Turkish Journal of Computer and Mathematics Education
DOI: https://doi.org/10.17762/turcomat.v9i3.13914 *Vol.09 No.03 (2018), 1220-1228*

Research Article

An Active Storage Framework for Utilisation of the Embedded Storage Array's Processing Capabilities

Manika Manwal

Asst. Professor, Department of CSE (Computer sc)

GEHU-Dehradun Campus

Abstract:

The adoption rate of digital products and services has skyrocketed. As the user base grew, so did the volume of data and the need for knowledge. The trend towards storing and processing this influx of data has been upward. Technology has been steadily trending forward, and with that, storage space and computing power. However, typical spinning discs incur a delay in transmitting data to processing components since they are slower than the computer capability. There is a growing Processing - I/O performance gap that contributes to this lag. When it comes to processing large amounts of data, the CPU-I/O performance difference widens. The Processing - I/O performance gap may be reduced with the help of suggested active storage architecture. This study presents a test environment for investigating the effect on application performance.

Keywords: Active Storage Framework, Parallel and Distributed Systems, Storage Arrays, Storage systems, Compute Storage system, Storage Virtualization, Active Disks, RPC

Introduction

The IT industry is no stranger to rethinking the foundation on which its operations are built. These revolutionary shifts in invention and innovation are occurring at a much accelerated rate. There has been a sea change in the IT environment from the first Platform to the third Platform. Gens and McAfee (2013). The original system was built solely around the mainframes and terminals of the "dawn of computing" period. The development of PCs, servers, DBMSs, and client-server architectures laid the groundwork for the Second Platform. The idea of a third platform evolved with the rise of cloud computing, mobile devices, big data analytics, and social media. Big data, mobile engineering, cloud computing, and social networking platforms are all part of the zeitgeist in IT right now. All of these descriptors are for widely used technology. Each of the three platforms has its own strengths and weaknesses when it comes to scaling apps and services and making the most of the technology they use. To what degree and by how many people are these applications and services being used is what ultimately determines the platform that supports them. Spindle technology speed complimented compute technology speed in the first platform period. Data retrieval speed was a defining feature of the spindles' era of technology.

Data access time was comparable to the processing power of the time. The length of time that passed after each section was evaluated. Access to storage and computation time were both measured in microseconds across both the compute and storage segments. Two dynamics, improvement in magnetic heads and improved head movement, propelled spindle technology forward as the disc development era advanced. This enhancement is mirrored in the smaller size of magnetic blobs, which allows for more information to be stored per track. The engineering of the memory and logic chips relied on the same technology that was utilised to create the spindle heads. Read/write magnetic head technology improved at a pace consistent with Moore's law. The current rate of improvement suggests a 35 percent gain annually.[Gens, 2014] Cost per spindle storage capacity has increased in tandem with processing power. However, the pace at which data might be read or written on a disc has only increased by 10-12% annually. Disc bandwidth was unaffected by the fact that the disk's outer edge was getting closer and closer to the speed of sound as spindle technology advanced.

Access times in the IT trended platform are measured in milliseconds, whereas processor cycles are recorded in nanoseconds, due to the rotational speed limitations of the magnetic spindles. Improvements to magnetic spindles during the second platform period were primarily made to aid in the progress of storage arrays. Technology has emerged in the storage array industry to help reduce the disparity between cycle and access times. This is the period that saw the introduction of dynamic random access memory (DRAM). Smart algorithms and read buffers improve DRAM's chances of delivering the necessary data quickly. DRAM write buffers are powered by batteries and capacitance, allowing for high data burst rates to be written with little delay. With RAID, data is striped across many spindles to boost transfer speeds. If applications are built in a certain manner, using minimal datasets and restricted functionality, then the aforementioned technologies may be useful.

Spindle storage constraints have had a harsh effect on application design, despite the fact that attempts to reduce the mechanical challenges have been remarkable. Previous studies have shown that apps are built with compact data sets and limited features so that they may fit into mobile devices' limited memory. Because to disc storage's significant unreliability, businesses could only do a limited number of database queries each transaction. The operational transactional system is isolated from the data warehouse and analytics. Wikibon and D. Fleury (2014). The reality is that modern programmes rely on very complex infrastructure software to help them deal with the widening gap between I/O and CPU cycles. The complex infrastructure software is brought all the way down to the storage controllers, via the many levels of the storage stack. Storage-related problems account for a disproportionate share of programme failures, and almost all design restrictions are data-related. The third setup is geared on increasing the storage domain's intelligence and speed so that it can keep up with the processing power. There are now four technologies that serve as the conceptual backbone of the third platform era. Cloud computing, large-scale data analysis, mobile devices, and social media all play a role.

As can be seen in Fig. 1, these four pillars have developed in tandem with the rising scalability of users and varied applications driven by the need to ensure business continuity. There is a massive explosion of data that is being shared, analysed, and utilised as the number of users increases with the variety of applications and services accessible. Increased data transmission rates between computation and storage nodes are a result of these technological advancements.

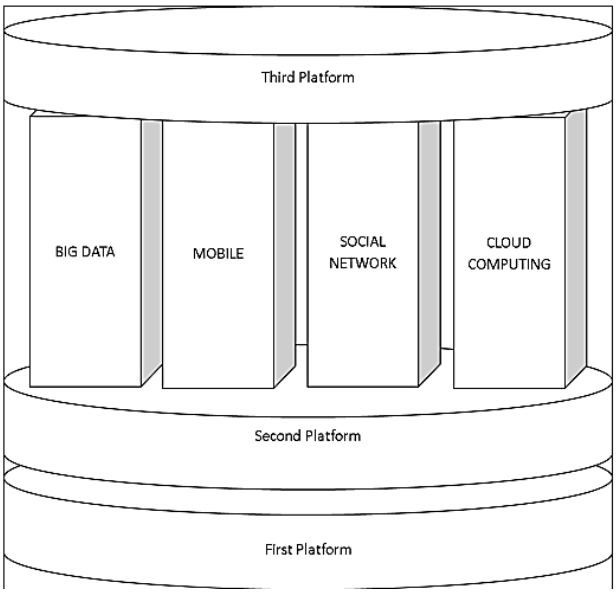

Figure 1: Platform Trend of IT Industry

Storage System and Platform Trend:

IT departments are experiencing dramatic changes to help them stay up with the rate of innovation and remain competitive. Applications that can operate on mobile devices and use the distant computation and storage have been created. As was previously indicated, the necessity to move to a third platform is enabled by the business model due to the number of users and the application deployment mechanism. The infrastructure and the data expand exponentially as the number of users increases. There is a flood of data from various devices that has to be analysed and processed in order to extract meaningful insights. The workflow of storage systems remains same from the first to second platform. These storage systems are often designed as large, centralised units. The present trend in the development of storage systems is a major roadblock to the continuing digital revolution. The reason for this is the complexity in datastorage management and the fact that storage systems are already struggling to keep up with the ever-increasing storage demand brought on by the proliferation of data. Data storage, connectivity, and processing power are all becoming more important as data volumes continue to explode. Data may be accessed or stored from any place in the globe due to the features of the user's location and the business model. Dispersed storage systems that need more work to manage and organise the dispersed data. Different sorts of applications exist on each of these four technologies, which is why the third platform relies so heavily on them: big data, the Cloud, mobile, and social networks. Applications built on different technologies offer very distinct demands and interact with storage systems in very unique ways. The environment is dynamic and diverse since it is the result of a combination of workloads and applications. The storage system you choose should be able to respond quickly to requests from your applications. Both data services for data-intensive applications—where speed is paramount—and storage services like replication, snapshot, failover, etc., should be performed by the storage systems. Robustly

Literature Review

Wang et al. 2009 network bandwidth capacity has been noted as a limiting factor in distributed environments because to the high amount of data and frequent connections between different nodes..

Ahrens et al. 2011 the existing design of CPUs is dependent on temporal and geographical proximity for efficient memory hierarchy utilisation, it has been reported. However, many I/O-heavy programmes suffer from subpar location. By designing and creating high performance parallel algorithms, these I/O-intensive applications may run in a distributed setting. Researchers have shown that increasing the number of cores in a CPU makes it easier to use parallel algorithms in that setting.

Runde et al. 2012; Chandy 2010; Pop et al. 2013 Developed a strategy for regulating OS operations in order to make the most efficient use of limited resources. Scheduling the scientific procedure in a way that worked with the model was successful. A mechanism called DualPar was mentioned in the suggested model. The parallel programme model-implemented application might run in two distinct modes. Data-driven and generic computation execution modes are used. Normal execution occurs in the latter mode, while I/O service scheduling is performed between process executions to optimise overall execution for maximum I/O efficiency in the former mode. The murder is committed close to the disc. The authors have examined the testbed and latency in order to assess the model.

Kim et al. 2012 discovered and proposed that the Storage system, not the CPU or Memory, is to blame for the slowdown of programmes in a high-performance computing setting. The effects of I/O traffic on data storage systems are a primary area of study. Finding and correcting a performance problem in a parallel application execution environment requires a knowledge of the I/O behaviour in the I/O stack, the author argues. Gathering and analysing the massive number of logs created from tens or hundreds of machine on which the parallel programme runs is a significant challenge to understanding the I/O behaviour.

Research Methodology:

Fig.2 depicts the research strategy that was implemented. The process begins with an analysis of the active storage spectrum, or the many active storage models currently in use, and proceeds to the extraction of the storage components' performance characteristics and their cumulative effect on the application's overall performance. With the knowledge gained from this research, storage system administrators should be better equipped to determine whether or not offloading the computational part from the program's host node viz. server to the storage devices will result in performance improvement. Finally, two test environments with it hardware are developed to validate the statistical model and verify the proposed active storage paradigm. Some setups use the tried-and-true method of execution, such as copying files from storage to the server, while others use more modern methods, such as active storage. Both the Mobility RPC and the Hadoop frameworks are used in the analysis. Additionally, simulation is included in the test bed evaluation process.

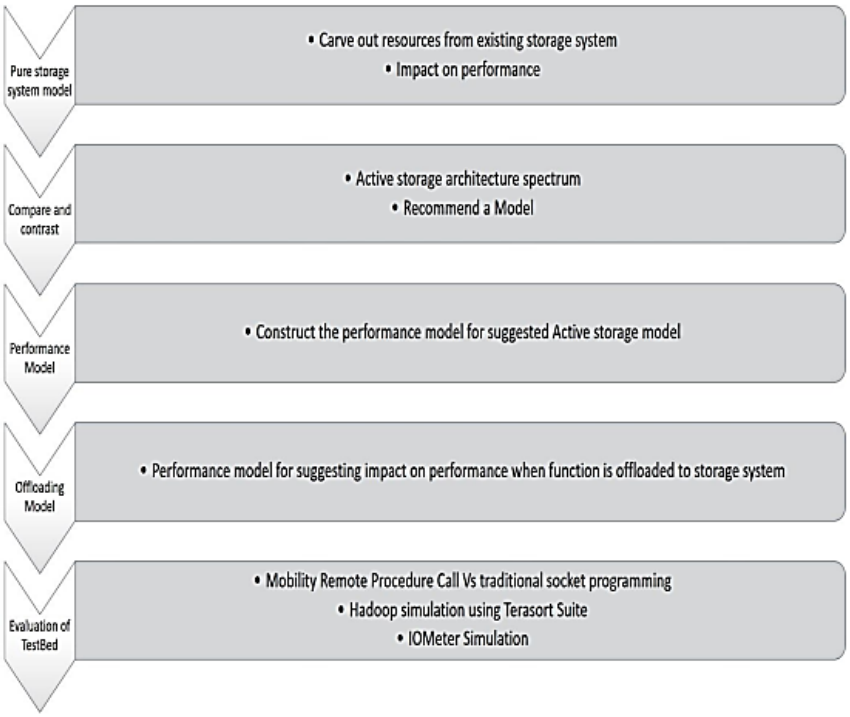

Figure 2: Research Methodology

Availability of Proposed Active Storage (Paras) Node in Cluster

When one or more of an array's parts fails to function as expected or entirely ceases working, the whole system or at least the affected device(s) may become unavailable as a result of a chain reaction. Mean time to failure (MTTF) and mean time between failures (MTBF) are often used to convey the dependability of a system or product. Both of these parameters are tested against the time distribution that is already known. Due to the use of virtualization in the proposed active storage concept, many guest operating systems may use the same hardware. Time-sharing allows several Virtual machines access to the same physical hardware, known as the bare metal. Virtualization's benefits include a lower total cost of ownership, less time spent on management and setup, more adaptability, and a higher tolerance for system complexity. The apps themselves are unaware of the fact that a virtualization layer has been developed that supplies all resources to them in the form of a virtual entity. It gives the impression to the programme that it is being executed on a single computer with all of its hardware resources at its disposal. Since the bare metals are not visible to the application, data loss or service interruptions may occur if a component fails. Due to the fact that virtualization technology performs reliability

checks on a bundled set of hardware and software, it is important to evaluate the dependability of each individual part. The longer the system is down, the more money is lost. The time it takes for the system to be down is distinct from the time it takes for the support staff to begin maintenance, for the client to manually reboot the system, etc. When a system fails and users are left waiting for promised services, such period is known as downtime. This chapter's discussion of the storage array's dependability and availability is grounded on the array's susceptibility to hardware faults. In the case of virtualization, problems with virtual computing and virtual discs may be fixed using existing techniques. As a consequence, the failures have a greater effect, resulting in the loss of functionality of individual parts or the whole storage array. The hypervisor or virtual machine monitor is the hardware-level software responsible for sharing the underlying physical hardware across many guest operating systems and creating a shared pool of virtual resources.

Reliability of Software & Hardware:

Through trials and the representation of data in a graph resembling a bathtub, researchers have analysed and verified the dependability of the system and hardware. Whether the rate of failure is growing, staying the same, or decreasing, the graph depicts all three of these scenarios. See the graph below for a visual depiction of the bathtub curve. The graphs depict the percentage of failure over time. The failure rate is presented speculatively. There are two ways to interpret the curve's first stage. The first stage, known as the newborn mortality period, occurs immediately after component deployment. Second, while the component is still in use, the rate of failure goes down with time. While manufacturing tools like six sigma and quality improvement techniques, as well as advances in technology, have helped reduce the failure rate, failures still occur due to a lack of fine-grained control over the process. The operational lifetime is short due to burn in and temperature cycling. The rate of failure is shown to be consistent over time in the second phase. This is the actual running time of the system or component. The time span during which the system functions without significant breakdowns is known as its operational life period. Predicting the system's uptime and availability is possible at this point. The third stage is the depreciation or exhaustion of the hardware. Due to the ongoing electrical and thermal stress placed on the system's electrical components, failure rates increase during this phase. This time frame is appropriate for determining the median time before failure.

Testbed Setup, Evaluation and Result Analysis

The IOMeter's GUI and Dynamo were separated and hosted on separate nodes so that the second testbed could be built. The Dynamo is configured to work with Testbed 3b and the virtual machine that was previously defined in the system. The graphical user interface is deployed on a different node, and the storage node is used to setup and move the burden to the dynamo. The GUI sends orders to the dynamo, which then runs the commands against the test file. The former configuration uses a dynamo as its central processing unit (CPU), which is responsible for creating test workloads and reading and writing to the test file. Dynamo transmits I/O operations to the discs using the Compute node's CPU cycles and utilises the storage node's processing cycles for I/O activities. By relocating the dynamo to the storage node, the storage processor cycles may be used for both workload generation and I/O activities, simulating a real-world active storage deployment as closely as possible. The IOMeter is shown in use in an active storage scenario in Fig. 3.

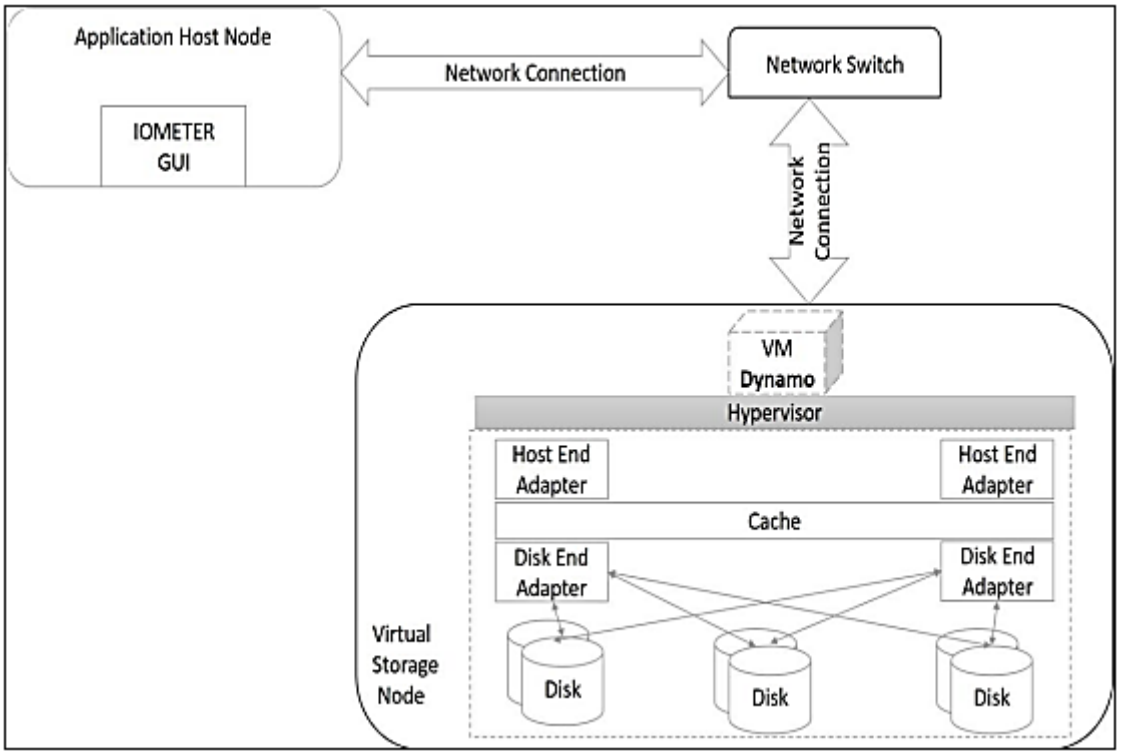

Fig. 3: Testbed 3b: IOMeter deployed on architecture model 2 Testbed

Workload Profiles:

Table 1: Workload profile applied using IOMeter

APPLICATION CLOSE RESEMBLANCE	TYPE OF ACCESS	% OF ACCESS		% & TYPE OF OPERATION		REQUEST SIZE (KB)
		Random	Sequential	Write	Read	
OLTP Database	Random	100%	0	30%	70%	8 KB
Microsoft Exchange	Random	100%	0	35%	65%	4 KB
File Server	Random	75%	25%	10%	90%	8 KB
Video on Demand	Random	100%	0	0%	100%	512 KB
Media Stream	Sequential	0%	100%	5%	95%	64 KB

Since just the VM's performance with its own virtualized resources is represented in Testbed 3b, the CPU utilisation is greater since the virtual component and overheads take more processing cycles than is shown here.

This virtual machine (VM) will host the IOMeter's Dynamo component, and the VM's virtual disc will be linked to the VM that will be subjected to the burden created by Dynamo. Here, IOMeter utilises the fabric that links the disc to the die in order to do all data transfers locally. In order to enhance utilisation, it is necessary for the processors to spend its cycles on the dynamo at the same time as they pay for I/O receives and acknowledgements. Due to the fact that the same processor performs read and write operations while also providing support to create the workloads through dynamo, this architecture mimics the active storage idea. In this case, dynamo is treated as an application, and the disc read/writes performed locally are I/O activities. Therefore, an active storage pattern emerges in which the application and disc access get closer together. When looking at the entire aggregated CPU utilisation of storage node, the average is close to 30%, whereas the utilisation in Testbed 3b system testbed measures at just 24.84%. The virtualization method used to bring computation closer to data causes this 6% gap between IOMeter-measured CPU utilisation efficiency and the aggregation CPU utilisation%.

Table 2: Processor Utilization in Testbed 3a & 3b

WORKLOAD PROFILE	TESTBED 3A CPU UTILIZATION (%)	TESTBED 3B CPU UTILIZATION (%)
OLTPDB	13.507637	15.278398
MS Exchange	6.626504	9.481507
Fileserver	2.534552	14.166516
Video on Demand	10.681488	36.250785
Media Stream	11.050595	34.036541

Fig. 4 is a bar chart depicting two series, series 1 being testbed 3a and series 2 being testbed 3b.,

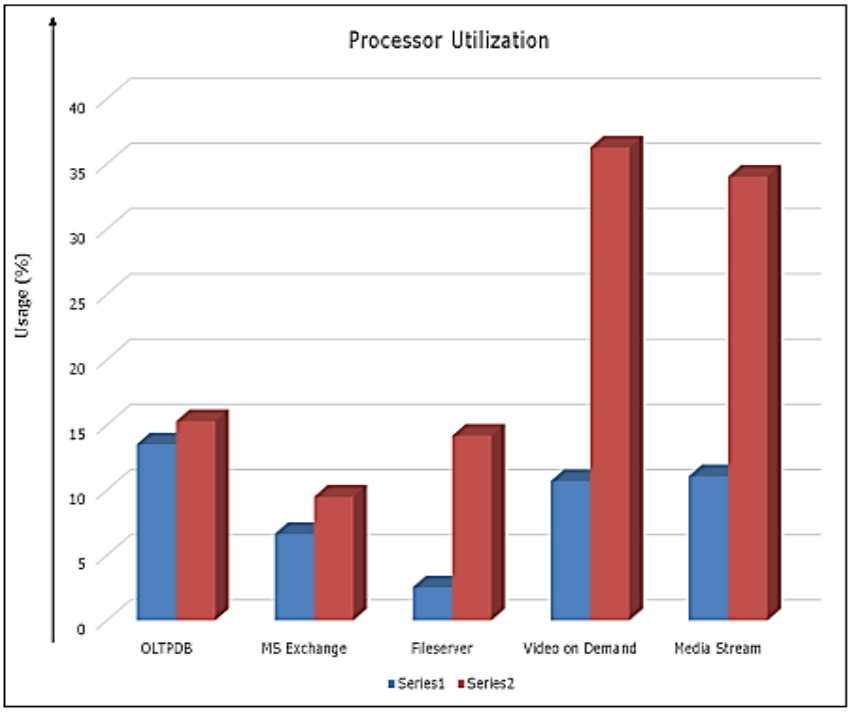

Figure 4: Graph to compare the storage processor utilization of both testbeds 3a & 3b

Conclusion

Input/output operations per second (IOPS), bandwidth, and latency are the primary measures used to assess the performance of storage systems. The disc sub system performance model has been described in previous works of study, or it has been applied. Different writers explain the performance of active storage systems by modelling both the application and the storage system. None of the offered models shed light on the question of how much more storage space per unit of application consumption is required to achieve the same level of performance gains. Both the MobilityRPC and Hadoop Map reduction frameworks were used in the assessment of the suggested architecture paradigm. These frameworks were selected mostly because to their open source nature and their adaptability. However, much of the focus of earlier studies has been on database applications rather than those that handle unstructured data. Both the conventional approach and the active storage model were evaluated using two test beds.

References

1. Acharya, A., Uysal, M. & Saltz, J., 1998. Active disks: programming model, algorithms and evaluation. SIGPLAN Not., 33, pp.81–91. Available at: http://doi.acm.org/10.1145/291006.291026.
2. Adomavicius, G. & Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), pp.734–749.
3. Ahrens, J. et al., 2011. Data-intensive science in the US DOE: Case studies and future challenges. Computing in Science and Engineering, 13(6), pp.14– 23.
4. Ali, M.F., Barnawi, A.M. & Bashar, A., 2012. Performance analysis framework to optimize storage infrastructure for Cloud Computing. In 2nd International Conference on Innovative Computing Technology, INTECH 2012. pp. 285–290.
5. Ananthanarayanan, R. et al., 2009. Cloud analytics: Do we really need to reinvent the storage stack? Proceedings of the 1st USENIX Workshop on Hot Topics in CLoud Computing, pp.1–5.
6. Anastasiadis, S. V., Wickremesinghe, R.G. & Chase, J.S., 2005. Lerna: An active storage framework for flexible data access and management. Proceedings of the IEEE International Symposium on High Performance Distributed Computing, pp.176–187.
7. Băsescu, C. et al., 2011. Managing data access on clouds: A generic framework for enforcing security policies. In Proceedings - International Conference on Advanced Information Networking and Applications, AINA. pp. 459–466.
8. Bechini, A. & Vetrano, A., 2013. Management and storage of in situ oceanographic data: An ECM-based approach. Information Systems, 38(3), pp.351–368. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0306437912001305 [Accessed January 23, 2015].
9. Boboila, S. et al., 2012. Active flash: Out-of-core data analytics on flash storage. In IEEE Symposium on Mass Storage Systems and Technologies.
10. Carlsson, G. et al., 2012. Computational topology for configuration spaces of hard disks. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics, 85(1).
11. Caulfield, A.M. et al., 2010. Moneta: A high-performance storage array architecture for next-generation, non-volatile memories. Proceedings of the Annual International Symposium on Microarchitecture, MICRO, pp.385–395. Available at: http://dl.acm.org/citation.cfm?id=1934984.
12. Chen, C. & Chen, Y., 2012. Dynamic active storage for high performance I/O. In Proceedings of the International Conference on Parallel Processing. pp. 379–388.
13. Chen, Z. et al., 2005. Empirical evaluation of multi-level buffer cache collaboration for storage systems. ACM SIGMETRICS Performance Evaluation Review, 33(1), p.145. Available at: http://portal.acm.org/citation.cfm?doid=1071690.1064230

14. Crespo, A., Ripoll, I. & Masmano, M., 2010. Partitioned embedded architecture based on hypervisor: The XtratuM approach. EDCC-8 - Proceedings of the 8th European Dependable Computing Conference, (April), pp.28–30.
15. Donnelly, P. & Thain, D., 2013. Design of an active storage cluster file system for DAG workflows. Proceedings of the 2013 International Workshop on Data-Intensive Scalable Computing Systems, pp.37–42
16. Elerath, J.G. & Pecht, M., 2007. Enhanced reliability modeling of RAID storage systems. Proceedings of the International Conference on Dependable Systems and Networks, pp.175–184.
17. Ge, Z., Lim, H.B. & Wong, W.F., 2005. Memory Hierarchy HardwareSoftware Co-design in Embedded Systems. Journal of Computer Science, 1, pp.1–9.
18. Gulati, A., Kumar, C. & Ahmad, I., 2010. BASIL : Automated IO Load Balancing Across Storage Devices. 8th USENIX Conference on File and Storage Technologies (FAST '10), pp.13–26
19. Jannen, W., Tsai, C. & Porter, D.E., 2013. Virtualize Storage , Not Disks The Cost of Virtual Disks. In HotOS'13: Proceedings of the 14th USENIX conference on Hot Topics in Operating. USENIX Association'
20. Kumar, R., Zyuban, V. & Tullsen, D.M., 2005. Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling. Proceedings - International Symposium on Computer Architecture, 00(C), pp.408–419.
21. Lammie, M., Brenner, P. & Thain, D., 2009. Scheduling grid workloads on multicore clusters to minimize energy and maximize performance. Proceedings - IEEE/ACM International Workshop on Grid Computing, pp.145–152
22. Li, S. & Huang, H.H., 2010. Black-Box Performance Modeling for Solid-State Drives. 2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp.391–393