098742 = 33.1153417$$

Παρακάτω βλέπουμε το αντιστοιχο plot, το οποίο παράχθηκε απο το script plotter.py που υπαρχει και αυτο στα αρχεια .



Εδώ βλέπουμε οτι για Linearizeability με την αυξηση του k το write throughput πέφτει όπως ειναι λογικό αφού για κάθε insert περιμένουμε απάντηση απο τον k-1 επόμενο κόμβο που ειναι ο τελευταιος που έχει το replica μας .

Απο την άλλη για Eventual Cons η αύξηση του k δεν επηρεάζει το throughput αφού όπως περιμέναμε ,τα insert πραγματοποιούνται κατα τον ίδιο τροπο αφού το αιτημα μας ολοκληρώνεται μόλις γίνει εισαγωγή στον υπευθυνο κόμβο και η εισαγωγή των replicas γίνεται ασυγχρονα απο αλλο process.

2)

Αντιστοιχα για τα 6 διαφορετικά setups κάναμε ολα τα query που υπάρχουν στο query.txt με κλήση `python cli.py queryarxio` .Μέσω αυτής της κλήσης δημιουργείται και το αντιστοιχο αρχείο

RepFXConsTypeY_query.txt με X=1,3,5 και Y=0,1(0 για linearizeability ,1 για chain replication) με ολα τα αποτελέσματα των insert και τον συνολικό χρόνο.Τα αρχεία αυτα βρίσκονται στον φάκελο.

Plot gia Linearizeability se sxesh me k kai read throughput

Linearizeability-k=1

Read(throughput)=500/12.910984=38.7267152

Linearizeability-k=3

Read(throughput)=500/18.565755=26.9313044

Linearizeability-k=5

Read(throughput)=500/24.009306=20.8252583

Eventual Consistency-k=1

Read(throughput)=500/13.865647=36.060344

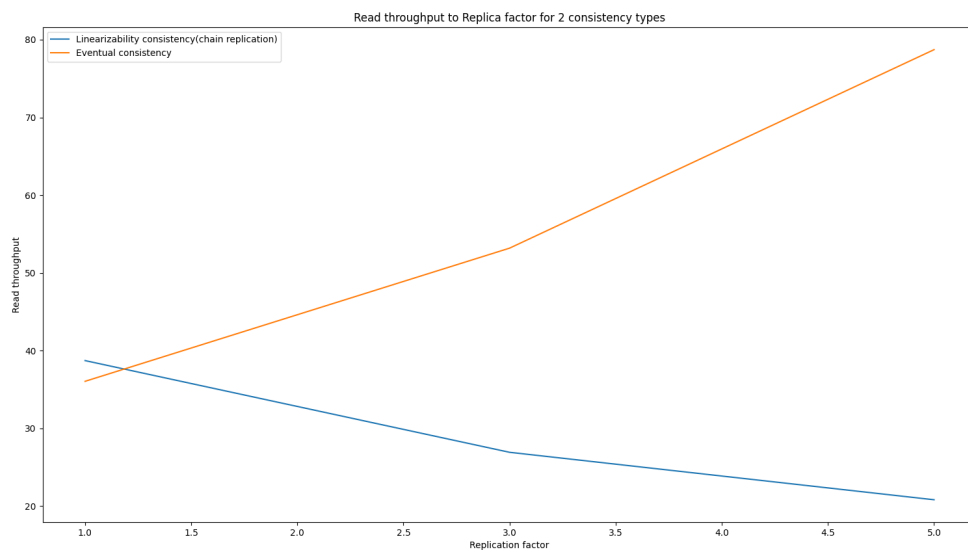
Eventual Consistency-k=3

$$\text{Read(throughput)} = 500 / 9.400489 = 53.1887224$$

Eventual Consistency-k=5

$$\text{Read(throughput)} = 500 / 6.351620 = 78.7200746$$

Παρακάτω βλέπουμε το αντιστοιχο plot, το οποίο παράχθηκε απο το script `plotter.py` που υπαρχει και αυτο στα αρχεια .



Εδώ οπως και πριν βλέπουμε οτι με Linearizeability με αυξηση του k το read throughput πεφτει αφού πρεπει να βρούμε τον υπεύθυνο κόμβο για το κλειδί και μετά να διαβάσουμε την τιμή του απο τον $k-1$ επόμενο γειτονα του που εχει το replica .

Αντίθετα για Eventual Cons με αυξηση του K ανεβαινει και το throughput αφού χρησιμοποιουμε οποια τιμή βρουμε πρώτη για ενα key ειτε σε replica ειτε σε responsible node.

3)

Στο συγκεκριμένο πείραμα τρέξαμε τα αιτηματα του request αρχιου για $\text{rep factor}=3$

και για τα δυο είδη consistency .Τα αποτελέσματα φαίνονται στα αντιστοιχα αρχεία RepF3ConsTypeX_request.txt X=0,1 .Συγκρίναμε τα δυο αρχεία με ένα python script που είναι στον φακελο(comparator.py) και είδαμε πως και στις δυο περιπτώσεις παίρνουμε fresh τιμές .Εφόσον λοιπον βεβαιώθηκαμε οτι έχουμε υλοποιησει σωστά τα δυο είδη consistency (άλλωστε αυτο φαίνεται απο τα αποτελέσματα στα 1,2) δοκίμασαμε να τρέξουμε με repfactor=7 και repfactor=10 για να δουμε αν το eventual consistency θα μας δώσει πιο παλίες τιμές χωρίς όμως να έχουμε τέτοιο αποτέλεσμα.Τα αντιστοιχα αρχεία των πειραμάτων είναι τα RepF7ConsType1_request.txt και RepF10ConsType1_request.txt καθένα απο τα οποία συγκρίναμε με το αρχείο RepF3constype0_request.txt .