

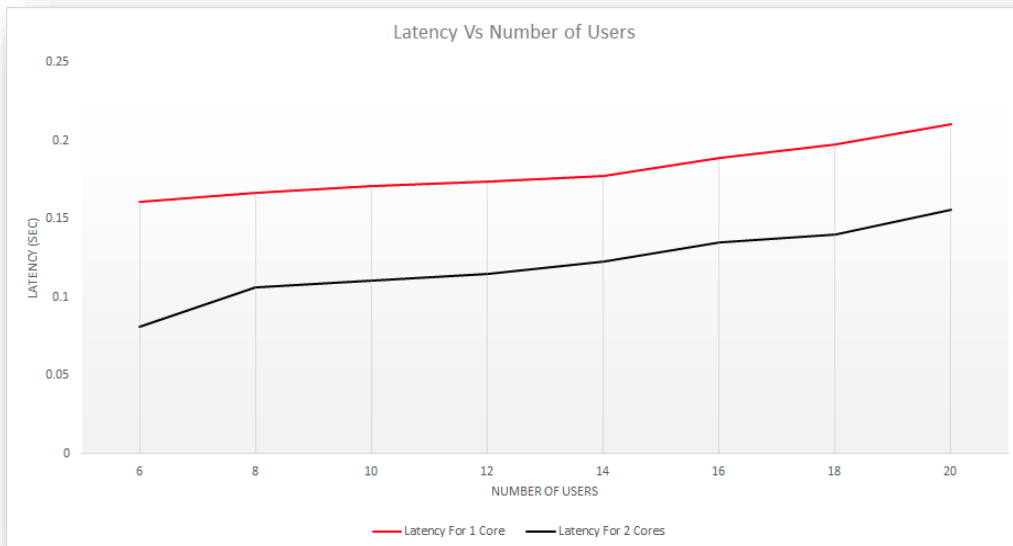
CS425 - Assignment 1:

Docs

Γενικές Οδηγίες:

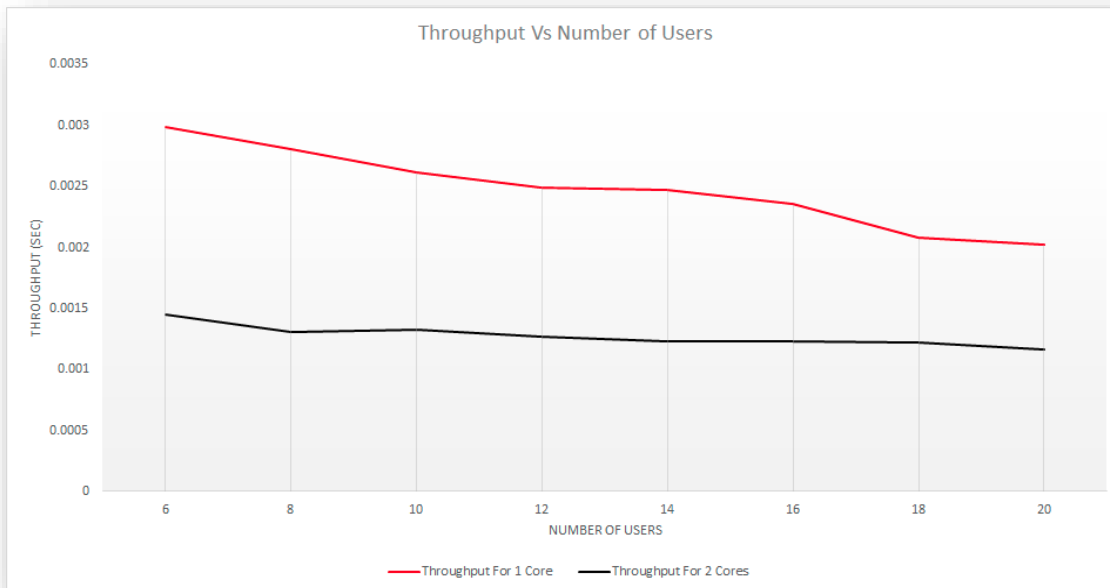
Υλοποιήσαμε δύο κλάσεις που αντιπροσωπεύουν τον Server και δύο που αντιπροσωπεύουν τους Clients. Η μία κλάση (ClientGenerator.java) είναι υπεύθυνη να δημιουργήσει N Clients αντικείμενα, όπου N είναι ο επιθυμητός αριθμός των πελατών. Η Client.java, είναι μία thread - class που δημιουργεί σύνδεση μέσω sockets με τον Server και κάνει 300 αιτήσεις στέλνοντας ένα απλό “Hello” μήνυμα μαζί με το IP και το port του user. Έπειτα η κλάση Server.java αναλαμβάνει να δεχθεί τις συνδέσεις και τα αναθέτει σε ServerThreads. Τα server threads παίρνουν τα αιτήματα, διαβάζουν το μήνυμα του πελάτη και απαντούν με ένα “Welcome” μήνυμα, το id του πελάτη και με payload size. Τέλος, το server thread στέλνει το payload.

Latency Vs Number of Users:



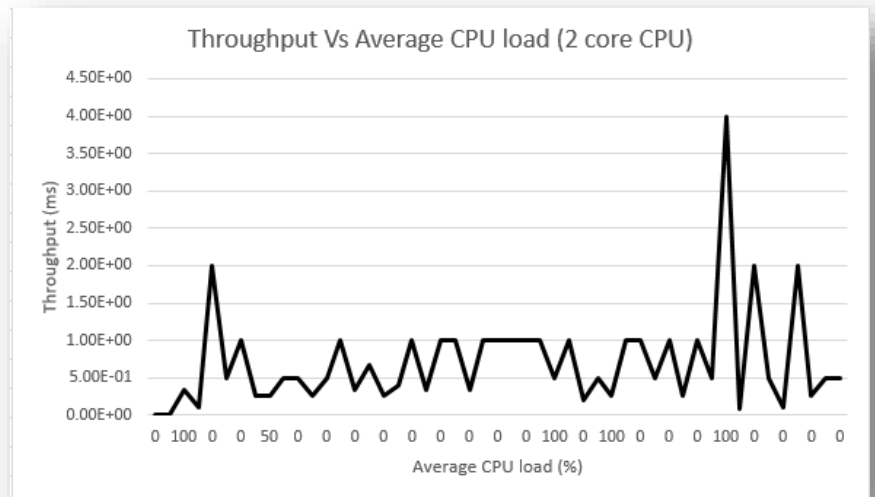
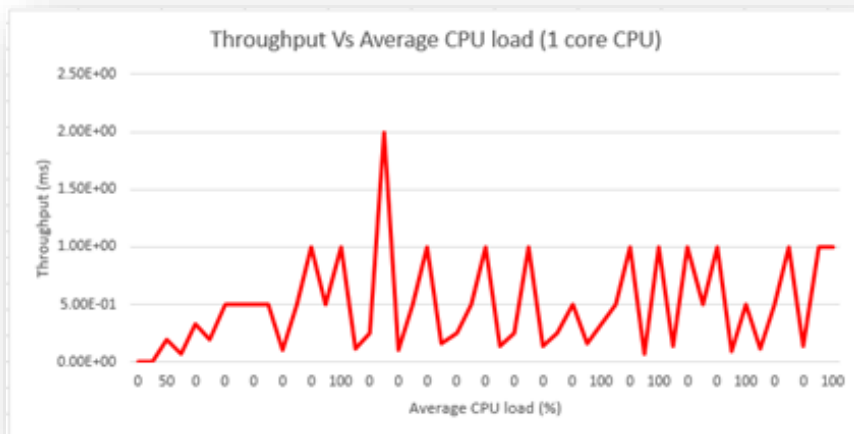
Στην πρώτη γραφική παρατηρούμε ότι όσο περισσότεροι πελάτες ενώνονται στον εξυπηρετητή τόσο περισσότερο latency έχουμε. Αυτό ήταν αναμενόμενο αφού οι πόροι του εξυπηρετητή πρέπει να μοιραστούν ανάλογα με τα αιτήματα που παίρνει, άρα υπάρχει καθυστέρηση. Η άλλη παρατήρηση ήταν ότι υπάρχει λιγότερο latency στο vm που είναι με 2 cores παρά στο vm με 1 core, όπου ήταν επίσης αναμενόμενη. Με περισσότερη επεξεργαστική δύναμη μπορεί ο εξυπηρετητής να αναλάβει και περισσότερες αιτήσεις ανα δευτερόλεπτο.

Throughput Vs Number of Users:



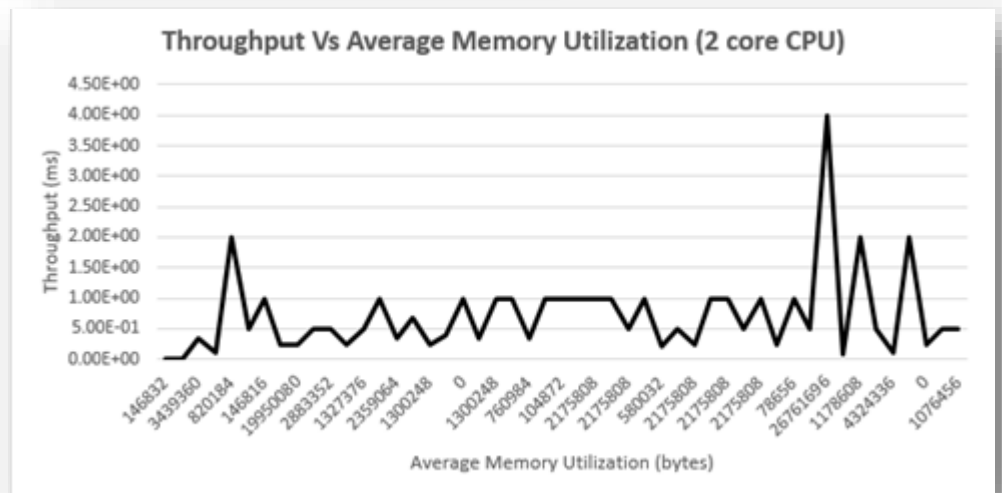
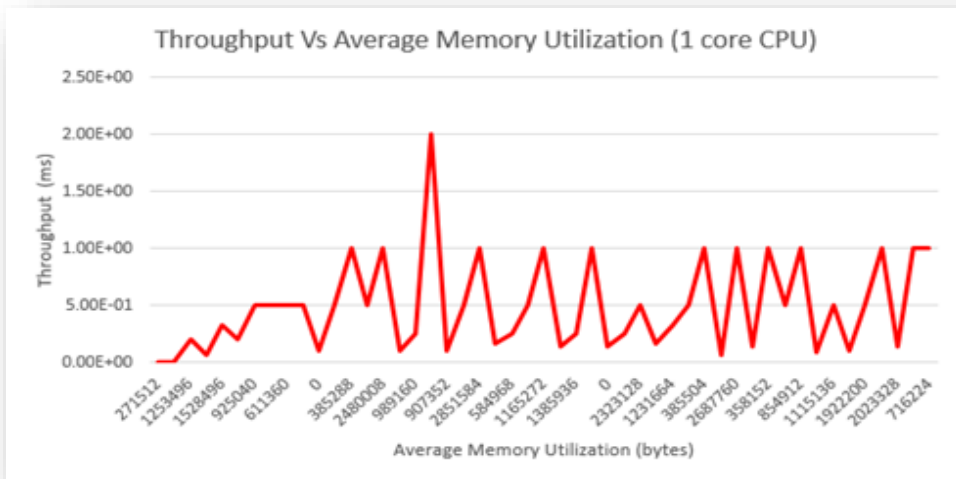
Στη γραφική αυτή παρατηρήσαμε ότι όσοι περισσότεροι πελάτες ενώνονται με τον εξυπηρετητή τόσο πιο μικρή είναι η τιμή που παίρνει το throughput. Αυτό συμβαίνει επειδή ανά δευτερόλεπτο ο εξυπηρετητής μπορεί να εξυπηρετήσει λιγότερα αιτήματα επειδή παίρνει πιο πολλά αιτήματα ταυτόχρονα.

Throughput Vs Average CPU load:



Αρχικά, παρατηρούμε ότι το throughput για 1 αλλά και για 2 cores CPU δεν παίρνει τις ίδιες τιμές. Το ίδιο συμβαίνει και για το average CPU load. Επίσης, στο σύνολο, όταν το πρόγραμμα του server τρέχει σε CPU με 2 cores παίρνει πιο πολλές φορές υψηλότερες τιμές για το average CPU load σε σύγκριση με τον 1 core CPU. Για τα δυο πιο πάνω γραφικές παραστάσεις παρατηρούμε ότι το CPU load παίρνει αρκετές φορές την τιμή 0, δηλαδή η CPU ούτε υπό-χρησιμοποιείται, αλλά ούτε υπερ- χρησιμοποιείται.

Throughput Vs Average Memory Utilization:



Αρχικά, το throughput για 1 αλλά και για 2 cores CPU δεν παίρνει τις ίδιες τιμές. Το ίδιο συμβαίνει και για το average memory utilization. Επίσης, παρατηρούμε ότι στο σύνολο, όταν το πρόγραμμα του server τρέχει για CPU με 2 cores το memory utilization παίρνει πιο μεγάλες τιμές σε σύγκριση με το CPU με 1 core.

Τέλος, μια γενική παρατήρηση είναι ότι, το πρόγραμμα για CPU με 2 cores αυτό έτρεχε πιο γρήγορα σε σχέση με το CPU με 1 core.