PROCEEDINGS OF SCIENCE

New Directions in Storage, Hard-disks With Built-in Networking

Patrick Fuhrmann[1], Yves Kemp, Tigran Mkrchtyan, Paul Millar
DESY
Notkestr. 85, D-22607 Hamburg, Germany
E-mail: `firstname.lastname@desy.de`

Christopher Squires
HGST
5500 Central Ave., Suite 220, Boulder, CO 80301
E-mail: `christopher.squires@hgst.com`

PoS(ISGC2015)029

The dCache project provides software allowing sites to offer a reliable storage service using heterogeneous storage components, including SSD, HDD and tape. The software provides native support for many protocols and has a number of advanced data-management features that allow scaling into the multi-petabyte capacity domain. Based on recent advances in hard-disks and system-on-a-chip (SoC) designs HGST developed a novel form of hard-disk: a disk that includes both, a network interface and an ARM processor., allowing the individual disk to run an operating system and to communicate with other nodes autonomously using wired Ethernet. No additional hardware or infrastructure is required. The dCache team took advantage of their early access to those devices, investigating a possible deployment in grid and cloud environments. By operating dCache software and observing the system's behavior we investigated how petabyte-sized storage infrastructures, based on these disks, can be build, even considering changes to the dCache software. We present possible deployment scenarios for those new devices and compare them to existing setups at the Deutsches Elektronen-Synchrotron (DESY) research centre, where direct attached RAID systems are used. The results of our initial investigation are presented along with an outline of future work.

International Symposium on Grids and Clouds (ISGC) 2015
15 -20 March 2015,
Academia Sinica, Taipei, Taiwan

[1] Speaker

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike Licence. http://pos.sissa.it/

1.Introduction

Current storage systems are composed of sophisticated building blocks: Large file server, often equipped with special RAID controllers, powerful and expensive CPUs, large amounts of RAM. We want to investigate whether there are options to build storage systems that are based on small, independent, low-care and also low-cost components.

If such devices exist, we want to investigate in which application scenarios they are best used and can offer both a TCO and performance advantage over classical large building blocks.

We also wanted to investigate whether other options for a massive scale-out using such novel techniques can be achieved.

It is clear that if one would use such simple devices, a stronger focus lies on the software that manages the system, as some features that are taken care of by dedicated hardware controllers must be implemented in software.

We are also interested in learning whether storage system setups based on such simple devices could enable different operational models, and how the TCO composition might change.

1.1 Software considerations

A software stack running a setup consisting of simple devices must present some advanced features:

- The software must be able to handle independent data nodes. This is the case, amongst others, for CEPH [1] and dCache [2].

- The software must be able to show a massive scale-out without bottlenecks. Again, this is the case for CEPH and dCache, amongst others, if adequate protocols are used. The protocol used in CEPH to allow massive scale-out is implemented in the robust hashing code of the CEPH client. dCache offers such a scaling through standard protocols like NFS 4.1/pNFS, GridFTP or WebDAV.

- As running a system with a high number of simple data nodes, the frequency and the severity of failures will increase. The management software must be able to react to these and offer both operational safety and data integrity. Both CEPH and dCache continue operation when a data node is failing. Data integrity can be achieved via several means. The simplest one, creating several copies of files on separated data nodes, is supported by CEPH and dCache.

Since DESY is the core of dCache development, it is natural to investigate on dCache as management software in the following. Other people already investigated on CEPH as the management software in a scenario with small data nodes [5].

1.2 Hardware considerations

Several systems with small units currently exist on the market. As examples, we cite the DELL C5000 blade series, the HP Moonshot system, and PANASAS.

These systems, however, have in common that they share much of the same infrastructure. Often they have several disks attached to the CPU. This makes these systems too large for a simple system, but too small for a serious RAID system.

Thinking this further, the smallest possible unit would be composed of only one single disk drive, to which a CPU, RAM and network interface is attached.

Up to date, two manufactures have announced such devices: Seagate with their Kinetic drive, HGST with their Open Ethernet drive [4].

We have chosen the HGST Open Ethernet (OE) drive for our investigations, as it allows for running general purpose code, which seems more easily feasible than with the Seagate Kinetic drive.

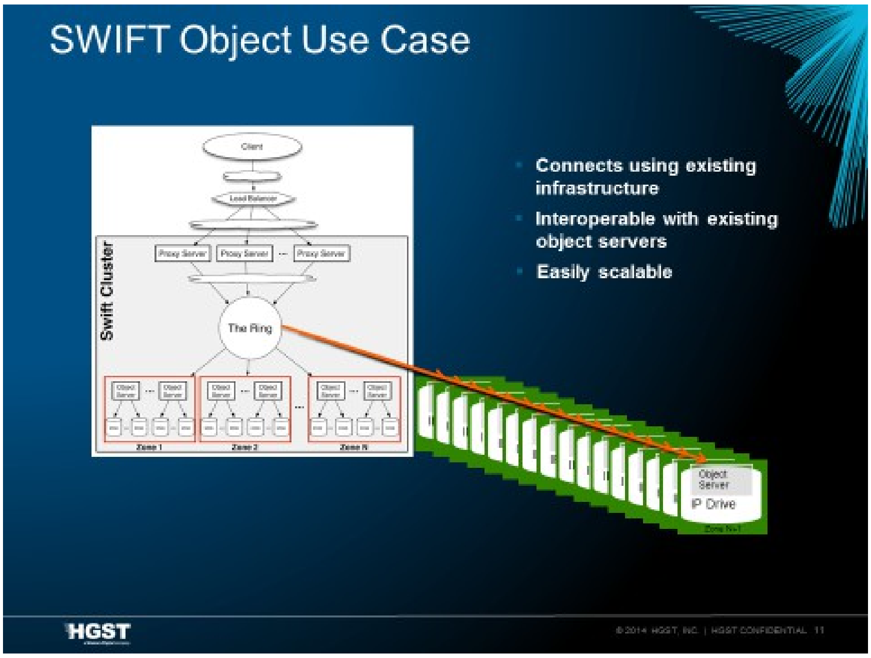
PoS(ISGC2015)029

2. HGST and Open Ethernet Drive

HGST took a holistic approach to solving upcoming media recording challenges, while keeping in mind the difficulty of changing datacenter ecosystems. Due to changes in upcoming media recording methodologies, specifically Shingled Magnetic Recording (SMR – details omitted here due to the fact that there are many white papers on this subject), there are difficulties in managing data using traditional methodologies.

By incorporating knowledge of the file/object size and other attributes into the storage subsystem, more intelligent placement of data can be achieved, which alleviates much of the burden of using SMR technology. Additionally, having this knowledge enables accelerated and more efficient caching of data.

Looking at the dCache use case below, it is easily understood how the use of HGST's Open Ethernet Drive connects to the existing network, works seamlessly with existing object servers, and scales easily. Additionally, by running the SWIFT Object server directly on the Open Ethernet Drive, the traditional object servers can be replaced, as shown in figure 3.


SWIFT Object Use Case

- Connects using existing infrastructure
- Interoperable with existing object servers
- Easily scalable

Figure 1: The SWIFT Object Use Case

Looking more in depth at total cost of ownership (TCO), HGST's Open Ethernet Drive has the ability to reduce acquisition cost is reduced by:

- Eliminating servers to manage storage
- Using backplane-based network eliminates cables (and cost)
- Removing 1 layer of datacenter switches
- Additionally, power reduction opportunities exist due to:
- Using less equipment (eliminating Servers & SAS/SATA expanders, adding Ethernet switching)
- Using more efficient processors (ARM vs. Intel)
- Reducing bandwidth requirements by using a non-blocking interface (2.5Gb Ethernet vs 12Gb SAS)
- Intelligently and independently powering off Ethernet Drives & Switch channels

Cooling efficiency can be improved because:

- Reducing power means reducing cooling costs (fewer switches and servers required)
- Higher density enables more efficient cooling

Long term reliability is improved due to:

- Backplane-based network eliminates cables (reduces cabling issues)
- Integrated storage/server enable HDD failure recognition
- Open Ethernet Drives can offload/backup data to "near-by" drives automatically (without the need for a traditional storage server to manage transfers)

Finally, opportunities exist to reduce bandwidth requirements within the data center by having the Open Ethernet Drive CPU operate on data locally and transmit only the "results", thus reducing data in-flight and thereby creating an opportunity to reduce network bandwidth requirements (and cost).

In addition to opportunities to reduce TCO, the Open Ethernet Drive can improve performance by:

- Scaling the connectivity to each drive, maximizing throughput to match the disk drive's native transfer rate, giving data centers the lowest possible "time to fill" rate
- Shortened data path improves performance
- Tuned/optimized Ethernet Drive operating system (e.g. intelligent data placement)

Finally, by utilizing HGST's Open Ethernet Drives data centers will improve scalability & flexibility by having the ability to:

- Add storage and compute in smaller increments
- Mix storage, compute, and flash nodes to meet workload requirements (future)
- Utilize storage that is designed for Software Defined Storage

Looking at the infrastructure from a developer's point of view, one will see a Linux kernel (currently Debian on prototype devices), an ARM 32-bit CPU with 512KB of L2 cache, 2GB DRAM (1794MB available to Linux), a block storage driver (enumerated as 'sda'), and an Ethernet network driver (enumerated as 'eth0'). If one is familiar with Linux, one already

PoS(ISGC2015)029

knows this device intimately. Software that one is used to seeing on your typical Linux installation is available. Users usually can run 'apt-get install' and start using the application immediately. Custom software can be compiled either using an off the shelf cross-compiler or directly on the device.

3. One current dCache setup and potential future alternative setups

We will describe a current dCache setup, the CMS instance at DESY. We then will shortly present some potential alternative setups.

3.1 The CMS dCache instance at DESY

DESY acts as a Tier-2 center and hosts the NAF for the LHC experiment CMS. In this context, DESY operates a storage system based on the dCache technology. We will shortly present the key points important for our future investigation.

Currently, this dCache instance consists of 199 pool nodes, of which the vast majority, 172 pool nodes, is composed of 2-unit boxes with 12 locally attached near-line SAS disks. A hardware raid controller operates in RAID-6 mode, so 2 disks are used for parity, 10 disks net are available as storage space. These systems contain either 2, 3, or 4 TB NL-SAS disks, depending on the date of purchase. The net capacity therefore varies between 20 TB and 40 TB. All these pool nodes are furthermore equipped with 2 Intel CPU (XEON or E5-26xx, depending on the purchase date) as well as a minimum of 24 GB RAM. Gbit or 10 Gbit network is used to export the data to the client nodes.

27 nodes have remote attached storage, either FC or SAS. As their number is small compared to the number of 2-unit boxes, we are excluding those systems from our considerations, and hence will not describe them. The same is true for four controlling and accounting nodes and ten protocol initiators machines.

3.2 Alternative 1: Decomposing the storage box using HGST OE disks and iSCSI

Instead of a 2-unit box with disks attached via SAS to a hard-ware RAID controller, one could use a very small form-factor Intel CPU box with similar specs. To each box, 12 HGST OE drives are put on the network and communicate with the Intel CPU box using the iSCSI protocol. The Intel CPU box uses software raid or similar software techniques to ensure data redundancy.

This setup presents no real advantage compared to the current setup. No cost benefit is to be expected, management complexity raises without the possibility for different operational regimes that could compensate for the higher complexity. Network usage will grow as data has to pass over the network twice.

3.3 Alternative 2: Building a CEPH pool using HGST OE disks to be used by dCache

CEPH has been shown to work on the HGST OE disks. A CEPH disk pool can export its data in three different ways: Object storage, block storage, or file system.

dCache internally saves files as objects on a file system. It would be natural to expand dCache such that it would use an underlying CEPH storage pool as object storage. All dCache pool nodes would see the same CEPH object name space, and could export these objects using any of the different dCache protocols. These pool nodes would have a similar CPU, RAM and network configuration as the currently employed pools, but would not has directly attached disks.

The advantage of this setup would be a clear separation between the low level data store and the high level data access methods. This setup would benefit from the many different possible setups of a CEPH cluster in terms of performance, data safety and data integrity methods. Scaling of capacity and scaling of performance can be achieved independently as either the CEPH pool with HGST OE disks can be enlarged for mode capacity, or the number of dCache pool nodes can be increased to boost performance. This setup also would alleviate one of CEPH's short-comings: The exposure of the same name space via different protocols simultaneously.

This setup is, however, not possible as of today. dCache would need to undergo code changes to account for a shared object name space among all pool nodes.

Some drawbacks are also clear: Again, network usage will grow as data has to pass over the network twice. Hardware RAID controller would be eliminated, the pool nodes and the rather expensive hardware in it would still exist.

3.4 Alternative 3: HGST OE disks acting as dCache pool nodes

This scenario is close to the classical scenario again, except for the fact that the pool nodes are composed of only one single drive (the HGST OE drive), and a rather small CPU system (the ARM CPU with its limited amount of RAM attached). The pool nodes again directly export data to clients using any of the dCache protocols, this time without the layer of CEPH or any RAID system – be it hardware or software.

To achieve the same data capacity as pool nodes equipped with RAID controllers and many direct attached disks, a setup consisting of HGST OE drives would need many more pool nodes – the next section will list an example setup.

Nothing would need to be changed in the dCache code. However, it has be verified that the dCache software as well as the management nodes can scale up to a substantially higher number of pool nodes. Especially the resilience manager must be well performing as multiple copies of files are the only way dCache can ensure data safety.

This setup would completely eliminate hardware RAID controllers and expensive CPUs in the pool nodes. The network usage would be optimal, as data only passes once over the network.

4. An estimation of the performance of a dCache system using HGST OE drives as pool nodes

Since the approach sketched under “alternative 3” seems promising and can be implemented with existing dCache releases, we decided to further investigate and make a qualitative estimation of the performance behavior of such a system.

PoS(ISGC2015)029

4.1 Simulation assumptions and sizing parameters

Storage systems are complex systems with many parameters to tune. Access patterns to a storage system are very diverse. In order to make an estimation of the performance of a dCache system composed of HGST OE drives as pool nodes, and compare it to the performance of a dCache system composed of RAID systems, we deliberately reduce the complexity and make some simple assumptions.

- All files written to the storage system are identical in size, 4 GB each.

- The files are equally distributed over all pool nodes in a flat, random way.

- Files are read in streaming mode only. The local performance is equivalent to the performance delivered to the clients over the network.

- The RAID setup consists of 100 RAID systems with 12 disks each and a capacity of 4 TB in a RAID 6 configuration. This is equivalent to 40 TB net capacity per pool node, or 4 PB total net capacity. Only a single copy of each file is present in the system. In total, one million files are stored within the system when being completely filled up.

- The HGST OE disks have a capacity of 4 TB each. As discussed previously, sufficient data loss protection can only be achieved using a second file copy. Two million files need to be written, which then requires a total brut capacity of 8 PB or 2000 HGST OE disks to be equivalent with the 4 PB net capacity of the RAID setup.

In summary, we are comparing a dCache setup consisting of 100 RAID file servers with a dCache setup composed of 2000 HGST OE drives.

4.2 Initial performance measurements

As input to a following simulation, we measure how the total bandwidth varies with the number of concurrent read streams. Both measurements are performed for a RAID system and an HGST OE drive. The measurement is done using IOzone [3], specifying the file size, the number of concurrent streams, and the streaming read access pattern. The RAID system tested is a DELL R720XD system with 12 2-TB NL SAS disks attached to a PERC H710 controller, configured as RAID-6. The RAID system is running ScientificLinux 6.6 with its default Kernel. The HGST OE system tested is a prototype made available to us in the HGST labs.
Figure 2 visualizes the results of our measurements, including the fits applied in the simulation.

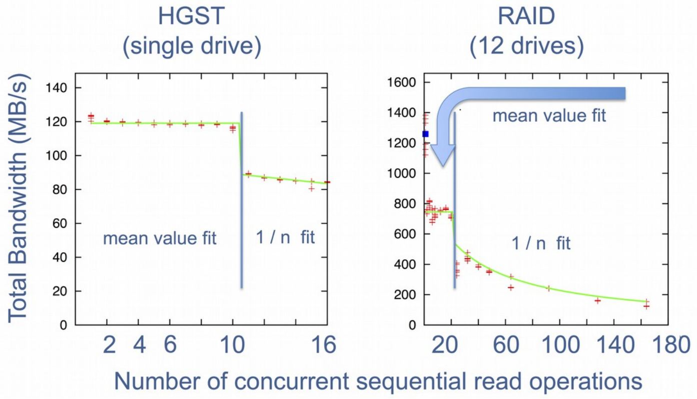


Figure 2: Total bandwidth vs. number of concurrent reads

PoS(ISGC2015)029

4.3 Simulation of the dCache setups

In the following simulation, the placement of the files is simulated. In the HGST OE case with two copies, these copies are placed on different disks.

A set of n files is chosen at random. These files should be read concurrently. Each file of this set is contained only once in this set. For each pool node, the number of files from the file set placed on it is computed. In the HGST OE case with two copies, only one pool is chosen randomly. For each pool node, the total bandwidth is looked up in the measurements performed previously. The aggregated total bandwidth is computed as the sum over all pool nodes.

These simulations are performed for a varying number 'n' of files contained in a file set. For each 'n', multiple simulation-runs are performed and the mean value of the aggregated bandwidth is computed.

The results of the simulation are shown in figure 3. One can see that the RAID curve has a steep increase up to n≈50, after which the bandwidth enters a plateau followed by a decrease after n≈1500. The HGST OE curve has a slower increase, which is however linear until n≈1000. The increase continues for n≈4000 being the maximum computed in the simulation. A cross-over point is reached at n≈750.

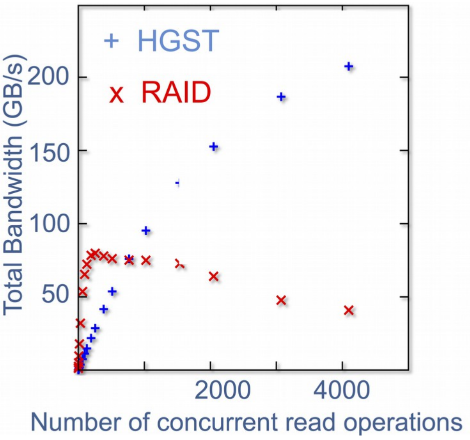


Figure 3: Total aggregated bandwidth vs. number of concurrent reads

This behavior can be explained such that single RAID systems perform better than individual HGST OE drives. With a small number of concurrent reads, these higher single bandwidth add up. Once the probability of two or more files on a pool node increases for the RAID setup, the aggregated bandwidth reaches a plateau. The advantage of the HGST OE setup is that this limit is reached only for higher number of concurrent streams.

4.4 Discussion of the simulation results and comparison with real experience

Depending on the access pattern, a dCache setup based on RAID systems or a dCache setup based on HGST OE drives is preferable performance wise.

DESY operates several dCache instances. The CMS instance, being the largest installation, has already been described. In average it serves over 1000 simultaneous reading threads, often reaching into the area of 4000 threads.

A second instance, serving the DESY photon community, only contains ~1.4 PB of data. The building blocks are however similar to those of the CMS instance. The photon instance usually serves well below 100 simultaneous reading threads.

Adding HGST OE drives to the CMS dCache, or even replacing the current RAID systems in the CMS instance would likely result in a drastic boost in overall performance.

The photon dCache instance however, wouldn't benefit from adding HGST OE drives.

We would like to stress that these results only reflect the internal disk performance in a streaming read scenario.

PoS(ISGC2015)029

5. Composition of TCO

Discussing the total cost of ownership is only possible in a qualitative way, as both the price per disk unit as well as the delivery form factor are not yet known.

For latter, we assume that a delivery unit consisting of 60 HGST OE drives, an internal Ethernet switch with up to 8x10 GE connections, power supplies and a 4 rack unit box are being evaluated by HGST.

We will compare the TCO composition for the two previously simulated setups, keeping in mind that the HGST OE dCache setup holds two copies per file. A CEPH setup with a more advanced data safety method would change the following considerations drastically.

5.1 Network

The RAID setup consists of 100 servers, resulting in 100 network ports. The HGST OE setup consists of 2000 drives, delivered in boxes of 60 disks with eight 10 GE links each. So, a maximum of 272 network ports can be used. However, depending on the desired total throughput, less links can be used. If only three 10 GE links per 60 disks are used, there is no overhead in network ports.

100 IP addresses have to be reserved for the RAID setup, whereas 2000 IP addresses are required for the HGST OE setup, not being suitable for public IPv4 addresses. One either has to use IPv6 or private IPv4 addresses routed within DESY only – dCache supports both.

5.2 Power consumption

The RAID setup consists of 1200 disks, 200 Intel CPU and 100 RAID controllers. The HGST OE setup consists of 2000 disks, 2000 ARM CPU and 34 switches. We didn't measure the power consumption, but we do not expect significant differences between the two scenarios.

5.3 Space utilization

Space utilization depends heavily on the exact composition of the building blocks. Where the HGST OE setup is rather dense, the currently deployed RAID systems are not very dense, systems with more disks per space can be found.

In the current setup, the RAID system would use 200 rack units (RU), whereas the HGST OE setup would add up to 136 RU. An advantage of the latter is that two computing rooms can be used for redundancy, each then holding 68 RU.

5.4 Management

On one hand, 100 systems with RAID controllers need to be managed. On the other hand, 2000 systems without RAID controller need to be managed. In either ways, a good and scaling life-cycle management system is mandatory. The higher number of nodes might be compensated by the nodes being less complex.

5.5 Operational aspects

We will focus on issues related to disk failures. We assume that the single disk failure probability is invariant in both scenarios.

If a disk fails in a RAID-6 system, it must be promptly replaced, and the RAID set needs to be rebuilt using information from the other 11 disks. A RAID-6 system can guarantee data integrity even with two failed disk – however, a third one failing e.g. during rebuild results in the loss of all files on the system. With increasing disk size, rebuilds tend to need longer, and the probability of such an event increases.

If a HGST OE drive fails, the files on it are gone. In our case, dCache must restore data integrity, and can do so because of each file, a second copy is held by another disk. As the files are distributed in a flat manner, many disks will have copies, so reestablishing redundancy can be achieved relatively fast without a huge impact on one single pool node. The manager of the dCache system must ensure that there is enough free space on all systems to create additional copies after a disaster.

As no timely action is required on the hardware level, an operational model can be envisaged where failed disks are not exchanged, but kept offline until the end of life of the whole enclosure they are part of.

6. Conclusion and outlook

Large storage systems build out of small building blocks seem feasible, and as our simulation indicates, depending on the exact usage pattern, can present a huge performance increase. HGST has presented the Open Ethernet Drive Architecture, and is preparing to introduce such drives to the market.

The TCO composition differs from the exact setup and the data safety features employed. We prospect that even in the case of a rather simple and inefficient second copy, the TCO composition of a setup consisting of the HGST Open Ethernet drives will not vary significantly compared to classical RAID server setups.

Our simulations are based on measurements made locally to the system, without considering network traffic to the data clients. In further works, we will perform server-to-clients measurements. We will also operate some HGST OE drives in our production dCache environment once final units with adequate integration into computing rooms are on the market. We will observe performance and operational aspects to refine our models and TCO estimations.

We will also investigate on using CEPH as one of the possible dCache storage backends, as soon as a CEPH driver for dCache is available.

References

[1] CEPH storage platform: http://ceph.com/

[2] dCache storage system: http://www.dcache.org/

[3] IOzone Filesystem Benchmark; http://www.iozone.org/

[4] HGST Open Ethernet; http://www.hgst.com/de/science-of-storage/emerging-technologies/open-ethernet-drive-architecture

[5] Ceph and the Open Ethernet Drive Architecture; Christopher Squires, HGST, presented at the CEPH Day New York, October 8 2014 (slides available at http://de.slideshare.net/Inktank_Ceph/05-ceph-day-new-york-hgst)

PoS(ISGC2015)029