Senior Project

Spin Down Policy for Device Mapper Cache

Student: Salma Rodriguez, Florida International University
Mentor: Prof. Ming Zhao, VISA Research Laboratory
Instructor: Masoud Sadjadi, Florida International University

Problem Statement

Can we save power in storage servers by spinning down the disk dynamically while caching data on solid state drives?

Motivation

Solid State Technology

High capacity EEPROM devices have been shown to reduce energy consumption when used as local cache for hard disk drives.

Distributed SSD Caching

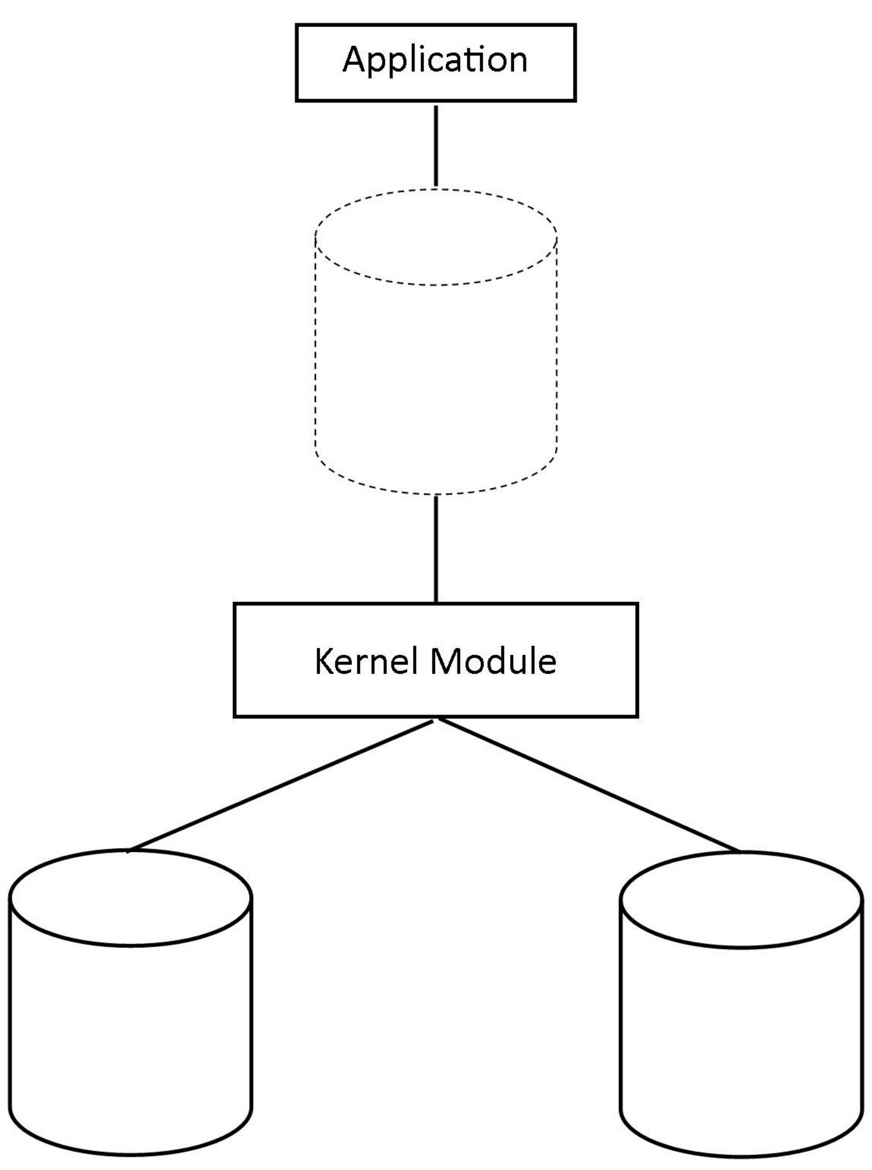
With our caching solution, we hope to increase the working set in flash memory and keep server disks spun down.

Properties to Explore

We explore the properties of dynamic spin down of storage server disks.

Linux Device Mapper

- Device Mapper facilitates mapping between two block devices.
- Only pseudo device in volatile memory is visible to applications.
- Kernel module uses device mapper to map block I/Os sent to pseudo device onto real devices (source and target).
- Devices may not necessarily be locally attached.

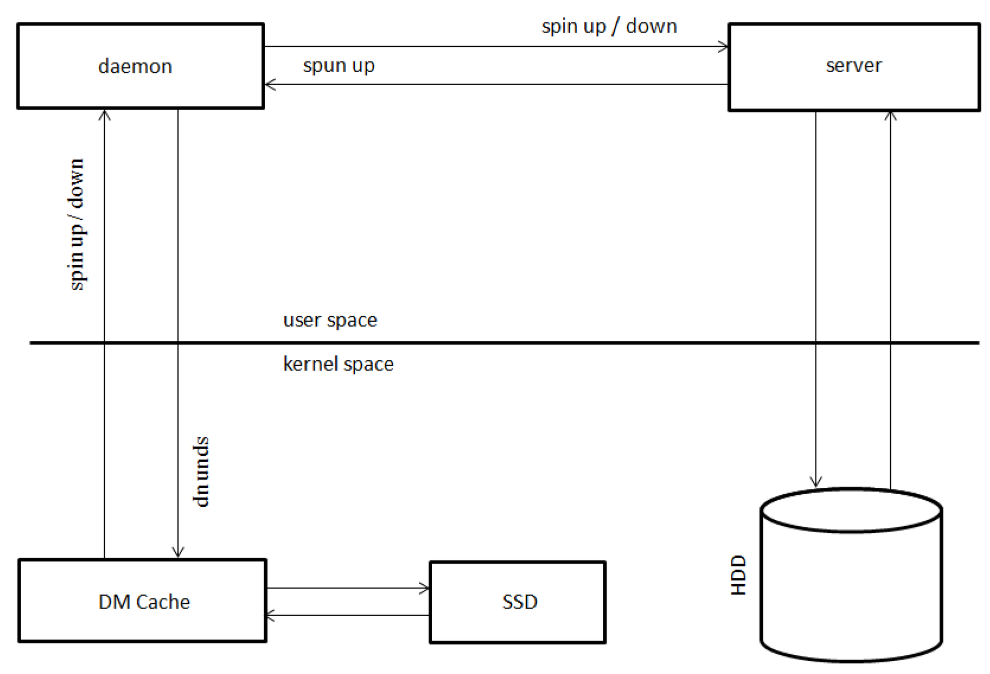


Spin Down Policy

Design

Dynamic spin down policy for Device Mapper Cache (DM Cache) Keep server disks (source device) spun down when cache is bigger than the working set.

Figure 1: The state of each disk is controlled dynamically.



Implementation

Algorithm 1 spinning the disk up or down dynamically

Precondition: server disk is spinning

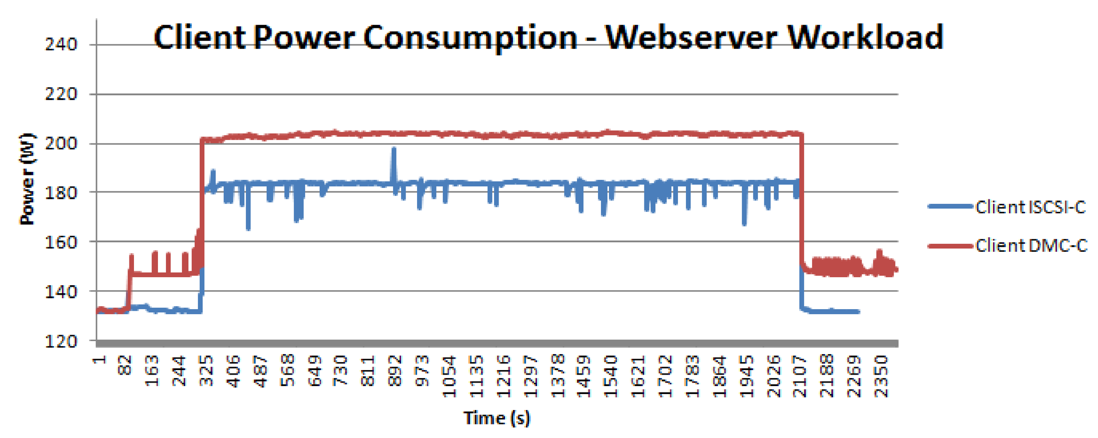
```
procedure Spin Up or Down
    T ← constant time
    while true do
        sleep for T seconds
        if disk is spinning then
            k ← current time in sec
            c ← last cache miss
            if c + T ≤ k then
                spin down the disk
                state ← not spinning
        else        ▷ disk is not spinning
            if blocking then
                spin up the disk
                state ← spinning
                unblock DM Cache
```

Evaluation

The baseline for our results is a simple iSCSI storage setup. We ran a 30 min test using a webserver-based workload, and observed that context switches increase power by 14W more than the baseline.

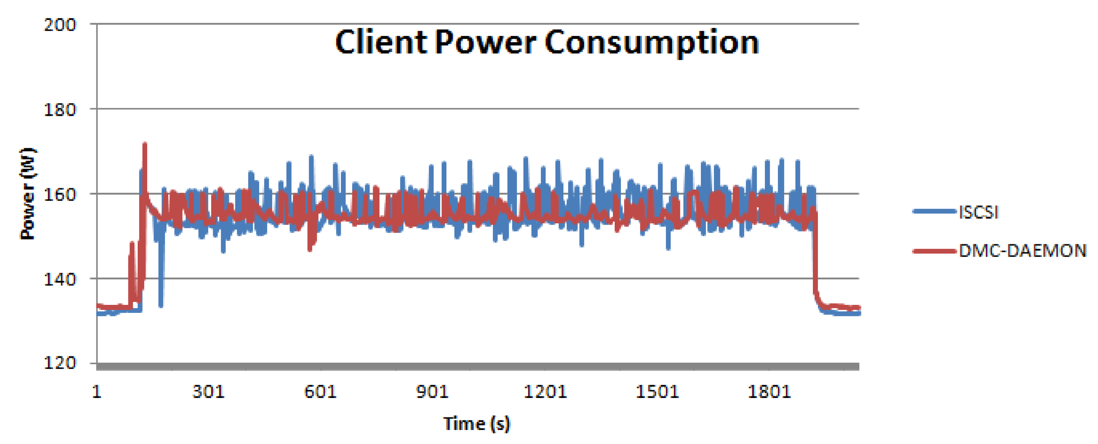
Spin Down with Schedule

Figure 2: Red plot: DM Cache with spin down daemon. Blue Plot: iSCSI without DM Cache.



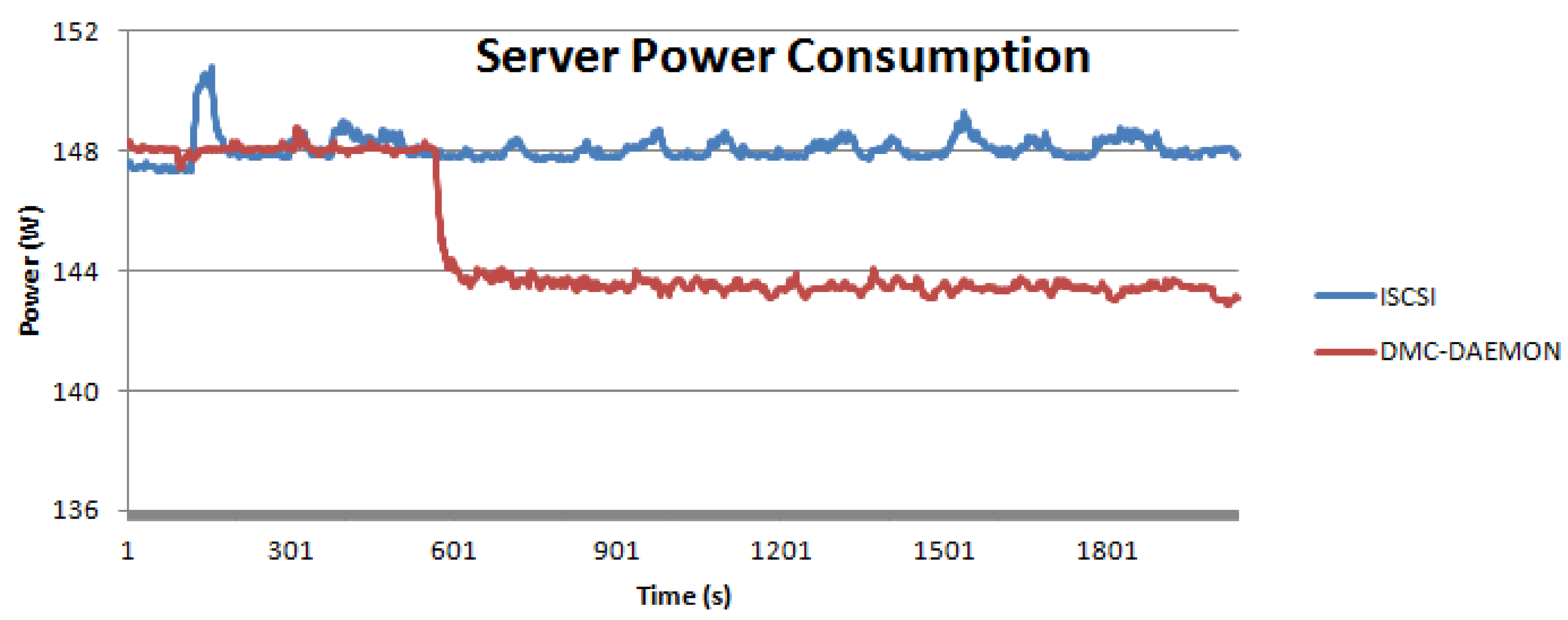
Spin Down with Sleep

Figure 3: Red plot: DM Cache with spin down daemon. Blue Plot: iSCSI without DM Cache.



Spin Down Results for Server

Figure 4: Spin down daemon reduces power on storage server.



Acknowledgements

I extend my gratitude to Jorge Cabrera for assisting me with the benchmarks and the shell scripts that allow the spin down daemon to interact with the disks on the storage server. I am also thankful to Jesus Ramos for pointing out the "grammar squiggles" in my draft of figure 1.