|  |  |
| --- | --- |
| ΕΠΛ425-ΤΕΧΝΟΛΟΓΙΕς ΔΙΑΔΙΚΤΥΟΥ  Ομαδική Εργασία 1(Client-Server) | Μαριάννα Χατζηδημητρίου (925889) Αναστασία Γιαννάκη (940871)  Ημερομηνία Παράδοσης: 8/10/2017 |

Πώς τρέχουμε το πρόγραμμα

javac MultiThreadedTCPServer.java

java MultiThreadedTCPServer port repetitions

* Τρέχουμε τον Server στο AWS server instance μας

Javac TCPClient.java

java TCPClient IP port

Σύντομη περιγραφή της άσκησης:

Για τον σκοπό της άσκησης αυτής δημιουργήσαμε 4 κλάσεις στην Java οι οποίες υλοποιούν τoν client και τον server, οι οποίοι επικοινωνούν μεταξύ τους με απλό «handshake» μήνυμα. Ο user αποστέλλει hello στον server μαζί με την διεύθυνση IP του, το port στο οποίο συνδέεται και ένα user id το οποίο γίνεται generate αυτόματα μέσω του προγράμματος μας. Στην συνέχεια ο server λαμβάνει το μήνυμα και αποστέλλει πίσω ένα μήνυμα καλωσορίσματος με το user id, και μια random μεταβλητή, την payload.

Οι 4 κλάσεις που υλοποιήσαμε έχουν ως εξής:

1. TCPClient

Η κλάση αυτή δημιουργεί ένα thread pool στο οποίο θα περιλαμβάνονται οι 10 threads όπου κάθε thread αντιστοιχεί σε ένα user τους οποίους δημιουργούμαι αυτόματα. Οι χρήστες αυτοί τρέχουν παράλληλα μέσω της συνάρτησης execute του threadpool η οποία με την υλοποίηση του thread καλεί αυτόματα την συνάρτηση run. Μετά το τέλος της επικοινωνίας με τον server, εφόσον τερματίσει ο executor, τυπώνουμε το συνολικό latency για όλους τους users.

1. ClientThread

Η κλάση αυτή αντιστοιχεί σε έναν χρήστη συγκεκριμένα, ο οποίος εκτελεί 300 requests σε διαφορετικό thread. Για κάθε χρήστη υπολογίζεται το latency του το οποίο καταχωρείται στον πίνακα rtt του αντικειμένου client.

1. MultiThreadedTCPServer

Αυτή η κλάση είναι υπεύθυνη για την δημιουργία του server μας ο οποίος ανταποκρίνεται σε ένα port το οποίο παίρνει σαν argument το πρόγραμμα μας. Επίσης δημιουργεί ένα ServerThread καλώντας την αντίστοιχη κλάση η οποία αποτρέπει τον τερματισμό επικοινωνίας ενός χρήστη με τον server επειδή άλλος χρήστης έχει κάνει το request του στον server.

Επίσης υπολογίζουμε το συνολικό throughput, το συνολικό Memory utilization του server και το συνολικό CPU load.

1. ServerThread

Τέλος, η κλάση αυτή αντιπροσωπεύει ένα server thread το οποίο ο κάθε user χρησιμοποιεί για να αποστείλει τα requests του. Υπολογίζουμε και το throughput, το Memory utilization του server και το CPU load για κάθε user ξεχωριστά.

Παρατηρήσεις και Συμπεράσματα από Γραφικές Παραστάσεις

Από τα στοιχεία που συλλέξαμε στο πρόγραμμα μας και μετά από την δημιουργία γραφικών παραστάσεων έχουμε φτάσει στα εξής συμπεράσματα. Αρχικά όσον αφορά το latency σε σχέση με τον αριθμό των χρηστών, παρατηρούμε ότι όσο οι χρήστες αυξάνονται το latency μειώνεται, λόγω παραλληλισμού. Όσον αφορά το latency δεν παρατηρήσαμε καμία διαφορά σε κανένα από τα instances του server που δημιουργήσαμε με διαφορετικό vCPU. Επίσης παρατηρήσαμε ότι το throughput είναι ανάλογο του αριθμού των χρηστών, αφού όταν οι χρήστες αυξάνονται, τότε και το throughput αυξάνεται. Αυτό ισχύει και για τα 3 διαφορετικά instances μας. Επιπλέον το memory utilization και το CPU load αυξάνονται όταν αυξάνεται το throughput. Τέλος το memory utilization για το τρίτο instance που δημιουργήσαμε με 4 vCPU είναι μικρότερο από τα άλλα instances, το οποίο είναι λογικό εφόσον έχει περισσότερη μνήμη.