



Cloud Computing & Performance Modeling

Deadline: 28/1/2023

Στην παρούσα εργασία θα ασχοληθούμε με μια σειρά από μηχανισμούς για την παρακολούθηση, διαχείριση και βέλτιστη εκτέλεση υπηρεσιών. Η εργασία αποτελείται από θεωρητικό και προγραμματιστικό μέρος. Θα παραδώσετε ένα report που περιλαμβάνει το θεωρητικό και το προγραμματιστικό ερώτημα. Για το προγραμματιστικό ερώτημα θα παραδώσετε και ένα notebook ή GitHub repo που θα περιλαμβάνει τον κώδικά σας με τα ενδιάμεσα αποτελέσματα.

Θεωρητικό Ερώτημα

Καλείστε να γράψετε ένα report (με αναφορές), στο οποίο θα περιγράψετε συνοπτικά τα ακόλουθα θέματα. Μια παράγραφος για κάθε θέμα είναι αρκετή:

- Task dependency graph
- Granularity στο parallel computing (λόγος του time computing προς το communication time)
- Embarrassing parallel workload
- Network Function Virtualization
- Intelligent Fault-Tolerant Mechanisms
- Workload Modeling

Προγραμματιστικό Ερώτημα

Καλείστε να μετρήσετε το resource utilization (i.e. CPU, RAM, BW) [\[1\]](#) [\[2\]](#) και το latency (i.e. response time, tail latency) κατά την εκτέλεση μιας εφαρμογής υπό διαφορετικά workloads με σκοπό να κάνετε performance modeling. Ο συνολικός φόρτος εργασίας θα είναι ο ίδιος αλλά θα αλλάζει το behaviour του workload [\[3\]](#). Φτιάξτε πινακάκια ή/και γραφικές παραστάσεις όπου οι γραμμές θα είναι τα διαφορετικού τύπου workloads και οι στήλες τα resource utilization και το latency. Γράψτε στο report σας: (α.) την διαδικασία του πειράματος, (β.) τα αποτελέσματα και (γ.) τα συμπεράσματα που έχετε. Μπορείτε να βρείτε configurations του application σας ή workload behaviours για τον βέλτιστο τρόπο εκτέλεσης της εφαρμογής;

Ακολουθούν 5 ενδεικτικά applications, αντίστοιχα με την επιλογή σας θα επιλέξετε τα διαφορετικού τύπου workloads.

1. **RESTful API.** Θα κάνετε request σε ένα RESTful API. Τα requests θα έρχονται σε batches διαφορετικού μεγέθους και συχνότητας [\[4\]](#).
2. **Multithreading/Multiprocessing.** Θα δοκιμάσετε σε ένα application την παράλληλη εκτέλεση νημάτων/processes με διάφορες τεχνικές [\[5\]](#) [\[6\]](#).



3. **SQL queries.** Σε μια σχεσιακή βάση δεδομένων θα δοκιμάσετε διάφορα queries που έρχονται σε batches διαφόρων μεγεθών και συχνότητας.
4. **NoSQL queries.** Σε μια NoSQL [7] βάση δεδομένων θα δοκιμάσετε διάφορα queries που έρχονται σε batches διαφόρων μεγεθών και συχνότητας.
5. **Inference/Training ANN.** Θα δοκιμάσετε το inference σε διαφορετικά batches δεδομένων. Μπορείτε να δοκιμάσετε και το training με διαφορετικό πλήθος mini-batches, epochs ή και άλλων υπερπαραμέτρων.

Ως application μπορείτε να χρησιμοποιήσετε έναν dummy sentiment analysis [8]. Σας παροτρύνω να χρησιμοποιήσετε ένα διαφορετικό application ή βάση δεδομένων της επιλογής, με την προϋπόθεση ότι η πίεση φόρτου εργασίας είναι σημαντική ούτως ώστε να έχουμε ενδιαφέροντα συμπεράσματα στα πειραματικά αποτελέσματα. Η εργασία αυτή μπορεί να επεκταθεί για να γίνει απαλλακτική εργασία σε συνεννόηση με τον διδάσκοντα.

[1] <https://psutil.readthedocs.io/en/latest/>

[2] <https://www.geeksforgeeks.org/how-to-get-current-cpu-and-ram-usage-in-python/>

[3] <https://iris.unipv.it/retrieve/e1f104fd-648d-8c6e-e053-1005fe0aa0dd/paper.pdf>

[4] 📄 lec_3. Flask_gr.pdf

[5] 📄 lec_4 Multithreading_gr.pdf

[6] 📄 Lesson Threads.ipynb

[7] 📄 lec_10 MongoDB_gr.pdf

[8] 📄 sentiment_analysis.ipynb