

Beyond Power Proportionality: Designing Power-Lean Cloud Storage

Lakshmi Ganesh
Cornell University
lakshmi@cs.cornell.edu

Hakim Weatherspoon
Cornell University
hweather@cs.cornell.edu

Ken Birman
Cornell University
ken@cs.cornell.edu

Abstract

We present a power-lean storage system, where racks of servers, or even entire data center shipping containers, can be powered down to save energy. We show that racks and containers are more than the sum of their servers, and demonstrate the feasibility of designing a storage system that powers them up and down on demand; further, we show that such a system would save an order of magnitude more energy than current disk-based power-proportional storage systems. Our simulation results using file system traces from the Internet Archive show over 44% energy savings, a 5x improvement over disk-based power management systems, without performance impact. We explore the tradeoffs in choosing the right unit to power off/on, and present an automated framework to compute the optimal power management unit for different scenarios.

Categories and Subject Descriptors C [Computer Communication Networks]: Distributed Systems

General Terms Power Management

Keywords Cloud Storage, Power-Aware Storage

1. Introduction

This is an account of our exploration of low-power storage designs for the Internet Archive (IA) [2]. The IA is a petabyte-scale (and growing) online data repository, whose aim is to archive all of the world’s (public) data. Its collection currently comprises over 150 billion

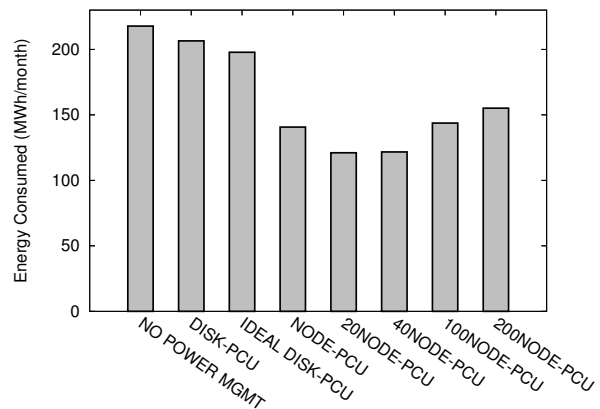


Figure 1. Energy Consumed For Different PCU Choices

web pages, as well as a media collection that includes millions of image and text files, and hundreds of thousands of audio and video files [2]. Brewster Kahle, the founder of this non-profit organization, is credited with inventing the concept of data center containers a full decade before they were adopted and made fashionable by market giants like Microsoft, Google, Amazon, Yahoo, etc. [11, 22]. He and his team at IA are now interested in designing the next-generation power-lean data center container – the GreenBox [19]; our work is related to this project.

In this paper, we share a key finding from our study - the role of the power cycle unit in storage power management. Power-proportional storage solutions power down idle IT components to save energy; we define the PCU as the unit chosen for powering down – e.g. disk, server, rack, or data center shipping container. Current solutions limit themselves to disk power cycling. Data Center (DC) containerization offers the unprecedented opportunity of treating an entire container as a component that can be turned on and off on demand.

Accordingly, we simulated a range of current solutions and compared them against a model where racks, or entire containers could be powered down. Our results strongly indicate that using larger PCUs can result in an order of magnitude more savings, and should be explored further. Figure 1 gives an overview of our results for the IA; a 20-node PCU results in an 80% improvement in power savings over single disk PCU, and a 30% improvement in power savings over single node PCU.

We posit that our findings have value beyond the IA. Firstly, low-power cloud storage design is of central importance today. We are in the midst of a data deluge[7] – even as you read this paper, about 100 GB of data are being generated every second, principally to be stored on hard disks [21]. Moreover, this number is doubling every 18 months [28] – faster than hard disk capacity growth (which doubles every two years [3]); with the result that the number of disks needed to store the world’s data is growing exponentially. An energy footprint that is proportional to the total data stored is, therefore, simply not sustainable. In this paper, we study the problem of scaling IA’s storage to meet the demands of the data deluge.

Secondly, we demonstrate the importance of looking beyond disk-power in designing low-power storage. Disk-power-based approaches [13, 15, 17, 23–25, 29] overlook a simple fact: 40% of the power drawn by a data center goes towards the overheads of cooling and power distribution [16], and is untouched by current solution designs (disks themselves, by comparison, consume only about 27% of the delivered power [29]). The great missed opportunity of cloud storage is in not doing more to amortize this sizeable chunk of the power cost of a data center. In this paper, we take a stab at quantifying the benefit of amortizing this overhead. We also take the first steps towards designing a practical system that can spin down entire DC containers.

We make the following contributions:

- We present a simple abstraction that concisely encapsulates the current power-proportional storage space, and shows its limitations.
- We present an automated framework, based on the above abstraction, that computes the performance and energy profile of different power management solutions for a range of storage system parameters.

- We quantify the benefits of moving to larger PCUs for the Internet Archive, and show the relevant trade-offs.

In the next section, we give some background on the IA. Section 3 surveys the power-proportional storage space, and presents a simple abstraction to describe it. We formally define PCUs in section 4, and show how to enable larger PCUs. Section 5 presents our simulation framework and our results. We discuss some practical implementation issues in section 6 and conclude in section 7.

2. The Internet Archive

The IA was founded in 1996 with the mission of providing “universal access to all knowledge” [2]. In a world turned digital, where information is as easy to wipe out as it is to create, the value of such an endeavor cannot be overstated. Society’s memory is only as large as its libraries, and it is the stated intent of the IA to help build as complete and long-lived a memory for society as possible, and make it accessible to everyone. What does “all knowledge” consist of? IA’s repository currently spans billions of webpages, millions of text files, hundreds of thousands of audio and video files, as well as a new software archive containing over a hundred thousand program files [2]. “Universal access” currently translates to everyone with access to the Internet; however, the IA has been actively working on broadening their reach beyond the Internet [20].

Before we go into further details about the IA, let us briefly explain why it makes for a uniquely interesting case study in the area of power-aware cloud storage: Firstly, it epitomizes the problem of scaling storage to meet the demands of the data deluge; its charter, after all, is to store *all* data. Secondly, the IA targets long-term preservation of (and immediate access to) data, rather than high-throughput data analysis and allied issues; in this it differs from data intensive computing services (which have tended to dominate the literature of late – ([9], [10], [12], etc.)). We believe these are orthogonal problems; once there is a sustainable framework for storing data at truly vast scales, data management/analysis services can be supported in a staged fashion. Finally, the IA is a not-for-profit organization, and operates under constraints (limited resources - money, people, etc.) that make the problem of scaling it more challenging; lean operation is not just desirable, but necessary in this context.

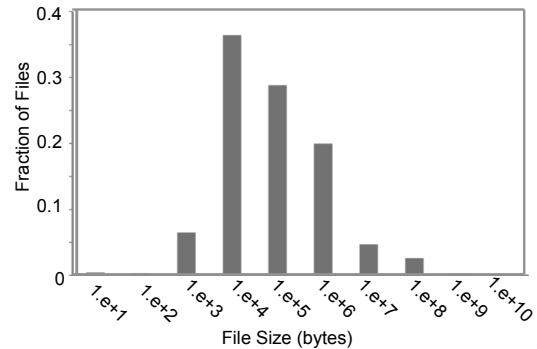
Let us now examine the current architecture and operation of the IA in order to identify its chief power-saving opportunities.

2.1 Architecture

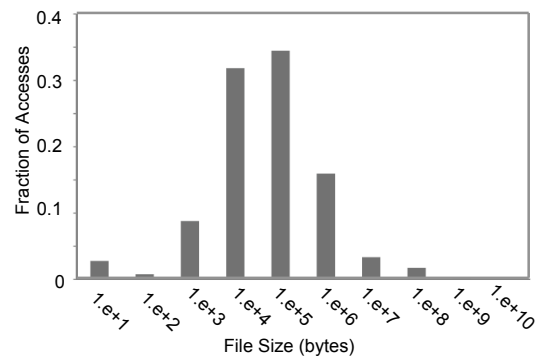
The IA offers two distinct (free) services: Wayback Machine, and Media Collection. The former offers snapshots of the World-Wide Web over time (since 1996), while the latter houses IA’s collection of text, image, audio and video files. There are some differences in how these two services function; in this study, we shall focus on IA’s Media Collection.

The IA’s Media Collection (MC) currently spans about 2 PB of data [8] that include public domain books, images, audio and video files (not including replicas). The service handles millions of requests daily, amounting to over 40 TB [18]. The MC stores contents as groups of files called “elements”. For example, video elements usually consist of the video file in several popular formats, while a book element might consist of several pages, each being represented as an image or text file. Elements are uploaded by users as well as IA staff, and are written to two dedicated “import nodes”; when these nodes fill up, two new nodes are drafted for the purpose. A monitoring service ensures that this two-way replication is maintained in the face of failures. Additional replicas are created based on item popularity – highly popular elements are manually replicated and distributed to other nodes for load balancing.

The IA employs about six front-end *web nodes* to handle user requests, and over 2500 back-end *storage nodes* to host content in their primary data center. The web nodes maintain a content index (a replicated MySQL database), where elements are indexed by their name and metadata (if any). User search terms are matched against this index to retrieve relevant element names. However, the content index maintains no information about the element location; to retrieve location information, web nodes broadcast a UDP message containing the relevant element names to all the storage nodes. Storage nodes maintain a list in memory of the names of all elements they store; and if they find a match among the requested elements, they respond to the broadcast message. The web node then redirects the user to the storage node, which serves the requested content. Storage nodes are typically commodity servers with a low-power CPU and four disks.



(a) Distribution of File Sizes Among Files



(b) Distribution of File Sizes Among Accesses

Figure 2. IA Workload Characteristics

It is worthwhile to pause here to partially explain what may appear surprising choices in the above description (manual load balancing, lack of location indices, item location by UDP-broadcasts, etc.). One of the IA’s guiding principles is simplicity in design [18]. Indeed, this is a necessity given their lean operations, with a very small staff count, high staff turnover, and limited resources. Complex systems potentially mean a higher initial outlay, greater risk of bugs, longer training time for new staff; all of which are luxuries the IA can ill-afford. Finally, the reality is that this extremely basic design has worked satisfactorily for over a decade, and has gained the IA a wide user base.

2.2 Workload

Figures 2 and 3 present some details of the IA workload. It is an almost read-only workload as data uploads

Data Type	HTTP logs
Data Duration	1 year
Data Source	886-node IA MC data center
Fraction of reads (GETs)	$\approx 99\%$
Access rate	$\approx 30\text{m}$ accesses/day

Figure 3. IA Workload Details

typically occur over a separate interface. Some important observations are:

1. There is a lot of data that never gets accessed (dark data)
2. Idle periods for nodes often last hours
3. Both primary and secondary (mirror) nodes respond to file requests
4. Average CPU utilization is very low

Many of these observations translate to power-saving opportunities.

2.3 Power-Saving Opportunities

- **Live data is a small fraction of total data:** The amount of data that is accessed at any time is a small fraction of the total data stored; further, a lot of data is never accessed for the interval we studied. This indicates that much of the data need not be kept live for much of the time.
- **All data is duplicated:** It is not necessary that all replicas be kept live at all times. We shall show how smart replica placement can create significant energy saving opportunities.
- **CPU utilization is low:** For every four disks consuming 10 W (each), there is one CPU consuming 200 W. Given the low average CPU utilization, one way to save power would be to increase the disk-to-CPU ratio.
- **Greenbox ideas:** For some other ideas to save power (that are orthogonal to the ones we discuss here), see [19].

3. Related Work: Power-Proportional Storage

The principle behind power-proportional storage is that power should track utilization; live data is usually a very small fraction of total data in any large-scale storage system, and it follows that considerable power can

be saved if the disks housing non-live data can be powered down.

The concept of power-proportional storage is less than a decade old, but the literature is already crowded with creative solutions. We present a brief survey here, and in doing so attempt to distil the principles that govern this space of solutions. A close examination of power-proportional storage solutions leads to the observation that they can be uniquely specified by two basic parameters:

1. *Data Localization Target:* Power-proportional storage schemes attempt to localize data accesses to a subset of the system so that the rest can be powered down. The data localization target parameter encodes this concept. For instance, MAID [13] concentrates popular data on a new set of “cache” disks, while PDC (Popular Data Concentration) [24] uses a subset of the original disk set to house the popular data. Power-aware caches [14] attempt to house the working set of spun-down disks in the cache, to increase their idle time. Write-offloading [23] is a technique that can layer on top of each of these solutions to temporarily divert write-accesses from spun-down disks to spun-up ones, and so is a scheme to localize *write* accesses. SRCMap [25] is similar to MAID and PDC (and additionally uses write-offloading), but is a more principled version of both. KyotoFS [15] is similar to write-offloading, but uses the log-structured file system to achieve write diversions.
2. *Architecture:* Power-proportional storage systems often add levels to the storage hierarchy in order to create disk power-down opportunities. The architecture parameter encodes the storage hierarchy of a given solution. For instance, the standard storage hierarchy puts primary memory (RAM) ahead of spinning disks. Power-proportional storage solutions add spun-down disks to the tail of this hierarchy. MAID uses an additional set of disks (cache-disks) between memory and the original disk set. PDC, power-aware caching, SRCMap, write-offloading, and KyotoFS all use the original disk set, and add no new levels. Hibernator [29] uses multi-speed disks, as does DRPM [17]. HP AutoRAID [27] divides the disk-set into a smaller, high-performance, high-storage-overhead RAID 1 level, and a larger, low-

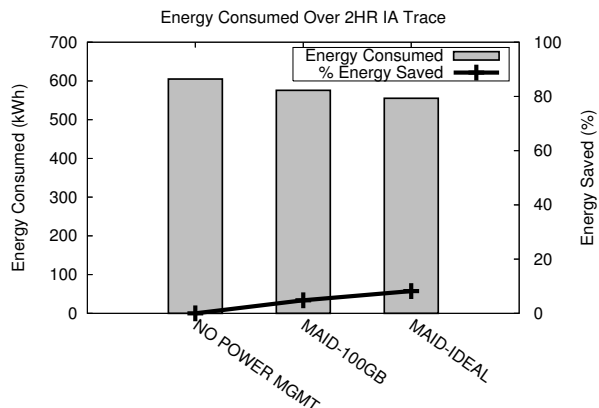


Figure 5. Limited Energy Savings From Disk Power Management

performance, low-cost RAID 5 level. PAR RAID [26] is a power-aware variant of AutoRAID.

Figure 4 describes a representative set of current solutions as instantiations of this basic abstraction. Under the “data localization target” heading, figure 4 lists the invariant that each solution attempts to maintain over time. The following notation is used:

D	$= \{d_1, d_2, \dots, d_n\}$	$=$ the set of disks
A	$= \{d_{n+1}, d_{n+2}, \dots, d_{n+m}\}$	$=$ additional set of disks, $m \geq 0$
C_i	$= \{\text{the contents of disk } d_i \text{ (excl. replicas)}\}$	
W_i	$= \{\text{the working set of disk } d_i\} \subseteq C_i$	
R_i	$= \{\text{the set of replicas on disk } d_i\}$	
S_i	$= \{\text{set of possible states of disk } d_i\}$	
s_i	$= \{\text{state of disk } d_i\}$	

3.1 Limitation of Disk Power Management Solutions

We built a power-aware storage system simulator, based on the above abstraction. Section 5.1 describes the simulator in detail, but we present a relevant result here in figure 5. For a 2-hour file access trace from IA, we configured the simulator to mirror one of the MC data centers (see table 9 for a listing of the parameters). We then simulated a MAID system with a 100GB MAID disk; the result was a 4.8% saving in energy. Further, we compared this with an idealized case where all of the back-end disks are powered off for the entire run; this case represents an ideal for any of the above disk power management solutions. The ideal

case resulted in a saving of 8.2%. The takeaway here is that disk power management solutions are limited in their benefit, with an upper limit (for IA) of less than 10% energy savings. Given the complexity of several of these solutions, the argument for their adoption is weak.

3.2 Beyond Power-Proportionality

As we saw above, the current power-proportional storage space has inherently limited benefit. The reason is that power-proportionality alone is not enough; a power-proportional storage solution could still waste significant amounts of energy in the following ways:

- As much as 40% of the power consumed by the storage system goes towards power distribution and cooling overheads [16]. While power-proportional storage solutions might help reduce cooling needs, they leave much of this overhead untouched. (Compare with disk power, which accounts for only about 27% of total storage power [29].)
- Storage systems typically replicate data for failure-resilience and/or performance. Mindful replica placement could allow some or all replicas to be turned off during periods of light load.
- Additional consumers of power, that are neglected by current solutions, include:
 1. The data center networking infrastructure
 2. Non-disk components of servers, such as CPU, memory, fan, etc.
 3. Non-IT DC components, such as lights, fail-over power generators, etc.

In essence, an extensive infrastructure exists to support the storage system – providing services such as power distribution, cooling, failure-resilience (redundancy), etc. – and any power-saving solution that neglects to take this into account is necessarily incomplete. The next section discusses how to go from power-proportional to power-lean.

4. Power Cycle Unit

We define the power cycle unit as the resource unit that the power management scheme operates over. This is the unit whose power state is manipulated to track utilization. For example, disk power management schemes manipulate the disk power state (ON/OFF/possibly low-power states corresponding to

Solution	Data Localization Target	Architecture
MAID	$\sum_{d_i \in A} C_i \supseteq \sum_{d_i \in D} W_i$	$m > 0, S_i = \{0, 1\}$
	Cache disks (A) used to hold working set of orig. disk set D . Disks can be on/off	
PDC	$\sum_{d_i \in D \wedge s_i=1} C_i \supseteq \sum_{d_i \in D} W_i$	$m = 0, S_i = \{0, 1\}$
	Powered-up disks in D contain working set of D . Disks can be on/off	
PA Cache	$C_{\text{cache}} \cup \sum_{d_i \in D \wedge s_i=1} C_i \supseteq \sum_{d_i \in D} W_i$	$m = 0, S_i = \{0, 1\}$
	Cache and powered-up disks in D together contain working set of D . Disks can be on/off	
SRMap	$\sum_{d_i \in D \wedge s_i=1} R_i \cup \sum_{d_i \in D \wedge s_i=1} C_i \supseteq \sum_{d_i \in D} W_i$	$m = 0, S_i = \{0, 1\}$
	Powered-up disks in D (orig. contents + replicas) contain working set of D . Disks can be on/off	
Hibernator	$\sum_{d_i \in D \wedge s_i > 0} C_i \supseteq \sum_{d_i \in D} W_i$	$m = 0, S_i > 2$
	Powered-up disks in D contain working set of D . Disks are multispeed	

Figure 4. The Current Power-Aware Storage Solution Space

lower speeds); CPU power management schemes manipulate CPU power (typically through frequency tuning). Our contention in this paper is that other PCU options, which have not been explored thus far, promise significantly bigger energy savings.

4.1 Key Opportunity: Modularity

The online services hosting space is evolving so rapidly that data center design standards are a moving target. However, they are characterized by one guiding principle – modularity. Agility, and rapid scalability are imperatives for successful online services – and both require modularity in design. Consider the facts: rapid expansion needs ushered in the concept of “commodity servers” – preassembled servers conforming to the most popular configurations prevalent in industry, ready for purchase off the shelf, deployable simply by plugging them into the data center. The concept has now expanded to racks, which are increasingly becoming the unit of choice for expansion. “Commodity racks” have servers, top-of-rack switches ([6]), power distribution units ([4]), and even in-rack cooling equipment ([1]) pre-installed. Purchasing and commissioning a rack is now a mere matter of hours – the “rack-and-roll” phenomenon [5]. Further along this path, entire data centers have now been commoditized – the data center shipping container – an idea that originated with the IA’s founder - Brewster Kahle.

This modularity at multiple levels translates to a new opportunity for power management solutions: we now have the ability to power down racks, or even entire containers. Each of these potential PCUs houses not only servers and disks, but also their correspond-

ing power distribution, networking, and cooling equipment; powering these down offers energy savings far beyond the limited disk power management space.

4.2 Enabling Different PCUs: PCU-Aware Data Organization

While larger PCUs are now physically possible, work is required to make them practical. Powering down a rack is not practical if it would result in service interruption or network disruption. However, as we suggest above, commodity racks exist that can be introduced into, or taken out of, the data center network without interrupting or disturbing service. These have their own network switch, power distribution unit (often software-controlled), and cooling equipment, and thus provide fault-isolation from the rest of the network. There is another issue, however – without some work, rack power-down opportunities (that is, all of the servers in the rack being simultaneously idle) are likely to be few. We shall now show how we could create power-down opportunities for different PCUs in the IA context, through appropriate data organization.

PCU-aware data organization essentially consists of two steps:

1. Each data item must be spread (striped/mirrored) *across* PCUs, rather than within them. Thus, assuming some degree of data redundancy, one or more host PCUs may be down without impacting the availability of that item.
2. Data access must be localized (as far as possible) to a subset of the PCUs so that others are idle and may be powered down. This is achieved by directing

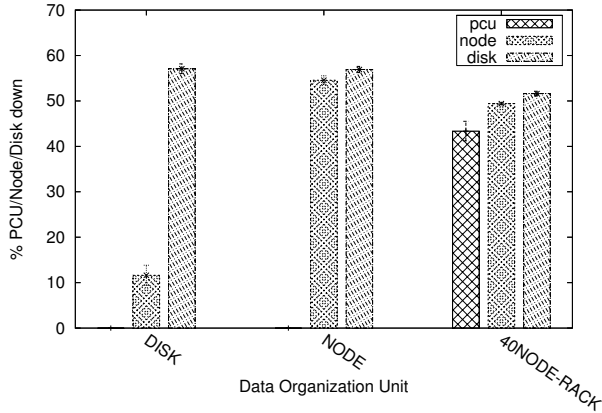


Figure 6. Impact of Data Organization Scheme on PCU Power-Down Opportunities

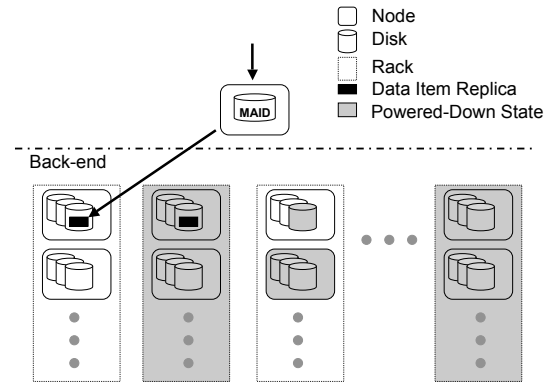
accesses to an item to the more active among its host PCUs.

Figures 7(a), and 7(b) illustrate PCU = Rack, and PCU = Node, respectively. Note how replica placement changes with PCU; note, also, the creation of idle PCUs through selective access of more active replica hosts. Figure 6 shows us the importance of PCU-aware data organization. Having set the PCU to 40-node racks, we varied the data organization unit (the unit across which replicas are distributed). As expected, we see that unless replicas are distributed across the given PCU (40-node racks, in this case), there are no opportunities for powering them down. When the replicas are distributed across disks, or nodes, we see plenty of disk and node power-down opportunities, but no rack power-down opportunity. Thus, PCU-aware data organization (and retrieval) is key to enabling larger PCUs.

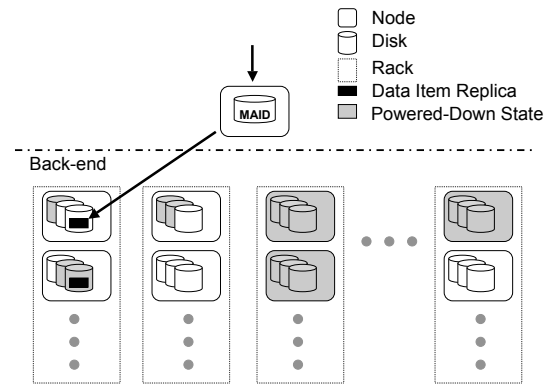
5. Evaluation

The aim of this study is to quantify the potential energy savings from using larger PCUs, for IA and beyond. We wish to answer the following questions:

1. *Internet Archive*: What choice of PCU maximizes energy savings for the Internet Archive without hurting performance?
2. *Beyond the IA*: How is this choice affected by system parameters such as data organization, request rate, and cache size?
3. *Simulator Sensitivity*: How far do simulator parameters affect the results?



(a) PCU=Rack



(b) PCU=Node

Figure 7. System Model

We describe our methodology, and then present our findings.

5.1 Methodology

We use simulations to explore the PCU space, for two reasons: Firstly, for a problem of this scale, a real deployment study is impractical. Secondly, we wish to explore a number of different PCU options, and the large combinatorial space of solutions and their configuration parameters does not allow for a practical deployment study.

5.1.1 Simulator

Our simulator models the power-proportional storage abstraction described in section 3, and allows different solutions to be simulated by specifying their architecture and data localization target. The model we

Parameter	Description	Value
Data Layout	Redundany scheme employed	PCU-aware, 2-way mirroring
Disk Power (W) (Up/Down/Tran)	Power consumed by disk when up, down, or transitioning between up and down	10/2/10
Node Power (W) (Up/Down/Tran)	Power consumed by node (over and above that consumed by its disks) when up, down, or transitioning between up and down	200/5/200
Rack Power Overhead (%) (Up/Down/Tran)	Power consumed by rack (over and above that consumed by its nodes) when up, down, or transitioning between up and down	50/0/50
Disk Access Time (ms)	Time taken to retrieve data from disk that is up	8
Disk Transition Time (s)	Time taken by disk to go between up and down states	6
Node Transition Time (s)	Time taken by node (over and above that taken by its disks) to go between up and down states	30
Rack Transition Time (s) (20/40/100/200)-node rack	Time taken by rack (over and above that taken by its component nodes) to go between up and down states	300/300/420/600
Power Check Interval (hr)	The intervals at which all PCUs are examined and idle ones powered down	0.5
Power Management Start Time (hr)	The interval after start of simulation when power checking begins	0.5
Disk Power Down Threshold	An exponentially weighted disk access count threshold below which the disk is considered idle	10
Target Disk Down Count	(optional) Force this target number of disks to be powered down during power checks, whether idle or not	50%
Cache Size	MAID disk capacity	100 GB
Number Of Nodes	Actual number from an IA MC data center	886
Number Of Disks/Node	Actual number from an IA MC data center	4

Figure 9. Simulator Parameters (applicable unless specified otherwise)

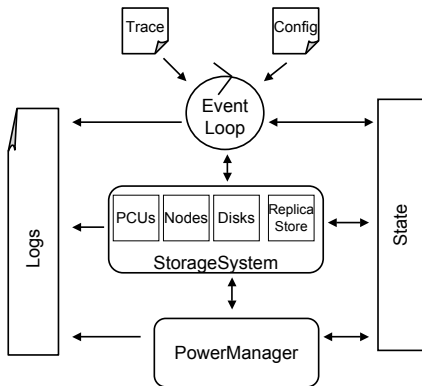


Figure 8. Simulator Architecture

work with for our PCU explorations is a MAID-style system, with PCU-aware back-end data organization. Given the system specifications (node and disk capacity, bandwidth, power ratings, PCU membership information, PCU power overhead, and transition time, etc), we simulate the progress of each file request through

the system, recording latency, power consumption, etc. Figure 7 shows the system model with PCU = Rack, and PCU = Node respectively.

The simulator is written in Python, and comprises less than 2000 lines of code. It is event-based, and takes as input a trace file of data accesses, as well as a configuration file that specifies the solution architecture, and the capacity, power and latency specifications of its components. It then models an execution of the specified solution on the input trace, and returns an execution log that details the power and performance profile of this run. Figure 8 shows the simulator architecture, while figure 9 presents the standard simulation parameters.

5.1.2 Data

For our experiments, we use the Internet Archive’s MC access logs for the week of April 3-9, 2009. These access logs have a read-ratio ($\frac{\# \text{ reads}}{\# \text{ accesses}}$) of very close to 1 (0.9926); we have, therefore, limited ourselves to reads in our experiments. Supporting writes is the subject of future work (see section 6 for more on this).

Attribute	Trace 1	Trace 2	Trace 3
Duration	6 hrs	6 hrs	6 hrs
# accesses	6.5m	7m	6.6m
Avg. access size (MB)	1.7	1.3	1.5
Max access size (GB)	7.73	20.74	7.73
Avg # accesses to a node	7797.77	8338.12	7862.95
Max # accesses to a node	110322	184424	120983
# Nodes accessed	833	838	835

Figure 10. Trace Characteristics

Unless otherwise specified, each data point presented in the following section is the averaged result of running 6-hour traces from three different days of this week (a Monday, Tuesday, and Friday, the same set of hours being picked from each day). Figure 10 gives details of these traces.

The traces are basically HTTP GET logs, and specify, for each file access, the access time, the file details (name, size), as well as the storage node details (id, disk number). These accesses are essentially cache-misses from the front-end web nodes. Recall that file location (for a cache-miss) is obtained by UDP broadcast to all the storage nodes. These accesses, thus, provide the storage node data as well. However, we manipulate this information slightly to conform to different data organization layouts. Given a data organization scheme – PCU-aware, 2-way mirroring, for example – we statically map each disk to a “mirror disk” such that the mirror disk is on a different PCU from the original disk. An access request to any item on either disk is then directed to the more active of the two. Support for dynamic, per-file mapping is planned in future work.

5.1.3 Internet Archive

Question: What is the optimal PCU size for the IA? For the parameter set listed in table 9, which is intended to approximate the IA store, we ran a 24-hour trace (from April 3, 2009). Our findings are shown in figure 11. Figure 11(a) shows that as we increase PCU size, a sweet-spot (minimum) is achieved for energy at the two configurations PCU = 20-node rack, and PCU = 40-node rack. At these configurations, we obtain energy savings over disk power management solutions of over 44%, and over node power management solutions of 30%. In energy, this translates to over 3MWh saved per day. Further, figure 11(b) shows that the 20-, and 40-node PCU configurations actually perform somewhat better than the node PCU configuration! Each set

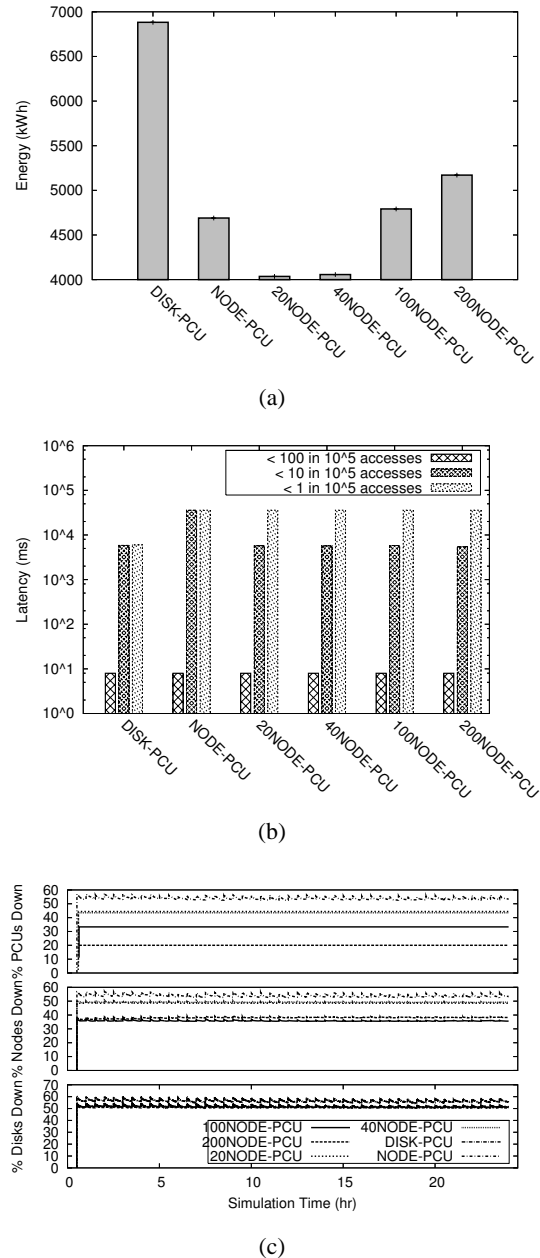
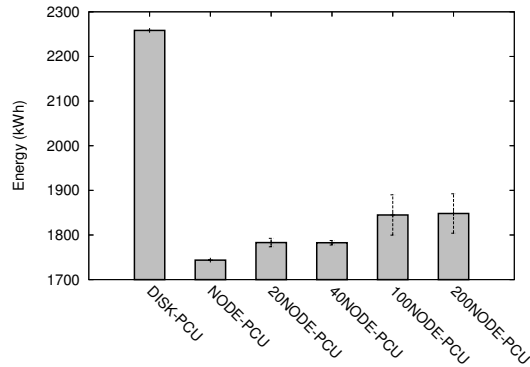


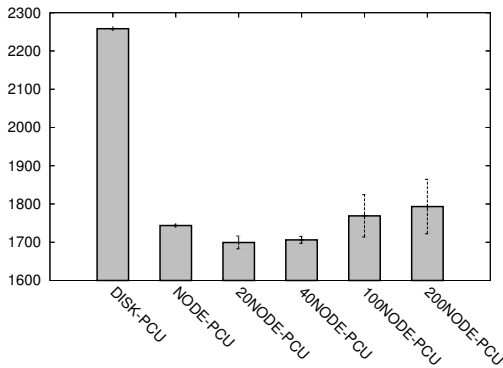
Figure 11. Computing Optimal PCU Size for the Internet Archive

of three bars in this graph shows the the highest latency seen in the 99.9-, 99.99-, and 99.999- th percentile of accesses respectively (left-to-right). We see that all configurations have acceptable performance, with over 99.9% accesses seeing no delay.

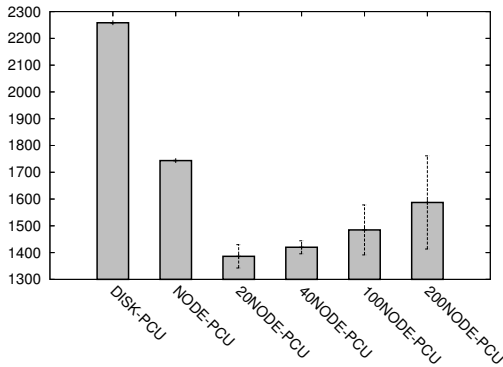
Figure 11(c) explains this latency distribution. For each configuration, it tracks the number of PCUs, nodes, and disks that are powered down over the length of the simulation. We see that for all of the configu-



(a) 5% Rack Overhead



(b) 25% Rack Overhead



(c) 100% Rack Overhead

Figure 12. Effect of Rack Power Overhead on Optimal PCU Size

rations with $PCU > node$, the number of PCUs down stays constant after the initial power check interval. This means that no access goes to a powered-down PCU, with the result that PCU power downs have no performance penalty!

Note: the remaining results all use three 6-hour traces to generate each data point.

Question: How does this choice of optimal PCU depend on Rack Power Overheads? Clearly, the higher the power overhead of a rack, the more energy savings obtained by powering it down. For our results above, we used a rack power overhead of 50%; so, for example, a 40-node rack would have a power overhead of $\frac{50}{100} * (40 * 200) = 4000W$. We chose this as a reasonably conservative value, given the industry rule-of-thumb that 1W of cooling is needed for every Watt going to servers (ie.. a 100% overhead). However, we would like to compute the minimum rack power overhead, at which it becomes worthwhile to consider PCUs that are greater than node. Figure 12 shows that this minimum overhead value is close to 25%. At overhead values of 5% or less, we actually waste energy if we power down racks. However, at overhead values of 25% and over, a 20-, or 40-node rack is the optimal PCU for the IA, with energy savings increasing with overhead.

Question: How does this choice of optimal PCU depend on Rack Transition Time? As rack transition time increases, we expect that the energy savings from powering the rack down, decreases. Therefore, we can expect a maximum rack transition time beyond which powering down racks does not make sense. However, as we see in figure 13, this limit is not reached for the transition time values we explored. Even at a conservative estimate that it takes 10 minutes to power up a 40-node rack – this is in addition to the power-up time of its component nodes – we see that the 40-node rack continues to be an optimal PCU choice for the IA (figure 13(b),13(d)). We also see that halving, or doubling the transition time does not affect performance significantly – which is in agreement with our earlier observation that no accesses hit powered-down racks.

Takeaway: Expanding the PCU from node to rack results in a 655kWh energy savings in just one day, in a very small (886-node) container. Extrapolating from these results, we could expect to save about 20MWh per month, per container, just using this simple technique. In what follows, we show how these savings can be further increased, and generalized beyond the IA context.

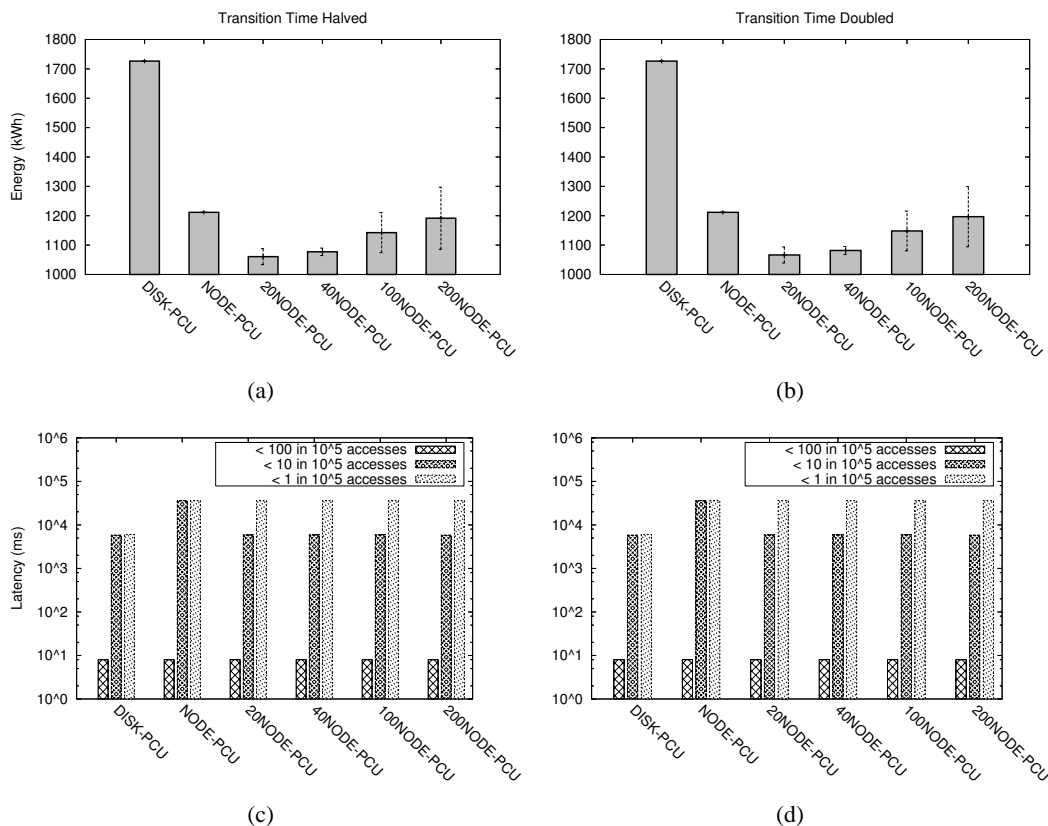


Figure 13. Effect of Rack Transition Time on Optimal PCU Size

5.1.4 Beyond IA

Question: How does Optimal PCU Choice depend on Data Organization Scheme? We now look beyond IA’s two-way mirroring, and see how PCU choice is affected by different data organization schemes. Figure 14 shows the result of running the same trace over a succession of data striping schemes – (n, m) , where n is the total number of chunks in a stripe, and m is the least number of chunks needed to reconstruct the data item. Each configuration is represented as nm_PCU -size, where PCU-size can either be node, or 40-node rack. We see that energy savings increase as overhead (n/m) increases (the higher the overhead of the striping scheme, the more redundant fragments there are whose host PCUs can be powered down), and decrease as fragmentation rate (n) increases (the higher the fragmentation rate, the bigger the set of PCUs each data item is spread over; thus increasing inter-PCU dependencies, and reducing PCU power-down opportunities). As a concrete example, we see that energy savings increase as we increase overhead from $(6,4)$ to $(6,3)$. On

the other hand, energy savings decrease as we increase fragmentation from $(2,1)$ to $(6,3)$ to $(8,4)$. We also see that, for all striping schemes, setting the PCU to node leads to having higher node and disk down-counts; consequently, node power cycling has more latency spikes than rack power cycling.

Question: How does Optimal PCU Choice depend on Cache Size? Another parameter of interest is cache size (note that ‘cache’ here refers to MAID disks). Large storage designs increasingly use significant amounts of non-volatile memory as a read/write cache. Terabytes of NV-RAM, or solid state storage are now well within the realms of possibility. We wished to see how much cache size impacted power-down opportunities. Figure 15 shows the surprising result that cache size has little impact on energy savings. The explanation is that our traces consist of accesses that missed front-end caches; this workload, therefore, is inherently resistant to caching.

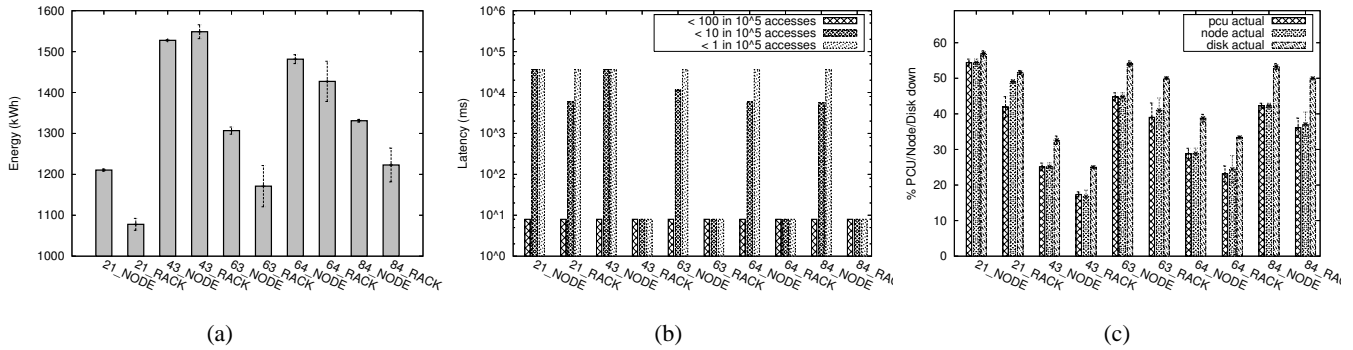


Figure 14. Impact of Data Organization Scheme on Optimal PCU Size

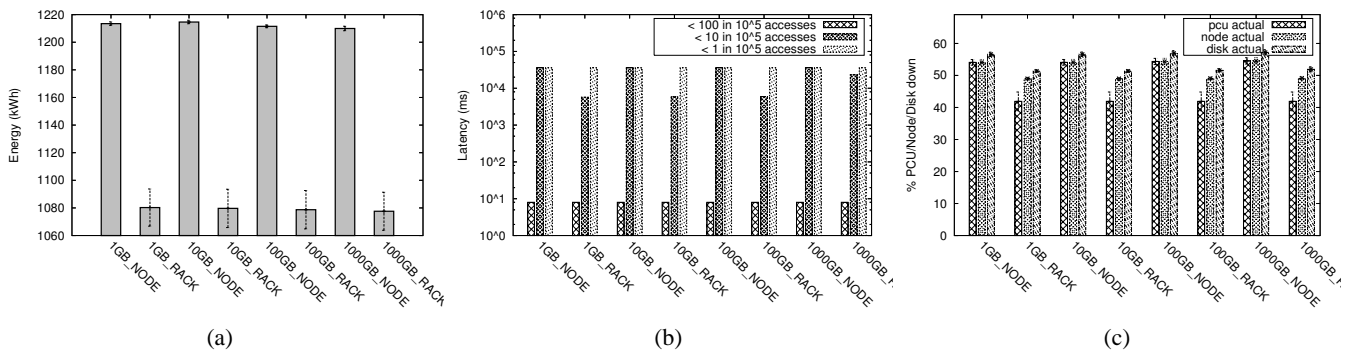


Figure 15. Impact of Cache Size on Optimal PCU Size

Question: How does Optimal PCU Size depend on Access Rate? Finally, we partially address the question of how optimal PCU size depends on file access rates by looking at the results from two different traces, one having 1.4 times the original access rate (figure 16(a)), the other 0.4 times the original access rate (figure 16(b)). While this doesn't comprise a wide range of access rates, it does show that energy savings increase when access rate is lower, but the choice of optimal PCU size does not change over the access rates we explored.

5.1.5 Simulator Sensitivity

It is important to ensure that our results are not artifacts of the simulator settings. We now show that our results are fairly robust to simulator fine-tuning.

Figure 17 shows that our findings are largely independent of changes in the power-check interval, target disk-down count, and disk-down threshold. In figure 17(b), we see that energy savings are significantly reduced by over-aggressive disk power down (forcing 100% of the disks to be down at every power-check interval). However, PCU-size ordering with respect to

total energy consumption is not affected by these simulator tuning parameters. Thus, whatever the value of the power-check interval, target disk-down count, or disk-down threshold, the choice of optimal PCU stays the same.

6. Discussion

We have examined PCU choices for a range of different storage system settings; our findings strongly suggest that disk power management is a dead end, and larger PCUs are a very promising direction to follow. However, there are some issues we did not address during our discourse; we examine some of them here.

- **What About Writes?** IA's read-mostly workload allowed us to concentrate on reads to the exclusion of writes. While we plan, in future work, to revisit our design to add support for writes, we believe that our findings are still widely applicable. The reason is that any truly global-scale service will need to design separately for the hot data (which includes writes) and the less-hot data. With very large volumes of data, storage for hot data (necessarily

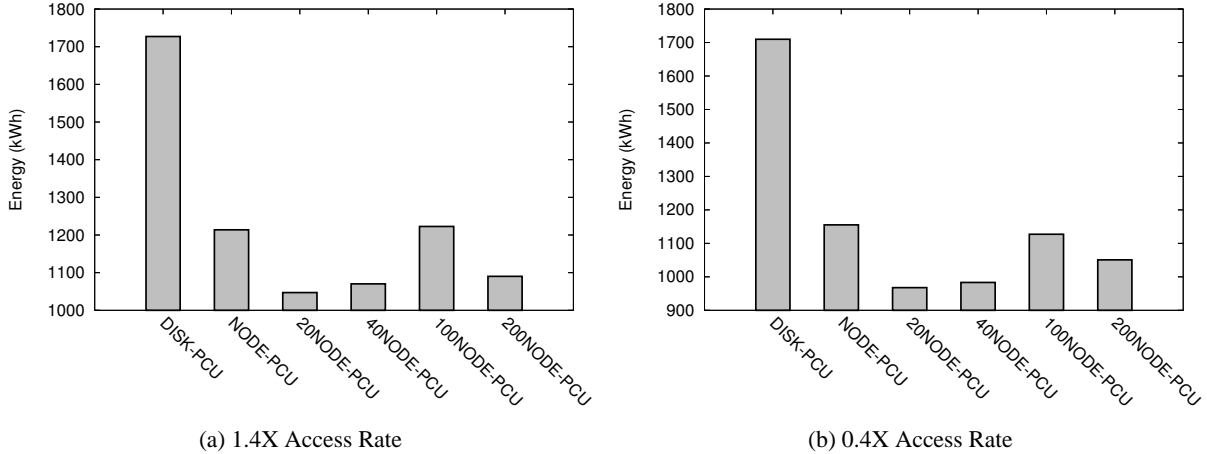


Figure 16. Effect of Access Rate on Optimal PCU Size

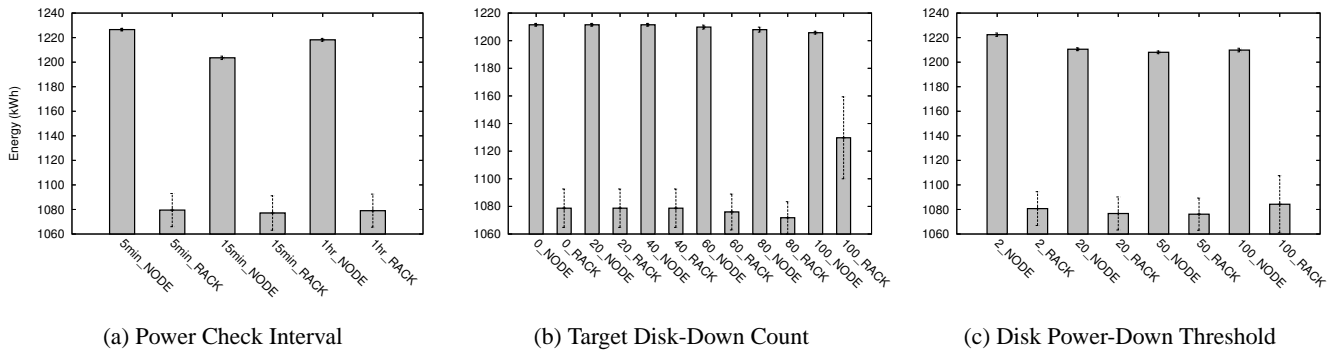


Figure 17. Result Sensitivity to Simulator Settings

a small fraction of the total data) can be designed for performance, while the much larger back-end store will need to be designed for volume. Our PCU schemes are targeted at this back-end store.

- *What About Bandwidth?* One issue with PCU-aware data organization is that it goes against the design principle of putting nodes that communicate a lot in the same rack, or under the same switch. When PCU = Rack, we spread data across PCUs, thus potentially impacting data retrieval latency. We are working on factoring this cost into our model.
- *Powering Down versus Over-Subscription:* An oft-made argument against power-aware storage solutions is that it is economically better to put idle equipment to use rather than to power it down. This argument breaks down in a large-scale data storage scenario. PB-scale data stores, even at the outer limit of their bandwidth capabilities, cannot serve all of the data they host simultaneously. As data contin-

ues to grow, it is necessary to separate the problem of storage from computation; the former must emphasize scalability and hence power-awareness. The latter can be designed on top of the storage solution in a staged fashion.

7. Conclusion

Information is the currency of our times, and as the volume of digital data continues to grow exponentially, designing power-lean, sustainable storage systems assumes central importance. We show that the current power-proportional storage space has limited potential, and that in order to scale with the data, we need to go beyond power-proportionality towards power-lean systems that address the overheads of cooling, power distribution, and networking. We show how to design systems that can power cycle over racks, or even entire data center containers, with an order of magnitude improvement in energy savings.

Acknowledgments

We would like to thank Brewster Kahle and the IA folk for sharing their data, and for their whole-hearted support and enthusiasm. We hope this work will be useful to them. We would also like to thank Deniz Altinbuken for her help in gathering statistics about the IA data. Finally, we are grateful to AFRL, NSF and Microsoft for their support of this effort.

References

- [1] High density in-rack cooling solutions for server racks, computer rooms, server rooms & data centers. URL <http://www.42u.com/cooling/in-rack-cooling/in-rack-cooling.htm>.
- [2] The internet archive. <http://www.archive.org>.
- [3] Moore's law. Wikipedia.
- [4] Switched rack pdu. URL <http://www.apc.com/products/family/index.cfm?id=70>.
- [5] Cisco data center infrastructure 2.5 design guide, 2007.
- [6] Data center top-of-rack architecture design, 2009.
- [7] Data, data everywhere. Special Report, The Economist, February 25 2010.
- [8] In Personal Communication with Brewster Kahle and the Internet Archive Staff, January 14 2010.
- [9] H. Amur, J. Cipar, V. Gupta, G. Ganger, M. Kozuch, and K. Schwan. Robust and flexible power-proportional storage. In *Proceedings of Symposium on Cloud Computing (SOCC)*, 2010.
- [10] D. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan. Fawn: A fast array of wimpy nodes. In *Proceedings of Symposium on Operating Systems Principles (SOSP)*, 2009.
- [11] B. Baumgart and M. Laue. Petabyte box for internet archive, November 2003.
- [12] A. Caulfield, L. Grupp, and S. Swanson. Gordon: Using flash memory to build fast, power-efficient clusters for data-intensive applications. In *Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2009.
- [13] D. Colarelli, D. Grunwald, and M. Neufeld. The case for massive arrays of idle disks (maid). In *Proceedings of the conference on File and Storage Technologies (FAST)*, 2002.
- [14] Q. Z. Francis, F. M. David, C. F. Devaraj, Z. Li, Y. Zhou, and P. Cao. Reducing energy consumption of disk storage using power-aware cache management. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, February, 2004.
- [15] L. Ganesh, H. Weatherspoon, M. Balakrishnan, and K. Birman. Optimizing power consumption in large scale storage systems. In *HotOS*, 2007.
- [16] A. G. Greenberg, J. R. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. *Computer Communication Review*, 2009.
- [17] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. Drpm: Dynamic speed control for power management in server class disks. *Computer Architecture, International Symposium on*, 2003.
- [18] E. Jaffe and S. Kirkpatrick. Architecture of the internet archive. In *Proceedings of The Israeli Experimental Systems Conference (SYSTOR)*, 2009.
- [19] B. Kahle. Project greenbox. <http://backyardfamilyfarm.wikispaces.com/Project+Greenbox>.
- [20] R. Kaushik. Spreading the digital word. ExtremeTech, April 29 2003. URL <http://www.extremetech.com/article2/0,3973,1047454,00.asp>.
- [21] P. Lyman, H. Varian, P. Charles, N. Good, L. Jordan, and J. Pal. How much information? executive summary. School of Information Management and Systems, UC-Berkeley, 2003.
- [22] C. Metz. Sun packs 150 billion web pages into meat locker. March 2009.
- [23] D. Narayanan and A. Donnelly. Write off-loading: Practical power management for enterprise storage, 2008.
- [24] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers, 2004.
- [25] A. Verma, R. Koller, L. Useche, and R. Rangaswami. Energy proportional storage using dynamic consolidation. In *Proceedings of the File and Storage Systems*, 2010.
- [26] C. Weddle, M. Oldham, J. Qian, A.-I. A. Wang, P. Reiher, and G. Kuenning. Paraid: A gear-shifting power-aware raid. In *Proceedings of File And Storage Technologies (FAST)*, 2007.
- [27] J. Wilkes, R. Golding, C. Staelin, and T. Sullivan. The hp autoraid heirarchical storage system. In *Proceedings of ACM Transactions on Computer Systems (TOCS)*, 1996.
- [28] E. Woollacott. Digital content doubles every 18 months. TG Daily, May 19 2009.
- [29] Q. Zhu, Z. Chen, L. Tan, and Y. Zhou. Hibernator: helping disk arrays sleep through the winter. In *Proceedings of the twentieth ACM Symposium on Operating Systems Principles (SOSP)*, 2005.