

# Oracle Database Capacity Analysis Using Statistical Methods

Ajith Narayanan  
Dell  
Bangalore, India

**Keywords:** Oracle, Database, Capacity Planning, Performance Tuning, Statistics, Matrices

## Introduction

Why is my database slow? I bet this would be the most common question any Oracle DBA would have come across in his/her career and end of the day the decision is made to go for a database infrastructure capacity increase. It's a myth that a server capacity increase in terms of CPU & RAM is the solution for all performance issues, but if it is really a CPU or IO bound system we need to prove it with numbers. This attempt would be to crack the myth and share my experience and knowledge on analyzing the mostly unexplored area of capacity planning statistical methods, which can be used proactively to avoid the potential problems immediately or in future.

## Database Server, Are you being served properly?

The database server would have been provisioned in the implementation stage of the project. As, the time goes by the application workload increases and remains unnoticed and that is where the actual performance starts due to lack of awareness of our own application workload trend.



*Illustration. 1: A server fully loaded with work, refuses to take more orders from clients*

Are you sure database is the problem center always?

In case there is no significant workload increase, there are high chances of problems with applications that would be vulnerable to memory leaks due to faulty designs, such cases remains untraceable or even the SQL's taking a bad execution plan. But in any case the immediate finger pointing happens towards the infrastructure like processing power (CPU) and Memory (RAM).

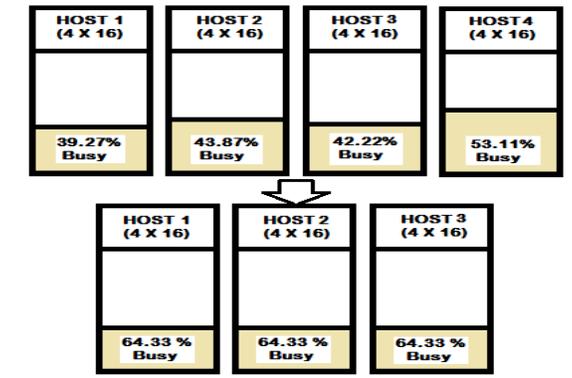
Do you understand you system workload fully?

In most cases the applications that connect to database are complex in terms of workload and there people who believe performance tuning is best performed by starting at the "system" level and

moving down through the layers to find the bottleneck(s). So to do so, we need to be fully aware of our system workload and its trend.

**What Are The Different Statistical Methods Of Analyzing The System Capacity Database?**

Simple Maths - This model can take single component inputs either application or technical metrics. This method is usually involved with short-duration projects. The precision is usually low, but sufficient when used appropriately.



*Illustration. 2: A server consolidation result using Simple Maths Method.*

Essential Forecasting Mathematics – This method can produce relatively precise forecasts. This is again a single component input method using technical metrics. This method is used with short-duration projects.

Linear Regression Analysis – This method is typically used to determine how much of some business activity can occur before the system runs out of gas. Linear regression analysis is a method for investigating relationships among variables. E.g. Logical Reads vs. CPU utilization

Correlation formula used in linear regression model:

CPU utilization = user calls \* m + c  
 Corrélation coefficient  $r = \frac{\sum(Y_i - Y)(X_i - X)}{SQRT(\sum(X_i - X)^2 \sum(Y_i - Y)^2)}$

Correlation Coefficient (r)	Practical meaning
=====	=====
0.0 - 0.2	Very weak
0.2 - 0.4	Weak
0.4 - 0.7	Moderate
0.7 - 0.9	Strong
0.9 - 1.0	Very Strong

Note: - MS Excel provides predefined function CORREL() for such complex mathematical equation (Good News!).

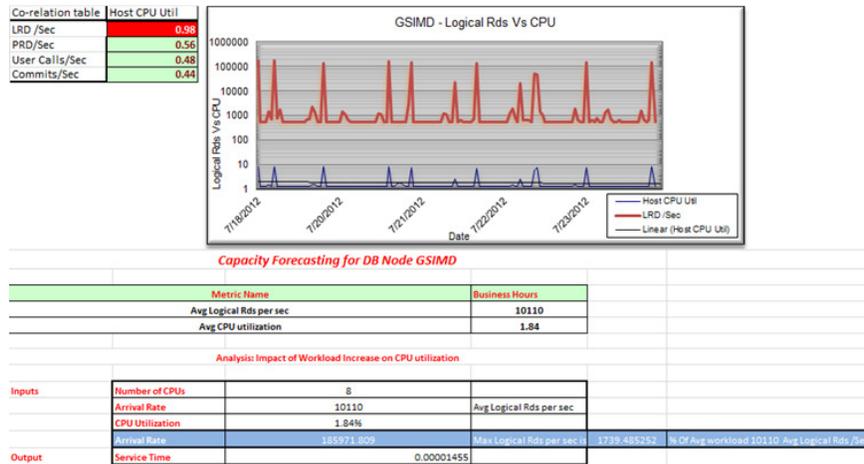


Illustration. 3 An Excel template for linear regression modelling.

Queuing Theory – This is basically an upgrade to essential forecasting mathematics, this method is used for high precision forecasting.

Agner Krarup Erlang(1878 -1929) is the man behind this math. He studied the performance of telephone networks. When Erlang C function is used, we do not apply the essential forecasting response time formulas, instead, we have a single new queue time formula that can be applied to both CPU & I/O subsystem.

For CPU subsystems, there is only one queue, so the entire system arrival rate is ( $\lambda_{sys}$ ), but for I/O subsystems, the arrival rate at each queue ( $\lambda_q$ ) is the system arrival rate ( $\lambda_{sys}$ ) divided by the number of I/O devices.

Correlation formula used in queuing model:

$$U = \frac{St \lambda_q}{m}$$

$$Q = \lambda_q Q_t$$

$$E_c = \text{Erlang}(m, St, \lambda_q)$$

$$Q_t = \frac{E_c St}{m(1-U)}$$

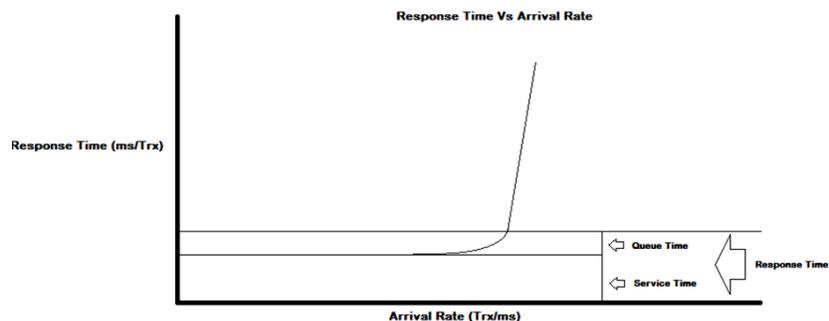


Illustration. 4 System response time Vs. System arrival rate.

Gathered Workload Data										Conversions	
Date	Time	System Arrival Rate (sys_lambda)	Queue Arrival Rate (lambda)	Service Time (Ts)	Queue Time (Tw)	Utilization % (Util)	Response Time (Tq)	Servers	ErlangC	Random#	Actual Workload UserCalls/30Secs
6/11/2012	6:30:15 AM	2.176966667	2.176966667	6.291322789	1.23041	0.856	7.521728	16	0.450597414	0.6750285	65309
6/11/2012	6:30:45 AM	2.178566667	2.178566667	6.272748137	1.19589	0.8541	7.468634	16	0.445048296	0.2761636	65357
6/11/2012	6:31:15 AM	0.000266667	0.000266667	51402	10148	0.8567	61549.97	16	0.452652745	0.8039326	8

Workload Baseline	
Peak workload (90%)	71214.3
Peak workload (95%)	71496.45

(1) RAW DATA:	
Dump your raw data in this worksheet. You'll have atleast 120 rows after the data gathering for 1 hr steady state duration; The first 60 rows of data is used to develop the forecast model and the next 60 rows for model validation.	

(3) Analysis Model:													
This page computes the weighted average of service time and queue time and gives the residual error by comparing it with actual service time and queue time.	<table border="1"> <thead> <tr> <th>Metrics</th> <th>Weighted Avg</th> <th>Average</th> <th>Std Dev</th> </tr> </thead> <tbody> <tr> <td>Service Time</td> <td>0.06839203</td> <td>25.61537</td> <td>78.61813</td> </tr> <tr> <td>Queue Time</td> <td>1.304223405</td> <td>485.547</td> <td>1345.278</td> </tr> </tbody> </table>	Metrics	Weighted Avg	Average	Std Dev	Service Time	0.06839203	25.61537	78.61813	Queue Time	1.304223405	485.547	1345.278
Metrics	Weighted Avg	Average	Std Dev										
Service Time	0.06839203	25.61537	78.61813										
Queue Time	1.304223405	485.547	1345.278										

Gathered Workload Data								Residual Error Finding				
Date	Time	System Arrival Rate (sys_lambda)	Server Arrival Rate (lambda)	Service Time (Ts)	Queue Time (Tw)	Utilization % (Util)	Response Time (Tq)	Servers	Residual (Derived Ts)	Residual^2(Ts)	Residual (Derived Tw)	Residual^2(Tw)
6/11/2012	7:19:16 AM	0.001033	0.001033	133.8735464	2943.051321	0.86	16330.41	16	133.8051564	17903.81987	2941.747096	8653875.989
6/11/2012	6:38:45 AM	2.204533	2.204533	0.062779727	1.387759108	0.87	7.665732	16	-0.005612304	3.1498E-05	0.083535703	0.006979214
6/11/2012	6:43:16 AM	0.000967	0.000967	141.897931	2824.156269	0.86	17013.95	16	141.829539	20115.61813	2822.852046	7968493.674
6/11/2012	7:19:46 AM	2.330567	2.330567	0.058460217	1.334480847	0.871	7.288021	16	0.008931813	7.9777E-05	0.030267442	0.00015811

Illustration. 5 An Excel template for Queuing modelling.

## Conclusion

This is an introduction to few statistical methods and the models that can be used to replicate our database systems workload trend. This capacity planning or workload measurement when done proactively on a regular basis can predict an unscheduled outage due to shortage of system resources like CPU or RAM.

## References

Forecasting Oracle Performance by Craig Shallahamer (Apress, 2007)

## Contact address:

**Ajith Narayanan**

Dell IT

Bagmane Tech Park Byra Sundra,

C V Raman Nagar Bagmane Parin Katha

Bangalore South Taluk

Bangalore INDIA-560093

Phone:

+91 9008488882

Email

ajith\_narayanan@dell.com

Internet:

http://oracledbascrptfromajith.blogspot.com