THE QUEEN'S AWARDS
FOR ENTERPRISE
2001

BBC

R&D White Paper

WHP 122

September 2005

The development of performance models for media storage systems

Y. Xiao

Research & Development
BRITISH BROADCASTING CORPORATION

BBC Research & Development
White Paper WHP 122

The Development of Performance Models for Media Storage Systems

Y. Xiao

Abstract

Modern professional media storage systems usually comprise of RAIDs, which help to improve the capacity and throughput of the system. Different performances from various manufacturers are easily measurable but difficult to predict and compare, because of the proprietary implementations of unique hardware structures and software configurations.

In addition, as technologies spread across all levels of the market, storage systems made of commodity components start to exhibit similar throughput capacities too, blurring the boundaries between the tailor-made and homemade while rendering the comparisons more confusing.

A model is therefore developed to evaluate the performances more analytically. This paper firstly categorises the various RAID levels and evaluates their features. Then it introduces the special test software used for the analysis. Hence it explains the underlying mathematic theory behind the different throughput behaviours. Any other factor affecting the throughput of certain types of storage systems is also identified and taken into consideration in the model. With annotated performance graphs plotted from test results, the development of the performance model is described.

This document was originally published in the Proceedings of the International Broadcasting Convention, September 2005

Additional key words: Performance model, RAID, BBCMeter, NAS, PC-DAS, Throughput, Degradation Index, Probability, Latency,

BBC Research & Development
White Paper WHP 122

The Development of Performance Models for Media Storage Systems

Y. Xiao

1 Introduction

As computerised HD production becomes the ultimate trend in the future of television, specifications for high performance media content servers rise sharply. In contrast the method used to analyse this feature remains unchanged. This is partly due to the large scope of the research in this area.

The most adopted way of accessing the performance is to look at throughput graphs obtained from testing software, which measures the absolute transfer rate under different access modes. However the difference in performance could be caused by various determinants and very often a better performance does not necessarily imply architectural superiority. Therefore this research aims to extract and make separate comparisons of the factors that shape the throughput curves, by establishing a mathematical model.

Throughput under sequential access mode varies significantly with the size of the local buffers and the prediction method used in the storage server, whereas random access throughput is only affected by the structure of the systems. Furthermore, write operations do not depend critically on the real-time performance of a system because data can be buffered onto cache before being written to hard disks. Hence this study will only focus on predicting the random read access performance.

The model established would be used for the evaluations of two typical storage solutions: a specialist NAS used as a benchmark and a PC-DAS made of commodity components.

2 RAID storage system overview

Redundant Array of Independent Disks became the most commonly adopted solution in many storage systems because of its large capacity and improved performance. In most modern RAID systems, the file system is distributed across several hard disks through block-level striping and redundancy is added to prevent data loss from disk failures. In general, there are three major types of RAIDs: RAID_0, parity-striped RAID and mirrored RAID, which have different impacts on performance under certain access conditions.

2.1 Striped RAID without parity (RAID_0)

Technically RAID_0 should be named as AID because it does not include any redundancy in the structure. Data is simply split into equal-sized blocks and spread evenly into all the member disks. In this case, maximum capacity is achieved with no fault tolerance and data security is reduced dramatically.

As I/O tasks can be shared among multiple disks, throughput for both random and sequential read-write is improved in RAID_0. For single-client situation, the performance improvement may not be obvious due to the small amount of data access. The random read performance of RAID_0 is representative of many other more complicated striped RAIDs and therefore will be studied thoroughly in this study.

2.2 Parity-striped RAID

The most popular configuration of stripe with parity is RAID_5, which generates a parity block for every data stripe. The parity blocks are stored evenly on different disks and effectively take the size of one disk in the array. Though capacity is reduced, the system is capable of recovering from single-disk failures. Other RAID levels, especially many proprietary ones, which implement parity redundancy, share the same feature with deviations in parities computation methods and the numbers of parity blocks in one stripe.

Read access performances for these kinds of RAID are very close to RAID_0 because the parity blocks are not accessed on normal data reads and hence, all member disks can be engaged with I/O operations as being the case for RAID_0. The speed for storing data is slightly reduced because the process of generating and storing parity blocks takes time. When there is a disk failure or during recovery, the performance is degraded tremendously, which occurs infrequently and thus is not the interest of this study.

2.3 Mirrored RAID

RAID_1 is perhaps the most extreme approach of adding redundancy in which every member disk contains a mirrored copy of the complete file system. It is rarely used singularly but often found in nested levels such as RAID_01 and RAID_10. Although the structures are completely different, the amount of computations required for random access is the same. Hence there is no performances difference between them. This type of RAID is very practical for media content server from commodity components for small-to-medium scale television productions, which requires the data to be available at full throughput bandwidth despite of any possible disk failure.

Mirrored RAID provides enhanced read performance by having extra copies of the original data. Nevertheless this benefit cannot be enjoyed in a single-client environment. The common misconception on mirrored RAID is that it tremendously hinders write performance by taking time to store the redundant data. In fact the redundant data is stored simultaneously by different hard disks without spending any extra time. Actually it is even faster than parity-striped RAID because it does not calculate parity. Depending on the controller hardware, the write performance can differ significantly. This is again not the main concern of this research. However the random read behaviour of mirrored RAID will be investigated and the performance modelled.

3 The BBCMeter

Developed by former BBC senior R&D engineer R.Walker, the BBCMeter is a storage-testing tool tuned specifically to simulate real-life television production scenarios. It has been used extensively in the BBC as a standard testing tool. Therefore it will be the measurement results by BBCMeter that the performance model is built upon.

3.1 Performance testing

Typical BBCMeter test settings comprise of a number of clients linked though a gigabit Ethernet connection to the storage test server. A PC on the network runs the console application that controls the test applications on all the clients. Previous experiments have shown that it is possible for a small test file set to be cached completely onto a storage system's local memory, eliminating completely the need to read directly from the hard disks. Hence twenty 1GB test files are copied onto the storage system, equalling the size of an hour-long 50Mbps compressed HD video.

During the test, a mixture of sequential or random read-write commands is sent from the clients to the storage system. By default, 50% of the all accesses are random and 10% are write operations. This scheme best mimics a small-to-medium scale production environment where a number of clients are steaming, skimming over and editing HD contents centrally stored on a content server. It is also a very demanding test on a storage system in terms of hard disk operations. All tests are left running for at least half an hour to obtain a sustained average performance reading.

Similar to other software, BBCMeter makes measurements in two categories: transfer rate and latency. While latency reflects the delay caused by the network and storage structures, the transfer rate determines the overall performance of a storage system. A results file is output at the end of the test showing the readings separately for the different access modes, which can then be used for plotting a performance graph.

3.2 The non-queuing feature

One of the BBCMeter's unique features that makes its results suitable for modelling is that each test client only issues one request at a time. This is very important in the construction of the model because with requests queuing, a very small number of clients would be capable of saturating the random throughput of a system consisting of a small number of hard disks.

In an actual studio condition, every frame of motion picture must be delivered on-demand. A queue of requests is only useful for streaming. Very often an editor would jump over scenes and expect the picture to be displayed accordingly as they navigate through the timeline with their mouse. This action, also known as "scrubbing", is completely random and makes software buffering, which sends a queue of requests, totally useless. In another words, if requests for random data from a client queue at the server end, there must have been delay observed at the client end. Thus to measure the effective transfer rate of a sustainable on-demand random read process, the clients must not queue their requests.

The BBCMeter's special configuration suits this requirement perfectly as it was purposely tuned from its development to be the test tool dedicated for media storage systems only. This non-queuing feature has also made the statistical analysis of storage system simpler as the total number of requests will constantly equal the number of clients, which is where the entire hypothesis of the model is based.

4 Performance model for parity-striped RAID

All storage systems can handle simultaneously only a certain number of requests, especially those of random read, which is limited mainly by the physical properties and the number of the hard disks used in the array. As a result, throughput saturates over a large number of clients.

Fig 1 shows the saturation curve of random throughput with increasing number of clients. The raw figures are from the test on the BBC benchmark specialist parity-striped RAID NAS storage system.

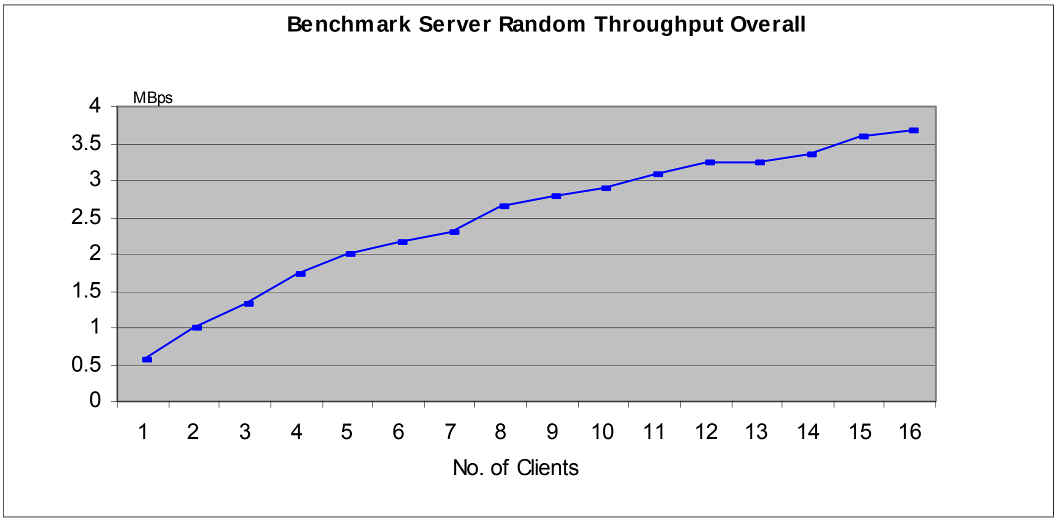

Fig 1 – Benchmark Server performance

As shown on the graph, the overall system throughput does not saturate linearly, which implies that the number of clients must have an impact on performances observed at individual clients.

Fig 2 shows the graph of average throughput per client against the number of clients from the same test result as above.

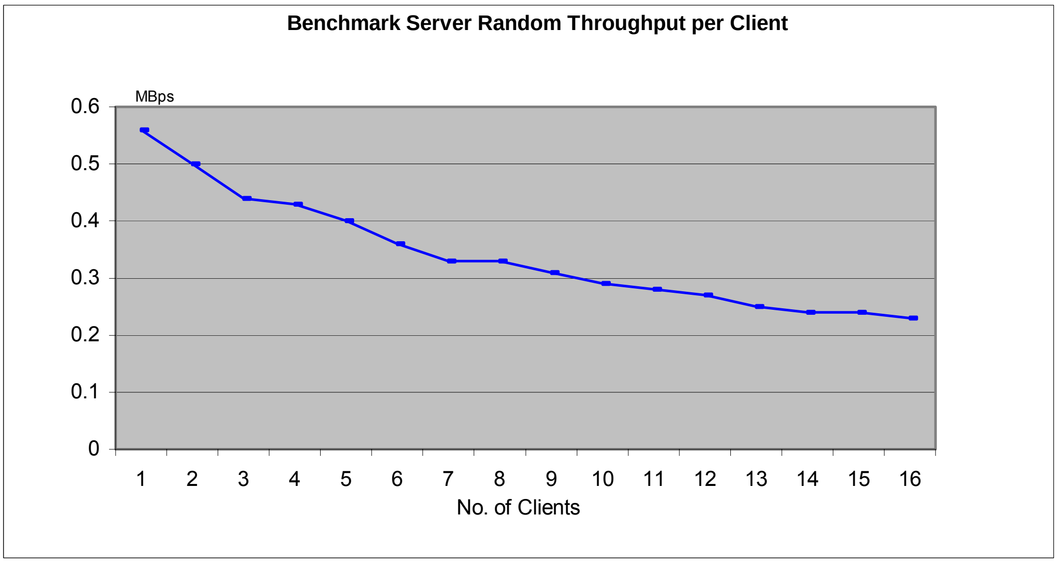

Fig 2 – Benchmark throughput per client vs. Total no. of clients

The graph reveals a curvilinear relationship between the throughput per client and the number of clients, resulting from the increasing requests at the server. To help analyse this throughput trend, a degradation index can be introduced as the ratio of single client throughput to throughput per client for other numbers of clients.

$$\text{Degradation Index for (1 to 16) Clients} = \frac{\text{Throughput per client for 1 Client}}{\text{Throughput per client for (1 to16) Clients}} \quad \text{.........(1)}$$

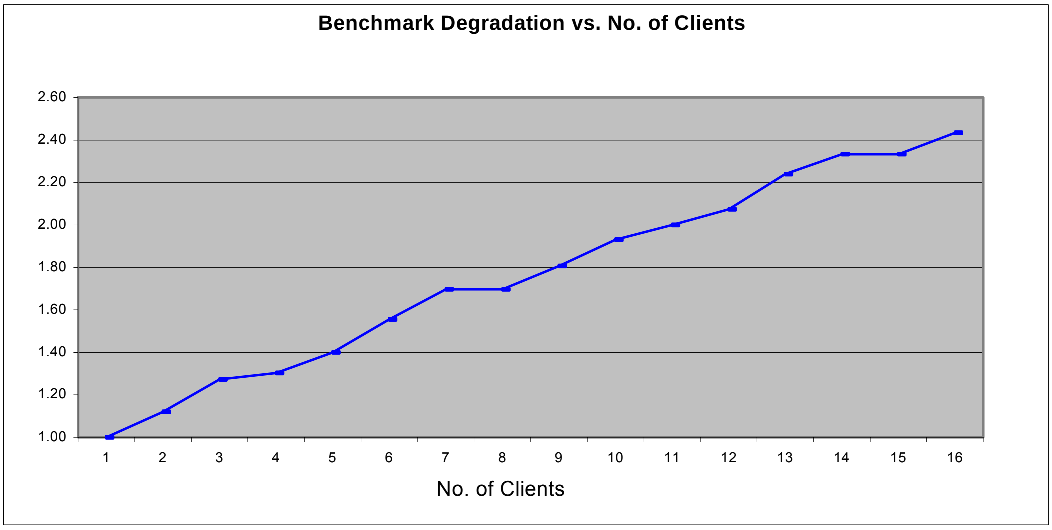

Fig 3 – Benchmark throughput degradation vs. No. of clients

Directly from Fig 3, it is clear that the degradation index at 1 client is 1 while at 16 clients is approximately 2.4. In another words, the transfer speed for a single client is 2.4 times as fast as that for 16 clients.

This degradation index is a system performance indicator that can be modelled mathematically. Nevertheless to predict the actual throughput figures of a storage system, a single client test is required to acquire the initial value for the calculations.

4.1 Degradation By Probability

In a BBCMeter test, if no command is queued at any hard disk in the system, there is no delay and thus the degradation index equals 1, like in the case of single client. However in reality when there are a number of clients reading data randomly from the system, it is always possible that some disks contain the data simultaneously requested by two or more clients

Hence in multiple client tests, a command might have to wait for others to finish, which causes delays. In general the delay is an integer multiple of the average access time of the hard disk, which is the time taken to complete one command. The number of access times that a command has to wait is equal to the degradation index for that client.

Some disks in the system can also be idle because they do not contain the requested data. Only non-idle disks are responsible for the throughput. The degradation index for a particular situation is then the ratio of the number of clients, which equals the number of commands, to the number of non-idle disks, defined as ‘N_c’ and ‘n’ respectively.

$$\text{Particular Degradation Index by probability} = N_c/n \qquad \text{.........(2)}$$

The value for n varies in different situations. Hence to find the aggregate degradation index for all situations, each particular situation must be weighed by its respective probability, defined as P(n).

$$\text{General Degradation by Probability} = \sum[N_c/n * P(n)] \qquad \text{.........(3)}$$

For example in a simple scenario where a RAID system comprises of 3 disks is tested with 3 clients on BBCMeter. It is possible to have 3 different possible numbers of non-idle disks, 1, 2 and 3, with particular degradation indexes of 3/1, 3/2 and 3/3 respectively. As shown in Fig4, each of them has different numbers of possible situations.

DISK 1	DISK 2	DISK 3
3	0	0
2	1	0
2	0	1
1	2	0
1	1	1
1	0	2
0	3	0
0	2	1
0	1	2
0	0	3

Legends:		No. of Situations:
🟥	: 1 Non-Idle Disk	3
🟪	: 2 Non-Idle Disks	6
🟨	: 3 Non-Idle Disks	1
0,1,2,3	: No. of Clients	Total = 10

When n=1:	$N_c/n = 3/1$,	P(n) = 3/10
When n=2:	$N_c/n = 3/2$,	P(n) = 6/10
When n=3:	$N_c/n = 3/3$,	P(n) = 1/10

Fig 4 – Example of a 3-disk, 3-client system

Therefore the degradation by probability for this system with 3 clients is calculated as follows:

$$\sum_{n=1}^{3} [N_c/n * P(n)] = (3/1)*(3/10)+(3/2)*(6/10)+(3/3)*(1/10) = 1.90 \qquad \text{.........(4)}$$

This number is the predicted purely on the basis of probability theory, which does not take into account of any structural characteristics of the storage system hardware. BBCMeter test results yield an actual degradation index of 1.96, very close to the empirical value.

Fig 5 shows the probability distributions for having different numbers of idle disks for other equal numbers of disks and clients.

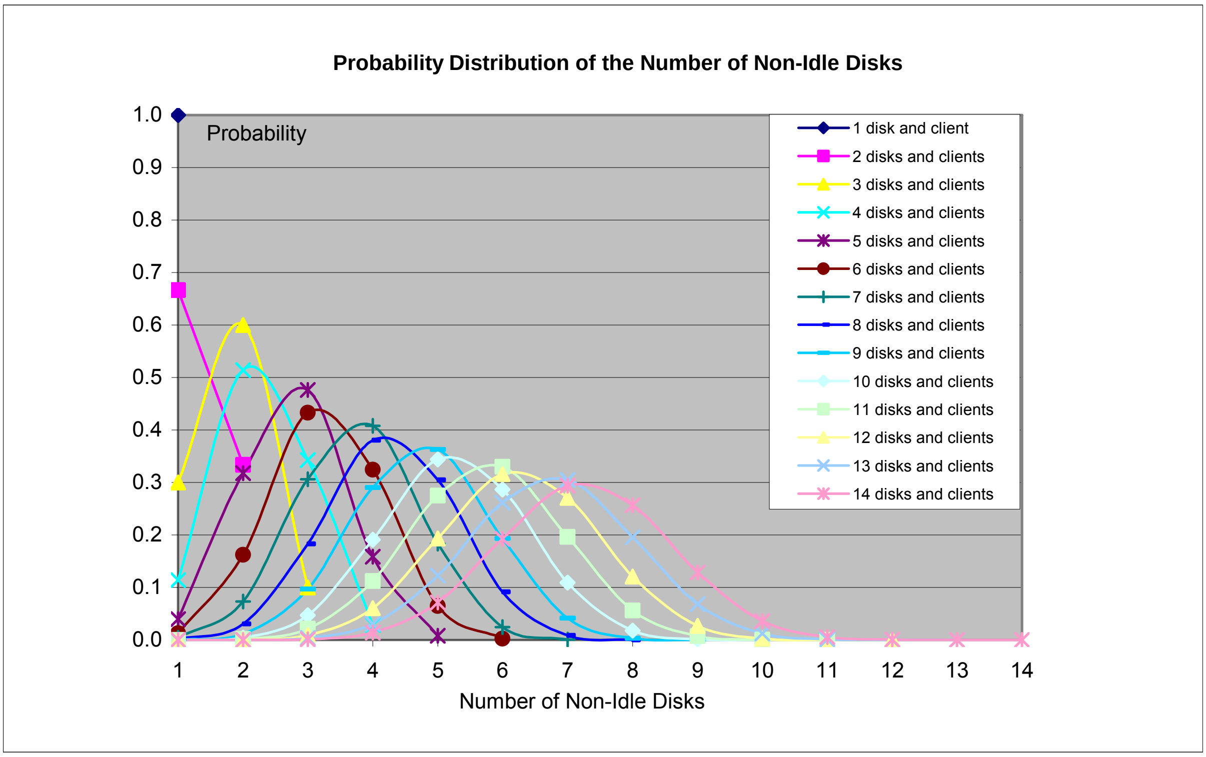

Fig 5 – Probability distribution of the number of non-idle disks in different situations

As observed from the graph, for 4 or more disks and clients, the highest possible number of non-idle disks is only half of the total number of disks. Also from the normally distributed probabilities, the degradation by probability index is calculated as 2 for all large numbers. Thus the throughput per client is only half the amount of reading from a stand-alone PC. Experimental results obtained from the benchmark server reveal that the actual figure is even lower.

Therefore the main benefit of a stripped RAID system is the centralisation of storage and enhancement of capacity, which is a critical server requirement only when the same content has to be made available to multiple clients. For very small workgroups, peer-to-peer configuration sometimes is a better implementation in terms of maximising the random access performance.

4.2 Calculating degradation by probability index

From the example of the system having 3 disks with 3 clients, it can be observed that when there is only one non-idle disk, it can be any one of the three. By applying simple probability function ‘Choose’ (symbol ‘***C***’), the number of possible situations can be easily calculated as: 3***C***1=3. When there are two non-idle disks, the disk with two commands can be any one of the three and that with one command can be either one of the other two. Therefore the number of situations is (3***C***2)*(2***C***1)=6. Finally when all of the three disks are non-idle, there is only one possible situations as 3***C***3=1.

A general formula for finding the total number of all situations in this case can be written as: (3***C***n)*[(3-1)***C***(n-1)], where ‘n’ is the number of non-idle disk. This applies perfectly as when n=1 or n=3, the term (3-1)***C***(n-1)=1, hence (3***C***n)*[(3-1)***C***(n-1)]=3 and 1 respectively.

The number “3” in this formula is for both the disks and the clients. In term (3***C***n), it is clear that ‘3’ is the total number of disks in the array, defined as ‘N_D’. In this way the formula can be written as: (ND***C***n)*[(3-1)***C***(n-1)] at this stage.

In the second term, [(3-1)***C***(n-1)], though it seemed to be following the argument of the previous term, however the number (3-1) is in fact restricted by the number of clients especially when the numbers of clients and disks are not equal. It is possible to have fewer clients than disks, in which case choosing a number from a smaller one is impossible.

For example, when there are 2 clients and 3 disks, the maximum number of non-idle disks is 2 as only 2 commands can be present simultaneously in the system. The calculation would then only be carried out for n=1 and 2 because the maximum possible value for (n-1) to choose from is limited by (N_C–1).

Therefore the general formula for calculating the number of situations for having a particular number of non-idle disks can be expressed as:

$$(N_D\boldsymbol{C}n)*[(N_C-1)\boldsymbol{C}(n-1)] \qquad \text{.........(5)}$$

The number of all possible situations is the sum of these numbers:

$$\sum_{n=1}^{u}(N_D\boldsymbol{C}n)*[(N_C-1)\boldsymbol{C}(n-1)] \qquad \text{.........(6)}$$

where ‘u’ is the smaller of N_D and N_C

The probability for a particular value of n is therefore:

$$P(n) = \frac{(N_D\boldsymbol{C}n)*[(N_C-1)\boldsymbol{C}(n-1)]}{\sum_{n=1}^{u}(N_D\boldsymbol{C}n)*[(N_C-1)\boldsymbol{C}(n-1)]} \qquad \text{.........(7)}$$

Joining with the formula for the particular index, the throughput degradation by probability index for all striped RAID system is:

$$\text{Degradation by Probability Index} = \sum_{n=1}^{u}\{N_C/n * \frac{(N_D\boldsymbol{C}n)*[(N_C-1)\boldsymbol{C}(n-1)]}{\sum_{n=1}^{u}(N_D\boldsymbol{C}n)*[(N_C-1)\boldsymbol{C}(n-1)]}\} \qquad \text{......(8)}$$

where ‘u’ is the smaller of N_D and N_C

Fig 6 shows the comparison between the calculated and actual degradation indexes of the benchmark NAS, which has 12 member disks, and three PC-DAS systems, two contain 6 member disks and the other has 3. The 3-disk PC-DAS is on RAID_0 and marked as Cmdt1.05. One of the 6-disk PC-DAS uses hardware RAID_5 and the other one uses a more complicated RAID_50 configuration by having two hardware-controlled RAID_5 joined by software RAID_0. They are represented as Cmdt1.1 and 1.2 respectively. The benchmark server uses a proprietary parity-striped RAID level.

As shown on the graph, the error of the predictions is small for the NAS but large for the PC-DAS. This error is consistent for all the DAS storages attached to the same PC. It is evident that apart from the degradation by probability there must be another factor that clearly affects the performance of PC-DAS while having very minor impact on NAS. Further more the effect of this factor become more significant with increasing number of clients.

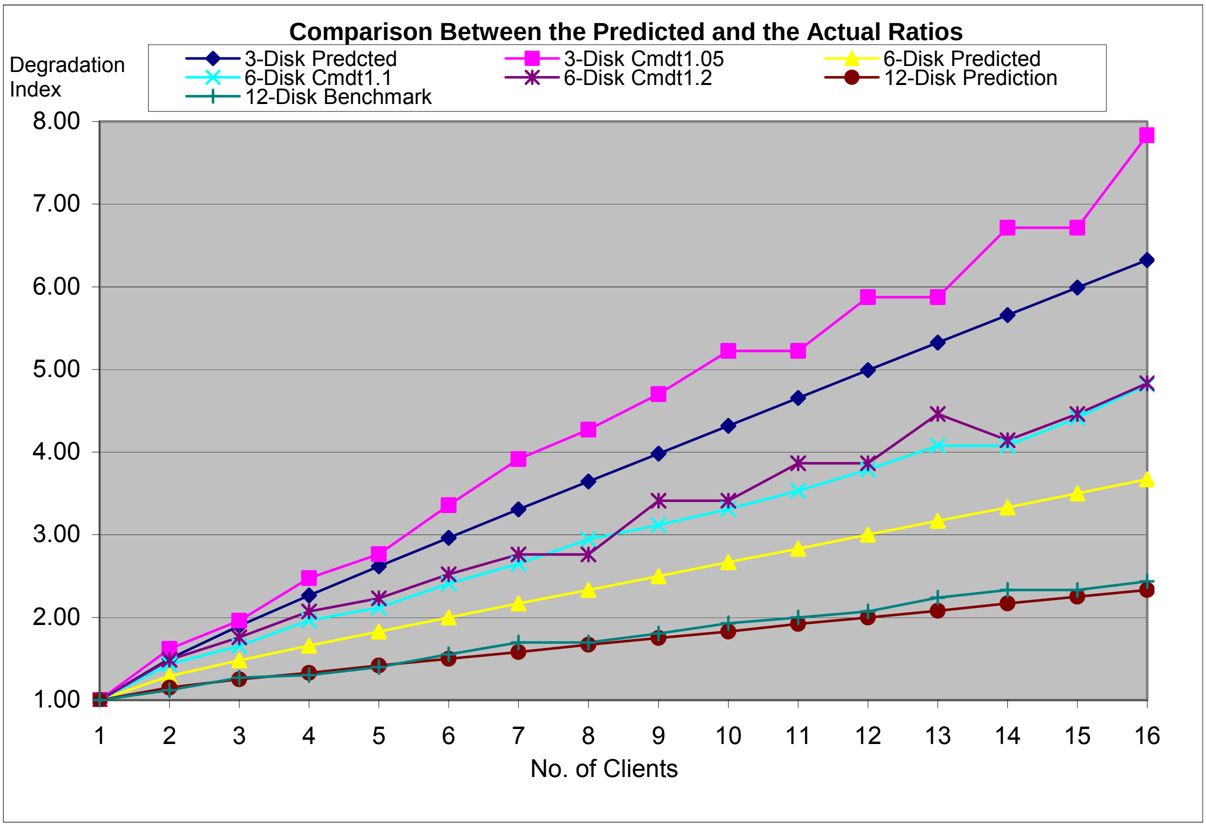

Fig 6 – Comparison between the predicted and actual degradation indexes

4.3 Degradation by latency

It is believed that the line between NAS and PC-DAS is drawn on their different characteristics in terms of latency. Depending on the network structure and storage controller circuitry, this figure varies considerably for different systems. In BBCMeter tests, the network factor has been made negligible so that the latency measured is solely the delay caused by the storage systems.

During storage I/O operations, the controller holds every command for a certain amount of time before sending it to the hard disks. This delay is the major cause of the system latency. It is unavoidable, as the controller has to locate the address of the data and communicate with the hard disk. It is hence believed to be the reason for the error between the predicted and actual degradations.

Many specialist storage system controllers are capable of processing multiple commands in parallel. Such systems can sustain low constant latencies over a large number of clients before the figures start to increase. For the benchmark server, which is used in this study, this number is so large that it would cause the network to incur notable amount of latency before the server itself does.

Fig 7 shows the average latency graph of the benchmark server. From the graph, the average latency for all number of client remains at 1.1ms, which is equal to the delay caused by a single client. Part of the superiority of this specialist storage system is clearly demonstrated.

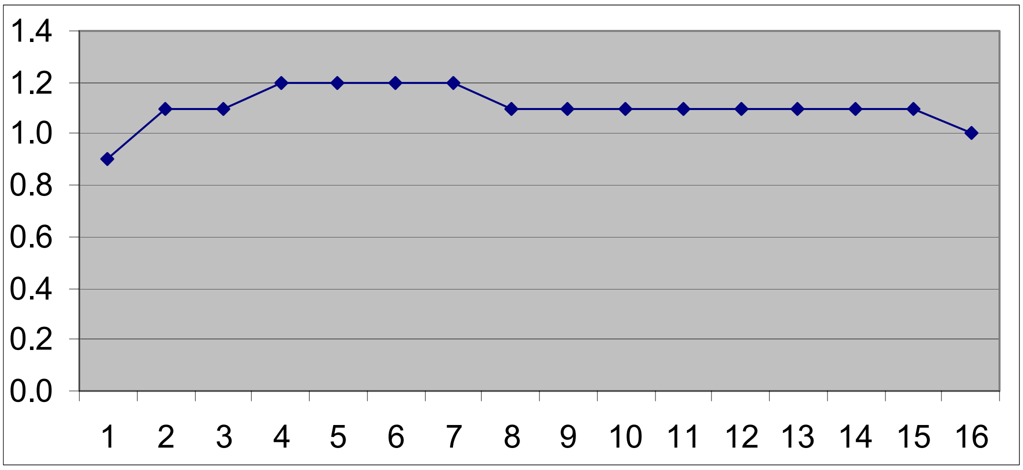

Fig 7- Benchmark NAS latency vs. No. of clients

In contrast, when several commands arrive at a PC-DAS simultaneously, the latency escalates as a result of the commands being processed serially.

Fig 8 shows the average latency graph of a 6-disk PC-DAS system. The latency suffered by 16 clients in this PC-DAS is 180 times higher than that in the benchmark NAS. It is also noted that the latency suffered by single client in both systems is rather similar. It seems the difference is mainly on their capability of handling additional clients.

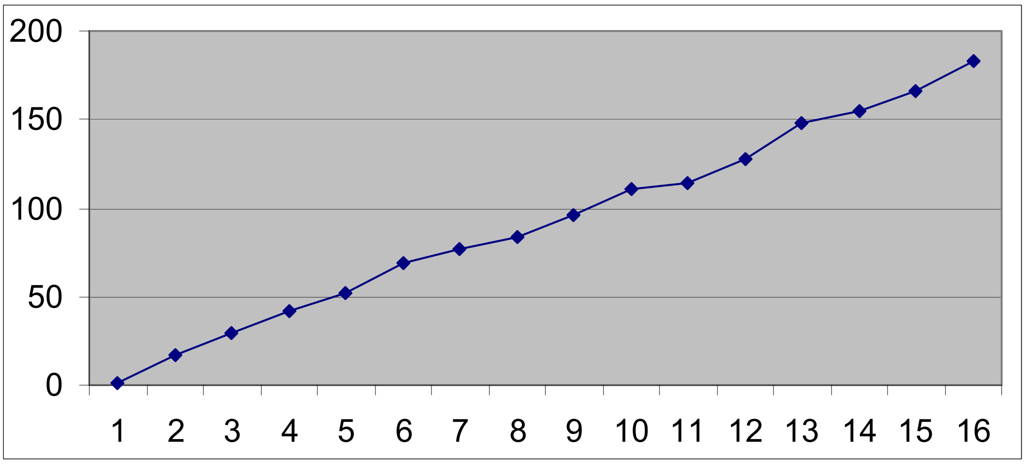

Fig 8 - Average PC-DAS latency vs. No. of clients

The latency graphs for both systems are observed having a linear relationship with the number of clients. For the benchmark NAS server, the coefficient is 0. This linearity that makes it possible to predict latency based on two different readings. If the latency for x clients is ‘L_X’ and for y clients ‘L_Y’, then the latency increment for every added client, defined as ‘ΔL’, can be found as:

$$\Delta L=(L_X-L_Y)/(x-y) \qquad \text{.........(9)}$$

In the case of the benchmark server, when ΔL=0, the accuracy of the degradation estimation is not affected. Since single client latency is the most significant delay, it can be used as ΔL to obtain a more accurate performance prediction.

Nevertheless the degradation caused by latency cannot be calculated linearly because it increases exponentially with the number of clients to the single client. There must be a further delay suffered, which is not measured as latency but reflected in the performance. Though this phenomenon requires further explanation, it is perfectly viable to assume that the actual delay is the product of latency and the number of added clients.

4.4 Calculating degradation by latency index

To translate latency figures into terms compatible with throughput degradation, it is essential to convert the throughput ratio into time ratio. Since the throughput is a measure of transfer speed, it is calculated by dividing the data length with the time taken to complete the data access, defined as ‘T’:

$$\text{Throughput} = \text{Data Length} / T \qquad \text{………(10)}$$

The data length in BBCMeter is set to be 1MB, thus Throughput = 1MB/T. The time taken for a single client to finish this data access is defined as ‘T_S’ and for other numbers of clients as ‘T_{Nc}’, where ‘N_C’ is the total number of clients. Therefore the throughput degradation due to any number of clients is:

$$\text{Degradation Due to } N_C \text{ Clients} = \frac{\text{Single Client Throughput}}{N_C \text{ Clients Throughput}} = \frac{1MB/\ T_S}{1MB/\ T_{Nc}} = T_{Nc}/T_S \qquad \text{……(11)}$$

Where T_S is found by:

$$T_S = \text{Data Length / Single Client Throughput} = \text{1MB / Single Client Throughput} \qquad \text{……(12)}$$

T_{Nc} can be further expressed as the sum of delay caused by other commands arriving earlier at the hard disks, ‘T_P’, and that caused by latency, ‘T_L’.

$$T_{Nc} = T_P + T_L \qquad \text{………(13)}$$

Therefore:

$$\text{Degradation Due to } N_C \text{ Clients} = (T_P + T_L)/\ T_S = T_P/T_S + T_L/T_S \qquad \text{………(14)}$$

The term ‘T_P/T_S’ is effectively the degradation by probability and ‘T_L/T_S’ the degradation by latency. Nevertheless this T_L is not the latency itself, but actually the product of latency and the number of added clients as mentioned earlier.

$$T_L = L_{Nc} * (N_C -1) \qquad \text{………(15)}$$

In this equation, ‘(N_C -1)’ gives the number of clients that is added to the single client and ‘L_{Nc}’ denotes the latency for N_C clients, which is calculated from ΔL and N_C:

$$L_{Nc} = \text{Single Client Latency} + \Delta L * (N_C -1) \qquad \text{………(16)}$$

The value of ‘Single-Client Latency’ is negligible compared to the large value of ΔL in this case. Hence T_L can be calculated as:

$$T_L = \Delta L * (N_C -1) * (N_C -1) = \Delta L * (N_C -1)^2 \qquad \text{………(17)}$$

Therefore the formula for finding the degradation by latency index is:

$$\text{Degradation by Latency Index} = T_L/T_S = \frac{\Delta L * (N_C -1)^2}{\text{1MB / Single Client Throughput}} \qquad \text{……(18)}$$

A universal formula applicable in all system is:

$$\text{Degradation by Latency Index} = T_L/T_S = \frac{(L_X-L_Y)/(x-y) * (N_C -1)^2}{\text{1MB / Single Client Throughput}} \quad \text{.........(19)}$$

where values for L_X, L_Y, x, y and 'Single Client Throughput' are obtained from actual BBCMeter test results

For $\Delta L = 0$, the single client latency can be used instead for a more precise prediction.

4.5 Final Model

Combining the degradations by probability and latency, the overall empirical degradation index for any parity-striped RAID can be calculated as:

$$\sum_{n=1}^{u}\{N_C/n * \frac{(N_D\mathit{C}n)*[(N_C-1)\mathit{C}(n-1)]}{\sum_{n=1}^{k}(N_D\boldsymbol{C}n)*[(N_C-1)\boldsymbol{C}(n-1)]}\} + \frac{(L_X-L_Y)/(x-y) * (N_C -1)^2}{\text{1MB / Single Client Throughput}} \quad \text{.........(20)}$$

where 'u' is the smaller of N_D and N_C, and values for L_X, L_Y, x, y and 'Single Client Throughput' are obtained from actual BBCMeter test results

By dividing the single client throughput with the empirical degradation index, then multiplying the number of clients, the performance of any parity-striped RAID systems on any number of clients can be modelled. Fig 9 compares the predicted and actual performances of the various systems described in the previous section.

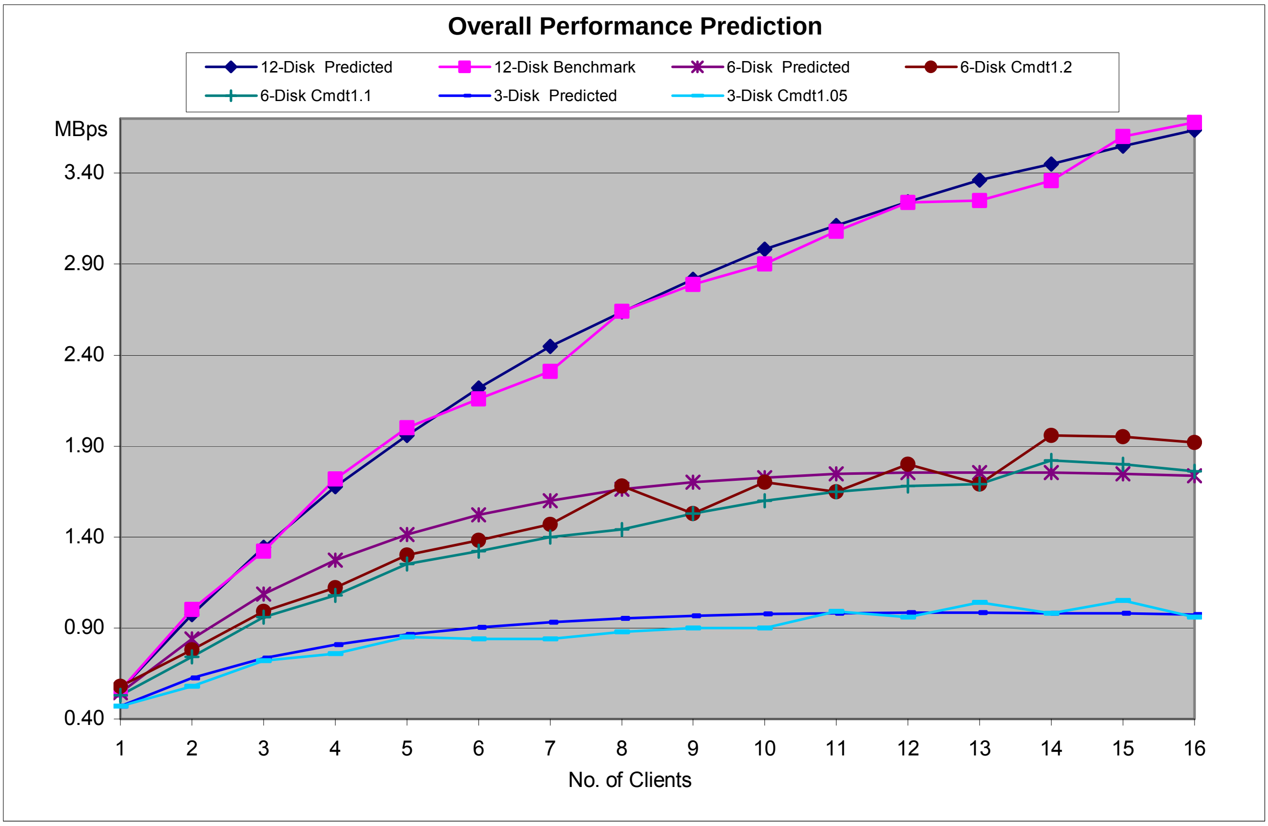

Fig 9 – Comparison between the predicted and actual Performance for various systems

5 Performance model for mirrored RAID

In a mirrored RAID system, unless there is only 1 client accessing the data, there would always be at least 2 non-idle disks present in the system, an obvious advantage over non-mirrored RAID. Nevertheless the degradation index must still be calculated in the same way as it is for the parity-striped RAID. For convenience and consistency, the short form expressions defined in the earlier part of the paper will be followed in this chapter.

5.1 Calculating Degradation By Probability Index

Finding the number of possible combinations for having a certain number of non-idle disks in a particular situation during the random data read process remains the centre of interest in this section. It becomes more complicated than the simple parity-striped RAID levels because of the extra copies that are available for access, which generates further possible combinations in addition to the single-copy situations.

Using the example of a 6-disk, 3-client system, in which data is striped among three disks and mirrored onto another three, all possible combinations can be listed and degradation by probability index calculated as shown in Fig 10.

Disk1	Disk1 Mirror	Disk2	Disk 2 Mirror	Disk 3	Disk 3 Mirror
2	1	0	0	0	0
1	1	1	0	0	0
1	1	0	0	1	0
1	0	1	1	0	0
1	0	1	0	1	0
1	0	0	0	1	1
0	0	2	1	0	0
0	0	1	1	1	0
0	0	1	0	0	0
0	0	0	0	2	1

Legends:		No. of Combinations:
🟪	: 2 Non-Idle Disks	3
🟨	: 3 Non-Idle Disks	7
0,1,2	: No. of Clients	Total = 10

When n=2: $N_C/n = 3/2$, $P(n) = 3/10$

When n=3: $N_C/n = 3/3$, $P(n) = 7/10$

$$\text{Degradation by probability} = 3/2 * 3/10 + 3/3 * 7/10$$

$$= 1.15$$

Fig 10 - Example of a 2x3-disk, 3-client system

The same index for a parity-striped system with same numbers of disks and client is 1.48, but with twice of the capacity. Apparently mirrored RAID for a small number of clients is extremely inefficient.

It is noticed that the number of total possibilities is equal to that of a 3-disk, 3-client, parity-striped RAID. Every mirrored disk pair can be considered as a RAID member that behaves in the same way as individual disks in a non-mirrored system. Therefore if ‘N_M’ denotes the total number of members in the RAID and ‘m’ denotes the number of the active ones, the term ‘($N_M C m$)’ can be used in the same way as ‘($N_D C n$)’ in the non-mirrored RAID.

For the situation of having 2 non-idle disks to occur, the number of active members in the system must be 1. When there are 3 non-idle disks, there can be either 2 or 3 active members, and thus their numbers of possible situations must add up to give the total figure for having 3 non-idle disks.

Detailed derivation of the calculation method is very complex and not the focus of this study. In general, if 'r' is the number of mirrored copies in a RAID, the number of possible situations of having certain numbers of active members can be calculated as:

$$(N_M\mathit{C}m) * \underbrace{(m\mathit{C}k) * (k\mathit{C}j) * \cdots * (a\mathit{C}z)}_{(r\text{-}1)\ \text{terms}} * [(N_C-1\underbrace{-m-k-j-\cdots-a}_{\text{the }(r\text{-}1)\ \text{terms}})\mathit{C}(z-1)] \qquad \ldots\ldots\ldots(21)$$

In this equation, with the same value of 'm':
when k=1, j=1, ··· and a=1, the number of non-idle disks is (r+m-1), A in Fig11;
when k=2, j=1, ··· and a=1, the number of non-idle disks is (r+m), B in Fig11;
when k=2, j=2, ··· and a=1, the number of non-idle disks is (r+m+1), C in Fig11;
...
when k=m, j=m, ··· and a=m, the number of non-idle disks is (r*m), D in Fig11;

A	Mirror 1	Mirror 2	Mirror 3	Mirrors ···	Mirror r
Member 1	Active	Active	Active	Active	Active
Member 2	Active	Idle	Idle	Idle	Idle
Member 3	Active	Idle	Idle	Idle	Idle
Members ···	Active	Idle	Idle	Idle	Idle
Member m	Active	Idle	Idle	Idle	Idle
Members ···	Idle	Idle	Idle	Idle	Idle
Member N_M	Idle	Idle	Idle	Idle	Idle

B	Mirror 1	Mirror 2	Mirror 3	Mirrors ···	Mirror r
Member 1	Active	Active	Active	Active	Active
Member 2	Active	Active	Idle	Idle	Idle
Member 3	Active	Idle	Idle	Idle	Idle
Members ···	Active	Idle	Idle	Idle	Idle
Member m	Active	Idle	Idle	Idle	Idle
Members ···	Idle	Idle	Idle	Idle	Idle
Member N_M	Idle	Idle	Idle	Idle	Idle

C	Mirror 1	Mirror 2	Mirror 3	Mirrors ···	Mirror r
Member 1	Active	Active	Active	Active	Active
Member 2	Active	Active	Active	Idle	Idle
Member 3	Active	Idle	Idle	Idle	Idle
Members ···	Active	Idle	Idle	Idle	Idle
Member m	Active	Idle	Idle	Idle	Idle
Members ···	Idle	Idle	Idle	Idle	Idle
Member N_M	Idle	Idle	Idle	Idle	Idle

D	Mirror 1	Mirror 2	Mirror 3	Mirrors ···	Mirror r
Member 1	Active	Active	Active	Active	Active
Member 2	Active	Active	Active	Active	Active
Member 3	Active	Active	Active	Active	Active
Members ···	Active	Active	Active	Active	Active
Member m	Active	Active	Active	Active	Active
Members ···	Idle	Idle	Idle	Idle	Idle
Member N_M	Idle	Idle	Idle	Idle	Idle

Fig 11 – Different numbers of non-idle disks can be resulted from the same number of active members

Higher values of 'r' and 'N_C' also enlarge the range of values for k, j, and other variables. For the same value of 'm', it is possible for the sums of different sets of k, j and other values to equal one another. In such cases, the values of 'n' are also equal. For example, the numbers of non-idle disks when k=2, j=2 is equal to that when k=3, j=1, as shown in Fig 12.

Furthermore as different values of ‘m’ have overlapping values for ‘n’, all suitable values of ‘m’ must be considered in the calculation for a particular value of ‘n’. Depending on the number of mirrored copy in the RAID system, the number of different ‘m’ values involved in the calculation of a particular ‘n’ value varies as shown in Fig 12.

For N_C = Any Natural Number:

r=2	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8
m=1	N_C=1	k=1						
m=2		N_C=2	k=1	k=2				
m=3			N_C=3	k=1	k=2	k=3		
m=4				N_C=4	k=1	k=2	k=3	k=4
m=5					N_C=5	k=1	k=2	k=3
m=6						N_C=6	k=1	k=2
m=7							N_C=7	k=1
m=8								N_C=8
No. of m involved:	1	2	2	3	3	4	5	5

r=3	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8
m=1	N_C=1	N_C=2	k=1,j=1					
m=2		N_C=2	N_C=3	k=1,j=1	k=2,j=1	k=2,j=2		
m=3			N_C=3	N_C=4	k=1,j=1	k=2,j=1	k=2,j=2 & k=3,j=1	k=3,j=2
m=4				N_C=4	N_C=5	k=1,j=1	k=2,j=1	k=2,j=2 & k=3,j=1
m=5					N_C=5	N_C=6	k=1,j=1	k=2,j=1
m=6						N_C=6	N_C=7	k=1,j=1
m=7							N_C=7	N_C=8
m=8								N_C=8
No. of m involved:	1	2	3	3	4	5	5	5

Fig 12 – Different numbers of ‘m’ values are needed for the same value of ‘n’, for different values of ‘r’

The total number of possible situations can be found as follows:

$$\sum_{m=1}^{v}(N_M\boldsymbol{C}m) * \underbrace{\sum_{k=1}^{m}(m\boldsymbol{C}k) * \sum_{j=1}^{k}(k\boldsymbol{C}j) * \cdots * \sum_{z=1}^{a}(a\boldsymbol{C}z)}_{(r-1)\text{ terms}} * [(N_C-1\underbrace{-m-k-j-\cdots-a}_{\text{the }(r-1)\text{ terms}})\boldsymbol{C}(z-1)] \qquad ....(22)$$

where ‘v’ is the smaller of N_M and N_C

The ratio between the number of situations for a particular ‘n’ value and that for all the possible ‘n’ values can then be used for finding ‘P(n)’, the probability of having n non-idle disks in the system. The throughput degradation is still calculated as ‘N_C/n’. Hence, the degradation by probability index can be computed in the same way as for the parity-striped RAID.

5.2 Calculating degradation by latency index

The latency measured on the tested mirrored RAID PC-DAS is approximately equal to the previous readings on the parity-striped RAID, directly attached to the same PC. It proved that latency is only dependent on the controller hardware structures, and totally independent of the RAID levels used.

In theory the latency by degradation for a mirrored RAID can be calculated simply in the same way as the parity-striped RAID. However experimental results show that in a mirrored RAID with r=2, degradation by latency measured is only 1/4 of the magnitude as it is in parity-striped RAIDs. This observation can be regarded as the result of enhanced access availability produced by the extra copy of the original file system. The precise degree of improvement needs more detailed study on controller I/O operations. For the purpose of estimating the overall performance, it can be accepted to use $1/r^2$ in the calculation, with r^2 subject to a maximum of N_C.

Therefore for a mirrored RAID containing r copies, the degradation by latency index is calculated as:

$$\text{Degradation by Latency Index} = \frac{(L_X-L_Y)/(x-y) * (N_C-1)^2 / w}{\text{1MB / Single Client Throughput}} \qquad \text{.........(23)}$$

where 'w' is the smaller of r^2 and N_C, and values for L_X, L_Y, x, y and 'Single Client Throughput' are obtained from actual BBCMeter test results

5.3 Final model

Again by adding up both indexes, the final degradation can be obtained and used for predicting the overall throughput performance. Fig 13 shows the predicted random access throughput for a 2-mirror, 3-disk PC-DAS system plotted on the same graph with the actual performance.

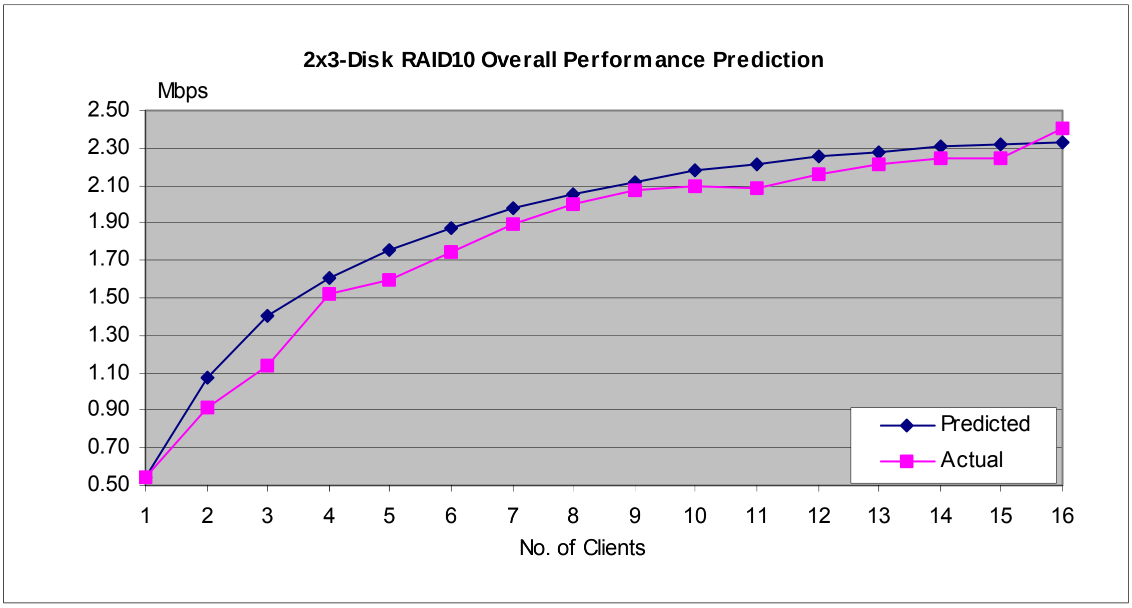

Fig 13 - Comparison between the predicted and actual Performance for 2x3-Disk RAID_10 PC-DAS system

6 Conclusion

The non-linear saturation of performance over large numbers of clients, suffered by all storage systems, is a combined result of the natural degradation by probability and the controller hardware properties. While the former is caused by the randomness of the access operations, which is governed by the rule of physics, the latter depends on the electronics of the controller that can be improved by better circuitry designs.

The model established based on this hypothesis is able to make accurate predictions from two samples, for the random access performances of any RAID systems. It is useful when actual tests of a server with certain numbers of clients are unable to be carried out. The degradation by probability index calculated can also be referenced as a benchmark when analysing the performance of a specialist storage controller.

It is clearly seen from this paper that some specialist servers do have great advantages over the ones built on consumer technologies in terms of storage controller structures. However not all the professional purpose-built storage servers are suitable for media content. They must be evaluated with BBCMeter before any conclusion can be made.

It is also recommended for small production groups that when storage capacity and synchronisation are not greater issues than random access performance, centralised RAID solutions should be reconsidered in favour of peer-to-peer configurations.

The study on media storage servers will gain more momentum with the growing demand for performance. The next step towards more comprehensive understanding on media servers will go beyond the hypothesis set in this paper and produce more detailed analysis of the performance for various types of storage systems.

7 Acknowledgement

It has been an ongoing study within the BBC R&D to try to build a workable model of media contents servers. Before this paper, Mr. R.Walker did a series of research and testing experiments on the performance of numerous server systems, and published several internal reports, which this paper used extensively as reference. Many experimental results, conclusions, concepts and methodologies are also inherited from the earlier works of R.Walker. Most importantly the BBCMeter software and access time calculation are the foundations to the entire study. The author is very grateful for the knowledge and help gained from him.

The other person important to this paper is senior R&D engineer C.Chambers, who leads the project team investigating storage and networks. During the whole period of this research, he offered the greatest support and encouragement, ensuring the final completion of the paper.

The author would also like to thank all of his colleagues in the BBC R&D for the kind help they have offered.