# Streaming Media from Flash with Less Power Consumption

**Ankit N Sanghvi[1]**
[1]Department of Electronics and Communication
[1]L. J. Institute of Engineering and Technology, GTU, Gujarat, India

*Abstract*---Reducing energy consumption has been an important design issue for large-scale streaming media storage systems. Existing energy conservation techniques are inadequate to achieve high energy efficiency for streaming media computing environments due to media is streamed from the hard disk drive. To address this problem, we propose in this paper a new energy-efficient method stream media from flash memory. Streaming media from flash memory significantly reduces power utilization because of two reasons. First, a set of hard disks in media server side consumes more power compared to flash memory. Second, heat dissipation on the media server side. Therefore Streaming media from the flash memory is conducive to saving more energy for the long media data. This approach is achieved using the kernel and device driver concepts on the Linux operating system. Our experimental results show that it can save up to 28.13%-29.33% energy consumption. Energy efficiency of streaming media storage systems can be improved by 3.3~6.0 times.

**Key words:** Energy Consumption, Streaming Media, Flash Memory, Hard Disk Drive, Kernel.

## I. INTRODUCTION

Streaming Media may be defined as listening or viewing media in real time as it comes across the World Wide Web. With streaming technology, users can watch and listen to media while it is being sent to their browser, instead of waiting for it to completely download and then playing it. Before streaming technology was available, a user might wait an hour (or more!) to completely download a short media file.

In general, media files are huge. For example, five minutes of uncompressed video would require almost one gigabyte of space! So, when the audio and video is prepared for streaming, the media file is compressed to make the file size smaller. When a user requests the file, the compressed file is sent from the video server in a steady stream and is decompressed by a steaming media player on the user's computer to play automatically in real time. A user can jump to any location in the video or audio presentation. Streaming media generally tries keep pace with the user's connection speed in order to reduce interruptions and stalling. Though general network congestion is unavoidable, the streaming server attempts to compensate by maintaining a constant connection. [7]

The traditional mainstay in storage technology has been the hard disk drive (HDD). However, while the capacity of HDDs has increased 40% annually, their random input/output (I/O) performance has increased only 2% annually. This means that for some of today's enterprise, web, cloud, and virtualized applications that require both high capacity and performance, HDDs may not deliver a cost-effective storage solution, even with their significant drop in cost per GB. Recently, an alternative storage device technology, the solid state drive (SSD), has started to gain prominence. SSDs offer exceptionally high performance but have much less capacity per drive. They are also relatively expensive when compared to HDDs, and have a write endurance limit. Given the properties of HDDs and SSDs, IT departments now have a choice, but also a challenge, in determining the best way to cost-effectively fulfill the performance and capacity requirements of their enterprise applications. To meet this challenge and determine how they should integrate HDDs and SSDs into their storage fabric, we must first quantify the performance, capacity, and cost value of SSDs vs. HDDs for different applications. The objective is to most effectively take advantage of the different cost, performance, and capacity characteristics provided by HDDs and SSDs. [8]

Existing model is storing and streaming the media from the Hard Disk Drive on the server side so power consumptions very much and also the heat dissipation is very much. To overcome this power consumption by Hard Disk Drive we will use the Flash memory whose power consumption is less comparatively to Hard Disk Drive. Now we are going to Store the media or data on the Hard Disk Drive and streaming from the Solid State Drive. We will storage the data from the HDD to SDD so we can stream the media and data from the SDD. At the time Storing the Data or Media from the HDD to SSD the HDD will go into the sleep mode after transferring the media and data so the now only flash memory will be used for streaming .By doing this power consumed and heat dissipation by the flash memory is comparatively less than Hard disk drive . By implementing this model power management can also be done.

## II. BACKGROUND AND RELATED WORK

### A. Streaming Media and Its Characteristics

In streaming media applications, users can enjoy video or audios as they are being downloaded to computers. For streaming media servers, disk bandwidth is usually the performance bottleneck, because a large number of online users who need video data from server every second bring a mass of small-block random disk accesses. Generally speaking, streaming media services are considered as a data-intensive application. However, compared with traditional data-intensive applications, streaming media systems have their own special features:

*1) High QoS Constraints:*-The QoS (Quality of Service) of streaming media applications is usually measured in startup delay and jitter rather than response time Moreover, the QoS constraints are much stringent than most online and all offline applications; low QoS services often mean unacceptable services for streaming media users. High bandwidth must be preserved to serve user requests to guarantee high QoS, so there is very limited bandwidth dedicated for data migrations among disks. In other words, streaming media application has weak tolerance for high data migration overheads.

*2) Large and Fast-Growing Storage Demand:*-In the world's newly produced data each year, multimedia data, especially videos, are forming the largest category. Videos are accumulated extremely fast. For example, video materials about 12.6 million hours are published on YouTube each year [2]. The number of accumulated videos in Facebook grows up as high as 239% per year [1]. Such a large and fast-growing storage demand makes streaming media application extraordinarily relies on highly cost-effective storage devices, i.e. cheap and large-capacity SATA disks instead of expensive modern storage devices like flash and multi-speed disks. The storage capacity required by media libraries is tremendously large compared with storage space offered by the main memory. Another negative factor is that the streaming media users dynamically change their interests all the time (e.g., Yu et al. [5] discovered that after one hour, the changing rate of the top 200 videos may be high up to 60%). Such a high rate of changing and the limited capacity of memory compared with disks make the cache hit rate very low. In addition, among such a large data volume of streaming media, a considerable part of videos are seldom or never accessed after being uploaded to servers [3]. Evidence indicates that the primary data set of streaming media itself has great potential and requirements to save energy, even without relying on redundant storage.

3) *Distinctive Behaviour of Streaming Media Users:*-The daily workload is drastically fluctuant for streaming media users. In a considerable part of daily time, the load is very high (for example, there are a numerous of requests in the afternoon and evening, but few requests in the early morning [5]). In addition, streaming media users usually abort watching much before the end of videos. Yu [5] pointed out that 37.44% users leaves in the first 5 minutes of watching, and the majority of partial sessions (52.55%) are terminated by users within the first 10 minutes. Therefore, the accuracy of prefetching streaming media data will be low in the prefetching-based energy saving algorithms [4] because of the numerous unpredicted early aborted users. In fact, much of the prefetched data may be useless at all, with the results of wasting much storage and transmission resources in streaming media applications.

### *B. Disk Power Management*

Conventional disks spin at full speed regardless of the *active* or *idle* modes. The disks can completely stop spinning when they are placed in the *standby* mode to save energy without being able to serve any request. The goal of disk power management (DPM) schemes is to conserve energy by turning disks to the low-power mode without adversely affecting I/O performance. The FT scheme. The Fixed Threshold scheme or FT is the most popular DPM for conventional disks [6]. FT places a disk in the low-power mode after a fixed threshold time has elapsed since the last I/O request. The threshold is usually set to the *break-even* time – defined as the smallest period for which a disk must stay in the low power mode to offset the extra energy spent in spinning the disk down and up.

## III. COMPARISONS OF FLASH MEMORY AND HARD DISK DRIVE

Unlike mechanical hard disks, the capabilities of Flash solid state storage match quite well with our requirements. First, Flash SSS has no mechanical moving parts, making the system more reliable and less sensitive to vibrations and rough landings... Also, Flash SSS weighs substantially less than HDD storage, and it uses much less power. SSS processes around 1,000 transactions per Watt, while HDD processes 5- 15 per Watt. In crowed electronics areas this translates into much less heat and lower cooling requirements. SSS is far denser than HDD systems as well, which means it takes up less space for the same amount of storage. [8]

## IV. EXISTING WORK

Client application request media content for streaming. Media server requests kernel to open the file after that Kernel open the media file from hard disk and Media server read the file from hard-disk and stream to client.

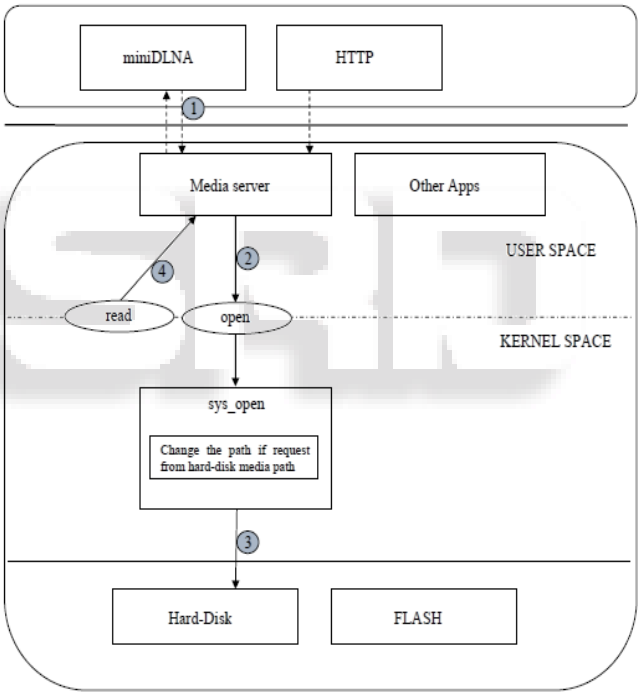


Fig. 1: Existing System

## V. KERNEL SPACE AND USER SPACE

A kernel is an OS. In general, the OS is used to denote the entire package that controls resources of a computer such as GUI, file utilities, and command-line interpreters. A kernel is a small version of OS, and it is a resource manager. Whether the resource being managed is a process, memory, or hardware device, the kernel manages the access to the resource between multiple competing users (both in the kernel and in user space).

Even though we can run a program without a kernel, but the very presence of a kernel greatly reduces the complexity of writing a program or using other programs. In other words, it increases the power and flexibility by providing a software layer to manage the limited resource of a machine. - From the Linux Programming Interface.

The Linux kernel executable typically resides at the pathname/boot/vmlinuz or something similar (Fedora18, its /boot/vmlinuz-3.6.10-4.fc18.x86_64)

CPU can typically have at least two different modes: user mode and kernel mode. Hardware instructions allow switching from one mode to the other. Areas of virtual memory can be divided as user space or kernel space.

When running in user mode, the CPU can access only memory that is in user space, and attempting access memory in kernel space may result in a hardware exception. However, when running in kernel mode, the CPU can access both user and kernel memory space.

### A. *User Space*

This is where the user applications are executed.

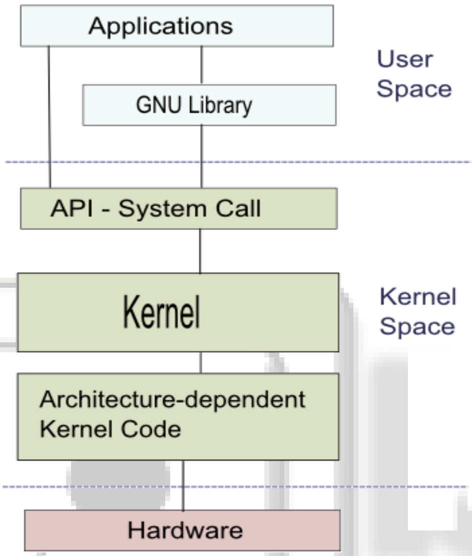


Fig. 2: Layers of Execuion

### B. *GNU Library*

This provides the system call interface that connects to the kernel and provides the mechanism to transition between the user-space application and the kernel. This is important because the kernel and user application occupy different protected address spaces. While each user-space process occupies its own virtual address space, the kernel occupies a single address space.

### C. *Kernel Space*

The Linux kernel can be further divided into three levels. At the top is the system call interface, which implements the basic functions such as read and write. Below the system call interface is the kernel code, which can be more accurately defined as the architecture-independent kernel code. This code is common to all of the processor architectures supported by Linux. Below this is the architecture-dependent code, which forms what is more commonly, called a BSP (Board Support Package). This code serves as the processor and platform-specific code for the given architecture

## VI. Daemon And Event Handler

A daemon is a program that runs as a "background" process (without a terminal or user interface), commonly waiting for events to occur and offering services. A good example is a web server that waits for a request to deliver a page, or a ssh server waiting for someone trying to log in. While these are full featured applications, there are daemons whose work is not that visible. Daemons are for tasks like writing messages into a log file (e.g. syslog, metalog) or keeping your system time accurate (e.g. ntpd). The word daemon is sometimes used for a class of programs that are started at boot but have no process which remains in memory. They are called daemons simply because they utilize the same startup/shutdown framework (e.g. systemd service files of Type oneshot) used to start traditional daemons. For example, the service files for alsa-store and alsa-restore provide persistent configuration support but do not start additional background processes to service requests or respond to events.

From the user's perspective the distinction is typically not significant unless the user tries to look for the "daemon" in a process list.

## VII. Proposed Work

When Client application request media content for streaming. After that Media server requests kernel to open the file then after Kernel checks if the file is requested from hard-disk media path if the file is requested from the hard disk media path then it Lookup the media file and change the path of media file if required and Generate a path change event and blocks for response after that User space daemon listen the event and start file copy from hard-disk to FLASH after the file is copied to the flash a Daemon gives path change response. Then after the sys_open resumes and Sys_open () opens the file from FLASH. if the file is not requested from the hard disk media path Sys_open() opens the file from FLASH and Media server read the file from FLASH and stream to client.

## VIII. Conclusion

Streaming media from the Hard disk drive at media server side there is power consumption is comparable more to Flash. So over here we just copy the data from the hard disk drive on request to the flash memory .so after the media is copied to the flash memory the hard disk drive will be in sleep mode until the interrupt is given to it. Now the media will be streamed from the flash memory .so that the less power is consumed on streaming the media and also heat dissipation will be reduced and vibration will be reduced on the media server side and also the power management is can be done

## References

[1] J. Yarow, “Videos On Youtube Grew 123% Year Over Year, While Facebook Grew 239%”, Http://Www.Strangelove.Com/Blog/2010/06/ Videos-On-Youtube-Grew-123-Year-Over-Year-While-Facebook-Grew-239, June 2010.
[2] C. Yates, “Top 10 Video Stats”, Http://Www.Huddleproductions.Com/?P=1014, June, 2011.
[3] X. Cheng, C. Dale and J. Liu, “Statistics and Social Network of Youtube Videos”, 16th International Workshop on Quality of Service, Pp. 229-238, 2008.

[4] A. Manzanares, S. Yin, X.-J. Ruan, and X. Qin, “Pre-Bud: Prefetching For Energy-Efficient Parallel I/O Systems with Buffer Disks,” Acm Trans. Storage, 2010.
[5] H. Yu, D. Zheng, B. Zhao And W. Zheng, “Understanding User Behavior In Large-Scale Video-On-Demand Systems”, Proc. 1st Acm Sigops/Eurosys Conf., Pp. 333-344, Leuven, Belgium, April 2006.
[6] E. Pinheiro and R. Bianchini, “Energy Conservation Techniques For Disk Array-Based Servers”, Proc. 18th Int’l Conf. Supercomputing, Pp. 68-78, June 2004.
[7] Http://Www.Cod.Edu/It/Streamingmedia/Intro.Htm
[8] Ssd Vs Hdd Price And Performance Study, Www.Dell.Com/Downloads/Global/Products/Pvaul/En / Ssd_Vs_Hdd_Price_And_Performance_Study