



CLOUD COMPUTING & VIRTUALIZATION:

HOMEWORK 1



ASSIGNED BY:

TAHEL BUTBUL - 315259168
YONATAN BANSAY - 207599499

שאלה 1:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Celsius	17	18	21	26	30	32	33	34	32	29	24	20
Fahrenheit	62.6	64.4	69.8	78.8	86	89.6	91.4	93.2	89.6	84.2	75.2	68
Kelvin	290.15	291.15	294.15	299.15	303.15	305.15	306.15	307.15	305.15	302.15	297.15	293.15
Rankine	522.27	534.07	529.47	538.47	545.67	549.27	551.07	552.87	549.27	543.87	534.87	527.67

	AVG	STD	CV
Celsius	26.3333	6.1683	0.2343
Fahrenheit	79.4	11.1047	0.1398
Kelvin	299.4833	6.1693	0.0206
Rankine	539.07	11.1047	0.0206

1. נשים לב שמקדם הווריאציה שקיבלנו בעבור מעלות צלסיוס שונה מיזה שקיבלנו בעבור מעלות פרנהייט.
הסיבה לכך היא הקשר בניהם שהוא: $^{\circ}\text{F} = ^{\circ}\text{C} \cdot \frac{9}{5} + 32$
הכפלה בקבוע משנה את הממוצע ואת סטיית התקן באותו אופן, אך הזזה משפיעה רק על הממוצע.
מכיוון שמקדם הווריאציה מחושב ע"י הנוסחה: $CV = \frac{\sigma}{|\mu|}$
נקבל: $CV_{^{\circ}\text{C}} = \frac{\sigma_{^{\circ}\text{C}}}{|\mu_{^{\circ}\text{C}}|} \neq CV_{^{\circ}\text{F}} = \frac{\sigma_{^{\circ}\text{C}}}{|\mu_{^{\circ}\text{C}} \cdot 1.8 + 32|}$ כלומר ניתן לראות שבעקבות ההזזה מקדמי הווריאציה שהתקבלו שונים.

2. מקדמי הווריאציה שקיבלנו בעבור מעלות קלווין ובעבור מעלות רנקין זההים.
הסיבה לכך היא שהקשר בניהם הוא: $^{\circ}\text{R} = ^{\circ}\text{K} \cdot \frac{9}{5}$
מאחר שמדובר בהכפלה בקבוע בלבד הממוצע סטיית התקן ישתנו באותו אופן ולכן המקדמים שיתקבלו יהיו זהים.
נקבל: $CV_{^{\circ}\text{K}} = \frac{\sigma_{^{\circ}\text{K}} \cdot 1.8}{|\mu_{^{\circ}\text{K}} \cdot 1.8|} = CV_{^{\circ}\text{R}} = \frac{\sigma_{^{\circ}\text{R}}}{|\mu_{^{\circ}\text{R}}|}$

3. התשובות שקיבלנו בשני בסעיפים שונות.
ההבדל בניהן נובע מכך שבסעיף 1 פונקציית ההמרה כוללת הכפלה בקבוע שמשפיעה על הממוצע ועל סטיית התקן יחד ובשונה מיזה ההזזה משפיעה רק על הממוצע ולכן היחס שהוא צקדם הווריאציה משתנה.
בשונה מסעיף 2 שבו פונקציית ההמרה כוללת רק הכפלה בקבוע שמשפיע על סטיית התקן והממוצע באותו האופן ולכן היחס ביניהם שהוא מקדם הווריאציה לא משתנה למרות ההמרה.

שאלה 2:

SR-IOV (*Signal Root Input/Output Virtualiation*) הינה טכנולוגיה שמאפשרת לרכיב המחובר ב**PCI Express**, להתחלק למספר רכיבים וירטואליים שיכולים להיות מוקצים למכונות וירטואליות שונות ואפשר להם בעצם גישה נפרדת לפונקציות הפיזיות של הרכיב המקורי.

לדוגמה בעבור כרטיס רשת כאשר מבצעים שימוש בוירטואליזציה *hypervisor* מנהל ואחראי על תעבורת הרשת בין ה**VMs** השונים לבין כרטיס הרשת הפיזי. לעומת זאת כאשר מתבצע שימוש ב**SR-IOV** לכל **VM** מוקצה כרטי וירטואלי המאפשר לו גישה ישירה ומהירה יותר אל הכרטיס הפיזי.

לכל כרטיב יש **PF** שהיא הפונקציונליות שלו ומהווה חלק ממחיצת האב של *hypervisor*. על מנת לאפשר גישה של הכרטיסים הוירטואלים לכרטיס הפיזי מגדירים **VF** כל אחת מהן מקושרת לכרטיס רשת וירטואלי ולכל אחד מה**PF** וה**VF** יש כתובת **PCI** ייחודית המאפשרת להבדיל בין זרמי המידע השונים ובכך לאפשר לכל אחד מה**VMs** לתקשר ישירות מול הכרטיס הפיזי מבלי התערבות של *hypervisor*.

שימוש ב**SR-IOV** בכרטיס רשת לדוגמא מאפשר שיפור ביצועים והקטנת ההשהייה, מאחר ויש קשר ישיר בין הכרטיסים הוירטואלים לכרטיס הפיזי המאפשר ביצועי רשת שקרובים מאוד לביצועים ללא וירטואליזציה

EPT (*Extended Page Tables*) הינה טכנולוגיית וירטואליזציה נתמכת חומרה המאפשרת ל**VM** גישה ישירה למיפוי של הזיכרון של המארח ובאמצעות זה היא משיגה זמן ריצה יעיל יותר והפרדה טובה יותר.

בסביבות וירטואליות רבות לכל **VM** יש מרחב כתובות וירטואלי משלו שממופה לכתובות פיזיות של המארח על ידי *hypervisor*, החיסרון שעולה משיטה זו הינו שהמיפוי יכול להיות מסובך ובכך לבזבז זמן רב במקרה שבו ה**VM** מריץ הרבה אפליקציות או מערכות הפעלה במקביל.

כאשר משתמשים ב**EPT** ה**hypervisor** מגדיר שכבה נוספת של *page table* שמיועדים ספציפית ל**VM** שזה מאפשר לו למפות ישירות בין הכתובות הוירטואליות לכתובות הפיזיות של המארח, ללא צורך בחיפוש טבלאות אצל ה**hypervisor** או התערבותו וזו הדרך שבה מתבצעות גישות יעילות יותר לזיכרון אשר משפרות את הביצועים של המערכת.

בנוסף שימוש ב**EPT** משפר את רמת האבטחה של הערכת מאחר וישנה הפרדה חומרתית מלאה בין הזיכרון של ה**VMs** השונים, כלומר לכל **VM** יש את *page table* משלו לכן נמנעת גישה של **VM** אחד לזיכרון של אחר ובצורה זו לא מתאפשרות הפצות או זליגות בין ה**VMs** השונים.

שאלה 3:

Comparison of Containerization and Virtualization in Cloud Architectures [Comparison of Containerization and Virtualization in Cloud Architectures | IEEE Conference Publication | IEEE Xplore](#)

המאמר שבחרנו מבצע השוואה בין שימוש בוירטואליזציה לבין שימוש בcontainers ב-3 ארכיטקטורות ענן:

IAAS (*infrastructure as a service*) הלקוח אחראי על כל החלק התכנותי ממערכת ההפעלה ועד האפליקציה ולא אחראי על החומרה.

PAAS (*platform as a service*) הלקוח אחראי על המידע והאפליקציה וכל השאר באחריות ספק השירות.

SAAS (*software as a service*) ספק השירות אחראי על הכל והלקוח משתמש באפליקציה רק בעבור שירות.

במאמר בוצעו הערכת ביצועים, יישום, והשוואה בין 3 הארכיטקטורות הנ"ל בשיטה של וירטואליזציה ובשיטה של containers.

מימוש IAAS:

קונטיינר: הרצת קונטיינר מבוסס *ubuntu image* באמצעות דוקר.
וירטואליזציה: שימוש בוירטואל בוקס ליצירת *ubuntu VM*
הערכה: בוצעה באמצעות הכלי *sysbench* אשר מודד ביצועי מעבד וזיכרון

מימוש PAAS :

קונטיינר: הרצת 2 קונטיינרים האחד מבוסס *ubuntu image* והשני מבוסס *MySQL image*.
וירטואליזציה: שימוש בוירטואל בוקס ליצירת *ubuntu VM* והתקנה *MySQL server*.
הערכה: בוצעה באמצעות הכלי *sysbench* אשר מודד ביצועי מעבד וזיכרון ובנוסף מדידת קריאה וכתיבה לשרת *SQL*

מימוש SAAS:

קונטיינר: יצירת קונטיינר של *MySQL* ושל *Django* והקמת אתר אינטרנט בסיסי.
וירטואליזציה: יצירת 2 מכונות וירטואליות אחת בעבור *MySQL image* ואחת בעבור *Django* תקשורת בין המכונות אפשרית לטובת פונקציונליות בסיס נתונים תקינה והקמת אתר אינטרנט.
הערכה: מדידות עומס באמצעות *Locust* על אפליקציית האתר.

תוצאות:

IAAS מדידות הביצועים הניבו תוצאות כמעט זהות לשימוש בוירטואליזציה לעומת הקונטיינר. אין יתרון מהותי לאחת השיטות מבחינת ביצועי המעבד.

PAAS יסתמן יתרון ברור לשימוש בקונטיינר לעומת וירטואליזציה. השיהוי של רוב המשימות קטן יותר בעשרות אחוזים עבור השימוש בקונטיינר ובנוסף קצב המטלות והספקם גבוה יותר.

SAAS ניכר יתרון ברור לשימוש בקונטיינר. הם מסוגלים להתמודד עם משפר פעולות גדול יותר ולכן זמן התגובה הממוצע קטן יותר לכל סוגי הפעולות שבוצעו באתר נבדקו : *update, rerieve, delete, create*

מסקנות:

מתוצאות המאמר עולה כי לקונטיינר יש עדיפות על וירטואליזציה עבור שימוש בארכיטקטורות *SAAS, PAAS* בהן משיגים ביצועים טובים יותר ולעומת זאת בארכיטקטורה *IAAS* לא ניכר יתרון לאף אחת מהשיטות מאחר שהיא מהווה בסיס לפיתוחים עתידיים. לפי כותבי המאמר ישנה חשיבות רבה יותר לעמידות מאשר ניידות ולכן עדיף יהיה לבחור בוירטואליזציה במקרים אלו ונכתב כי בעת כתיבת המאמר זה היה טרנד בשוק.