A Note on SCSI Bus Waits

Alexandre Brandwajn

Jack Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064 USA
alexb@cse.ucsc.edu

Abstract

In the SCSI-2 standard, the unique IDs of devices on the bus define a fixed priority whenever several devices compete for the use of the bus. Although the more recent SCSI-3 standard specifies an additional fair arbitration mode, it leaves such fair mode an optional feature. Despite a number of allusions to potential unfairness of the traditional SCSI bus arbitration scattered in the trade literature, there seem to be few formal studies to quantify this unfairness.

In this paper, we propose a simple model of SCSI bus acquisition in which devices on the bus are viewed as sources of requests with fixed non-preemptive priorities. We use the model to assess the expected extent of unfairness, as measured by the mean bus wait, under varying load conditions. Effects of tagged command queueing are not considered in this note. Numerical results obtained with our model show that there is little unfairness as long as the workload is balanced across devices and the bus utilization is relatively low. Interestingly, even for medium bus utilization a significant fraction of bus requests find the bus free which might correlate with the service rounds noted in a recent experimental study. For unbalanced loads and higher bus utilization, the expected wait for the bus experienced by lowest priority devices can become significantly larger than the one experienced by highest priority device. This appears to be especially true if the higher priority devices have higher I/O rates and occupy the bus for longer periods. As might be expected, even for balanced workloads, unfairness tends to increase with the number of devices on the bus.

1. Introduction

The SCSI (Small Computer System Interface) bus is widely used for interconnecting disks in higher-end workstations, as well as at the back end of medium to large storage controllers (e.g. EMC 8000 series [1]) where SSA (e.g. [2, 3]) and Fibre Channel (e.g. HDS 9900 [4]) are competing interconnection architectures.

In the SCSI-2 standard [5], the unique IDs of devices on the bus define a fixed priority whenever several devices compete for the use of the bus. The more recent SCSI-3 standard [6] specifies an additional fair arbitration mode but leaves it an optional feature. Despite a number of allusions to potential unfairness of the traditional SCSI bus arbitration scattered in the trade literature (e.g. [7, 8]), there seem to be few formal studies to quantify this unfairness. Recently, Barve *e.a.* [9] presented the results of a detailed study of workstation I/O under synthetic I/O workloads. With one exception, the configurations reported in [9] consisted of a small number of devices on the SCSI bus, and the measurement results point to convoy-like service rounds with little unfairness observed in terms of starvation of lower priority devices.

This paper proposes a simple model of SCSI bus acquisition in which devices on the bus are viewed as sources of requests with fixed non-preemptive priorities. We use the model to assess the expected extent of unfairness, as measured by the mean bus wait, under varying load conditions. Effects of tagged command queueing are not considered in this note. Numerical results obtained with our model show that there is little unfairness as long as the workload is balanced across devices and the bus utilization is relatively low. Interestingly, even for medium bus utilization a significant fraction of bus requests find the bus free which might correlate with the service rounds noted in [9]. The expected wait for the bus experienced by lowest priority devices can become significantly larger than the one experienced by highest priority device for unbalanced loads and higher bus utilization. This appears to be especially true if the higher priority devices have higher I/O rates and occupy the bus for longer periods. In addition, as might be expected, even for balanced workloads, unfairness tends to increase with the

number of devices on the bus.

We use a priority class aggregation method akin to [10] to solve approximately our model. A comparison with discrete-event simulation indicates that, for the set of values considered, the class aggregation approach tends to be considerably more accurate than the approximate Mean Value Analysis BKT method (cf. [11]).

In the next section, we describe our model and outline its solution via class aggregation.

2. A simple model of SCSI bus acquisition

Our model of SCSI bus acquisition is shown in Figure 1. There are N devices, numbered 1 through N, competing for the use of the bus (the server). Each device issues requests at a fixed priority level corresponding to its number. Device 1 has highest priority. The priority is non-preemptive so that it determines only which request gets to use the server when the bus finishes serving a request and there are several requests waiting.

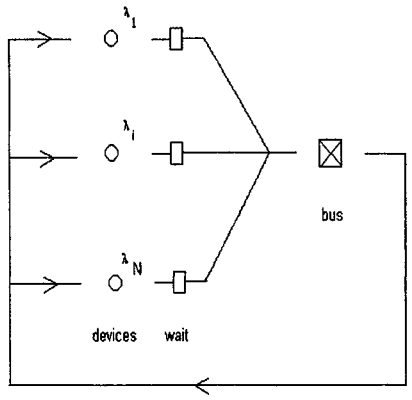

Figure 1: Finite source model of SCSI bus priorities.

We denote by $1/\mu_i$ the mean bus service time of a request issued by device i. We lump bus setup and cleanup overheads with the time device i uses the bus, and we assume that the resulting service times are exponentially distributed. We denote by λ_i the rate with which device i issues new requests for bus use when the device is idle (no request for the bus), and we assume that device idle times are exponentially distributed.

We plan to use our model to study the expected time a device has to wait before acquiring the bus under a given attained rate of I/Os for each device. Denote by Θ_i this attained I/O rate for device i, and let $\overline{w}_i$ be the corresponding expected bus wait time. In our formulation, the server (bus) utilization is known for each request class (device on the bus). We are seeking to determine the expected number of class i requests waiting for the SCSI bus, denoted by $\overline{k}_i$. Since in our model each device can only have one bus request outstanding, this expected number is also the fraction of time device i spends waiting for the bus. By Little's law [12] we have

$$\overline{w}_i = \overline{k}_i / \Theta_i . \qquad (1)$$

Note that we must have

$$(1 - \overline{k}_i - u_i)\,\lambda_i = \Theta_i , \qquad (2)$$

where $u_i = \Theta_i/\mu_i$ is the utilization of the bus by device i. In practice, when solving the model, we use a fixed-point iteration to determine the proper values for λ_i, i=1,...,N such that we have the desired attained I/O rates Θ_i for all devices.

We now briefly discuss the class aggregation approach, assuming that the total number of priority classes (devices on the bus) exceeds three. We focus on priority level i, i=2,...,N-1, and we use the subscripts h and l, respectively, to refer to priority levels higher and lower than the selected focus level. We denote by n_i the current number of class i requests in the system (waiting or using the bus), and by n_h and n_l the numbers of higher and lower priority requests in the system, respectively. We describe the system by $p(n_i,n_h,n_l,s)$, the steady state probability that there are n_i requests of class i (n_i=0,1), n_h (n_h=0,...i-1) requests of priority higher than i, n_l (n_l=0,...N-i) requests of priority lower than i, and that the device currently using the bus is of priority s (s=i, h, l). Clearly, s is meaningless when the server is idle ($n_i=n_h=n_l=0$).

To be able to write the balance equations for $p(n_i,n_h,n_l,s)$, we need conditional rates of increase and of decrease for n_h

$$\alpha_h(n_i,n_h,n_l,s) = \sum_{j=1}^{i-1} [1 - \overline{m}_j(n_i,n_h,n_l,s)]\,\lambda_j, \qquad (3)$$

$$\beta_h(n_i,n_h,n_l,s) = \sum_{j=1}^{i-1} q_j(n_i,n_h,n_l,s)\,\mu_j, \qquad (4)$$

where $\overline{m}_j(n_i,n_h,n_l,s)$ is the conditional expected number of class j requests in the system, and $q_j(n_i,n_h,n_l,s)$ is the conditional probability that device j is using the bus given n_i, n_h, n_l, and s. Similarly, we need analogous rates for lower priority devices

$$\alpha_l(n_i,n_h,n_l,s) = \sum_{j=i+1}^{N}[1-\overline{m}_j(n_i,n_h,n_l,s)]\lambda_j, \quad (5)$$

$$\beta_l(n_i,n_h,n_l,s) = \sum_{j=i+1}^{N} r_j(n_i,n_h,n_l,s)\,\mu_j, \quad (6)$$

where $r_j(n_i,n_h,n_l,s)$ is the conditional probability that a lower priority device j is using the bus.

As an approximation, we assume that the quantities pertaining to higher priority classes depend only on n_h, and those pertaining to lower priority classes depend only on n_l. Since we know the server utilization contributed by each class of requests, a simple "natural" approximation is

$$q_j(n_n, s=h) \approx u_j / \sum_{k=1}^{i-1} u_k$$

and (7)

$$r_j(n_n, s=l) \approx u_j / \sum_{k=i+1}^{N} u_k .$$

To approximate $\alpha_h(n_h)$, we postulate an arrival rate of the form

$$\alpha_h(n_h) = (N_h - n_h)\gamma_h, \quad (8)$$

where N_h = i-1 is the total number of higher priority devices. We can set γ_h to maintain the correct average rate of arrivals for higher priority requests, viz.

$$[N_h - \sum_{j=1}^{i-1}(\overline{k}_j + u_j)]\,\gamma_h = \sum_{j=1}^{i-1}\Theta_j. \quad (9)$$

Similarly, we approximate $\alpha_l(n_l)$ as

$$\alpha_l(n_l) = (N_l - n_l)\gamma_l, \quad (10)$$

where N_l = N-i, and we set γ_l so as to maintain the correct average rate of arrivals for lower priority requests

$$[N_l - \sum_{j=i+1}^{N}(\overline{k}_j + u_j)]\,\gamma_l = \sum_{j=i+1}^{N}\Theta_j. \quad (11)$$

The average numbers of requests waiting to use the bus, $\overline{k}_j$, are not known until we solve our models for focus priority level i = 2,...,N-1, and at the same time they are needed to solve these models. This suggests the use of a fixed-point iteration scheme where we solve the set of N-2 (i=2,...,N-1) three class models until the values of $\overline{k}_j$ stabilize. In practice, this tends to occur in just a few iterations. No such iteration is needed with three or fewer devices on the bus.

The balance equations for $p(n_i,n_h,n_l,s)$ together with the normalizing condition

$\sum_{n_i=0}^{1}\sum_{n_h=0}^{N_h}\sum_{n_l=0}^{N_l}\sum_{s} p(n_i,n_h,n_l,s) = 1$ form a linear system of moderate dimensions, and can be solved by any of a number of methods. We use a solution based on conditional probabilities according to the probability identity $p(n_i,n_h,n_l,s) = p(n_i,s|n_h,n_l)p(n_h|n_l)p(n_l)$.

We apply the results of our model to study the expected bus wait experienced by devices at different priority levels. In this context, it is of interest to assess the probability that a given device experiences no wait, i.e., finds the bus free. A simple conservation argument allows to express this probability for device i as

$$s_i = (1-u)/(1-\overline{k}_i - u_i) \quad (12)$$

where u is the total bus utilization, u_i is the bus utilization contributed by device i, and $\overline{k}_i$ is the expected number of device i requests waiting for the bus.

In the next section, we present numerical results obtained from our model.

3. Numerical results

We start by a bus configuration with four devices under balanced workload. The data rate is taken to be 20 MB/s, and the setup and cleanup overheads are each assumed to be 0.5 ms per I/O. We show in Figures 2 through 6 the expected time each device has to wait for bus acquisition, as well as the probability that the device finds the bus free, for I/O transfers of 4 Kbytes, 8 Kbytes, 16 Kbytes, 32 Kbytes and 65 Kbytes, respectively.

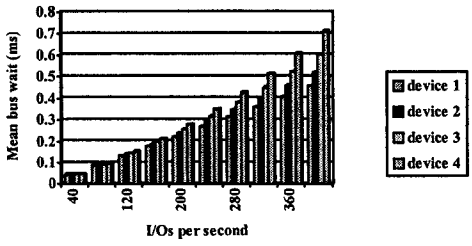

Figure 2a: Mean bus wait time with 4 Kbytes transfers.

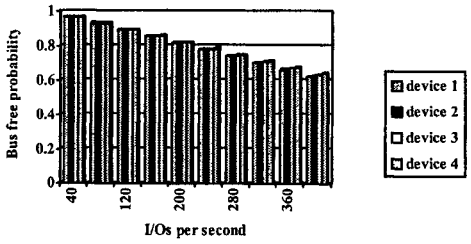

Figure 2b: Probability device finds bus free with 4 Kbytes transfers.

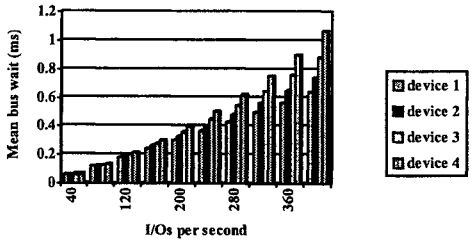

Figure 3a: Mean bus wait time with 8 Kbytes transfers.

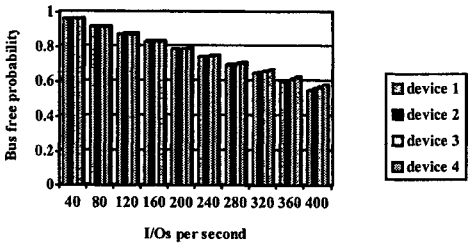

Figure 3b: Probability device finds bus free with 8 Kbytes transfers.

We observe that, for the balanced workloads under consideration, the relative difference between the expected bus wait for highest priority device and lowest priority device becomes significant only under heavy bus loads, say, for bus utilization approaching or exceeding 45%. It is interesting to note that a large fraction of bus requests find the bus free, even under moderately heavy I/O rates. It is also interesting that, as the bus load increases, lower priority devices have a somewhat higher probability of finding the bus free. The implication is that when they do find the bus busy, their expected waiting time is longer than for the higher priority classes. Because requests are generated by a small number of non-preemptive priority sources which cannot generate a new request if they are waiting for or using the bus, there seems to be a fair amount of "self-regulation" in the model. This can be correlated with the convoy-like behavior described in [9] although it is not clear how much of that behavior is attributable to any particularities of the workload generators or some other features not accounted for in our model.

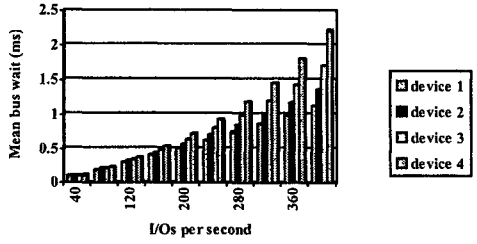

Figure 4a: Mean bus wait time with 16 Kbytes transfers.

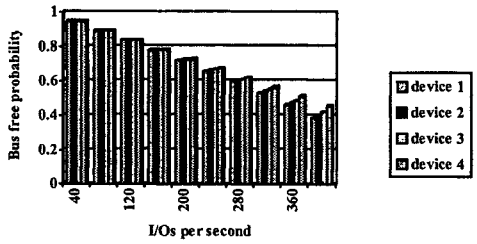

Figure 4b: Probability device finds bus free with 16 Kbytes transfers.

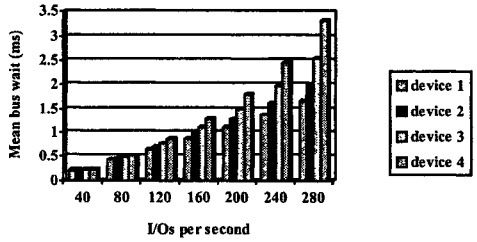

Figure 5a: Mean bus wait time with 32 Kbytes transfers.

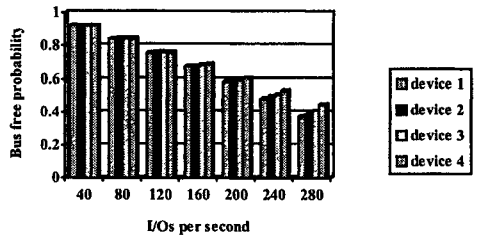

Figure 5b: Probability device finds bus free with 32 Kbytes transfers.

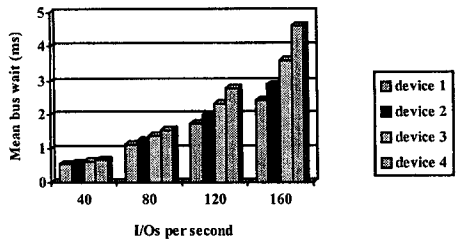

Figure 6a: Mean bus wait time with 64 Kbytes transfers.

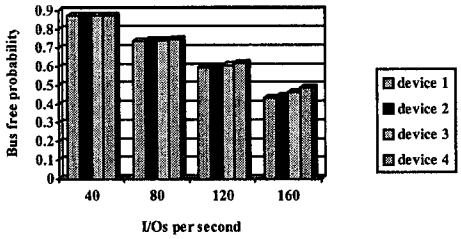

Figure 6b: Probability device finds bus free with 64 Kbytes transfers.

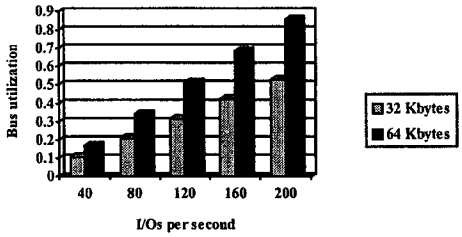

Figure 7: Bus utilization (as a fraction) with 32 and 64 Kbytes transfers.

Figure 7 shows the bus utilization for a set of I/O rates with an average transfer length of 32 Kbytes and 64 Kbytes. Intuitively, one expects the relative difference in bus waits between the lowest and the highest priority devices to tend to increase with the number of devices. Figure 8 shows an example of results with only three devices on the bus to be compared with those in Figure 6 for four devices. Note that the bus utilization in Figure 8 with a total of 120 I/Os per second is the same as in Figure 6 with the same I/O rate. Figure 9 shows the results for a slower bus with a data rate of 5 MB/s and a total of seven devices on the bus.

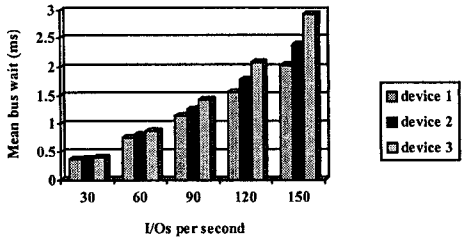

Figure 8: Mean bus wait time with 64 Kbytes transfers and 3 devices.

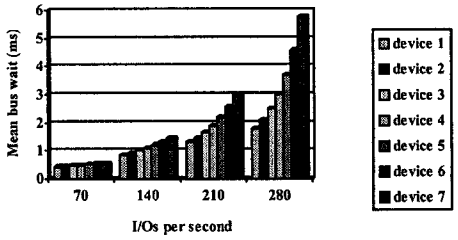

Figure 9a: Mean bus wait time with 8 Kbytes transfers and 7 devices.

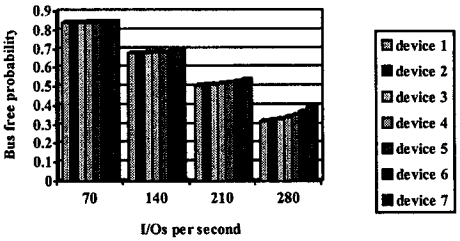

Figure 9b: Probability device finds bus free with 8 Kbytes transfers and 7 devices.

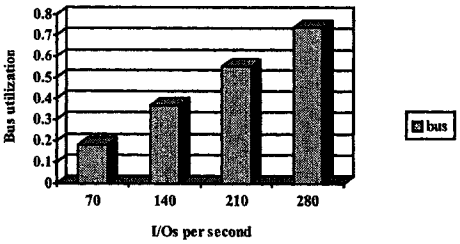

Figure 9c: Bus utilization with 8 Kbytes transfers and seven devices.

We observe, as before, that the relative difference in the bus wait times remains small as long as the bus utilization does not exceed some 40%. As the bus utilization grows, the relative difference between the highest and the lowest priority devices grows and can attain significant values. Because the workload is balanced and there is a limited number of devices on

the bus, true starvation does not seem to occur for the lowest priority device. Intuitively, one would expect the lowest priority device to be most penalized if the higher priority devices dominate the bus in terms of transfer lengths and bus utilization. Figure 10 shows an example of an unbalanced workload with four devices. The mean transfer lengths are 64 Kbytes, 32 Kbytes, 16 Kbytes and 8 Kbytes for devices 1, 2, 3, and 4 respectively. For simplicity, the I/O rates were kept the same for all four devices. The bus data rate was taken to be 20 MB/s.

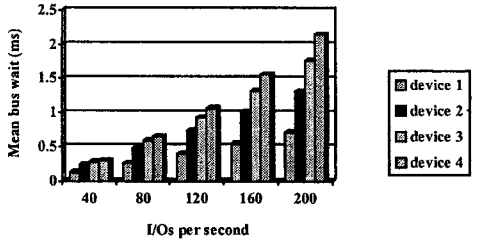

Figure 10a: Mean bus wait time with different length transfers.

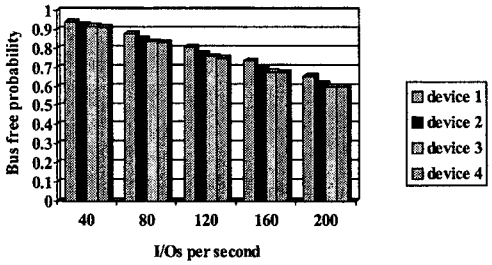

Figure 10b: Probability device finds bus free with different length transfers.

Here, the expected wait for lowest priority device can grow to be three times that for the highest priority device, but, because there are only four devices on the bus, it is difficult to totally starve the lowest priority device. Clearly, the bus wait penalty could become very substantial with a higher number of devices sharing the SCSI bus.

We used a class aggregation approach to solve our model. There are several other approximation techniques that can be applied to our simple model. One of them is an approximate Mean Value Analysis method proposed by Bryant, Krzesinski and Teunissen (cf. [11]). We refer to this method as the BKT AMVA. For the workloads explored in this note, the BKT AMVA method is, in our experience, not only generally significantly less accurate than the class aggregation approach, but seems to exhibit erratic errors. Table 1 shows an example of the accuracy of both approaches for a system with four devices studied in Figure 10 at a bus utilization of some 50%.

Solution Method	Class	Mean Wait	+-	Mean Number	+-
simulation	1	0.724	0.043	0.250	0.003
	2	1.316	0.049	0.198	0.003
	3	1.785	0.049	0.181	0.004
	4	2.187	0.052	0.180	0.003
class aggregation	1	0.708		0.249	
	2	1.304		0.197	
	3	1.758		0.179	
	4	2.148		0.178	
BKT MVA approx.	1	1.320		0.280	
	2	1.430		0.203	
	3	1.465		0.164	
	4	1.536		0.147	

Table 1: Example of accuracy of class aggregation and AMVA.

4. Conclusion

We have presented a simple model of SCSI bus arbitration with fixed device priorities. In our model, the devices on the bus are viewed as sources of requests with non-preemptive priorities, and the bus is the server. We apply a class aggregation approach to obtain an approximate solution to our model. We use our model to assess the degree of unfairness that can be expected on the SCSI bus due to the fixed priority scheme. Our results indicate that there is little unfairness as long as the workload is balanced across devices and the bus utilization is relatively low. We find that, even for medium bus utilization, a significant fraction of bus requests arrive when the bus is free which might correlate with the service rounds noted in a recent study. The expected wait for the bus experienced by lowest priority devices can become significantly larger than the one experienced by highest priority device for unbalanced loads and higher bus utilization. This seems especially true if the higher priority devices have higher I/O rates and occupy the bus for longer periods. As might be expected, even for balanced workloads, unfairness tends to increase with the number of devices on the bus.

Our model was solved under the assumption of exponentially distributed service times on the bus. It is interesting to get some insight into the influence of the service time distribution on the SCSI bus wait. Figures 11 and 12 show simulation results for the unbalanced workload considered in Figure 10 but with constant and high variability bus service times, respectively. The high variability case uses a hyperexponential distribution of bus busy times with a coefficient of variation of 3.

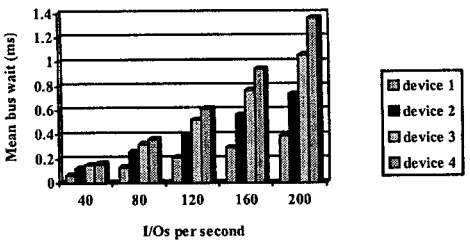

Figure 11: Mean bus wait with different length constant transfers

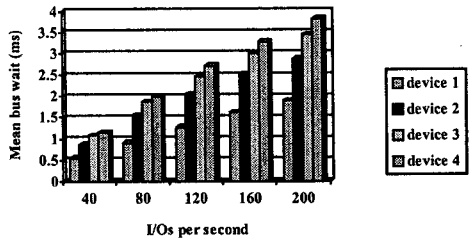

Figure 12: Mean bus wait with different length higher variability transfers

Interestingly, we observe that while constant transfers result in a shorter expected wait for the bus, they appear, in relative terms, to penalize more lower priority devices. Higher variability transfers, on the other hand, result in a longer expected wait for the bus but with a relatively smaller penalty for lower priority devices. These effects appear more pronounced as the bus utilization increases.

5. Bibliography

1. EMC Symmetrix 8000 Series, information available on www.emc.com, 2001.

2. IBM Enterprise Storage Server, IBM Publication SG24-5665, 1999.

3. StorageTek V960 Shared Virtual Array Disk System, information available in pdf format on www.storagetek.com, 2001.

4. Hitachi Lightning 9900 Series Architecture Guide, HDS Publication available in pdf format on www.hds.com, 2001.

5. SCSI-2, X3T9.2 Project 375D Small Computer System Interface-2, Rev. 10 ISO/IEC 9316-1, April 19,1996.

6. SCSI-3, Working Draft American National Standard T10, Project 1365 D, Rev. 9, Jan, 2002.

7. Chen, S., Towsley, D.: A performance evaluation of RAID architectures, IEEE Transactions on Computers 45, 10, 1116-1130 (1996).

8. Sinclair, J.B., Tang, J., Varman, P.J.: Placement-related problems in shared disk I/O, vol. 362 of The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, 271-289 (1996).

9. Barve, R., Shriver, E., Gibbons P.B., Hillyer, B.K., Matias, Y., Vitter, J.S.: Modeling and optimization of multiple disks on a bus, Performance Evaluation Review 27, 83-92 (1999)

10. Sauer, C., Chandy, K.: Approximate Analysis of Central Server Models, IBM J. Res. Devel. 19, 301-313 (1975).

11. Eager, D., Lipscomb, J.N.: The AMVA Priority Approximation, Performance Evaluation 9, 173-193 (1988).

12. Stidham, S.: A last word on $L = \lambda W$, Operations Research 22, 417-421 (1974).