IOmeter Performance Comparison of SBOD and MBOD
FRANCK CHEVALIER, ROLAND IBBETT
Institute for Computing System Architecture, School of Informatics, University of Edinburgh, The King’s Buildings, Edinburgh, EH9 3JZ
UK
TIM COURTNEY
Xyratex Technology Ltd, Langstone Technology Park, Langstone Road, Havant, PO9 1SA, UK

Abstract: In this paper, the performance of the emerging Switched Bunch Of Disks (SBOD) RAID architecture is compared to the more traditional Managed Bunch Of Disks (MBOD) using both analytical and simulation models. The benchmark used for this comparison is the conventional IOmeter that enables Throughput and IOPS of these systems to be compared. The SBOD configuration that maximises its performance is first investigated. Using both analytical and simulation models, it was found that the SBOD performs between 104% to 275% better than the MBOD, depending on the size of the requests in the IOmeter traffic workload.

Key-Words: - RAID, MBOD, SBOD, Storage Systems, IOmeter Benchmark, Interconnect Architectures, Fibre Channel.

1. Introduction

Redundant Arrays of Inexpensive/Independent Devices/Disks (RAID) have been used for hard-disk storage systems for many years [1]. Enterprise class RAIDs typically use the Fibre Channel Arbitrated Loop (FC-AL) protocol [2, 3] to move SCSI commands between disks and RAID controllers. Initially, the disks were organised in a daisy-chain style loop, known as Just a Bunch Of Disks or JBOD. This architecture suffers from a single point of failure as a failure from any disk or link in the system results in the overall system being unavailable. To overcome this, a hub containing port-bypass and management functionality can be placed in the system. This architecture is commonly referred to as Managed Bunch of Disks or MBOD. This results in a system that is tolerant to failures at the expense of an additional delay introduced by the hub. In systems containing many devices, there is already a large delay as communications travel around the loop. To overcome these delays, a Switched Bunch Of Disks (SBOD) configuration was recently introduced. In an SBOD, a simple cross-bar switch replaces the port-bypass function of the hub. When two devices wish to communicate, the switch is used to create a virtual loop, just consisting of these devices with the switch as an interposer. With such a system, the observed loop delay no longer scales with the number of devices on the loop but remains constant.

In this paper, the performance of the SBOD will be compared to the more traditional MBOD using the IOmeter [4] as the reference benchmark.

2. IOmeter Benchmark

Until 2001, the IOmeter benchmark [4] was the only standard benchmark available to compare the performance of storage systems. Since then, the Storage Performance Council (SPC) has introduced another benchmark that produce more random workloads – i.e. the SPC-1 benchmark [5]. The performance benefits of the SBOD architecture under SPC-1 workload are presented in [6].

This paper focuses on fully sequential workloads as:
1. Many applications generate such a traffic profile, e.g. backup and video streaming.

2. IOmeter is still the reference benchmark used by many customers when purchasing storage systems.

The IOmeter test determines two system parameters: the IOPS (Input Output requests Per Second) and the Throughput. To measure Throughput and IOPS, the IOmeter issues fixed-size sequential I/O requests to the system under test, while maintaining the number of I/O requests in the storage system input queue constant at 75 commands. The relationship between Throughput and IOPS is defined below:

\[
\text{Throughput} = \text{IOPS} \times \text{request\_size} \quad (1)
\]

IOPS and Throughput have contradictory aims: the former is maximised by concurrently reducing both I/O command size and strip size while the latter is maximised by increasing both I/O command size and strip size in the RAID sub-system. In this paper, we investigate the IOPS and Throughput for a number of strip sizes with all I/O requests being full strips.

### 3. SBOD Operation

While the operation of the MBOD is fully defined by the FC_AL standard [2, 3], the operation of the SBOD is not so intuitive, although it is designed to be fully interoperable with devices using the FC-AL standard. As opposed to the MBOD, the SBOD switch [7] we are investigating is not transparent to the Fibre Channel protocol and plays an active part in the FC_AL arbitration process. In the rest of the paper, Fibre Channel disks and controllers attached to the MBOD or SBOD switch will be referred to as AL_Nodes.

When an AL_Node wishes to arbitrate, it transmits \textit{ARB} (arbitration) words to the SBOD switch. After a short delay, the switch returns these \textit{ARB} words to the arbitrating AL_Node, thereby allowing that device to gain tenancy of the loop and to transmit an \textit{OPN} (open) word. This AL_Node is known as the Initiator. When the switch receives an \textit{OPN} word, it first determines if the port of the target AL_Node is busy or not. If the target port is busy the switch is able to transmit a \textit{CLS (close)} to the initiator to terminate the connection. If the target port is not busy, the switch will connect the initiator port and the target port together and forward the \textit{OPN} to the target device. This forms a virtual FC_AL with two devices in the loop, with a delay in the transmission path reduced to the latency across the switch.

As apposed to the MBOD, the SBOD architecture does not naturally implement a fairness algorithm to allow fair/unfair access to all competing AL_nodes (disks and controllers). Hence some fairness mechanism must be built into the switch instead. In the SBOD considered in this paper, a fairness algorithm is implemented using queues on each port of the switch to keep a note of the AL_nodes that tried to open the currently busy device. As we shall see in the next section, the fairness algorithm in the switch plays a crucial role in maximising the system’s performance.

### 4. SBOD Configuration

Because the loop round trip delay in the SBOD is so small, the main limiting factor is disk latency and not bandwidth. Hence, making sure that each request is treated as sequential will maximise the system’s performance. A request being sent to the disk is considered ‘sequential’ when the disk only needs a small incremental rotation to access the data compared to the command that is being processed. In the 15krpm Fibre Channel disks we considered for this study, the difference in response time (for a 4kbytes request) between sequential and non-sequential requests is a factor of 7 (5 ms average response time for non-sequential and 0.7 ms average response time for sequential request). It is therefore crucial that when a new request arrives in the disk queue, the request that was processed before it has not completed. If it has completed, the disk will treat the incoming request as non-sequential and significantly decrease the system performance. Hence empty queues on the disks will significantly decrease system performance.

When using the IOmeter traffic generator, the disk queues in an SBOD system are allowed to
empty for 2 main reasons:

a) Disks are allowed to empty themselves because no new requests are generated in time to replace the completed requests

b) The average number of requests per queue is small due to insufficient offered workload.

In Section 3, it was shown that the SBOD cannot implement an intrinsic arbitration policy. Instead the fairness algorithm is implemented by the manufacturer using different queues associated with each switch port. To understand the impact of the fairness policy algorithm, let us assume that there is no policy implemented in the switch and that all disks can win arbitration and empty their queues before the RAID controller accesses the SBOD to transmit its new requests to the disks. This is illustrated in Figure 1(a).

In such a scenario, requests are sent in bursts of 75 to the disks (phase 1, 2 and 3), and each new burst of requests finds the disk with empty queues. This results in the requests being treated as non-sequential. This results in idle times in the SBOD, significantly decreasing the Throughput as the system spends its time waiting for the disks to complete their requests.

Because the IOmeter benchmark restricts the total number of outstanding commands to 75, no new commands can be generated in the idle periods, significantly underloading the tested system.

In contrast, Figure 1 (b) shows the Throughput of a system with an unfair RAID Controller. Having arbitration priority over the disks, the RAID controller can send a new request for each completed response received from the disks. This, on average, ensures that each disk in the system has a constant number of requests in its queue, provided that each disk is equally likely to receive a request.

Taking the example of an N disk system, there are on average, 75/N processing requests in each disk queue (at any time, there should be 75 outstanding requests in the system due to the Iometer benchmark definition.). Since the number of outstanding commands cannot be increased in the Iometer benchmark, the number of active disks has to be kept reasonable. If we consider a 30 disk SBOD system where each disk is equally likely to receive a request, the average number of requests in each disk queue is 2.5, and hence there is a high probability that the queue will become empty, decreasing the Throughput.

In an MBOD, this phenomenon is less apparent as the performance of the system is more limited by the communication bandwidth than by the disk latency.

5. Experimental Setup

The simulation tool used to perform the experiments is HASE, developed by Edinburgh University [8]. HASE provides a discrete event simulation environment in which to model the storage system under test. As previously described [9], we use an abstraction from the full FC_AL communication specification that provides accurate results with simulation time running only 10-40x slower than real time; this compares to 100,000x slower for simulating a full FC_AL system.

The RAID architectures analysed in this paper are shown in Figure 2 and 3. In the MBOD topology there is a single RAID controller, as there can only be one active communication between a disk and a controller. In practical situations, a second controller is added to provide failover protection but, under normal
operating conditions, this additional controller simply adds to the delay around the loop. In the case of the SBOD the switch allows any number of simultaneous communications and so the second controller is modelled.

In the SBOD model, the two RAID controllers are given I/O instructions such that one controller issues commands to one half of the attached disks and the other controller issues commands to the other half of the disks. This is how such systems are configured in practice since it removes contention and blocking. Under failover conditions, the remaining controller will access all the drives. In this configuration, the SBOD switching architecture behaves as two independent systems, each addressing half of the disks, working through a single switch unit.

In the experiments, an IOMeter traffic generator was used to generate I/O commands. This generator feeds commands into the RAID controllers which then translate the high level accesses (volume addresses) to strip level accesses (disk level addresses).

The operation of the RAID controller is based on the specification for a commercially developed RAID controller. In our models, the cache in the RAID controller has been disabled to allow us to focus solely on the performance of the network connecting the controller to the disks. The characteristics of the SBOD switch and the MBOD hub are also based on commercial devices. The final component is the 15 Krpm Fibre Channel disk which is modelled as a look-up table of delays. This look-up is based the total number of commands currently in the disk command queue and on whether or not the requests are sequential. The look-up data was obtained from experiments on real drives. Each of the model components has been designed to be parameterisable so that the same model can be used in each simulation.

6. Analytical Models

To validate the simulation results, the MBOD and SBOD maximum Throughput and IOPS were approximated using a simple analytical model. The models presented in this paper give the upper limit of the MBOD and SBOD IOPS by considering the communication bandwidth limitations. First, a few variables are defined (Table 1).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Total number of AL nodes</td>
</tr>
<tr>
<td>$k$</td>
<td>I/O request size (bytes)</td>
</tr>
<tr>
<td>$Tx_rate$</td>
<td>Transmission rate in bits/s</td>
</tr>
<tr>
<td>$Node_latency$</td>
<td>Time for each AL node to process a word (number of FC words)</td>
</tr>
<tr>
<td>$MBOD_latency$</td>
<td>Time for the MBOD to transmit a word from an input port to the next output port (number of FC words)</td>
</tr>
<tr>
<td>$SBOD_setup_time$</td>
<td>Time for the SBOD to set up the connection (number of FC words)</td>
</tr>
<tr>
<td>$SBOD_latency$</td>
<td>Time for the SBOD to transmit a word from an input port to the next output port (number of FC words)</td>
</tr>
</tbody>
</table>

Table 1: Parameter Definition

Based on hardware characteristics, the following values were chosen for each parameter defined in Table 1.

- $Tx\_rate = 2.125\ Gbit/s$
**Node latency = 4 FC words**
**MBOD latency = 4 FC words**
**SBOD latency = 6 FC words**
**SBOD_setup time = 6 FC words**

The same values were used in the simulation models. The main assumptions in the derivation of the analytical models are the following:

- There is always at least 1 AL_Node wanting to arbitrate in the system
- Negligible propagation time – only a few meters of optical fibre connecting all AL_Nodes.
- Error-free transmission

Using the FCP protocol description and the Fibre Channel Arbitrated Loop (FC-AL) standards, it can easily be shown that the IOPS for the MBOD is given by:

\[
\text{IOPS}^{-1}_{\text{MBOD}} = \frac{(40N+78+\lceil k/4 \rceil 512) * 527}{40/Tx\_rate} \quad (2)
\]

and that the IOPS for the SBOD with dual RAID controller is given by:

\[
\text{IOPS}^{-1}_{\text{SBOD}} = \frac{(182+\lceil k/4 \rceil 512) * 527 * 40/Tx\_rate}{2} \quad (3)
\]

It can be observed that the response time in an SBOD system is independent of the number of AL_Nodes (N). This is in marked contrast to the MBOD system. This expected result highlights one of the advantages of the SBOD over the MBOD: the response time is independent of the number of AL_Nodes.

Equation 2 and 3 can be substituted into Equation 1 (given in Section 2) to derive the analytical Throughput of both architectures.

### 7. Experimental results

#### 7.1. MBOD results

In our pre-experiments, it was found that the SBOD response was maximised when limiting the number of active disks to 14 and ensuring that RAID controller arbitration policy was unfair (RAID controller has priority over the disks). This is based on the observations made in Section 2. We will use this configuration in the rest of the paper.

In Figure 4, the Throughput for the MBOD architecture was simulated for systems with 14, 20 and 30 disks. 14, 20 and 30 disks were chosen because they relate to disk enclosures that are available today. For each system size, different I/O request sizes were considered: 4k, 8k, 16k, 32k and 64kbytes. Although it is not shown (for clarity), the analytical model matches our simulation results within 5%.

Our first observation is that the Throughput for the MBOD architecture varies significantly with the total number of disks in the system. The Throughput increases as the number of disks decreases. This first obvious result is due to the fact that an increased number of disks results in an increased loop round trip delay.

The second obvious observation is that the MBOD Throughput increases with the request size. This is explained by the fact that the larger the request, the less the overhead. As the I/O size increases, performance is dominated by the data transmission time and the protocol overhead becomes increasingly negligible (in MBOD the 3\(\mu\)s round loop time is not significant compared to the 300\(\mu\)s transmission time of a 64kbytes data frame). The same remark is also valid for IOPS parameter.
7.2. SBOD results

In Figure 5, it can be observed that the Throughput improvement of the SBOD over the MBOD is greater for smaller I/O request and a larger number of disks. The SBOD Throughput improvement over the MBOD ranges from 104% (14 disks and 64k requests) to 275% (30 disks and 4k requests).

It can be noted that the SBOD is particularly attractive in comparison with the MBOD for system with a large number of disks issuing small requests. This is explained by the fact that in large systems, the SBOD round trip delay (SBOD latency) remains constant and independent of the number of AL_Nodes. Furthermore, when the request size is small, the transmission overhead of the MBOD is significantly larger than that associated with the SBOD for the same reasons as explained above.

8. Conclusions

In this paper, an evaluation of the performance of MBOD and the SBOD architectures was presented, considering the IOmeter benchmark as a traffic generator. This evaluation was based on a different number of disks and on different request sizes.

It was first shown that the configuration of an SBOD is crucial to maximize its performance:

- The SBOD switch should implement an unfair arbitration policy for the RAID controllers
- The number of active disks should be kept to a reasonable number (14).

If the two above conditions are not met, the IOmeter can severely “underload” the SBOD and the performance improvement would not be so evident.

We also derived an analytical model to validate our simulation results. The analytical model gave the upper limit in terms of IOPS for both the MBOD and the SBOD, solely considering the communication bandwidth available in both systems.

Finally, we showed simulation results for both systems which indicated that the SBOD offers IOPS and Throughput improvements, compared to the MBOD, in the range of 104% to 275% depending on the system configuration.

9. Acknowledgements

This work is supported by the UK EPSRC grant GR/S28143 and is funded under the DTI/EPSRC LINK Information Storage and Displays programme.

10. Reference