Interpreting Load-Runner Analysis time series graphs

Introduction The LoadRunner Analysis tool can display data points from a run as a time series graph, for example *Average Transaction Response Time*. These graphs are one of the methods the Analysis tool uses to summarise the vast amount of data collected during a test run. Creating a summary in a standard form, the graph, removes the need to plough through all of the raw data.

Their role is similar to that of a set of company accounts. The accounts present a summary of the financial transactions the company has been involved in and of the financial state of the company. People do not have to read all of the company records to establish what is happening to the company. In a similar way the graphs reduce the need to read the raw data.

However a user of company accounts must understand their limitations. Understand what they do and don't show and how they can mislead as well as inform. Understand when it is necessary to look at the underlying data to get the true picture. Similarly it is necessary to understand how the LoadRunner time series graphs behave in order to interpret them correctly. This article discusses the issues involved.

The Items on Display When a time series graph is selected the graph appears in the main window. The X axis represents time and the Y axis the values. Information is displayed as a line graph, a set of linked data points. In addition to this main window the lower window contains multiple sheets for *Legend*, *Graph Details*, *User Notes*, *Graph Data* and *Raw Data*.

The *Legend* sheet gives minimum, average, maximum, median and standard deviation figures for each plot. The *Graph Data* sheet contains the graph data points. The *Raw Data* sheet is an optional data set and contains the actual values captured during the run.

Getting the Wrong Impression from the Graphs

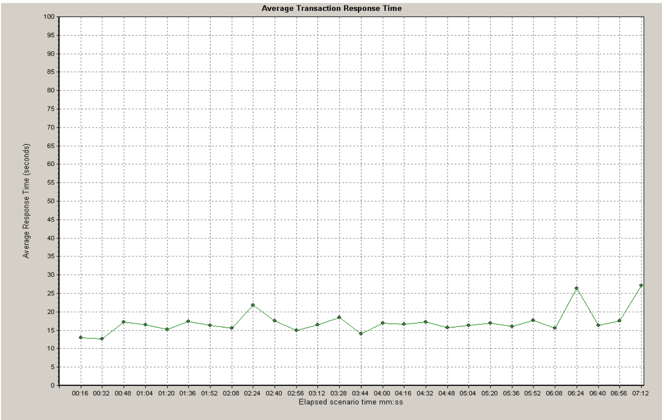
A quick glance at one of the graphs and the associated summary figures on the *Legend* sheet can be misleading. It is easy to understand why a user could take the maximum and minimum values as the limits of the value being plotted. It is even more understandable that a user would assume the average on the *Legend* sheet is the average value of the item for the period on the graph. Though understandable, interpreting the data in this way is wrong and can lead to the user getting an incorrect view of the behaviour of the system that is being monitored.

What is Actually Being Displayed? The first step when the Analysis Tool time series graphs are being prepared is to generate the values that can be seen in the *Graph Data* sheet. This is done by dividing the graph time span into slots and taking the mean of the raw values falling within the slot as the *Graph Data* value for that slot. Then it is the *Graph Data* that is plotted on the graph. Also the maximum, average, minimum, median and standard deviation figures are not calculated from the raw data values, instead they are based on *Graph Data* values.

The Effects of Granularity The duration of the slots in the *Graph Data* is referred to as the *Granularity* of the graph. This can be set[1] to be a number of seconds, minutes or hours[2]. Shorter periods provide more detail, longer ones more of an overview. The choice of granularity has a very significant effect on the impression conveyed by the analysis graph.

Effect on the 'Appearance' of the Graph Two graphs produced for the same set of raw data values are shown. Figure 1 Shows the data with the granularity set to 16 seconds whilst figure 2 uses a setting of 1 second. Notice that the 16 second graph line is 'smoother', it is less 'spikey' than the 1 second graph. Also note that the range of the values has been reduced. The values for the 16 second graph are all closer to the mean value. The 'appearance' of the 16 second graph is of fairly consistent data with minor variations around the mean value. The 'appearance' of the 1 second graph is of highly variable values with very large values appearing occasionally.

Figure 1 Granularity of 16 seconds

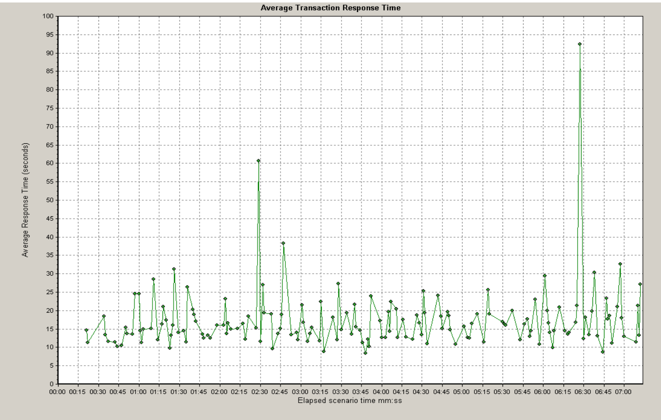


1. The Granularity is set from the View menu or by right clicking on the graph.

2. The user interface drop down list also offers milliseconds but if this is selected a message is displayed indicating that millisecond granularity is not supported.

sqc
www.sqc.co.uk
SQC Technology Ltd.
Coton Park House. Linton. Swadlincote.
DE12 6RA. UK.

Figure 2 Granularity of 1 second



Effect on Maximum and Minimum Values

Table 1 shows the summary values from the *Legend* sheet at different granularity durations for the data set used for figures 1 and 2. Notice the differences in the minimum and maximum values for the different settings. As the period gets larger more raw values are included in the calculation for the period. The values in a period become more representative of the overall population and so the values tend towards the mean. This masks the effects of occasional extreme values. As a result the minimum value that occurs for a period increases and maximum value that occurs for a period decreases[1].

Table 1 Data set 1 analysis with varying granularity

Granularity	Min	Av	Max	Median
1sec	8.366	17.321	92.384	15.328
2sec	8.366	17.381	92.384	15.677
4sec	8.838	16.944	43.490	15.869
5sec	9.861	16.779	56.885	16.181
6sec	10.714	16.828	43.490	15.977
16sec	12.695	17.165	27.152	16.524
1min	14.750	17.096	20.264	16.683

Effect on Average Value

The average values in table 1 are effectively the same and they do correspond to the mean of the raw data values which has a value of 17.161. So is the average value always consistent and does it always match the mean of the raw data values?

Under certain circumstances the average value can, if used in isolation, be misleading. The value can be significantly different to the mean of the raw data values. This arises because the average is a straight forward mean of the values for each slot. The calculation of the average takes no account of the number of values that are actually present in each slot, there is no weighting.

This issue is best illustrated by an example. Table 2 shows the characteristics of the response times for a set of transactions. Transactions that use cached data form 42% of the data set whilst the other 58% have to access a data server. Table 3 shows the summary values from the *Legend* sheet when this data is graphed with the granularity set at a periods from 1 second to 5 minutes.

Table 2 Data set 2 raw data characteristics

Data Set	Min	Mean	Max	Median
Raw-Cached	0.189	0.200	0.272	0.197
Raw-Server	0.920	29.245	57.839	29.096
Raw-Combined	0.189	17.143	57.839	9.061

Table 3 Data set 2 analysis with varying granularity

Granularity	Min	Av	Max	Median
1sec	0.189	4.764	41.747	0.199
2sec	0.189	3.186	29.913	0.198
10sec	0.189	3.377	27.654	0.198
20sec	0.192	5.848	24.603	0.199
1min	0.193	13.632	20.001	19.162
5min	16.104	17.058	18.012	18.012

Why can the average values in table 3 for the 1-20 sec granularities be so far from the mean value of the raw data? It is because in this test run the server responses are bunching. Server transactions are completing in groups whereas cache transactions are spread evenly. The result is a large proportion of the data points in *Graph Data* only contain cached transaction times. The averaging process does not weight *Graph Data* points according to the number of transactions they contain so the average can skew towards the average for cached transactions.

Bunching can occur because of problems with the system or interactions between the Vusers and the system. Systems can go into cycles where a backlog of work builds up in the system with no responses being supplied by the system. Then at some point in the cycle, perhaps as something clears, a large set of responses are delivered in a short period.

Given the possibility of bunching it is not safe to assume that data values that are captured when an event occurs[2] are distributed uniformly over time. The average values needs to be treated with caution and should not be used or reported in isolation. The LoadRunner transaction duration values are an example of this, each value is collected when its transaction completes.

Opposite Effects

It can be seen from the data in table 3 that the average moves closer to the mean value of the raw data as the granularity time increases. This is the reverse of what happens to the maximum / minimum values which converge with the raw data as the granularity time decreases.

1. At the limit if all of the raw data samples fall within one slot then the slot value is the mean value of the raw data.

2. The other method for collecting data is to sample a value on a regular periodic basis. The distribution of data that is collected in this way is independent of the behaviour of the system and so is not vulnerable to bunching effects.

Impact on User Perception

So how do these effects influence a user's perception? Take, for example, the *Transaction Response Time* graph. The smoothing effect may mask sporadic responses that take far longer than the typical responses. If a user looks at data via a graph that 'smooths' the data they may get one impression. However if they were to look at the same data set via a graph that does not 'smooth' the data their impression may be very different. A 'smooth' graph may give the user the impression that all transactions have similar response times and these are acceptable whereas, in reality, there are transactions with problems. The 'spikey' graph would reveal the problems.

The smoothing effect can also make two sets of results look similar when the behaviour that occurred was significantly different. Suppose a system has been modified and a regression test has been run. Graphs for the two runs are put side by side, the minimum, average and maximum figures are effectively the same and the general appearance is the same so this seems to indicate that the change has affected the system. Is this valid? Well not necessarily, the smoothing may be masking variation within a period. Occasional anomalies may be occurring or priority may be being given to transactions from source A which receives very fast responses, whilst source B is getting slow responses. Observing the data without the smoothing effect would reveal this.

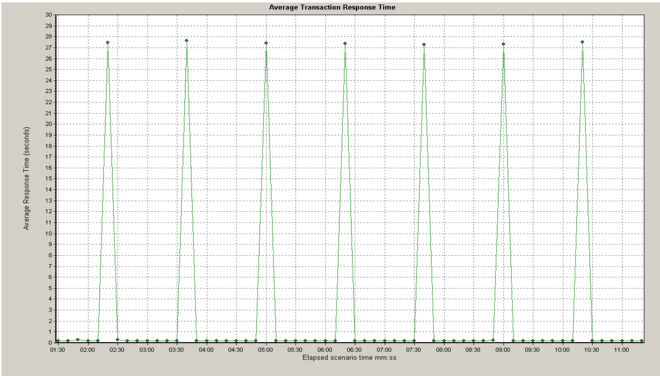
Useful information can be obtained from the 'appearance' of the graphs but users should not rely on a single granularity. A number of settings should be used to obtain both a smoothed view of the overall trend and a more detailed view of the distribution of the values.

Obviously a user's perception of the system will also be influenced by the summary information, maximum, average, minimum and median displayed on the *Legend* sheet. A user could take the maximum response time for the graph as being at or close to the maximum value that occurred in the test. The average value could be taken as the general response time users will experience. Given the characteristics of these values this can lead to an erroneous interpretation of the results.

Considering the results for data set 1 first. If a user were to graph this using a 16 second granularity and then review the graph and the *Legend* sheet values (see figure 1 and table 1) then what impression could they get? The impression could be of consistent values averaging around 17 with a worst case value around 27. Whilst the average is correct the impression of stability and of the worst cases is not. The real worst case value is over three time higher.

For data set 2 at 10 second granularity (see figure 3 and table 3) the impression could be of transactions with a typical response time around 3 seconds and a worst case around 28. The impression could be that there must be a big bunch of transactions experiencing extended, but still single figure, response times every so often with occasionally ones approaching the 28 second mark. These impressions, though understandable, are wrong, the data server transaction response time is being hidden.

Figure 3 Data set 2 with 10 second granularity



Avoiding the Wrong Impression

The time series graphs produced by the LoadRunner Analysis tool provide an extremely powerful and useful capability. It is when they are used in isolation or with just one granularity that they can give a user the wrong impression. To avoid this pitfall it is advisable to adopt a number of practices when using the Analysis tool.

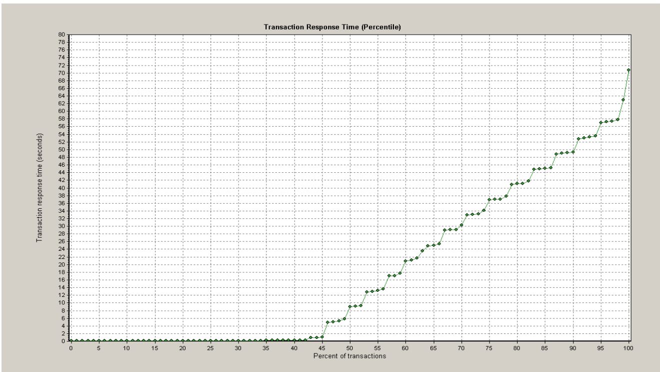
Practice-1 Get an overview of the data. Graph the data with a granularity that provides a good overview of the medium term pattern of behaviour. Don't take too wide a view or else long term trends that are developing, for example a gradual increase in response time, may be masked.

Practice-2 Look at the data in more detail. Analyse the data with a granularity that ensures a small number of data values per slot. This provides a detailed view of the data. It reveals short term patterns in the data. It provides a better indication of the extreme values and of the frequency with which they occur.

Where the Analysis tool can directly report mean, minimum and maximum based on the raw data ensure this information is reviewed. This is mainly applicable to the transaction response time where this data is available in the *Summary* report and in the *Transaction Performance Summary* graph. This information is not directly available for other values like transactions per second.

Practice-3 Do not use the time-series analysis on its own. Where the Analysis tool allows data to be presented in additional ways use them. Do not just rely on the time-series presentation of the data. For example Figure 4 Shows the percentages of transactions under a given response time for data set 2. It indicates that something untoward is happening.

Figure 4 Percentile Distribution of data set 2



Practice-4 Check the raw data for extreme values. The raw data for a graph can be displayed in the Raw Data sheet of the analysis tool. It is updated to match the graph settings[1] when the sheet is first selected and by using the *View Raw Data* command[2]. The raw data should be exported as an Excel spreadsheet[3]. Excel can then be used to check for extreme values.

Practice-5 Establish, Document and Be Consistent. A conscious effort should be made to establish the analysis to be performed for each test set. This should not simply be the default or whatever springs to mind when the tests are run. Getting this right may require some experimentation. When it is right it should be documented and an Analysis tool *Template* created to act as a starting point for the analysis.

The analysis and reporting should be applied consistently across test runs and on different versions of the system. The approach should evolve with experience. When something is missed the analysis should be refined to make certain it is not missed next time.

Practice-6 Transition Periods. Determine how to deal with transition periods like scenario ramp-up and ramp-down periods. Should data from these periods be excluded from the analysis? Decide what is required and document this.

Practice-7 Post Test Reviews. An informed review of the analysis generated by LoadRunner Analysis tool should be a formal activity within the testing process. It doesn't have to be a meeting, it could be a review by one person, but it should generate a record.

Ideally a checklist / data collection form should be used to guide the review. The questions on the checklist and data to be completed on the form should be designed to ensure the reviewer(s) study the analysis and if necessary the raw data and understand what is going on.

Generating the review record should require thought and hopefully some work with a calculator. It should definitely not be possible to complete the record by tickling boxes and copying summary information from the LoadRunner report.

Practice-8 Detailed Analysis. When anomalies are suspected or the 1 second granularity can not provide enough detail resort to analysing the raw data. Export it to Excel and investigate.

LoadRunner Version

This article is based on LoadRunner 7.6 Analysis tool build number 7.6.0.5.

1. The data is not 'live' it does not automatically change if the graph settings are changed.

2. From the View menu or by right clicking on the graph. Note that the data is first filtered by the graph settings and then by the period specified in the raw data command.

3. Right click on the *Raw Data* sheet.