

Elimination of Bootload Time of an Operating System

Sohan Nipunage, Surbhi Shukla, Mayuri Patil

Abstract—There has been significant improvement in the computing power by inclusion of various technologies that are needed to be integrated into one single operating system for the convenience of users. But there is no improvement in the boot load area of the operating system. Hence our focus has been apprehended on the boot load time since it was of immediate concern to decrease the booting time. The proposed operating system is being developed by modifying the Linux kernel and will use a log based system which will resume the leftover scenario when restarted. We propose a software solution by modifying Linux kernel that can replicate the left over scenario of the desktop of the user when the user logs off and loading operating system eliminated boot time.

Index Terms—Bootting, GRUB (Grand unified Boot Loader), Hibernation , Kernel, Log based Bootting,

1 INTRODUCTION

THERE has been significant improvement in the computing power by inclusion of various technologies that are needed to be integrated into one single operating system for the convenience of users. Hence the focus has been apprehended on the boot load time since it was of immediate concern to increase the computing power. Boot load is a technology that has not seen many advances in recent years. It has only improved in the terms of computing technology but there has been no significant improvement in the boot load area of the Operating system till date since there are hardware limitations to the same.

We propose a software solution by modifying Linux kernel that can replicate the left over scenario of the desktop of the user when he logged off. The loading of other drivers and features can happen in the background. As they say, Time is Money; we are providing a solution to completely eliminate the boot time of Linux based operating systems. Linux is the first truly free Unix-like operating system. The GNU Project was launched in 1983 by Richard Stallman to develop a Unix-compatible operating system called GNU, which is free software. Many programs and utilities were contributed by developers around the world, and by 1991 most of the components of the system were ready. Still missing was the kernel. The use of logs for debugging is as antediluvian as computing itself. Log-based responses are useful not only for understanding the internal operation of a system but also the timing and relationship of activities within the system through the time-ordered messages within a time-stamped log.

2 EXISTING SYSTEM

2.1 Operating System

An operating system is the program that, after being initially loaded into the computer by a boot program, manages all the other programs in a computer. The other programs are called applications or application programs. The application programs make use of the operating system by making requests for services through a defined application program interface (API). In addition, users can interact directly with the operating system through a user interface such as a command language or a graphical user interface (GUI). For example, Linux OS, Mac OS, Windows OS.

An operating system performs these services for applications:

- In a multitasking operating system where multiple programs can be running at the same time, the operating system determines which applications should run in what order and how much time should be allowed for each application before giving another application a turn.
- It manages the sharing of internal memory among multiple applications.
- It handles input and output to and from attached hardware devices, such as hard disks, printers, and dial-up ports.
- It sends messages to each application or interactive user (or to a system operator) about the status of operation and any errors that may have occurred.
- It can offload the management of what are called batch jobs (for example, printing) so that the initiating application is freed from this work.

2.2 Kernel

The Kernel within the Windows operating system is somewhat different than that of LINUX because Microsoft has taken a somewhat different approach to the Kernel's design. The term "Kernel mode" refers to not only the Kernel itself but the

- Author Sohan Nipunage is currently pursuing bachelors degree program in information technology engineering in Mumbai University, India, PH-+919867509494. E-mail: sohan.nipunage@gmail.com
- Co-Author Surbhi Shukla is currently pursuing bachelors degree program in information technology engineering in Mumbai University, India, PH-+918149343226. E-mail: shuklasurbhi010@gmail.com
- Co-Author Mayuri Patil is currently pursuing bachelors degree program in information technology engineering in Mumbai University, India, PH-+919869559666. E-mail: patilmayuri2293@gmail.com

HAL (hall.dll) and various system services as well. This would include the HAL (hal.dll) at the bottom layer, followed by the Kernel at the second layer. The third layer is the Kernel. Above that (fourth level) we have various managers for Objects, processes, Memory, Security, Cache, Plug in Play (PnP), Power, Configuration and I/O. The file system and Win32 GDI are also at this level. Many of the items at this fourth level are referred to collectively as the Windows Executive (ntoskrnl.exe). At the top and fifth layer of the Kernel mode are system services.

2.3 Bootloader

Boot loader or boot strap loader is a program that starts the 'chain reaction' which ends with the entire operating system being loaded. Boot loader comprises of BIOS (Basic Input Output System) or Boot monitor which executes Master Boot Record (MBR) located in the first sector of bootable disk. MBR contains information about Grand Unified Boot loader (GRUB) or Linux Loader(LILO). GRUB loads and executes kernel and kernel executes init() method. Init identifies the default init level from /etc/inittab and uses that to load all appropriate programs.

2.4 Hibernation

Now a days it is common in several laptops that they have a suspend option. Suspend option technically saves the state of the machine to a file system or to a partition and switches to standby mode. Later resuming the machine the saved state is loaded back to RAM and the machine can continue its work. Upon hibernation, the computer saves the contents of its random access memory (RAM) to a hard disk or other non-volatile storage. When the system is resumed, the computer is exactly as it was before entering hibernation. When used to save power as computing isn't being performed, hibernation is similar to sleep mode; however, it saves more power at the cost of slower resumption. TuxOnIce (formerly known as Suspend2) is an implementation of the suspend-to-disk (or hibernate) feature which is available as patches for the 2.6 Linux kernel. During the 2.5 kernel era, Pavel Machek forked the original out-of-tree version of swsusp (then at approximately beta 10) and got it merged into the vanilla kernel, while development continued in the swsusp/Suspend2/TuxOnIce line. TuxOnIce includes support for SMP, high memory and preemption.

2.5 Problems In Existing System

- Booting time is comparatively more (10sec with high-end hardware).
- No solution for loading process snapshot on boot.
- No software solution for independent hardware platform.
- Slow operating boot load response and hence time consuming.

PROPOSED SYSTEM

We propose a software solution by modifying Linux kernel that can replicate the left over scenario of the desktop of the user when he logged off. The loading of other drivers and fea-

tures can take place in the background. The use of logs for debugging is as old as computing itself. Logs are useful not only for understanding the internal operation of a system but also the timing and relationship of activities within the system through the time-ordered messages within a time-stamped log. Some of the unique advantages of our proposed system are:

- Significant improvement in the computing power for RAID systems.
- Improvement in the CPU cycles by enhancing the CPU utilization to its maximum performance.

3.1 Log Based Booting

Log based systems provide a system time stamp that helps to perform the tasks that may not be existent after the system is restarted. This helps in defining a predetermined path for execution of processes. With this log based system we can implement the user process before booting the operating system so that we can execute the same processes that the user left before shutting down the operating system.

3.2 Device Drivers

The device drivers are soft wares that control the basic system hardware devices such as keyboards, graphics, display adapters, memory and externally connected devices. These device drivers are important as they are the building blocks of the operating system control.

Our proposed system will provide loading of those drivers that are needed for the system to load so as to decrease the unnecessary booting time of the operating system. We consider a specific memory limit so that if the user process has memory that is excess to be handled by the memory management of the operating system then we can schedule it at the end since we want immediate system boot up.

3.3 Kernel (Modified)

Our proposed system is based on modifying the Linux Kernel which is available for free as it is open source development. The kernel will load the parameters of the processes from the log and ensure that the booting time is reduced. We are modifying Kernel at its basic level where we are concerned about reducing the booting time and not reducing the process lag or increasing the process speed.

3.4 Replication Of Hibernation Process

Hibernation is a process that is designed to save power. Here the hibernation enters the operating system into sleep mode by saving the user processes with their memory management and addresses into RAM and then turns of the power for the devices that need not consume the efficient power such as Display, HDD, CD-DVD drive, Keyboard, mouse, external devices, etc. Still the power supply can't be turned off since to keep memory and addresses bus active it needs to have power supply on. Now our proposed system will replicate and not implement the hibernation process by modifying the kernel. The system will save the basic application processes that the user is accessing before shutting down the system and thus will be added into the system log which will be accessed later

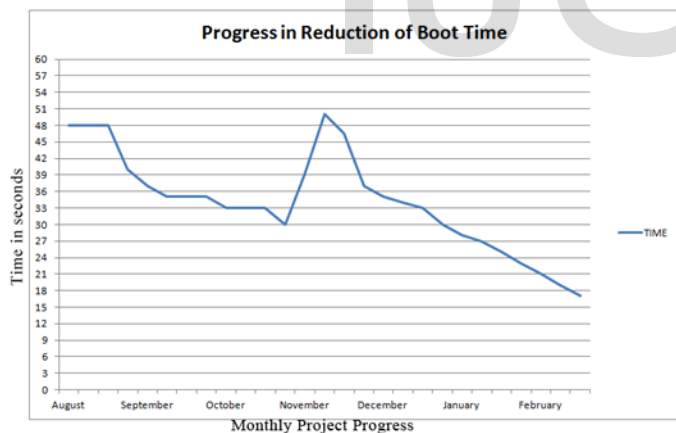
by the kernel during the next boot of the operating system.

3.5 Requirement Analysis

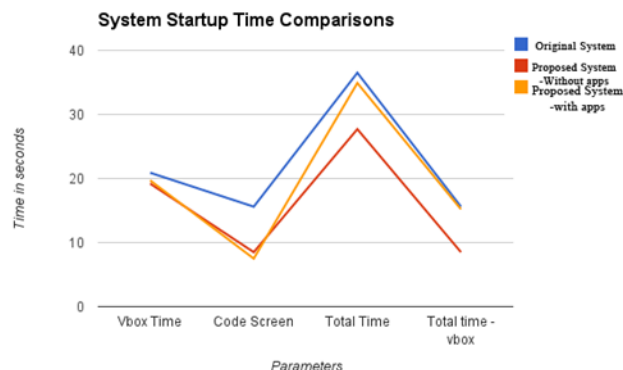
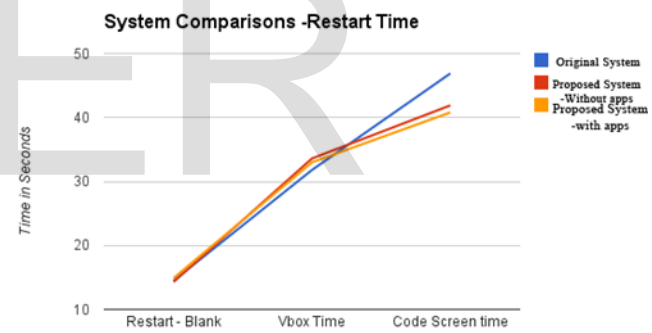
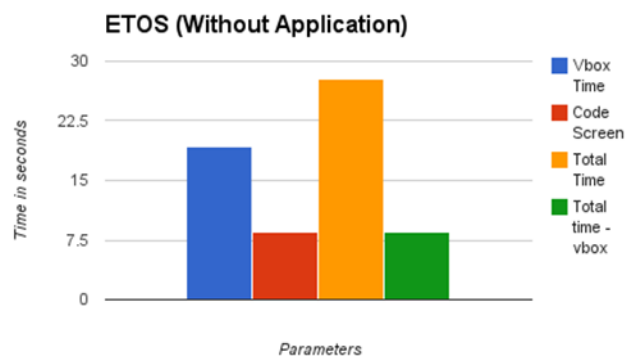
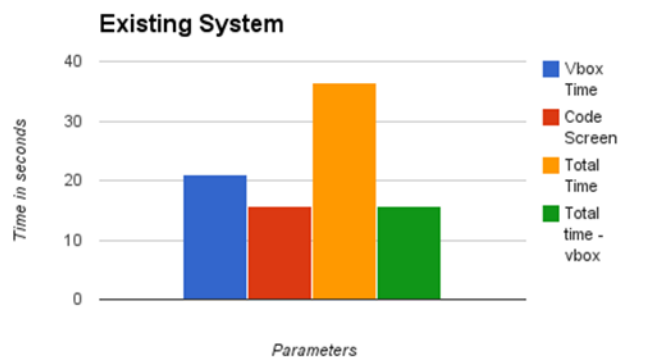
Our proposed system is based on modification of the Linux kernel so as the requirements for our project are comparatively less and are more basic in terms of functionality and operating functions. Hence we provide a basic standard of requirements that we are working and since our project can be scaled from low parametric hardware to high-end parametric hardware we are more focused on its system independent implementation.

4 CONCLUSION

With development of research in elimination of boot load time, it will be possible to develop a fast responsive operating system that will result into a better performance and integration of high speed tools. Not only a pc would be benefitted from the research but with more modification for suitable devices such as personal PDA's phones and smart devices will enhance the usage of technology in a more improved way. Research will be carried out by focusing on memory management and registry addressing by which a modified Linux kernel can load hibernation-like system to eliminate the boot time of Linux based operating system. A smart, fast, easy and highly responsive operating systems for future devices.



- Reduced boot time by 45 seconds to 11 seconds excluding VM delay – Probable reduction from 10seconds to 3seconds on fresh installation
- Processes booting according to the modified sequence.
- Removed SATA/Flash Drive input check from the boot menu, loads faster by acknowledging from the previous boot.



4.1 CERN

A Large Hadron Collider Project Carried out by CERN in Geneva generates 15TeraBytes of data in one single experiment. They need almost a year to process the data and then perform the next experiment. Our system can be installed on the super

computers and can result into an increased system response time as our system can be modified as an application and hence with eliminated boot time, system can switch on/off constantly on completion of the processes and thus conserving energy and reducing time of processing. This will improve their productivity and hence help in further development of their project.

REFERENCES

- [1] R.Love, "Linux Kernel Development".
- [2] J.Feldman, "The Linux startup process", Hewlett-Packard.
- [3] V.Mahajan, "Hibernation in Linux".
- [4] Daniel P. Bovet, "Understanding Linux Kernel".
- [5] Tim R. Bird, "Methods to Improve Bootup Time in Linux", in Proc. Of the Linux Symposium, 2004.
- [6] Kunhoon Baik, "Boosting up Embedded Linux device: experience on Linux-based Smartphone".
- [7] <http://www.elinux.org>
- [8] <http://www.tldp.org/>
- [9] <https://www.kernel.org>

IJSER