Energy-Efficient Cloud Storage using Solid-State Drive Caching

Jorge Cabrera Salma Rodriguez Jesus Ramos Alexis Jefferson Tiffany Da Silva Ming Zhao
Florida International University
{jcabr020, srodr063, jramo028, ajeff003, tdasi002, mzhao}@fiu.edu

Abstract

In this paper, we explore ways to save power with Solid-State Drive (SSD) caching. We devise an online power monitoring scheme using the Watts Up? Pro power meter to evaluate our implementation. Our framework offers high performance by keeping a portion of the working set of processes in a local cache, where data can be accessed faster than when it is stored on the storage server disk. Our results indicate that an SSD cache uses less power in Write Back (WB) mode for both cache hits and cache misses. To improve on the initial power saving, we implement a management policy with Least Recently Used (LRU) as the replacement policy for workloads. We also implement a spin down policy to save power on the storage server when there is low disk activity, such as when the cache is large enough to hold the entire working set of running programs.

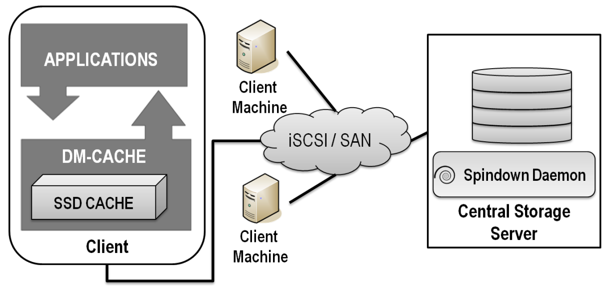

Figure 1: Architecture of proposed system

1 Introduction

Data center power consumption is growing very rapidly. In 2011 it was estimated that U.S. data centers consumed about 100 billion kWh of electricity at a staggering cost of $7.4 billion [4]. Because data centers house so many computers even small savings per machine can lead to large savings overall. Storage systems can consume up to 25% percent of energy in data centers [5]. The reason behind this is found in the primary storage device medium, the Hard disk drive (HDD). HDDs waste a significant amount of energy because they are kept spinning even during idle IO periods [1].

Power consumption of an SSD when compared to standard HDDs is much lower due to the lack of mechanical parts. We can leverage SSDs in data centers not only to improve throughput and performance of applications, but to save energy as well. Using an SSD as a local cache device can mitigate the cost of reading from a disk by providing a caching layer between physical memory and the HDD. Energy is saved not only by issuing less requests to the disk but also from the increased performance granted by the SSD. A workload can finish in much less time if requests are satisfied from faster storage which means the workload will consume less energy.

Our work focuses on energy saving in cloud storage systems through the use of block level caching mechanisms. We use DM-Cache [3] as the block level caching mechanism to evaluate our work on as it is freely available as a kernel module and is currently in use in production systems such as CloudVPS, a cloud service provider [2]. Our proposed approach consists of implementing DM-Cache in shared storage systems, and extending this framework to support an HDD spin down mechanism. Our results show that energy can be saved for workloads with high read hit ratios using a write-back cache.

2 Design and Implementation

The version of DM-Cache used in our framework uses a radix tree for managing the cached data. The original version only supports write-through caching, but for the purpose of our system we extended its functionality so that it could support write-back caching. The cache managament policy used is LRU. Eviction of cached blocks is done when there is no more space in the cache device, by replacing LRU blocks that are already stored in the main source storage. In our implementation, we use a user space daemon that interacts with the storage server for dynamically spinning the disk up or down. The setup for the spin down mechanism includes the user space client daemon, a web server that listens for incoming disk commands, and a function that checks the time since the last cache miss, notifying the client daemon when it is time to spin the disk.

	iSCSI-W	DMC-WT	DMC-WB-M	DMC-WB-H
Client	82566.1	84782.9	68361.1	54465.9
Server	90366	91715.7	72831.3	57913.1
Total	172932.1	176498.6	141192.4	112379

Table 1: Energy consumption (in Joules) of write operations

	iSCSI-R	DMC-R-M	DMC-R-H
Client	66433.4	101134.3	50929.7
Server	74024.8	109196.9	55010.2
Total	140458.2	210331.2	105939.9

Table 2: Energy consumption (in Joules) of read operations

3 Evaluation

We present the results of evaluating the energy-saving effects of using a block-level solution on a shared storage environment. To measure the power consumption of a node we used the Watts up? PRO power meter [6]. which measures the power consumption in watts with an interval as fast as one second. There are two configurations that are tested; the first is the baseline case where a client machine accesses its back-end storage through an iSCSI connection to a remote storage server. In this case, both client and server machines are utilized when I/O is performed. The client's memory capacity will determine the amount of data that is cached from the server-side and thus the amount of interactions with the server. As for the DM-Cache setup, an SSD is used an as additional cache layer between the client and the server. This layer provides a larger capacity to cache more data from the server-side, and thus reduce interactions with it.

Table 1 shows the energy consumption of both the client and server machines when performing write requests. The second column (iSCSI-W) shows the energy consumption of writes performed on the plain iSCSI system setup. This is the value that we'll use as a baseline to compare the energy consumption of a system running with DM-Cache. The next column shows the energy consumption of writes performed on a system using the write-through configuration of DM-Cache (DMC-WT). Because write-through operations on DM-Cache are similar to regular iSCSI writes, the difference between energy consumption of these two is not too much. The fourth column, a write-back miss (DMC-WB-M), shows energy power consumption of doing writes on a file that has not been yet cached. Even though this operation involves cache misses, the energy consumption for it is lower than the baseline, because the cache misses result in dm-cache allocating data structures to hold metadata about the new blocks of data, and no data is actually read from the storage server. The last operation involves write hits. These are the fastest and also the most energy efficient of all write operations.

Table 2 shows the energy consumption of sequential read requests. The second columns shows the energy consumed by reads performed on the system without DM-Cache. The next column shows the results of read misses using DM-Cache. The results in this particular category show the additional energy consumed when a read causes a cache miss. The additional energy is caused by the fact that both the client and server are actively performing I/O, and also because of the length of the operation. The last column shows the energy saving potential of client-side caching, as a read hit operation shows a significant reduction in energy consumption.

Results confirm previous expectations that an SSD cache can help reduce power consumption. For operations where I/O's are served from the cache device, SSD's can save up to 25% energy savings for read operations, and 35% for write operations.

4 Conclusion

This paper proposes an improvement to leverage current block level caching mechanisms to save energy in cloud storage systems. In order to better exploit caching mechanisms we adopt a caching solution and modify it to support smarter cache eviction policies. To take advantage of long idle periods on the storage nodes a daemon that interacts with the cache and storage node was created to spin down the mechanical disks on the storage node in order to save energy. These changes led to lower power consumption of storage nodes as some requests could now be serviced by more energy efficient SSDs.

References

[1] BISSON, T., BRANDT, S. A., AND LONG, D. D. E. A Hybrid Disk-Aware Spin-Down Algorithm with I/O Subsystem Support. In *IEEE International Performance Computing and Communications Conference* (2007).

[2] CLOUDVPS. CloudVPS activates Linux SSD Caching with dm-cache. `http://www.cloudvps.com/blog/cloudvps-activates-linux-ssd-caching-with-dm-cache/`.

[3] HENSBERGEN, E. V., AND ZHAO, M. Dynamic Policy Disk Caching for Storage Networking. Tech. Rep. RC24123 (W0611-189), IBM Research, November 2006.

[4] U.S. EPA. Report to congress on server and data energy efficiency. Tech. rep., U.S. Environmental Protection Agency, Aug. 2007.

[5] VERMA, A., KOLLER, R., USECHE, L., AND RANGASWAMI, R. SRCMap: energy proportional storage using dynamic consolidation. In *FAST USENIX Conference on File and Storage Technologies* (2010).

[6] WATTS UP? Watts up? Products: Meters. `https://www.wattsupmeters.com/secure/products.php?pn=0`.