



ΠΜΣ “Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης”

Υπολογιστική Νέφους

Διδάσκοντες:

Καθ. Δουληγέρης Χρήστος

Αν. Καθ. Μιχάλας Άγγελος

Υ.δ. Καλλέργης Δημήτριος

Εργαστήριο:

Εικονικές Μηχανές και Μετρήσεις Απόδοσης

Γιώργος Πάνου

ΑΜ: ΜΠΣΠ18019

Email: panou.g@gmail.com

Β' εξάμηνο

Πειραιάς 2019

Πίνακας Περιεχομένων

1	Εισαγωγή.....	2
1.1	Μεθοδολογία.....	2
1.2	Η Cloud υπηρεσία ~Okeanos.....	3
1.3	Το λειτουργικό σύστημα CentOS.....	4
1.4	Επιμέρους δυνατότητες των μηχανών.....	4
1.4.1	Στοιχεία Επεξεργαστών	5
1.4.2	Στοιχεία Μνήμης RAM.....	6
1.4.3	Στοιχεία δίσκων.....	7
1.5	Προετοιμασία μηχανών	8
1.5.1	Εγκατάσταση Apache	8
1.5.2	Ρύθμιση firewall.....	8
1.5.3	Βοηθητικά Εργαλεία.....	9
2	Δοκιμές Απόδοσης (Benchmarking)	10
2.1	Αποτελέσματα Μετρήσεων.....	10
2.1.1	Gzip.....	10
2.1.2	ApacheBench	11
2.1.3	PHPBench.....	13
2.2	Υποστηρικτικός Κώδικας	16
2.2.1	Gzip script	16
2.2.2	ApacheBench script	16
2.2.3	PHPBench script	16
2.3	Προβλήματα Μετρήσεων.....	16
2.4	Machine Learning on Cloud Platforms	17
2.4.1	Hardware Specifications	18
3	Συμπεράσματα	20

Πίνακας Γραφημάτων

Γράφημα 1.	Χρόνος εκτέλεσης Gzip	11
Γράφημα 2.	Συνολικός χρόνος εκτέλεσης ApacheBench	12
Γράφημα 3.	Μέγιστος χρόνος εκτέλεσης ApacheBench	13
Γράφημα 4.	Συνολικό Αποτέλεσμα PHPBench	15
Γράφημα 5.	Χρόνος εκπαίδευσης Machine Learning μοντέλου	19

Πληροφοριακοί Πίνακες

Πίνακας 1.	Σύνθεση εικονικών μηχανών	5
Πίνακας 2.	Προδιαγραφές επεξεργαστών	5
Πίνακας 3.	Προδιαγραφές Μνήμης	7
Πίνακας 4.	Προδιαγραφές Δίσκων	7
Πίνακας 5.	Αναλυτικά στοιχεία ελέγχων PHPBench.....	14
Πίνακας 6.	Ενδεικτική εσφαλμένη λειτουργία του ApacheBench λόγω προβλημάτων δικτύου	17
Πίνακας 7.	Σύνθεση μηχανών δοκιμής Machine Learning Training	18

Έξοδος Κονσόλας

Έξοδος Κονσόλας 1.	Εγκατάσταση και παραμετροποίηση Apache	8
Έξοδος Κονσόλας 2.	Ρύθμιση Firewall.....	9
Έξοδος Κονσόλας 3.	Παρακολούθηση φόρτου επεξεργαστή κατά τη διάρκεια του PHPBench	15
Έξοδος Κονσόλας 4.	Gzip script.....	16
Έξοδος Κονσόλας 5.	ApacheBench script	16
Έξοδος Κονσόλας 6.	PHPBench script.....	16

1 Εισαγωγή

Με την άνθιση των cloud based infrastructure υπηρεσιών έχει προκύψει η ανάγκη αξιολόγησής τους από πλευράς απόδοσης προσφερόμενων υπηρεσιών και κόστους. Οι cloud υπηρεσίες προσθέτουν ένα επίπεδο αφαίρεσης πάνω από το hardware που, αν και αποτελεί το συγκριτικό τους πλεονέκτημα και τον κύριο λόγο migration σε αυτές, δημιουργεί αδιαφάνεια ως προς τη σύνθεση του υποκείμενου εικονικού hardware (underlying hardware). Αυτό συμβαίνει διότι δεν είναι πάντοτε σαφές από τους παρόχους ποια είναι η σύνθεση των εικονικών συσκευών (virtual hardware) των εικονικών μηχανών που προσφέρουν.

Σαν αποτέλεσμα είναι εξαιρετικά δύσκολο να αξιολογήσουν οι ενδιαφερόμενοι τις εκάστοτε υπηρεσίες μελετώντας μόνο τα datasheets των προδιαγραφών των παρόχων. Η αξιολόγηση μέσω benchmarking είναι μια μέθοδος που μπορεί να δώσει μια πλήρη εικόνα όχι σε επίπεδο χαρακτηριστικών του hardware αλλά στο τελικό επίπεδο των αποτελεσμάτων χρήσης του. Η διαδικασία αυτή ωστόσο κοστίζει σε εργατοπροσπάθεια και ενέχει κινδύνους αν δεν αναλυθεί και αξιολογηθεί ως προς την πληρότητά της και τις μεθόδους της.

Επίσης οφείλει να είναι προσαρμοσμένη στις ανάγκες των συστημάτων τα οποία θα γίνουν migrate. Στις περισσότερες περιπτώσεις κρίνεται απαραίτητο να μετρηθούν τα επίπεδα απόδοσης / τιμής που σχετίζονται με τις συγκεκριμένες ανάγκες των εφαρμογών προς migration. Ένα web application έχει διαφορετικές απαιτήσεις σε virtual (η και μη) hardware από ένα σύστημα log analyzer όπου οι απαιτήσεις σε παράλληλη εξυπηρέτηση χρηστών είναι μειωμένες.

Στην παρούσα εργασία έγινε μια προσπάθεια αξιολόγησης της υπηρεσίας ~Okeanos με προσανατολισμό τα Web Applications. Χρησιμοποιήθηκαν τα εργαλεία Gzip, ApacheBench και PHPBench. Το Gzip παρέχει μια αξιολόγηση κατά κύριο λόγο του δίσκου αλλά και δευτερευόντως της CPU σε single-threaded mode και της μνήμης. Το ApacheBench μπορεί να μετρήσει πόσο καλά αποκρίνεται ένας Web Server σε πολλά ταυτόχρονα requests/responses. Το PHPBench προσομοιώνει τις συνθήκες λειτουργίας του Backend ενός Application Server.

Επιπρόσθετα των απαιτούμενων δοκιμών έγιναν και μετρήσεις απόδοσης σχετικές με την εκπαίδευση Machine Learning μοντέλων στην υπηρεσία ~Okeanos, στην υπηρεσία Google Collab και σε consumer computer.

1.1 Μεθοδολογία

Έγινε προετοιμασία των μηχανών για τις δοκιμές η οποία περιλάμβανε την εγκατάσταση του software αλλά και την ρύθμιση του firewall ώστε να επιτρέπεται η απαραίτητη κίνηση προς αυτές. Έπειτα εγκαταστάθηκε και ρυθμίστηκε το software που απαιτείται για την εκτέλεση των benchmarks. Καταγράφηκε τόσο το virtual hardware configuration όσο και τα στοιχεία του ώστε να υπάρχει ως μέτρο σύγκρισης.

Δημιουργήθηκαν bash scripts τα οποία αναλαμβάνουν να εκτελέσουν τα benchmarking εργαλεία και ανακατευθύνουν την έξοδο από το τερματικό σε αρχείο, ώστε να είναι ευκολότερη η εξαγωγή των δεδομένων των αποτελεσμάτων.

Για το εργαλείο μέτρησης web server: ApacheBench επιλέχθηκε το benchmarking tool να εκτελεστεί σε ξεχωριστό μηχάνημα ώστε να μην επηρεαστεί η μέτρηση από το overhead της εκτέλεσής του.

Σε πρώτο στάδιο ελέγχθηκε το idle state της μηχανής με τις εντολές htop και free m, ώστε να επιβεβαιωθεί ότι δεν τρέχει κάποια άλλη διεργασία η οποία καταναλώνει πόρους μνήμης ή χρόνο στον επεξεργαστή. Οι εντολές αυτές χρησιμοποιήθηκαν και κατά την διάρκεια του τεστ ώστε να υπάρχει εποπτεία της κατανάλωσης πόρων των μηχανών.

Τέλος έγινε εξαγωγή των αποτελεσμάτων από τα αρχεία κειμένου και μεταφορά σε αρχείο xls ώστε να δημιουργηθούν γραφήματα και να είναι ευπαρουσίαστα.

1.2 Η Cloud υπηρεσία ~Okeanos

Η πλατφόρμα ~Okeanos παρέχει υπολογιστικές υπηρεσίες Cloud Computing με τη μορφή δημόσιας υποδομής-ως-υπηρεσία (Infrastructure as a Service), και έχει αναπτυχθεί και υποστηρίζεται από την Grnet. Ο ~Okeanos παρέχει ανάμεσα σε άλλα τα εξής:

- δημιουργία εικονικών μηχανών,
- δημιουργία εικονικών δικτύων
- δυνατότητα αποθήκευσης δεδομένων
 - είτε σε εικονικούς δίσκους
 - είτε σε εικονικό αποθηκευτικό χώρο στο Cloud.
- στατική public IP και
- URL

Οι υπηρεσίες αυτές είναι διαθέσιμες για την ακαδημαϊκή κοινότητα μέσω διαδικτυακής εφαρμογής, που αποτελεί ένα κεντρικό σημείο διαχείρισης και provisioning των ανωτέρω πόρων. Οι συνολικοί πόροι που υπήρχαν διαθέσιμοι στον λογαριασμό μου ήταν οι εξής:

- 8 virtual CPUs
- 8 GB RAM
- 40 GB Storage

1.3 Το λειτουργικό σύστημα CentOS

Το λειτουργικό σύστημα που επιλέχθηκε για την πραγματοποίηση των μετρήσεων είναι το CentOS 7. Είναι ένα προϊόν της οικογένειας Linux και το όνομά του είναι ακρωνύμιο που σημαίνει Community Enterprise Operating System. Η άδεια χρήσης του είναι open source και διατίθεται δωρεάν. Το CentOS παράγεται αντλώντας κώδικα από το Red Hat Enterprise Linux (RHEL) της Red Hat.

Το διοικητικό συμβούλιο του CentOS απαρτίζεται από μέλη του CentOS project, πολλά από τα οποία είναι ιδρυτικά, καθώς και από νέα μέλη από τη Red Hat, τα οποία συμβάλλουν στο εγχείρημα. Το διοικητικό συμβούλιο επικεντρώνεται στην υλοποίηση και προώθηση του CentOS.

Το CentOS είναι ένα λειτουργικό που χρησιμοποιείται ευρέως από επαγγελματίες ως πλατφόρμα για την εγκατάσταση enterprise grade software, open source ή και paid. Είναι ιδιαίτερα ευέλικτο όσο αφορά τα πακέτα που θα προ-εγκατασταθούν και απαιτεί ελάχιστα resources όταν επιλεγεί το προφίλ εγκατάστασης minimal. Το προφίλ αυτό παρέχει μόνο τα ελάχιστα απαραίτητα στοιχεία που μπορεί να απαιτηθούν για την εγκατάσταση εφαρμογών και σαν αποτέλεσμα εξοικονομεί πόρους από το hardware, είτε αυτό είναι εικονικό είτε φυσικό.

Για την εκπόνηση της εργασίας επιλέχθηκε το minimal προφίλ που δεν παρέχει γραφικό περιβάλλον εργασίας – η διεπαφή με το σύστημα γίνεται κατά αποκλειστικότητα μέσω του τερματικού. Εγκαταστάθηκαν κάποια tools (όπως ο nano text editor) ώστε να διευκολυνθεί η διαχείρισή του.

1.4 Επιμέρους δυνατότητες των μηχανών.

Για την υλοποίηση των μετρήσεων των cloud based εικονικών μηχανών αξιοποιήθηκε η πλατφόρμα ~Okeanos. Στα πλαίσια αυτής της εργασίας μετρήθηκε η απόδοση δύο διαφορετικών τύπων (configurations) εικονικών μηχανών της υπηρεσίας ~Okeanos, των οποίων η σύνθεση ήταν η εξής:

Configuration 1:	
vCPUS	8 x 2.1 GHz
RAM	8 GB
HDD	20 GB
OS	CentOS

Configuration 2:	
vCPUS	4 x 2.1 GHz
RAM	4 GB

HDD	20 GB
OS	CentOS

Πίνακας 1. Σύνθεση εικονικών μηχανών

Ακολουθούν αναλυτικότερα στοιχεία σχετικά με τις εικονικές μηχανές τα οποία, για λόγους απλότητας και καθώς χρησιμοποιώντας την ίδια πλατφόρμα δεν παρατηρήθηκε διαφορά στο virtual hardware, παρουσιάζονται κοινά και για τις δύο.

1.4.1 Στοιχεία Επεξεργαστών

Αξιοποιώντας την εντολή `cat /proc/cpuinfo`, μπορούμε να αντλήσουμε δεδομένα σχετικά με τα στοιχεία των επεξεργαστών. Από ότι παρατηρούμε οι διαθέσιμοι επεξεργαστές είναι χρονισμένοι στα 2100 MHz και έχουν cache 512 KB. Παρόλα αυτά επειδή το επίπεδο αφαίρεσης που δημιουργεί η υπηρεσία υπολογιστικού νέφους δεν μας επιτρέπει να πάρουμε παραπάνω πληροφορία από τον κατασκευαστή του επεξεργαστή (AMD) και τον generic driver “QEMU Virtual CPU version 2.1.2” δεν γνωρίζουμε στοιχεία όπως Instructions Per Clock ή TDP (Thermal design power).

vendor_id	AuthenticAMD
cpu family	6
model	6
model name	QEMU Virtual CPU version 2.1.2
stepping	3
microcode	0x1000065
cpu MHz	2099.998
cache size	512 KB
core id	0
cpu cores	1
cpuid level	4
wp	yes
flags	fpu de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx lm art nopl pni cx16 x2apic popcnt hypervisor lahf_lm svm abm sse4a
bogomips	4199.99
TLB size	1024 4K pages
clflush size	64
cache_alignment	64
address sizes	40 bits physical, 48 bits virtual

Πίνακας 2. Προδιαγραφές επεξεργαστών

1.4.2 Στοιχεία Μνήμης RAM

Τρέχοντας την εντολή `cat /proc/meminfo` αντλούμε τόσο τρέχοντα στοιχεία της μνήμης όσο και στατικά, όπως το `page size`. Μια κρίσιμη πληροφορία είναι το `SwapTotal` το οποίο δείχνει το μέγεθος του τμήματος του δίσκου που χρησιμοποιείται ως μνήμη (στο παράδειγμα 0). Το `Swap space` χρησιμοποιείται σε οριακές περιπτώσεις όταν όλη η διαθέσιμη μνήμη χρησιμοποιείται ως ένας τρόπος να αυξηθεί η συνολική διαθεσιμότητα ώστε να μην χρειαστεί σταματήσει βίαια κάποια εφαρμογή ή (όπως είναι πιθανό) όλο το λειτουργικό σύστημα.

Παρατηρούμε ότι η διαθέσιμη μνήμη (`MemAvailable`) είναι η ποσότητα που είναι συνολικά διαθέσιμη για τις εφαρμογές χωρίς να υπολογίζεται η δεσμευμένη από το λειτουργικό. Ένα άλλο ενδιαφέρον στοιχείο όπως προκύπτει από τα παρακάτω δεδομένα είναι το μέγεθος της `cached` μνήμης. Η `cached` μνήμη μπορεί να γίνει `overwritten`, όμως επηρεάζει έντονα την ποσότητα της ελεύθερης μνήμης (`MemFree`). Όπως παρατηρούμε η `cached` και η ελεύθερη μνήμη αθροίζονται στο σύνολο της διαθέσιμης μνήμης (`MemAvailable`)

MemTotal	8009564 kB
MemFree	2397256 kB
MemAvailable	7076728 kB
Buffers	238768 kB
Cached	4770440 kB
SwapCached	0 kB
Active	2063120 kB
Inactive	3035572 kB
Active(anon)	295608 kB
Inactive(anon)	197464 kB
Active(file)	1767512 kB
Inactive(file)	2838108 kB
Unevictable	0 kB
Mlocked	0 kB
SwapTotal	0 kB
SwapFree	0 kB
Dirty	76 kB
Writeback	0 kB
AnonPages	89496 kB
Mapped	52508 kB
Shmem	403576 kB
Slab	429836 kB
SReclaimable	379816 kB
SUnreclaim	50020 kB
KernelStack	2352 kB

PageTables	8052 kB
NFS_Unstable	0 kB
Bounce	0 kB
WritebackTmp	0 kB
CommitLimit	4004780 kB
Committed_AS	691244 kB
VmallocTotal	34359738367 kB
VmallocUsed	21732 kB
VmallocChunk	34359704020 kB
HardwareCorrupted	0 kB
AnonHugePages	6144 kB
HugePages_Total	0
HugePages_Free	0
HugePages_Rsvd	0
HugePages_Surp	0
Hugepagesize	2048 kB
DirectMap4k	163708 kB
DirectMap2M	8224768 kB

Πίνακας 3. Προδιαγραφές Μνήμης

1.4.3 Στοιχεία δίσκων

Εκτελώντας την εντολή `df` αντλούμε στοιχεία σχετικά με τον κάθε δίσκο. Η ανάλυση είναι σε επίπεδο filesystem και παρέχονται πληροφορίες τόσο σε επίπεδο block όσο και σε ποσοστό επί της συνολικής χωρητικότητας του κάθε mount point. Παρατηρούμε ότι το root mount point είναι το πιο «ογκώδες» και επίσης αυτό με την μεγαλύτερη χρήση. Αυτό συμβαίνει διότι επιλέχθηκε κατά την δημιουργία του VM να χρησιμοποιηθεί ένα μόνο partition και σαν συνέπεια όλα τα δεδομένα τόσο των εφαρμογών όσο και των χρηστών αποθηκεύονται στο ίδιο σημείο.

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
devtmpfs	3995088	0	3995088	0%	/dev
tmpfs	4004780	0	4004780	0%	/dev/shm
tmpfs	4004780	403504	3601276	11%	/run
tmpfs	4004780	0	4004780	0%	/sys/fs/cgroup
/dev/vda1	20509264	4687896	14756664	25%	/
tmpfs	800960	0	800960	0%	/run/user/0

Πίνακας 4. Προδιαγραφές Δίσκων

1.5 Προετοιμασία μηχανών

Για την πραγματοποίηση των δοκιμών έγιναν μια σειρά από εργασίες ώστε να μπορούν να υποστηρίξουν τα εργαλεία μετρήσεων οι εικονικές μηχανές

1.5.1 Εγκατάσταση Apache

Για την εγκατάσταση του Apache χρησιμοποιήθηκε το standard repository του CentOS και ο package manager “yum”. Το πακέτο που επιλέχθηκε ήταν το httpd, το οποίο εγκαθίσταται ως service στο λειτουργικό και εκκινείται από τον service manager με την εντολή “systemctl”.

Με την εντολή systemctl status httpd Μπορούμε να πάρουμε πληροφορίες σχετικά με την κατάσταση του service:

```
[root@snf-872251 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2019-12-14 20:50:44 EET; 12min ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 15807 (httpd)
    Status: "Total requests: 31; Current requests/sec: 0; Current traffic:  0 B/sec"
    CGroup: /system.slice/httpd.service
            └─15807 /usr/sbin/httpd -DFOREGROUND
              └─15809 /usr/sbin/httpd -DFOREGROUND
                └─15810 /usr/sbin/httpd -DFOREGROUND
                  └─15812 /usr/sbin/httpd -DFOREGROUND
                    └─15813 /usr/sbin/httpd -DFOREGROUND

Dec 14 20:50:44 snf-872251 systemd[1]: Starting The Apache HTTP Server...
Dec 14 20:50:44 snf-872251 systemd[1]: Started The Apache HTTP Server.
```

Έξοδος Κονσόλας 1. Εγκατάσταση και παραμετροποίηση Apache

Παρατηρούμε ότι σε idle state οι μετρήσεις είναι μηδενικές, πέρα από τα total requests που έχουν προκύψει από δοκιμές, κάτι αναμενόμενο εφόσον η μηχανή έχει στηθεί αποκλειστικά για testing.

1.5.2 Ρύθμιση firewall

Επιλέχθηκε το benchmarking να μην γίνει τοπικά στην εικονική μηχανή ώστε να μην επηρεάζονται τα αποτελέσματα των μετρήσεων από το overhead της εκτέλεσης του ίδιου του benchmark. Για τον σκοπό

αυτό έπρεπε να ρυθμιστεί το firewall του λειτουργικού ώστε ο Web Server να είναι σε θέση να απαντήσει στο εργαλείο δοκιμών.

Η ελάχιστη ρύθμιση που χρειάζεται να γίνει στο firewall ώστε να γίνει published ο server είναι να επιλεγεί σαν ζώνη η public (default) και να προστεθεί σαν service το http. Αυτό έχει σαν αποτέλεσμα να επιτρέπονται συνδέσεις που αφορούν tcp πακέτα στην πόρτα 80 της εικονικής μηχανής που είναι η default πόρτα στην οποία ακούει ο Apache για εισερχόμενα requests.

Ακολουθούν οι εντολές που χρειάστηκαν και η επιβεβαίωση για το αποτέλεσμα τους:

```
[root@snf-872251 ~]# firewall-cmd --add-service http --permanent
success
[root@snf-872251 ~]# firewall-cmd --reload
success
[root@snf-872251 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0 eth1
  sources:
  services: dhcpv6-client http ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Έξοδος Κονσόλας 2. Ρύθμιση Firewall

1.5.3 Βοηθητικά Εργαλεία

Για την σύνδεση με τις εικονικές μηχανές για την πραγματοποίηση των εργασιών χρησιμοποιήθηκε το πρωτόκολλο SSH (Secure Shell) για την εκτέλεση εντολών και SFTP για την μεταφορά αρχείων. Τέλος για ευκολότερη διαχείριση και αυτοματοποίηση των συνδέσεων χρησιμοποιήθηκε ως τοπικό desktop software το SSH Connection Manager, Termius, το οποίο παρέχει δυνατότητες οργάνωσης των συνδέσεων σε λογικές ομάδες (groups) και αποθήκευσης των credentials πρόσβασης.

2 Δοκιμές Απόδοσης (Benchmarking)

Στο παρόν κεφάλαιο περιγράφονται οι δοκιμές απόδοσης που πραγματοποιήθηκαν και παρουσιάζονται τα αποτελέσματά τους. Ο προσανατολισμός των δοκιμών είναι η αξιολόγηση της υπηρεσίας και η μέτρηση της απόδοσης που θα μπορούσε να παρέχει σε ένα Web Application.

Επιλέχθηκε το μικτό τεστ ApacheBench και PHPBench να διεξαχθεί ταυτόχρονα ώστε να προσομοιωθεί ο φόρτος που μπορεί να παρουσιάζει ένας Web Server κάτω από πραγματικές συνθήκες λειτουργίας.

Με αυτή την επιλογή η εικονική μηχανή αναλαμβάνει να δημιουργήσει πολλά νέα threads για το κάθε ένα από τα οποία εξυπηρετεί την αποστολή ενός στατικού αρχείου, φόρτος που προσομοιάζει στην front end λειτουργία ενός Web Service. Από την άλλη με το PHPBench η εικονική μηχανή καλείται να πραγματοποιήσει πράξεις μεγάλης ποικιλίας που προσομοιώνουν την λειτουργία ενός back end. Η ταυτόχρονη χρήση τους δημιουργεί συνθήκες φόρτου που συνήθως παρουσιάζονται κατά την λειτουργία ενός Web Application.

2.1 Αποτελέσματα Μετρήσεων

Για την εκπόνηση της εργασίας χρησιμοποιήθηκαν τρία εργαλεία τα οποία ήταν προσανατολισμένα στην αξιολόγηση των εικονικών μηχανών με κατεύθυνση τα Web Applications. Το Gzip παρέχει μετρήσεις που κυρίως αφορούν τον δίσκο και τον επεξεργαστή, το ApacheBench μετράει την ταχύτητα εξυπηρέτησης μιας στατικής σελίδας με χρήση ταυτόχρονων παράλληλων συνδέσεων και το PHPBench προσομοιώνει πράξεις που θα μπορούσε να ζητηθεί να γίνουν από τον επεξεργαστή από ένα back end.

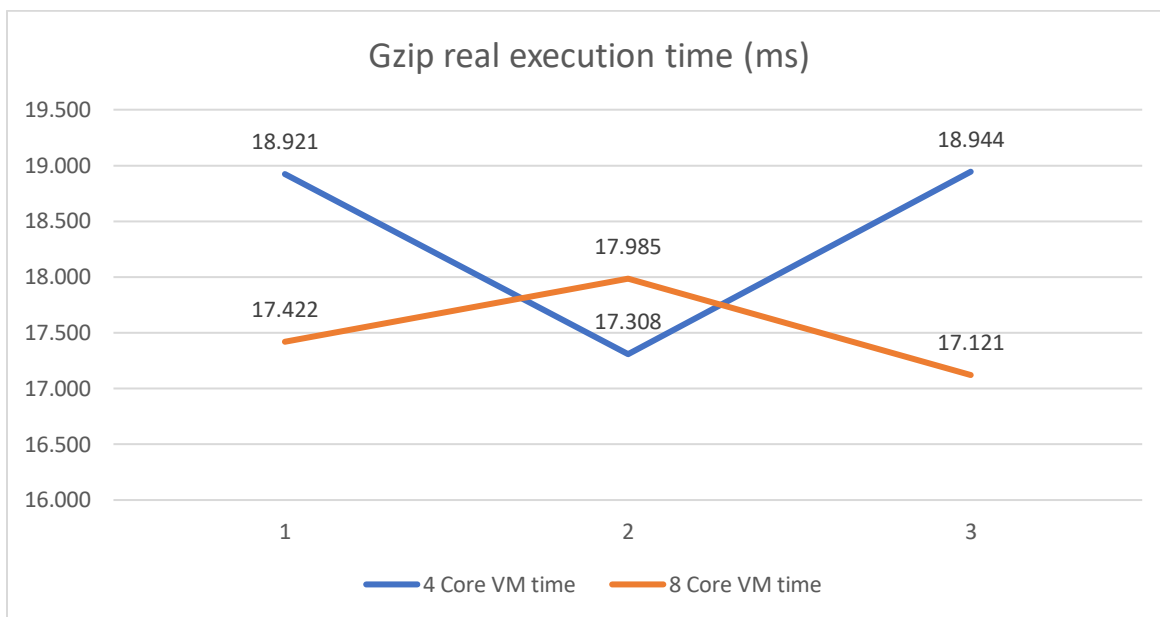
2.1.1 Gzip

Το Gzip αποτελεί ταυτόχρονα ένα file format και ένα εργαλείο συμπίεσης και αποσυμπίεσης αρχείων. Τα αρχεία που προκύπτουν από την επεξεργασία με Gzip έχουν την κατάληξη .gz και είναι συμπιεσμένα με βάση τον ομώνυμο αλγόριθμο. Το πακέτο εγκαταστάθηκε από τον Package manager του CentOS και κατά την διεξαγωγή των δοκιμών χρησιμοποιήθηκε με την εντολή -r ώστε να συμπίεσει αναδρομικά πολλά αρχεία που βρίσκονταν μέσα σε έναν φάκελο.

Το Gzip αποτελεί μέρος της σουίτας λογισμικού GNU και είναι ελεύθερα διαθέσιμο <https://www.gzip.org/>

Ο προσανατολισμός του test αυτού ήταν να μετρηθεί πέρα από τον επεξεργαστή και η απόδοση του εικονικού δίσκου αφού η λειτουργία της συμπίεσης/αποσυμπίεσης περιλαμβάνει μεγάλο κομμάτι I/O εντολών. Τα αρχεία που επιλέχθηκαν ποικίλουν σε μέγεθος με το μεγαλύτερο από αυτά να φτάνει τα 103 MB και ήταν η διανομή του Libre Office για CentOS. Το μέγιστο ποσοστό συμπίεσης που επιτεύχθηκε ήταν 48%.

Στα παρακάτω γραφήματα φαίνεται το αποτέλεσμα της εκτέλεσης του Gzip με τρεις επαναλήψεις και στα δύο μηχανήματα. Στον άξονα Χ έχουμε την κάθε επανάληψη του benchmark.



Γράφημα 1. Χρόνος εκτέλεσης Gzip

Οι διαφορές στον χρόνο εκτέλεσης είναι μικρές κάτι αναμενόμενο αφού τα δύο μηχανήματα έχουν το ίδιο configuration ακριβώς όσον αφορά το δίσκο. Βέβαια έστω και για λίγο το test είναι υπέρ της μηχανής με τους περισσότερους πόρους. Λόγω της μεγαλύτερης διαθεσιμότητας σε πόρους η γραφική παράσταση της μηχανής αυτής είναι και πιο σταθερή. Αυτό οφείλεται στο ότι επηρεάζεται λιγότερο από τον ανταγωνισμό με άλλες διεργασίες που τρέχουν στο λειτουργικό.

2.1.2 ApacheBench

Το ApacheBench είναι ένα εργαλείο για μέτρηση της απόδοσης ενός Apache Hypertext Transfer Protocol (HTTP) server. Έχει σχεδιαστεί ώστε να μπορέσει να δώσει μια εικόνα σχετικά με την απόδοση μιας τρέχουσα εγκατάστασης ενός Apache Server. Πιο συγκεκριμένα δείχνει πόσα requests το δευτερόλεπτο είναι ικανός ένας web server να εξυπηρετήσει.

Με αυτό το benchmark tool μπορούμε να μετρήσουμε την απόδοση του hardware κατά την ταυτόχρονη λήψη πολλών requests. Η λειτουργία του Apache Web Server σε αυτές τις περιπτώσεις περιλαμβάνει στο παρακίνητο την δημιουργία πολλών ξεχωριστών threads, η οποία έχει μεγάλο υπολογιστικό κόστος τόσο σε επεξεργαστική ισχύ όσο και σε μνήμη, σε μεγάλες τιμές ταυτόχρονων requests.

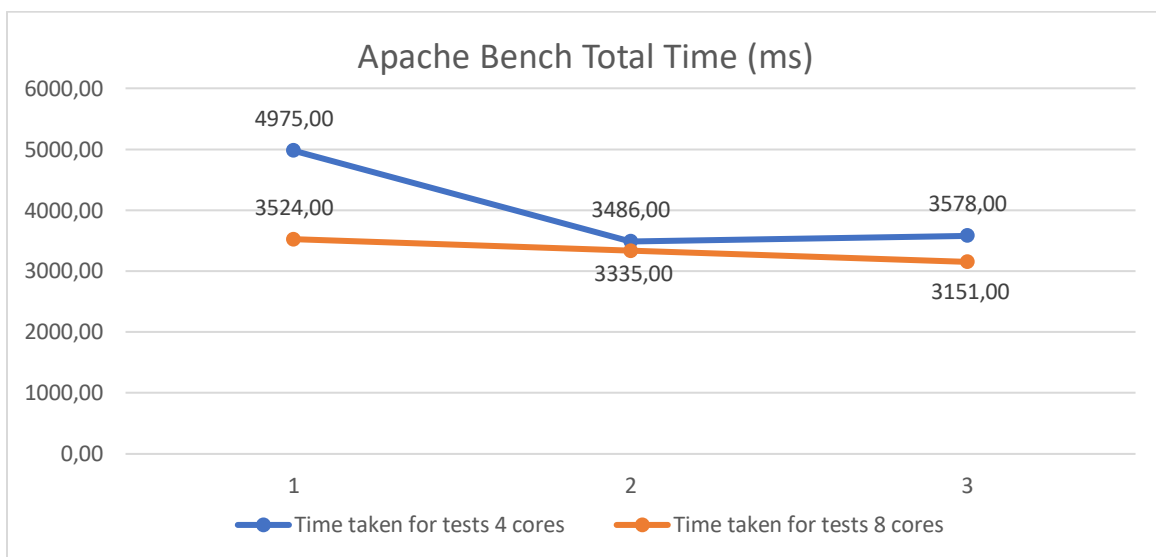
Το εργαλείο αυτό έρχεται bundled με τον Apache server και, παρότι μπορεί να ανοίξει πολλαπλά connections με τον web server που αξιολογείται, τρέχει σε ένα thread στον υπολογιστή στον οποίο εκτελείται.

Η εντολή για την εκκίνησή του είναι :

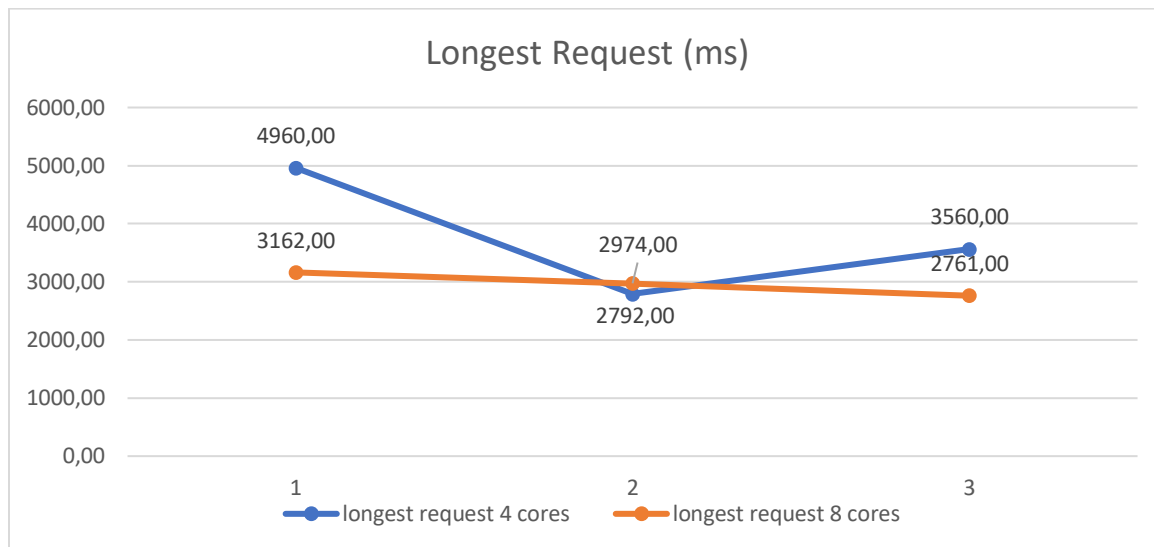
```
ab -n 1000 -c 500 "http://snf-872251.vm.oceanos.grnet.gr:80/"
```

Το οποίο θα εκτελέσει συνολικά 100 HTTP GET requests, με επεξεργασία 100 requests ταυτόχρονα, στο ορισμένο URL : «<http://snf-872251.vm.oceanos.grnet.gr:80/>», το οποίο είναι η default welcome page της βασικής εγκατάστασης του Apache index.html (5 KB) .

Στα παρακάτω γραφήματα φαίνεται το αποτέλεσμα της εκτέλεσης του ApacheBench με τρεις επαναλήψεις και ταυτόχρονη εκτέλεση του PHPBench στον Server. Οι δοκιμές πραγματοποιήθηκαν με 1000 συνολικά connections και 500 ταυτόχρονα. Στον άξονα X έχουμε την κάθε επανάληψη του benchmark.



Γράφημα 2. Συνολικός χρόνος εκτέλεσης ApacheBench



Γράφημα 3. Μέγιστος χρόνος εκτέλεσης ApacheBench

Οι διαφορές μεταξύ των δύο configuraitons είναι μικρές. Στα δύο γραφήματα παρουσιάζονται ο συνολικός χρόνος εκτέλεσης και ο μεγαλύτερος χρόνος που εξυπηρετήθηκε ένα request. Όπως φαίνεται οι δύο αυτές ποσότητες είναι συνδεδεμένες αφού η κλίση της μίας παράστασης φαίνεται να ακολουθεί την κλίση της άλλης.

Όπως παρατηρούμε από το γράφημα το εικονικό μηχανήμα με το configuration 8 πυρήνων έχει πιο σταθερή απόδοση καθώς και πιο γρήγορη απόκριση από ότι το μηχανήμα των 4 πυρήνων.

2.1.3 PHPBench

Για τον έλεγχο της απόδοσης backend λειτουργιών των εικονικών μηχανών χρησιμοποιήθηκε το software PHPBench <https://phpbench.com/>. Ως testing profile χρησιμοποιήθηκε το εξής πακέτο: <http://phoronix-test-suite.com/benchmark-files/phpbench-081-patched1.zip>

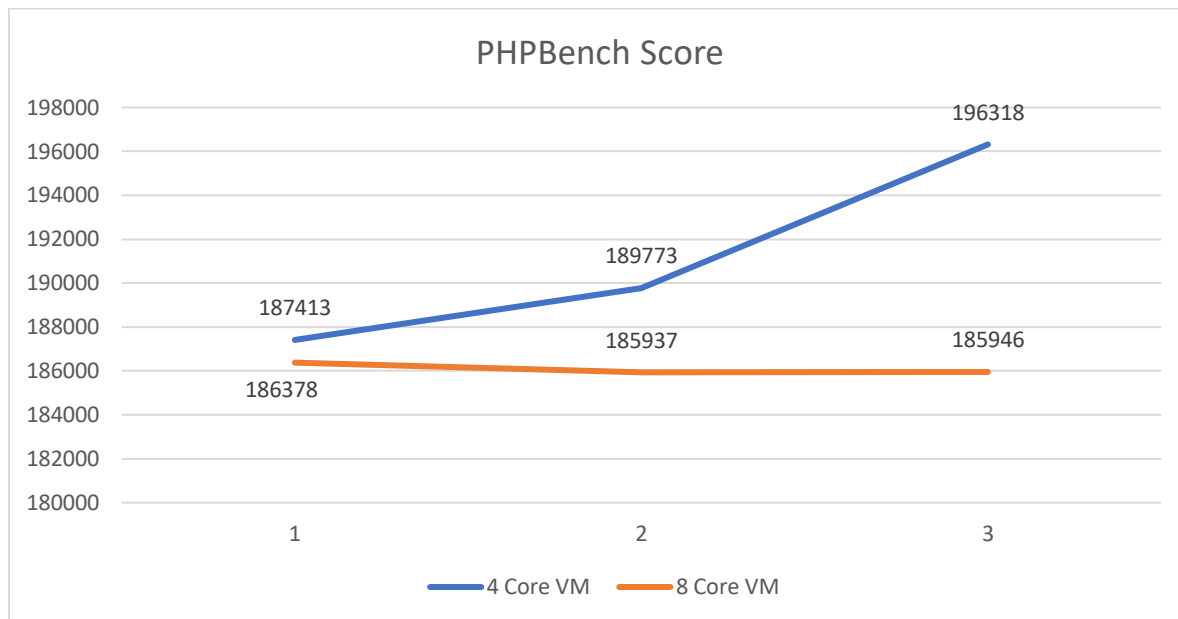
Το οποίο περιλαμβάνει στους ελέγχους του τα παρακάτω tests, και ανάλογα με τις επιδόσεις στην κάθε κατηγορία υπολογίζει το τελικό score, όπου το καλύτερο αποτέλεσμα έχει μεγαλύτερη βαθμολογία.

test_arithmetic	test_compare_false
test_global_scalar_assign	test_is_array
test_do_while_break	test_is_object
test_compare	test_casting
test_md5	test_do_while
test_constants	test_ordered_functions_references

test_variable_variables	test_ord
test_local_scalar_assign	test_foreach
test_bitwise	test_unordered_functions
test_local_hash_assign	test_chr_var
test_increment	test_while
test_compare_unstrict	test_ereg
test_crc32	test_array_operators
test_local_object_assign	test_microtime
test_local_boolean_assign	test_ordered_functions
test_local_float_assign	test_chr_hardcoded
test_local_integer_assign	test_chr_fixed
test_get_class	test_empty_loop
test_rand	test_switch
test_mt_rand	test_local_array_assign
test_if_constant	test_global_string_assign
test_references	test_local_string_assign
test_is_type	test_time
test_empty	test_preg_match
test_compare_strict	test_comment_loop
test_line	test_string_append
test_strlen	test_sha1
test_isset	
test_compare_invert	

Πίνακας 5. Αναλυτικά στοιχεία ελέγχων PHPBench

Το εργαλείο μετρήσεων ρυθμίστηκε ώστε να εκτελεί 150.000 επαναλήψεις για κάθε υπο-τέστ. Τα αποτελέσματα του test φαίνονται στον παρακάτω πίνακα:



Γράφημα 4. Συνολικό Αποτέλεσμα PHPBench

Εδώ παρατηρούμε κάτι που μοιάζει παράδοξο, ενώ θα αναμέναμε το μηχάνημα με τους 8 πυρήνες να έχει καλύτερη απόδοση παρατηρούμε ότι συνέβη το ακριβώς αντίθετο. Από ότι παρατηρήθηκε με την χρήση της εντολής htop κατά την διάρκεια του test δεν χρησιμοποιούνταν πολλαπλοί πυρήνες παρά μόνο ένας, συνεπώς η μηχανή με τους περισσότερους πυρήνες δεν είχε κάποιο προβάδισμα και αυτό αποτυπώθηκε στην δοκιμή.

1	[0.0%	Tasks: 49, 17 thr; 2 running
2	[0.0%	Load average: 0.24 0.11 0.07
3	[100.0%	Uptime: 05:09:36
4	[0.7%	
Mem	[223M/3.70G	
Swp	[0K/0K	

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
9548	root	20	0	322M	13820	7476	R	99.8	0.4	0:02.08	php phpbench.php -i 150000
9532	root	20	0	119M	2256	1424	R	1.3	0.1	0:00.46	htop
1013	root	20	0	236M	9456	6220	S	0.0	0.2	0:11.08	/usr/sbin/rsyslogd -n
579	root	16	-4	55528	1064	612	S	0.0	0.0	0:06.16	/sbin/auditd
1354	root	20	0	112M	2088	1612	S	0.0	0.1	0:00.31	-bash

Έξοδος Κονσόλας 3. Παρακολούθηση φόρτου επεξεργαστή κατά τη διάρκεια του PHPBench

Και σε αυτό το τεστ υπάρχει μεγαλύτερη σταθερότητα στις τιμές, επίσης όπως και με τα προηγούμενα εργαλεία οι διαφορές είναι μικρές.

Περισσότερα και αναλυτικότερα στοιχεία βρίσκονται στα αρχεία μορφότυπου .xls που συνοδεύουν την παρούσα εργασία.

2.2 Υποστηρικτικός Κώδικας

Για τις επαναλήψεις των μετρήσεων, αναπτύχθηκαν bash scripts ώστε να αυτοματοποιήσουν μέρος της εργασίας που χρειάζεται να γίνει για την εξαγωγή των αποτελεσμάτων. Συνολικά δημιουργήθηκαν 3 scripts τα οποία μπορούν να εκτελεστούν με παράμετρο τον αριθμό των επαναλήψεων του κάθε τεστ και αν δεν οριστεί στην είσοδο κάποια τιμή έχουν ως default τις τρεις επαναλήψεις.

Ο κώδικάς τους παρατίθεται παρακάτω:

2.2.1 Gzip script

```
max=${VARIABLE:=3} # If iterations not set or null, set it to 3.
for (( i=0; i < $max; ++i ))
do
  cp LibreOfficeBackup/ LibreOffice_6.3.4.2_Linux_x86-64_rpm -R
  time gzip -r LibreOffice_6.3.4.2_Linux_x86-64_rpm/ -v
  rm -rf LibreOffice_6.3.4.2_Linux_x86-64_rpm
done
```

Έξοδος Κονσόλας 4. Gzip script

2.2.2 ApacheBench script

```
max=${VARIABLE:=3} # If iterations not set or null, set it to 3.
for (( i=0; i < $max; ++i ))
do
  ab -n 1000 -c 500 "http://snf-872251.vm.okeanos.grnet.gr:80/index.html" >>
  results.txt
done
```

Έξοδος Κονσόλας 5. ApacheBench script

2.2.3 PHPBench script

```
max=${VARIABLE:=3} # If iterations not set or null, set it to 3.
for (( i=0; i < $max; ++i ))
do
  php phpbench.php -i 150000 >> results.txt
done
```

Έξοδος Κονσόλας 6. PHPBench script

2.3 Προβλήματα Μετρήσεων

Κατά την εκτέλεση του ApacheBench, παρατηρήθηκαν διακυμάνσεις που αφορούσαν σε προβλήματα της σύνδεσης διαδικτύου. Αυτό οφείλεται εν μέρει από την ποιότητα της διαδικτυακής σύνδεσης του παρόχου και του όγκου και της αποστολής πολλών παράλληλων requests.

Ακόμα και με την πρόνοια της επανάληψης των μετρήσεων 3 φορές, ήταν συχνό φαινόμενο να αποτυγχάνουν requests προς την εικονική μηχανή. Ενδεικτικά παρατίθεται μια μέτρηση που εμφανίζει 54 failed requests από τα 1000 συνολικά:

Server Software:	Apache/2.4.6
Server Hostname:	snf-872251.vm.oceanos.grnet.gr
Server Port:	80
Document Path:	/index.html
Document Length:	4897 bytes
Concurrency Level:	500
Time taken for tests:	4.975 seconds
Complete requests:	1000
Failed requests:	54
(Connect: 0, Receive: 0, Length: 54, Exceptions: 0)	
Total transferred:	4892712 bytes
HTML transferred:	4632562 bytes
Requests per second:	201.01 [#/sec] (mean)
Time per request:	2487.388 [ms] (mean)
Time per request:	4.975 [ms] (mean, across all concurrent requests)
Transfer rate:	960.45 [Kbytes/sec] received
Connection Times (ms)	
	min mean[+/-sd] median max
Connect:	16 103 189.6 51 1042
Processing:	249 1318 1340.4 590 4899
Waiting:	0 1110 1224.5 515 4899
Total:	288 1420 1362.6 689 4960

Πίνακας 6. Ενδεικτική εσφαλμένη λειτουργία του ApacheBench λόγω προβλημάτων δικτύου

2.4 Machine Learning on Cloud Platforms

Πέρα από το πλαίσιο της παρούσας εργασίας, πραγματοποιήθηκαν δοκιμές μέτρησης της απόδοσης της πλατφόρμας ~Okeanos ενός consumer grade υπολογιστή και της cloud υπηρεσίας collab της Google ως προς την απόδοση κατά την εκπαίδευση ενός Machine Learning μοντέλου. Στόχος ήταν η αποτύπωση των δυνατοτήτων των τριών διαφορετικών συστημάτων και η σύγκρισή τους.

Για την εξαγωγή των αποτελεσμάτων χρησιμοποιήθηκε κώδικας που αναπτύχθηκε για το μάθημα Audio Signal Processing του εαρινού εξαμήνου του προγράμματος. Χρησιμοποιήθηκε dataset των 2000 αρχείων

ήχου για λόγους επιτάχυνσης των δοκιμών. Η μέτρηση αφορά μόνο το κομμάτι της εκπαίδευσης του μοντέλου και όχι της προ-επεξεργασίας των αρχείων ήχου.

2.4.1 Hardware Specifications

Η σύνθεση των διαθέσιμων (εικονικών και μη) πόρων ήταν η εξής:

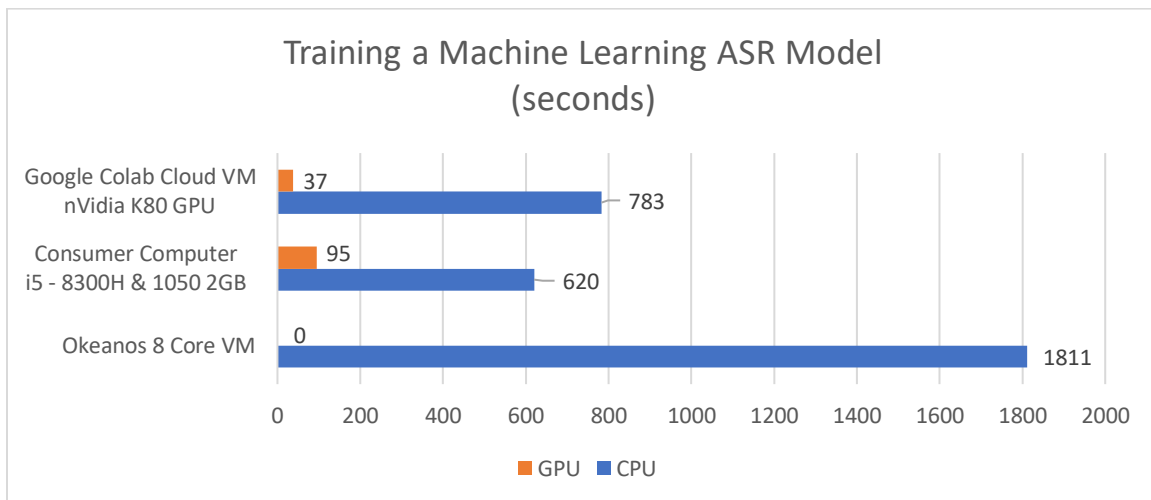
~Okeanos 8 core VM:	
CPU	8x QEMU Virtual CPU version 2.1.2
RAM	8 GB DDR4
GPU	Not available

Consumer grade computer:	
CPU	Intel Core i5 8300H – 4 physical cores
RAM	12 GB DDR4
GPU	nVidia 1050 2GB

Google Collab Cloud:	
CPU	1x Intel(R) Xeon(R) CPU @ 2.20GHz
RAM	12.6 GB
GPU	1x nVidia Tesla K80

Πίνακας 7. Σύνθεση μηχανών δοκιμής *Machine Learning Training*

Στο παρακάτω διάγραμμα παρατηρούμε ότι η διαφορά στον χρόνο εκτέλεσης διαφέρει σημαντικά όταν ο κώδικας εκτελείται στην κάρτα γραφικών χρίζοντας ασύμφορη την επένδυση σε cloud υπηρεσίες που δεν παρέχουν στις συνθέσεις των εικονικών μηχανών τους virtual GPUs. Ένα άλλο ενδιαφέρον στοιχείο που προκύπτει από το διάγραμμα είναι η μικρή διαφορά στο speedup (3x) μεταξύ της υπηρεσίας Collab και ενός consumer υπολογιστή. Λόγω του ότι το κόστος μιας Tesla k80 είναι το δεκαπλάσιο μιας GTX 1050 το speedup που παρέχει κρίνεται ότι δεν έχει τα αναμενόμενα αποτελέσματα.



Γράφημα 5. Χρόνος εκπαίδευσης Machine Learning μοντέλου

Σε αυτό το σημείο αξίζει να σημειωθεί ότι με την χρήση μιας k80 και την περεταίρω βελτιστοποίηση των μεθόδων εξαγωγής του μοντέλου θα μπορούσε να αυξηθεί επιπλέον το speedup, μέσω μείωσης της ακρίβειας των floating point μεταβλητών σε 16bit. Η Tesla k80 αυξάνει την απόδοσή της σε σχέση με τις κοινού τύπου κάρτες σε αυτές τις περιπτώσεις λόγω της αρχιτεκτονικής της.

3 Συμπεράσματα

Από την ανάλυση των μετρήσεων προκύπτει ότι η υπηρεσία ~Okeanos παρέχει εικονικές μηχανές με CPUs χαμηλών σε απόδοση single-thread με μέτρια ταχύτητα στο storage και με περιορισμένες δυνατότητες στην παράλληλη επεξεργασία πολλών ταυτόχρονων requests. Η απόκριση σε 500 ταυτόχρονα requests ήταν κατά μέσο όρο 5 sec – χρόνος απαγορευτικός για μια σύγχρονη Web υπηρεσία.

Επίσης παρατηρήθηκε ότι η παροχή περισσότερων πυρήνων δεν βελτιώνει τα αποτελέσματα των μετρήσεων σε συνθήκες προσομοίωσης Web Application. Οπότε ο υποψήφιος χρήστης θα πρέπει να λάβει υπόψη του ότι το παραπάνω κόστος μιας τέτοιας επιλογής δεν αποφέρει ανάλογο κέρδος (μάλιστα είναι σχεδόν μηδενικό) και να προσαρμόσει τις απαιτήσεις του σε αιτούμενο virtual hardware.

Όσο αφορά στις δοκιμές σχετικά με το machine learning αν και εκ πρώτης όψης τα αποτελέσματα συνηγορούν στο ότι το virtual cpu hardware είναι ανεπαρκές, θα πρέπει να λάβει κανείς υπόψιν τα κόστη του ρεύματος και της φθοράς και αστοχίας υλικού πέρα από την χρονική απόδοση κατά το training ενός μοντέλου. Αν ο χρόνος εκτέλεσης ενός τέτοιου training δεν είναι ζητούμενο υψηλής προτεραιότητας και δεδομένου ότι αυτές οι διαδικασίες μπορεί να διαρκέσουν ούτως ή άλλως μέρες υπάρχει η πιθανότητα να συμφέρει οικονομικά μια τέτοια λύση.

Δεδομένων των ανωτέρω η προτεινόμενη χρήση θα μπορούσε να είναι για την εγκατάσταση και λειτουργία ενός enterprise business software για εσωτερική χρήση μιας μεσαίου ή και μεγάλου μεγέθους εταιρείας. Το Software εσωτερικής χρήσης συνήθως σημειώνει μικρότερο αριθμό από ταυτόχρονες συνδέσεις και παράλληλα απαιτεί λιγότερους επεξεργαστικούς πόρους. Εναλλακτικά θα μπορούσε να χρησιμοποιηθεί και από μικρού μεγέθους επιχειρήσεις ως λύση για την λειτουργία των ιστοσελίδων τους.

Η πλατφόρμα ~Okeanos παρέχει ένα μεγάλο σετ λειτουργιών σχετικών με τις εικονικές μηχανές όπως :

- Scaling
- Power Management
- Interactive Console
- Create
- Delete

Απουσιάζει όμως η επιλογή για template import/export, μια πολύ ουσιώδης λειτουργία για τους επαγγελματίες. Με τα templates μπορεί να μειωθεί πολύ ο χρόνος μιας εγκατάστασης. Επιπλέον δεν υπάρχει η δυνατότητα τήρησης Snapshots ώστε να αποθηκευτεί το image της μηχανής σε κάποιο χρονικό σημείο. Αν και είναι δυνατό πλέον το provisioning να αυτοματοποιηθεί με εργαλεία DevOps οι λειτουργίες αυτές θεωρώ ότι δεν θα έπρεπε να απουσιάζουν ώστε να μπορεί να χρησιμοποιηθεί με αξιοπιστία.