ISSN: 2277-5528
Impact Factor: 4.015

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & MANAGEMENT

MINIMISING EDA COST OF WITHOUT IMPACTING R&D EFFECTIVENESS

Niyati Trivedi[1], Prof. Dr. R. K. Sharma
niyatit10@gmail.com[1], rksharma@thapar.edu[2]
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala

ABSTRACT

The EDA industry offers a wide variety of software licensing models. EDA software tool for integrated circuit (IC) design are most of the time the second biggest financial budget cost for fabless IC design organizations (after compensations). Fabless manufacturing is the design and sale of hardware devices and semiconductor chips while outsourcing the fabrication or "fab" of the devices to a specialized manufacturer called a semiconductor foundry. As a result, numerous organizations have concentrated on dynamic administration of this financial plan to secure a focused advantage. The steadily expanding many-sided quality of IC plans has been empowered by an always growing scope of progressively effective programming tools.

This paper presents the software package to monitor summarized or average peaks of license usage and a summary table of this monitored data is used to estimate required license pools for remixes and for renegotiating EDA rental agreements. We have design this project at "Infineon Technologies India Pvt. Ltd.", which was established at Bangalore in 1997 and was earlier known as Siemens Semiconductors.

Keywords: EDA, ECAD, fabless IC, Software Package, License. Infineon Technologies India Pvt. Ltd.

INTRODUCTION

Electronic Design Automation (EDA), also known as Electronic Computer-Aided Design (ECAD), is a process for designing electronic systems. EDA is a key technology enabler of Real-Time Enterprises (RTE). EDA complements existing enterprise technologies by providing the infrastructure needed to support the event-driven solutions that analyze complex relationships between business events to identify business opportunities, threats, and anomalies. Business users need timely information about significant business activities to improve the quality of the decisions they make and to close the insight-to-action gap.

Why EDA software licensing and management of EDA software licenses is an important topic in semiconductor industry at all ? The reason is that current technologies for EDA software licensing and management of EDA software licenses have been designed under the assumption that the license server (reason for the authorization of the usage of a license protected application) and the EDA tools are located in the same administrative and network domain, and EDA budget causes the second highest budget after employee salaries and there is a need for effective management of the EDA tool licenses Thus, these licenses are provided on the basis of named users or clients (NNU), hostnames (IP-addresses), or regularly as a geographical site license for the administrative domain of the firm.

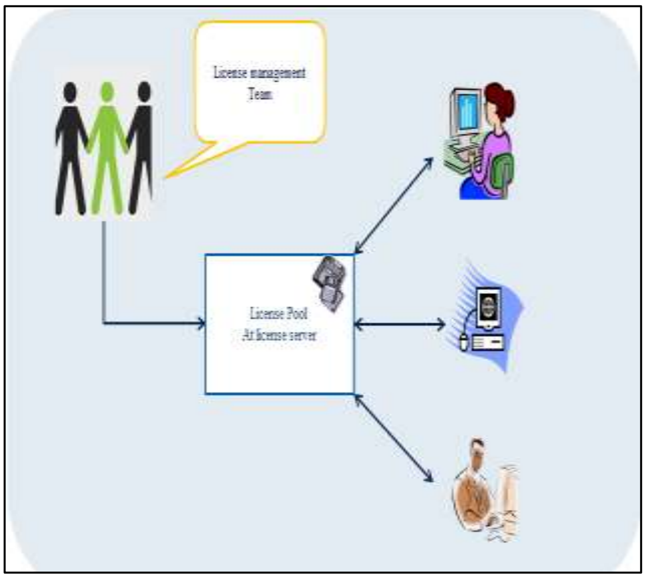


Figure 1: License Server setup

If we want to use this EDA tool software in a distributed service oriented setup, using assets that are spread across various administrative domains the network license server setup can be seen in figure 1.

WAITING LOUNGE

Waiting Lounge is an application for admins and managers, as well for end users. In this we are processing a big license statistics text file and capturing all the information for the checkout licenses, like, user id, host, job id, process id, start time of the checkout and all such information. Sometimes when the license limit is reached, some licenses are queued. We are also capturing the queued licenses information. This is for tracking queued license users for different tools; it was initially started with two big vendors and shows average, minimum and maximum waiting time for individual users for particular tool.

QUEUING THEORY

Queuing theory is the mathematical study of waiting lines, or queues. Queuing theory is used to understand the behaviour of queuing systems. The main components of a queuing system include queues (waiting line), customers (in need of service) and servers (who serve the customers). The basis of queuing theory lies with the understanding of arrival and service rates of the system.

SYSTEM ARCHITECTURE

System architecture is a conceptual model which explains structural behaviour of the system and architecture can comprise system components and for software architecture is the high level of structure of a software system. The following block diagram (Figure 2) shows the Heatmap tool architecture.

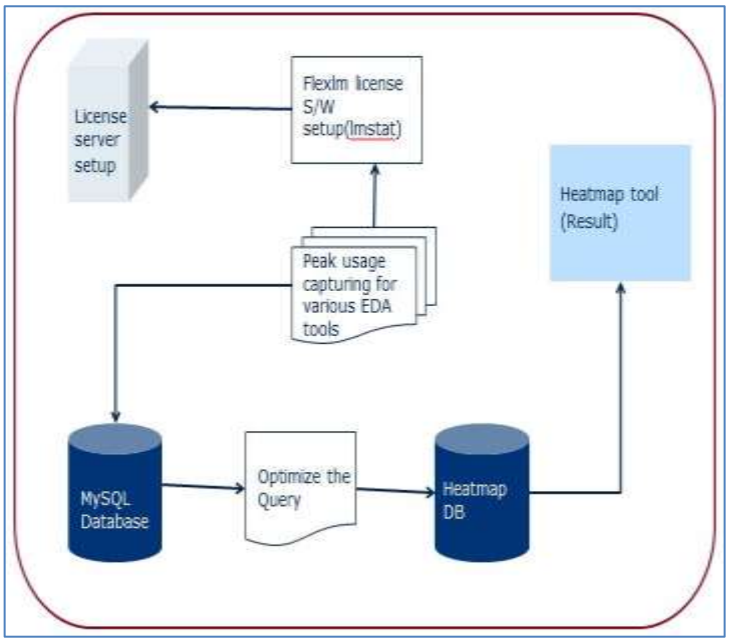


Figure 2: System Architecture of Heatmap

RESULTS ANALYSIS

A. Waiting Lounge and Prediction

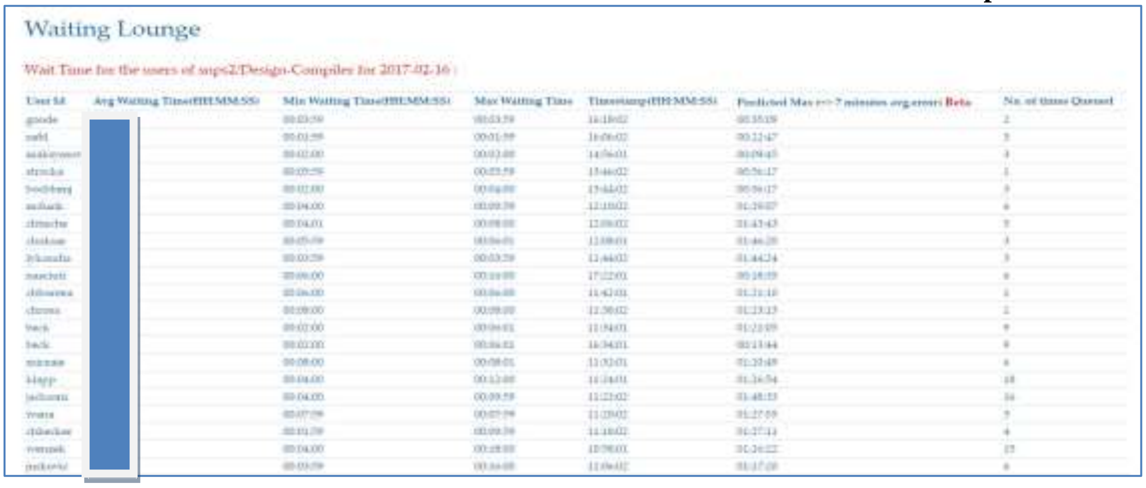
Waiting Lounge is for tracking queued license users for different tools initially started with two major Vendors. It shows average, minimum and maximum waiting time for individual users on tool basis. Sometimes when the license limit is reached for a particular tool, then licenses are queued. We are also capturing the queued licenses information,

Waiting time is predicted on almost 26 parameterswhich are prepared and passed in the "xgboost model" using R which uses eXtreme Gradient Boosting algorithm and prediction is done on the queued licenses, which tells the user that how much more time he has to wait for a particular license.

This application started with Vendor1 only, that time accuracy was around 80 %, then Vendor2 was added, and currently Vendor3 is added to this application. After Vendor3 was added, accuracy dropped down to 65%, but now it has raised to 82 %. Earlier, there were only two tables (Main Table and Archive Table) in database for all the three vendors, but now we have 6 tables, two tables handling each vendor.

We have done analysis on the accuracy dropdown; this was because of the different format of license statistics file of the vendors. Currently we are running 4 models with same xgboost method, but giving different parameters to them and testing them with different training-testing ratio. The best result is coming from 80:20 division of training and testing data.

ISSN: 2277-5528
Impact Factor: 4.015



Waiting Lounge

Wait Time for the users of snps2/Design-Compiler for 2017-02-16 :

User Id	Avg Waiting Time(HH:MM:SS)	Min Waiting Time(HH:MM:SS)	Max Waiting Time	Timestamp(HH:MM:SS)	Predicted Max (+-7 minutes avg error) Beta	No. of times Queued
goode	[redacted]	00:03:59	00:03:59	16:18:02	00:35:19	2
nahl	[redacted]	00:01:59	00:01:59	16:06:02	00:22:47	5
[illegible]	[redacted]	00:02:00	00:03:00	14:56:01	01:09:43	3
[illegible]	[redacted]	00:03:59	00:03:59	15:46:02	00:56:17	1
[illegible]	[redacted]	00:02:00	00:04:00	15:44:02	00:56:17	3
[illegible]	[redacted]	00:04:00	00:09:59	12:10:02	01:29:07	6
[illegible]	[redacted]	00:04:01	00:08:00	12:06:02	01:43:43	7
[illegible]	[redacted]	00:05:59	00:06:01	12:08:01	01:46:28	3
[illegible]	[redacted]	00:03:59	00:03:59	12:44:02	01:44:24	3
[illegible]	[redacted]	00:06:00	00:14:00	17:22:01	00:18:59	6
[illegible]	[redacted]	00:06:00	00:06:00	11:42:01	01:21:10	1
[illegible]	[redacted]	00:09:00	00:09:00	12:30:02	01:23:13	2
back	[redacted]	00:02:00	00:04:01	11:54:01	01:22:09	9
back	[redacted]	00:02:00	00:06:02	16:34:01	00:13:44	8
[illegible]	[redacted]	00:08:00	00:08:01	11:02:01	01:20:49	6
klapp	[redacted]	00:04:00	00:12:00	11:24:01	01:26:54	18
[illegible]	[redacted]	00:04:00	00:09:59	11:23:02	01:48:53	16
[illegible]	[redacted]	00:07:09	00:07:09	11:20:02	01:27:59	7
[illegible]	[redacted]	00:01:59	00:09:59	11:18:02	01:27:11	4
[illegible]	[redacted]	00:04:00	00:18:00	10:58:01	01:26:22	15
[illegible]	[redacted]	00:03:59	00:16:00	11:06:02	01:17:20	6

Figure 3: Waiting Lounge Snapshot

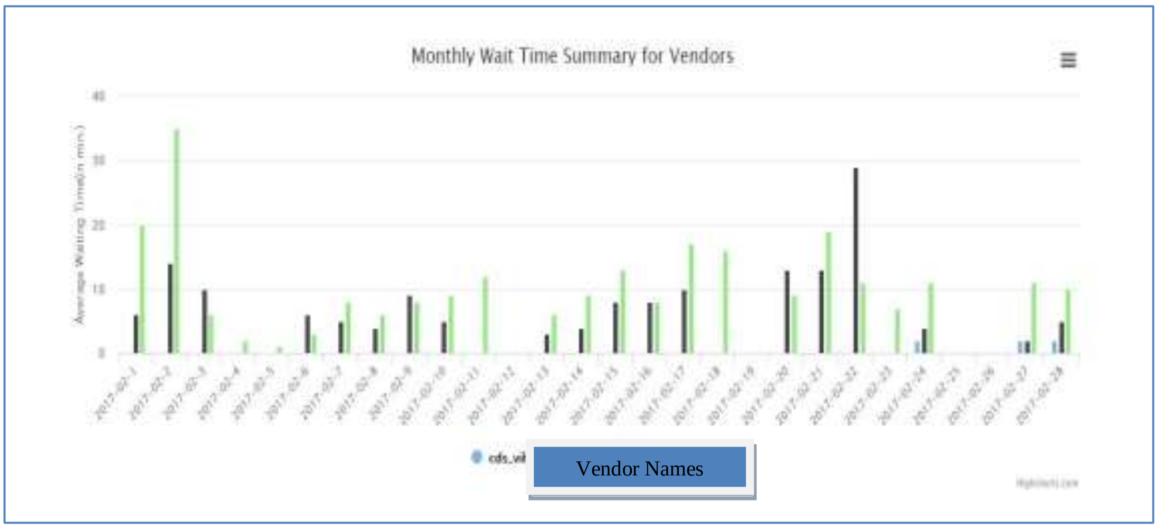


Figure 4: Waiting Lounge for Monthly wait time summary for vendors

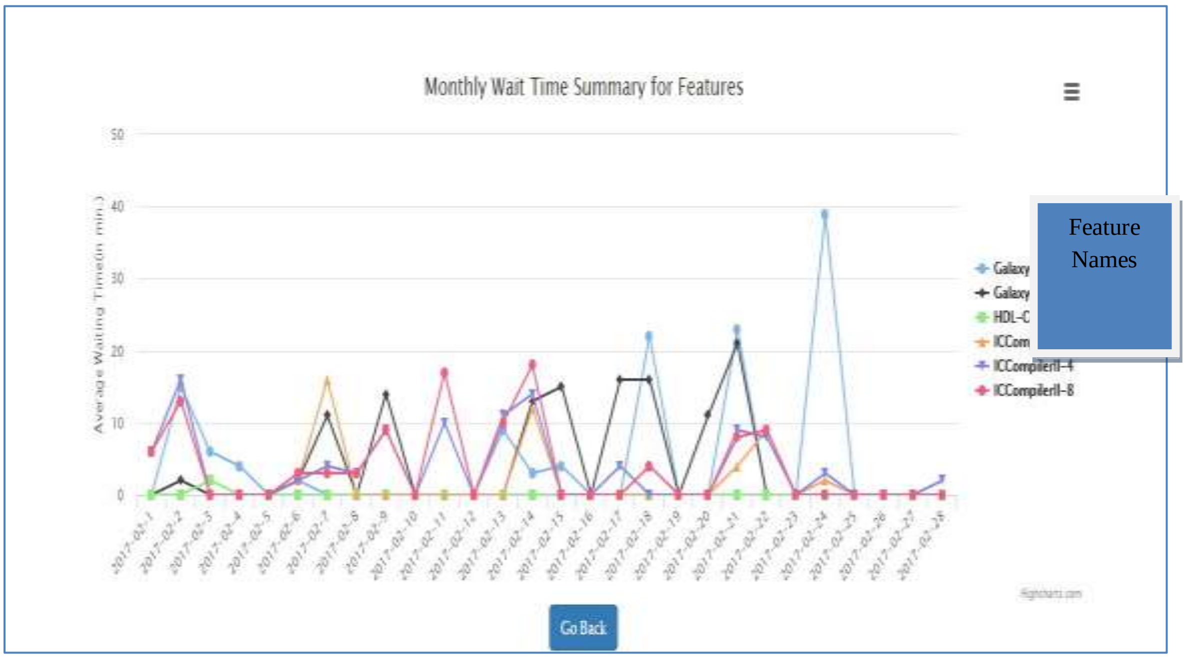


Figure 4: Waiting Lounge for Monthly wait time summary for features

For predicting wait times, we have used technologies like Perl, R – xgboost (xgbtree), based on gradient boost algorithm and PHP. The production output was 84% accurate initially, when Vendor3 is added, accuracy went down to 68% and the current accuracy 91%.

B. Experiment Results for Method I - Queuing Theory

License checkout data of March 2017 from 1 site for Vendor1's feature1 from LISA database was used. For March the number of available licenses varied between 18 and 20. The average arrival time was 3.75 minutes and average service time was 59.06 minutes.

B.1 Analysis of March 2017 M/M/C results

Table 1: March 2017 – M/M/C results

No of Licenses available	Arrival Rate	Service Rate	Average				
			Number of Request		Waiting Time in		Utilization of licenses
			System	Waiting in queue	System	Queue	
18	0.26667	0.01693	19.14	3.39151764	71.78	12.72	87%
19	0.26667	0.01693	17.39	1.63594654	65.19	6.13	83%
20	0.26667	0.01693	16.60	0.84896559	62.24	3.18	79%

With the help of the model we can calculate the waiting times, no. of requests and utilization for different values of available licenses. In March the number of licenses was increased from 18 to 20 during the course of the month. Each row in Table 1 shows the model results if the license number was set to a particular value for the entire month. This helps in critically analysing different scenarios for different conditions and designing the license configurations.

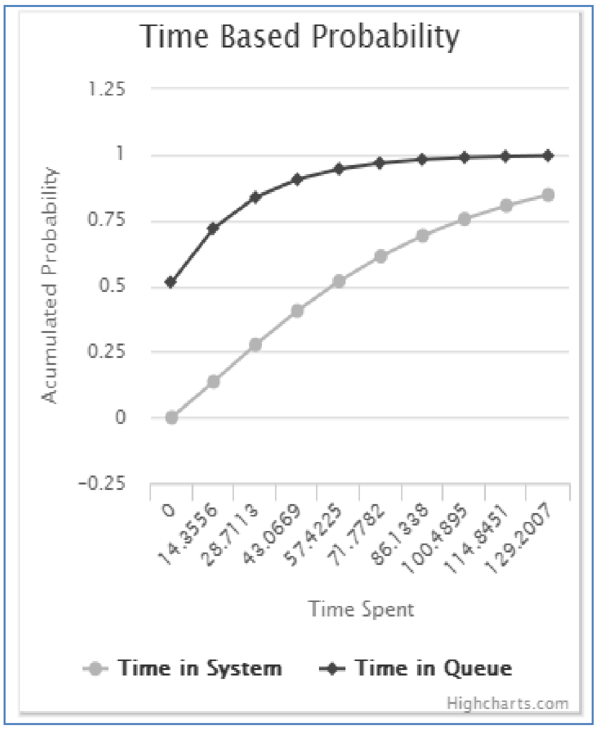


Figure 5: March 2017 – Time based probability

B.2 Analysis March 30th 2017 M/M/C Results

Though the utilization of the licenses was not at its maximum at 87%, the licences were still increased twice because of short term demands on 30th and 31st March. The reason for the increase was that there were many queues in the license servers than normal. We tried out the queuing model for all the 24 hours individually for 29th and 30th March. Queuing theory results for March 30th is shown in Table 2. For this period, Average arrival time was 3.27 minute and Average service time was 60.73 minute. With 18 licenses, the model fails for March 30th as the demand exceeded the capacity. This is a common behaviour as, in some day's maximum EDA users work in parallel and submit multiple requests in parallel, causing sudden rise in arrival rate. Because the arrival rate is greater than the servicing capacity of the server, the model fails. As the demand was great, it was essential to increase the license number to 19 to finish the jobs within the same day.

Table 2: January 13th 2017 – M/M/C Results

No of Licenses available	Arrival Rate	Service Rate	Average				
			Number of Request		Waiting Time in		Utilization of licenses
			System	Waiting in queue	System	Queue	
18	0.30057	0.01647	-	-	-	-	-
19	0.30057	0.01647	38.05	19.79	126.58	65.85	96%
20	0.30057	0.01647	24.51	6.26	81.56	20.83	91%

B.3 Analysis March 31st 2017 M/M/C Results

For the period of March 31st the arrivals were observed to be more frequent with an Average arrival time of 1.71 min, but at the same time service time reduced drastically to 30.11 minutes. This is the reason as shown in Table 3, with 18, 19 the model works, as the demand is within the capacity. Unlike March 30th, the actual increase in license could have been avoided on March 31st, by maintaining the license at 19.

Table 3: March 31st 2017 – M/M/C Results

No of Licenses available	Arrival Rate	Service Rate	Average				
			Number of Request		Waiting Time in		Utilization of licenses
			System	Waiting in queue	System	Queue	
18.00	0.5848	0.03321	57.80	40.19	98.84	68.73	98%
19.00	0.5848	0.03321	26.00	8.39	44.46	14.35	93%
20.00	0.5848	0.03321	21.15	3.54	36.17	6.06	88%

C Experiment Results for Method II-Average Based

We try an alternate method for prediction of wait times for license availability at any given time using descriptive statistics and exploratory data analysis. The same feature1 usage dataset from LISA for the month of March-2017 was used.

C.1 Understanding distribution of runtimes for month of March-2017

With the help of the available historical data of previous checkout durations at any given point in time we obtain a frequency distribution which explains the categories inside which majority of checkout durations were completed. The Figure 6 below represents the percentage of checkouts completed within different intervals of time during the month of March, 2017.

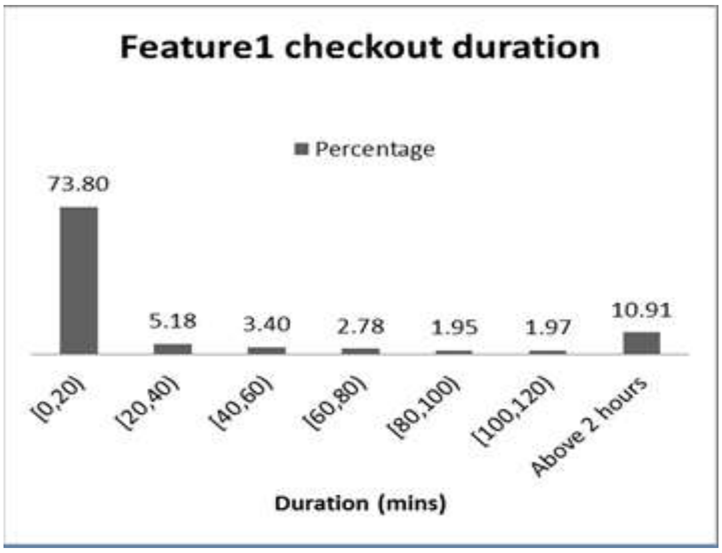
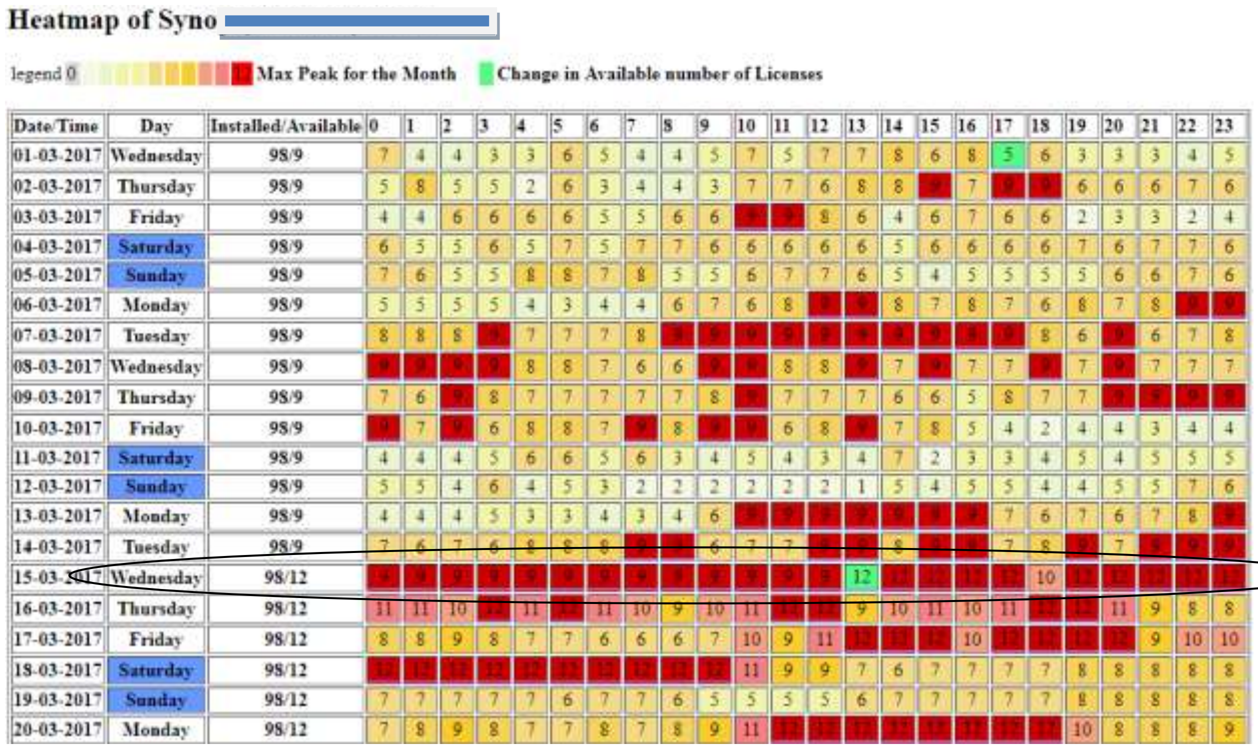


Figure 6: March 2017 feature1 checkouts

ISSN: 2277-5528
Impact Factor: 4.015

C.2 Fix initial conditions for month of March-2017

Wait time predictions are mostly useful in situations where the installed number of licenses does not meet the demand. In such situations, multiple queues are formed and user requests get delayed. In order to solve the problem of wait time prediction for such crucial situations, of the entire Jan data we need to select data which would be representative of the peak/busy days for our analysis. We will now select a busy day of March which will best represent such situations. Looking at the heatmap below Figure 7, we selected March 15th, 2017, since there were continuous peak hours with maximum license usage.



Heatmap of Syno [redacted]

legend 0 🟥 Max Peak for the Month 🟩 Change in Available number of Licenses

Date/Time	Day	Installed/Available	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
01-03-2017	Wednesday	98/9	7	4	4	3	3	6	5	4	4	5	7	5	7	7	8	6	8	5	6	3	3	3	4	5
02-03-2017	Thursday	98/9	5	8	5	5	2	6	3	4	4	3	7	7	6	8	8	9	7	9	9	6	6	6	7	6
03-03-2017	Friday	98/9	4	4	6	6	6	6	5	5	6	6	9	9	8	6	4	6	7	6	6	2	3	3	2	4
04-03-2017	Saturday	98/9	6	5	5	6	5	7	5	7	7	6	6	6	6	6	5	6	6	6	6	7	6	7	7	6
05-03-2017	Sunday	98/9	7	6	5	5	8	8	7	8	5	5	6	7	7	6	5	4	5	5	5	5	6	6	7	6
06-03-2017	Monday	98/9	5	5	5	5	4	3	4	4	6	7	6	8	9	9	8	7	8	7	6	8	7	8	9	9
07-03-2017	Tuesday	98/9	8	8	8	9	7	7	7	8	9	9	9	9	9	9	9	9	9	9	8	6	9	6	7	8
08-03-2017	Wednesday	98/9	9	9	9	9	8	8	7	6	6	9	9	8	8	9	7	9	7	7	9	7	9	7	7	7
09-03-2017	Thursday	98/9	7	6	9	8	7	7	7	7	7	8	9	7	7	7	6	6	5	8	7	7	9	9	9	9
10-03-2017	Friday	98/9	9	7	9	6	8	8	7	9	8	9	9	6	8	9	7	8	5	4	2	4	4	3	4	4
11-03-2017	Saturday	98/9	4	4	4	5	6	6	5	6	3	4	5	4	3	4	7	2	3	3	4	5	4	5	5	5
12-03-2017	Sunday	98/9	5	5	4	6	4	5	3	2	2	2	2	2	2	1	5	4	5	5	4	4	5	5	7	6
13-03-2017	Monday	98/9	4	4	4	5	3	3	4	3	4	6	9	9	9	9	9	9	9	7	6	7	6	7	8	9
14-03-2017	Tuesday	98/9	7	6	7	6	8	8	8	9	9	6	7	7	9	9	8	9	9	7	8	9	7	9	9	9
15-03-2017	Wednesday	98/12	9	9	9	9	9	9	9	9	9	9	9	9	9	12	12	12	12	12	10	12	12	12	12	12
16-03-2017	Thursday	98/12	11	11	10	12	11	12	11	10	9	10	11	12	12	9	10	11	10	11	12	12	11	9	8	8
17-03-2017	Friday	98/12	8	8	9	8	7	7	6	6	6	7	10	9	11	12	12	12	10	12	12	12	12	9	10	10
18-03-2017	Saturday	98/12	12	12	12	12	12	12	12	12	12	12	11	9	9	7	6	7	7	7	7	8	8	8	8	8
19-03-2017	Sunday	98/12	7	7	7	7	7	6	7	7	6	5	5	5	5	6	7	7	7	7	7	8	8	8	8	8
20-03-2017	Monday	98/12	7	8	9	8	7	7	8	7	8	9	11	12	12	12	12	12	12	12	12	10	8	8	8	9

Figure 7: March 2017 Feature1 heatmap

Based on our selection of the representative data (a busy day – March 15th), we calculate the distribution of runtimes on that particular day as shown in Table 4.

Table 4: March 31st 2017 Runtime distribution

Interval	Frequency	Percentage
[0, 10)	343	72.36
[10, 20)	18	3.80
[20, 30)	5	1.05
[30, 40)	9	1.90
[40, 50)	18	3.80
[50, 60)	3	0.63
Above 60	78	16.46

From the above distribution for the run times on March 15th, 2017, it is evident that:

1. 76% of the license requests were completed within 20 minutes.
2. 24% of the license requests were completed after 20 minutes.

It is important to note that 76% of the license requests are invoked by short runners (requests which use a license within 20 minutes). This is in line with the Pareto principle (also known as 80-20 rule) which states that for many events; roughly 80% of the effects come from 20% of the causes. Since, the top 20% of the license requests i.e., requests which use a license within 20 minutes have around 76% impact on the overall pattern of license requests, we concentrate more on these requests. We set these findings as a base to predict wait times

Arrived base conditions

1. The percentage split between short runners and long runners is 76:24 at any point in time.
2. The standard duration for short runners is less than or equal to 20 minutes.

D Validation on month March 2017

We test the above two base conditions if these can be generalized and applied to any situation by validating them against:

D.1 Validation using data of a complete month March 2017

The below Table 5 explains the percentage split between short and long user checkouts during the month of March, 2017, and Table 6 explains the wait time statistics.

Table 5: Runtime Statistics for complete month March 2017

Run time parameters	Results
Total no of checkouts	6034
Total no of checkouts less than 20 minutes	4277
Percentage of checkouts less than 20 minutes	71%
Split between short runners to long runners for the whole month	71, 29

Table 6: Wait time Statistics for complete month March 2017

Wait time parameters	Results
Total no of queued checkouts	125
No of wait times less than 20 minutes	123
Percentage of wait times less than 20 minutes	98.4%
Average of all the wait times for the complete month	3.47 minutes

D.2 Validation using data at a random point on 10th March, 2017

First random validation point: 10th March, 2017 at 8:45 A.M. Let us see if the base conditions are true when data from the start of the month till the random point 8:45 A.M on 10th March 2017 is taken. The below Table 7 for Runtime Statistics and 8 for Wait Time Statistics, explains the percentage split between short and long user checkouts before 8:45 A.M on 10th March, 2017.

Table 7: Random runtime Statistics on 10th March, 2017

Parameters (related to run times)	Results
Percentage of checkouts less than 20 minutes	69.23%
Ratio between short runners to long runners for the whole month	69 : 31

Table 8: Random wait time Statistics on 10th March, 2017

Parameters (related to wait times)	Results
Total no of queued requests before 8:45 A.M	4
No of queued requests below 20 minutes	4
Percentage of wait times less than 20 minutes	100%
Average of all the wait times for the complete day	2.40 minutes
Note: For the whole day, only one request was above 20 minutes, and it had lasted for 20.98 minutes	

CONCLUSION

These requirements target increased flexibility of the license management to cope with the flexibility of and security of the licenses token and integrity of the execution environment. The first requirement describe deployment of parts of the license management components. In the course of the project these requirements will result in a number of developments enhancing the Dash board to monitor licenses and which will be used for Data Analytics and applying Machine Learning on that to be used in monitor live license statistics.

REFERENCES

I. Sergiu Costea and Bogdan Warinschi "Secure Software Licensing: Models, Constructions, and Proofs", *29th Computer Security Foundations Symposium, IEEE*-2016.

II. Jan Schmidt and Petr Fiser "On Robustness of EDA Tools", *17th Euromicro Conference on Digital System Design*-2014,

III. Bo Chen Wei-wei Zhang and Ling yu "Cloud Licensing Model for .NET Software Protection" *The 7th International Conference on Computer Science & Education (ICCSE 2012).*

IV. Petr Fiser and Jan Schmidt "Sources of Bias in EDA Tools and Its Influence" *IEEE*- 2014.

V. Jethro B. de Guzman "Mobile Emergency Response Application Using Geolocation for Command Centers" *IJCCE*, Vol.3 (4), 2014

VI. James T. O. Connor "Present Value Analysis Applied to EDA Software Alternatives" *Economics of Design, Test and Manufacturing IEEE*-1994.

VII. Mukesh Singhal & Niranjan G. Shivratri. Voting and Election Algorithms. *In Advanced concept in operating Systems*, pages 209 & 343, 2002

VIII. George Coulouris, Jean Dollimore and Tim Kindberg. Election Algorithm, Bully Algo & Ring based algo. *In Distributed Systems*, page 445-448, 2006

IX. Leili Noorian and Mark Perry. Autonomic Software License Management System: an implementation of licensing patterns. *Fifth International Conference on Autonomic and Autonomous Systems. IEEE*, 2009

X. Mikhail J. Atallah and Jiangtao Li. Enhanced Smart-card based License Management. *IEEE International Conference on E-Commerce IEEE*, 2003.