ΕΠΛ425:Εργασία 1

Ομάδα06: *Παντελίτσα Λεωνίδου, Κρίστια Χαριλάου*

Περιγραφή:

Το πρόγραμμα που υλοποιήσαμε ήταν η αποστολή μηνυμάτων από Client που τρέχει στον υπολογιστή μας σε απομακρυσμένο Server τον οποίο εγκαταστήσαμε στο AWS Amazon.

Ο Client στέλνει ένα Hello Message με το IP Address του, τον αριθμό χρήστη και το μέγεθος του αρχείου που ζητά. Ο Server δέχεται το αίτημα, δημιουργεί ένα byte array του μεγέθους που ζήτησε ο χρήστης και του το στέλνει.

Ο κάθε χρήστης στέλνει 300 μηνύματα. Ο Server περιμένει να επικοινωνήσουν μαζί του οι χρήστες.

Οι μετρήσεις που χρησιμοποιήσαμε για τα τις γραφικές ήταν:

* Ο αριθμός των Query που εξυπηρετεί σε 1000 milliseconds (Throughput)
* Ο χρόνος που χρειάζεται από την στιγμή της αποστολής του αιτήματος και της παραλαβής του μηνύματος.(Latency)
* Το Average CPU Load και το Average Memory Utilization ανά 1000 milliseconds

Για τα πειράματα του Latency, Throughput τρέξαμε για 5,10,15,20.25 users 10 repetitions.

Για τα πειράματα του Average CPU Load, Average Memory Utilization τρέξαμε για 15 χρήστες και 10 επαναλήψεις.

***Γραφικές Παραστάσεις:***

y:/Latency (in milliseconds), X:/Users

Οι μετρήσεις γίνονται για 5-25 πελάτες οι οποίοι στέλνουν παράλληλα στον εξυπηρετητή 300 requests περιμένοντας την απάντηση του. Το latency στην γραφική παρατηρούμε να αυξάνεται και να μειώνεται. Η αυξομείωση που παρατηρείται επηρεάζεται από 2 παράγοντες:

καθώς αυξάνονται οι πελάτες αν και αναμένουμε να αυξηθεί η καθυστέρηση και όταν είναι λιγότεροι να μειώνεται , ωστόσο δεν πραγματοποιείται αυτό αφού σημαντικό ρόλο στην καθυστέρηση εκτός από τον αριθμό τον πελατών, παίζει το μέγεθος του playload. Υπάρχουν δηλαδή περιπτώσεις όπου αν και οι πελάτες είναι ελάχιστοι ωστόσο σε κάθε αίτηση τους να ζητούν κάτι το οποίο να έχει μεγάλο μέγεθος (στην περίπτωση του προγράμματος μας το μέγεθος παράγεται τυχαία). Η δεύτερη περίπτωση είναι να ζητούν περισσότεροι πελάτες να εξυπηρετηθούν όμως το μέγεθος της αίτησης τους να είναι αρκετά μικρό. Ως αποτέλεσμα προκύπτει η πιο πάνω αυξομείωση στην καθυστέρηση των μηνυμάτων. Το πείραμα έχει γίνει με 1 και με 2 επεξεργαστές. Με δύο επεξεργαστές όπως αναμέναμε το latency είναι μικρότερο απ’ ότι είναι με ένα επεξεργαστή αφού τα αιτήματα εξυπηρετούνται πιο γρήγορα.

Y:/Throughput (Queries per 1000 milliseconds), X:/Users

Στο παραπάνω γράφημα παρατηρούμε ότι το Throughput ανάλογα με τον αριθμό των χρηστών μειώνεται και από ένα σημείο και έπειτα τείνει να σταθεροποιηθεί. Αυτό συμβαίνει γιατί καθώς ο αριθμός των χρηστών αυξάνεται θα αυξηθεί και ο αριθμός των αιτημάτων που ο εξυπηρετητής πρέπει να εξυπηρετήσει. Όμως όταν ο εξυπηρετητής δέχεται το μέγιστο των αιτημάτων που μπορεί υπολογιστικά να εξυπηρετήσει το throughput μειώνεται και ξεκινά να γίνεται σταθερό.

Παρατηρούμε επίσης ότι με 2 CPU το Throughput είναι μεγαλύτερο αφού ο εξυπηρετητής με μεγαλύτερη υπολογιστική δύναμη θα μπορεί να εξυπηρετεί μεγαλύτερο αριθμό αιτημάτων σε συγκεκριμένη μονάδα χρόνου.

y/ Average CPU load Percentage,x/Throughput(Queries per 1000 milliseconds)

Στη παραπάνω γράφημα παρατηρούμε το Average CPU Load ανάλογα με το Throughput για 15 χρήστες και 1 CPU.

Παρατηρούμε ότι κατά την αύξηση του Throughput αυξάνεται και το Average CPU Load.

Αυτό ήταν και το αναμενόμενο αφού όσα περισσότερα αιτήματα θα δέχετε ο server τόσo περισσότερο θα φορτώνεται ο επεξεργαστής. Μερικές αυξομειώσεις που παρατηρούνται πιθανόν να είναι μη ακριβείς μετρήσεις των συναρτήσεων που χρησιμοποιούνται.

y/ Average Memory Utilization Percentage,x/Throughput(Queries per 1000 milliseconds)

Στη παραπάνω γράφημα παρατηρούμε το Average Memory Utilization ανάλογα με το Throughput για 15 χρήστες και 1 CPU.

Παρατηρούμε ότι κατά την αύξηση του Throughput αυξάνεται και το Memory Utilization.

Αυτό ήταν και το αναμενόμενο αφού όσα περισσότερα αιτήματα θα δέχετε ο server τόση μεγαλύτερη μνήμη θα δεσμεύεται. Μερικές αυξομειώσεις που παρατηρούνται πιθανόν να είναι μη ακριβείς μετρήσεις των συναρτήσεων που χρησιμοποιούνται.