GreenHDFS: Towards An Energy-Conserving, Storage-Efficient, Hybrid Hadoop Compute Cluster

Rini T. Kaushik
The University of Illinois, Urbana-Champaign and Yahoo! Inc.
kaushik1@illinois.edu

Milind Bhandarkar
Yahoo Inc.
milindb@yahoo-inc.com

ABSTRACT

Hadoop Distributed File System (HDFS) presents unique challenges to the existing energy-conservation techniques and makes it hard to scale-down servers. We propose an energy-conserving, hybrid, logical multi-zoned variant of HDFS for managing data-processing intensive, commodity Hadoop cluster. Green HDFS's data-classification-driven data placement allows scale-down by guaranteeing substantially long periods (several days) of idleness in a subset of servers in the datacenter designated as the *Cold* Zone. These servers are then transitioned to high-energy-saving, inactive power modes. This is done without impacting the performance of the *Hot* zone as studies have shown that the servers in the data-intensive compute clusters are under-utilized and, hence, opportunities exist for better consolidation of the workload on the *Hot* Zone. Analysis of the traces of a Yahoo! Hadoop cluster showed significant heterogeneity in the data's access patterns which can be used to guide energy-aware data placement policies. The trace-driven simulation results with three-month-long real-life HDFS traces from a Hadoop cluster at Yahoo! show a 26% energy consumption reduction by doing only *Cold* zone power management. Analytical cost model projects savings of $14.6 million in 3-year total cost of ownership (TCO) and simulation results extrapolate savings of $2.4 million annually when GreenHDFS technique is applied across all Hadoop clusters (amounting to 38000 servers) at Yahoo.

1. INTRODUCTION

Cloud computing is gaining rapid popularity. Data-intensive computing needs include advertising optimizations, user-interest predictions, mail anti-spam, data analytics and deriving search rankings. With the increase in the sheer volume of the data that needs to be processed, storage and server demands of computing workloads are on a rapid increase. Yahoo!'s datacenters already have 170 petabytes of data and deploy 38000 servers [4]. Over the lifetime of IT equipment, the operating energy cost is comparable to the initial equipment acquisition cost [9] and constitutes a significant part of the TCO of a datacenter [7]. Hence, energy-conservation of the extremely large-scale, commodity server farms has become a priority.

Server energy consumption costs can be cut down significantly by using low-power, high-energy-saving inactive power modes during idle periods of utilization[1]. However, inactive power modes cannot be used in an adhoc fashion as there are significant latencies associated with the power state transitions. Effective usage of inactive power modes mandates presence of significantly long periods of idleness in the system.

There is significant amount of research literature about datacenter energy management. A large number of these techniques aim to classify and place the computational load in an energy-efficient manner [6, 14, 16, 17, 19]. They try to scale-down servers by manufacturing idleness by migrating workloads and their corresponding state to fewer machines during periods of low activity. This can be relatively easy to accomplish when using simple data models, when servers are mostly stateless (e.g., serving data that resides on a shared NAS or SAN storage system). However, in complex data distribution models that have significant state, such techniques cannot manufacture enough idleness to make usage of inactive power modes feasible. One such example is the Hadoop compute cluster.

Given the massive bandwidth requirements and the sheer amount of the data that needs to be processed, data-intensive compute clusters such as those running Hadoop have moved away from NAS/SAN model to completely clustered, commodity storage that allows direct access path between the storage servers and the clients [10]. Hadoop's data-intensive computing framework is built on a large-scale, highly resilient Hadoop distributed filesystem (HDFS) managed cluster-based storage [13]). HDFS distributes data chunks and replicas across the servers for higher performance, load-balancing and resiliency. With data distributed across all servers, any server may be participating in the reading, writing, or computation of a data-block at any time. Such a data placement *complicates* power-management and makes it hard to generate significant periods of idleness in the Hadoop clusters and renders usage of inactive power modes infeasible [15].

Recent research on increasing energy-efficiency in GFS and HDFS managed clusters [2, 15] propose maintaining a primary replica of the data on a small covering subset of nodes that are guaranteed to be on and which represent the lowest power setting. The remaining replicas are stored in larger set of secondary nodes. Performance is scaled up by increasing the number of secondary nodes. However, these solutions suffer from degraded write-performance and increased DFS code complexity [3]. These solutions also do not do any data differentiation and treat all the data in the system alike.

We take a different approach to generating significant periods of idleness in a Hadoop cluster. Instead of energy-efficient placement of computations or replicas, we use *energy-*

[1]Some SATA disks consume only 0.9W in sleep state vs. 7.5W in idle state and 17W in active state

aware placement of *data* and focus on *data-classification* techniques to differentiate the data. A study of Yahoo!'s Hadoop cluster illustrated in Section 2 shows significant variation in the access patterns of the data stored in the cluster. We seek to utilize the heterogeneity in the data towards our energy-conserving data-classification-driven data placement. We proposed this approach in the context of cloud storage in our paper [12].

We propose GreenHDFS, an energy-conserving, self-adaptive, hybrid, logical multi-zone variant of HDFS. GreenHDFS trades performance and *power* by logically separating the Hadoop cluster into *Hot* and *Cold* zones. Zone's temperature is defined by its power consumption and the performance requirements. Data's temperature evolves based upon its availability/performance requirements. We use data-clas-sification policies to place data onto a suitable *temperature* zone. Since computations exhibit high data locality in the Hadoop framework, the computations flow naturally to the data in the right temperature zones.

GreenHDFS technique results in a number of servers in the *Cold* zone with very *low* utilization and guaranteed periods of idleness. The CPU, DRAM and Disks on these servers can then be transitioned to *inactive power* modes resulting in substantial energy savings. Thus, GreenHDFS provides a mechanism to have an energy-proportional behavior in data centers built with non-energy-proportional components.

We argue that zoning in GreenHDFS will not affect the *Hot* zone's performance adversely and the computational workload can be consolidated on the servers in the *Hot* zone without exceeding the CPU utilization above the provisioning guidelines. A study of 5000 Google compute servers, showed that most of the time is spent within the 10% - 50% CPU utilization range [5]. Hence, significant opportunities exist in workload consolidation. And, the compute capacity of the *Cold* zone can always be harnessed under peak load scenarios as discussed in Section 3.4.

The remainder of the paper is structured as follows. In section 2, we describe the data characteristics of a Yahoo! cluster. In Section 3, we discuss the design, and the conceptual basis of the policies in GreenHDFS. We focus on the major design primitive: an adaptive multi-zoned Hadoop cluster. In Section 4, we include simulation results demonstrating the effectiveness of our design in conserving energy in an enterprise Hadoop cluster at Yahoo!. Finally, we conclude.

2. ANALYSIS OF A YAHOO CLUSTER

We analyzed three months of HDFS logs [2] in one of Yahoo!'s enterprise Hadoop clusters. The cluster had 2600 servers and hosted 34 million files, and the data set size was 5 Petabytes. We found existence of a significant amount of data in the system which either wasn't accessed at all or rarely accessed after some amount of elapsed time. We introduced four file lifespan metrics to analyze the evolution and lifetime of the files in the cluster. $FileLifeSpan_{CFR}$ metric is defined as the file lifespan between the file creation and first file read access. This metric is used to understand the gap between a file's creation and the start of a file's hotness lifespan. $FileLifeSpan_{CLR}$ metric is defined as the file lifespan between the file creation and the last file read access. This metric is used to determine the *Hotness* profile of a file, i.e., the period of the time for which the file is actively accessed. $FileLifeSpan_{LRD}$ metric is defined as the file lifespan between last file read access and file deletion. $FileLifeSpan_{LRD}$ metric helps in determining the *Coldness* profile of a file, i.e., the period for which a file lies dormant in the system without getting accessed.

As shown in Figure 1, the $FileLifeSpan_{CFR}$ of 90.26% of data is less than 2 days. Thus, data is accessed soon after its creation. 89.61% of data has a $FileLifeSpan_{CLR}$ of less than 10 days. This indicates that majority of the data is *hot* for less than 10 days after its creation in the system. This clearly illustrates a news-server-like behavior for 89% of the data whereby the reads to the data are clustered around its creation and accesses die down after a short *hot* lifespan. 40% of the data in the cluster has a $FileLifeSpan_{LRD}$ of higher than 20 days. This indicates that 40% of the data lies untouched in a dormant state in the cluster for more than 20 days. We didn't account for temporary data that is stored in the grid in this analysis. Had we accounted for the same, a higher percentage (60%) would be dormant for more than 20 days.

As shown in Figure 2(mid), in steady-state, almost 60% data was not accessed at all in a 20 day window. This cold data amounted to 70% of the files in the system as shown in the Figure 2(left). This cold data needs to exist in the system for regulatory compliance and historical trend analysis. This study indicates that there are tremendous opportunities to differentiate the data into different temperature classes in a Hadoop computer cluster.

3. GREENHDFS'S HYBRID MULTI-ZONE LAYOUT

We give an overview of GreenHDFS's zones in this section and we only discuss the functionality and policies which are relevant to Hadoop cluster's energy-management. The zone with the higher *temperature* is designated as the *Hot* zone. The zone with the lower *temperature* is designated as the *Cold* zone. High-level data placement policy decides the zone on which data will be placed initially based on per-zone data-classification policies illustrated below. The data transitions adaptively between the zones as its *temperature* changes in response to the energy-management policies covered in Section 3.3. GreenHDFS requires few changes to the HDFS code such as differentiating inactive servers from failed servers in the heartbeat handling mechanism [3] and directing data accesses to the right zone. We have built a prototype which is out of the scope of this paper.

3.1 *Hot* Zone

Data Class: Consists of hot, popular data that is accessed very frequently. The popularity can be spatial or temporal. **Hardware Class:** consists of high performance, high power and hence higher cost CPUs. **Data Chunking Policy:** Uses a *Chunk Server Placement* policy that considers the problem of assigning n chunks $f_1, f_2, \ldots, f_n$ among m servers and aims to optimize the mean response time and the system throughput by minimizing the queuing delays on

[2] HDFS has the ability to log all filesystem access requests. The logging is implemented using log4j and once enabled, logs every HDFS event in the Namenode's log [20]. We used the HDFS metadata checkpoint and logs for our analysis.

[3] This is necessary as HDFS rereplicates the blocks residing on a server if it doesn't receive a heartbeat from the server within a configured threshold of time.

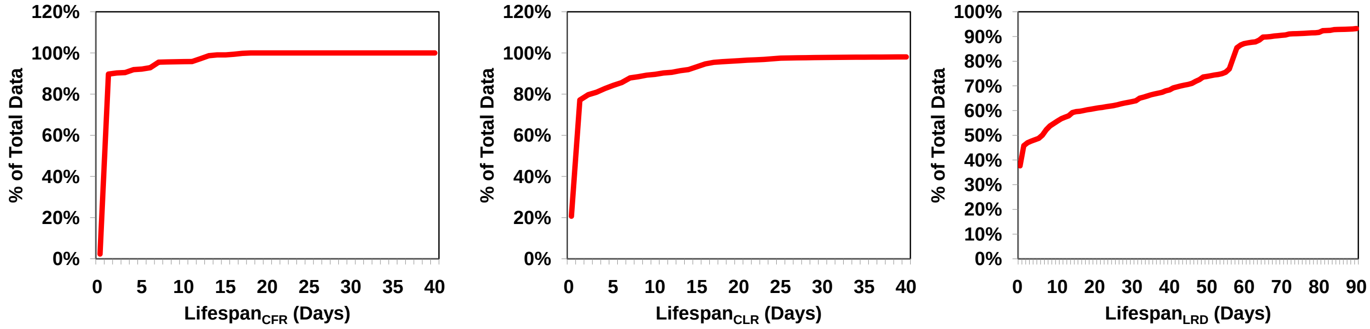

Figure 1: Cumulative Frequency Distribution of the LifeSpan metrics in the cluster. 90.26% of data has a $FileLifeSpan_{CFR}$ of less than 2 days. 89.61% of the data has a $FileLifeSpan_{CLR}$ of less than 10 days. 40% of the data has a $FileLifeSpan_{LRD}$ higher than 20 days.

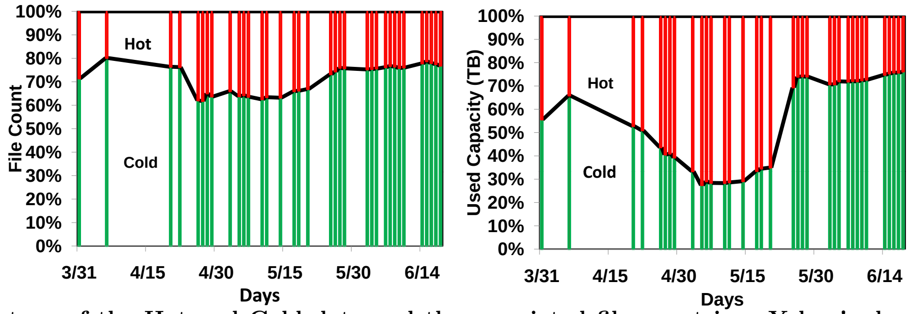

Figure 2: Percentage of the Hot and Cold data and the associated file count in a Yahoo! cluster. In steady-state, 60% data is Cold in the system. Cold data was determined as the data that was not accessed within the last 20 days in this graph. The anamoly during the periods of 04/30-05/15 arose from a planned internal enterprise initiative that resulted in unprecedented data accesses.

the servers' disks in *Hot* zone. We assume $m \geq n$. Full description of the algorithm is beyond the scope of this paper. **Power Policy:** None, *Hot* zone has strict service level agreement (SLA) requirements and hence, performance is of the greatest importance. We trade-off power savings in interest of higher performance and servers in *Hot* zone will remain in the active mode at all times. **Zone Server Assignment:** Majority (70% +) of the servers in the cluster are assigned to the hot zone upfront. We are working on a dynamic server assignment policy which is self-adapts to the changes in the workload patterns similar to [8].

3.2 *Cold* zone

Data Class: consists of files with low spatial or temporal popularity with few to rare accesses. We trade-off performance for higher energy-conservation in this zone. **Hardware Class:** We propose using a larger number of disks per server in these zones compared to the *Hot* zone to accommodate the huge amount of cold data. This has two advantages: 1) fewer servers will need to be assigned to the *Cold* zone, hence, performance-critical *Hot* zones will get a higher share of servers. Higher number of hot servers is critical in a high-performance and high-availability distributed systems software which tends to spread data and computation. **Power Policy:** *Aggressive*, Performance and SLA requirements are not critical for *Cold* zone and we employ *aggressive* power management schemes and policies in *Cold* zone to transition servers to a very low power consuming, inactive power mode. **Zone Server Assignment:** *On-Demand* servers are powered-on and assigned to *Cold* zone on-demand. **Data Chunking Policy:** *None*, For optimal energy savings, it is important to increase the idle times of the servers and limit the wakeups of servers that have transitioned to the inactive-power-saving mode. Keeping this rationale in mind and recognizing the low performance needs and infrequency of data accesses to the *Cold* zone; this zone *will not chunk* the data. By not chunking the data, we ensure that future access to a particular data is limited to just one server hosting that data. **File Allocation Policy** tries to avoid powering-on a server and maximizes the use of the existing powered-on servers in its *server allocation* decisions in interest of maximizing the energy savings. We used *In-order placement* policy where a data structure maintains a sorted list of all the server IDs and the first few servers in the data structure are chosen as a target for data placement. These servers are kept powered-on and are filled completely to capacity before the next set of servers is chosen from the list. **Data Integrity** To ensure *data integrity* in the *Cold* zone, disks in the *Cold* zone are scrubbed from time to time. Every block on the disk is read and checked for agreement with its signature. [4]

3.3 Energy-management Policies

Files are moved from the *Hot* zone to the *Cold* zone as their temperature changes over time. In this paper, we use *age* of a file, as defined by the last access to the file, as the

[4] We have added 1 day per month per server to the servers in the *Cold* zone, to account for a monthly scrub in our evaluation.

measure of temperature of the file. *File Migration Policy* monitors the *age* of the files and moves old files to the *Cold* Zone. The advantages of this policy are two-fold: 1) it leads to higher *space-efficiency* as space is freed up on the *Hot* zone for files which have higher SLA requirements by moving rarely accessed files out of the servers in these zones; furthermore, these files can be aggressively compressed on the lower zone to save on storage capacity2) it leads to higher *energy-efficiency* as the concentration of the *cold* data on these servers allows for aggressive power management techniques illustrated earlier. *Server Power Conservation Policy* determines the servers which can be transitioned into a power saving standby/sleep mode in the *Cold* zone. We rely on Wake-on-LAN to wake the system upon arrival of a packet. Servers transition back to active power state upon receipt of data access, data migration or disk scrubbing events. All the components in the servers typically illustrate poor energy-proportionality and servers consume 50% peak power even when idle [5]. Thus, it is not sufficient to just scale-down one component in the system and we scale-down all the components (CPU, DRAM, Disks) in the system. *File Reversal Policy:* ensures that the QoS, bandwidth and response time of files which become popular again after a period of dormancy is not impacted. A once-again-popular file is moved back from the *Cold* zone to the *Hot* zone if the number of accesses to the file exceed a certain threshold.

3.4 Discussion

Based on our observation that a large amount of data in the cluster is cold, GreenHDFS will move this data to the *Cold* zone. The observation that the accesses to the data have a news-server-like access pattern, ensures that once a data is deemed cold (i.e., was not accessed in past n days), the probability of it being accessed again is lower. This will guarantee significant periods of idleness in the *Cold* zone and allow a large number of servers in the *Cold* zone to transition to inactive power modes. Hence, there can be a significant reduction in the energy costs of the cluster.

Given that the servers are under-utilized, the workload can be consolidated in the *Hot* zone servers. Furthermore, the boundary between the *Hot* and *Cold* zone is a logical boundary and in periods of peak utilization, the compute power of the *Cold* zone servers can still be harnessed. We also argue that the resulting power reduction courtesy of a percentage of always-sleeping servers could allow for provisioning more servers in the data center than a baseline case within the same power budget. [11] quotes in favor of our argument that - 'Maximizing usage of the available power budget is also important for existing facilities because it can let the computing infrastructure grow or enable upgrades without requiring the acquisition of new datacenter capacity, which can take years if it involves new construction'. These additional servers can be provisioned to the *Hot* zone, allowing the *Hot* zone to offer similar availability, performance and bandwidth as the baseline case.

Finally, Oozie, which is used to manage and schedule Hadoop workflows, can be used to proactively power-on inactive servers since it knows upfront when/where a job needs to be launched [1]. Such proactive powering-on will help hide and amortize the wakeup latency of the power state transition. Also, given the batch-processing mode of the workloads, the wakeup penalty is of less concern. Still, we are taking several steps to mitigate the performance penalty of state transitions in our ongoing research.

4. EVALUATION RESULTS

We used a trace-driven simulator [5] to evaluate GreenHDFS and used HDFS traces generated by a Hadoop cluster at Yahoo! as an input to the simulator. We used models for the power levels of the disk, processor and the DRAM in the simulator [6]. Table 1 lists the power values used in the Simulator. There were 34 million files in the trace file and the total size of the dataset was 5 PetaBytes.

All experiments and analysis were performed on the nodes of a development cluster at Yahoo. We used PIG extensively for the data analysis of the traces. We used a hybrid-model with storage-heavy servers (12, 1TB disks/server) in the *Cold* zone and normal (4, 1TB disks/server) servers in the *Hot* zone. Our comparisons were done with the baseline HDFS which doesn't do any energy-management. The Simulator assumed 2600 servers to be consistent with the Hadoop cluster under consideration. For performance, load balancing and availability reasons, we provisioned 70% (i.e., 1820) servers to the *Hot* zone and 30% (i.e., 780 servers) to the *Cold* zone in the simulator.

Server	Active Power (W)	Idle Power (W)	Sleep Power (W)
Hot server (2 CPU, 8 DRAM DIMM, 4 HDD)	442.7	105.3	14.1
Cold server (2 CPU, 8 DRAM DIMM, 12 HDD)	578.7	165.3	21.3

Table 1: Power Number Used in Simulator

Energy-Conservation In this section, we show the energy savings made possible by GreenHDFS over three months in comparison to the baseline HDFS. The cost of electricity was assumed to be $0.063/Kwh [11]. Applying GreenHDFS to 2600 servers results in $41,607 energy savings in a three-month period which is 26% of the baseline energy costs. If GreenHDFS technique is applied to all of the Yahoo! Hadoop clusters (amounting to 38000 servers), $2,432,417 can be saved in the energy costs annually. And, these results are with power management only in *Cold* zone. Energy saving from off-power servers will be further compounded in the cooling system of a real datacenter. For every Watt of power consumed by the compute infrastructure, a modern data center expends another one-half to one Watt to power the cooling infrastructure [18].

Cold Zone Reliability Consideration We analyzed the frequency distribution of the power transitions incurred by the servers in the *Cold* zones. As shown in Figure 3, the maximum number of power state transitions incurred by a server in a three-month simulation run was 70 times. Frequently starting and stopping disks is suspected to affect disk longevity. Given the very small number of transitions incurred by a server in the *Cold* zone in a year, there is no risk of exceeding the start/stop cycles [7] during the typical 5 year service life time of the disks.

[5] The GreenHDFS simulator was implemented in Java and MySQL distribution 5.1.41 and executed using Java 2 SDK, version 1.6.0-17.

[6] Information was extracted from the datasheet of Seagate Barracuda 7200.7 which is a SATA hard drive, and a Quad core Intel Xeon X5400 processor. This is an approximation of the cluster hardware.

[7] Most of the SATA disks can tolerate 50,000 start/stop cycles

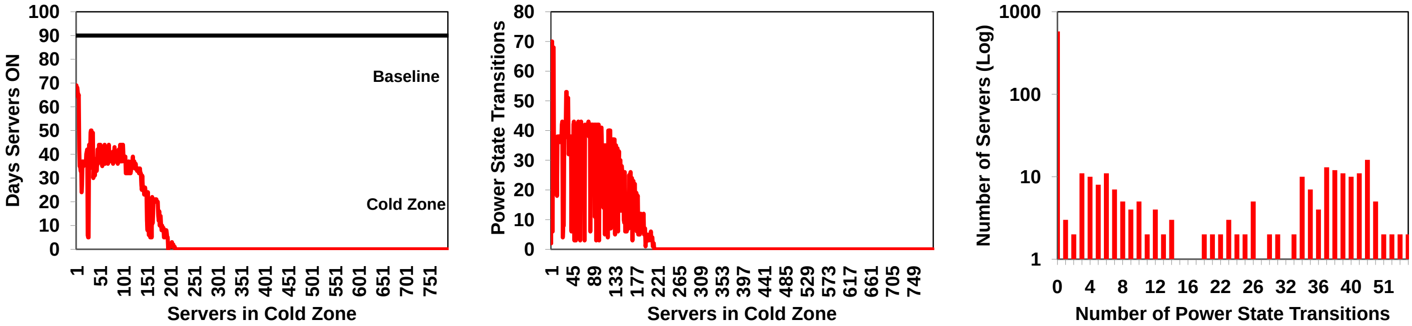

Figure 3: Days servers in *Cold* Zone were ON compared to the Baseline and the number of power state transitions incurred by the servers in the *Cold* Zone.

Analytical Cost Model Projections We changed the homogeneous Analytical Cost Model used in [11] to allow heterogeneity [8] in the power consumption rate and in the costs of the servers in the hybrid model. In the hybrid model with 4, 1TB disks server (lower cost) in the *Hot* zone and 12, 1TB disks server (higher cost) in the *Cold* zone, the 3-year TCO with no energy management is $15.1million for 2600 servers. In our simulation run, on average only 198 servers were awake in the *Cold* zone out of the assigned 780 servers. Taking the sleeping servers into consideration, the Analytical Cost Model shows savings of $1 million in the 3-yr TCO for 2600 servers in the hybrid model while giving an extra 6.8 Petabytes of storage. 3-yr TCO saving will amount to $14.6 million by applying GreenHDFS technique across all the hadoop clusters (around 38000 servers overall),

5. CONCLUSION AND FUTURE WORK

We presented GreenHDFS, an energy-conserving, hybrid, logical multi-zoned variant of Hadoop's compute cluster. We rely on *data classification* driven data placement to realize guaranteed, substantially long periods of idleness in a subset of servers designated as the *Cold* zone in the Hadoop cluster. These long periods of idleness allow us to use aggressive inactive power modes in all components (CPU, disks, DRAM) of this subset of servers projecting significant energy cost savings. We argue that the multi-zone data center layout won't have an adverse impact on the performance of the *Hot* zone as there are ample opportunities for workload consolidation given the low utilization (10-50%) in compute servers [5]. Simulation results show that GreenHDFS is capable of achieving 26% savings in the energy costs of a Hadoop cluster in a three-month simulation run. Analytical cost model projects a savings of $14.6 million in 3-year TCO and simulation results extrapolate a savings of $2.4 million annually when GreenHDFS technique is applied across all Hadoop clusters (amounting to 38000 servers) at Yahoo. We are working on optimizations to enhance the energy-efficiency by using low-cost processors, amortizing the power transition penalties, reducing replication by exploring RAID alternatives and compression mechanisms to make the *Cold* zone more energy- and storage-efficient.

6. ACKNOWLEDGEMENT

We are thankful to Ryota Egashira and Viraj Bhat at Yahoo! for giving us a jump-start on the cluster analysis. We are thankful to Raj Merchia at Yahoo! Inc and the anonymous reviewers for their comments. Lastly, we are grateful to Prof. Klara Nahrstedt, and Prof. Roy Campbell for their feedback in the initial version of this work.

[8]The spreadsheets used in these calculations are located at the author's website and can be tweaked for individual clusters

7. REFERENCES

[1] http://yahoo.github.com/oozie/.

[2] H. Amur, J. Cipar, V. Gupta, G. R. Ganger, M. A. Kozuch, and K. Schwan. Robust and flexible power-proportional storage. In *SoCC*, 2010.

[3] H. Amur and K. Schwan. Achieving power-efficiency in clusters without distributed file system complexity. 2010.

[4] E. Baldeschwieler. http://developer.yahoo.com/events/hadoopsummit2010/agenda.html.

[5] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *Computer*, 40(12), 2007.

[6] C. Bash and G. Forman. Cool job allocation: measuring the power savings of placing jobs at cooling-efficient locations in the data center. In *ATC'07*.

[7] C. Belady. *In the data center, power and cooling costs more than the it equipment it supports.* Electronics Cooling, February, 2010.

[8] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *NSDI*, 2008.

[9] Y. Chen, A. Ganapathi, A. Fox, R. H. Katz, and D. A. Patterson. Statistical workloads for energy efficient mapreduce. Technical report, UC, Berkeley, 2010.

[10] S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. *SIGOPS Oper. Syst. Rev.*, 37(5):29–43, 2003.

[11] U. Hoelzle and L. Barroso. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines.* Morgan and Claypool Publishers, May 29, 2009.

[12] R. T. Kaushik, L. Cherkasova, R. Campbell, and K. Nahrstedt. Lightning: Self-adaptive, energy-conserving, multi-zoned, commodity green cloud storage system. *HPDC*, 2010.

[13] S. Konstantin, H. Kuang, S. Radia, and R. Chansler. The hadoop distributed file system. *MSST*, 2010.

[14] K. Le, R. Bianchini, M. Martonosi, and T. Nguyen. Cost- and energy-aware load distribution across data centers. In *HotPower*, 2009.

[15] J. Leverich and C. Kozyrakis. On the energy (in)efficiency of hadoop clusters. *SIGOPS Operating Systems Review*, 2010.

[16] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making scheduling "cool": temperature-aware workload placement in data centers. In *ATC '05*.

[17] D. Narayanan, A. Donnelly, and A. Rowstron. Write off-loading: Practical power management for enterprise storage. *Trans. Storage*, 4(3):1–23, 2008.

[18] C. Patel, E. Bash, R. Sharma, and M. Beitelmal. Smart cooling of data centers. In *IPACK*, 2003.

[19] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase. Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing*, 9:42–49, 2005.

[20] T. White. *Hadoop: The Definitive Guide.* O'Reilly Media, May, 2009.