Lightning: Self-Adaptive, Energy-Conserving, Multi-Zoned, Commodity Green Cloud Storage System

Rini Kaushik, Ludmila Cherkasova*, Roy Campbell, Klara Nahrstedt
University of Illinois at Urbana-Champaign

ABSTRACT

The objective of this research is to present an energy-conserving, self-adaptive Commodity Green Cloud Storage, called Lightning. Lightning's File System dynamically configures the servers in the Cloud Storage into logical *Hot* and *Cold* Zones. Lightning uses data-classification driven data placement to realize guaranteed, substantially long, periods (several days) of idleness in a significant subset of servers designated as the *Cold* Zone, in the commodity datacenter backing the Cloud Storage. These servers are then transitioned to inactive power modes and the resulting energy savings substantially reduce the operating costs of the datacenter. Furthermore, the energy savings allow Lightning to improve the data access performance by incorporation of high-performance, though high-cost Solid State Drives (SSD) without exceeding the total cost of ownership (TCO) of the datacenter. Analytical cost model analysis of Lightning suggests savings in the upwards of $24 million in the TCO of a 20,000 server datacenter. The simulation results show that Lightning can achieve 46% energy costs reduction even when the datacenter is at 80% capacity utilization.

1. INTRODUCTION

The Internet Era of computing is upon us and it brings with it a new data model which is much more global in nature. Data is shared and used across continents and ubiquitously accessed via the internet. The astronomical growth of data requires highly scalable storage technologies, and *storage-efficiency* and *energy-efficiency* have become an urgent priority. IDC and the Environmental Protection Agency indicate that *'Digital data will surpass 1.8 zettabytes by 2011'* and *'U.S. data centers will consume 100 Billion Kilowatt hours annually by 2011'*. 100 Billion Kilowatt hours will cost an upwards of $10B annually at a common price of $100 per MWh.

Cloud storage [3] has an on-demand, pay-per-usage business model. It is elastic in nature and offers instant scalable capacity, reduced management overhead, and involves no capital and facilities costs. Hence, Cloud storage is gaining in popularity as the storage technology of choice.

Economies of scale are possible in Cloud Storage through multi-tenancy; shared infrastructure allows multiple customers to share costs of infrastructure with capabilities they don't possess themselves. As per [2], Cloud Storage will be used to store a multitude of data types: backup, archives, media, websites, photos, videos and disaster recovery data. These data have varied access patterns, and data allocation needs.

There is a large body of related work in the area of datacenter energy management. Several techniques [4, 5, 9, 11] aim to classify and place workload (e.g., computation) in an energy-efficient manner. Existing energy management techniques, achie-ve only short sub-second periods of idleness in servers in a typical distributed system where data striping and load balancing are a norm. Since, the penalty associated with a power state transition from an inactive mode back to an active mode is high; these systems are unable to utilize high power saving, inactive modes [7].

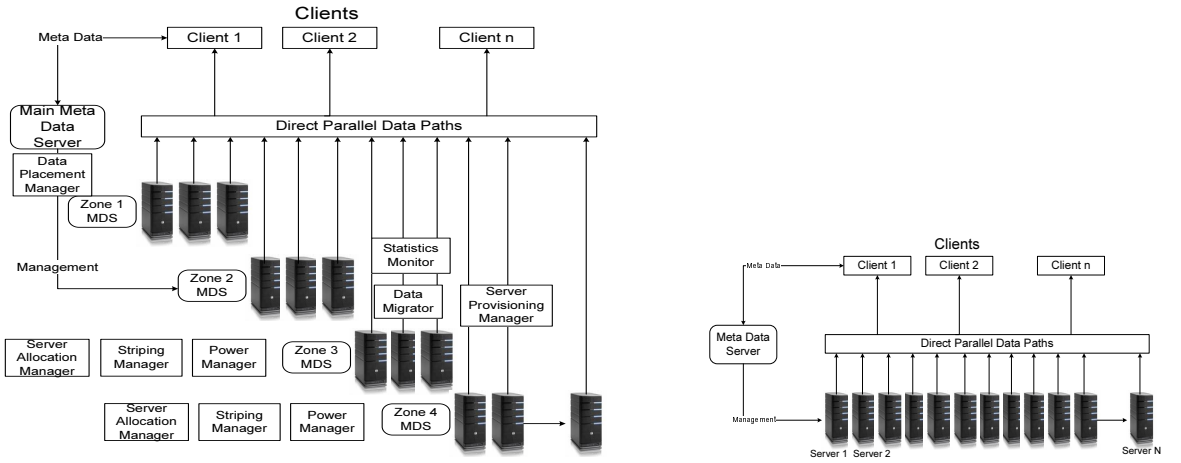
We introduce a new approach to solving the energy-efficiency challenge. Instead of energy-efficient placement of workloads, we propose energy-conserving placement of *data* and focus on proactive as well as reactive *data-classification* techniques to extract energy savings. We exploit the heterogeneity inherent in the data in the Cloud Storage in our data-classification techniques.

In this paper, we present a design and architecture of a Commodity, Green Cloud Storage system called *Lightning*. Lightning uses a filesystem-managed, logical, *power* differentiated, multi-zoned layering of the storage servers as shown in Figure 1(a) to save on energy costs. Commodity distributed object-based file systems [6] do not do any energy management.

The novelty and contributions of Lightning are in achieving: (a)**High Energy-Conservation**, we use *data-classification* policies, application hints and filesystem derived insights to place files with similar measures of *Coldness* onto a zone of servers designated as the *Cold* Zone. Long periods of idleness are inherent in these *Cold* Zones and the entire server (processors, DRAM, disks) can be transitioned to an inactive power mode for maximal energy savings. The resulting reduction in the power usage can have a large strategic impact as it will allow an existing datacenter to accommodate the business growth within a given power budget. Additional servers can be deployed in the datacenter; thus obliterating the need to build a new datacenter. TCO

*Hewlett-Packard Laboratories

HPDC '10, June 20-25, 2010, Chicago, Illinois, USA



(a) Lightning (b) Typical Internet-Scale File System

Figure 1: Comparison of Lightning's File System Managed Logical, Multi-Zone View of a Commodity Datacenter with the Single-Zone view of a Typical Internet-Scale Commodity Datacenter File System

of the datacenter consists of Capex(capital expenses) and Opex(operating expenses). Energy costs are a significant portion of the operating costs (Opex) of a datacenter [7]; hence, reduction in energy costs will also considerably reduce the TCO of the datacenter. (b)**Higher Response Time Performance**, Reduced TCO will allow incorporation of high performance though high cost Solid-State Devices (SSD) [8]. The analytical cost model discussed briefly in (Section 2) addresses the optimal number of SSDs that can be incorporated without impacting the TCO of the datacenter.

The paper is structured as follows: Section 2 introduces Lightning's design principles and Section 3 presents its architectural components. Section 4 presents preliminary results and validates some of the architectural design decisions. Section 5 concludes the paper.

2. ARCHITECTURE

Lightning aims to deliver an highly Energy-Conserving, Multi-Zoned, Automated Policy-Driven, and Self-Adaptive Commodity Cloud Storage System, managed by a Cloud File System. There are two major design principles to Lightning. **Adaptive Multi-zoned Data Center Layout:** Instead of organizing the servers in the data warehouses in a single zone as it is done in the current cloud storage solutions such as Amazon S3, the servers get organized in multiple dynamically provisioned logical zones. Each zone has distinct performance, cost, data layout and ***power*** characteristic and is managed by a set of policies most conducive to the data residing in the zone as shown in Table 1. The paper assumes four zones denoted as *Zone 1, Zone 2, Zone 3 and Zone 4*; however, any number of zones can be chosen. *Zone 1, Zone 2* are *Hot* zones and *Zone 3, Zone 4* are *Cold* zones. **Hint-based, Policy-Driven File System:** A hints-based, and policy-driven file system manages the data placement and data movement between multiple logical zones. The filesystem determines the degree of the *Coldness* of a file in two ways: a)*proactively* based on the hints received from the application; b)*reactively* based on the insights derived by the filesystem. Data is moved between the zones in response to the changes in the data's *Coldness* over time.

The metadata store at the top level simply contains an indirection to the metadata store of the new zone where the file is residing. File system-based data management includes semantic information such as metadata and hints about the files which proves valuable in aiding the data placement in a zone-based system.

The Lightning architecture consists of different management components, services, algorithms, and protocols to execute required Lightning's file system and storage operations. Below, we briefly discuss some of the major components and their functional capabilities.

Data Placement Manager: Any *file create* request gets directed to the Data Placement Manager (DPM) component of the filesystem. DPM takes the file attributes, and any hints provided by the application as input and decides the zone on which the file will be placed initially. DPM consists of the following two data classification sub-components in its decision-making process: (1)**Hints Manager** takes hints such as the popularity of a video, rating of a movie or classification of file as backup/archival and proactively places the file on the most conducive zone. Hints implicitly yield (proactive) *power savings and performance guarantees* and reduce the need for reactive data migration. Frequently accessed data such as high popularity and high rating videos are placed in *Hot* zones. Less frequently accessed data such as backups, archivals, disaster recovery and long-term retention data are placed in the aggressively power-managed *Cold* zones upfront. (2)**Insights Manager** derives insights about the file from its own observations and monitoring. Insights assist in an *optimal proactive placement* of the file in a zone. For example, the insight that a file is a highly accessed profile file (small, read-only file) allows the DPM to place the file into *Zone 1* consisting of SSDs which have a very good read access performance. We opt to place only small-sized and read-only data in *Zone 1* as SSDs have limited write cycles and storage capacity.

Striping Manager: Per-zone Striping Manager takes the file's zone allocation as an input and decides the striping strategy for the file, based on the striping policy of the zone. For optimal energy savings, it is important to increase the idle times of the servers and limit the wakeups of servers

Table 1: Description of the associated striping, power, reliability and data classification policies in the Logical Zones of Lightning

	Zone 1	Zone 2	Zone 3	Zone 4
Storage Type	SSD	SATA	SATA	SATA
Number of Disks		Few	Large (storage-heavy)	Large (storage-heavy)
File Striping Policy	None	Performance-Driven, Same as [6]	Energy-Efficiency Driven,None	Energy-Efficiency Driven, None
Server Power Policy	Always-on	Always-on	Aggressive, Server Standby, Wake-On-Lan	Aggressive, Server Sleep, Wake-On-Lan
Replication Policy	None	n-way	1-way	1-way
Data Classification	Small read-only files	Frequently accessed, Popular, High Rating Files	Rarely accessed, Less Popular, Low Rating Files	Backups, Archivals, Disaster Recovery, Long-term Retention Data
Power Transition Penalty	None	None	Medium	High
Energy Savings	Medium (SSD consume less power)	None	High	Highest
Performance	High	High	Low	Lowest

that have transitioned to the power-saving mode. Keeping this rationale in mind and recognizing the low performance needs and infrequency of data accesses to the *Zone 3* and *Zone 4*; these zones *will not stripe* the data. This will ensure that upon a future access, only the server containing the data will be woken up. However, given that reliability is becoming exceedingly important with the increasing disk capacities, 1-way replica of the files will be maintained in the *Zone 3*. Given, the small sizes of the files in *Zone 1*, striping is *unnecessary* in the first place. Hence, at this point, only *Zone 2* has a *striping policy* similar to the striping policy used in [6].

Server Allocation Manager: Server Allocation Manager (SAM) in *Zone 1* and *Zone 2* aims to optimize data access performance; hence it determines the servers in the zones in which the file stripes and replicas are placed. SAM takes as an input the number of stripe units and replicas. It then relies on the *server allocation* algorithm to determine the optimal set of servers in the zone. This means that SAM focuses on maximizing the performance of the data access in a load- and capacity-balanced manner. In the *Zone 3* and *Zone 4*, SAMs are driven by a goal to *maximize the energy* savings by minimizing the number of days servers are in an active power mode. By default, the servers in *Zone 3* and *Zone 4* are in a sleeping mode. A server is woken up when either new data needs to be placed on it or when data already residing on the server is accessed. The server then stays on till the *Server Power Conserver* policy, described in 2, power cycles the server. SAM maximizes the use of already powered-on servers in its server allocation decisions to avoid waking up sleeping servers. We used an *In-order Placement Policy* which maintains a sorted list of the server IDs in increasing order, and the first server in the list is chosen as a target for data placement. This server is filled completely to its capacity before next server is chosen from the list. The first server in the list is left powered on till it is filled to capacity to minimize unnecessary power transitions.

Power Manager: PM is a per-zone entity which relies on per-zone power policy to determine the servers which can be transitioned into a power saving mode and to decide the appropriate power saving mode (idle/standby/sleep) of the server. The PM invokes a power saving policy, called *Server Power Conserver*, at a recurring time interval. *Zone 3* transitions to a standby power state that has a lower wakeup time than the sleep state. Thus it trades off better performance with slightly lower energy savings. *Zone 4* doesn't need to make the performance tradeoff and can transition to sleep mode leading to maximal power savings. *Zone 1* and *Zone 2* have strict performance requirements and needs to comply with SLAs (Service Level Agreement). Hence, a power saving scheme which doesn't compromise performance can be used in the future. We aim to use hardware techniques similar to [10] in our prototype to transition the processors, disks and the DRAM into low power state. We plan on using Wake-on-LAN to wake up sleeping servers upon a future data access. Thus, power transitions will be software-driven and will not require any manual intervention.

Data Migrator: DM is a cross-zone component. It is responsible for migrating the files between the zones as the *Coldness* of the data changes over time and it is policy-driven. We considered *Insight-driven* policies. An insight-driven policy is based on an observed pattern in the system. It is a reactive policy and has a recurring frequency. For example, the *File Migration Policy* monitors the last access-time of the files in the *Zone 1* and *Zone 2*, and moves the file to the *Zone 3* if the last access time is greater than the specified threshold. The advantages of this scheme are two-fold: (a) Space is freed up on the high-performance *Zone 1* and *Zone 2* for files which have higher SLA requirements; (b) The file movement leads to energy-efficiency as these files will be aggressively de-duplicated and compressed in the *Cold* zones and servers will be eventually transitioned to low power modes. A *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.

Server Provisioning Manager: (SPM) SPM allows self-management of the system by automatically rebalancing and optimizing resources without human intervention. A logical server allocation into zones would be deemed ineffective if the server assignment and allocation were static in nature. A static split of the servers into zones will not be responsive to the changes in the capacity, performance or power demands of the workloads observed by the system. SPM does server *re-assignment* and *re-configuration* into respective zones in two ways: **(1)Initial provisioning:** SPM is responsible for the *initial assignment* of servers to zones when the datacenter first comes into effect. The goal is to maximize the set of servers kept *powered-off* for optimal power savings initially.

The server **provisioning of the Zone 1** is done statically and it is driven by an analytical cost model. The

homogeneous analytical cost model used in [7] is changed in this paper to allow heterogenity in the power consumption rate and in the costs of the servers in the datacenter. Table 2 shows the net savings in the TCO if 25% servers are powered-off in a 20,000 node datacenter and varying percentages of SSD are incorporated in the system. The assumptions and spreadsheets used in this paper along with the formulas are available at the author's website [1] for reference. The server **provisioning of *Zone 2*** uses an algorithm to calculate the number of servers to provision initially. We will describe the algorithm in future work. Only two servers are provisioned to *Zone 3* and *Zone 4* initially since these zones do not need to meet any SLAs and hence, can tolerate the power-on penalty. The rest of the servers are put in the *excess* category and are kept powered-off. **(2) Dynamic re-provisioning**: During the runtime, the goal is to (a) *re-configure the server assignment*, and (b) *re-assign and power-on* servers between zones as the capacity, bandwidth, QOS (Quality of Service) demands of the workload change. The algorithm is beyond the scope of this paper.

Table 2: TCO savings with 25% Servers OFF

SSD %	Total TCO	TCO Savings
0%	$147,410,056	$(24,222,382)
20%	$155,708,703	$(15,923,735)
40%	$164,007,350	$(7,625,088)
60%	$172,305,997	$ 673,559

3. EVALUATION

Evaluation Methodology: We built a trace-driven simulator for Lightning and evaluated Lightning using media traces generated by Medisyn [12], representative of accesses observed by HPC media servers at HP. All experiments were performed on a HP laptop (2.4GHz Intel Core2 Duo) with 3GBytes RAM, running Windows Vista. The simulator software was implemented in Java and MySQL distribution 5.1.41 and executed using Java 2 SDK, version 1.6.0-17. The simulator contains the main energy-conservation related functionality illustrated in the earlier sections. The simulator used models for the power levels, power state transition penalties and performance from the datasheets of Seagate Barracuda 7200.7, a Quad core Intel Xeon X5400 processor and Intel X25-E 64GB SSD. The number of files in the trace were a million and the total size of the files was 290 Terabytes. The cost of electricity was assumed to be $0.063/KWh. We assumed that a server consumed 442.7W (active), 105.3W (idle), 14.1W (sleep). We kept track of the number of days spent in a particular power mode for each server in the datacenter to calculate the energy costs. In our experiments, we compared Lightning's energy costs over a baseline case (Datacenter without energy management). Preliminary results show (without using hints or insights) a 46% reduction in energy costs of a 500 server datacenter with 80% capacity utilization in a one year simulation run. Cost savings will be further compounded by reduction in the cooling costs of a real datacenter.

4. CONCLUSION

This paper introduces on-going research on a novel energy-saving Commodity Cloud Storage, called Lightning. Lightning yields 46% savings in energy costs as illustrated in the preliminary results. We have just touched the tip of the iceberg here. For our system to be practical, it is not enough

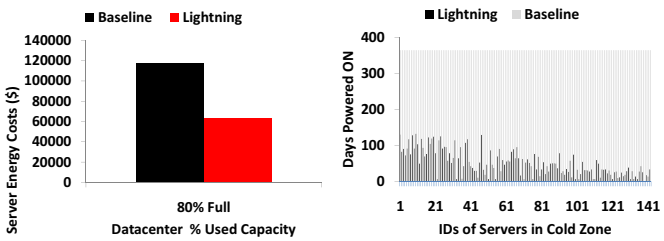


Figure 2: Energy Savings with Lightning and Days servers in *Cold* Zone were ON compared to the Baseline. Lightning achieves 46% savings in the energy costs simply by doing power management in the *Cold* Zones.

to just minimize the energy costs; we must also guarantee that performance, availability, reliability and integrity of the data is not impacted by our design. In the future, we will illustrate several techniques to mitigate the performance penalty of the power transitions and to limit the wakeups of the servers in the inactive power state in the *Cold* Zones. We will also argue that neither availability nor reliability is adversely impacted by our design. We will elaborate on the various algorithms, policies, associated thresholds and their sensitivity analysis. Finally, we are developing a prototype which utilizes hints and insights to do proactive, energy-efficient data placement to further strengthen our results. Part of this work was supported by NSF grant CNS 05-51665.

5. REFERENCES

[1] *www.cs.illinois.edu/homes/kaushik1/TCOspreadsheets.*
[2] *CloudStorageUseCasesv0.5.pdf.* SNIA, June, 2009.
[3] *ManagingDataPublicCloud.pdf.* SNIA, October, 2009.
[4] C. Bash and G. Forman. Cool job allocation: measuring the power savings of placing jobs at cooling-efficient locations in the data center. In *USENIX ATC*, 2007.
[5] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing energy and server resources in hosting centers. *SOSP*, 2001.
[6] S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. *SOSP*, 2003.
[7] U. Hoelzle and L. A. Barroso. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines.* Morgan and Claypool Publishers, May 29, 2009.
[8] W. Josephson, L. Bongo, D. Flynn, and K. Li. Dfs: A file system for virtualized flash storage. *FAST*, 2010.
[9] K. Le, R. Bianchini, M. Martonosi, and T. Nguyen. Cost- and energy-aware load distribution across data centers. In *HotPower*, 2009.
[10] D. Meisner, B. T. Gold, and T. F. Wenisch. Powernap: eliminating server idle power. In *ASPLOS*, 2009.
[11] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making scheduling "cool": temperature-aware workload placement in data centers. In *ATC '05: USENIX Annual Technical Conference.*
[12] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat. Medisyn: a synthetic streaming media service workload generator. In *NOSSDAV*, 2003.