A Framework for Measuring the Performance and Power Consumption of Storage Components under Typical Workload

DongJin Lee, Michael O'Sullivan, Cameron Walker
{dongjin.lee, michael.osullivan, cameron.walker}@auckland.ac.nz
Department of Engineering Science
The University of Auckland, New Zealand

***Abstract*—Although the cost of storage components are reported accurately by the vendors, it is not clear whether the performance (IOps, MiBps) and power consumption (W) specifications they provide are accurate under 'typical' workloads. Accurately measuring this information is a vital step in providing input for optimal storage systems design. This paper measures storage disk performance and power consumption using 'typical' workloads. The workloads are generated using an open source version of the (industry standard) SPC-1 benchmark. This benchmark creates a realistic synthetic workload that aggregates multiple users utilizing data storage simultaneously. A flexible current sensor board has also been developed to measure various storage devices simultaneously.**

This work represents a significant contribution to data storage benchmarking resources (both performance and power consumption) as we have embedded the open source SPC-1 benchmark `spc1` within an open source workload generator `fio`, in addition to our flexible current sensor development. The integration provides an easily available benchmark for researchers developing new storage technologies. This benchmark should give a reasonable estimation of performance with the official SPC-1 benchmark for systems that do not yet fulfill all the requirements for an official SPC-1 benchmark. With accurate information, our framework shows promise in alleviating much of the complexity in future storage systems design.

***Index Terms*—Performance, Power Consumption, Benchmark, Measurement, Storage Component, HDD, SSD, Open Source SPC-1**

I. Introduction

With an increasing number of networked users and growing volume of complex data, information is becoming more valuable and access to that information is critical in many organizations. This necessitates modern storage systems with large capacity that are fast and reliable capacity. High performing storage systems, for example, deliver a high number of operations (IOps), high throughput (MiBps) and low response times (ms). The current generation of such systems mainly consist of storage components made up of a large number of HDDs and SSDs. These disks have greatly improved over time, both in density (higher volume) and cost (lower price).

Storage systems such as Storage Area Networks (SANs), are becoming more versatile as they need to be able to accommodate many different types of workload on the same system. Instead of using a different server with customized disk configurations (e.g., one set-up for email, another for a database, one for web, one for video streaming and so on), organizations are centralizing their data storage and using the same storage system to simultaneously support all services. Critical design factors for storage system development include: 1) total system cost; 2) system performance; and 3) power usage. Therefore, mathematical models [36], [37] developed to design storage systems rely on accurate information about cost, performance and power consumption of storage components. In addition, as more storage components and different device types emerge, technologies have also been focused towards components that are more 'green-friendly' so as to reduce the amount of power consumption. For example, it is estimated that globally installed disk storage has increased by 57% between 2006 and 2011, with a 19% increase of power consumption [32].

Storage architects are often confronted with a multitude of choices when designing storage systems. When selecting the best option, architects take into account a large number of metrics such as performance, quantity, cost, power consumption, etc. Often the best systems will use various types of disks to support differing workloads so as to increase total performance, yet decrease costs. In addition, finding disk configurations and data management policies that yield low total power consumption is of rising importance in storage systems. Mathematical models have automatically designed SANs [39] and core-edge SANs [37] for various end-user demands, but, in many instances, the demands are not known before the storage system is designed. In these cases, typical workload is the best benchmark, hence the properties of the components under typical workload need to be accurate. Our framework should become established as a barometer for vendor-supplied values, such as performance and power consumption, under *typical* workload.

For practical evaluation of storage devices, various open source tools have been utilized by the technical and research community. Data storage benchmarks have been developed with many parameters to build specific workloads for examining; a file system [8], a web server [6], a network [9] and so on. However, benchmarks for cutting edge storage systems need to create a workload that combines the workload for the file system, web server, network and other services.

Producing synthetic benchmarks with a realistic workload to capture overall performance is a non-trivial task because it is difficult to model systematic user behavior. Typically, communities have chosen their own set of methods to represent a typical workload. Results from these methods are, however, subject to the parameters and configurations particular to each community. To date, especially for storage systems, few tools have attempted to capture overall performance of storage systems.

Another issue is the variability of power consumption reported by individual disks depending on the workload benchmarked. Realistic readings could be different to the vendor's specified reports, which can be lower or higher depending on the application workloads [5]. Also, power consumption measurement is non-trivial because the disk does not have an internal power sensor for reporting. A current-meter is needed to independently read the currents flowing through each rail (such as 12V, 5V and 3.3V) so as to compute the total power (W). These measurements need to be recorded and computed simultaneously for multiple storage disks, so require a flexible sensor system that is scalable to a large number of disks of different types. To date, little research has attempted to develop set-ups to measure disk power consumption for a set of disks simultaneously.

Broadly our research advances storage systems technology in two important ways. First, it provides a framework for benchmarking performance and power consumption on various storage components such as HDDs, SSDs, RAID arrays and entire storage systems. The performance is measured by using the industry standard SPC-1 benchmark (open source 'typical' workload generator) and the power consumption is measured by using a Current Sensor Board developed

DOI: 10.5176_2010-2283_1.2.55

during this research. Second, our work then uses the proposed framework for the validation of vendor specifications of storage components. We demonstrate our framework using different configurations of commodity storage devices. The properties of these configurations are vital inputs to mathematical models that promise to alleviate much of the design complexity currently faced by storage architects. Ultimately we would like to develop a mathematical model that, given vendor information as an input, produces an accurate estimate of performance and power. This model will provide valuable input into our existing mathematical models and corresponding optimization algorithms for storage system design.

In Section II, we introduce the workload generator developed by combining the open source SPC-1 library `spc1` [19] and an open source workload generator `fio` [3]. We describe how this integrated tool produces typical workloads for benchmarking storage components. Despite the wide range of benchmarking tools available, no practical studies have been done using the open source SPC-1 library. We also describe our Current Sensor Board (CSB), a current-sensor kit that measures the current flowing through individual rails of multiple disks, in order to obtain their power usage, both passively and simultaneously. In Section III, we present experiment results. Sections IV and V discuss related work and summarize our findings.

II. Measurement

Widely used disk benchmarking tools such as `IOmeter` [7] measure performance by generating as much IO as possible to disk, thus obtaining upper bounds on IO operations (IOps), throughput (MiBps) and response times (ms). Often four types of conventional workload – sequential read, sequential write, random read and random write – are tested, and to mimic 'typical' (user) workloads, the proportions of sequential/random, read/write requests are adjusted and the block size of requests are similarly tuned. While such tools are designed for flexibility, they do not generate typical IO workloads to practically represent the workload experienced by a centralized storage system.

The SPC-1 benchmark tool (developed by the Storage Performance Council – SPC) [13] has the ability to produce typical workloads for evaluating different storage systems and is the first industry standard tool with supporting members from large commercial vendors of storage systems (e.g., HP, IBM, NetApp, Oracle). SPC-1 particularly focuses on mixed behaviors of simultaneous, and multiple user workloads which were empirically obtained from various real user workloads for OLTP, databases, mail servers and so on. It performs differently from other tools in that it has built-in proportions of random/sequential read/write requests that aggregate to give an overall request level and it uses a threshold (of 30ms) for an acceptable response time.

Specifically, SPC-1 generates a population of Business Scaling Units (BSUs) which each generate 50 IOs per second (IOps), all in (minimum) 4KiB blocks. Each BSU workload produces the IO requests to read/write to one of three individual Application Storage Units (ASUs) – ASU-1 [Data Store], ASU-2 [User Store], and ASU-3 [Log]. Both ASU-1 and ASU-2 receive varying random/sequential read/write requests, whereas ASU-3 receives sequential write requests. The proportion of IO requests to ASU-1, ASU-2, and ASU-3 are 59.6%, 12.3%, and 28.1% respectively. For example, a population consisting of two BSUs (generating 100 IOps) will produce about 59.6 IOps to ASU-1, 12.3 IOps to ASU-2, and 28.1 IOps to ASU-3. Also, physical storage capacity is allocated at 45%, 45%, and 10% for ASU-1, ASU-2, and ASU-3 respectively. Higher IO workloads are generated by increasing the number of BSUs, and the benchmark measures the average response time (ms) to evaluate the performance. Its detailed specifications are documented in [14].

The official SPC-1 benchmark is, however, aimed at use by commercial storage vendors, particularly the industrial members of the SPC, with associated fees. It is unavailable for experimental storage systems that are not commercially available. It is also not available for testing individual storage components themselves, thereby excluding most of the research community. Further, use of SPC-1 for research publications is subject to permission being granted by the SPC. To date, only a few publications from the industry members of the SPC have used the official tool to validate their research [17], [25], [29], [42].

The open source SPC-1 library (`spc1`) has been developed by Daniel and Faith [19]. This library reliably emulates and closely conforms to the official SPC-1 specification. The library produces SPC-1-like IO requests with timestamps, but requires a workload generator to send these requests to a storage system. For example, Fig. 1 shows scatter plots of IO request streams to the ASUs, for 20 BSUs accessing the ASU on a 1TB disk (262144 in 4KiB block). ASU-3 for instance shows IO requests for sequential write – increasing the BSU counts further produces more variable IO write behavior. The behavior matches the specifications described in the Tables 3-1, 3-2 and 3-3 in [14]. Here, the 4KiB block length distributions are drawn from the set {1, 2, 4, 8, 16} with converging proportions of the read IO [33.1%, 2.5%, 2.1%, 0.8%, 0.8%] (39.4%) and the write IO [43.8%, 6.8%, 5.6%, 2.2%, 2.2%] (60.6%) respectively. For any reasonable workload duration of IO requests, we observe an average IO block size of 6.75KiB read and 8.82KiB write, which results in an average of 8KiB per IO request; twice of the block size. This means that we can indeed multiply the total IO requests by 8KiB to calculate the total throughput requests.

Unfortunately, the actual workload generator used by Daniel and Faith is proprietary [18] and so no readily available tools exist that integrate the library in practice. Without integrating the library into a workload generator, only simulations of the library's workload can be conducted. Excluding the original study [19], the open source SPC-1 library has only been used for simulation [20], [28].

A. The `fiospc1` tool

An open source IO tool used for benchmarking hardware systems is `fio` [3]. It is highly flexible with a wide range of options supported, e.g., different IO engines and configurations of read/write mixes. It works at both a block level and a file level and provides detailed statistics of various IO performance measures. Here, we have produced `fiospc1` – an integration of `fio` and the `spc1` – which, when executed, generates IO requests in accordance with the SPC-1 benchmark via `spc1` and sends these requests to the storage system defined within `fio`. The integrated version is now available in [4] and a simple one line command

```
fio --spc1 spc1.config
```

executes the `fiospc1`. The `spc1.config` file contains simple arguments such as disk device locations, number of BSUs, test duration and IO engine selections.

B. Current Sensor Board

Storage devices come in different physical sizes (1.8″, 2.5″ and 3.5″) and voltage rail requirements (3.3V, 5V and 12V). For example, often 3.5″ disks require 5V and 12V to be operational, 2.5″ disks often require 5V, and 1.8″ disks require 3.3V. Depending on the disk type, it may also require all of the rails to be operational. To obtain the power consumption of a disk requires a way to measure the current (A) of individual rails simultaneously without interrupting the benchmark activities.

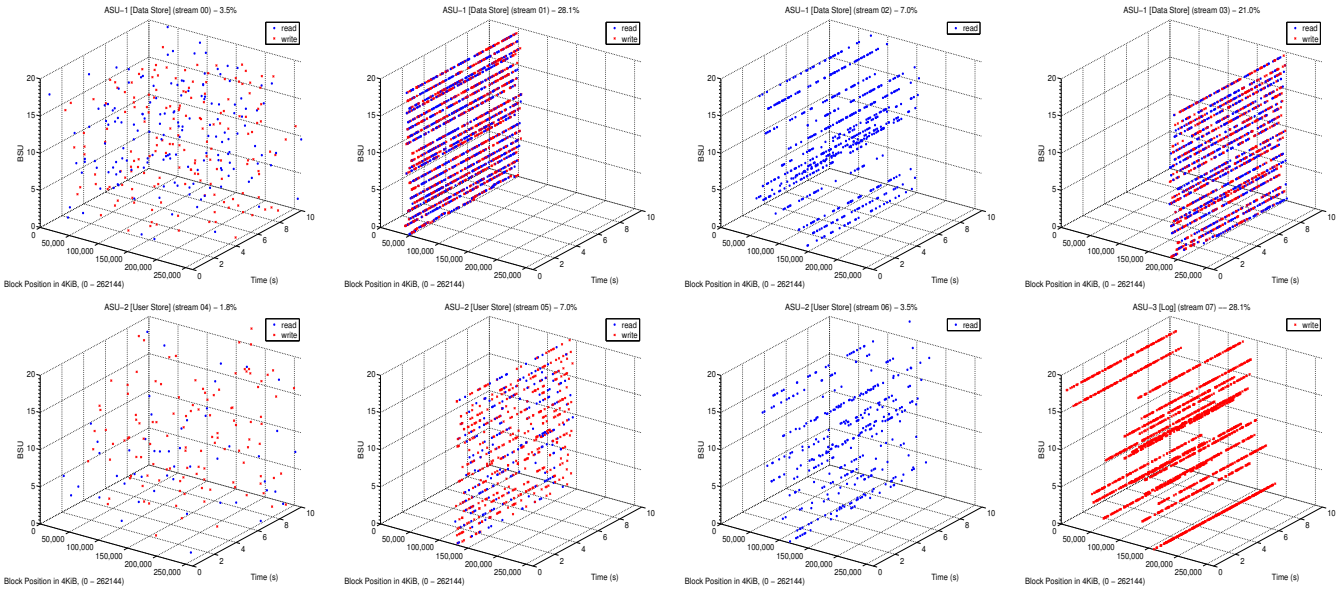

Fig. 1. IO workload distributions by the ASU streams: ASU-1 [Data Store] (stream 00–03), ASU-2 [User Store] (stream 04–06), ASU-3 [Log] (stream 07)

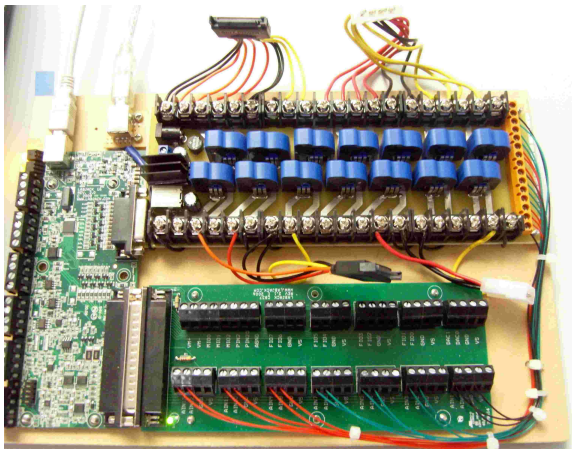

Fig. 2. Current Sensor Board (CSB)

We have built our own Current Sensor Board (CSB), shown in Fig. 2. The CSB allows us to passively examine power consumption of multiple disks in a scalable way. It can measure up to 14 current rails simultaneously and log the results using a Data Acquisition (DAQ) device. Each rail is measured (magnetically) by the Hall-Effect current transducer and the analogue to digital conversion (ADC) is performed with 16 bit resolution. Using these transducers has the advantage over current-shunt methods of adding no resistance to the rails. The transducers are rated 6A which means each rail can measure up to 19.8W, 30W and 72W for 3.3V, 5V and 12V respectively. A single disk typically uses less than 1A each rail, so each transducer scales well and can also measure multiple disks aggregated together. Both the transducer and DAQ specifications are detailed in [2] and [10] respectively.

C. System Components and Storage Devices

The test system has Intel 2.53Ghz CPU processors (2xE5630, 8 cores), Intel 5520 chipset (18GB DDR3-1333, ICH10R) and a separate HW RAID controller (LSI SAS 2108 BBU with 512MB DDR2 [1]). We use two commodity disk types (regarded generally as high-end); HDDs[1] and SSDs[2] (six disks each) which are powered individually through the CSB from a separate power supply. Both disks are configured to HW-RAID and SW-RAID, for x1, x3 and x6 RAID0 mode. For HW RAID, we followed the recommended configuration from the vendor [11] – [Read-Ahead, Write-Back] is set for HDDs and [Read-Normal, Write-Through] is set for SSDs. For SW RAID, we use the `mdadm` tool. All RAID stripe sizes are 8KiB, direct IO with disk cache enabled for optimal performance for transaction/random IO. The disks are then partitioned into ASU-1, ASU-2 and ASU-3 capacity to receive the IO workloads. 64-bit Linux OS (Ubuntu 10.04 LTS, April-2010) is installed.

Benchmarks are run with the asynchronous (`libaio`) IO engine following the original study by Daniel and Faith [19] closely. Since we are not benchmarking a complete storage system in this work, we started from the smallest IO requests – BSU=1 (50 IO requests per second) and increased consistently until the disk bottlenecked. Also, based on Daniel and Faith's scaling method [19], the IO depth is incremented every 25 IO requests, i.e., (BSU=1, depth=2), (BSU=2, depth=4), and so on. We then observed the behavior of the performance and power consumption.

Generally, HDDs perform better for sequential read/write requests than the SSDs, especially when the disk's edge sections are utilized (e.g., short-stroking). However, HDDs perform worse for random read/write requests due to a mechanical movements within the disk. For the SPC-1 IO workload patterns, consisting of both random and sequential requests, HDDs would be expected to score in between the conventional random and sequential IO workload. We have benchmarked individual ASU workloads on each disk, JBOD configurations and also low-end HDDs and SSDs. The results for the low-end disks are reported in [26]. The work here focuses on combined ASUs under the SPC-1 workload in a RAID0 (HW and SW) environment.

The work here could be expanded by testing a variety of storage devices with different characteristics available in the market. However, the focus of this work is the development of a consistent and easily repeatable testing framework. Benchmarking with other types of disks and different storage configurations (such as distributed file systems using Ceph [40] and XtreemFS [22]) is ongoing research

[1]WD Caviar Black 64MB, 2TB – `WD2001FASS` [16]
[2]OCZ Vertex 2, 60GB – `OCZSSD2-2VTXE60G` [12]

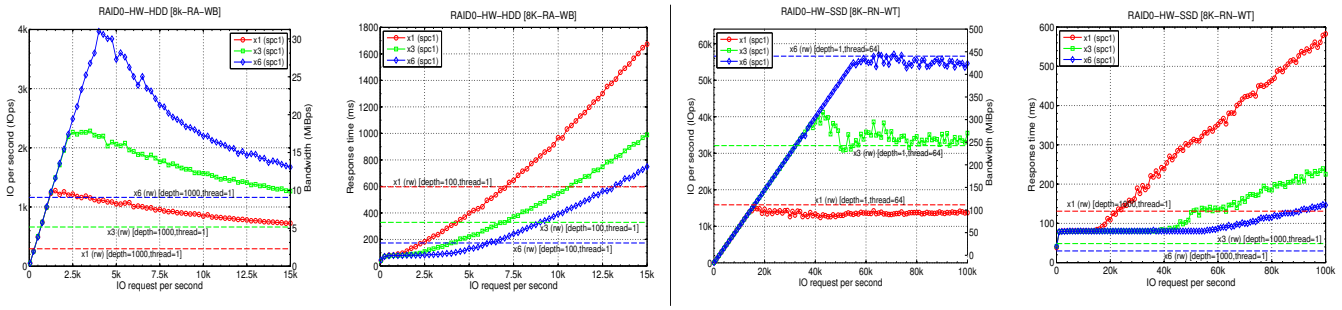

Fig. 3. Measurement of HW RAID0 for x1, x3 and x6 disk configuration – Left: HDD, Right: SSD

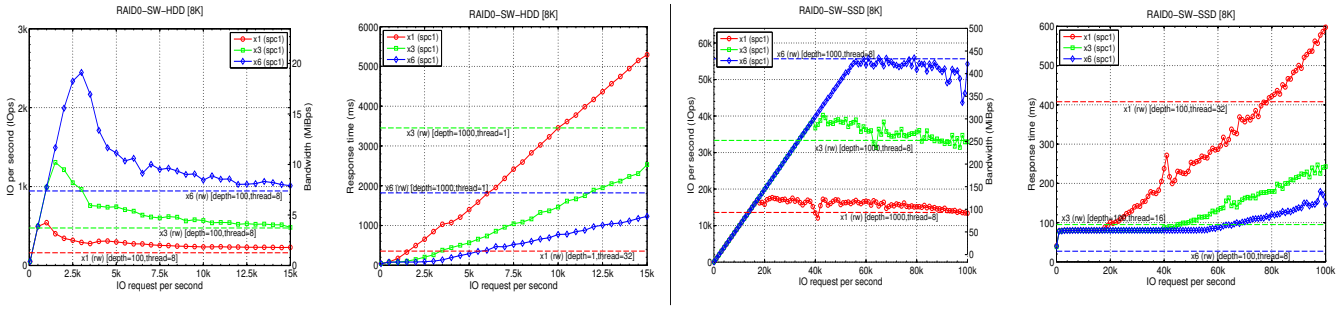

Fig. 4. Measurement of SW RAID0 for x1, x3 and x6 disk configuration – Left: HDD, Right: SSD

that will utilize the framework described here.

III. Experiments and Observations

A. HW RAID and SW RAID

Fig. 3 shows the performance measurements from HW RAID configurations: the first two plots show the IOps and MiBps and response time measurements respectively for the HDDs, and the last two plots show the same for the SSDs. For HDDs, the IOps initially increase with increasing BSU count (IO request per second) and after peaking, decrease in an exponential manner. Increasing IOps for lower BSU counts shows that the disk spends time waiting between requests (resulting in low IOps) and as more requests become available (as the BSU count increases) the disk "keeps up" and the IOps increase. The rate of increase of IOps increments is observed to be the same for all disk configurations because SPC-1 specifically does not push more than 50 IO requests per second for a single BSU. Striping more disks resulted a higher IOps peak, e.g., x1: BSU=30, x3: BSU=50, and x6: BSU=80. The highest throughput measured is just over 30MiBps for x6.

Once the BSU count increases past a given number, the IOps begin to decrease. This shows that the disk is unable to keep pace with the number of IO requests and so the requests are queued, resulting in less IOps and longer response times. For example, requesting 30s of 100 BSUs took 130s, 70s and 37s for x1, x3 and x6 configurations respectively. For 200 BSUs, it took much longer: 330s, 185s and 130s respectively. As mentioned previously, the BSU increase results in requests to the ASUs that are more diverse (including ASU-3 which experiences sequential write requests at low BSU counts), ultimately generating random IO behaviors. The plots also show some of the conventional IO workload patterns (50% read, 50% write, 8KiB random IO) measured with varying IO depths and threads (shown as horizontal lines). We observe that the disks handle the increasingly random SPC-1 workload better than other 'full-random' IO patterns.

In general, if response times from a storage device remain low with increasing IO requests, then the storage device (in our case, the disk) is regarded as capable of handling multiple tasks simultaneously. For the HDDs, at a lower BSU range (<10), the response time stayed constant at about 100ms, and as the BSU count increases up to the peak IOps, the response time increases accordingly. The response times are greatly reduced when using a multiple disk configuration, e.g., x1 to x3, but only slightly reduced when using more disks, e.g., from x3 to x6. We also find that average IO lengths queued by the disks (measured using `iostat`) correlate with the response time measurements.

SSD configuration response times also reach their own peak IOps, with much higher IOps than the HDDs, e.g., x1: BSU=15k, x3: BSU=43k, and x6: BSU=59k. Conversely, SSD response times do not decrease exponentially but plateau for higher BSU counts. This behavior also shows that the SSDs can have sustainable throughput even when the IO patterns received are more random. Furthermore, SSDs report consistently low response times, e.g., 80ms until IOps peaks, and slowly increase linearly (x3 and x6). Thus, SSDs overall outperforms HDDs in many aspects.

Fig. 4 plots measurement results from the SW RAID configuration. Generally the plots show similar shape to the previous HW configuration plots. In particular, we observe that the HDDs performed poorly, e.g., at highest peak IOps, SW reached 33%, 50% and 65% of HW's IOps. Also after the peak IOps, the performance deteriorates to a lower level than for the HW configuration. The response time is also observed to be a lot longer (e.g., at 15k IO request, x1 peaks 1.7s compared to HW's 5.2s). This shows that the HW RAID controller takes advantage of built-in memory to improve the performance. Changing from Write-Back to Write-Through mode, thereby effectively disabling the controller's cache, produced the similar results to the SW RAID configurations. On the other hand, SW RAID configurations for the SSDs appear to have no distinct disadvantage when compared with using HW RAID controller, as shown in the two plots on the right in Fig. 4.

In our previous work measuring low-end HDDs and SSDs [26], we found that the low-end disks' performance will quickly deteriorate after the peak IOps unlike the slow exponential curves we have observed for high-end disks. First generation low-end SSDs, known to suffer random-writes, performed much worse as the IO workload produces over 60% writes. For more details see [26].

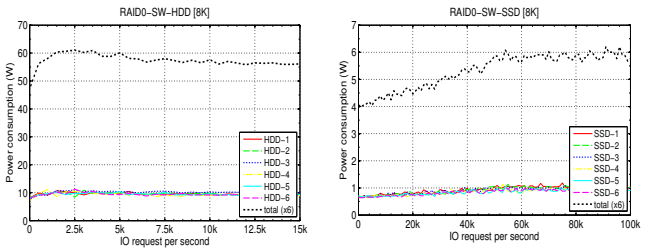

Fig. 5. Power Measurement for x6 disk configuration

B. Power Consumption

We find that in general each disk reaches a stable power consumption level, and these levels are observed to be correlated with peak IOps performance, e.g., high peak IOps (and throughput) coincided with a high stable power consumption level. The general pattern of power consumption is that an increase in power consumption occurs with an initial increase in IO requests, but then power consumption peaks and stays fairly constant for any further increase in IO requests. For high-end disks the increase happens slowly, but for low-end disks the power consumption reaches its peak quickly. For the HDDs we observed the power consumption ranged between 8W and 11W per disk; the most power (W) was consumed when the disk experienced the highest workload with random read/write IO requests requiring the most mechanical movement. When the disks receive sequential write requests (e.g., ASU-3), especially when configured in JBOD, they consumed the least power. As long as the disks are active with high IO requests, we find that individual disks consumed similar power to the vendor-supplied specifications.

Fig. 5 shows the x6 configured HDDs and SSDs power consumptions. Disks consume the least power at low IO requests and the most at around peak IOps (2.6k IO requests), and their power consumption actually drops after reaching the peak IOps. SSDs consumed the lowest power among all the disk types; as the IO requests increase, they consumed on average from 0.7W to 1.5W per disk. Again we observe power consumed was the highest at peak IOps (59k IO requests). Another advantage of the SSDs over the HDDs is that the power level variation is a lot smaller making it easier to plan for their power usage.

We also measured the power efficiency per IO request (mW/IOps) and clearly HDDs require a much higher mW/IOps than SSDs. We find that at low levels of IO requests, HDDs required about 950mW/IOps, and then drop to optimum point of 25mW/IOps. Similarly, SSDs start at 80mW/IOps and drop to an optimal point of 0.1mW/IOps. Generally, as IO requests increase mW/IOps drops quickly to the optimum point and then slowly increases again, indicating that the disks use more power per request for higher numbers of IO requests.

IV. Related work

One of the benchmarking standards that is related to SPC-1 is the TPC-C [15], which is an OLTP workload focusing only on database transactions (order-entry process) with its own measurement unit, transactions per minute (tpmC), for evaluating systems. Our CSB measurement functions similarly to the device used in [23], but ours do not use a current-shunt approach to find voltage-drops on the rails, and is modularized with separable parts to enable the changing of instruments (for example, to include a DAQ).

Performance and power consumption studies have been examined from small to large scales. For example, at the large scale of a cloud storage system, distributed meta-data servers (such as in [40]) are utilized to spread the load across multiple storage nodes and to increase reliability. Also, storage nodes can be configured to conserve energy with error correction [21] and with redundancy [38] approaches. A system-wide energy consumption model in [27] measures individual components of the CPU, DRAM, HDD, fan and system board, to combine them for benchmarking and predicting overall energy consumption. One of the conclusions of this study is that even when examining a single disk, there were varying results due to a diversity of application workloads.

Disk power management policies have been widely researched in particular. For instance, studies in [30], [35], [41] examined various optimal time-out selections for disk spin-up/spin-down to maximize the energy conservation while minimizing the impact on performance. Results of those studies were highly dependent on the workload generated by the user applications. Riska and Riedel [33] measured disk workload behaviors examining longitudinal traces. They observed that characteristics such as request/response times are environment dependent, however the ratio of read/write and access patterns are application dependent. They also found that the disk load is variable over a period of time, e.g., write traffic is more bursty than read traffic [34].

Allalouf et al. [17] measured separately 5V (board) and 12V (spindle) power values to model energy consumption behavior during random and sequential read/write requests. For example, with reasonable accuracy, a single IO per Watt is calculated to linearly approximate the total disk energy. Similarly, Hylick and Sohan [24] modeled power consumption levels of a disk's mechanical properties, e.g., estimating transfer and seek energy as well as finding major power level differences between the outer and inner section of a disk partition. Performance and energy measurement has also been studied in other system fields, such as in network routers [31].

V. Summary

With a wide range of versatile system configurations available, proper modeling and design of storage systems' configurations is becoming more important. Critical design factors within these models rely on accurate information about cost, performance and power consumption of storage system components. Our study introduced a measurement framework which has the ability to benchmark storage devices under 'typical' workload and measure performance and power consumption. In particular, we introduced the (`fiospc1`) tool for generating typical workloads and a CSB for power consumption measurement.

We tested our framework on commodity storage disks. We found that each disk has its range of performance (IOps and MiBps) with increasing IO requests. For instance, the IOps increase with increasing IO requests to a peak, and the IOps decrease after that. Behaviors across different disks and configurations were similar but the location and height of the peak depends on the disk and configuration used. Also, HDDs benefited a lot from using a HW RAID controller by IOps (improvement was shown over SW RAID of up to a factor of 3), while SSDs showed no apparent gain.

We also found that the power consumption by individual disks are similar to the vendor-supplied specifications or to that determined by conventional benchmarking. Nevertheless, one significant lesson is that one should not rely on vendor specifications for disk power usage without allowing for at least as much variability as we have observed in this study. Additionally, when designing storage systems, one can optimize the configuration of the ASUs on to the available disk types. For example, since ASU-1 generates mainly random read/write IO requests, high performing SSDs can be utilized. Also, since ASU-3 generates only the sequential writes, HDDs can be utilized, provided that the IO requests are kept low enough for the request pattern to remain "mostly" sequential.

Testing the different characteristics of various disk types available in the market would improve the diversity of our benchmark results, however measuring more disks in different configurations (e.g., including RAID 1) and testing distributed file system environments is ongoing research to be performed using our framework. Designing storage systems with good performance, low power use, high capacity and low total cost is our ultimate goal. The work presented here is a first step towards building a model that incorporates performance and power consumption of storage devices under typical workload.

REFERENCES

[1] "AOC-USAS2LP-H8iR," http://www.supermicro.com/products/accessories/addon/AOC-USAS2LP-H8iR.cfm.

[2] "Current Transducer LTS 6-NP," http://www.lem.com/docs/products/lts%206-np%20e.pdf.

[3] "fio," http://freshmeat.net/projects/fio/.

[4] "fio.git," http://git.kernel.dk/?p=fio.git.

[5] "Hard Disk Power Consumption Measurements," http://www.xbitlabs.com/articles/storage/display/hdd-power-cons.html.

[6] "httperf," http://sourceforge.net/projects/httperf/.

[7] "Iometer," http://www.iometer.org.

[8] "IOzone," http://www.iozone.org.

[9] "Iperf," http://sourceforge.net/projects/iperf/.

[10] "LabJack U6," http://labjack.com/u6.

[11] "LSI - MegaRAID Benchmark Tips, Jan-2010," http://www.lsi.com/DistributionSystem/AssetDocument/Benchmark_Tips_Jan_2010_v2_0.pdf.

[12] "OCZ Vertex 2 Pro Series SATA II 2.5" SSD," http://www.ocztechnology.com/products/solid-state-drives/2-5--sata-ii/maximum-performance-enterprise-solid-state-drives/ocz-vertex-2-pro-series-sata-ii-2-5--ssd-.html.

[13] "Storage Performance Council," http://www.storageperformance.org.

[14] "Storage Performance Council: Specifications," http://www.storageperformance.org/specs/SPC-1_SPC-1E_v1.12.pdf.

[15] "TPC-C," http://www.tpc.org/tpcc/.

[16] "WD Caviar Black 2 TB SATA Hard Drives (WD 2001FASS) ," http://www.wdc.com/en/products/products.asp?driveid=733.

[17] M. Allalouf, Y. Arbitman, M. Factor, R. I. Kat, K. Meth, and D. Naor, "Storage modeling for power estimation," in *SYSTOR '09: Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*. New York, NY, USA: ACM, 2009, pp. 1–10.

[18] S. Daniel, "Personal communication," 2010.

[19] S. Daniel and R. Faith, "A portable, open-source implementation of the spc-1 workload," *IEEE Workload Characterization Symposium*, pp. 174–177, 2005.

[20] J. D. Garcia, L. Prada, J. Fernandez, A. Nu nez, and J. Carretero, "Using black-box modeling techniques for modern disk drives service time simulation," in *ANSS-41 '08: Proceedings of the 41st Annual Simulation Symposium (anss-41 2008)*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 139–145.

[21] K. Greenan, D. D. E. Long, E. L. Miller, T. Schwarz, and J. Wylie, "A spin-up saved is energy earned: Achieving power-efficient, erasure-coded storage," in *Proceedings of the Fourth Workshop on Hot Topics in System Dependability (HotDep '08)*, 2008.

[22] F. Hupfeld, T. Cortes, B. Kolbeck, E. Focht, M. Hess, J. Malo, J. Marti, J. Stender, and E. Cesario., "Xtreemfs: a case for object-based storage in grid data management," in *Proceedings of 33th International Conference on Very Large Data Bases (VLDB) Workshops*, 2007.

[23] A. Hylick, R. Sohan, A. Rice, and B. Jones, "An analysis of hard drive energy consumption," in *Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on*, 2008, pp. 1 –10.

[24] A. Hylick and R. Sohan, "A methodology for generating disk drive energy models using performance characteristics," in *HotPower '09: Workshop on Power Aware Computing and Systems*. USENIX, October 2009.

[25] G. Laden, P. Ta-Shma, E. Yaffe, M. Factor, and S. Fienblit, "Architectures for controller based cdp," in *FAST '07: Proceedings of the 5th USENIX conference on File and Storage Technologies*. Berkeley, CA, USA: USENIX Association, 2007, pp. 21–21.

[26] D. Lee, M. O'Sullivan, and C. Walker, "Practical measurement of typical disk performance and power consumption using open source spc-1," in *Annual International Conference on Green Information Technology (GreenIT)*. GSTF, 2010, pp. 29–36.

[27] A. Lewis, S. Ghosh, and N.-F. Tzeng, "Run-time energy consumption estimation based on workload in server systems," in *HotPower '08: Workshop on Power Aware Computing and Systems*. USENIX, December 2008.

[28] M. Li, E. Varki, S. Bhatia, and A. Merchant, "Tap: table-based prefetching for storage caches," in *FAST'08: Proceedings of the 6th USENIX Conference on File and Storage Technologies*. Berkeley, CA, USA: USENIX Association, 2008, pp. 1–16.

[29] Y. Li, T. Courtney, R. Ibbett, and N. Topham, "On the scalability of storage sub-system back-end networks," in *Performance Evaluation of Computer and Telecommunication Systems, 2008. SPECTS 2008. International Symposium on*, 16-18 2008, pp. 464 –471.

[30] Y.-H. Lu and G. De Micheli, "Comparing system-level power management policies," *IEEE Des. Test*, vol. 18, no. 2, pp. 10–19, 2001.

[31] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2008, pp. 323–336.

[32] E. Riedel, "Green storage products: Efficiency with energy star & beyond," in *SNIA - Spring*, 2010.

[33] A. Riska and E. Riedel, "Disk drive level workload characterization," in *ATEC '06: Proceedings of the annual conference on USENIX '06 Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2006, pp. 9–9.

[34] ——, "Evaluation of disk-level workloads at different time scales," *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 2, pp. 67–68, 2009.

[35] D. C. Snowdon, E. Le Sueur, S. M. Petters, and G. Heiser, "Koala: a platform for os-level power management," in *EuroSys '09: Proceedings of the 4th ACM European conference on Computer systems*. New York, NY, USA: ACM, 2009, pp. 289–302.

[36] C. Walker, M. O'Sullivan, and T. Thompson, "A mixed-integer approach to core-edge design of storage area networks," *Comput. Oper. Res.*, vol. 34, no. 10, pp. 2976–3000, 2007.

[37] C. G. Walker and M. J. O'Sullivan, "Core-edge design of storage area networks-a single-edge formulation with problem-specific cuts," *Comput. Oper. Res.*, vol. 37, no. 5, pp. 916–926, 2010.

[38] J. Wang, H. Zhu, and D. Li, "eraid: Conserving energy in conventional disk-based raid system," *IEEE Trans. Comput.*, vol. 57, no. 3, pp. 359–374, 2008.

[39] J. Ward, M. O'Sullivan, T. Shahoumian, and J. Wilkes, "Appia: Automatic storage area network fabric design," in *FAST '02: Proceedings of the Conference on File and Storage Technologies*. Berkeley, CA, USA: USENIX Association, 2002, pp. 203–217.

[40] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: a scalable, high-performance distributed file system," in *OSDI '06: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2006, pp. 22–22.

[41] A. Weissel and F. Bellosa, "Self-learning hard disk power management for mobile devices," in *Proceedings of the Second International Workshop on Software Support for Portable Storage (IWSSPS 2006)*, Seoul, Korea, Oct. 26 2006, pp. 33–40.

[42] J. Yaple, "Benchmarking storage subsystems at home using spc tools," in *32nd Annual International Conference of the Computer Measurement Group*, 2006, pp. 21–21.

Dong,Jin Lee is a Post Doctoral Research Fellow since 2009 in the Department of Engineering Science at the University of Auckland, New Zealand. He is supported by The University of Auckland Faculty Research Development Fund (Engineering). He received PhD in Computer Science (Network Traffic Measurement) at the University of Auckland.

Michael O'Sullivan is a Senior Lecturer in the Department of Engineering Science at the University of Auckland. He received PhD in Management Science and Engineering from Stanford University, California, USA. He has worked on storage systems design since 1997 including collaborations with HP Labs (Palo Alto, California) and University of California Santa Cruz (UCSC).

Cameron Walker is a Senior Lecturer in the Department of Engineering Science at the University of Auckland. He received PhD in Pure Mathematics (Graph Theory) and Masters of Operations Research at the University of Auckland. He has been collaborating with Dr O'Sullivan on optimal storage systems design since 2001 and has also been involved in the collaboration with HP Labs and UCSC.

Acknowledgment. This project was supported by the KAREN Capability Build Fund administered by REANNZ.