



**דו"ח סיכום לקורס מידול עומסים במערכות מחשב –
203.3713**

מוגש ל: אורנה אגמון בן-יהודה

מגישים:

- אליאס מרגילה ת.ז 318386646
- גוליאן שעאר ת.ז 316050608
- פהד נאסר ת.ז 318489168
- יובל תמיר ת.ז 201277894

תוכן ענייניים

<u>2</u>	שלב 1 – בחירת TRACE
<u>3</u>	שלב 2
3	ניתוח ראשוני של המידע
3	NASA-Log File Graphs
8	MATLAB-Log File Graphs
11	מענה על שאלות
<u>15</u>	שלב 3 – התאמות בין התפלגיות לבין תוצאות ניתוח המידע
<u>16</u>	שלב 4 – התאמות בין גרפים לבין התפלגיות אמפיריות
16	NASA-Log File Graphs
16	CDF of Runtimes time – All users
18	CDF of Runtimes time – Other users
20	CDF of Runtimes time – Special users
22	CDF of Interarrival time
24	MATLAB-Log File Graphs
24	CDF of Runtimes
26	CDF of Interarrival time
<u>28</u>	שלב 5 – התפלגות ZIPF
<u>31</u>	שלב 6
31	NASA-Log File
33	MATLAB-Log File
<u>34</u>	שלב 7 – ייצור HYPER DISTRIBUTION
<u>36</u>	שלב 8 – ייצור מודל המדמה את המידע הראשוני
<u>37</u>	שלב 9- אימונות
<u>63</u>	ביבליוגרפיה ומקורות
<u>63</u>	COLOPHON

שלב 1 – בחרית Trace

בחרנו את ה-Trace של iPSC/860 NASA Ames. תוצאות ניתוח ה-gog מוצגות בפירוט בהמשך הדוח.

בנוסף, מתואר בדוח גם הניתוח של ה-gog היבשתי של MATLAB.

שלב 2

ניתוח ראשוני של המידע

NASA-Log File Graphs

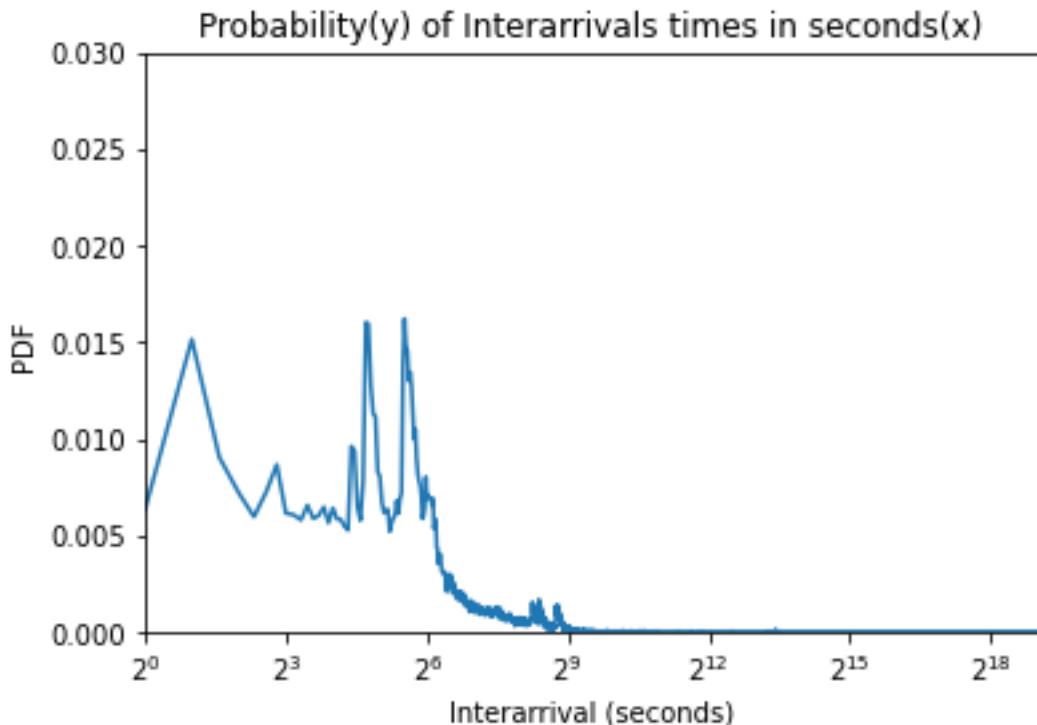


Figure 2-1

בחרנו לייצג את הנתונים בגרף 1-2 אשר בו ציר ה- x בסקאלה לוגריתמית. בחרנו ביצוג זה מכיוון שקיבלנו נתונים מודדים אשר אי אפשר לייצגם בציר עם ערכיים רגילים. בנוסף, שמננו לב למשתמש עם התנヘגות חריגה – משתמש "special", אשר לו הוא jobs ארוכים במיוחד. ואחד מהם אף נמשך במשך 6 ימים, מה שסביר את ה-Heavy tail שאפשר לראות בגרף.

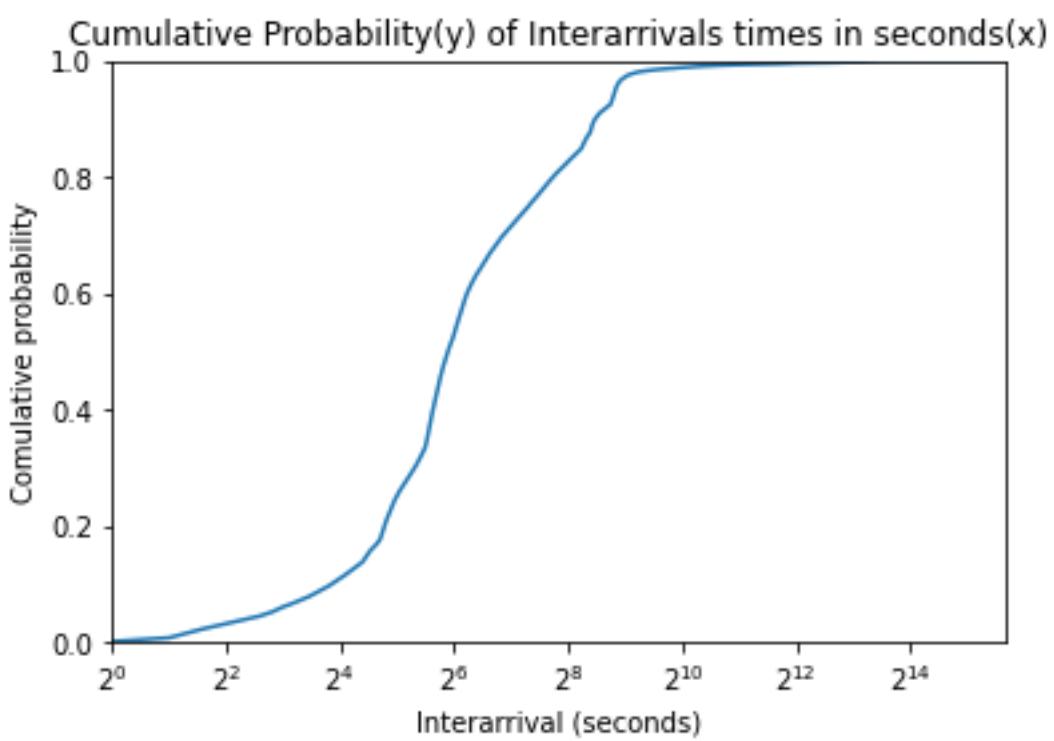


Figure 2-2

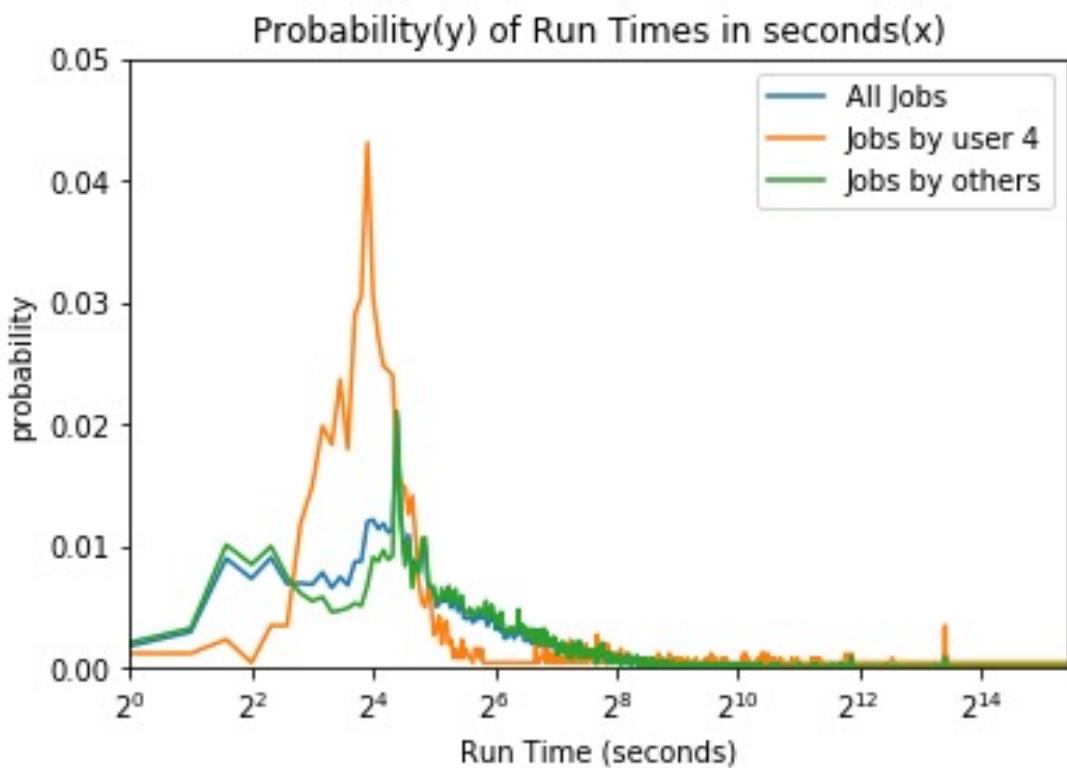


Figure 2-3

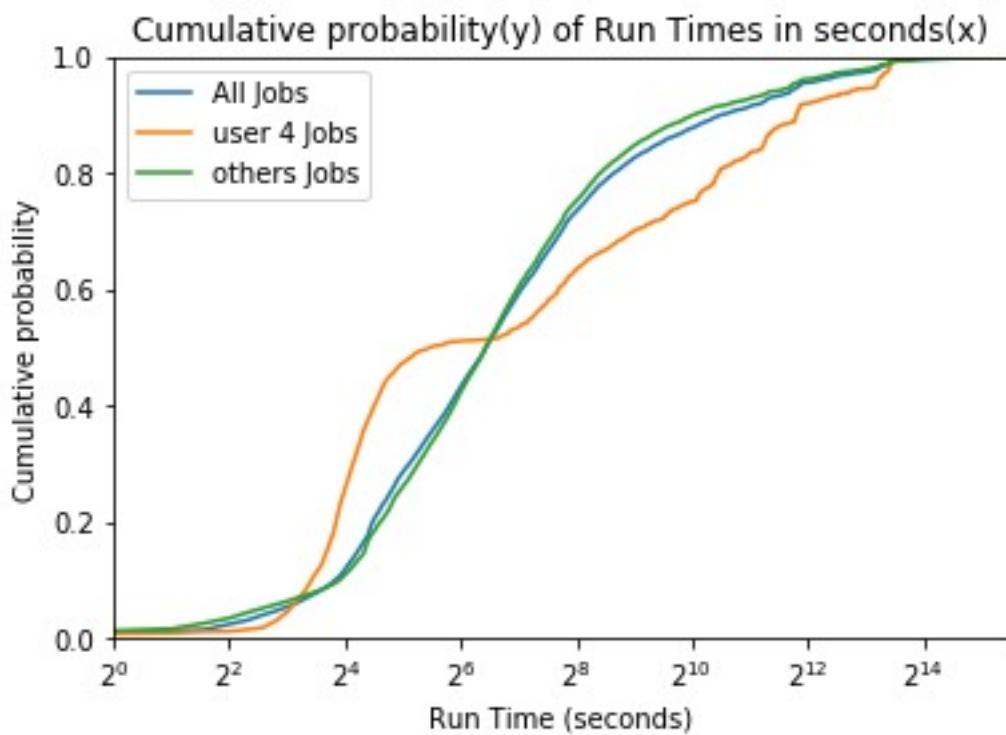


Figure 2-4

גרפים 2-2, 2-3, 2-4 מתארים התנהגות שונה של אחד ממשתמשי המערכת לעומת שאר המשתמשים, בפרק שזמני הריצה של ה-Jobs שלו, ארוכים לעומת השאר.

אפשר לראות בגרף 2-4, ש"בזבב" של הגרף שמתאר את המשתמש "Special", גם כאשר $2^{14} > X$, עדין ערך ה-Y אינו שווה ל-1, מכיוון כי יש Job באורך של שישה ימים.

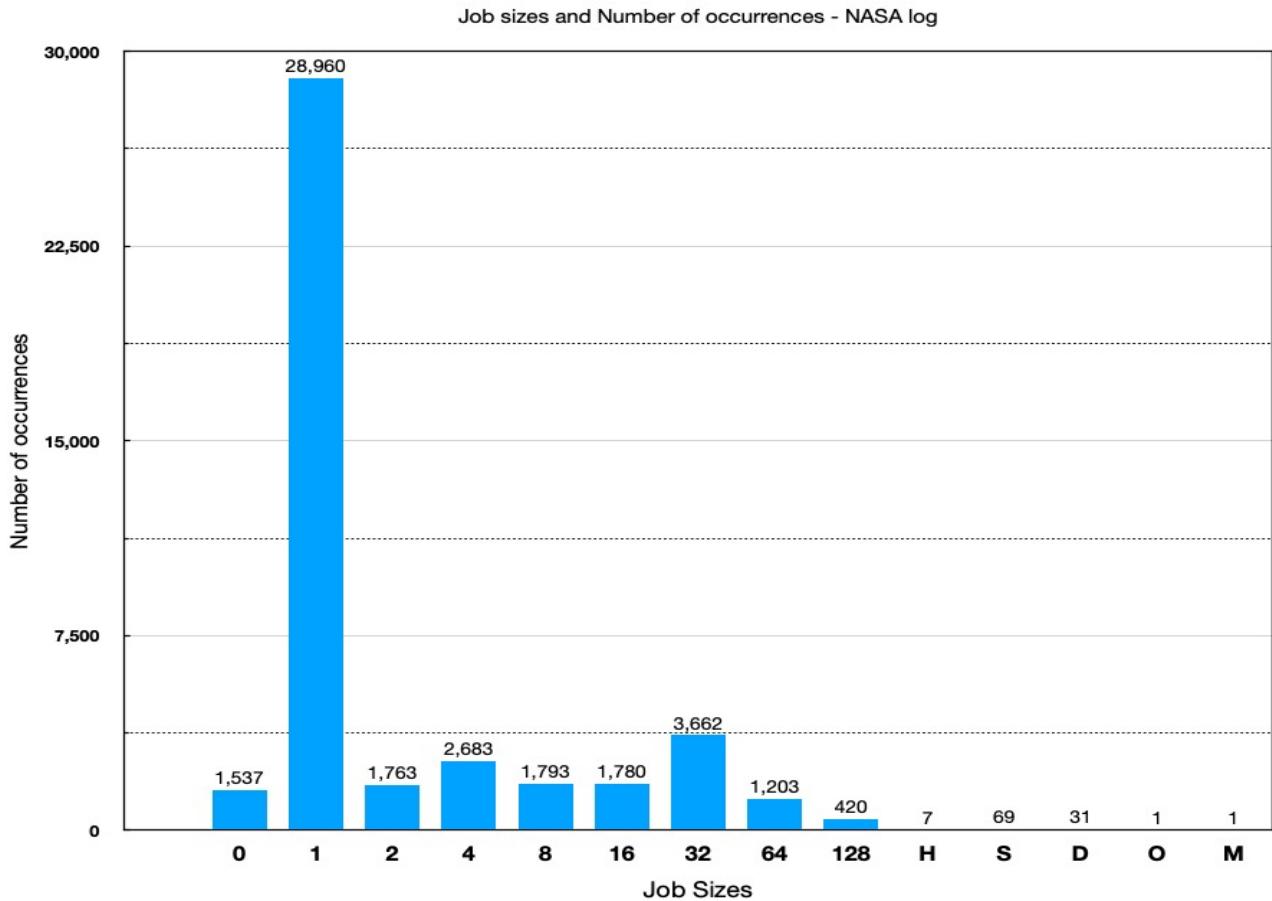


Figure 2-5

גרף 5-2 מתואר הצורה של היסטוגרמה, אשר מתרמת לנו את מספר המופיעים של גודלי *ספ* שונים במערכת. ניתן לראות שהגודל הנפוץ ביותר הוא בגודל 1, אשר נובע מהסיבה שישן המונ פועלות מערכת (sysadmin), שambilות פועלות קצרות אשר בודקות את תקיןות המערכת (למשל – pid).

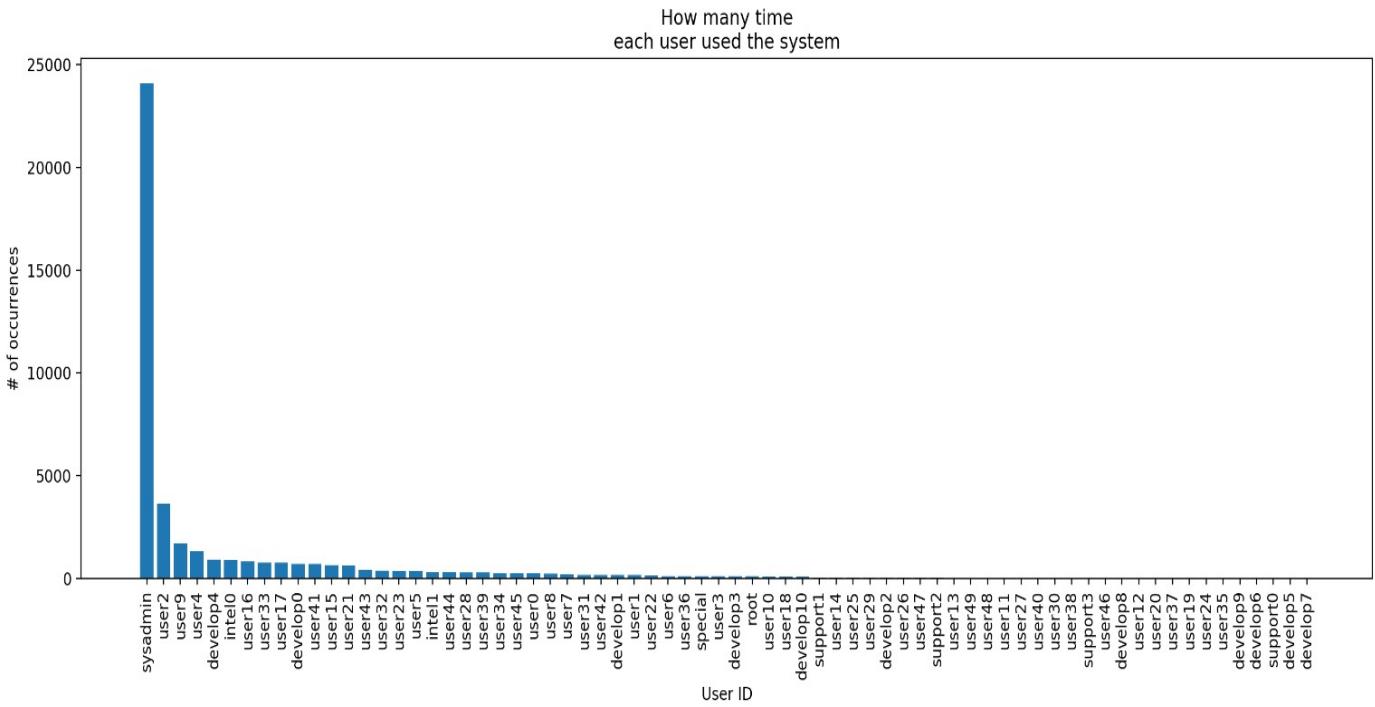


Figure 2-6

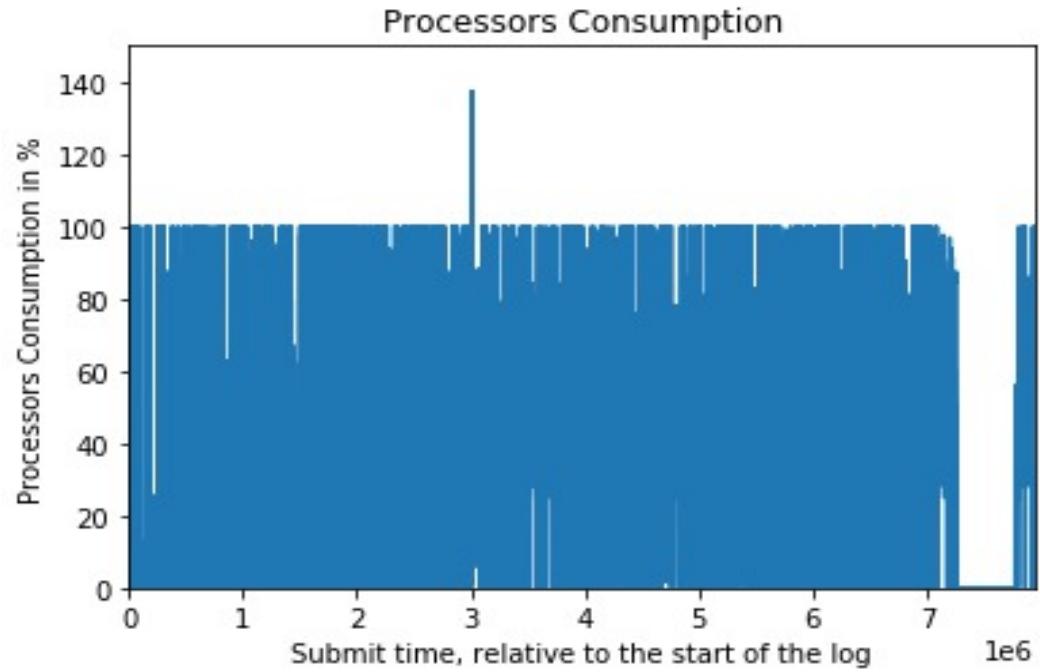


Figure 2-7

בגרף 6 ניתן לראות את במות ה-`submit` שבוצעו לפי כל משתמש במערכת, ובשים לב שההמשתמש `minsys` ביצע במותות גדולות במיוחד לעומת השאר (זהrat מבחן) שההמשתמש בנו"ל מבצע פעולות מערכתי – `pid` אשר בודק את תקינות המערכת בצורה תכופה).

גרף 7 מראה את אחוז המעבדים התativosים בכל שנייה מהתחילת תיעוד המערכת (תחילת ה-`log`) ועד סיוםו. המספרים בציר ה-*X* הם בכפоловות של 10^6 .

MATLAB-Log File Graphs

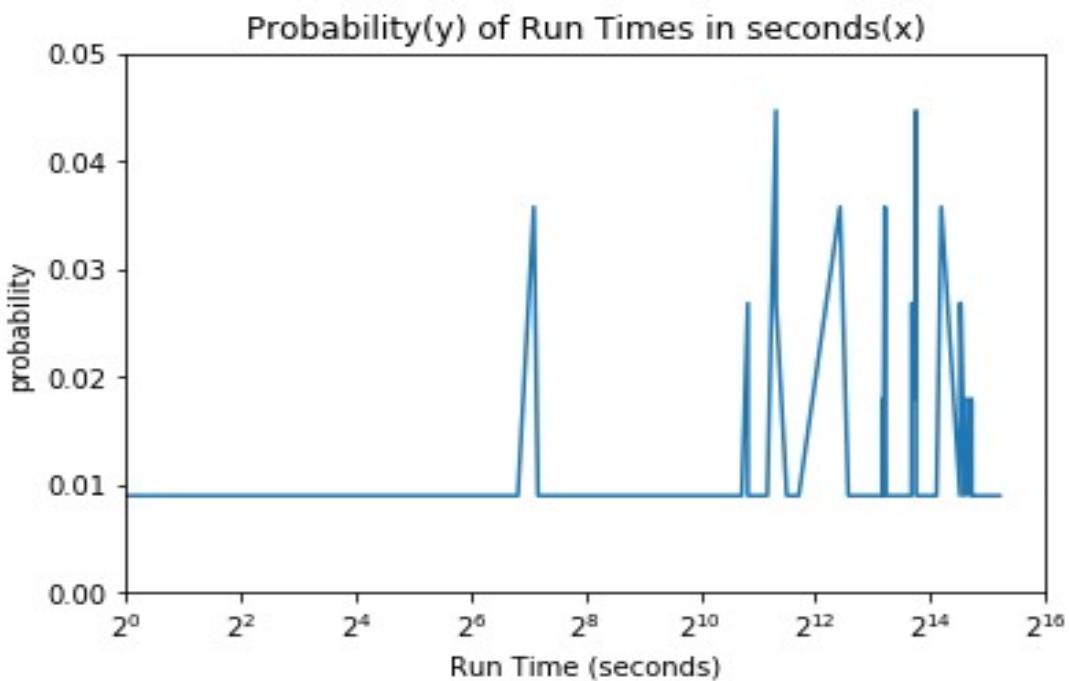


Figure 2-8

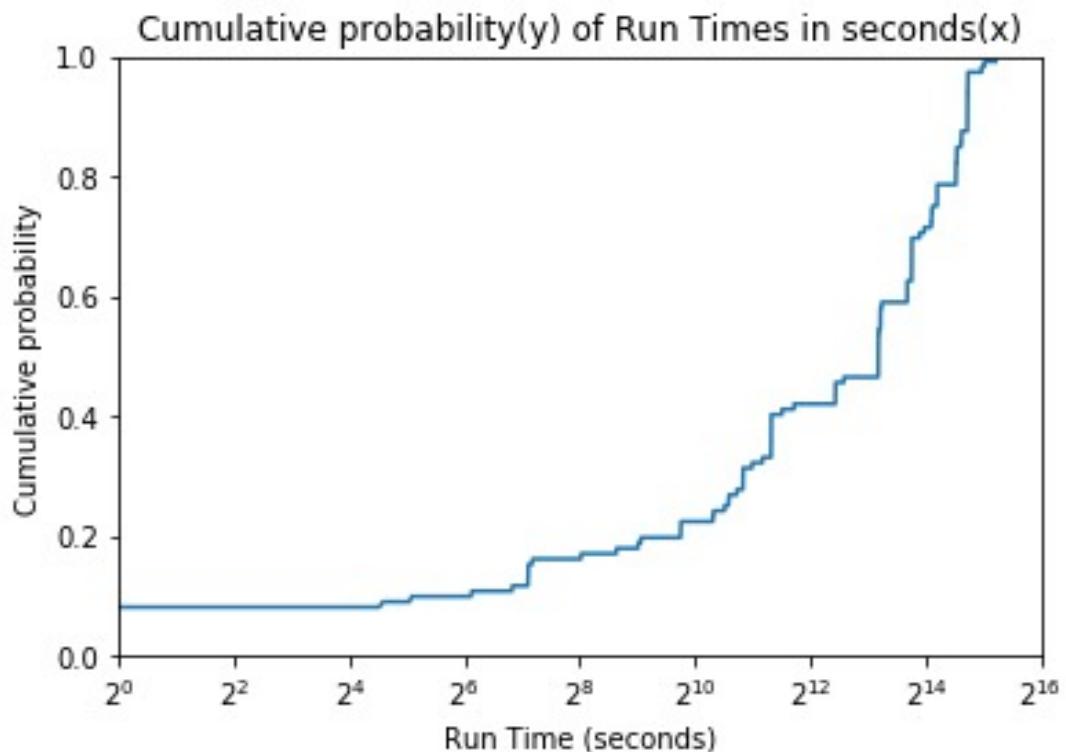


Figure 2-9

בגרף 2-8 (PDF) אפשר לראות כל מיני Flurries - Bursts אשר בעצם מראים שימוש שיט בכמה RunTimes נפוצים במערכת.

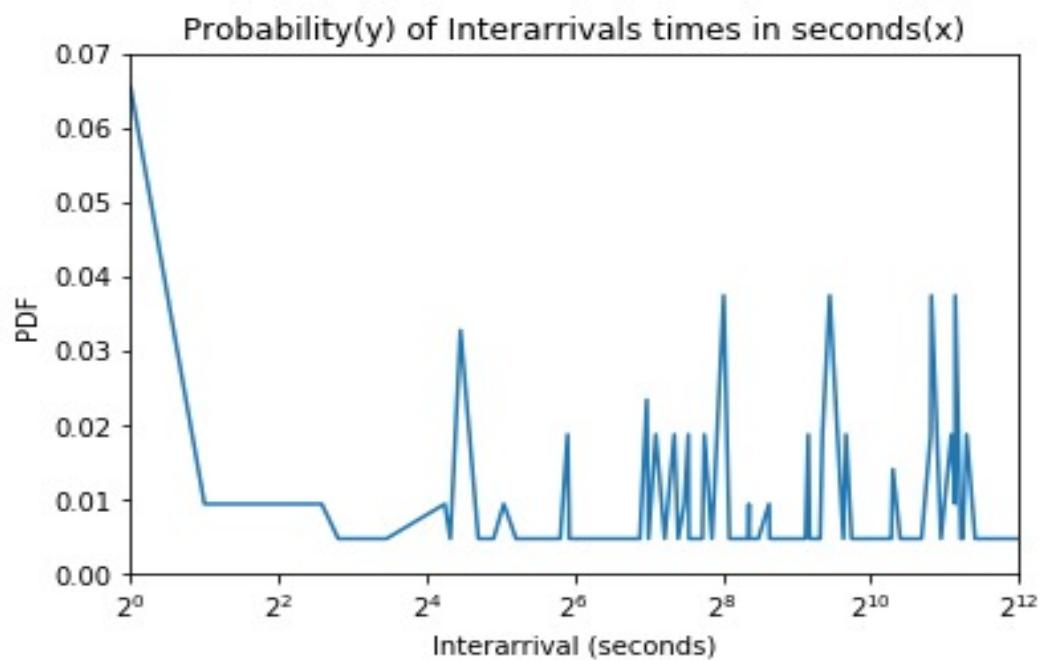


Figure 2-10

בדומה לגרף 8-2, גם בגרף 10-2 ניתן לראות כל מיני Flurries Bursts אשר מראים לנו شيئا נמה Interarrival times נפוצים.

בנוסף, ניתן לראות שכאשר $1 = X$, יש לנו הסתברות גדולה יחסית (~ 0.065) לשאר ה-Interarrivals, מה שמרמז שיש מגוון רחב של jobs אשר באים אחד אחריו שנייה אחת.

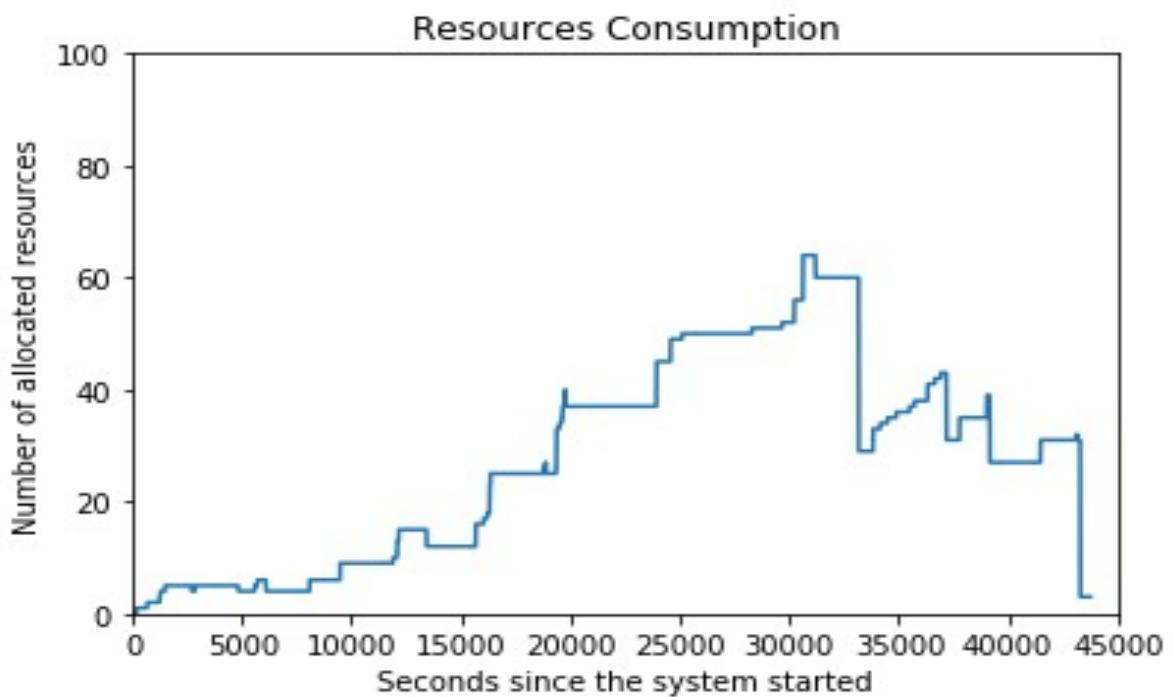


Figure 2-11

ניתן להסיק מגרף 2-2, ששעות העומס על המערכת זו בערך בין 17:40 ל-20:05. הגיוני להניח שהעומס ירד באופן משמעותי לאחר השעה 20:00 בערב, מכיוון בדרך כלל חלק גדול מהמשתמשים (עובד היי-טק וסטודנטים) מסיים עבודתו בסביבות השעה הוז. אפשרות נוספת היא שייתכן שהשתרת של MATLAB שחרר כל מיני משאבים ביחד באותו זמן.

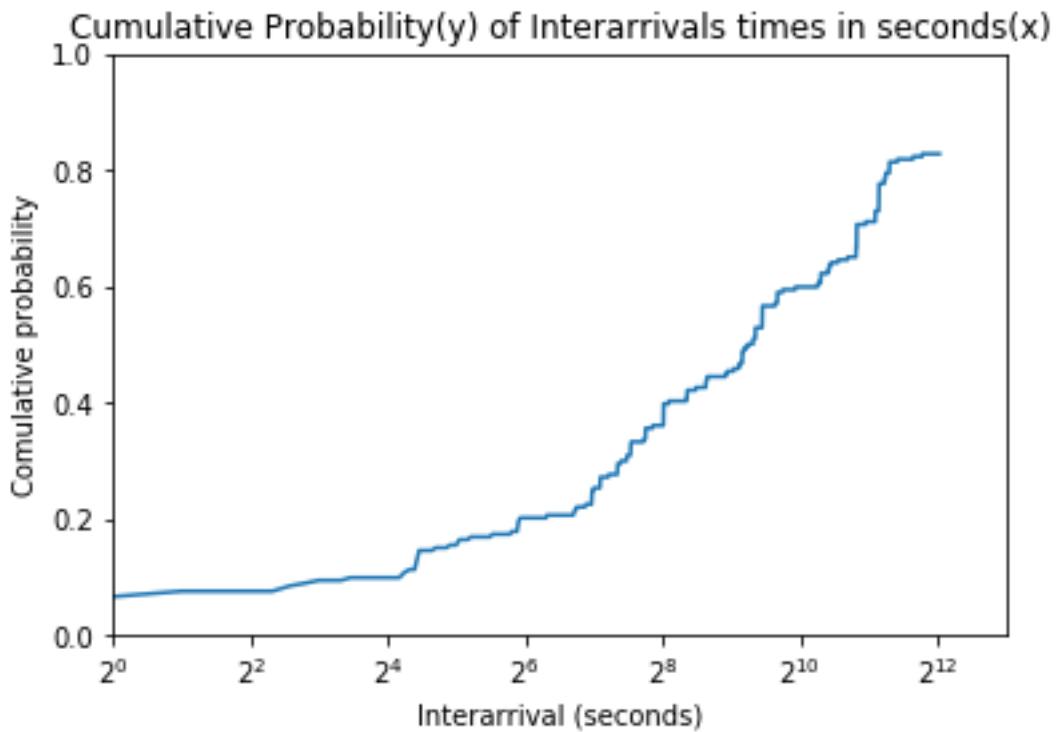


Figure 2-12

מענה על שאלות

שאלה 2:

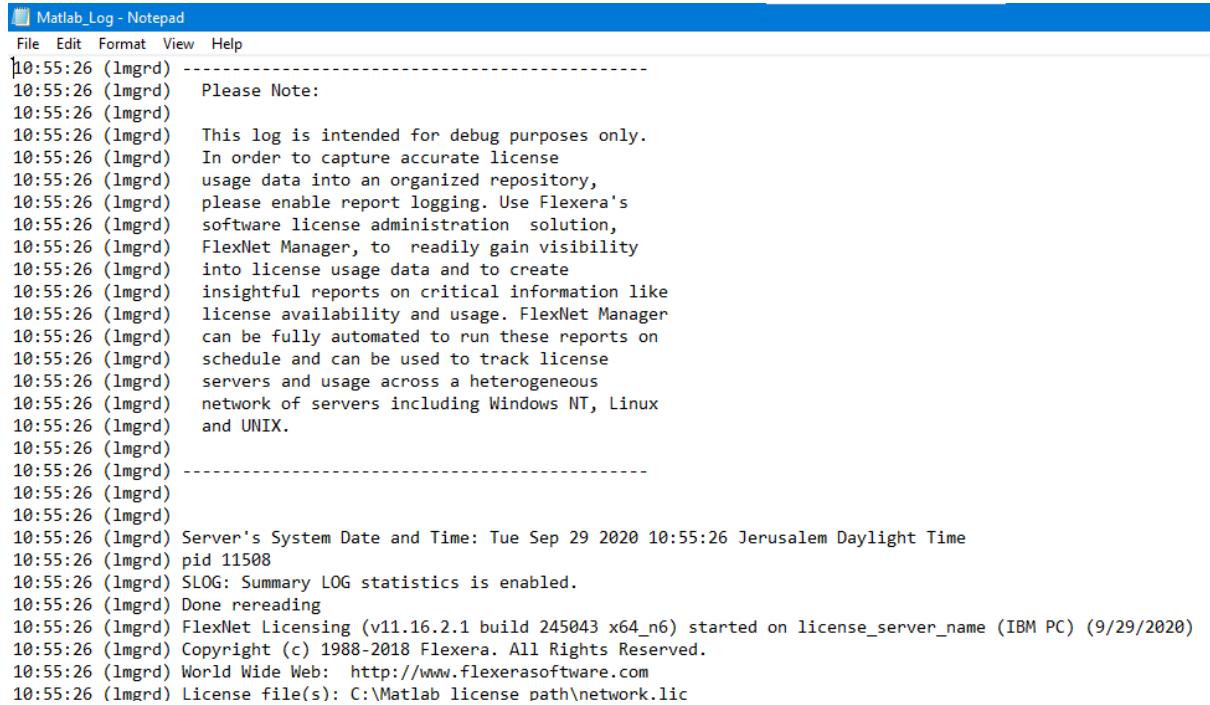
נקח למשל את גרף ה-*Processor Consumption* (גרף 7-2) של הלוג של NASA - ניתן לראות כי יש לנו פיק של 140% של Processors. זהה תופעה בעלת *z*-score Behaviour. Abnormal מיצג את מספר המעבדים ה"תפוסים" ע"י דבר זה נובע בغالל הסיבה פשוטה שבכל רגע נתון, הגרף הנויל מיצג את מספר המעבדים ה"תפוסים" ע"י הגיבים השונים ולא ניתן ל"שים את האצבע" על הבעיה והגיבים הספציפיים שגורם לפיק זהה.¹

שאלה 3:

בתריס של NASA, קיבלנו אותו נקי וmonic לשימוש, אך לא הצטרכנו לעשות הרבה בנדון, אלא רק לישם את הנדרש (SWF, גרפים וכו').

Dror G. Feitelson, "Workload Modeling: Computer Systems Performance Evaluation", p. 50 section 2¹

בטריס של ה-MATLAB שקיבלנו, אבן הינו למו כל מיני שורות שהיינו צריכים לנוקות – בಗו:



Matlab_Log - Notepad

File Edit Format View Help

```
10:55:26 (lmgrd) -----
10:55:26 (lmgrd) Please Note:
10:55:26 (lmgrd)
10:55:26 (lmgrd) This log is intended for debug purposes only.
10:55:26 (lmgrd) In order to capture accurate license
10:55:26 (lmgrd) usage data into an organized repository,
10:55:26 (lmgrd) please enable report logging. Use Flexera's
10:55:26 (lmgrd) software license administration solution,
10:55:26 (lmgrd) FlexNet Manager, to readily gain visibility
10:55:26 (lmgrd) into license usage data and to create
10:55:26 (lmgrd) insightful reports on critical information like
10:55:26 (lmgrd) license availability and usage. FlexNet Manager
10:55:26 (lmgrd) can be fully automated to run these reports on
10:55:26 (lmgrd) schedule and can be used to track license
10:55:26 (lmgrd) servers and usage across a heterogeneous
10:55:26 (lmgrd) network of servers including Windows NT, Linux
10:55:26 (lmgrd) and UNIX.
10:55:26 (lmgrd) -----
10:55:26 (lmgrd)
10:55:26 (lmgrd) -----
10:55:26 (lmgrd) Server's System Date and Time: Tue Sep 29 2020 10:55:26 Jerusalem Daylight Time
10:55:26 (lmgrd) pid 11508
10:55:26 (lmgrd) SLOG: Summary LOG statistics is enabled.
10:55:26 (lmgrd) Done rereading
10:55:26 (lmgrd) FlexNet Licensing (v11.16.2.1 build 245043 x64_n6) started on license_server_name (IBM PC) (9/29/2020)
10:55:26 (lmgrd) Copyright (c) 1988-2018 Flexera. All Rights Reserved.
10:55:26 (lmgrd) World Wide Web: http://www.flexerasoftware.com
10:55:26 (lmgrd) License file(s): C:\Matlab license path\network.lic
```

Figure 2-13

```

10:55:27 (MLM) Server started on license_server_name for: MATLAB
10:55:27 (MLM) SIMULINK MATLAB_5G_Toolbox AUTOSAR_Blockset
10:55:27 (MLM) Aerospace_Blockset Aerospace_Toolbox Antenna_Toolbox
10:55:27 (MLM) Audio_System_Toolbox Automated_Driving_Toolbox Bioinformatics_Toolbox
10:55:27 (MLM) Communication_Toolbox Video_and_Image_Blockset Control_Toolbox
10:55:27 (MLM) Curve_Fitting_Toolbox Signal_Blocks Data_Acq_Toolbox
10:55:27 (MLM) Database_Toolbox Datafeed_Toolbox Neural_Network_Toolbox
10:55:27 (MLM) Econometrics_Toolbox RTW_EMBEDDED_Coder Filter_Design_HDL_Coder
10:55:27 (MLM) Fin_Instruments_Toolbox Financial_Toolbox Fixed_Point_Toolbox
10:55:27 (MLM) Fuzzy_Toolbox GPU_Coder GADS_Toolbox
10:55:27 (MLM) Simulink_HDL_Coder EDA_Simulator_Link Image_Acquisition_Toolbox
10:55:27 (MLM) Image_Toolbox Instr_Control_Toolbox LTE_Toolbox
10:55:27 (MLM) MATLAB_Coder MATLAB_Builder_for_Java Compiler
10:55:27 (MLM) MATLAB_Report_Gen MAP_Toolbox Mixed_Signal_Blockset
10:55:27 (MLM) MPC_Toolbox MBC_Toolbox Motor_Control_Blockset
10:55:27 (MLM) Navigation_Toolbox OPC_Toolbox Optimization_Toolbox
10:55:27 (MLM) Distrib_Computing_Toolbox PDE_Toolbox Phased_Array_System_Toolbox
10:55:27 (MLM) Powertrain_Blockset Pred_Maintenance_Toolbox RF_Blockset
10:55:27 (MLM) RF_Toolbox ROS_Toolbox Reinforcement_Learn_Toolbox
10:55:27 (MLM) Risk_Management_Toolbox Robotics_System_Toolbox Robust_Toolbox
10:55:27 (MLM) Sensor_Fusion_and_Tracking SerDes_Toolbox Signal_Toolbox
10:55:27 (MLM) SimBiology SimEvents SimDriveline
10:55:27 (MLM) Power_System_Blocks SimHydraulics SimMechanics
10:55:27 (MLM) Simscape_Virtual_Reality_Toolbox SL_Verification_Validation
10:55:27 (MLM) Simulink_Code_Inspector Real-Time_Workshop Simulink_Compiler
10:55:27 (MLM) Simulink_Control_Design Simulink_Coverage Simulink_Design_Optim
10:55:27 (MLM) Simulink_Design_Verifier Real-Time_Win_Target Simulink_PLC_Coder
10:55:27 (MLM) XPC_Target SIMULINK_Report_Gen Simulink_Requirements
10:55:27 (MLM) Simulink_Test SoC_Blockset Excel_Link
10:55:27 (MLM) Stateflow Statistics_Toolbox Symbolic_Toolbox
10:55:27 (MLM) System_Composer Identification_Toolbox Text_Analytics_Toolbox
10:55:27 (MLM) Trading_Toolbox Vehicle_Dynamics_Blockset Vehicle_Network_Toolbox
10:55:27 (MLM) Vision_HDL_Toolbox WLAN_System_Toolbox Wavelet_Toolbox

```

Figure 2-14

למשל - תיעוד של השרת (תמונה 2-13) ושל ה-TOOLBOXES הקיימים שנמענו (תמונה 2-13).

```

14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:50:03 Jerusalem Daylight Time,#4,(3744K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:46:31 Jerusalem Daylight Time,#4,(3748K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:40:03 Jerusalem Daylight Time,#4,(3748K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:36:31 Jerusalem Daylight Time,#5,(3744K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:33:38 Jerusalem Daylight Time,#5,(3740K),(user7,computer7,696)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:30:03 Jerusalem Daylight Time,#5,(3740K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:26:31 Jerusalem Daylight Time,#5,(3744K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:23:38 Jerusalem Daylight Time,#5,(3744K),(user7,computer7,696)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:20:03 Jerusalem Daylight Time,#5,(3744K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:18:52 Jerusalem Daylight Time,#5,(3688K),(user7,computer7,696)
14:55:28 (MLM) (@MLM-SLOG@)

14:55:28 (MLM) (@MLM-SLOG@) === Top 10 Peak Client Requests Processing Time (in ms) ===
14:55:28 (MLM) (@MLM-SLOG@) Time: Tue Sep 29 2020 14:55:28 Jerusalem Daylight Time
14:55:28 (MLM) (@MLM-SLOG@) Request processing time, when, #concurrent clients, (private bytes (in KB)), client info (user, node, FD)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:46 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:55:29 Jerusalem Daylight Time,#3,(3900K),(user8,license_server_name,640)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:45 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:45 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:48 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:51 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 11:17:44 Jerusalem Daylight Time,#5,(3744K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 11:18:18 Jerusalem Daylight Time,#5,(3688K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 11:20:27 Jerusalem Daylight Time,#6,(3688K),(user4,computer4,696)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 11:20:32 Jerusalem Daylight Time,#6,(3688K),(user4,computer4,696)
14:55:28 (MLM) (@MLM-SLOG@)

14:55:28 (MLM) (@MLM-SLOG@) === Top 10 Peak In-house Operations time (in ms) ===
14:55:28 (MLM) (@MLM-SLOG@) NOTE: Peak times greater than 4 seconds get recorded.
14:55:28 (MLM) (@MLM-SLOG@) Time: Tue Sep 29 2020 14:55:28 Jerusalem Daylight Time
14:55:28 (MLM) (@MLM-SLOG@) In-house operation time, when, #concurrent clients
14:55:28 (MLM) (@MLM-SLOG@)

14:55:28 (MLM) (@MLM-SLOG@) === Active Connections Info ===
14:55:28 (MLM) (@MLM-SLOG@) Peak active connections #6 attempted at Tue Sep 29 2020 11:20:01 Jerusalem Daylight Time
14:55:28 (MLM) (@MLM-SLOG@)
14:55:28 (MLM) (@MLM-SLOG@) === Memory Usage Info ===
14:55:28 (MLM) (@MLM-SLOG@) Peak private bytes 3984K attempted at Tue Sep 29 2020 10:58:33 Jerusalem Daylight Time

```

Figure 2-15

ובנוסף, סיכום מעת לעת של המשתמשים ושל החיבורים הפעילים במערכת ברגע (תמונה 2-15).

במובן של אחר הנקוי, נשารנו עם מידע נקי, אשר יוכל להשתמש בו לצורך בניית ה-.SWF. חלק מՄՏת נקיון המידע, החלנו להשאיר את השורות בעלות הצורה של IN/OUT, ובאמצעותן יכולנו לקבל את מלאו המידע הדרוש במשימה הנוכחית ובנסיבות הבאות.

אין כל מה להשוות בין התוצאות שהיינו מקבלים בלי הנקוי לעיל, מכיוון שבכל מה שניקינו – יכולנו לקבל מהמייד שנסחר (חיבורים פעילים, שמות משתמשים, שעת התחלת סיום וכו'...).

לכן, התוצאה הסופית של הלוג הנקוי הייתה רק שורות של OUT/IN ובאמצעותן יכולנו לדעת עברו כל משתמש וכל משאב מי, מתי ובמה השתמשו בו למשל.

שלב 3 – התאמות בין התפלגות לבן תוצאות בתחום המידע

נתחיל להתאים כל גרפף להתפלגות מסוימת.

Figure 2-1 - Multiple Erlang distributions:

ניתן לראות שגרף זה "מורכב" משתי התפלגות Erlang, כאשר הפרמטר k יהיה $2 = .$

Figure 2-2 - CDF of Erlang distribution:

באוטו אופן לגרף מס' 1, נראה כי גרפף זה מייצג לנו CDF של התפלגות Erlang כאשר הפרמטר k יהיה $= 2.$

Figure 2-3 - with tail of exponential distribution: More than one Hyper-Erlang

ניתן לראות שגרף זה הינו שילוב של שתי התפלגות – התפלגות Erlang של user ותפלגות Erlang של user special. לכן, שילוב של שתיהן ייתן לנו Hyper-Erlang distribution, כאשר הפרמטרים יהיו $k = 1, p = P(\text{user}), 1 - p = P(\text{special user})$

Figure 2-4 - CDF of Hyper-Erlang:

באוטו אופן לגרף מס' 3, נראה כי גרפף זה מייצג לנו CDF של התפלגות Hyper-Erlang כאשר ערכי הפרמטרים זהים לgraf ה-PDF.

Figure 2-5 אין התפלגות שמתאימה להיסטוגרמות

Figure 2-6 אין התפלגות שמתאימה להיסטוגרמות

Figure 2-7 אין התפלגות שמתאימה להיסטוגרמות

Figure 2-8 Mixed lognormal:

ניתן לראות שגרף זה מתפלג בצורה "יחסית" נורמלית, אך גרפף זה נמצא במרחב לוגריתמי (ציר ה- x הינו סקללה לוגריתמית), ולכן גרפף זה מתאים להתפלגות לוג-נורמלית. בסיסן נשים לב שה-mean וה-standard deviation נמצאים במרחב לוגריתמי.

Figure 2-9 -

לפי המידע על התפלגות הלוג-נורמלית, אין צורה חד-משמעות לגבי ה-CDF.

Figure 2-10 -

גרף זה מטעה קלות – ראשית, נשים לב שהוא נמצא במרחב לוגריתמי. אם נסתכל על ה-interval שנמצא מ- $2^{12} - 2^4$, ניתן לחשב שמדובר בהתפלגות לוג-נורמלית (מרחב לוגריתמי, התפלגות "יחסית" נורמלית). מצד שני, ה-interval שנמצא בתחלת הגרפף עד 2^4 לא ברור לగמרי.

Figure 2-12 - Normal distribution:

ניתן לראות בgraf זה "עקומת פעמון" אשר מאפיינת את העומס על המערכת (הSKU ששהעומס נובע ישירות מפיזור שעות העבודה בכל יום – עלייה בבקרים וירידה בערבבים).

שלב 4 – התאמות בין גרפים לבון התפלגיות אמפיריות

NASA-Log File Graphs

CDF of Runtimes time – All users

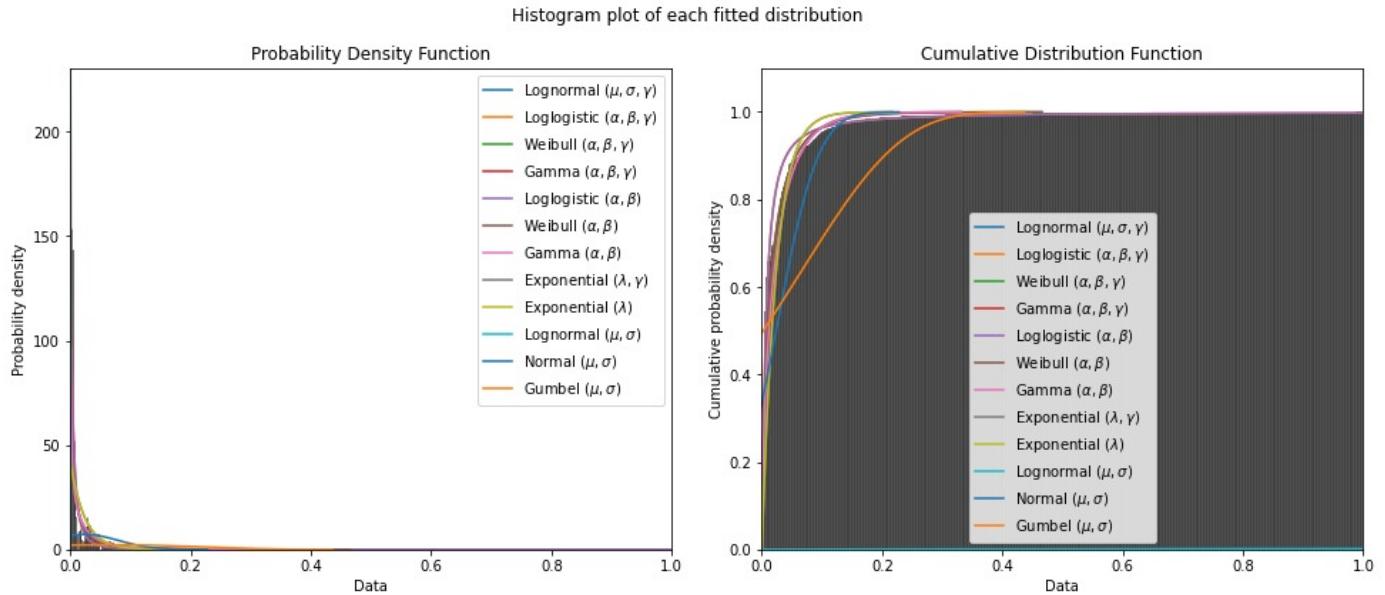


Figure 4-1

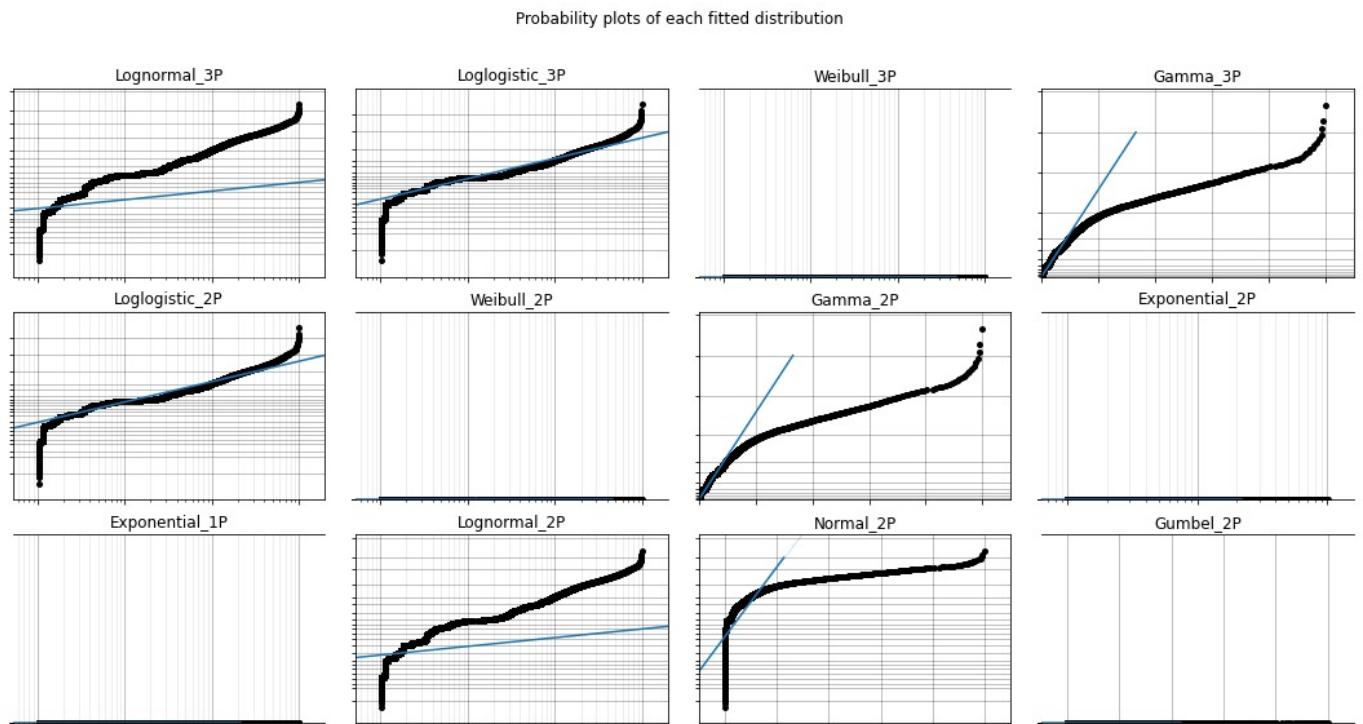


Figure 4-2

Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)

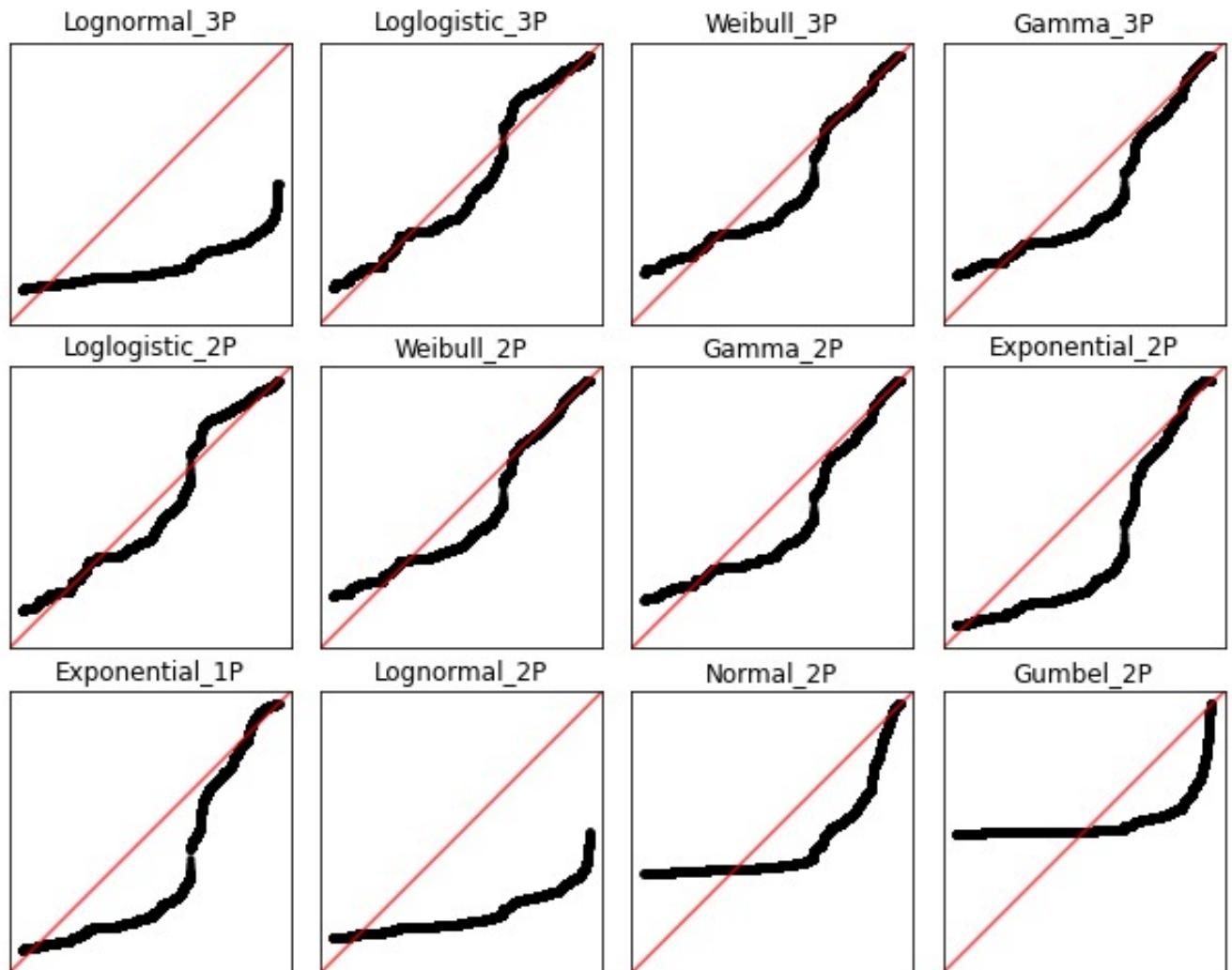


Figure 4-3

CDF of Runtimes time – Other users

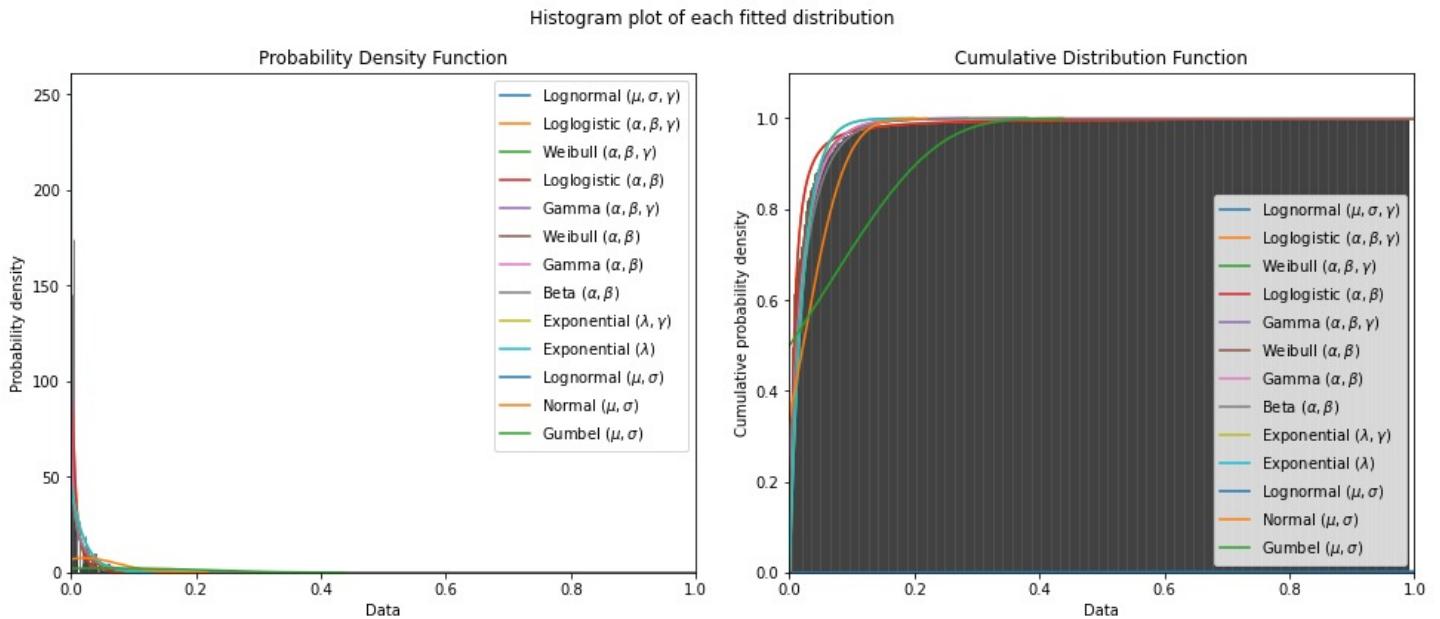


Figure 4-4

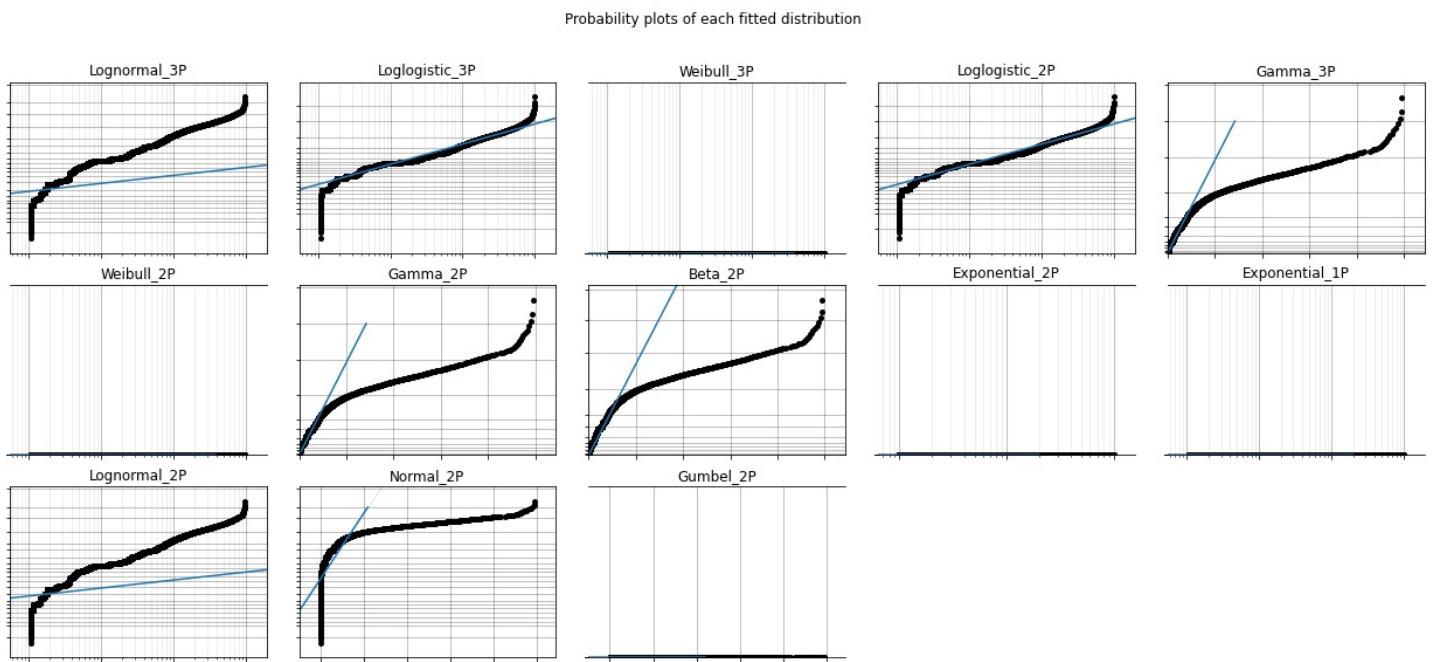


Figure 4-5

Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)

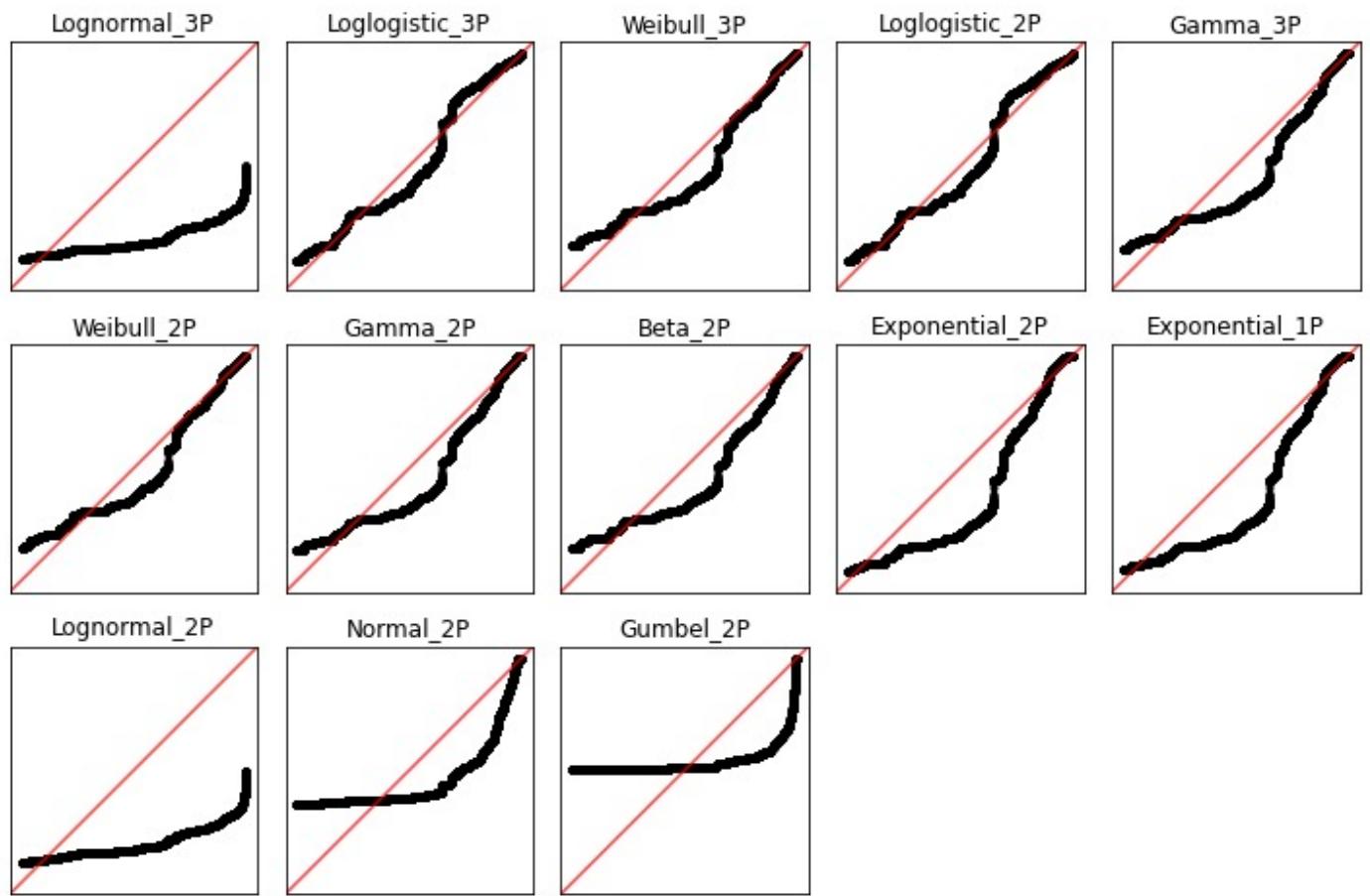


Figure 4-6

CDF of Runtimes time – Special users

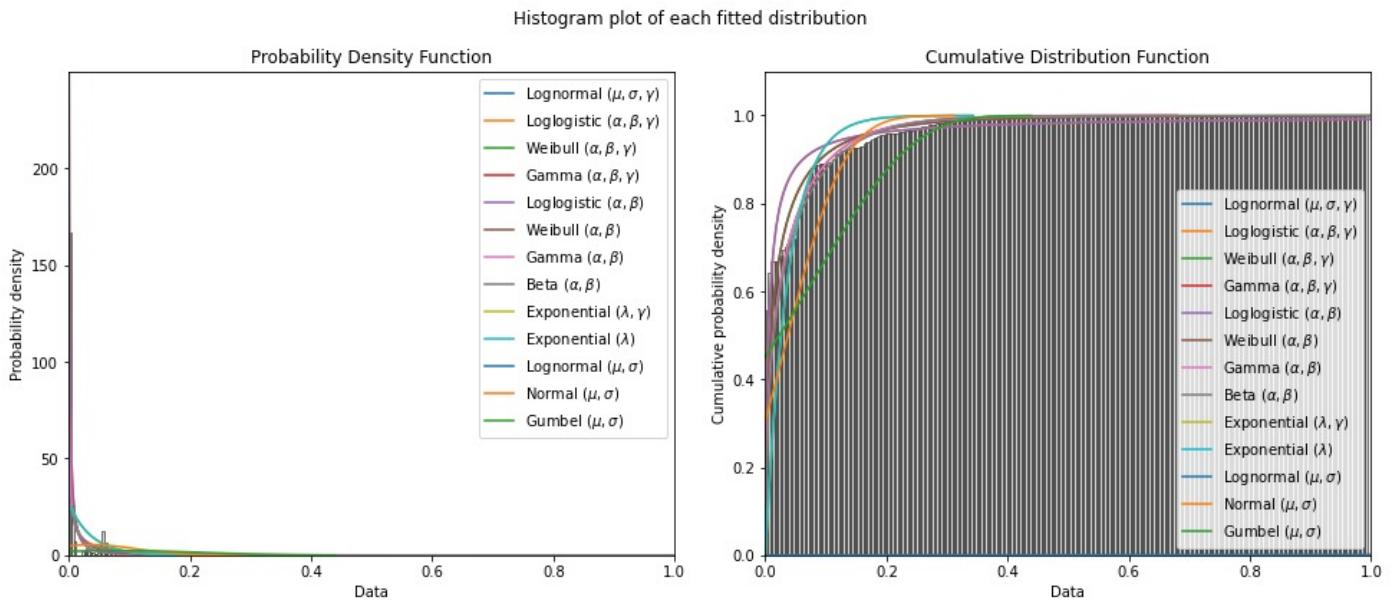


Figure 4-7

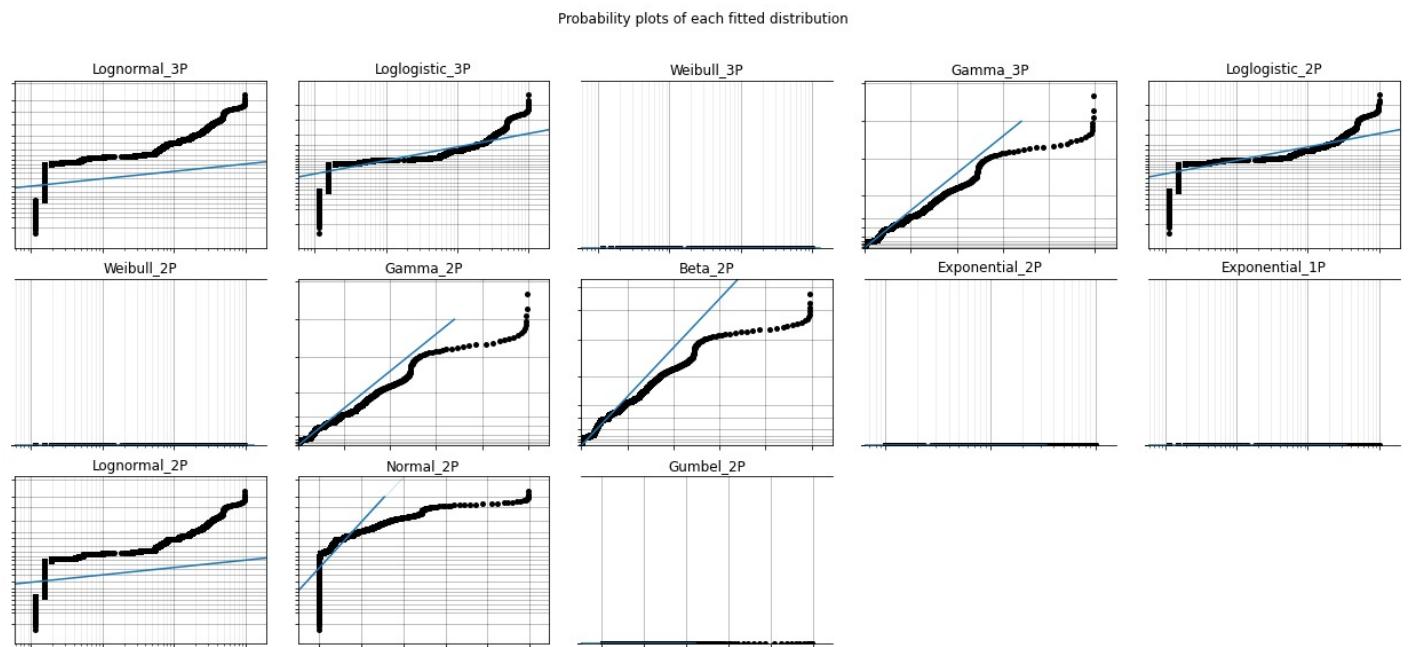


Figure 4-8

Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)

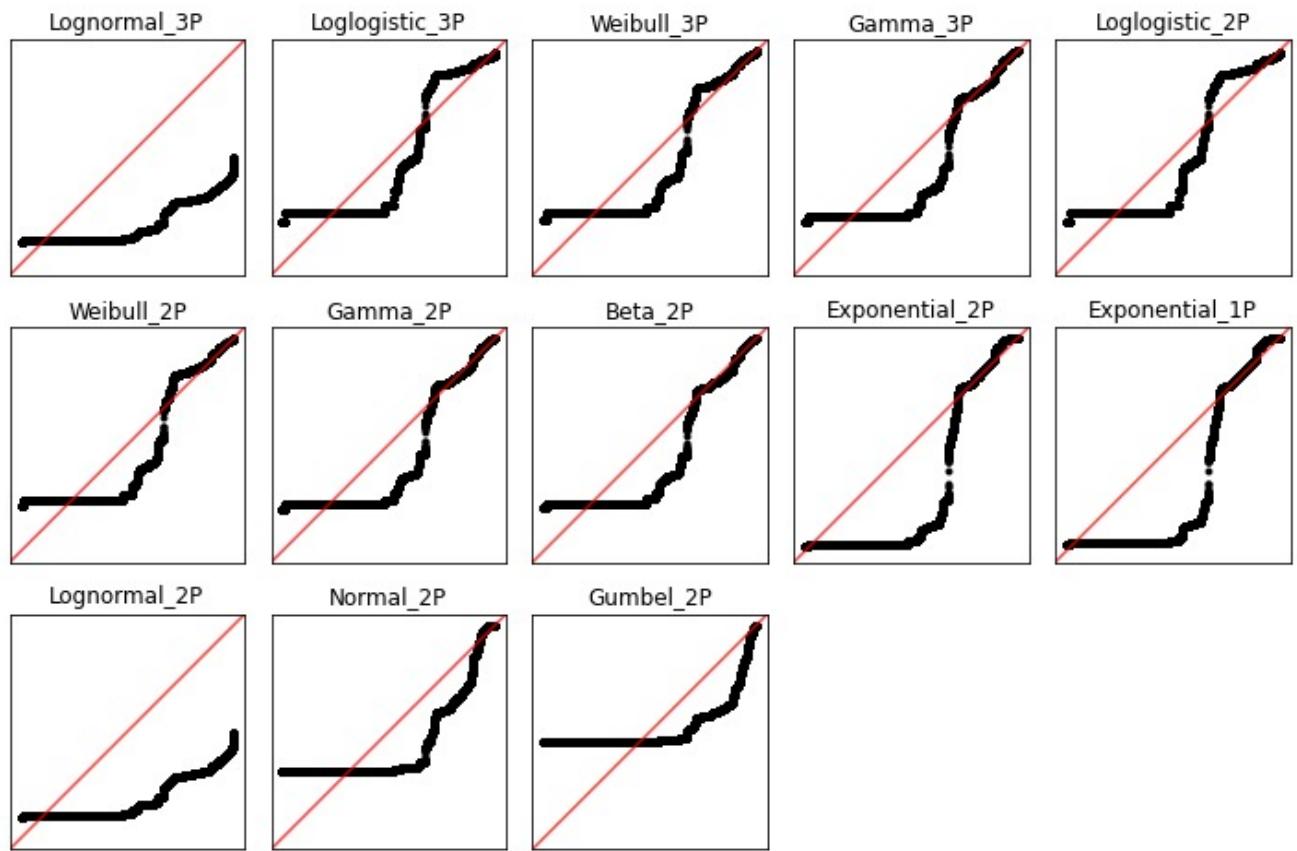


Figure 4-9

CDF of Interarrival time

Histogram plot of each fitted distribution

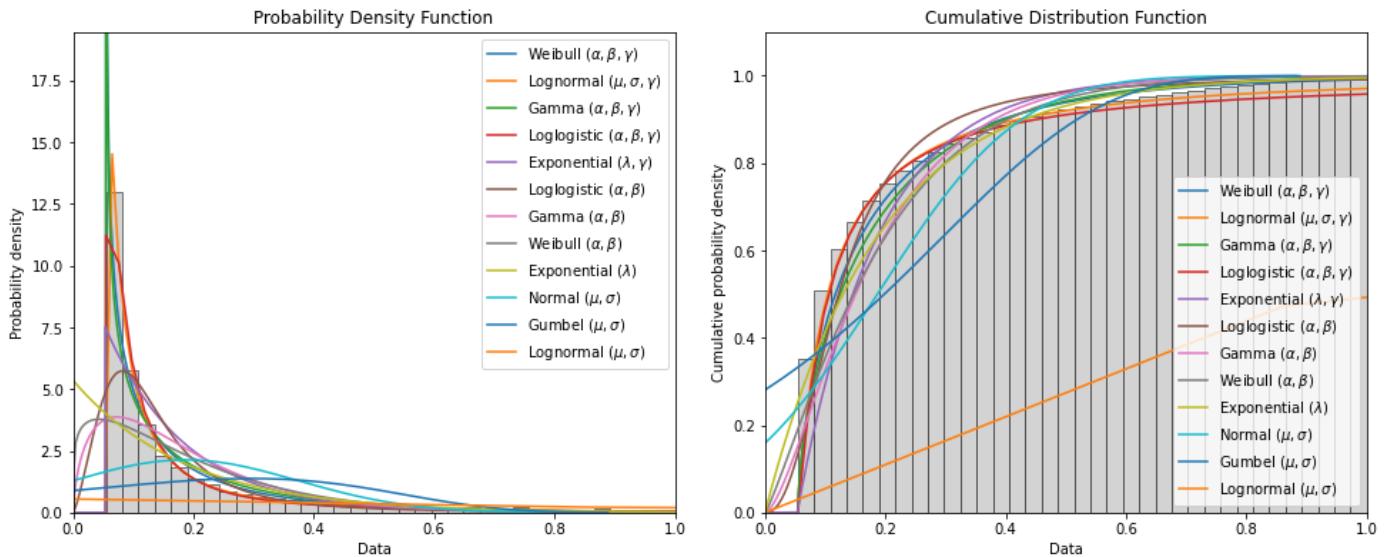


Figure 4-10

Probability plots of each fitted distribution

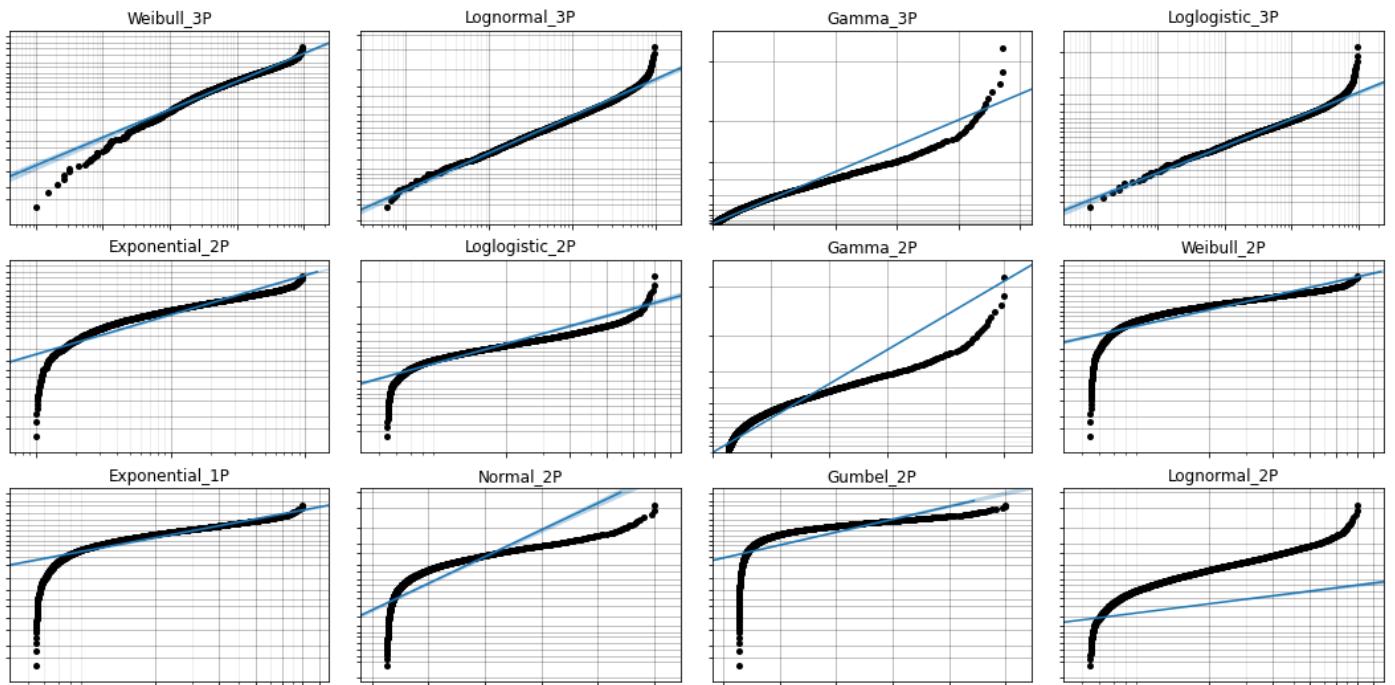


Figure 4-11

Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)

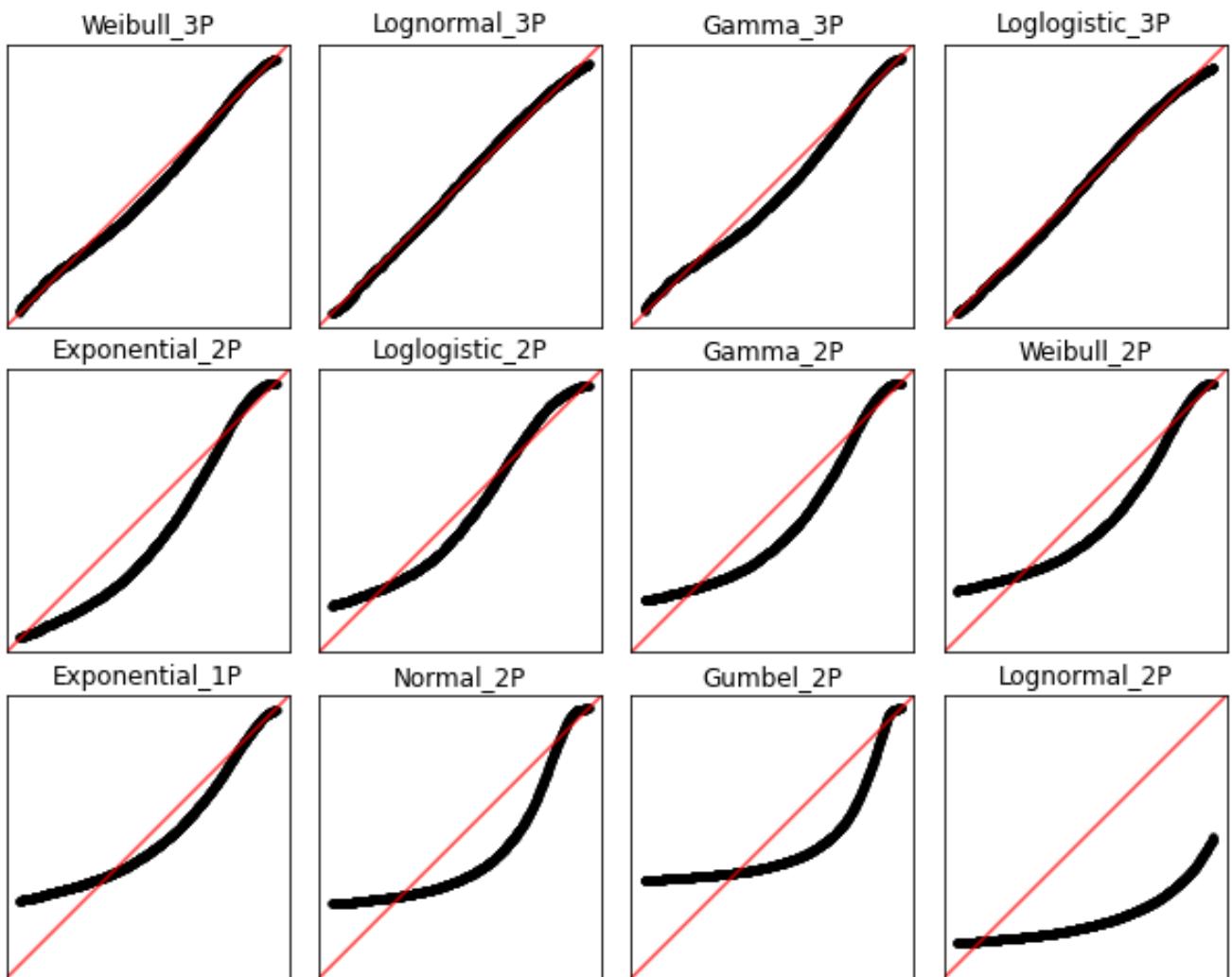


Figure 4-12

MATLAB-Log File Graphs

CDF of Runtimes

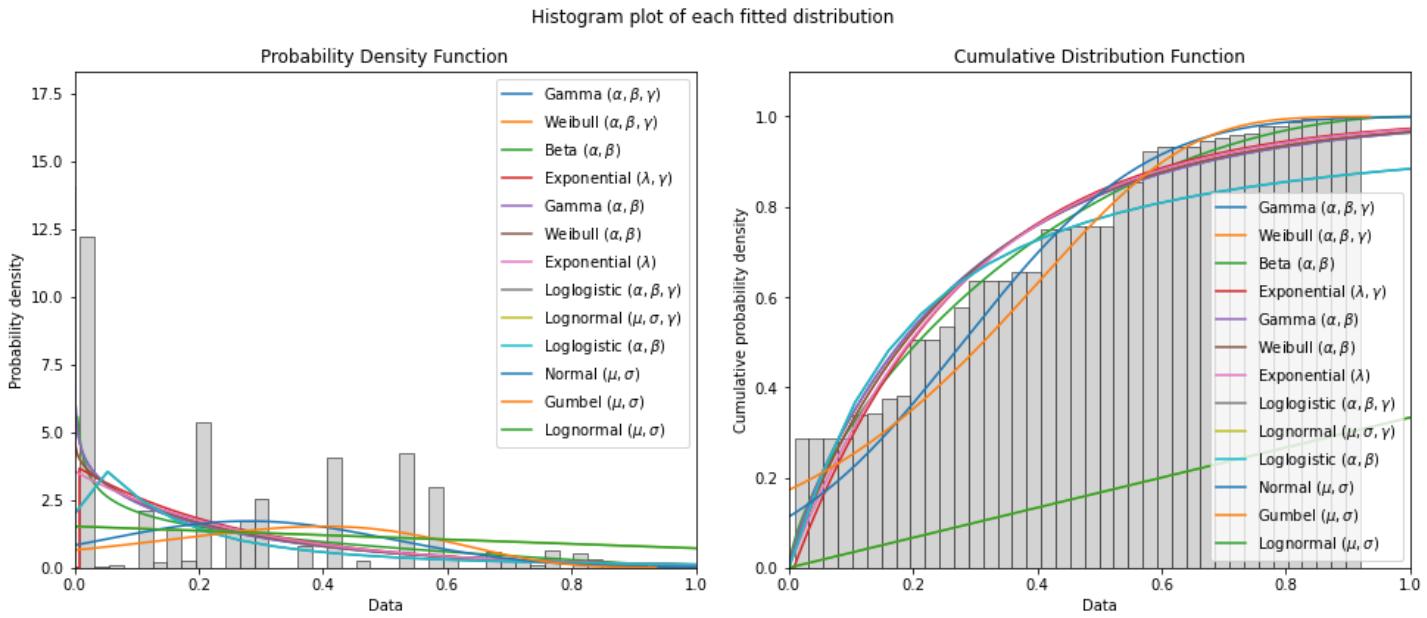


Figure 4-13

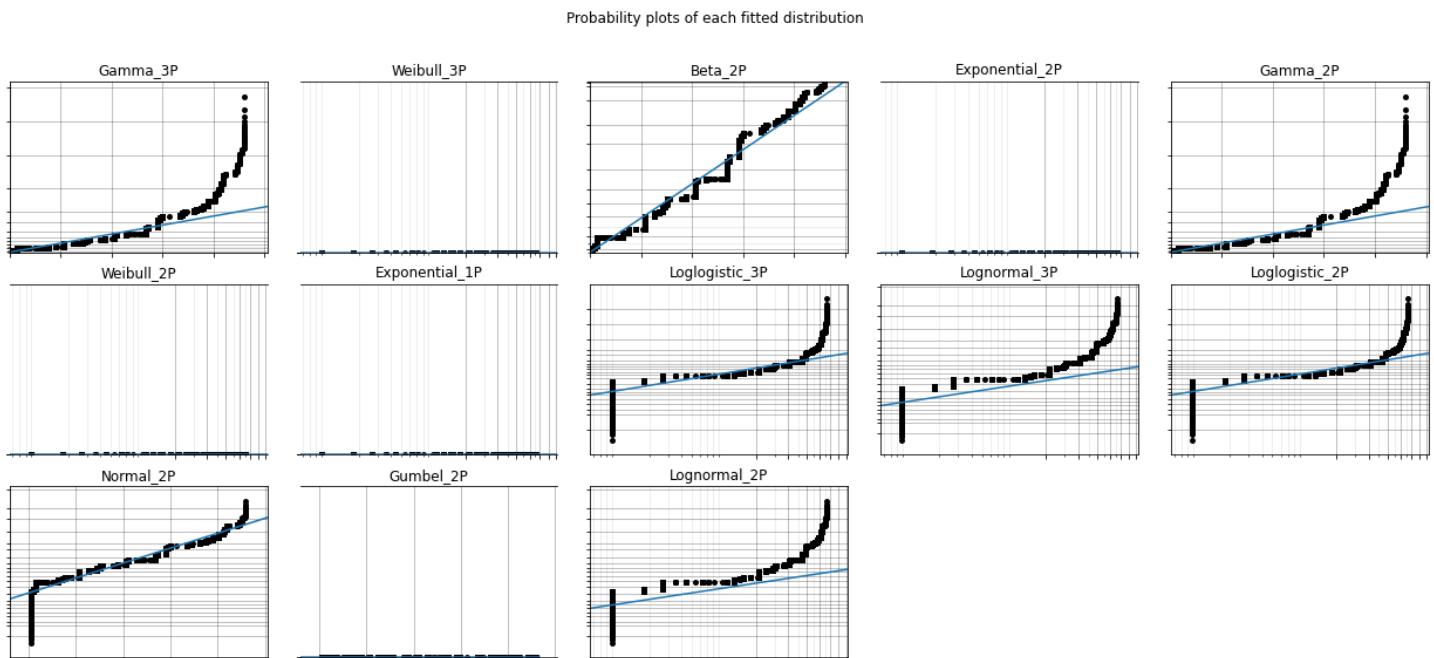


Figure 4-14

Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)

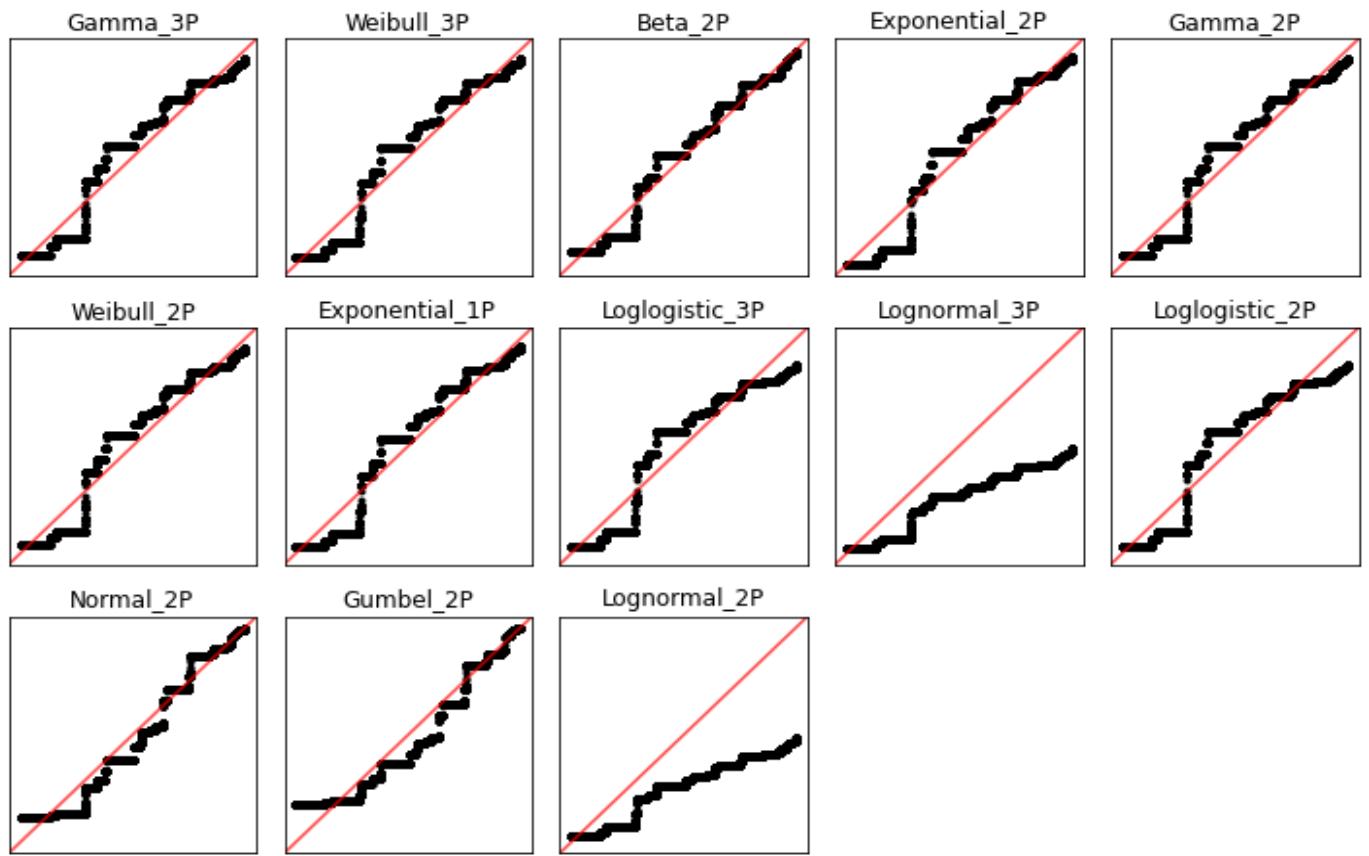


Figure 4-15

CDF of Interarrival time

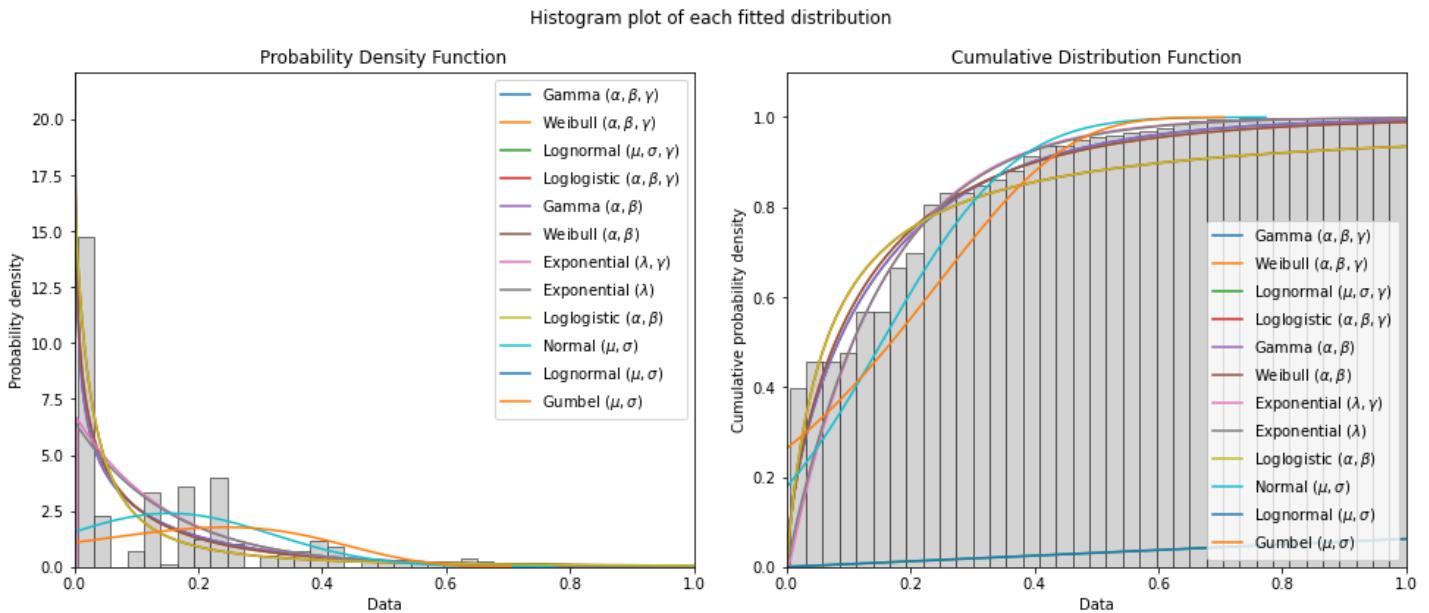


Figure 4-16

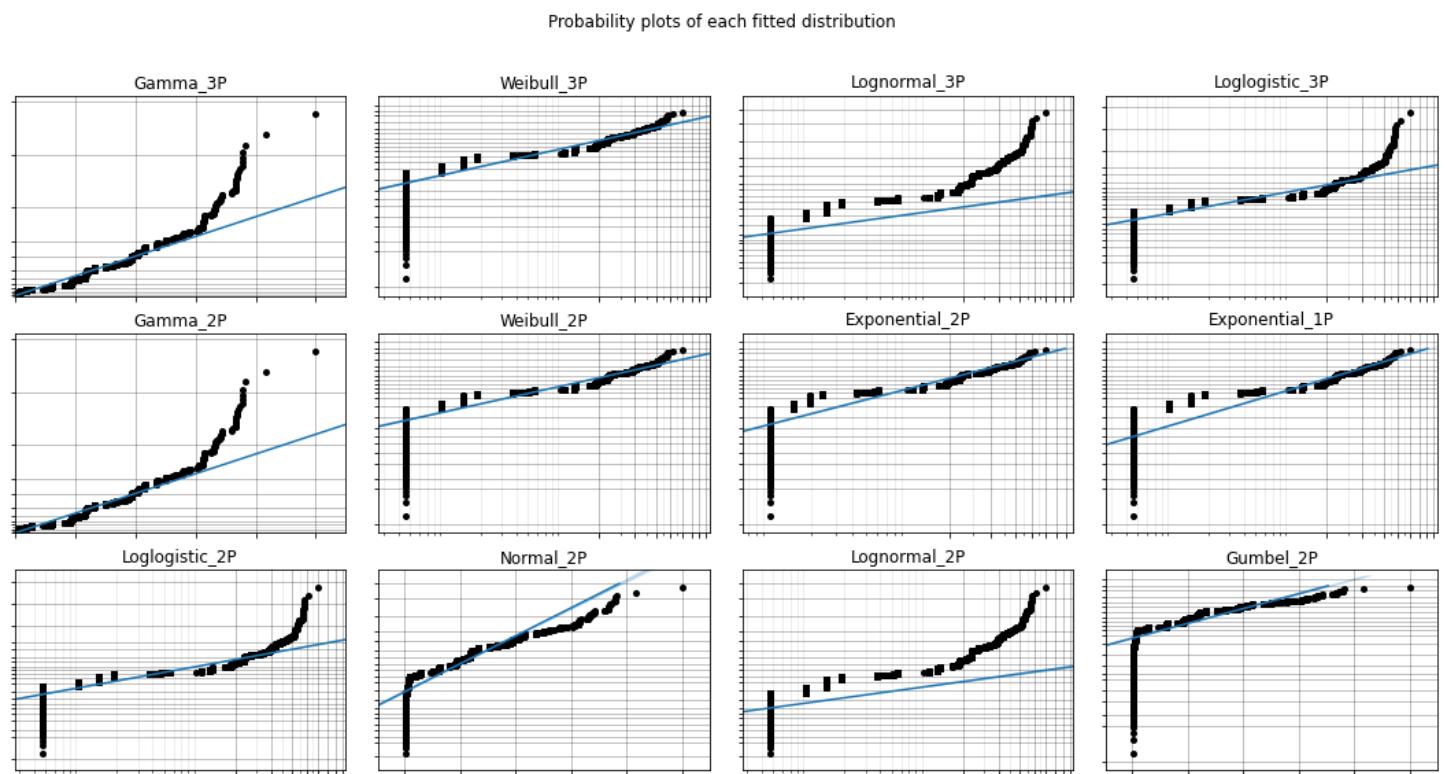


Figure 4-17

Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)

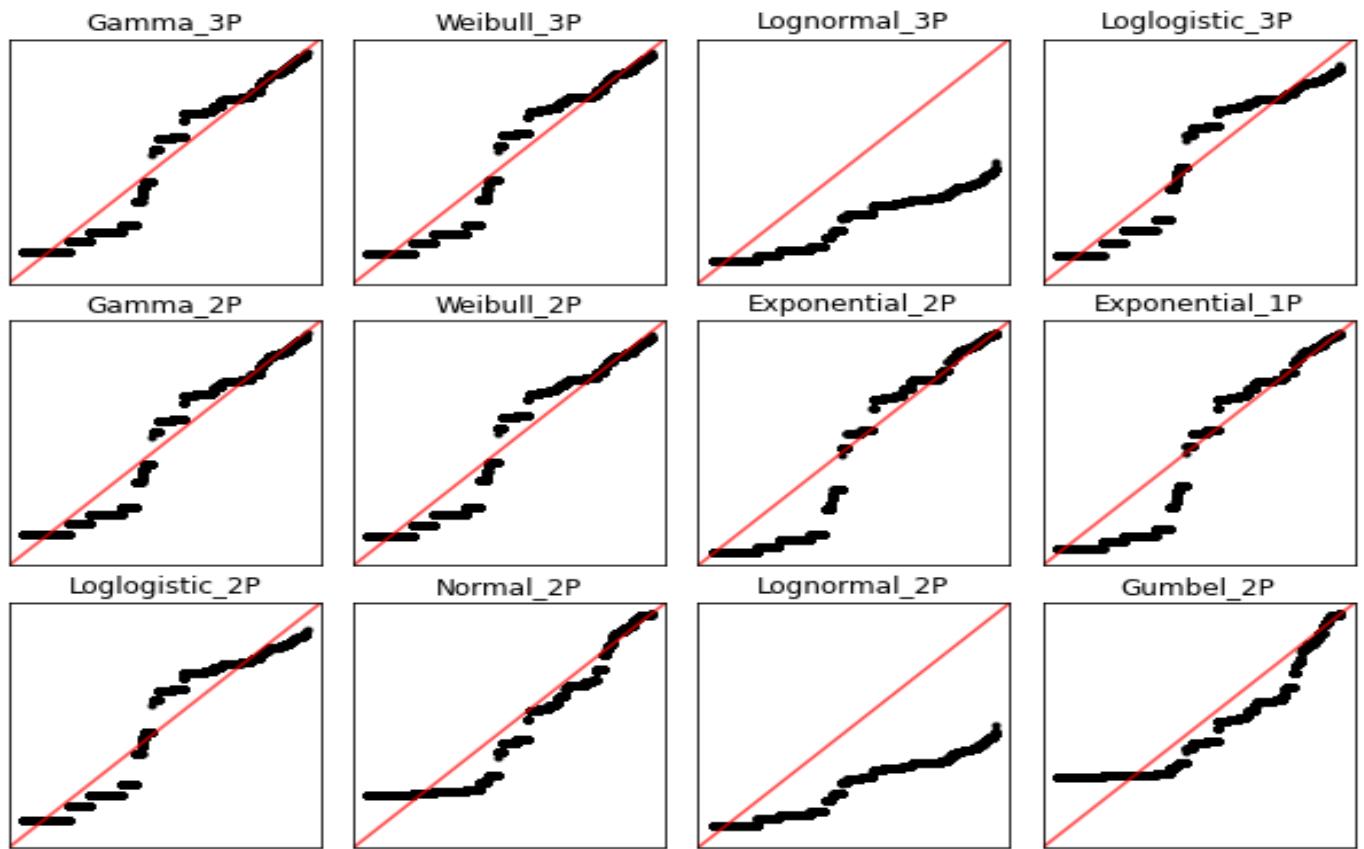


Figure 4-18

שלב 5 – התפלגות Zipf

נראה ש-2 figure (במאות jobs לכל user במערכת) יכול להיות מתואר לפי התפלגות Zipf.

בדומה לגרף Zipf של הסתברות הופעות מילים בקטע טקסט, ניצור גרף לפי אותו אופן. באמצעות גראף מספר 6 אשר מתאר לנו את במות ה-jobs לכל user, נוכל למצוא את ההסתברות להופעת job של user ספציפי.

נציג זאת בגרף:

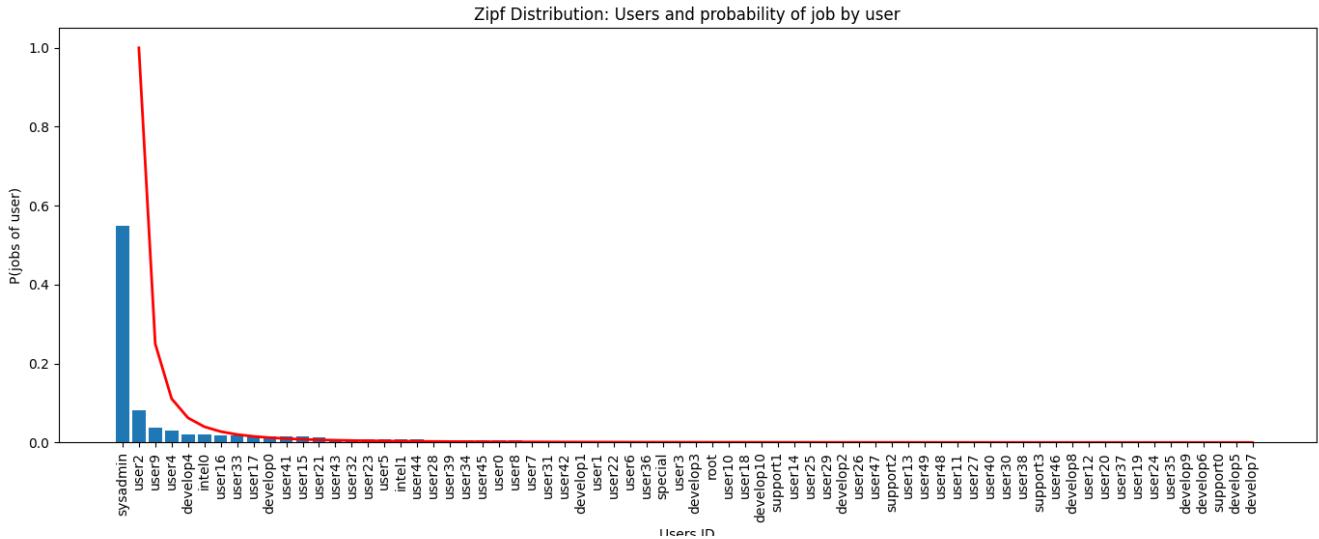


Figure 5-1

ציר ה-x יהיה המספר הסידורי של users, ציר ה-y יהיה ההסתברות להופעת job מאותו user. הקן האדום מתאר את הפונקציה שנוצרת משימוש בתפלגות Zipf על הערכים הנ"ל.

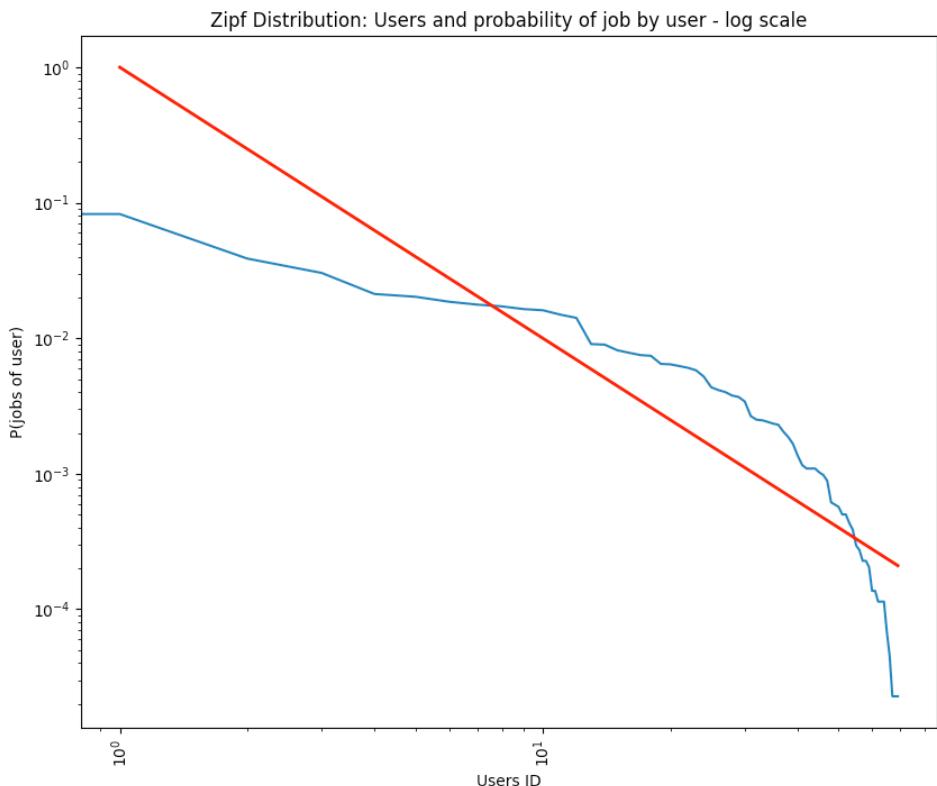


Figure 5-2

בשביל לראות יותר בנוחות את הגרף, נהפוך את העמודות לקו רציף וכייצג את צירים בסקללה לוגריתמית: ניתן לראות שההתאמה לא טובה כפי שציפינו. ננסה לקבץ מספר דוגמאות ייחודי בשבייל התאמה טובה יותר. בגרף הבא, נעשה קיבוץ על 5 דוגמאות. ניתן לראות שיש לנו שיפור כל.

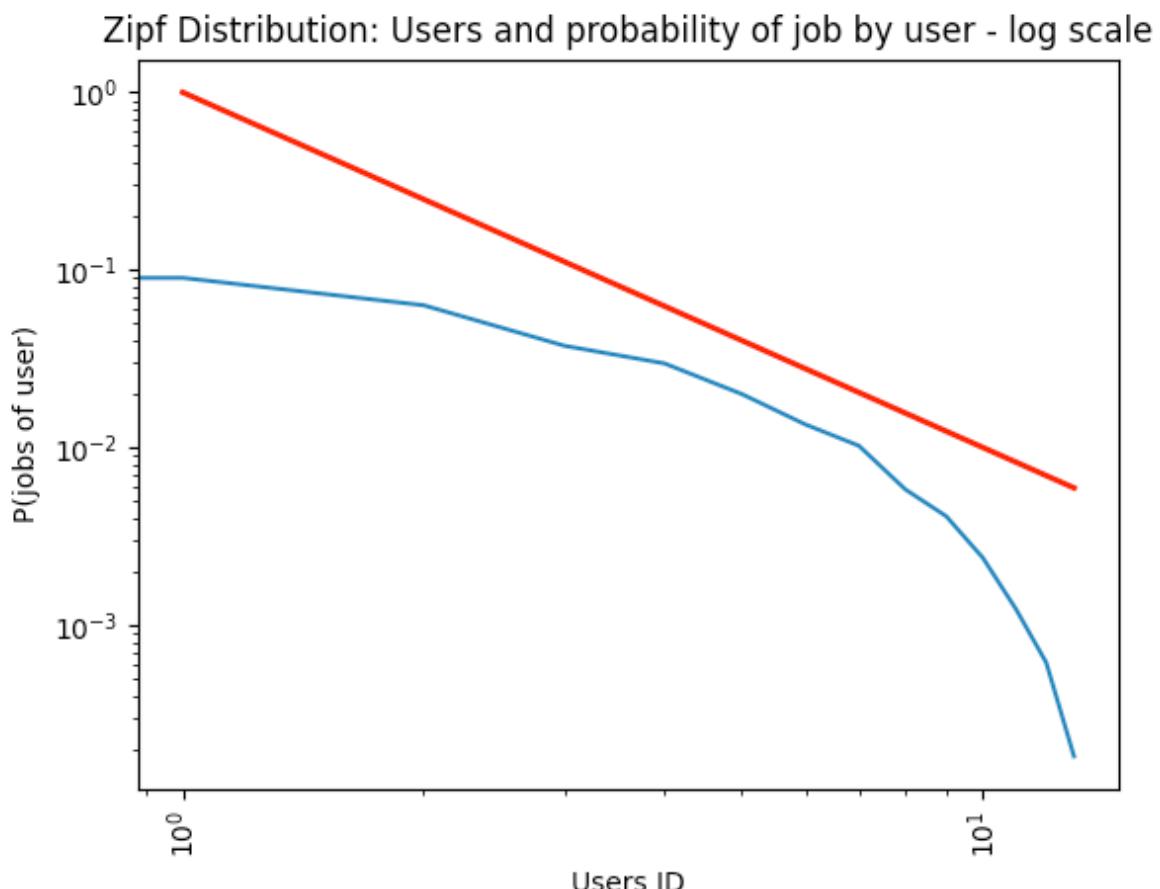


Figure 5-3

נססה לקבץ כמהות גדולה יותר של דוגימות – הפעם נקבע 7 דוגימות ונראה את הגרף הבא:

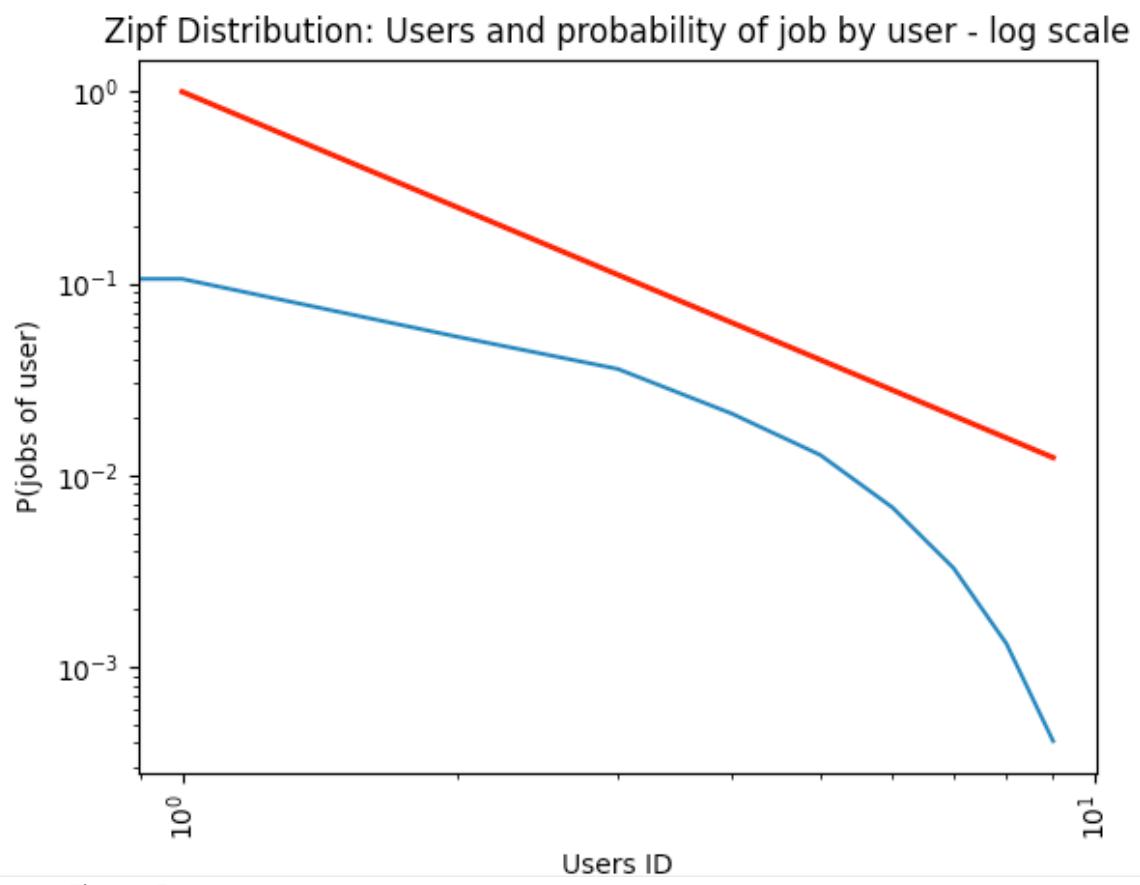


Figure 5-4

נשים לב שה-fitting בגרף עם קיבוץ של 7 דוגימות יותר מתאים לפונקציית Zipf.

שלב 6

NASA-Log File

אם נסתכל על גרף ה-CDF (גרף 4), ניתן לראות שלרוב ה-Jobs יש Run time הקטן יותר מ-12,000 שניות (פחות או יותר).

מצד שני, המשתמשים במחשב של NASA לא "אהבים" לחכות עד אשר ה-Job הקודם מסתיים, והם מתחילה להריץ את ה-Job שלהם במקביל עם ה-Job הקודם (שעוד לא נגמר). בתוצאה מתוופה זו, אנחנו מסיקים שהוא לא חובה של-Jobs ארוכים יותר, יש Interarrival Arivals יותר.

ניתן לשימוש לב לוחת גם בגרף ה-CDF של זמני ה-Interarrival (גרף 2).

Figure 6-1

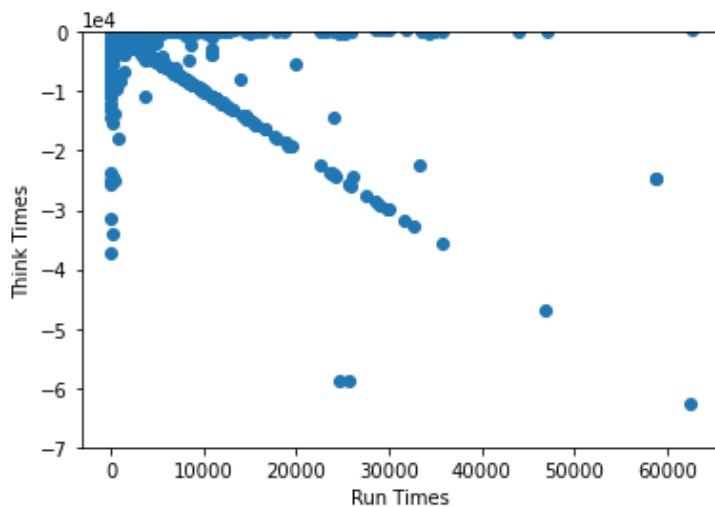
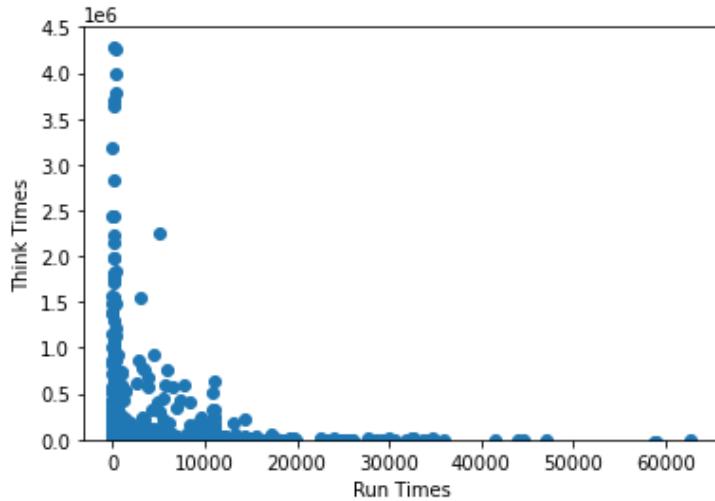


Figure 6-2



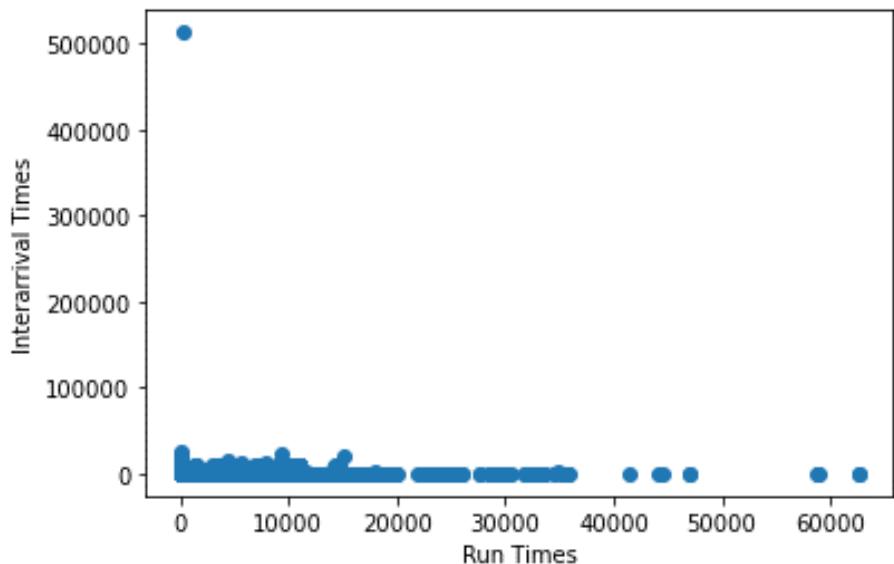


Figure 6-3

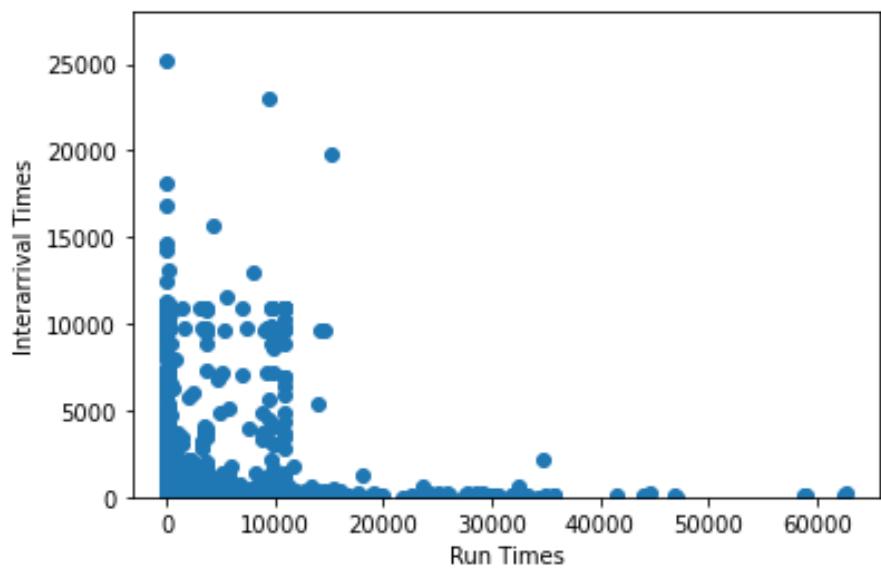


Figure 6-4

```
In [3]: runfile('C:/Users/elias/OneDrive/Documents/GitHub/SWF-Parser/main_parser.py', wdir='C:/Users/elias/OneDrive/Documents/GitHub/SWF-Parser')
Reloaded modules: RowClass
Correlation Coefficient between run times and think times:
0.0133
Correlation Coefficient between job sizes and run times:
0.1997
Correlation Coefficient between run times and interarrivals times:
0.0211

In [4]: |
```

Figure 6-5

MATLAB-Log File

מקדם הקורלציה בין Run Times למשתנה Think Times הוא: -0.06555923348325185

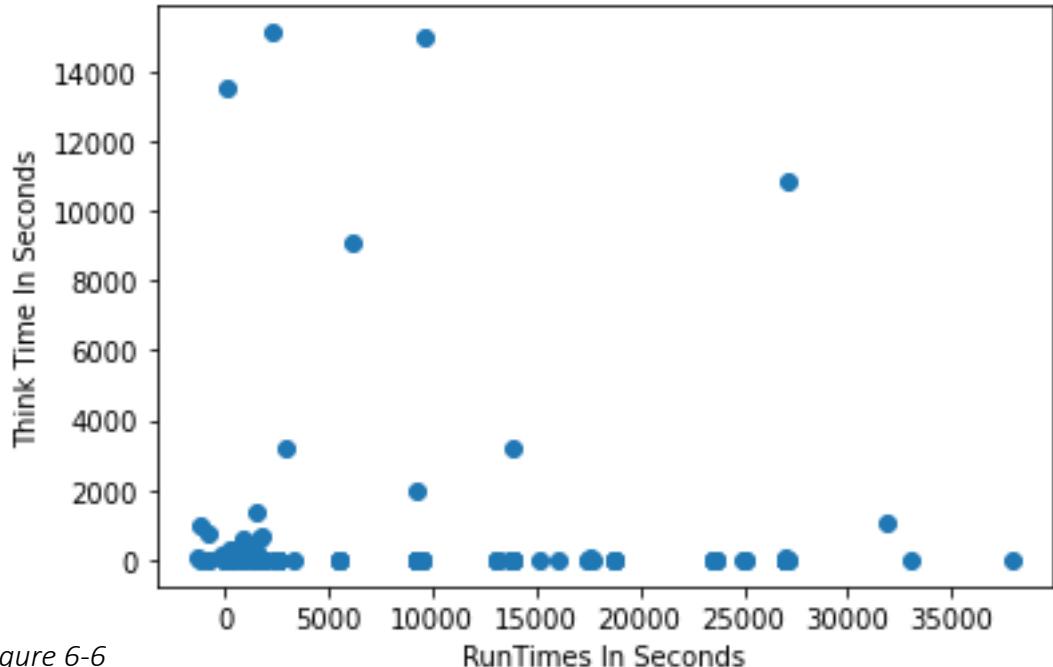


Figure 6-6

שלב 7 – יצירתיות distribution

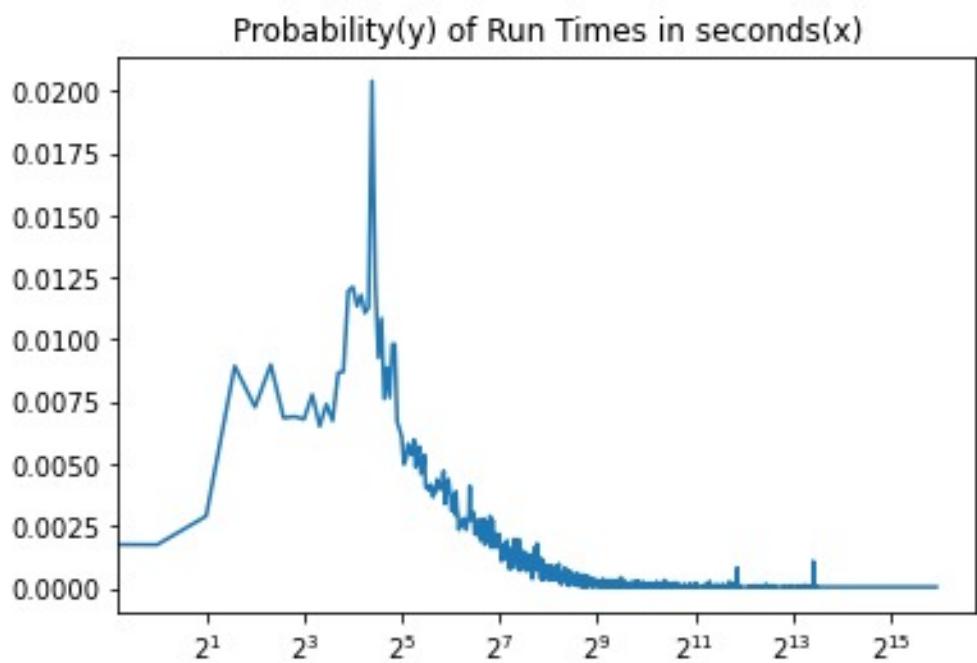


Figure 7-1

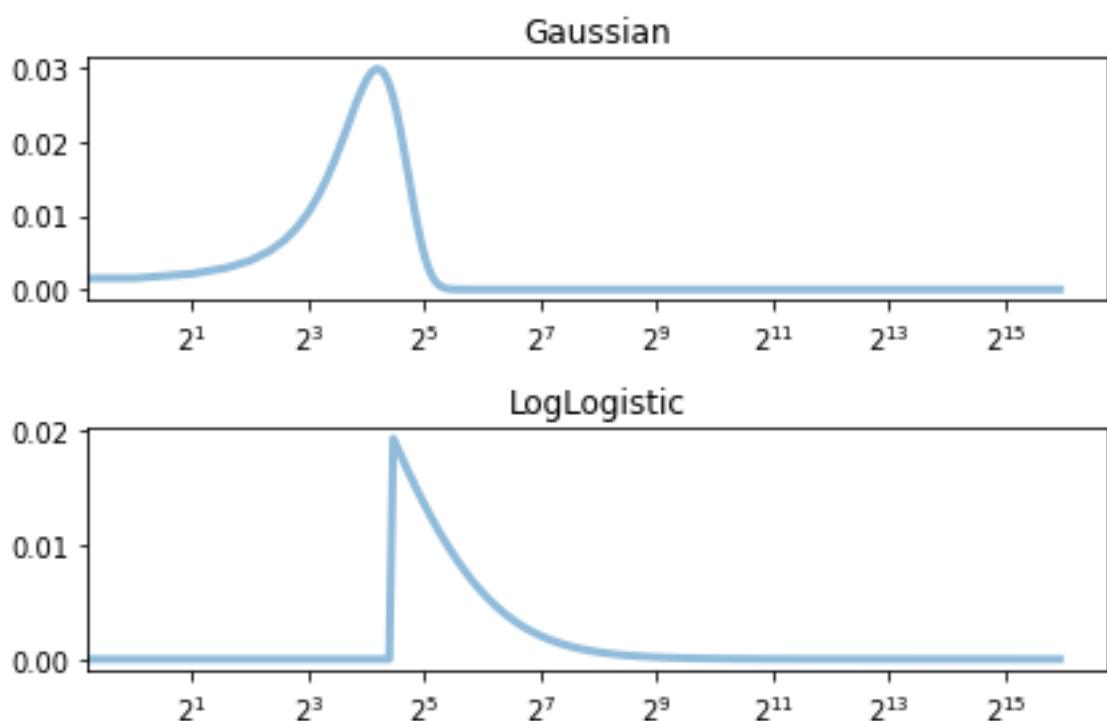


Figure 7-2

The two distributions

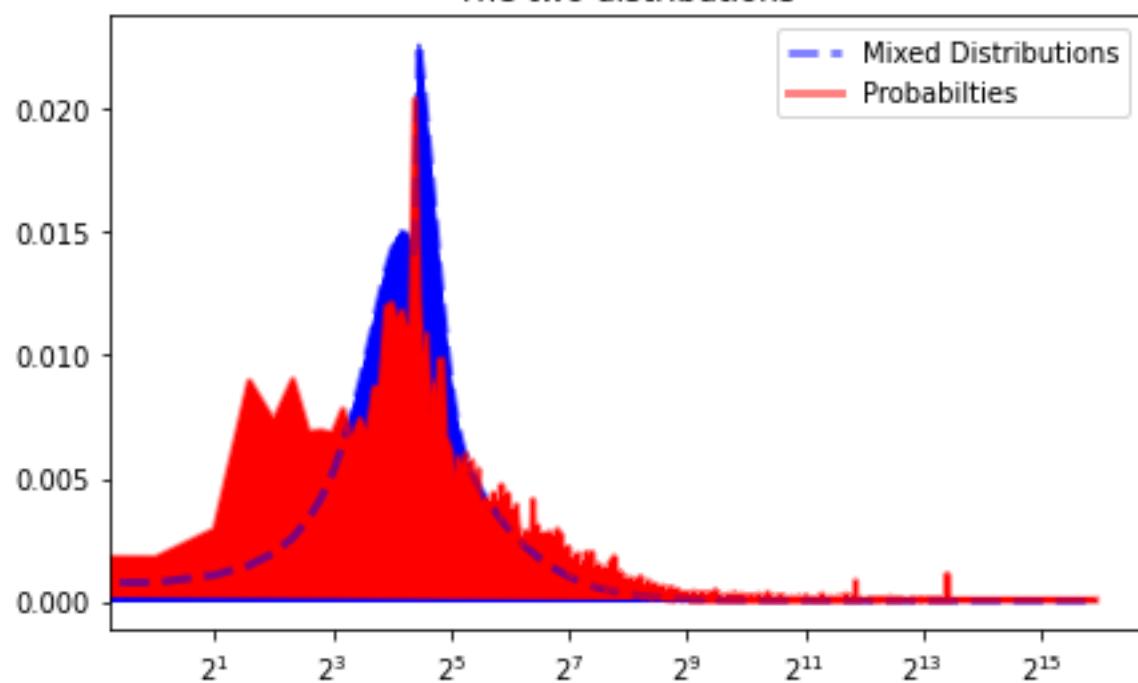


Figure 7-3

שלב 8 – יצירת מודל המדמה את המידע הראשון

ביצירת ה-Traces שלנו, השתמשנו בעיקרו User Resampling. עשינו זאת על ידי כך שחילקנו את ה-Users ל- 7 Clusters (השתמשנו ב-K-MEANS ובחרנו את המספר 7 משומש שיש לנו 69 Users, ו- 10% מהכמות הזאת היא בערך 7 Users). לאחר מכן, בחרנו את הקבוצות שקיבלו, בחרנו נציגים מכל קבוצה – על מנת לדאוג שהיה לנו לפחות שני users Short term users, והנותרים יהיו Long term users.

לאחר מכן, יצרנו קובץ קונפיגורציה בפורמט הבא:

```
"Residence           User1:<ResidenceTime1>           User2:<ResidenceTime2>           ....  
User69:<ResidenceTime69>"
```

//For Example: Residence User1:4892 User2:12099 User3:123 User69:3

```
"Activity           Week1:<NumberOfNewUsers1>           Week2:<NumberOfNewUsers2>           ....  
Week13:<NumberOfNewUsers13>"
```

"Random_Seed 1"

<Jobs of the first user chosen>

.

.

.

.

.

<Jobs of the last user chosen>

//The number of users chosen must be 2 long term users (i.e are active the whole time) and
6-7 short-term users (you could add more if you want)

לאחר בחירת ה-Users, יצרנו שלושה קבצי Config, אשר כל אחד מהם מדמה עומס שונה במערכת; 80%, 100% ו- 120%.

הוספנו את קבצי הקונפיגורציה לתיקייה שלנו, ובתוכנית Python, בנינו את שיטת ה-User Resampling, אשר בהתבסס עליה הפקנו את הגרפים שנראה בשלב 9.

שלב 9- אימות

1. השוואת העומסים

באיורים הבאים (9-1, 9-2, 9-3) מתחארת השוואת של ה-Traces עם העומסים שייצרנו אל מול ה-Trace עם העומס המקורי.

יהי לנו קושי לקבל עומסים של 80% ושל 120% ב-Traces שהקפנו, מכיוון והדגימה של משתמשים הייתה מתוך כמות קטנה יחסית לאורך כל ה-Trace, ולכן Submit time היו צפופים מיד, מה שגרם לעומס רב על המערכת.

Consumption Graph Of Each One Of The Realistic Traces With 80% Load And The Original Trace

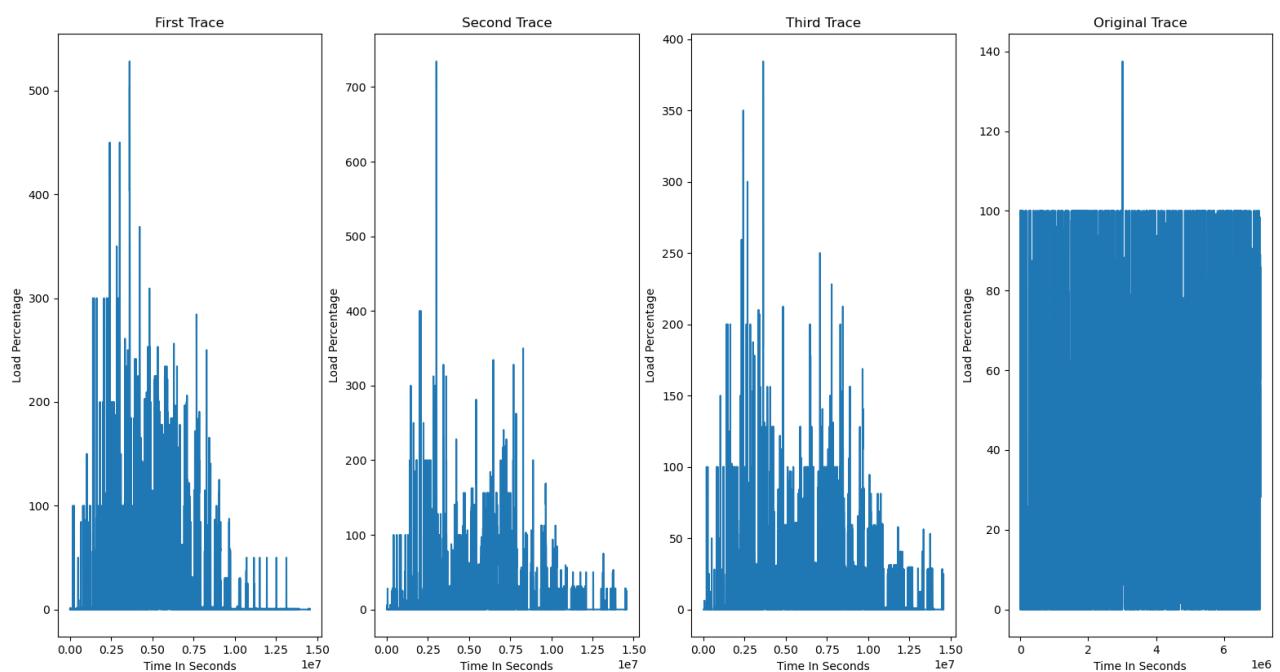


Figure 9-1

Consumption Graph Of Each One Of The Realistic Traces With 100% Load And The Original Trace

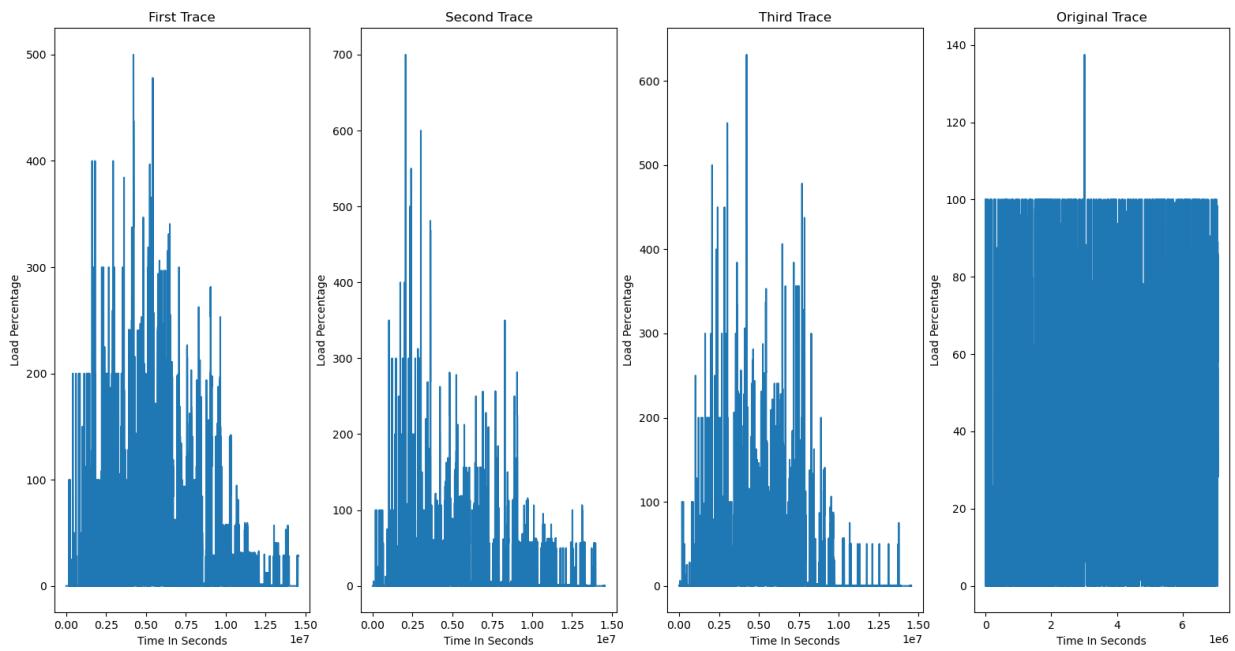


Figure 9-2

Consumption Graph Of Each One Of The Realistic Traces With 120% Load And The Original Trace

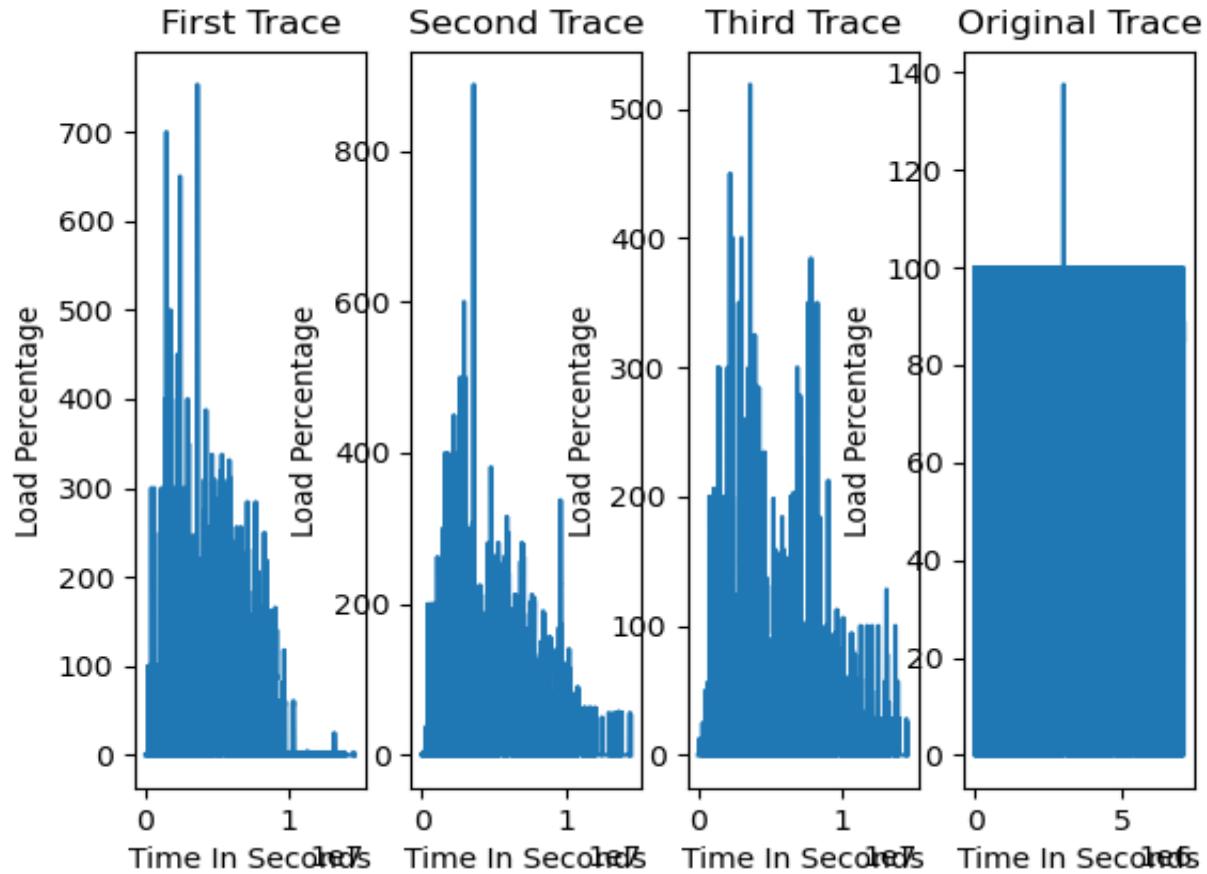


Figure 9-3

2. התפלגויות .1 Interarrival time

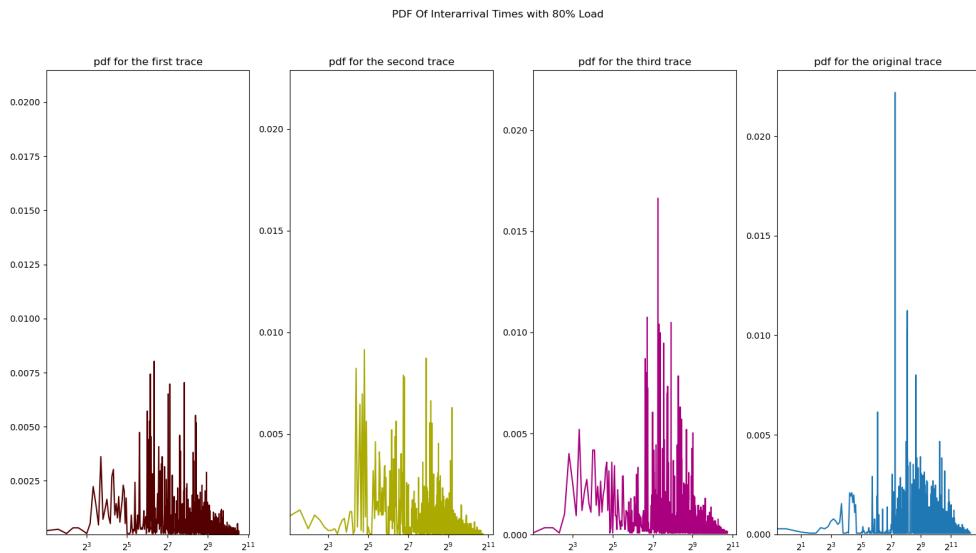


Figure 9-4

נראה כי באיר 4-9, ניתן לראות ב-80% עומס שיש לנו הטענות די דומה בהשוואה לגרף הראשון מצד ימין. ניתן לראות את אזור הדשדוש בתחילת ובסוף של הגרף. כਮובן שההבדל בין ה-Traces Random – שבאמצעותה הפקנו שלושה Traces. ניתן להבחין את הדמיון בשיא בין שני הגרפים האחרונים בטוויח של הזמןים של $2^{11} - 2^5$. ניתן לראות דמיון די ברור בין הגרף הראשון לגרף השלישי בטוויח של הזמןים של $2^5 - 2^3$.

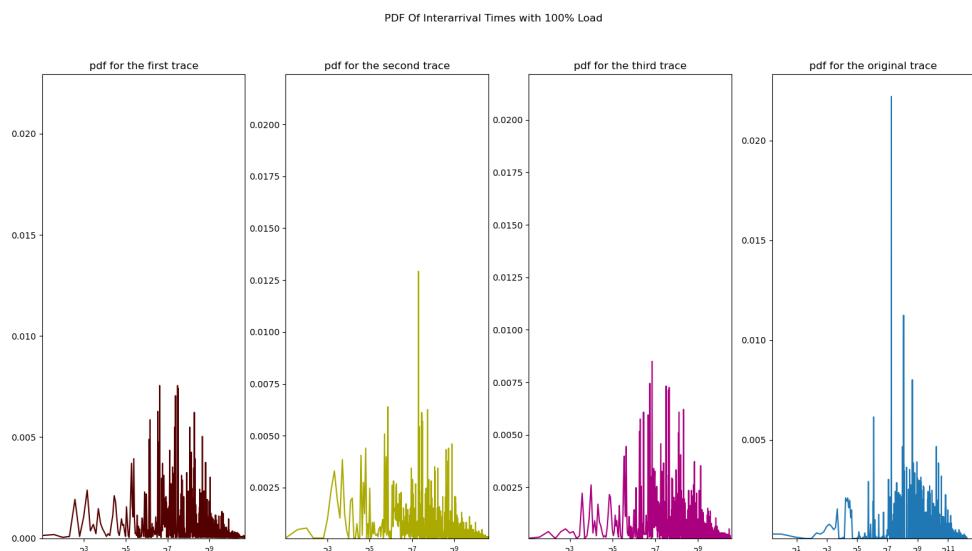


Figure 9-5

באיור 5-9, ניתן לראות ב-100% עומס שיש לנו הטענות די דומה בהשוואה לגרף הראשון מצד ימין. רואים את אזור הדשדוש בתחילת ובסוף של הגרף. את הדמיון-peaks ניתן לראות בבירור בין הגרף השני והאחרון, בטוויח הזמןים של $2^{11} - 2^5$. בנוסף, ניתן לראות דמיון די ברור בין הגרף הראשון לגרף השלישי והאחרון בטוויח הזמןים של $2^5 - 2^3$.

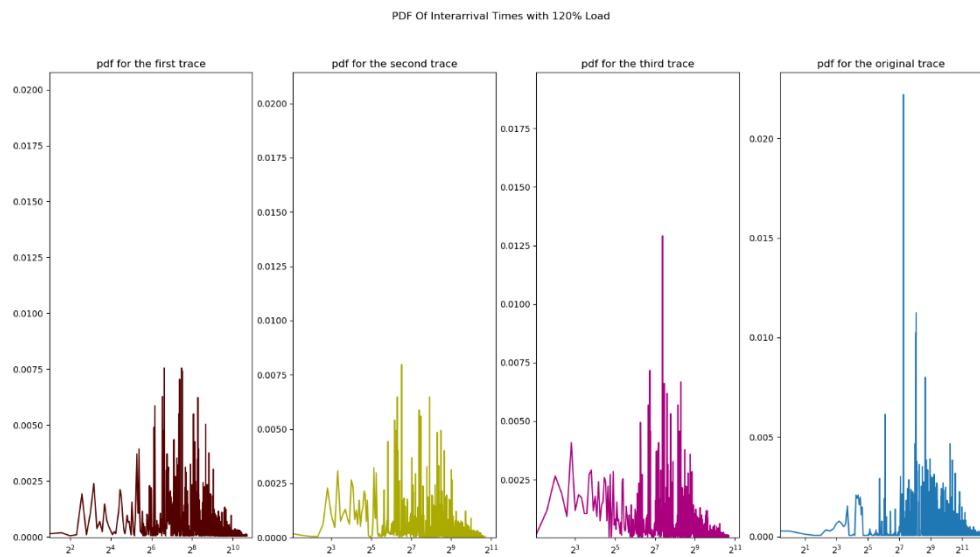


Figure 9-6

את איור 6-9, ניתן לתאר לפי אוטם הסברים שהוצגו מעלה פחות או יותר.

:Runtimes .2

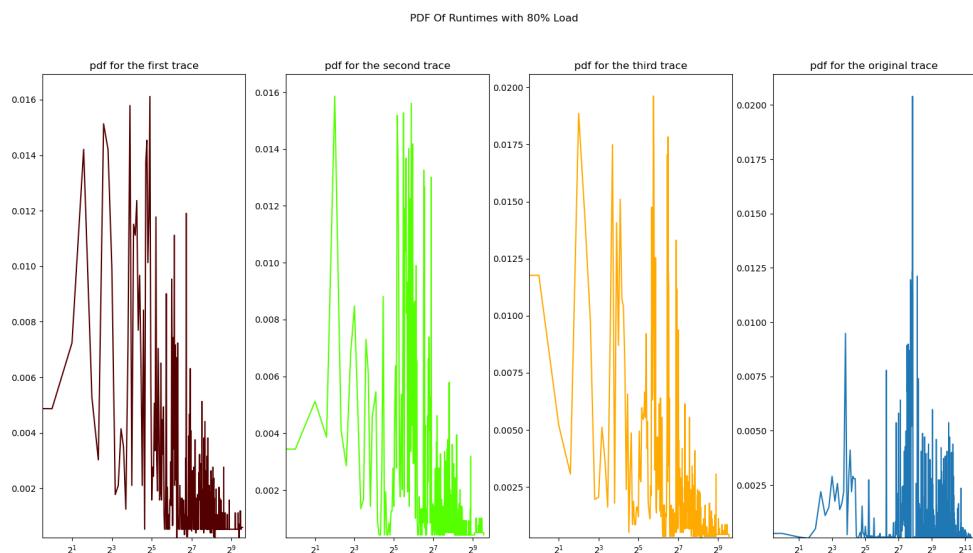


Figure 9-7

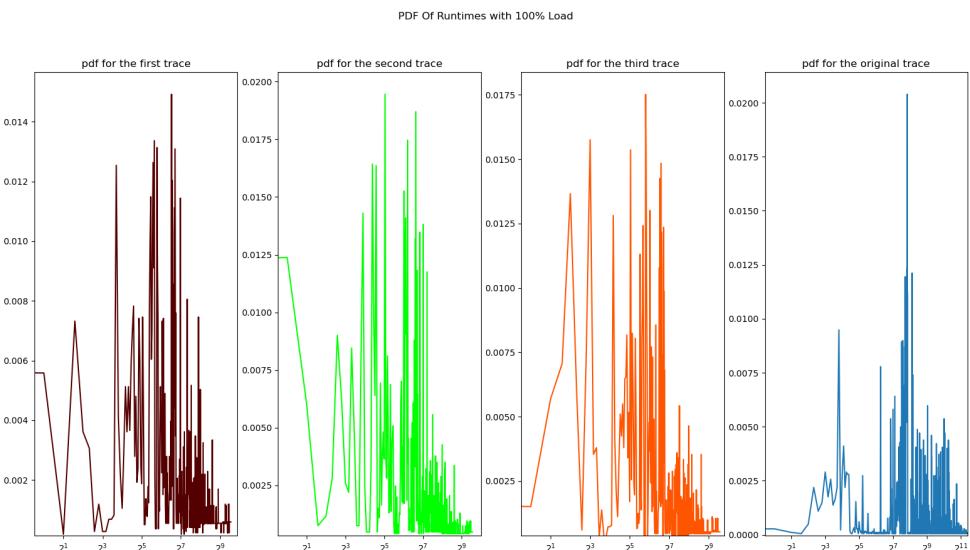


Figure 9-8

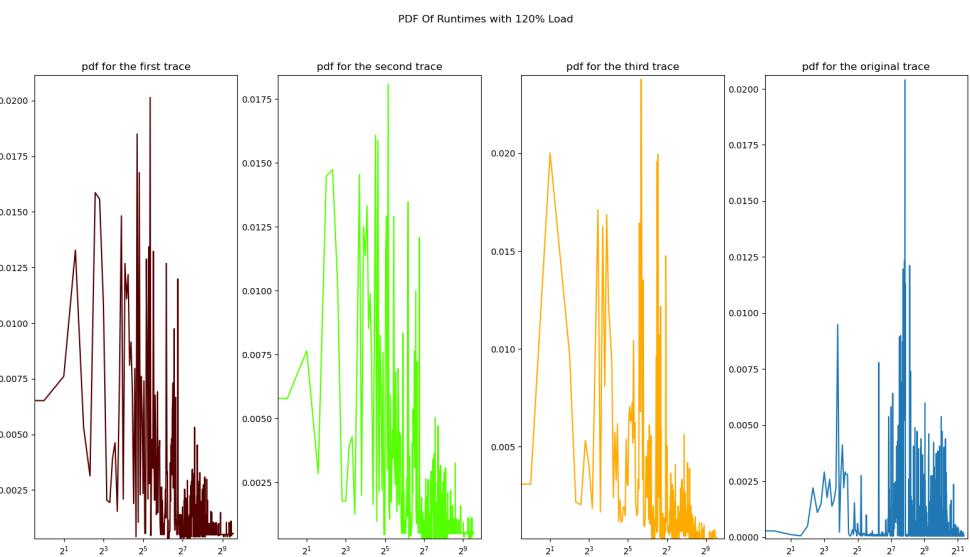


Figure 9-9

בהתכלות כללית על האיורים 7-9 עד 9-9, ניתן לראות שישנו דפוס התפלגות די דומה בין שלושת ה-Traces לעומת ה-Trace המקורי. ניתן לבדוק שישנו הבדל בתחום הזמן של $2^3 - 2^1$ מasinיבת הפשוות שיש יוזרים שלא נמצאים בקובץ הקונפיגורציה שלנו.

:User Distribution .3

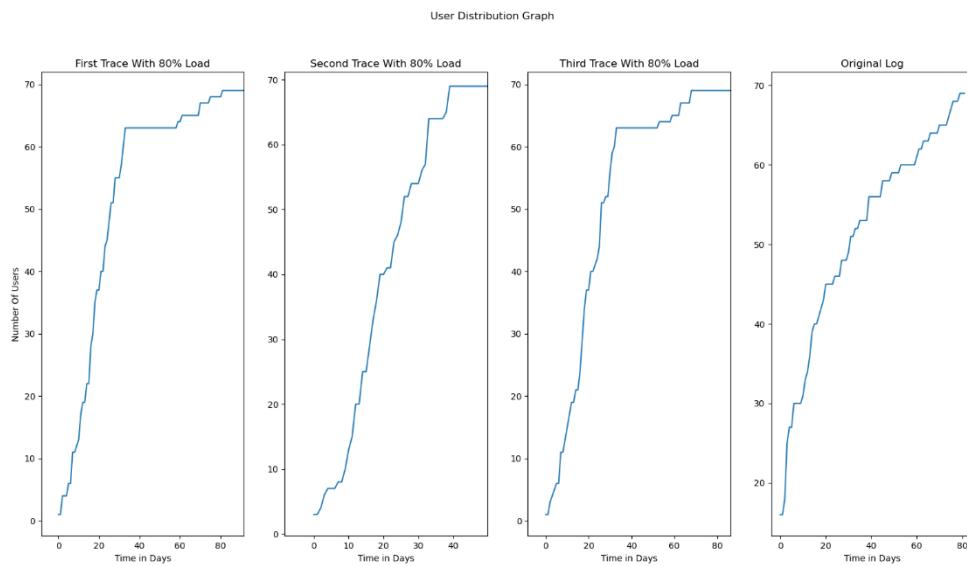


Figure 9-10

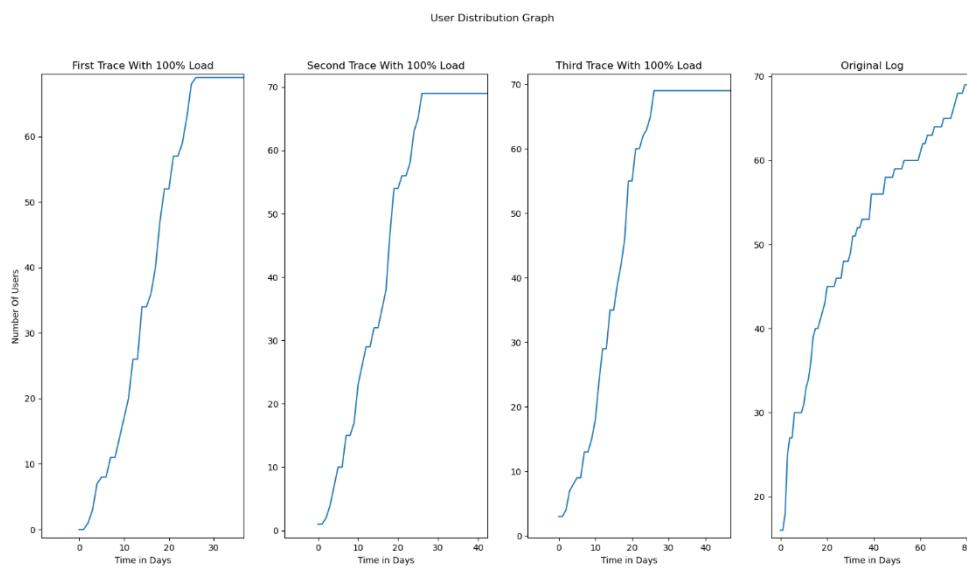


Figure 9-11

בשלושת האיורים (10-9 ו-12-9) ניתן לראות גרפים, ועליה בהפלגות של ה-Users, באופן דומה ל-Trace המקורי שלנו. ההבדל בציר ה-X הוא בגלל הסיבה שחתכנו את הגראפים, מכיוון שהקו ממשיך באופן יציב עד סוף הימים. ציר ה-Y מייצג את מספר ה-Users שראינו עד לרגע זה.

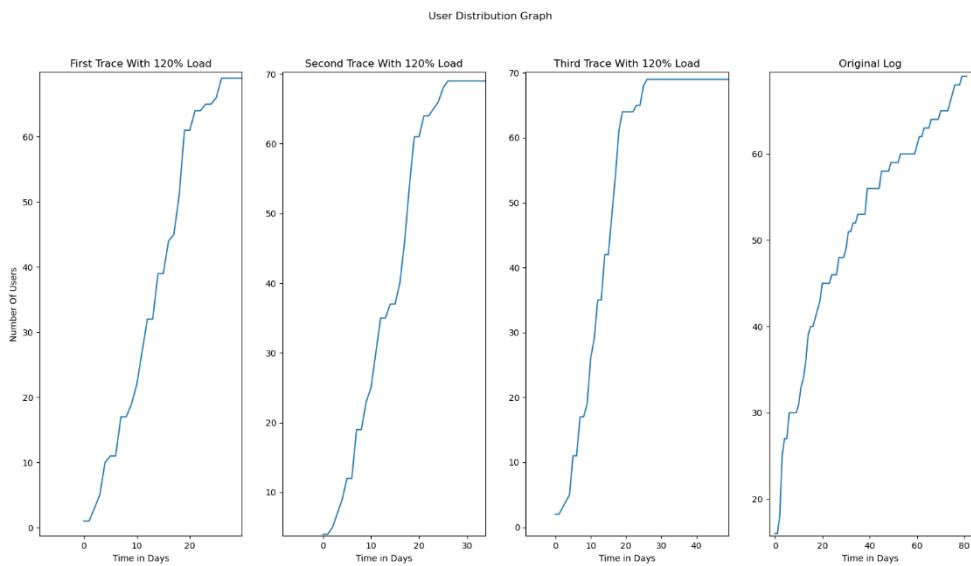


Figure 9-12

3. קורלציה

כדי לבדוק את הקורלציה, בחנו את ה-Job sizes ו-Run times לשולות העומסים ושלושת ה-Traces, ובנוסף הראינו את ה-Think times ואת זהותם. הגрафים מותווים בהמשך:

:Job Sizes and Run Times .1

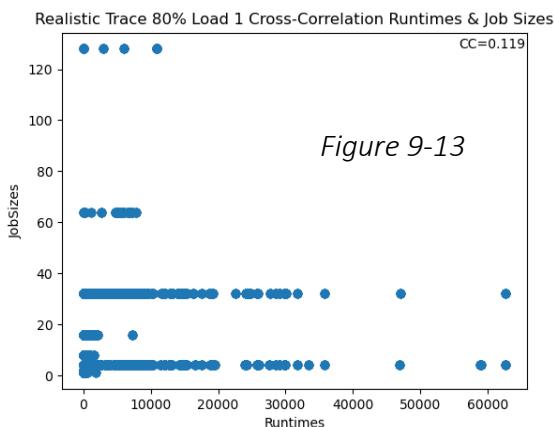


Figure 9-13

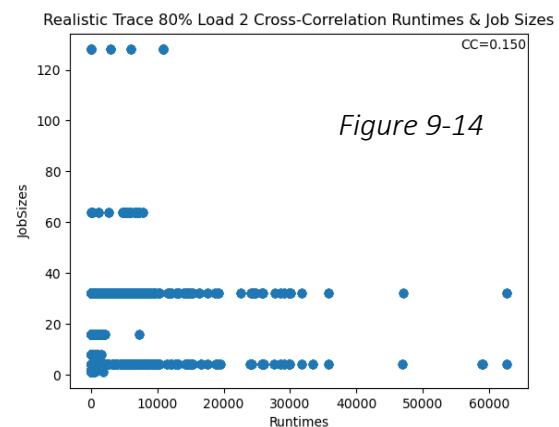
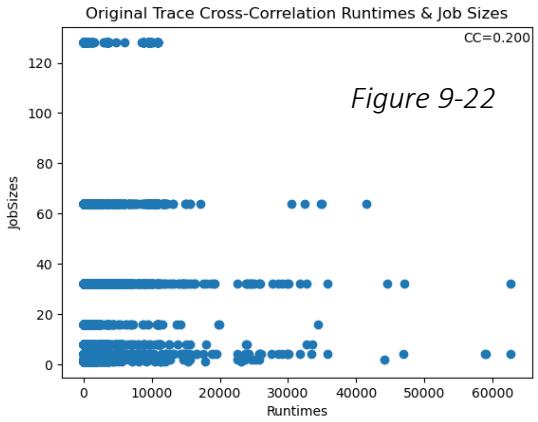
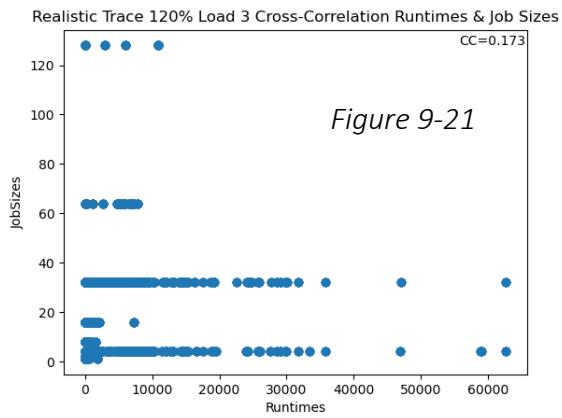
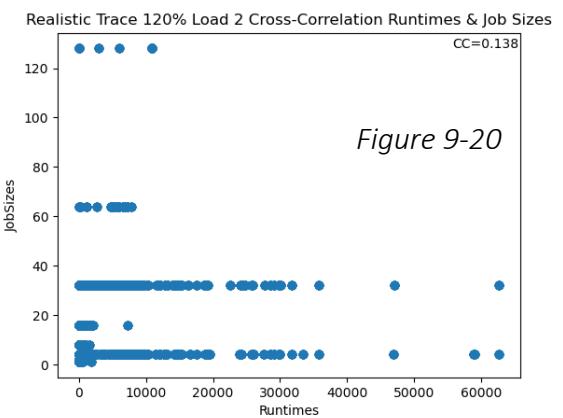
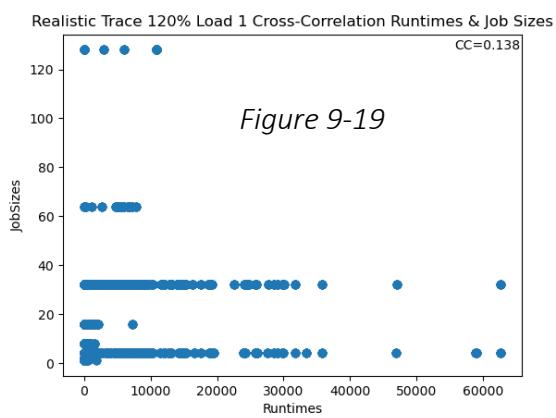
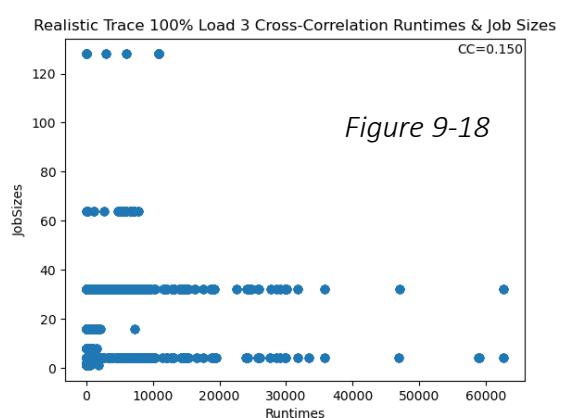
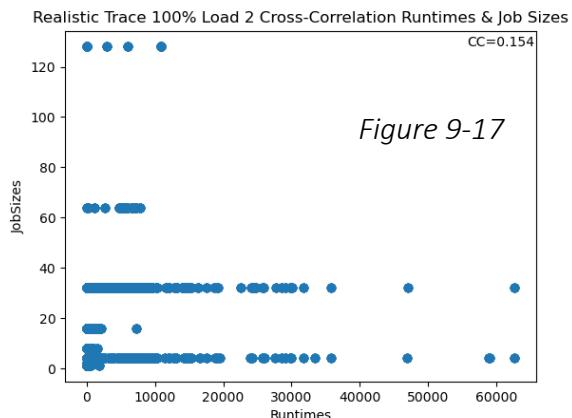
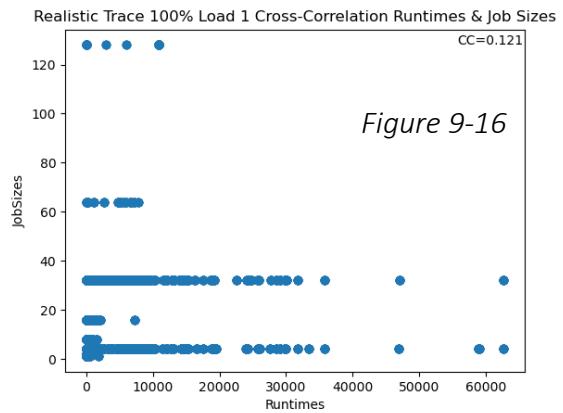
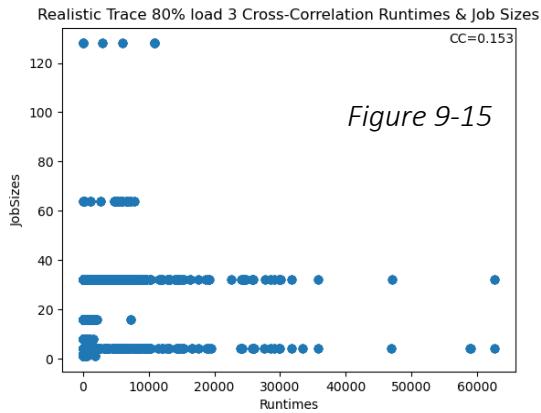
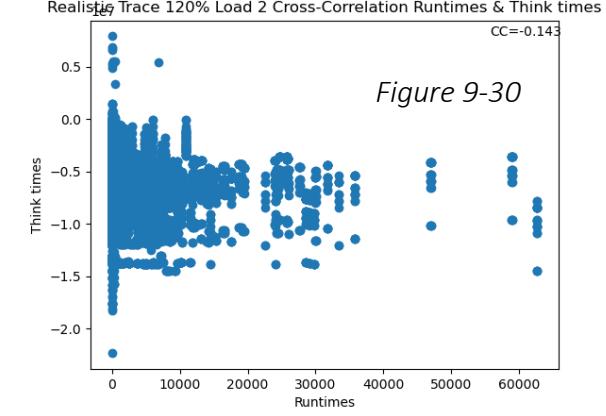
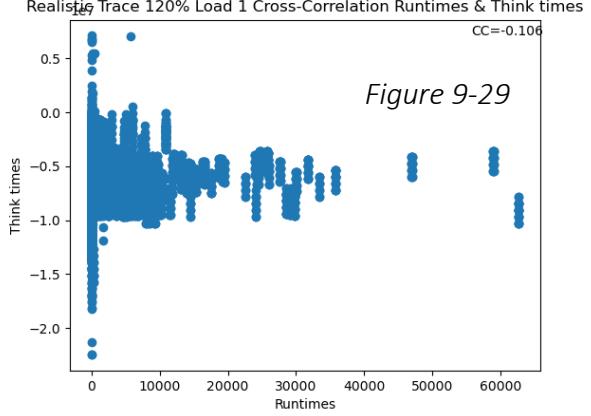
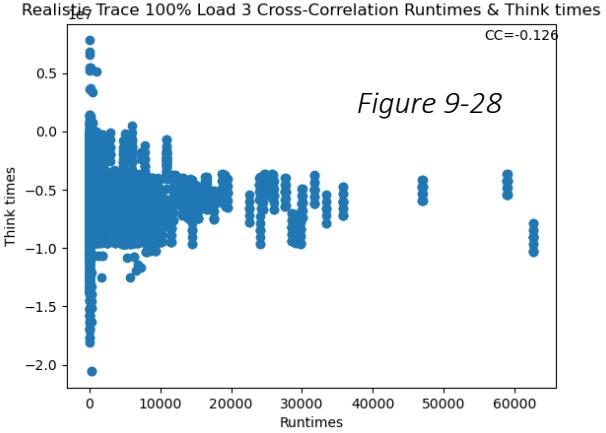
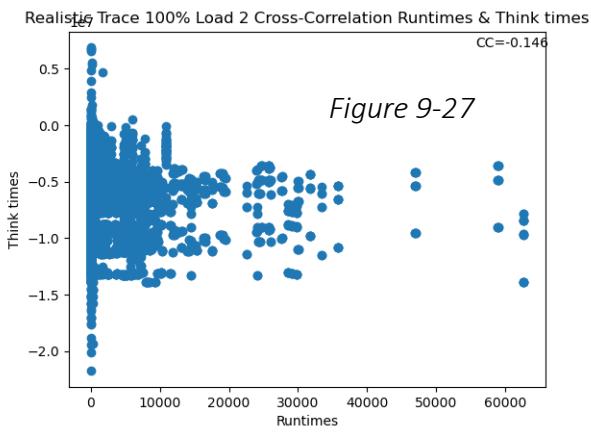
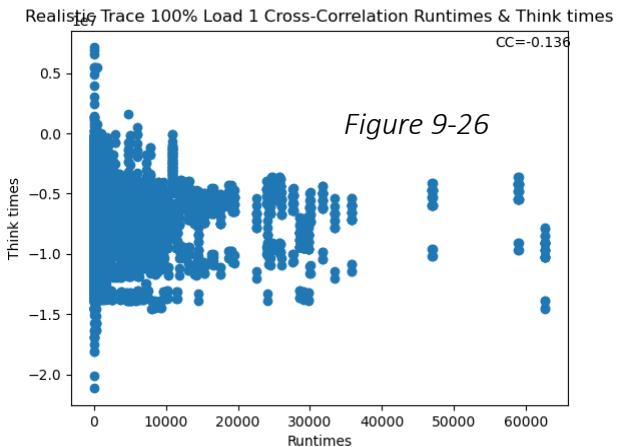
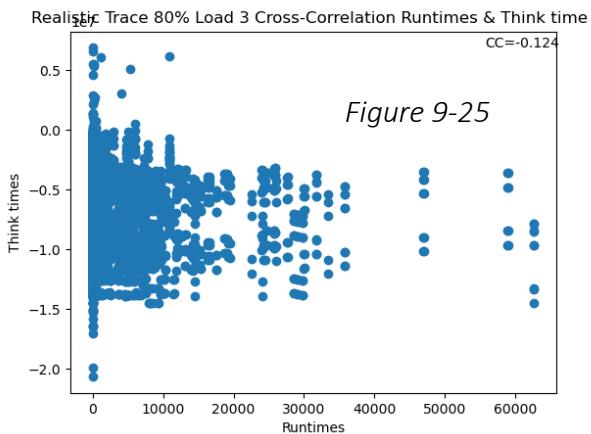
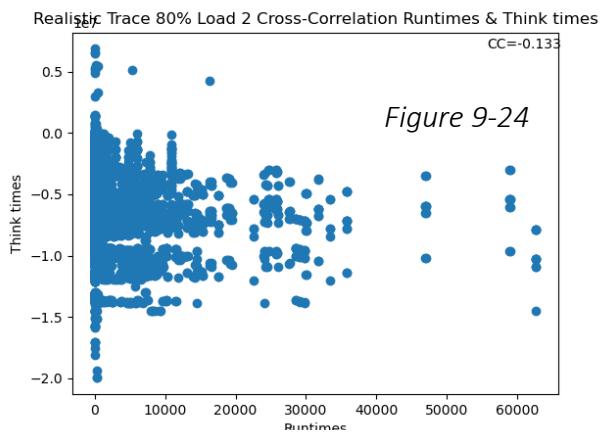
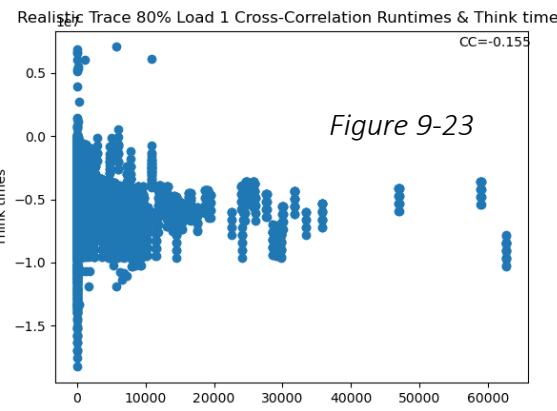


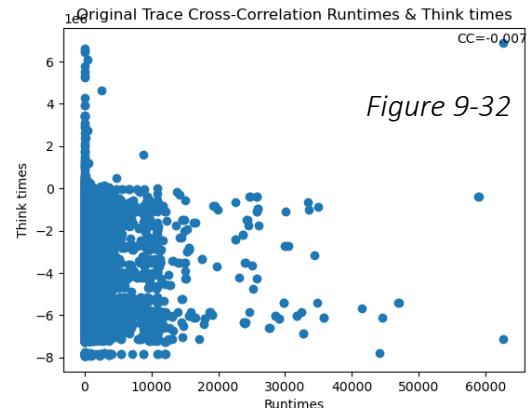
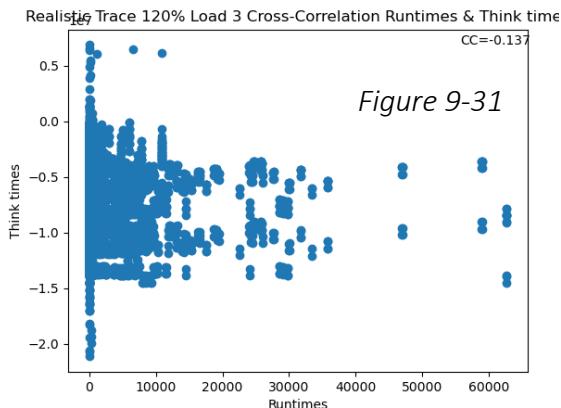
Figure 9-14

ניתן לראות כי מקדם הקורלציה אכן מתפלג בין 0.1 ו- 0.2 בכל הגרפים. כמו כן, ניתן לראות כי לא הייתה לנו קורלציה די ברורה, שכן אין זה המצב בשלושת-Traces שהפקנו ובכל שלושת העומסים.



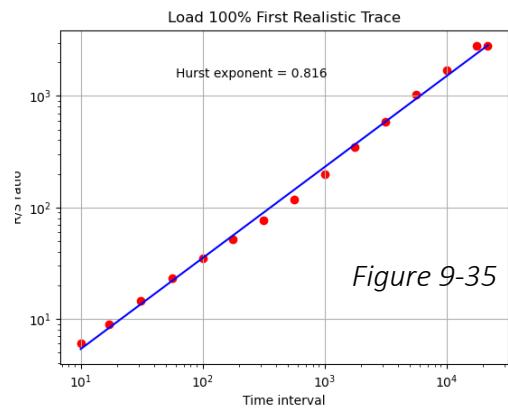
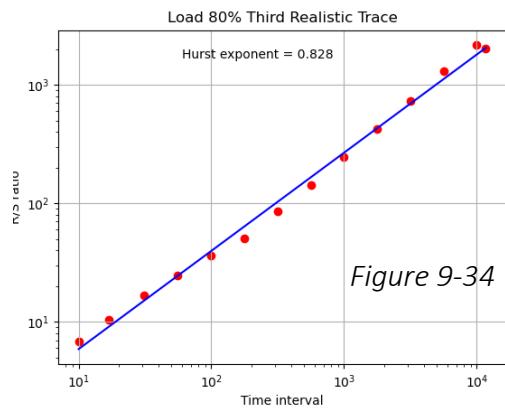
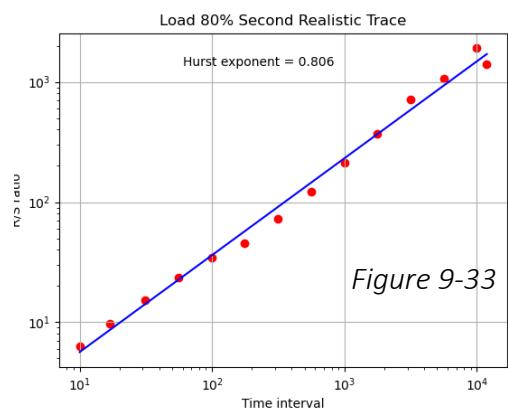
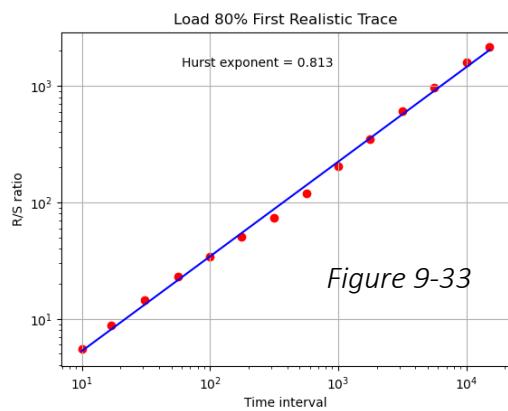
Think Times and Run Times .2

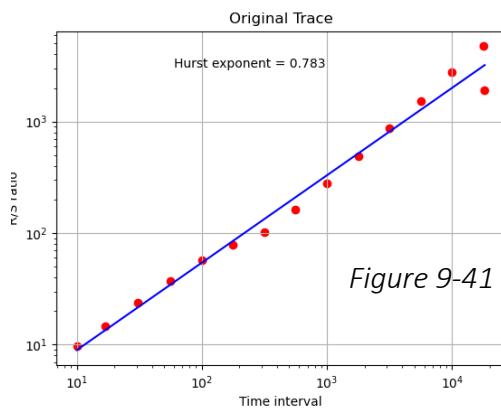
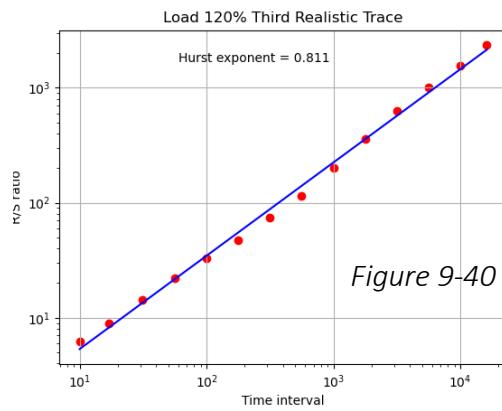
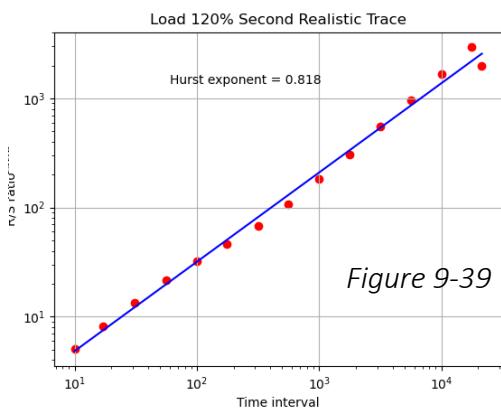
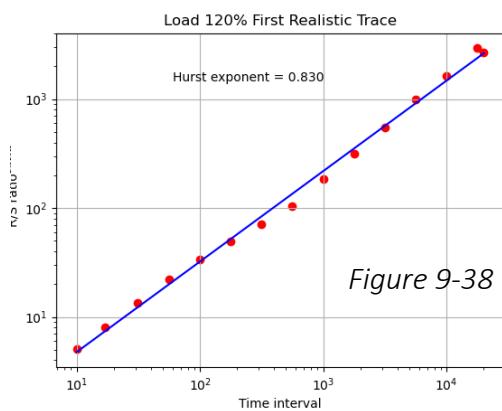
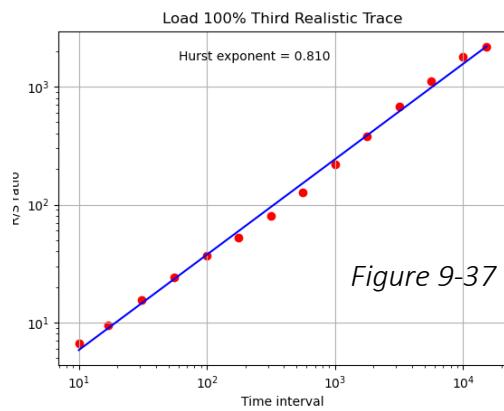
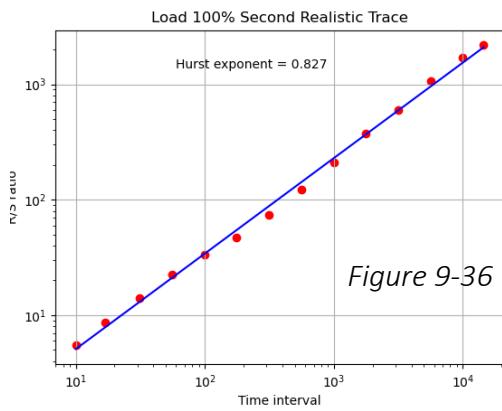




באיורים 9-23 ו-9-32, ניתן לראות כי מקדמי הקורלציה התרפלו בין 0.007- 0.15- ל- 0.137- בכל הגрафים, גם לפי ההגדרה אין לנו קורלציה, ותבוננה זו נשמרה בשלושת ה-Traces שלנו.

Self-Similarity .4

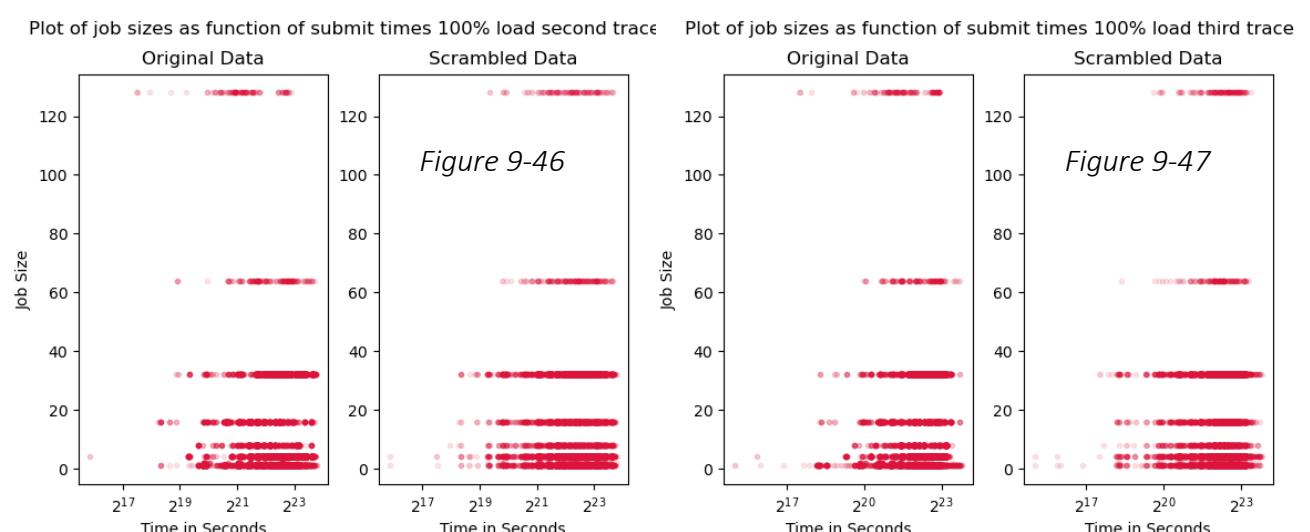
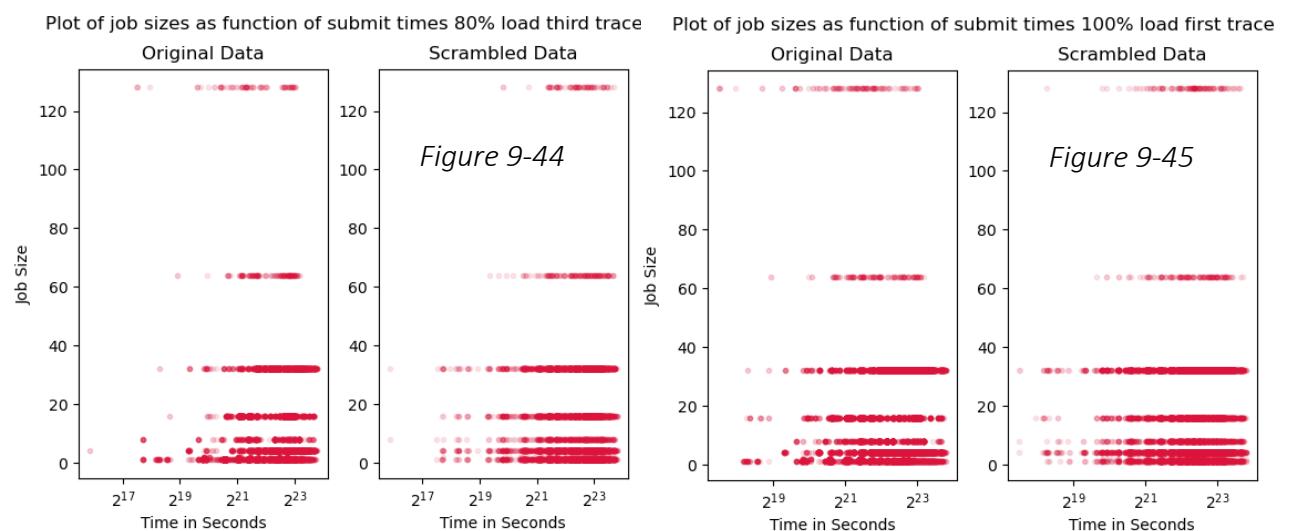
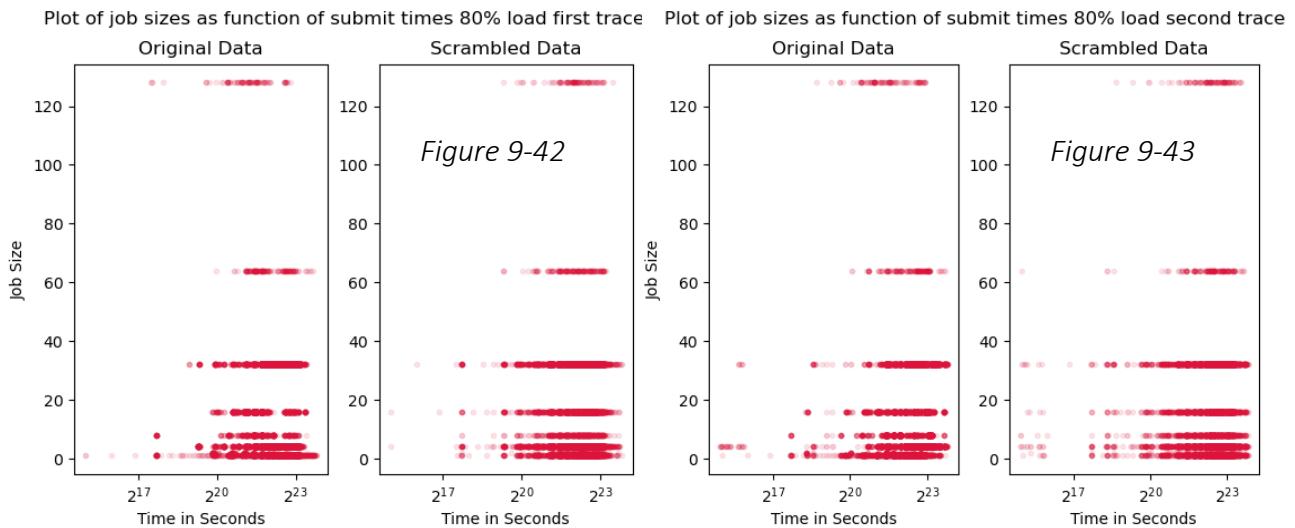


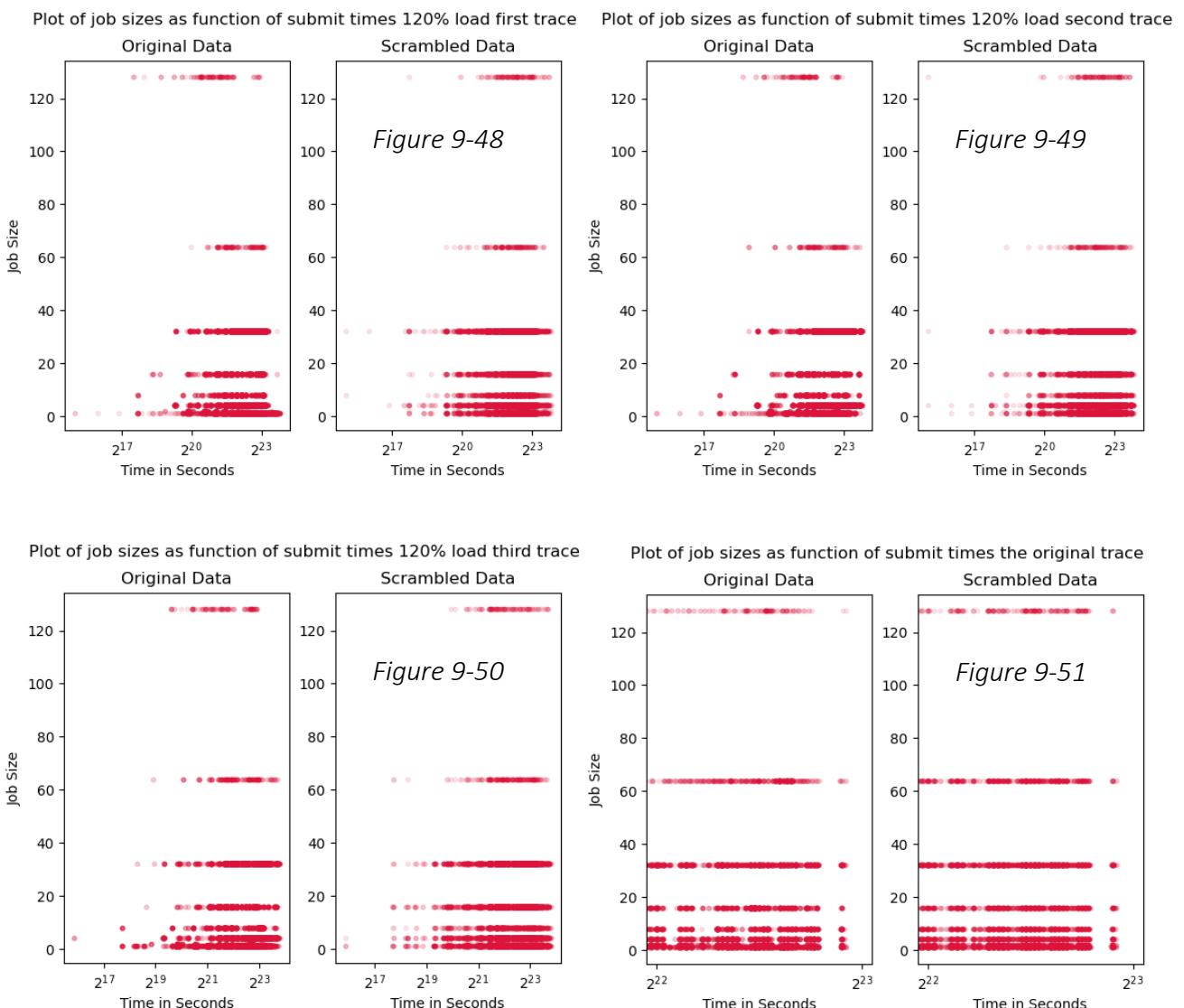


ניתן לראות כי בשלושת traces, בכל רמות העומס, שמרנו על גרף די דומה, עם צווח הדוק של ה-Hurst Exponent לפי ההגדרה בפרק 7 בספר.

Locality of Sampling .5

בגרפים מוצגיםisp Job Sizes כפונקציה של Submit Times





אם נסתכל טווח הזמן $2^{22} - 2^{23}$ – Locality of Sampling נשמר בכמה וכמה נקודות למרוחת הירובות שהציגו, ובנוסף, גם אם נסתכל על כל גרסה של כל Trace בנפרד, במיוחד באזור אשר בו ה-Job Sizes שוים ל- 20,40,60, 120, ניתן לראות כי קיימים דינומות בין כל הגרסאות, עם נקודות בהירות ובהות, מה שמעיד לנו על ה-Locality of Sampling כפי שהוצג בספר².

² Dror G. Feitelson, “Workload Modeling: Computer Systems Performance Evaluation”, p. 247

Submission Rate .6

בסעיף זה, מתוארכות השוואות בין ה-Users, בהתאם לעומס שהמערכת נמצאת בו. גرافים 60-9 עד 52-9 מתארים לנו בצורה גרפית מספר תתי גרפים (PDF, CDF, ECDF, Submissions) – PDF, CDF, ECDF, Submissions (Trace, Traces, Submission rates) אשר יצרנו את ה-Trace, נשים לב לכךון שלא הוכנס מנגנון מפורש לטיפול ב-Submission rates כאשר יצרנו את ה-Trace, נשים לב שקיבלנו תוצאות זהות ב-Traces שלמו – כפי שמתואר בגרפים בהמשך.

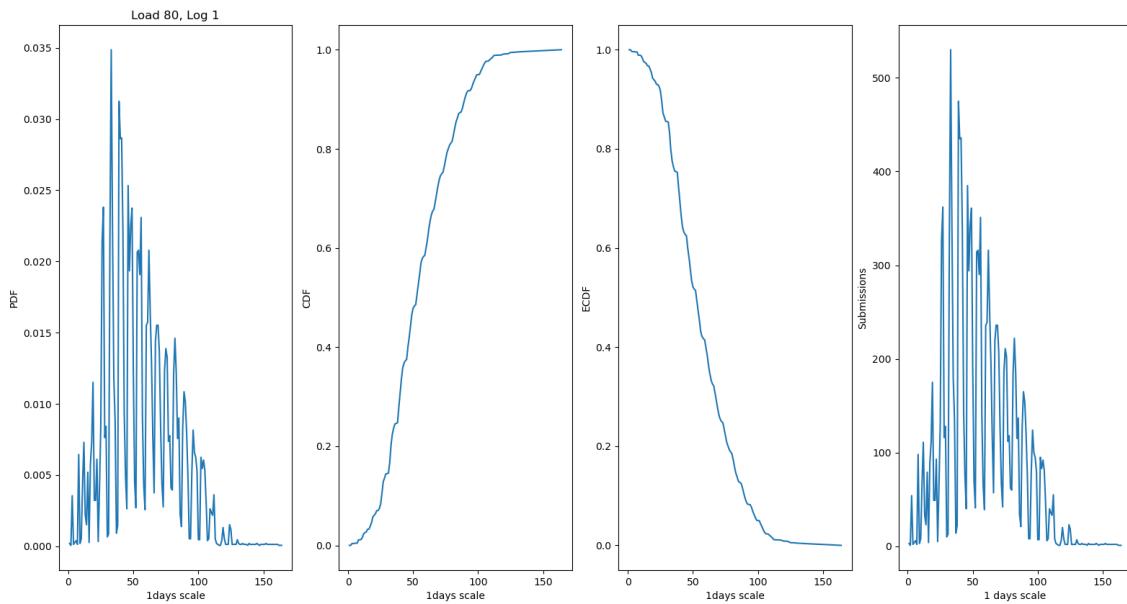


Figure 9-52: Load 80%, Trace 1

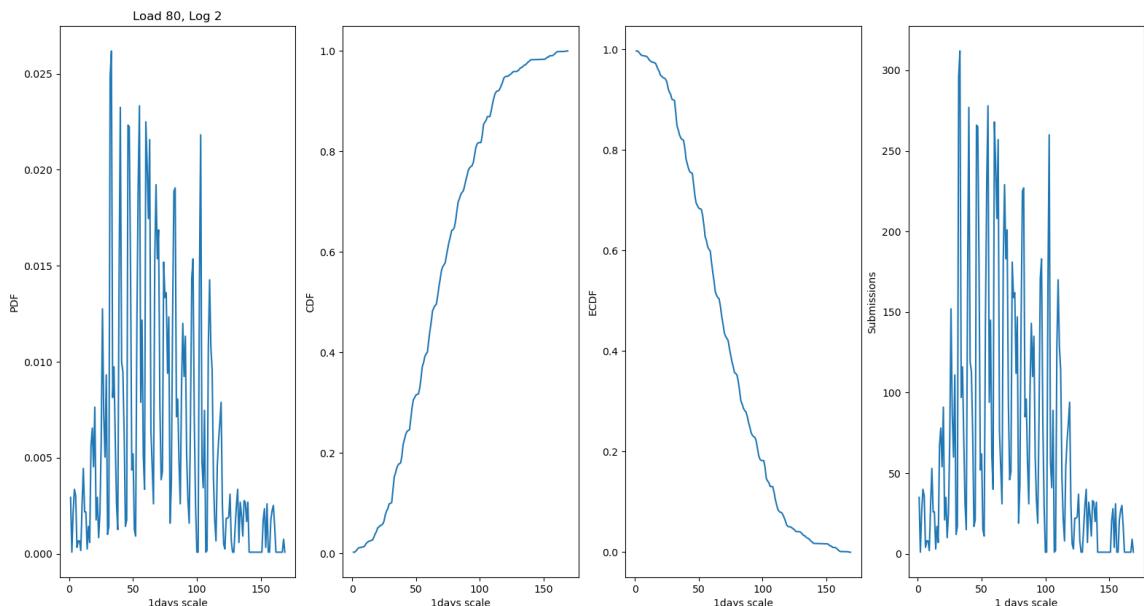


Figure 9-53: Load 80%, Trace 2

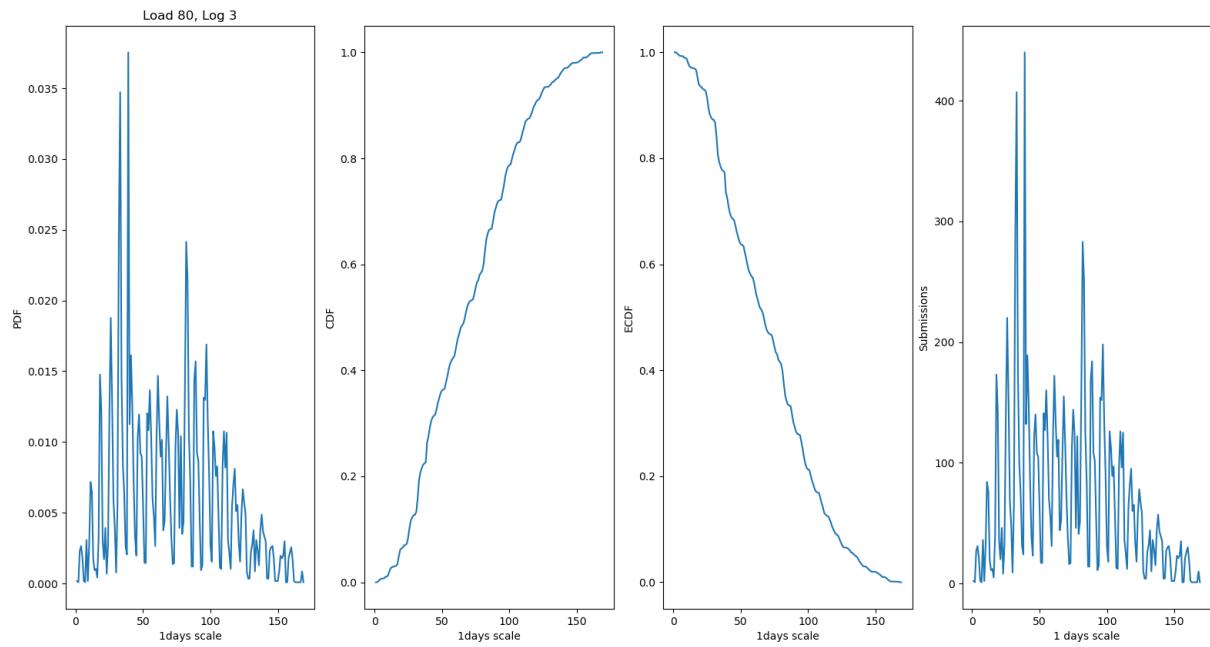


Figure 9-54: Load 80%, Trace 3

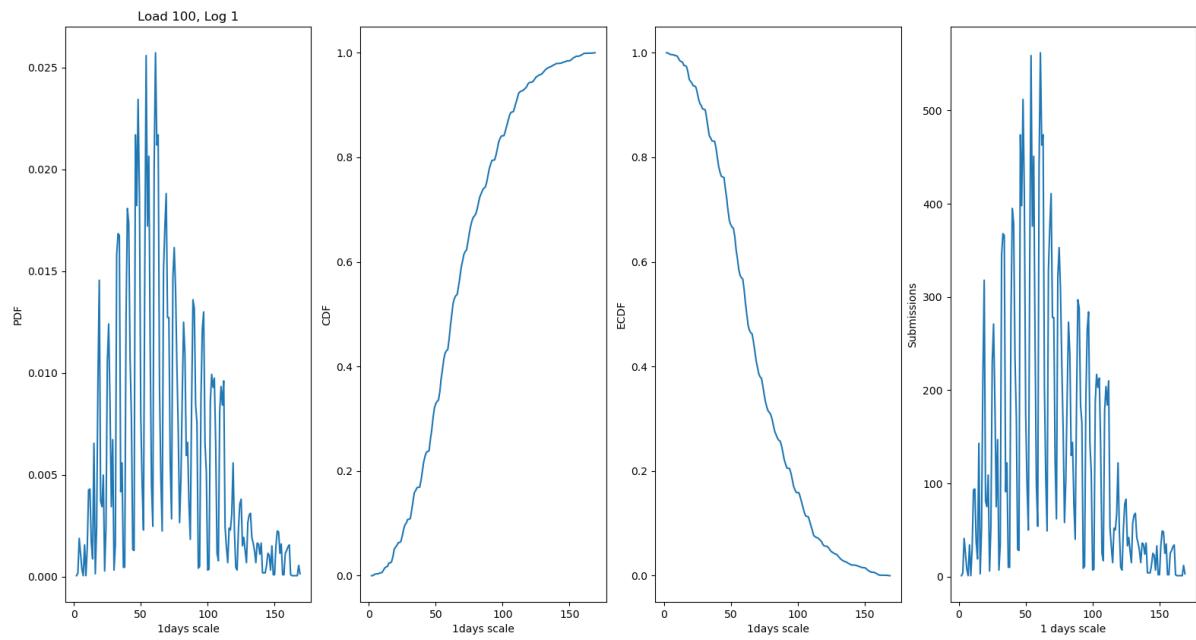


Figure 9-55: Load 100%, Trace 1

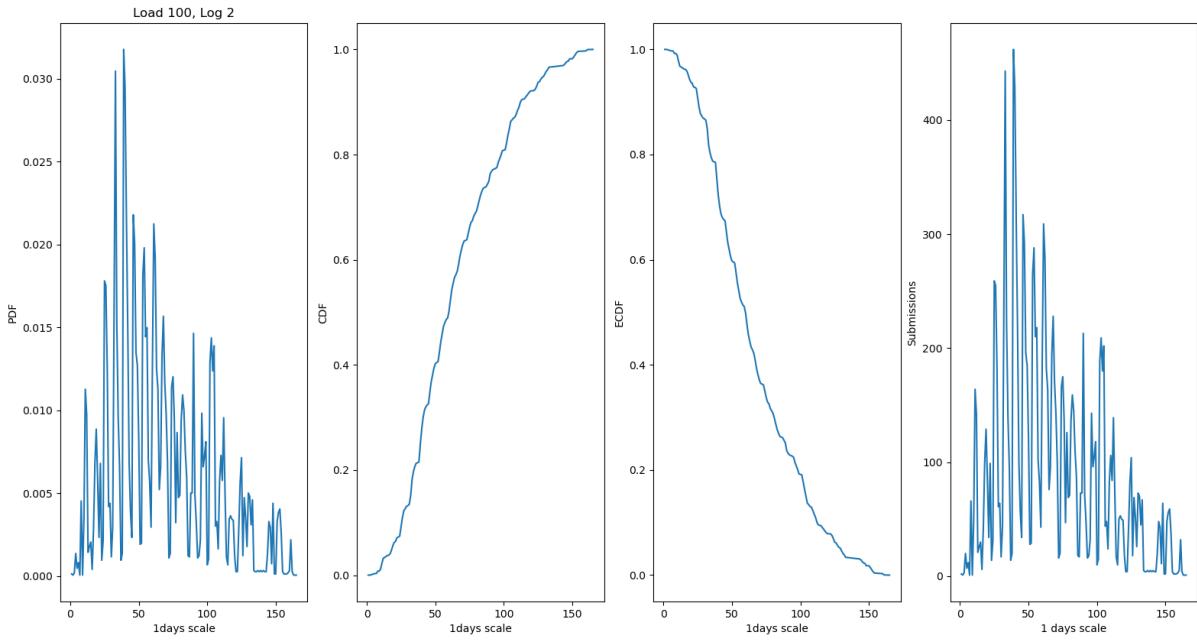


Figure 9-56: Load 100%, Trace 2

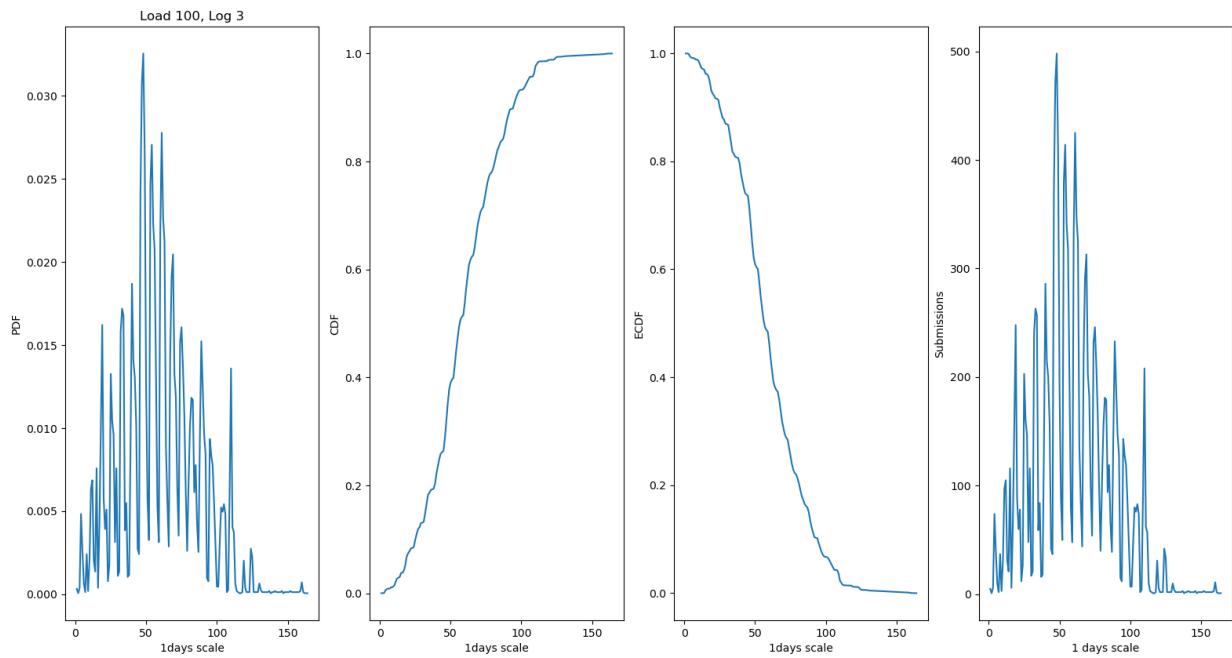


Figure 9-57: Load 100%, Trace 3

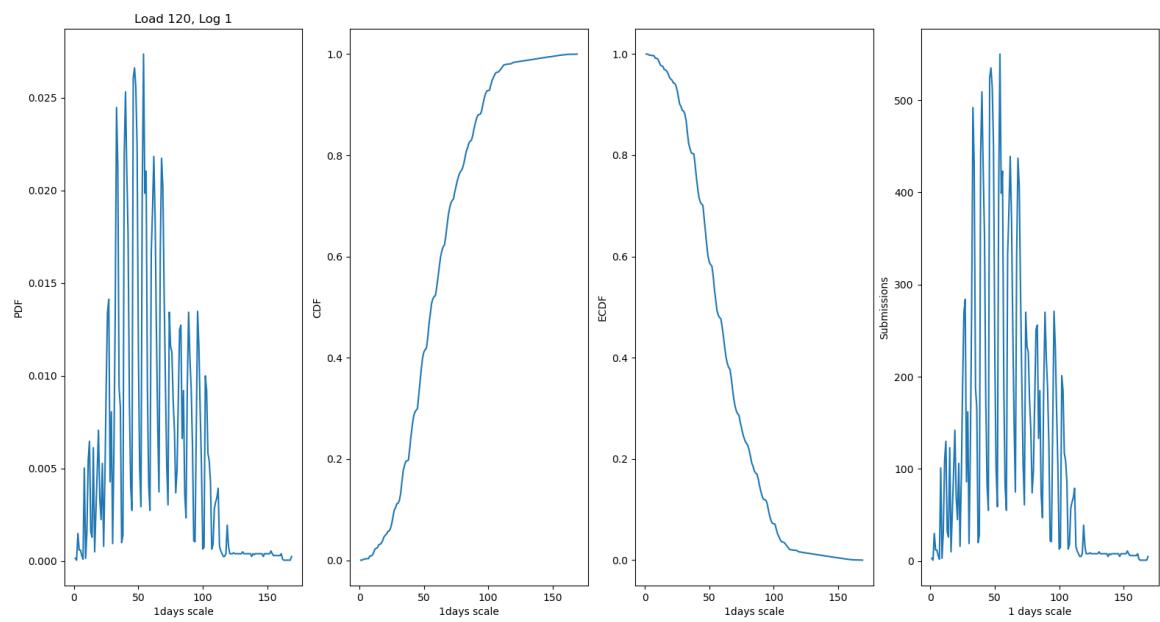


Figure 9-58: Load 120%, Trace 1

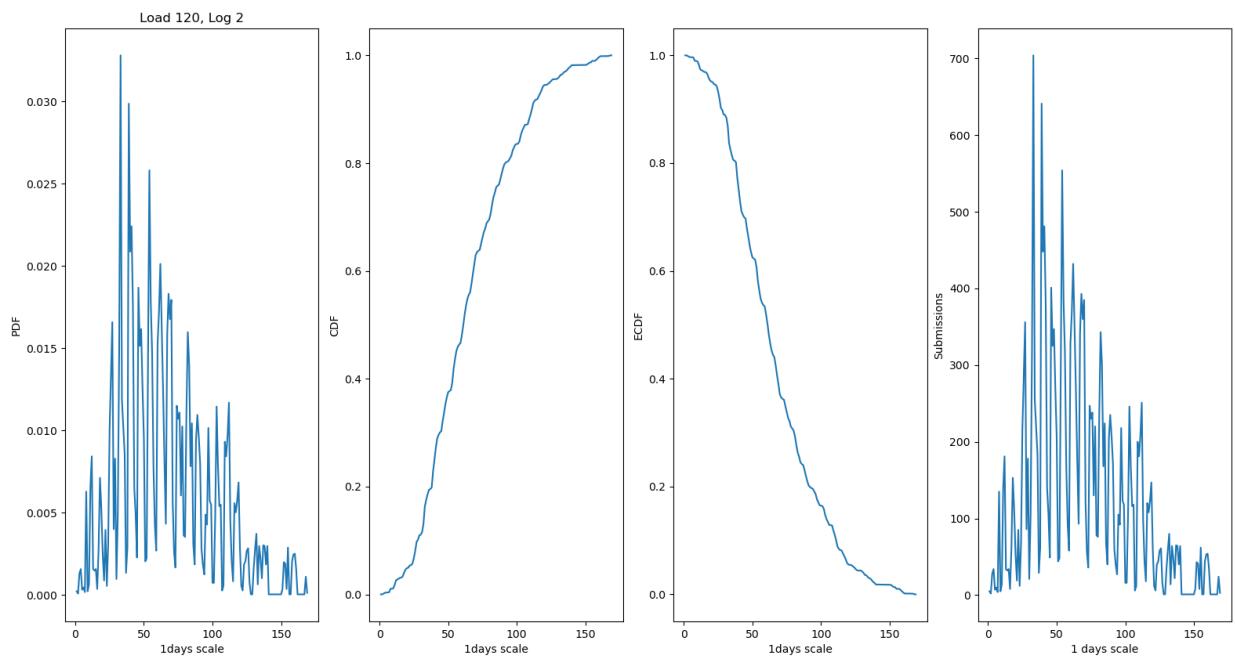


Figure 9-59: Load 120%, Trace 2

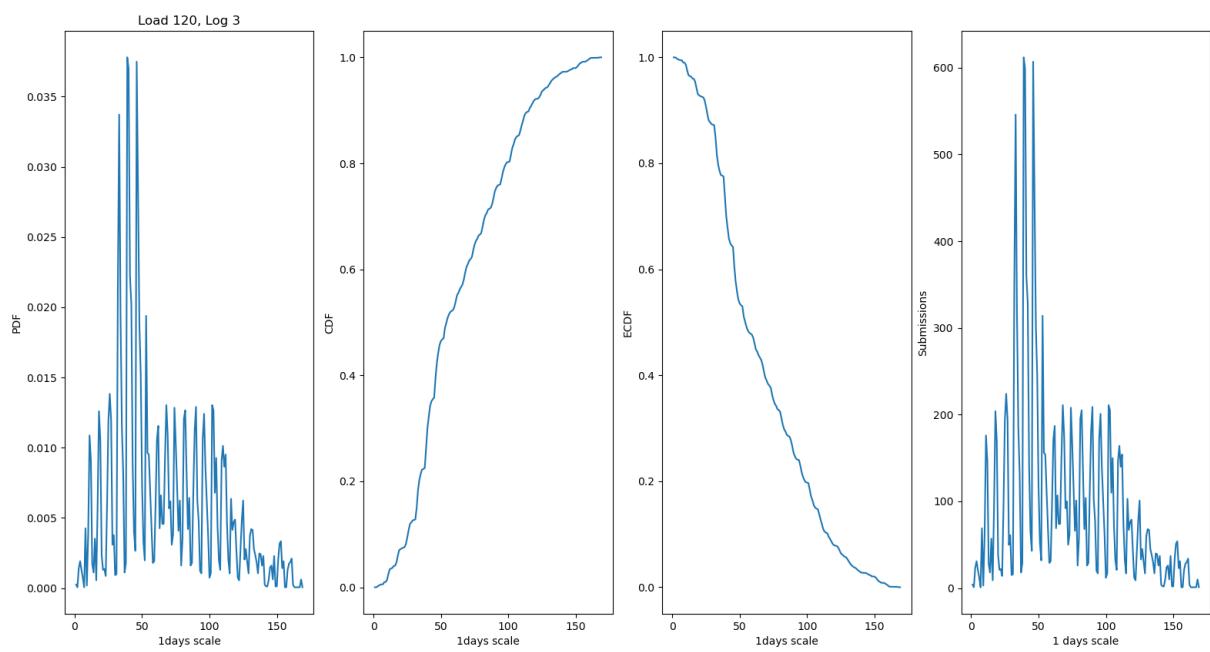


Figure 9-60: Load 120%, Trace 3

Trends and Cycles .7

נתחיל עם השוואת הטרנדים בגרפים.

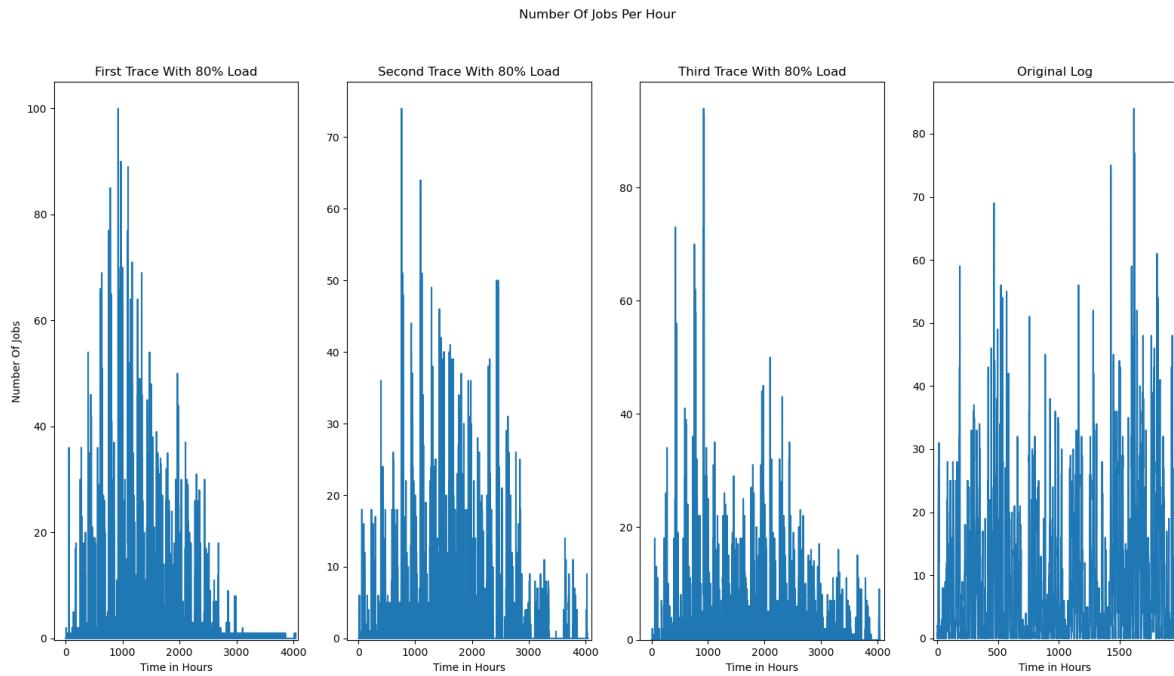


Figure 9-61

ראשית נסתכל על ה-Traces עם 80% load (תמונה 9-61). ניתן לראות שאנו בודקים את מספר ה-jobs שמתבצעים בשעה. ניתן לשים לב שישנו דמיון כל בין ה-Traces שייצרנו לבין ה-Trace המקורי. בחלק משיטת ה-User resampling יש לנו מספר Users שונים בחורם בכל שבוע מחדש באופן רנדומלי. בעקבות סיבה זו, קשה לטרנד להישאר נאמן לאופי הטרנד המקורי.

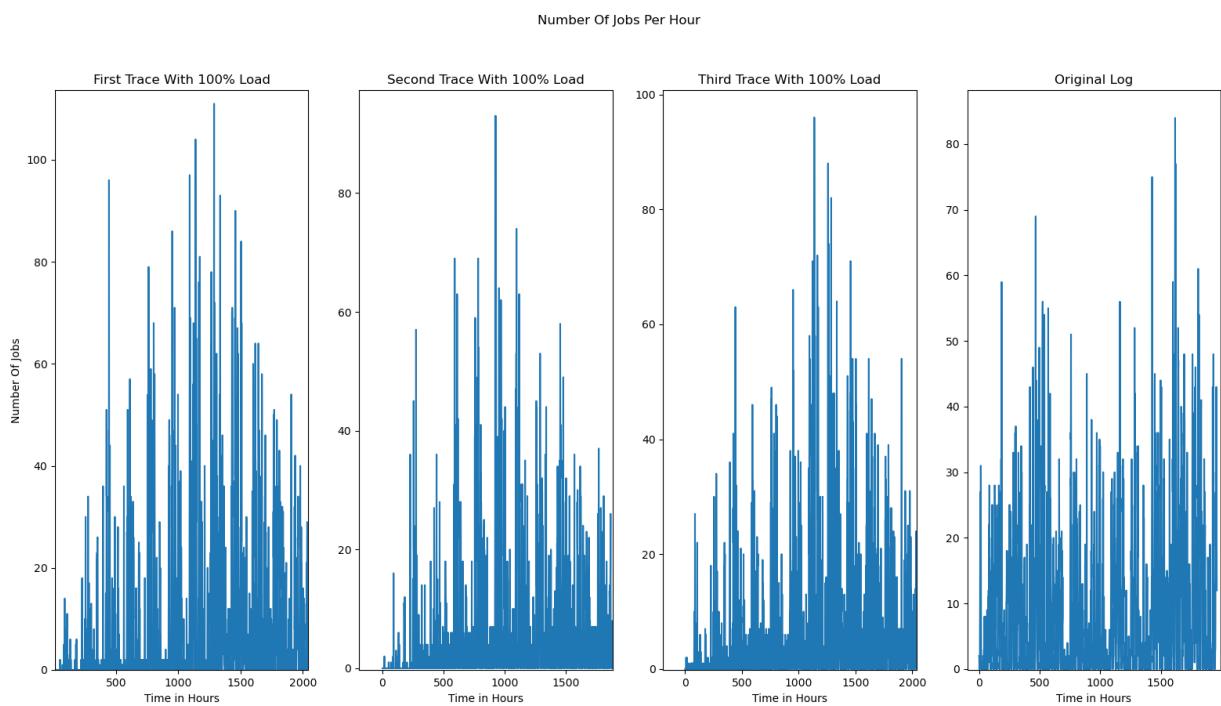


Figure 9-62

במהר לcker, ב-*batch* עם 100% load (תמונה 9-62), נשים לב שיחסו שונה בין התוצאות, כאשר ב-*batch* Trace המקורי ישנים שני "פיקים" שונים לב אליהם, בעוד ב-Traces שלנו יש "פיק" אחד יותר דומיננטי, זאת בעקבות אotta שיטת בחירת *Users*, אשר "שוברת" לנו את הטרנד.

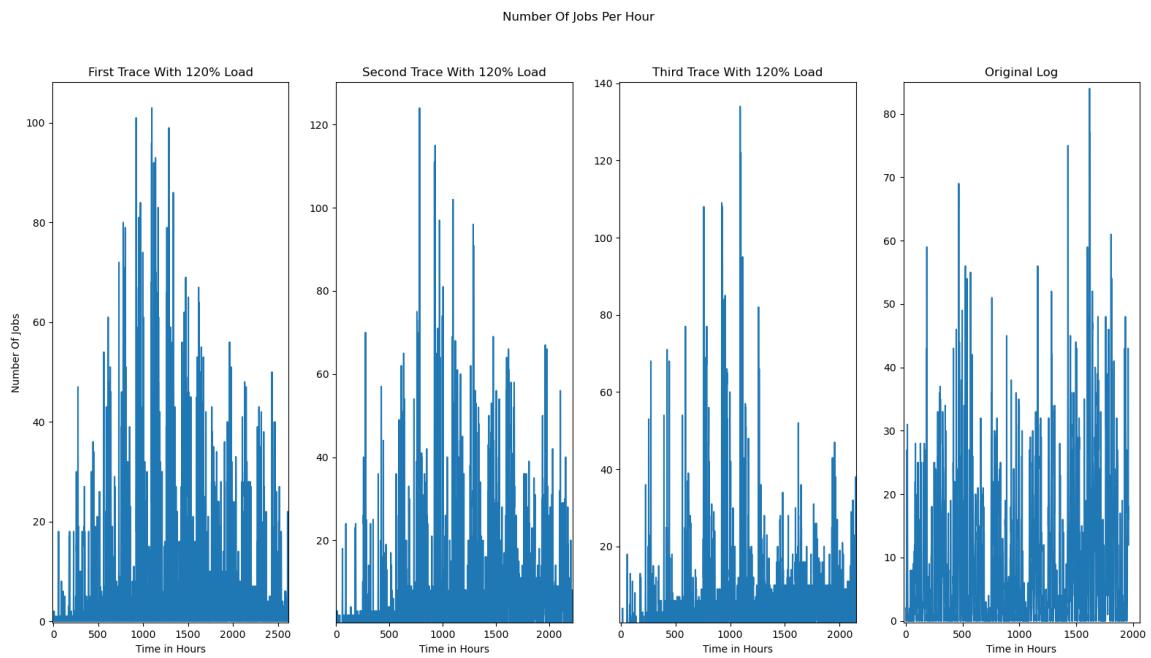


Figure 9-63

וב-*batch* עם 120% load (תמונה 9-63), בדומה לתיאור של ה-Traces ב-80% וב-100%, נראה כי קיימן קל (מצהים שני "גלים" בהיסטוגרמות, אך לא כמו ב-Trace המקורי).

כעת נבחן את ניתוח ה-Cycles.

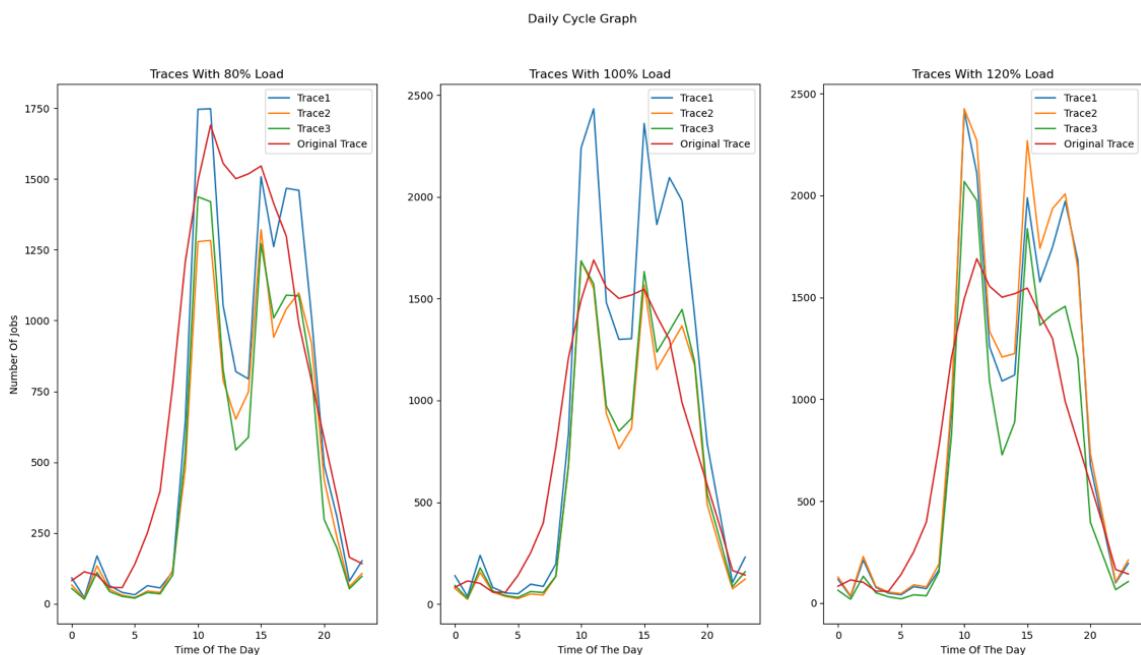


Figure 9-64

ראשית נביט על תמונה 9-64 אשר מתארת לנו מחזוריות יומיות. בכל אחד מהגרפים, מתחאים שלושת ה-Traces שלנו ובנוסף גם ה-Trace המקורי.

נשים לב, כי למעט זמנים ספציפיים (בין השעות 11-14) אשר בהם המגמה של הגרף דומה אך הערכים לא דומים, כל שאר שעות היום מאוד דומים.

ניתן בклות להוות את שעותימי העבודה, את הפסקות (השעות הממצוות של הפסקה), את סיום העבודה ואת הלילות. נראה כי למעט Trace אחד ב-100% load, כל ה-Traces האחרים דומים גם מבחינת הערכים וגם מבחינת המגמה המחזורית ל-Trace המקורי.

בעת נביט בגרף של מחזוריות שבועית (תמונה 9-65), המתואר באותו אופן של הגרף הקודם.

זהה כי גם בגרף שבועי, הדמיון נשאר דומה ונitin בклות להוות מחזוריות של שבוע – ימי חול וימי חופש. נראה כי בחלק מה-Traces, הערכים אמורים שונים מה-Trace המקורי, אך המחזוריות נשמרות בצורה טוביה מאוד למטרות מגבלות שיטות ציירת ה-Traces.

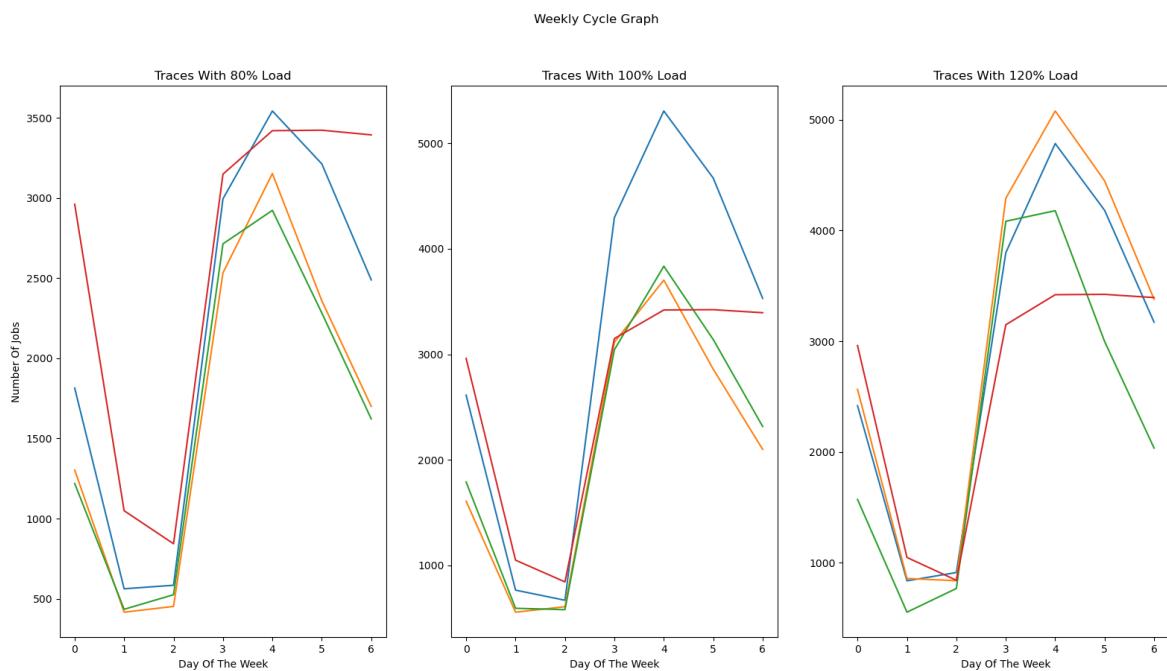


Figure 9-65

Wait Times .8

ראשית נראה את גраф ה-ECDF של Wait times תחת Loads שונים (תמונה 9-66-9).

נשים לב כי בכל Trace, ההסתברות לקבל אוטםWait times דומה מכך שמשתנה. זה קורה מכיוון שענו דוגמים את אותו Users בטעמים שונים – מה שמשנה את ההסתברות לכל Wait time.

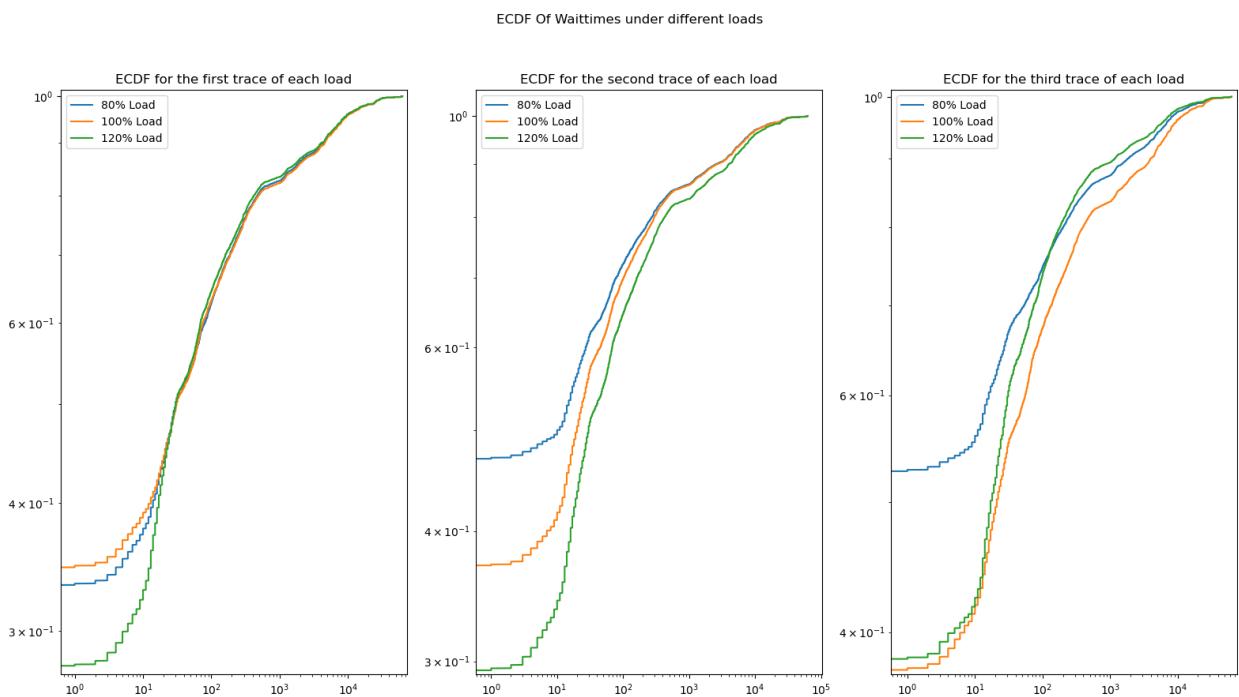


Figure 9-66

הגרפים הבאים (איורים 9-67, 9-68, 9-69) מתארים לנו Scatter plots של Wait times בפונקציה של Submit time בכל שלושת העומסים.

ניתן לראות כי באמצע הטווח של Submit times, Wait times הבי גדולים. בנוסף, בעקבות שיטת User Resampling, ומכיון שגם דוגמים מאותם משתמשים בעומסים שונים, לכל עומס של המערכת נקלט פיזור שונה מעט אשר נובע מאותם שינויים בעומסים.

Scatter Plot Of Wait Times As Function Of Subtimes

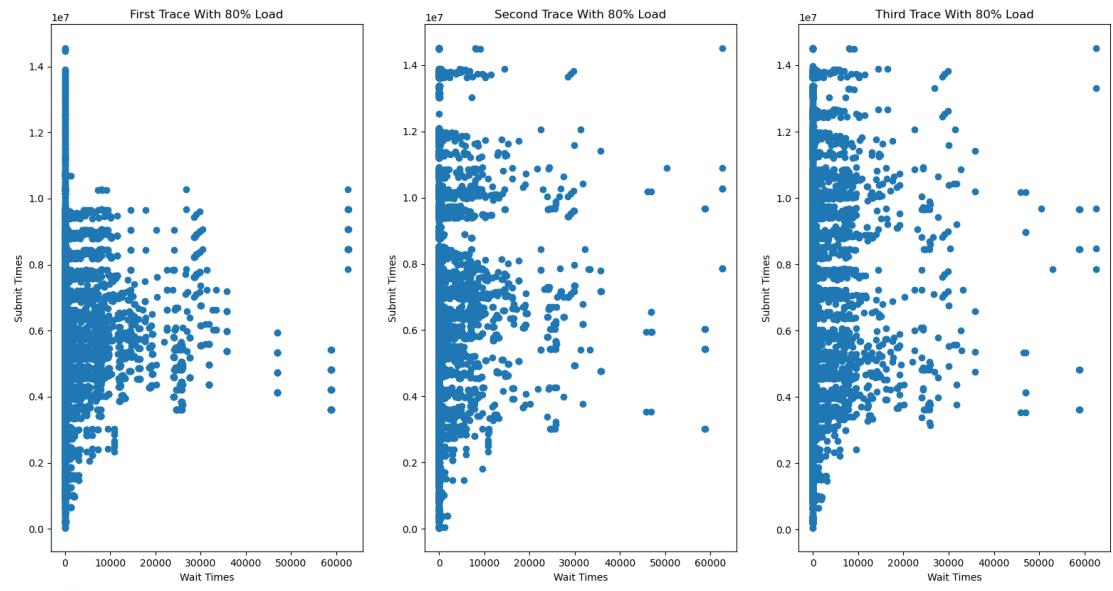


Figure 9-67

Scatter Plot Of Wait Times As Function Of Subtimes

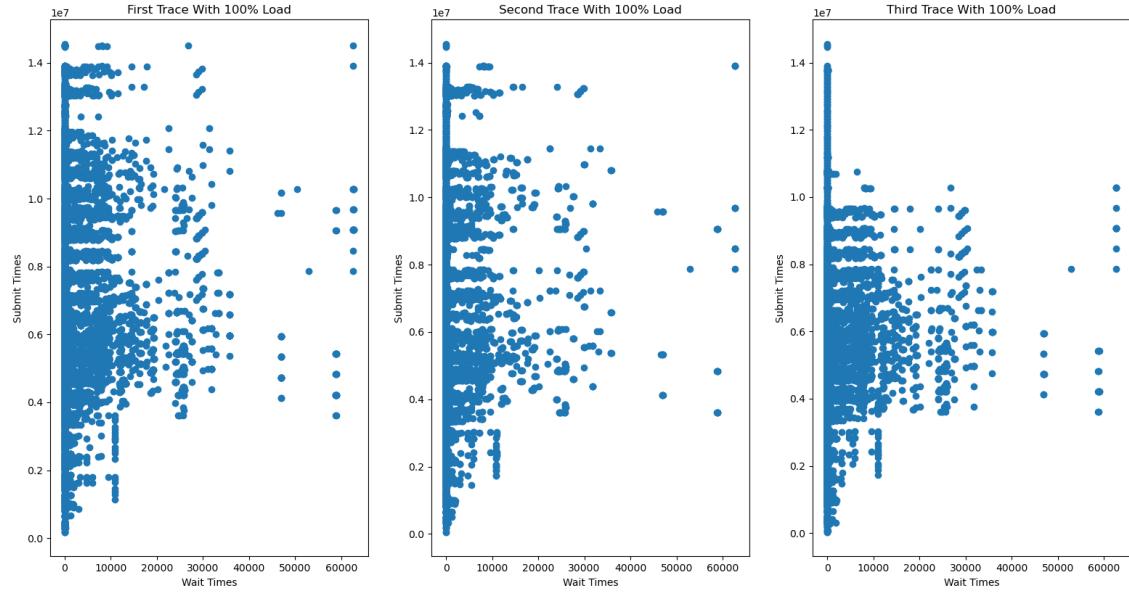


Figure 9-68

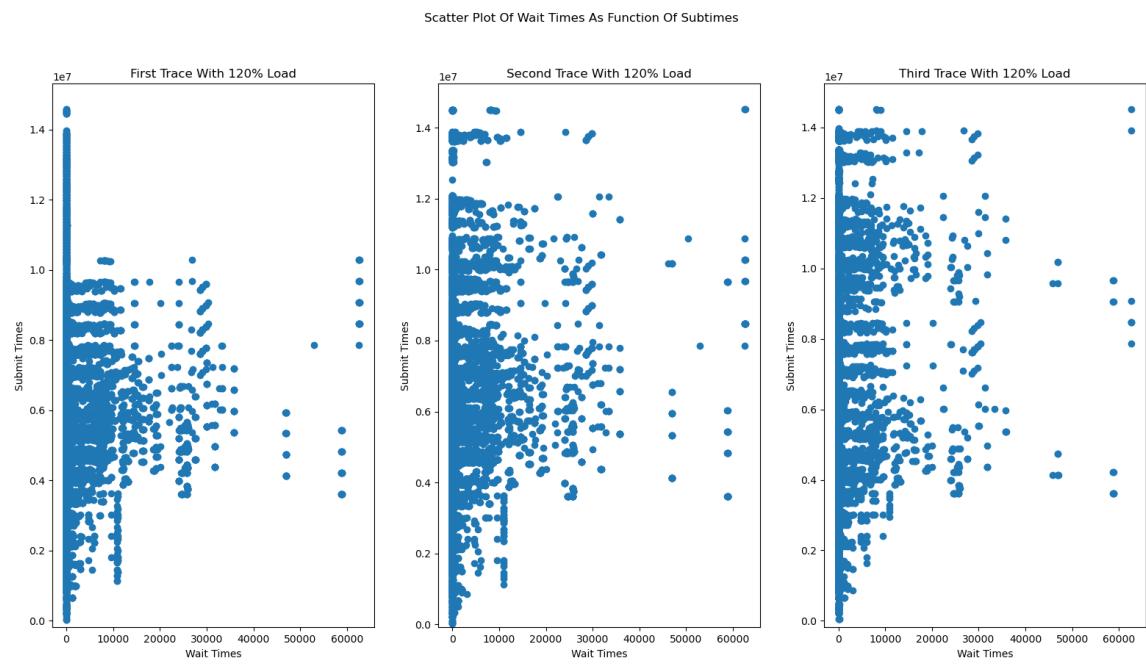


Figure 9-69

Different Analysis – Think Time . 9

בסעיף זה בחרנו לנתח ולהשווות את ה-Traces שהפקנו אחד עם השני ועם ה-Trace המקורי על פי ה-Think times שלהם. נשים לב שהאיורים הבאים מתארים זאת בצורה גרפית. מצד אחד, נראה כי קיימן דמיון רב בין ארבעת ה-Traces. אכן ארבעתם מתקדמים באותה מגמה והגרפים שלהם שומרים על אותה מונוטוניות לאורך כל הדרכן. מצד שני, ערכו ה-Think time מעט שונים זה מזה. ב-Traces שלנו, טווח העובדים גבוה יותר בממוצע – דבר אשר לא מפתיע אותנו כאשר יש עומס רב על המערכת (load 120%) אך קצת גבוה מיידי כאשר המערכת פחותה עמוסה. בנוסף, טווח הערכים של ה-Think time ב-Traces שלינו מתחילה מתחילה ציר ה-X (-1.4) לעומת ה-Trace המקורי אשר מתחילה ב-(0.8). ההבדל זה יכול לנובע מגוון סיבות, אך ככל הנראה הבדל זה קורה בעקבות שיטת ה-User Resampling, וכנגד לנו דוגמים במעט מוגבלת של Users.

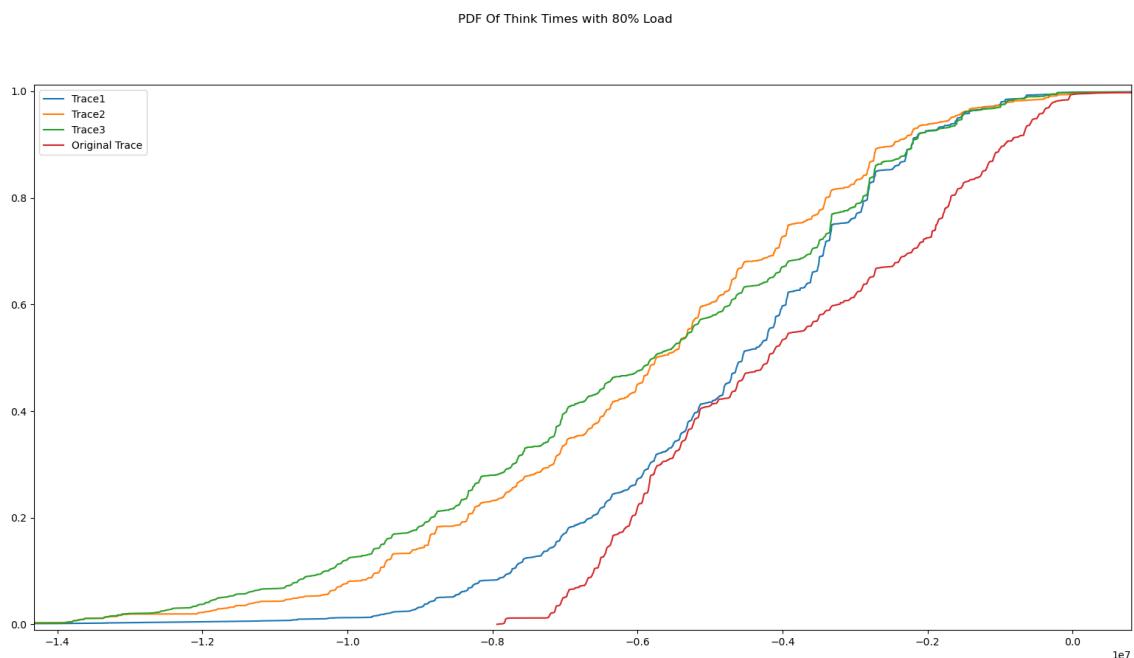


Figure 9-70

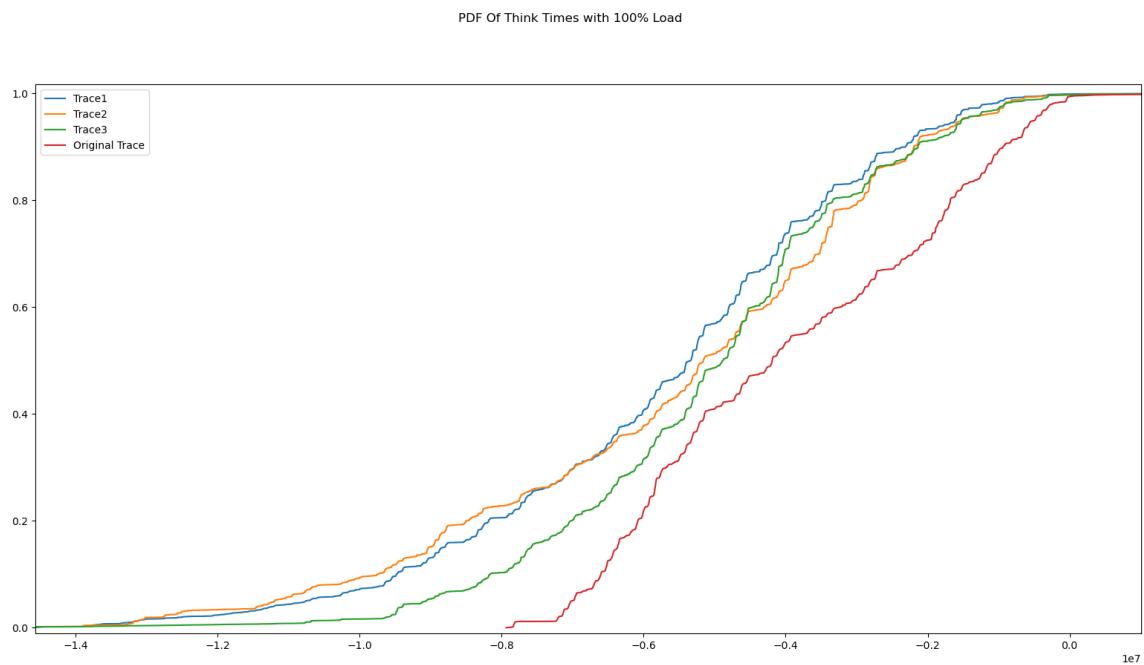


Figure 9-71

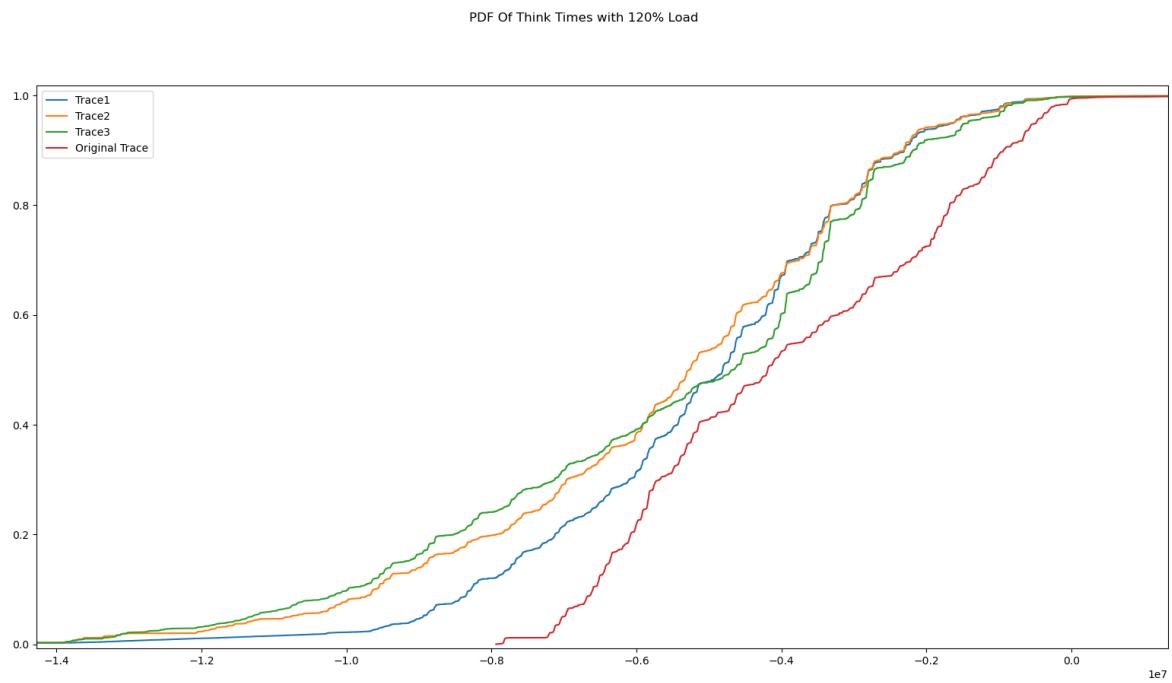


Figure 9-72

ביבליוגרפיה ומקורות

- Dror G. Feitelson, “Workload Modeling: Computer Systems Performance Evaluation”

Colophon

- Pycharm, PyCharm 2020.2.3 (Community Edition), Build #PC-202.7660.27, built on October 6, 2020
- Spyder – The Scientific Python IDE , Spyder 4.2.1