

# Performance Tuning of Storage System using Design of Experiments

---

WHITE PAPER

**Author: Prabu D, Devi Prasad B, Lakshminarayana Prasad K.**

---

## Abstract

This paper discusses the suitability of the Design of Experiments (DOE) methodology for storage performance tuning. The DOE estimates the main and interaction effects of three major storage factors viz, Cache Partition Size (CPS), LUN Strip Size (LSS) and Cache Segment Size (CSS) settings of an enterprise level storage system on the overall storage performance. Other factors influencing performance are kept constant throughout the experiment to minimize experiment duration and complexity. DOE has provided key ideas for effectively analyzing the relationship of storage factors in storage performance tuning.



## Introduction

The Storage System is a key component in Storage Area Networks (SAN) [BARK2001]. Applications, Servers, Host Bus Adapters (HBA), SAN Switches and Storage Systems are the other essential components in a SAN. These components individually and collectively contribute to overall SAN performance. Performance bottlenecks can be studied at each component level or at an overall SAN level depending on the need. For this paper we have studied the SANs performance at the storage system level with other components kept constant. Within the storage system, there are several factors affecting performance. The objective of this experiment is to understand the relationship among three factors viz: CPS, CSS and LSS with regard to how they affect performance individually and through interaction. DOE is a powerful statistical methodology, often applied to manufacturing, agriculture, psychology and clinical medicine to study and model processes. This technique has not been widely used in the area of computer performance testing [MART1981]. To the best of our knowledge in the research literature on DOE methodology, it has not been adopted for performance testing in the storage domain. The application of DOE methodology has potential use in performance testing and tuning of a SAN. Performance testing and tuning efforts are intensive and block expensive resources for long durations. Using DOE we can reduce experimental runs, and the resultant model can be used to select optimal values for performance tuning. The objective of this experiment is to study the suitability of DOE methods for SAN performance testing also. The white paper discusses:

- An overview of SAN and storage system performance
- Design of experiments
- Details of experimental work
- Interpretation of the experimental results
- Conclusions and future work

## SAN and Storage System Performance

The purpose of Storage Area Networks is to interconnect various components involved in data exchange between the user (application) and the provider (storage system). In this experience, we have attempted to study how to optimize the storage system's performance by tuning the CPS, LSS and CSS for a given application profile. At the outset, we present terms which are used in document:

- Cache Partition Size (CPS): The storage system employed provides a feature which enables the administrator to reserve a part of available cache exclusively for an application. This exclusive cache is called a cache partition. An exclusive cache partition will increase cache hits and thus improve performance.
- Cache Segment Size (CSS): The Cache Segment Size is the unit page of the cache memory configured at the storage cache level. When configured in conjunction with the application block size, CSS can reduce the fragmentation of the available cache for the application and hence improve the over all throughput.
- LUN Stripe Size (LSS): LUN (Logical UNit) is the organization of physical devices into logical units.

- LUNs are used by hosts to access the logical device. "LSS is the amount of data written on a disk before the successive data is written to the next disk, while striping data" [HUSE2003]. Again when configured in conjunction with application block size, LSS reduces the number of disk reads / writes to complete the I/O cycle thus improving performance.
- Application Profile: Typically, performance testing would be carried out using work load generators that simulate the target application profile. For the current experiment we have chosen an OLTP (OnLine Transaction Processing) application profile [SHEK1990] which is characterized as short random IOs from multiple concurrent users. In this experience we have kept the application profile the same as given in Table 1. We have used IOmeter [INTE1998] for simulating and measuring the raw performance of the data storage system. Using IOmeter, a user can easily simulate realistic workloads and produce detailed profiling data for the storage system.

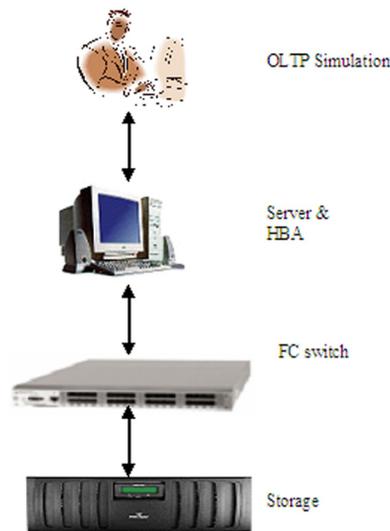


Figure 1. Experimental Setup

Figure 1 provides a high level data flow through various components of SAN. Performance of the SAN is quantified by how much data it can provide to an application in MBPS (Mega Bytes per Second), how many concurrent data requests it can handle in IOPS (Input Output Per Second) and how fast it can process the requests i.e. latency in seconds. Each of the components of a SAN setup is shown in Figure 1(Page no 4) and contributes to the overall performance of the SAN. Apart from the above, technologies employed in each component like protocols [CLAR2001], RAID [CHEN1996], Disks [LEE1993], Cache [ARI2004], etc. determine the overall performance of the individual component. The overall performance of the SAN would be equal to the weakest link in Figure 1. It is essential to study the performance capabilities of each component and develop methods to optimize it for maximum performance.

The remainder of the components are kept constant to keep the experiment simple.

## An Overview of Design of Experiments

Design of Experiments (DOE) [MONT1992] [ANTOI1998] is a structured method for conducting experiments and analyzing the results. This method was first developed in the 1920s and 1930s, by Sir Ronald A. Fisher [SIRR1920], the renowned mathematician and geneticist. There have also been many other contributors [ASQ1993] to the DOE theory viz, George Box, Dorian Shainan and Genichi Taguchi and others. DOE is also called statistical experimental design; it is a tool for determining the main and interaction

effects of different factors affecting process quality and for calculating optimal setting for controllable factors. In order to understand the main and interaction effects very clearly, let us consider a real life example. We know attitude and aptitude individually affects the rate of success of a person. Experimenters, who are studying how aptitude and attitude influence behaviour, should look to not only how attitude and aptitude determine the success of individual but how they interact with each other. This will give complete information to enable the experimenter to predict the optimal level for a guaranteed success rate. Generally resources are limited for experimentation, so it is very important to obtain the most information from each experiment we perform.

A system's or process's performance depends on a certain set of factors. These factors could be set at a different "Level" which alters the performance. To find optimal levels for each factor, experiments should be carried out by changing one factor at a time keeping the rest constant: this is also called a full factorial design. For example, in an experiment with 5 factors each with 2 levels 32 experimental runs would be required to gather data to find the optimal level for each factor. When factors and their levels increase or the duration of each experimental run is long or the cost of conducting each experimental run is high, the experimenter is forced to select fewer runs. This approach is called fractional factorial design. DOE provides suitable methods to select fewer runs such as Plackett Burman [PLAC1946], Box-Behenken [BOX1960], Taguchi designs [TAGU2004], etc. Fractional factorial designs can also be used to screen the vital few factors among all factors and hence are sometimes called screening designs. Well-designed experiments can produce significantly more information and often require fewer runs than fortuitous experiments. Thus, DOE provides an efficient means for determining the "Levels" of the "Factors" that would result in optimal performance of a process or system with fewer experiments. Experimental designs with many factors and with 2 levels are very common. They provide information of all main effects and first order interaction effects. But it is uncommon to find factors with 3 and 4 levels. In this case, one needs to study 2nd order and 3rd order interactions also. Any thing above 3rd order interaction is very complex and not worth studying for commercial experiments.

DOE uses General Linear Model (GLM) [DOBS1990] to analyze the results when factor levels are more than two. GLM is an advanced methodology for studying the correlation between multiple predictors (x) or factors influencing one or more outputs (y). GLM is an extension of multivariate regression. Multivariate regression can analyze the correlation between multiple factors and one output. For this it assumes several things about the base data, like normality, randomness and continuous data. GLM not only eliminates the need for these assumptions, but can also analyze the correlations between multiple factors influencing more than one output.

The mathematical model for GLM is given by:

$$Y = XB + E$$

where:

Y = matrix of n x m order

X = design matrix

B = matrix of coefficients

E = matrix of error terms.

Other Key Features of DOE are:

- Replication - where the experimenter takes multiple readings for the same experimental run at different times to reduce pure error
- Randomization - where the user runs the experimental trails in a random order to avoid the influence of previous runs
- Blocking - where the experimenter conducts the experiment in different blocks to reduce the systematic error due to time, environment, etc.

Minitab [MINI1972], a commercial statistical analysis tool was used in this work to design the experiment and the effect and interaction analysis among the factors. Minitab is one piece of software most often used in Six Sigma projects, a comprehensive, data-driven approach to defining and solving business problems and the quality improvement methodology.

### Advantages of DOE in Storage Performance Analysis

As discussed above, SAN performance depends on various factors. Better performance is achieved in SAN by tuning its individual components. Optimal values for each tunable factor should be obtained by conducting several experimental runs and analyzing the main effect and interaction effect of the factors. By its very nature, performance testing takes a long time and blocks valuable resources. In this context, DOE methodology can be used effectively in designing the performance testing experiments and analyzing the results. The data generated by the experiment can be used to build a mathematical model that can be used to predict the performance given the factor setting. These models would be handy for system administrators to modify the SAN settings to maximize performance.

There are many advantages [DOET1989] for using DOE to conduct performance analysis:

- Generate optimal number of experimental runs
- Study the behavior of a SAN performance over a wide range of operating conditions
- Minimize the effect of variation in testing conditions through replication, randomization and blocking
- Help us determine the important factors that need to be controlled through screening in an experiment where there are too many factors
- Help us measure interactions, which are much more important.

In this work, we used a DOE method to arrive at the optimal settings for three storage system parameters, namely: CPS, LSS and CSS to maximize the performance for a given application profile.

## Experimental Work

Figure 1 depicts the experimental SAN with a Server [FABB1998], Host Bus Adapter (HBA) [BARRY2000], Fibre Channel (FC) switch [MENG1994] and mid range storage system.

### Experimental SAN Setup

The description of storage server, HBA and storage system is shown in Table 2 and 3 respectively.

**Table 1. Shows the description of experimental application profile**

Application factors	Specifications
Application Block Size	8 KB
Read and Write Ratio	60 and 40 %
Application Type	Random
Percent of Random	100 %
Percent of Access	100 %
No of Outstanding I/Os	500
Burstiness	0
Run Time	3 Minutes
Ramp Up Time	60 Seconds

**Table 2. Shows the details of the Storage Server, HBA and SAN Switch**

Server	
Operating System	MS Windows Server Standard Edition
No of Processor	1
Type of Processor	Intel Xeon
Processor Speed	3.0 GHz
RAM	2 GB
HBA	
Speed	4Gbit/s
Queue depth used	32
Frame	2048
SAN Switch	
Speed	4Gbit/s
No of ports supported	16
No of ports used	2

**Table 3. Shows the description of the Storage System**

Storage	
Disk Drive Interface	Fibre Channel(FC)
Controller	Dual
Disk Drive Speed	15000 RPM
RAID Level	RAID-5
No of Disks/ Topology used	11D+1P/ loop
Disk Drive Total Capacity	730.5GB
LUN size used	10GB
Cache Memory	2 GB
Max/Min Cache partition in MB	1360/100
No of port used	1

We created a LUN of size 10GB in the storage subsystem to reduce the time preparing and configuring the disk. An OLTP application load was simulated using an IOmeter. A random application load, as per Table 1, was generated also by using the IOmeter tool.



## Factors and Levels

As briefed earlier in section 2, CPS, LSS and CSS were the factors we selected to tune the storage system Table 4 summarizes the factors and the levels.

**Table 4. Initial Factors and Levels**

CPS (MB)	LSS(KB)	CSS(KB)
100	64	8
600	256	16
1360	512	64

## Experimental Runs

As the number of factors and levels were low, we decided to use a full-factorial design to get the entire main and interaction effects of the factors. We took 3 replicates, i.e. three measurements for each unique run to reduce the pure error in the experiment. These runs were randomized to reduce the influence on the system state due to a previous run. As well, we decided to conduct the experiment in single block as the experimental duration was small. The initial design provided by Minitab [MINI1989] with single block and three replicates gave us 81 test runs. But we found out that certain combinations of CSS with 8kb size is infeasible for testing with 512 kb of LSS.

**Table 5. Corrected Factors and Levels**

CPS (MB)	LSS(KB)	CSS(KB)
100	64	16
600	256	64
1360	512	-

Owing to this condition, we later removed the 8 KB level of the CSS as shown in Table 5. Finally Minitab generated 54 test runs for the experiment as shown in Table 6.

## Measuring the Throughput

The storage system was configured for each experimental run setting the CPS, LSS and CSS as required. The IOs which simulated OLTP application was pumped using the IOMeter tool for a duration of 60 seconds for ramp up and three minutes to collect the throughput readings. The results are tabulated in Table 6.

**Table 6. Shows the final experimental runs designed using Minitab and the throughput recordings**

Trail no	CPS in MB	LSS in KB	CSS in KB	Throughput in MBPS	Fit Value
1	600	64	64	15.28	15.25
2	100	64	64	14.79	14.52
3	600	256	64	16.65	17.28
4	100	256	16	16.57	16.89
5	100	64	64	14.28	14.52
6	600	512	16	18.6	19.21
7	600	256	64	17.15	17.28
8	600	64	64	14.66	15.25
9	1360	256	64	18.08	18.45
10	600	64	16	18.11	18.12
11	100	64	16	16.39	15.82
12	100	512	64	14.69	15.20
13	1360	512	64	18.24	18.6
14	100	64	16	15.57	15.82
15	600	256	64	18.05	17.28
16	100	256	16	17.53	16.89
17	1360	256	16	21.37	21.39
18	100	256	16	16.58	16.89
19	100	64	64	14.5	14.52
20	1360	512	16	21.34	21.35
21	1360	64	64	17.52	17.41
22	1360	512	16	21.37	21.35
23	100	256	64	15.8	15.49
24	1360	512	64	18.7	18.60
25	600	512	64	17.2	16.64
26	1360	64	16	20.28	20.31
27	100	512	64	15.46	15.20
28	100	256	64	15.35	15.49
29	600	256	16	18.31	19.01
30	600	256	16	18.99	19.01
31	600	512	64	15.44	16.64
32	1360	64	16	20.28	20.31
33	1360	512	64	18.88	18.60
34	600	64	16	18.32	18.12
35	1360	64	16	20.37	20.31
36	100	64	16	15.5	15.82
37	1360	256	64	18.5	18.45
38	1360	64	64	17.45	17.41
39	600	256	16	19.73	19.01
40	100	512	16	16.29	16.78
41	1360	256	64	18.78	18.45
42	100	512	64	15.47	15.20
43	1360	256	16	21.39	21.39
44	600	512	16	19.52	19.21
45	600	64	16	17.94	18.12
46	100	512	16	17.06	16.78
47	1360	512	16	21.36	21.35
48	600	512	16	19.52	19.21
49	600	64	64	15.83	15.25
50	600	512	64	17.29	16.64
51	1360	64	64	17.26	17.41
52	1360	256	16	21.43	21.39
53	100	512	16	17.01	16.78
54	100	256	64	15.33	15.49



## Interpretation of Experimental Results

### Analysis of Data

The first step in analysis is to test the response data. For normality, distribution, run order and residual versus fitted value. The following graphs describe the data.

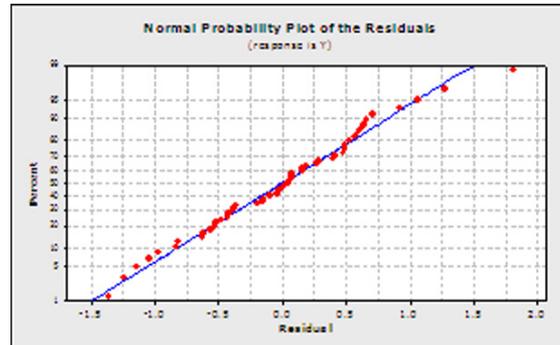


Figure 2. Normality test for the data

The normal probability plot in Figure 2 shows an approximately linear pattern consistent with normal distribution. The model does appear to be suitable for analysis and has adequate predictive ability. The purpose of any histogram is to evaluate shape and central tendency of data and to assess whether the data follows the normal distribution. Figure 3 shows the histogram of the data and bars represent the number of observations falling within consecutive intervals or bins. Each bar represents many observations; a histogram is most useful when we have a large amount of data for the examination (usually more than 50). It is evident from Figure 3 that histogram follows normal distribution and appears to be suitable for the analysis and has adequate predictive ability.

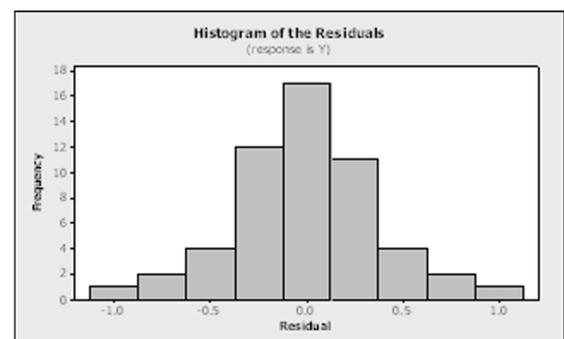


Figure 3. Histogram of the data

The plot of residuals versus the observation order shown in Figure 4, is to check for any time trends or other non-random patterns in the observed data. It indicates that the model fits the data well.

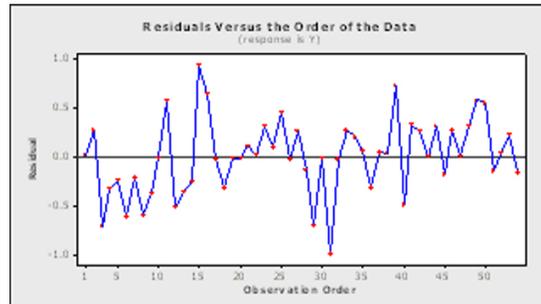


Figure 4. Run order graph for data

The plot of residuals versus the fitted values shown in Figure 5 depict that the residuals get smaller (closer to the reference line) as the fitted values increase, which may indicate the residuals have no constant variance.

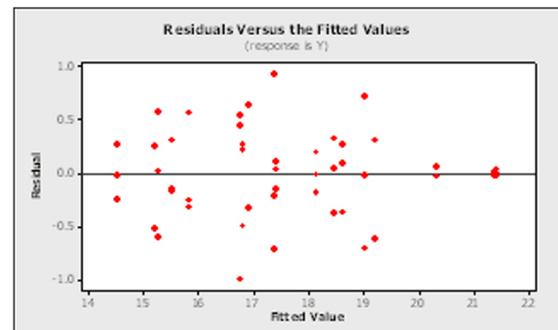


Figure 5. Shows the residual versus fitted value for the data

It indicates that the model fits the data well. The above graphs shown in Figure 2, 3, 4 and 5 suggest that the data is normal, free from skews and no known time dependencies.

### Analysis of Variance (ANOVA)

Minitab tool automatically calculates a variance analysis and provides F values (fixation indices) and P values (probability indices) for each factor and interaction term. Table 7 lists the ANOVA [LIND1974] calculations obtained for the experiment.

Source	DF	Seq SS	Adj MS	F	P
CPS	2	130.193	65.097	287	0
LSS	2	15.219	7.609	33.55	0
CSS	1	66.889	66.889	294.91	0
CPS*	4	0.505	0.126	0.56	0.696
LSS					
CSS					
CPS*CSS	2	4.835	2.418	10.66	0
LSS*	2	0.284	0.142	0.63	0.541
CSS					
CPS*					
LSS*	4	0.858	0.215	0.95	0.449
CSS					
Error		36	8.165	0.227	
Total			53	226.94	

S = 0.476251  
R-Sq = 96.40%  
R-Sq (adj) = 94.70%

Table 7. ANOVA Calculations

Careful analysis of the Table 7 reveals that all three factors from the main effects are significant (p value of 0.000) and that the interaction effect of CPS\*CSS is significant. The remainder of the interactions are not significant. Table 8 gives the coefficients for different factors and interactions. The information provided in Table 8 reiterates the fact that all main effects are significant and first order interaction of CSS and CPS is significant at 100 and 16. The rest of the first order and second order interactions are not significant.

The R2 and adjusted R2 values of 96.4% and 94.7 % shows a good model fit.

Term	Coef	SE Coef	T	P
Constant	17.6548	0.0648	272.41	0
<b>CPS</b>				
100	-1.86759	0.09165	-20.38	0
600	-0.06648	0.09165	-0.73	0.473
<b>LSS</b>				
64	-0.74759	0.09165	-8.16	0
256	0.43352	0.09165	4.73	0
<b>CSS</b>				
16	1.11296	0.06481	17.17	0
<b>CPS*LSS</b>				
100 64	0.132	0.1296	1.02	0.315
100 256	-0.0274	0.1296	-0.21	0.834
600 64	-0.1507	0.1296	-1.16	0.253
600 256	0.1248	0.1296	0.96	0.342
<b>CPS*CSS</b>				
100 16	-0.40019	0.09165	-4.37	0
600 16	0.08093	0.09165	0.88	0.383
<b>LSS*CSS</b>				
64 16	0.06426	0.09165	0.7	0.488
256 16	-0.1013	0.09165	-1.11	0.276
<b>CPS*LSS*CSS</b>				
100 64 16	-0.1287	0.1296	-0.99	0.327
100 256 16	0.0885	0.1296	0.68	0.499
600 64 16	0.1752	0.1296	1.35	0.185
600 256 16	-0.2293	0.1296	-1.77	0.085

Table 8. Main and Interaction coefficients of CPS, LSS and CSS

The second column in the table gives the B matrix of the equation  $Y = XB + E$

Using the above equation, one can calculate or predict the  $Y_i$  for factor setting of  $CPS_i$ ,  $LSS_i$  and  $CSS_i$ . Minitab automatically calculates the Y for each run using the above formula and provides that as Fits. Fits are given in Table 6. One can observe that the fits are well with in 5% deviation from the observed values except for outliers.



### Main Effect of Storage Factors on Throughput (Y)

The Figure 6 shows the main effect plot of CPS, LSS and CSS on the throughput of the storage system.

#### Effect of CPS on Throughput

Figure 6 also shows that throughput is directly proportional to the CPS. This is because the application has enough additional cache to overcome the cache misses [ANAN1989] and context switching [EDW2001] during the execution that causes the throughput to linearly increase with the increase in CPS. Figure 6 clearly shows that CPS has significant effect on the throughput of the system. But a cache size limitation is inevitable due to economic and system capability constraints.

#### Effect of CSS on Throughput

The storage system provides the facility to configure the cache segment size suitable to application block size or LSS to improve the cache utilization and hence the throughput. It is best practice to choose small CSS for small IO random application and large CSS for large IO application such as streaming video. Figure 6, shows that CSS has significant effect on the throughput of the system at smaller values because the application profile chosen for this experiment was a random IO profile with 8kb application block size. As shown in Figure 6, the throughput drops drastically are larger at CSS (64kb) and gives higher performance when configured at 16kb.

Configuring CSS at 8kb for this application profile could have increased the performance further. But this could not be validated in the experiment as 8kb CSS and 512kb LSS combinations were not feasible.

#### Effect of LSS on Throughput

LUN strip size individually does not play a significant role in this experiment. Figure 6 clearly shows that the effect of LUN stripe size above 256kb does not have much effect on the throughput when the application profile is kept constant.

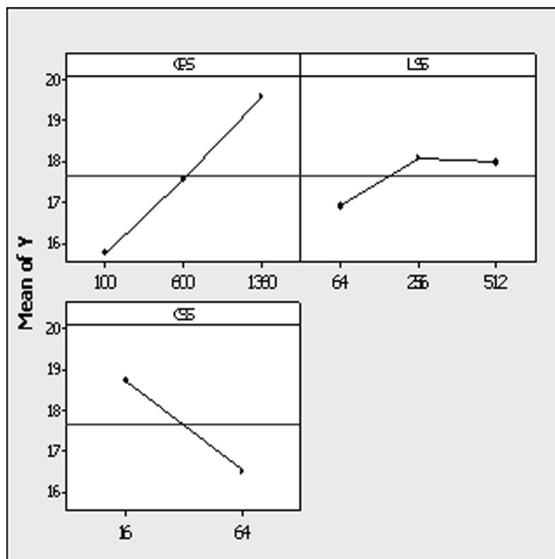


Figure 6. Main effects plot of CPS (MB), LSS (kb) and CSS (kb) given in X-axis on the Throughput (MBPS) on Y-axis.



### Interaction effect of the Storage Factors on Throughput (Y)

As discussed in the introduction, not only do the factors individually affect the output of an experiment, but their interaction will also affect the output.

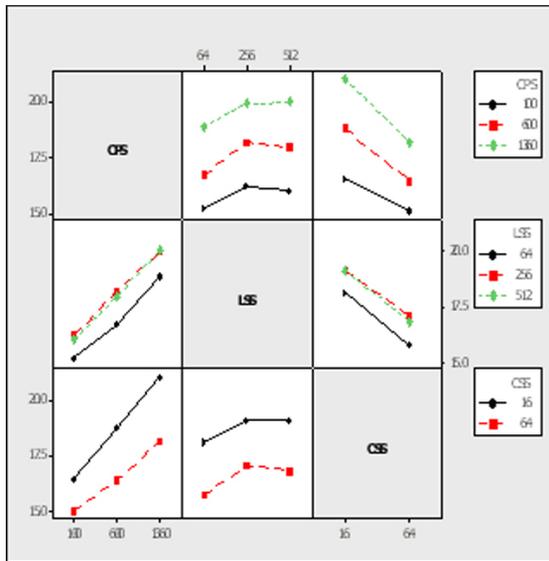


Figure 7. Interaction effect plot of the CPS (MB), LSS (KB) and CSS (KB) given in X-axis on the Throughput (MBPS) on Y-axis.

Thus, it is always important to understand the interaction effects of the factors. Figure 7 shows the interaction effects of CPS, LSS and CPS. In the graphs, the delta in the slope of the curves shows the interaction. While some interaction effect can be seen between CPS and CSS (right upper corner and left lower corner), there is no or very little interaction between CPS-LSS and CSS-LSS as shown in Figure 7. The P- value of interaction effects given in Table 8 reiterates this. Throughput is reduced with lower CPS and the drop is more profound when CSS is set at high value (64K). P values for CPS and CSS interaction is high at CPS 600 as shown in Table 8.

We conducted “best subset” regression analysis [HOCK1976] to compare the findings of the above experiment. Best subset regression analysis indicates how much each variable accounted for the variation in output. Table 9 shows the best subsets regression which identifies the right set of predictors that should be controlled to achieve the desired results.

It is evident from Table 9 that CPS and CSS together are the best subset (small number of predictors with low mallow C-p and R<sup>2</sup> adj value explaining 85.9% of the model). Adding LSS as an additional predictor would further improve the model, but marginally. The above finding, matches with DOE interaction analysis which indicated a very low level interaction of LSS with other factors. The main effect of LSS is not as strong as the other factors (LSS vs. throughput graph).

Vars	R <sup>2</sup>	R <sup>2</sup> adj	Mallows		C	L	C
			C-p	S			
1	56.9	56.1	172.8	1.3712	Y		
1	29.5	28.1	314.7	1.7544			Y
2	86.4	85.9	22.4	0.7781	Y		Y
2	60.9	59.3	154.4	1.3198	Y	Y	
3	90.3	89.8	4	0.6624	Y	Y	Y

Table 9. Best subset of the regression model Response is Y

## Conclusions and Future Work

This work gives substantial insight of how DOE can be used effectively in SAN performance testing at a storage system level which can be extended to different levels. This work has demonstrated key idea of using DOE for analyzing the storage factors relationship very effectively. In this experiment we found all three storage factors viz, CPS, LSS and CSS individually play an important role on the storage throughput. We found some interaction between CPS and CSS, but no or little interaction between LSS with CPS or CSS. The optimal values for CPS, LSS and CSS for a random IO application profile are found to be 1360MB, 256KB and 16KB respectively. Our future interests are rooted in conducting the SAN performance analysis involving more factors at different component levels. We are also investigating various techniques for performance modeling of the SAN.



## References

- [BARK2001] Barker, Richard and Paul Massiglia, Storage Area Network Essentials: A Complete to Guide to Understanding and Implementing SANs, John Wiley & Son, November 2001.
- [MART1981] Martin Schatzoff "Design of Experiments in Computer Performance Evaluation" IBM J. Res. IBM Corp, Develop. Vol. 25, no. 6, pp 848-859, 1981
- [CLAR2001] Clark, Tom, IP SANS: A Guide to iSCSI, iFCIP protocols for Storage Area Network, Addison-Wesely, December 2001.
- [CHEN1996] Chen, S. and D. Towsley, "Performance Evaluation of RAID Architectures", IEEE Transaction on Computers, Vol 45, no 10, pp.1116-1130, Oct 1996
- [LEE1993] Lee, Edward K. and Randy H. Katz., "Analytic Performance Model of Disk Arrays", Proceeding of the ACM Sigmetric Conference on Measurement and Modeling of Computer Systems, Performance Evaluation Review, ACM Press, Vol 21, No 1, pp. 98-109, May 1993
- [ARI2004] I. Ari, M. Gottwals, and D. Henze, "SANBoost: Automated SAN-Level Caching in Storage Area Networks", Proceedings of the 13th IEEE International Conference on Autonomic Computing (ICAC), pp. 164-171, 2004
- [HUSE2003] Huseyin Simitci, "Storage Network Performance Analysis" Wiley Publishing Inc, Indianapolis, Indiana, 2003
- [SHEK1990] Shekant S. Thakkur, Mark Sweiget, "Performance of an OLTP Application on Symmetry Multiprocessor System", Proceedings of the 17th Annual International Symposium on Computer Architecture, pp. 228 - 238, June 1990
- [INTE1998] Intel IO meter Project, <http://www.iometer.org/>
- [MONT1992] D. C. Montgomery, "The use of Statistical Process Control and Design of Experiments in Product and Process Improvement", IEE Transactions, 24, 5, pp. 4-17, 1992
- [1ANTO1998] Antony J, Antony F.J., "Teaching Advanced Statistical Techniques to industrial engineers and business managers", Journal of Engineering Design; 9(1), pp 89-100, 1998
- [SIRR1920] Sir Ronald A. Fisher, [http://en.wikipedia.org/wiki/Ronald\\_Fisher](http://en.wikipedia.org/wiki/Ronald_Fisher)
- [ASQ1993] DOE contributors, <http://www.asq.org/about-asq/who-we-are/>
- [PLAC1946] R. L. Plackett and J.P. Burman, "The Design of Optimum Multifactorial Experiments", Biometrika, Vol. 33, pp. 305-325, 1946
- [BOX1960] Box, G. E. P., and D. W. Behnken, "Some New Three Level Designs for the Study of Quantitative Variables." Technometrics, Vol 2, pp.455-475, 1960
- [TAGU2004] Taguchi, G., Phadke, M.S. "Quality Engineering Through Design Optimization", Proceedings of GLOBECOM 84 Meeting, IEEE Communication Society pp. 1106-1113, 2004
- [DOBS1990] Dobson, A. J. An Introduction to Generalized Linear Models. New York: Chapman & Hall. 1990
- [MINI1972] MinitabInc, <http://www.minitab.com/products/minitab/>
- [DOET1989] DOETutorials <http://www.doesinc.com/knowledge.htm>
- [FABB1998] Fabbrocino, F., Santos, J.R., Muntz, R. "An implicitly scalable, fully interactive multimedia storage server", Proceedings. 2nd IEEE International Workshop on Distributed Interactive Simulation and Real-Time Applications, Vol 19, no 20, pp.92 - 101, July 1998
- [BARRY2000] Barry J. Schoenborn, Ralph H. Thornburgh, Storage Area Network: Designing and Implementing a Mass Storage System, Prentice Hall PTR, New Jersey, 2000
- [MENG1994] Mengjou Lin, Jenwei Hsieh, James A. MacDonald, "Performance of high-speed network I/O subsystems: case study of a fibre channel network" Proceedings of the International Conference on Supercomputing, pp. 174 - 183, 1994
- [MINI1989] Minitab Guide, <http://mathstat.carleton.ca/~help/minitab/dorsfdes.pdf>
- [LIND1974] Lindman, H. R., Analysis of variance in complex experimental designs. W. H. Freeman & Co, San Francisco, 1974
- [ANAN1989] Anant Agarwal, Mark Horowitz, John Henness "An Analytical Cache Model" ACM Transactions on Computer Systems, Vol. 7, No. 2, pp 184-215, May 1989
- [EDWE2001] G. Edward Suh, Srinivas Devadas, Larry Rudolph, "Analytical cache models with applications to cache partitioning", Proceedings of the 15th International Conference on Supercomputing, pp 1-12, Sorrento, Italy, June 2001
- [HOCK1976] R.R. Hocking, "A Biometrics Invited Paper: The Analysis and Selection of Variables in Linear Regression" Biometrics, 32, pp 1-49, 1976

## Acknowledgements

The authors would like to thank anonymous reviewers for their invaluable feedback and review comments. The authors convey immense reverence and thankfulness to Kishor Arumilli, Group Head, PTG Lab, Wipro Technologies, Electronics City-IV, Bangalore, India, for providing suggestions, guidance, encouragement and kind support to this project. The authors would like to thank Mr. Devender Malhotra, Senior Consultant Software Quality Assurance, Wipro Technologies, Bangalore, India for discussion on DOE coverage.

---

### Wipro Technologies Consulting | IT Services | Testing Services | Product Design | BPO

For further information  
on our **Testing Solutions**,  
please contact  
**Sanjay Seth**  
Email: [sanjay.seth@wipro.com](mailto:sanjay.seth@wipro.com)  
Mobile: + 91 98451 97657

#### INDIA

Wipro Technologies,  
Sarjapur Road,  
Bangalore 560 035.  
Tel: +91 (80) 2844 0011.

#### FRANCE

Wipro Technologies,  
91, Rue Du Faubourg,  
Saint Honoré,  
75008 Paris.  
Tel: + 33 (1) 4017 0809.

#### UK

Wipro Technologies,  
Mimet House,  
5A- Praed Street,  
London W2 1NJ.  
Tel: +44 (20) 7087 3770.

#### USA

Wipro Technologies,  
1300, Crittenden Lane,  
Mountain View,  
CA 94043.  
Tel: +1 (650) 316 3555.

#### JAPAN

Wipro Technologies,  
# 911A, Landmark Tower,  
2-1-1 Minatomirai,  
2-Chome, Nishi-ku,  
Yokohama 220 8109.  
Tel: +81 (4) 5650 3950.

#### GERMANY

Wipro Technologies,  
Horn Campus,  
Kaistrassen 101,  
Kiel 24114.  
Tel: +49 (431) 77 55 713.

Wipro Limited is a USD 4.2 Bn IT Services company with over 718 clients globally. We have more than 88,000 employees and 48 Global Development Centers across 53 countries. We ensure stringent quality measures including using Six Sigma, PCMM, CMM Level 5, CMMi and Lean methodologies.

With 7500+ dedicated testers, the Testing Services Division is among the largest third party offshore testing services providers in the world.

© 2008 Wipro Technologies. All rights reserved.

All trademarks and copyrights, registered and unregistered, used in this document are properties of their respective owners. Testing as Managed Services (TMS)™ is the exclusive property of Wipro, pending registration.

To know more about our testing solutions, visit [www.wipro.com/testing](http://www.wipro.com/testing)