Beyond Power Proportionality: Designing Power-Lean Cloud Storage

Lakshmi Ganesh, Hakim Weatherspoon, Ken Birman
Cornell University
{lakshmi,hweather,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.**

I. Introduction

This is an account of our exploration of low-power storage designs for the Internet Archive (IA) [3]. 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 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 [3]. 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. [10], [22]. He and his team at IA are now interested in designing the next-generation power-lean data center container – the GreenBox [20]; 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 (PCU) 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 unprecendented 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. We find that 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[4] – 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 [29] – faster than hard disk capacity growth (which doubles every two years [2]); 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 ([14], [18], [25], [30], [16], [24], [26]) overlook a simple fact: 40% of the power drawn by a data center goes towards the overheads of cooling and power distribution [17], and is untouched by current solution designs (disks themselves, by comparison, consume only about 27% of the delivered power [30]). 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.

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

II. The Internet Archive

The IA was founded in 1996 with the mission of providing "universal access to all knowledge" [3]. The 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 [3]. "Universal access" currently translates to everyone with access to the Internet.

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 – ([7], [8], [11], 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.

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 [5] 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 [19]. Content is uploaded by users as well as IA staff, and is 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 centers. 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.

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 [19]. 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.

III. 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. 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 [14] concentrates popular data on a new set of "cache" disks, while PDC (Popular Data Concentration) [25] uses a subset of the original disk set to house the popular data. Power-aware caches [15] attempt to house the working set of spun-down disks in the cache, to increase their idle time. Write-offloading [24] 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 [26] is similar to MAID and PDC (and additionally uses write-offloading), but is a more principled version of both. KyotoFS [16] 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 [30] uses multi-speed disks, as does DRPM [18]. HP AutoRAID [28] divides the disk-set into a smaller, high-performance, high-storage-overhead RAID 1 level, and a larger, low-performance, low-cost RAID 5 level. PARAID [27] is a power-aware variant of AutoRAID.

We found that the power-aware storage abstraction specified by these two parameters encapsulates much of the current solution space; We simulated this abstraction to explore the space.

A. Limitation of Disk Power Management Solutions

We built a power-aware storage system simulator, based on the above abstraction. Section V-A describes the simulator in detail, but we present a relevant result here in figure 1. For a 2-hour file access trace from IA, we configured the simulator to mirror one of the MC data centers (see table 4 for a listing of the parameters). We then simulated a MAID system with a

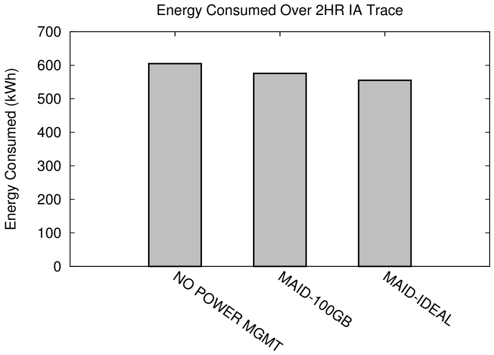


Fig. 1: Limited Energy Savings From Disk Power Management

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.

B. *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 [17]. 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 [30].)
- 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.

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

A. *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. 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 ([13]), power distribution units ([9]), and even in-rack cooling equipment ([6]) pre-installed. Purchasing and commissioning a rack is now a mere matter of hours – the "rack-and-roll" phenomenon [12]. 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 corresponding power distribution, networking, and cooling equipment; powering these down offers energy savings far beyond the limited disk/server power management space.

B. *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

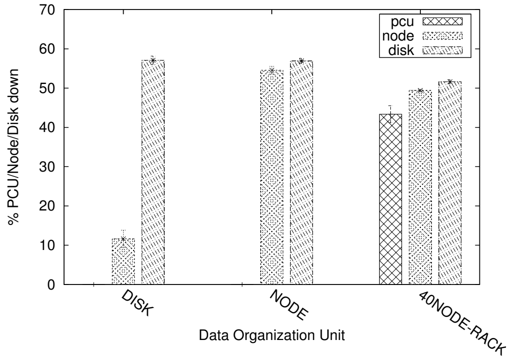


Fig. 2: Impact of Data Organization Scheme on PCU Power-Down Opportunities

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

Figures 3(a), and 3(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 2 illustrates 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.

V. Evaluation

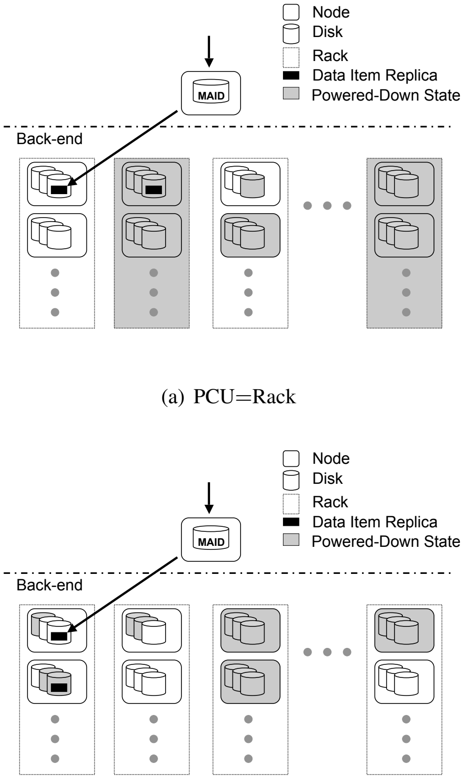
The aim of this study is to quantify the potential energy savings from using larger PCUs, for IA and beyond. We wish to answer the following questions:

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?

We describe our methodology, and then present our findings.

A. 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 makes it a natural candidate for a simulation study.



(a) PCU=Rack

(b) PCU=Node

Fig. 3: System Model

1) Simulator: Our simulator models the power-proportional storage abstraction described in section III, and allows different solutions to be simulated by specifying their architecture and data localization target. The model we work with for our PCU explorations is a MAID-style system, with PCU-aware back-end data organization. Given the system specifications, we simulate the progress of each file request through the system, recording latency, power consumption, etc. Figure 3 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 accessses, 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 4 presents the standard simulation parameters. We validated the simulator in two ways; we compared its findings with measurements from a real storage node and ensured that the simulator's disk-level storage model is sufficiently accurate. Further we used actual measurements from an IA production facility to inform our choice of node and rack transition times and power overheads.

2) Data: For our experiments, we use traces from one of the MC data centers, which contains 886 storage nodes.

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
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

Fig. 4: Simulator Parameters (applicable unless specified otherwise)

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

Fig. 5: Trace Characteristics

We use access logs for the week of April 3-9, 2009. 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 5 gives details of these traces.

The traces are HTTP logs, and specify, for each file access, the access time, the file (name, size), as well as the host storage node (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.

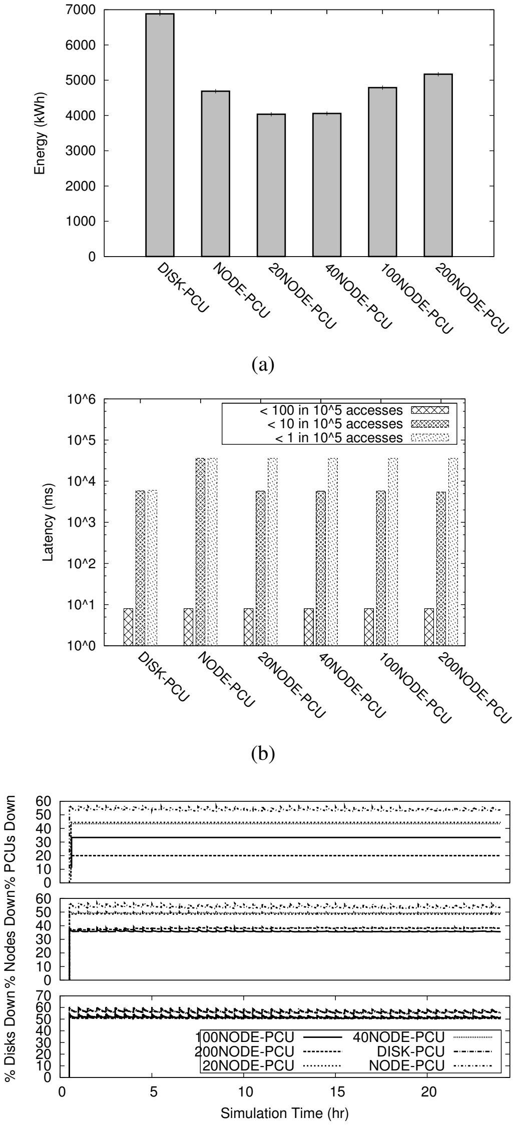
B. Results

1) Internet Archive:

Question: What is the optimal PCU size for the IA? For the parameter set listed in table 4, which is intended to approximate the IA store, we ran a 24-hour trace (from April 3, 2009). Our findings are shown in figure 6. Figure 6(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%. Further, figure 6(b) shows that the 20-, and 40-node PCU configurations actually perform somewhat better than the node PCU configuration! Each set of three bars in this graph shows 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 6(c) explains why the rack PCU configurations perform better than the node PCU configuration. 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 configurations 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 rack, with the result that rack 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



(a)

(b)

(c)

Fig. 6: Computing Optimal PCU Size for the Internet Archive

would like to compute the minimum rack power overhead at which it becomes worthwhile to consider PCUs that are greater than node. Figure 7 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 down the rack decreases. Therefore, we might expect a maximum rack transition time beyond which powering down racks does not make sense. However, as we see in figure 8, 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. We also found (though not shown here) 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.

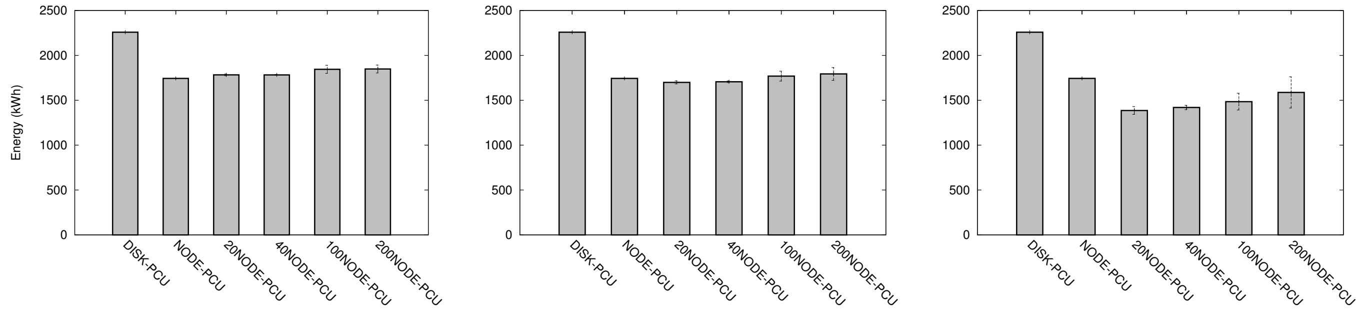
2) Beyond Internet Archive:

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 9 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? We wanted to isolate the effect of the cache size (note that 'cache' here refers to MAID disks) on PCU power-down opportunities. In our experiments with varying cache sizes, we observed the surprising result (omitted here due to space constraints) that cache size has negligible 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.

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 10(a)), the other 0.4 times the original access rate (figure 10(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.

Finally, though we omit the results due to space constraints, we also conducted experiments to confirm that our results are not artifacts of simulator-specific parameters such as power-

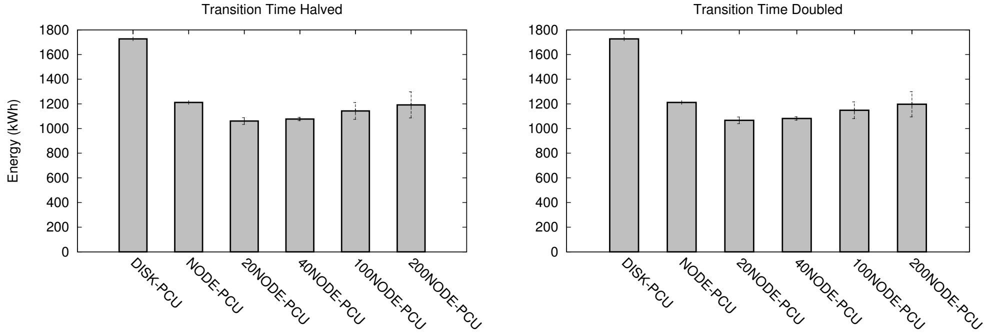


(a) 5% Rack Overhead

(b) 25% Rack Overhead

(c) 100% Rack Overhead

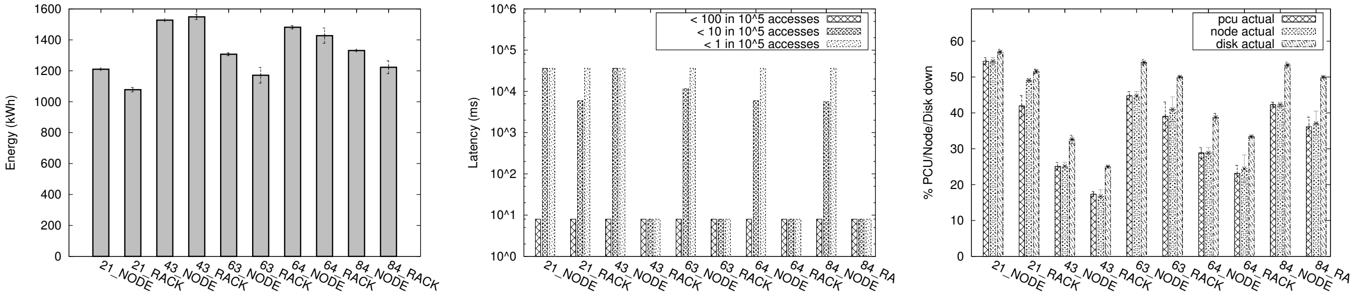
Fig. 7: Effect of Rack Power Overhead on Optimal PCU Size



(a)

(b)

Fig. 8: Effect of Rack Transition Time on Optimal PCU Size

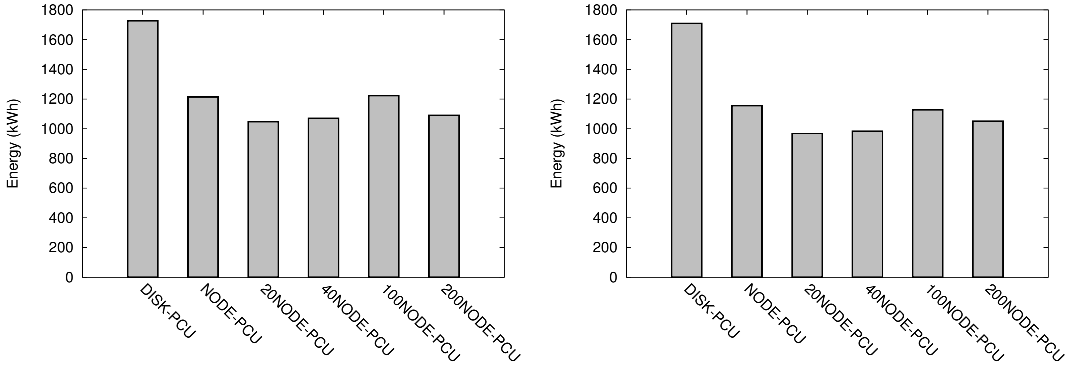


(a)

(b)

(c)

Fig. 9: Impact of Data Organization Scheme on Optimal PCU Size



(a) 1.4X Access Rate

(b) 0.4X Access Rate

Fig. 10: Effect of Access Rate on Optimal PCU Size

check interval, disk-down threshold, and disk power-down threshold.

VI. 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.

- *Power Usage Effectiveness (PUE):* PUE is defined as $\frac{\text{Total power}}{\text{IT power}}$, and the industry average is 2.0 to 2.5 [23]. As figure 7 shows, rack-sized PCUs improve energy efficiency for facilities with PUE exceeding 1.25, while node-sized PCUs suffice for more efficient facilities. Modern green data centers have reported PUE values as low as 1.07 [1]; a feat achieved by using outside air for cooling (*free cooling*), thus obviating the need for energy-hungry chillers. We note that in these cases the node is the optimal PCU – moving to a larger PCU has scant benefit. However, the vast majority of existing data centers do not fall into this category – they use chillers either because outside temperatures do not permit free cooling, or because their legacy design does not allow it.
- *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 continues 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.

VII. 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.

This work was supported by Internet Archive, AFRL, NetApp, Microsoft, NSF TRUST

References

[1] Facebook Open Compute Project. http://www.opencompute.org.

[2] Moore's Law. Wikipedia. http://en.wikipedia.org/wiki/Moore's_law.

[3] The Internet Archive. http://www.archive.org.

[4] Data, data everywhere. The Economist, February 25 2010.

[5] In Personal Communication with Brewster Kahle and the Internet Archive Staff, January 14 2010.

[6] 42U. *High Density In-Rack Cooling Solutions for Server Racks, Computer Rooms, Server Rooms & Data Centers*. http://www.42u.com/cooling/in-rack-cooling/in-rack-cooling.htm.

[7] Hrishikesh Amur, James Cipar, Varun Gupta, Gregory Ganger, Michael Kozuch, and Karsten Schwan. Robust and flexible power-proportional storage. In *Symposium on Cloud Computing (SOCC)*, 2010.

[8] David Andersen, Jason Franklin, Michael Kaminsky, Amar Phanishayee, Lawrence Tan, and Vijay Vasudevan. FAWN: A Fast Array of Wimpy Nodes. In *Symposium on Operating Systems Principles (SOSP)*, 2009.

[9] APC. *Switched Rack PDU*. http://www.apc.com/products/family/index.cfm?id=70.

[10] Bruce Baumgart and Matt Laue. Petabyte Box for Internet Archive, November 2003.

[11] Adrian Caulfield, Laura Grupp, and Steven Swanson. Gordon: Using flash memory to build fast, power-efficient clusters for data-intensive applications. In *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2009.

[12] Cisco. *Cisco Data Center Infrastructure 2.5 Design Guide*, 2007. http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCI_SRND_2_5_book.html.

[13] Cisco. *Data Center Top-of-Rack Architecture Design*, 2009. http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9670/white_paper_c11-522337.html.

[14] Dennis Colarelli, Dirk Grunwald, and Michael Neufeld. The Case for Massive Arrays of Idle Disks (MAID). In *File and Storage Technologies (FAST)*, 2002.

[15] Qingbo Zhu Francis, Francis M. David, Christo F. Devaraj, Zhenmin Li, Yuanyuan Zhou, and Pei Cao. Reducing energy consumption of disk storage using power-aware cache management. In *Symposium on High-Performance Computer Architecture (HPCA), Febuary*, 2004.

[16] Lakshmi Ganesh, Hakim Weatherspoon, Mahesh Balakrishnan, and Ken Birman. Optimizing power consumption in large scale storage systems. In *HotOS*, 2007.

[17] Albert G. Greenberg, James R. Hamilton, David A. Maltz, and Parveen Patel. The cost of a cloud: research problems in data center networks. *Computer Communication Review*, 2009.

[18] Sudhanva Gurumurthi, Anand Sivasubramaniam, Mahmut Kandemir, and Hubertus Franke. DRPM: Dynamic Speed Control for Power Management in Server Class Disks. In *International Symposium on Computing Architecture*, 2003.

[19] Elliot Jaffe and Scott Kirkpatrick. Architecture of the Internet Archive. In *Israeli Experimental Systems Conference (SYSTOR)*, 2009.

[20] Brewster Kahle. Project Greenbox, January 2008. http://backyardfamilyfarm.wikispaces.com/Project+Greenbox.

[21] Peter Lyman, Hal Varian, Peter Charles, Nathan Good, Laheem Jordan, and Joyojeet Pal. *How Much Information? Executive Summary*. School of Information Management and Systems, UC-Berkeley, 2003.

[22] Cade Metz. Sun packs 150 billion web pages into meat locker. March 2009. http://www.theregister.co.uk/2009/03/25/new_internet_archive_data_center/.

[23] Rich Miller. How A Good PUE Can Save 10 MegaWatts. *Data Center Knowledge*, September 13 2010. http://www.datacenterknowledge.com/archives/2010/09/13/how-a-good-pue-can-save-10-megawatts/.

[24] Dushyanth Narayanan and Austin Donnelly. Write off-loading: Practical power management for enterprise storage. In *File and Storage Systems (FAST)*, 2008.

[25] Eduardo Pinheiro and Ricardo Bianchini. Energy conservation techniques for disk array-based servers. In *International Conference on Supercomputing (ICS)*, 2004.

[26] Akshat Verma, Ricardo Koller, Luis Useche, and Raju Rangaswami. Energy proportional storage using dynamic consolidation. In *File and Storage Systems*, 2010.

[27] Charles Weddle, Mathew Oldham, Jin Qian, An-I Andy Wang, Peter Reiher, and Geoff Kuenning. PARAID: A Gear-Shifting Power-Aware RAID. In *File And Storage Technologies (FAST)*, 2007.

[28] John Wilkes, Richard Golding, Carl Staelin, and Tim Sullivan. The HP AutoRAID heirarchical storage system. In *ACM Transactions on Computer Systems (TOCS)*, 1996.

[29] Emma Woollacott. Digital content doubles every 18 months. *TG Daily*, May 19 2009. http://www.tgdaily.com/hardware-features/42499-digital-content-doubles-every-18-months.

[30] Qingbo Zhu, Zhifeng Chen, Lin Tan, and Yuanyuan Zhou. Hibernator: helping disk arrays sleep through the winter. In *Symposium on Operating Systems Principles (SOSP)*, 2005.