Developing Toward an SLA: Understanding the User Load

Tom Wilson

1 Introduction

The previous installments of this series looked at three facets of evaluating an SLA. In [Wil10c], we looked at the variation of the response times across transaction types; this was a notional analysis. In [Wil10b], we investigated the variation in average response time because of load; we also took a real transaction and studied its response time distribution. In [Wil11a], we considered how transaction performance varies because of the data referenced by the transaction. This aspect is complicated by the growth of the user base and the database over the operations lifecycle.

In this paper, we want to understand how the user load varies not only during the SLA charge period, but also during the operations lifecycle. The variation in user load during the SLA charge period is closely related to the transaction load variation presented in [Wil10b]. We also want to address a common misunderstanding concerning the number of concurrent users. The growth in the user base during the operations lifecycle has already been covered in [Wil11a]. Here, we suggest some possible values in order to understand the load during the SLA charge period from year-to-year.

2 Evaluating an SLA

The system providing example data supports logistics and maintenance for military equipment. For the purpose of anonymity, some general details are provided, and some specific details have been changed. This proprietary system has a response time SLA defined for it. For our purpose, the SLA is:

> The system shall respond to user requests with an average response time of 5 seconds for 98% of the transactions during the monthly charge period.[1] The system shall support 3000 concurrent users.

Other details exist in the SLA that are not pertinent to this discussion.

It is common in an SLA specification to define a number of users that must be supported. However, this specification may not be appropriate for the SLA definition. In this case, the response time aspect of the SLA calls for averaging most of the response time measurements across the charge period (e.g., one month). We already saw that transaction load varies across the month ([Wil10b]), and we know that users and transactions are highly correlated ([Wil10e]). Therefore, a thorough user specification should define counts for the whole charge period.

This thorough user specification may not seem necessary for the SLA (i.e., the customer does not want to bother with such details, nor is he likely to agree to them), but it is critical for the engineering effort. As we have discussed in the other installments of this series, response times are permitted to exceed the 5 second value defined in the SLA. Some of the response times are discarded because of the "98%" clause; some are retained but are compensated for by other, low response times when the averaging is performed. To better manage risk, we need to understand the performance during all intervals of the charge period. That means understanding the load during these intervals.

As previously mentioned in [Wil11a], what is also missing from the SLA is a description of the growth during the operations lifecycle; only the final goal is defined. Nonetheless, since most of the risk lies with the contractor, understanding this growth is critical to a cost-effective system maintenance plan. The system is used in a military environment, not a commercial one. Users will not be showing up when they hear how great the system is or when some interesting function becomes available.[2] New users arrive according to a deployment plan—a plan not detailed in the SLA. A deployment plan would define when organizations would transition to the new system. Such a plan is complicated by the numerous other activities that the organizations are supporting. Without understanding the deployment plan, the system will be over-engineered for a load that will not exist for a few years.

[1] This means that after the top 2% of response times are eliminated, the remaining 98% must have an average that is less than or equal to 5 seconds.

[2] Having no competition has both advantages and disadvantages.

2.1 Definitions

A *user* executes work on a system, typically in the form of transactions. A *session* is associated with a user and holds some resources whether the user is active or idle. We will define *active* as executing work (e.g., a transaction) and *idle* as not executing work. When a user is active, he is interacting with the system. An idle user may be thinking between actions or not interacting with the system at all (we cannot tell which). A *unique* user is active at least once during a time interval. The same user can have more than one non-overlapping session and still only be counted once, i.e., he can logout and login again. We will not consider the case where a user has two or more overlapping sessions.[3]

Concurrent users are a collection of unique users during an interval. For a closed system, the number of concurrent users is essentially the number of users in the system since the sessions persist and overlap. This assumes that unusually long idle times do not occur. For an open system, the number of concurrent users is difficult to define since users enter and leave the system. The average number of concurrent users can be defined as the number of unique users during an interval divided by the average percentage of the interval that the users are present (refer to [Wil11b]). For example, if all unique users are present for the entire interval, the number of concurrent users is equal to the number of unique users. If the users are present for half of the interval, then the number of concurrent users is half of the number of unique users. Note that using the average has all of the dangers discussed in [Wil10d]. A partly open system is an open system where users arrive, stay for a while like in a closed system, and then depart (refer to [SWHB06]). The number of concurrent users in a partly open system is similar to the number in an open system.

2.2 Sample Data

So, let's look at some operations data using different intervals, so that we can better appreciate the data. We gathered data from the system transaction logs for a one month period. There were 5,341 unique users present that month. But, how many users were there during a day, an hour, or a minute?

Six data sets were created from the gathered data. Table 1 shows some statistics for the data sets. The first data set, "All Days", contains counts of the unique users for every day of the month. A second data set, "All Hours", contains counts of the unique users for every hour of every day of the month. A third data set, "All Minutes", contains counts of the unique users for every minute of every hour of every day of the month. For each of these data sets, a subset is defined ("Work Days", "Work Day Hours", and "Work Day Minutes", respectively). For the three subsets, data corresponding to weekends are removed. For the hours and minutes subsets, only data between the [7:00,17:00) interval are included. We have constrained these subsets intentionally. For each data set, statistical measures like those in Table 1 can be reported. But are they appropriate?

Table 1: User Data Set Statistics

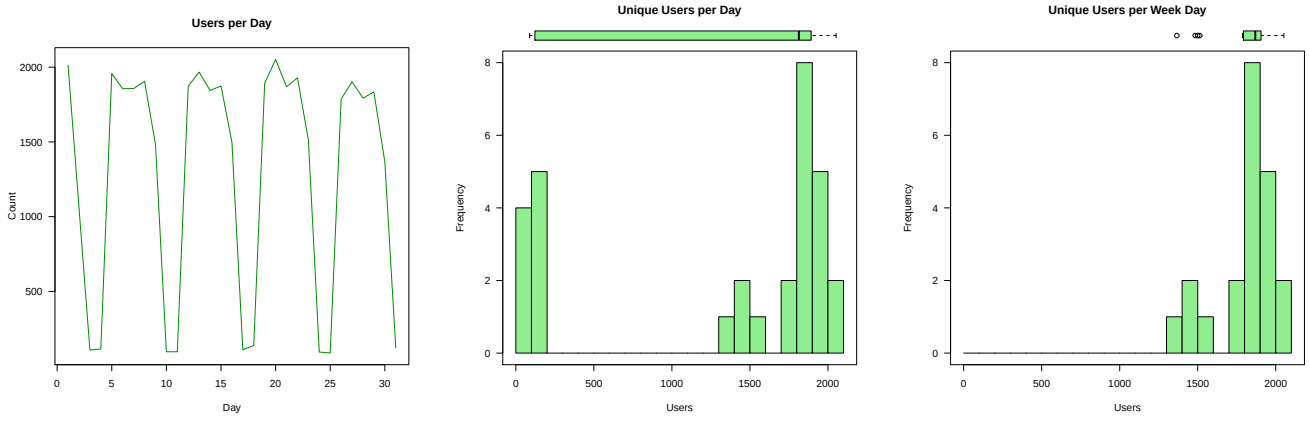
Data	Size	Min	Q_1	Q_2	Q_3	Max	Mean
All Days	30	88	126	1,301	1,888	2,052	1,814
Work Days	21	1,366	1,793	1,813	1,906	2,052	1,869
All Hours	720	0	5	163	218	779	17
Work Day Hours	210	31	448	509	654	779	549
All Minutes	43,200	0	1	23	36	180	3
Work Day Minutes	12,600	0	50	73	98	180	80

[Wil10d] provides an overview of various statistics (e.g., the quartiles Q_1, Q_2, and Q_3). It also discusses in what situations they are appropriate. Most summary statistics are not appropriate when data are multimodal. The summary statistics themselves do not reveal modality. The distribution of the data should be viewed graphically.

Figure 1(a) shows the number of unique users for each day of the month. The data are definitely sensitive to the day of the week. Based on the users-per-day, there are drastic differences in load. Figure 1(b) shows a histogram of the "All Days" data. It is clear that the distribution is multimodal, and so the data are not best summarized with common statistical measures. The boxplot above the histogram is also deceiving. Figure 1(c) shows a histogram of the "Work Days" data. While this distribution is also multimodal, summarizing the data using common statistical measures is not as offensive as it is for the "All Days" data. The boxplot more accurately represents these data.

Figure 2(a) shows the users per hour for the two most heavily loaded days. The two graphs are similar; other normal weekdays also have a similar shape as demonstrated in [Wil10a]. Figure 2(b) shows a histogram of the "All Hours" data. In this case, all data are accounted for (in contrast to Figure 2(a), where only two days are shown). Figure 2(c) shows

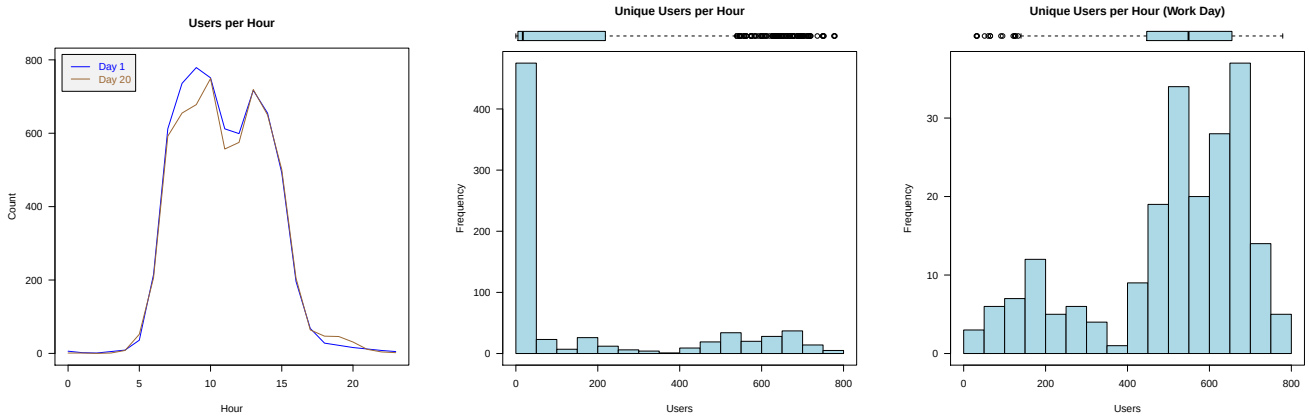
[3] This assumes that the user can login more than once concurrently. A single user cannot drive multiple sessions in the same way that multiple users can drive single sessions.



(a) **(b)** **(c)**

Figure 1: Chart **(a)** shows the number of unique users for each day of the month. Chart **(b)** shows the distribution of the "All Days" data. Chart **(c)** shows the distribution of the "Work Days" data.

a histogram of the "Work Day Hours" data. In both cases, we do have multimodal data and should not summarize the data using common statistical measures.



(a) **(b)** **(c)**

Figure 2: Chart **(a)** shows the number of unique users for each hour for two specific days. Chart **(b)** shows the distribution of the "All Hours" data. Chart **(c)** shows the distribution of the "Work Day Hours" data. Note the difference in y-axes on the histograms.

Figure 3(a) shows the number of users for two specific hours. One looks fairly uniform, showing little variation; the other is very erratic. Figure 3(b) shows a histogram of the "All Minutes" data. All data are accounted for (in contrast to Figure 3(a)). The distribution is skewed and barely bimodal. Figure 3(c) shows a histogram of the "Work Day Minutes" data. The distribution is unimodal and slightly skewed.

The erratic nature of the "Day 1 8:00" data in Figure 3(a) alerts us to the dangers of averaging. When the number of users spikes to larger values, contention for resources may cause response times to become non-linearly related. We may want our tests to encounter such loads. The statistics in Table 1 are sufficient for us to detect this situation, but the visualization of the data is better.

What can we conclude from the analyses? The load is definitely time-sensitive. We could make further statements about the behavior on Fridays, weekends, and night time hours, but we will not do that here. By grouping the data into certain time groups, distributions within the groups are likely to be unimodal and summary statistics can be computed for the groups. The variation in the minute-interval data warns us of the possibility of much-higher-than-average loads that may cause contention for resources.

Since the system is a combination of an open system and a closed system, our counts of unique users are not really counts of concurrent users. Let's say we decide to create an hour-long test based on 800 users since that number is roughly

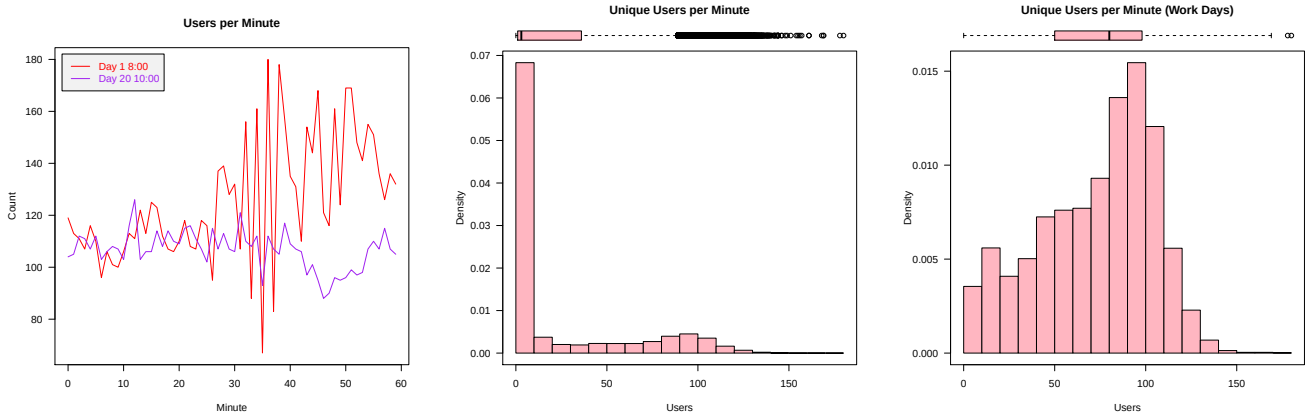


Figure 3: Chart **(a)** shows the number of unique users for each minute for two specific hours of the month. Chart **(b)** shows the distribution of the "All Minutes" data. Chart **(c)** shows the distribution of the "Work Day Minutes" data. Note the difference in y-axes on the histograms.

the maximum for any given hour. Is our test too heavy, too light, or just right? Existing analysis shows that it is far too heavy (refer to [Wil11b]).

Figure 4 shows how the unique users for an hour are not necessarily present for the whole hour. For any given user, a horizontal line is drawn to represent the interval that a transaction covers. Numerous lines show the user's activity; horizontal gaps between lines show his idleness (or absence, if he logged off). A user's session duration should be a factor in determining the equivalent number of test users. The session duration can be misleading when it contains idle periods that are not really think times.

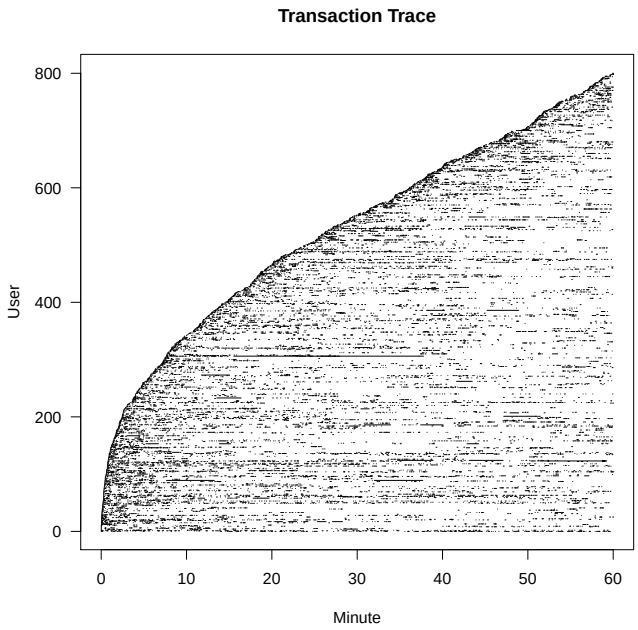


Figure 4: This chart plots each user's activities during the hour interval. The users are ordered by the start of their first transactions within the interval. What is not so obvious is when some users' last transactions occurred. While there are 801 unique users, there is only an average of 257 concurrent users.

In [Wil10b], we assigned hourly data to groups based upon the number of transactions present. The result was five groups with roughly the same total number of transactions. Each group can have an associated number of users that

generates the group's load. Table 2(a) defines what user loads are assigned to each load group. A test is executed for each group with a load equal to that defined by the group. For each test, a percentage of high response times can be removed (the given percentages were established in [Wil10b] and are only used here notionally). The group response times are averaged and the result is an estimated average response time for the charge period.

Table 2: User Load Profiles

(a) SLA Charge Period

Group	Users	Rem'd
5	300	3.1%
4	275	2.8%
3	225	1.6%
2	175	1.7%
1	50	0.7%

(b) Lifecycle

Year	Users
1	300
2	1000
3	1700
4	2400
5	3000

In [Wil11a], we discussed how the user base grows over the operations lifecycle. The load profile previously discussed needs to be scaled. Table 2(b) shows a possible maximum user load for each year of the operations lifecycle. In this case, the values are notional since no such data exist. For each entry in Table 2(b), a separate "Users" column in Table 2(a) would exist (only the users for year 1 are shown in Table 2(a)). This approach will be further detailed in the series' final paper.

The SLA calls for 3000 concurrent users. We have noted that is likely to be a number at the end of the lifecycle and should be scaled down for earlier years. However, should we consider that this number can really be exceeded? For a closed system like the performance test, we can let this be the maximum number of users. But for a partly open system, is this an average that can be exceeded for short periods? Because we are dealing with an SLA and not an engineering requirement, the risk should be assessed and appropriately handled (we probably do not care since averaging is done across the evaluation period and badly performing transactions are discarded). Accepting the SLA penalties is always an option.

3 Conclusion

This paper discussed how user load varies during the SLA charge period. This was very similar to the analysis presented in an earlier installment that looked at transaction load during the same period. This is the foundation for creating tests that will model the different parts of the charge period. The user load from the existing (partly open) system must be converted to user loads in the (closed) test system. Then we included the growth in user load that will occur over the system's operations lifecycle. Nonetheless, the number of concurrent users remains a vague term because of the lack of detail throughout the SLA definition.

Bibliography

[SWHB06] Bianca Schroeder, Adam Wierman, and Mor Harchol-Balter. "Open Versus Closed: A Cautionary Tale". In *Networked Systems Design and Implementation '06*, pages 239–252, 2006. Also available from `http://www.cs.caltech.edu/~adamw/papers/openvsclosed.pdf`.

[Wil10a] Tom Wilson. "Data Mining User Behavior". *CMG MeasureIT*, September 2010.

[Wil10b] Tom Wilson. "Developing Toward an SLA: Understanding the Testing Interval". *CMG MeasureIT*, October 2010.

[Wil10c] Tom Wilson. "Developing Toward an SLA: Understanding Transaction Performance". *CMG MeasureIT*, July 2010.

[Wil10d] Tom Wilson. "Statistics for the Performance Analyst". *CMG MeasureIT*, November 2010.

[Wil10e] Tom Wilson. "Workload Correlation and Visualization". *Proceedings of the CMG 2010 International Conference*, December 2010.

[Wil11a] Tom Wilson. "Developing Toward an SLA: Understanding Data Complexity". *CMG MeasureIT*, April 2011.

[Wil11b] Tom Wilson. "What Were They Thinking: Modeling Think Times for Performance Testing". *CMG MeasureIT*, March 2011.