

# Kernel Improvement

Authors:

Itzel Cordero Sanchez A01113122

Daniel Contreras Segura

November 24, 2016

## 1 Abstract

The kernel, is the most important thing inside every computer, without it the interaction with the computer would be very different, the computers wouldn't have a mouse or a keyboard, maybe we would only use bottoms and a black screen. But also, it depend a lot for what purpose you are using the computer, in this days we have many times of computers, and each one has a specific purpose. For example, the servers are a specific type of computer that count with his own operative system, imagine that we improve the kernel in order that the server run faster and the company you are working for increase his earnings; that is the reason why is so indispensable to learn about the kernel.

In this project we will modify the Linux kernel, trying to improve its performance by modifying *nr\_migrate* variable, and compare the results using a benchmark.

## 2 Introduction

Everyone uses a computer but almost no one knows how it works. We need to know this because once we know it we can use to administrate the memory for all the programs and processes in execution in the computer. But if almost no one knows how works a computer, they will not know what is the kernel.

The kernel is the heart of the operating system, and it's like the middle men, the one between the hardware and the software. It is the responsible for access the various devices, also it is the one that organizes how all the programs execute in memory.

The kernel is also important when we start talking about the type of hardware that we are trying to use, for example, it isn't the same the kernel of the Android operating system, that the one in Windows or Mac OS, so there exist

many types of operating systems and each of those works in different ways to trying to manage the system resources more efficiently.

On this paper we will see the results of us trying to improve the work of the kernel modifying the *nr\_migrate*. We played with that variable and saw how the kernel's improvement varied.

### 3 Theoretical Framework

In this case we use a Linux operating system, and its kernel was created in 1991 by Linus Torvalds on his personal computer, since that day the Linux kernel has evolved with the contributions from nearly 12,000 programmers from more than 1200 companies. In our project we modify the kernel of the operating system, but that makes us wonder, what is the kernel? The kernel, also called nucleus, is a computer program that constitutes the central core of a computer operating system. Is the one that executes the programs, the one who access the different programs, and it's the one between the software and the hardware.

In the project we modify the *nr\_migrate* variable, that is the one in charge of specify the number of tasks that will move at a time an idle CPU.

### 4 Objective

The main objective of this work is to learn how to improve the Linux kernel for a specific system, for that purpose we tried with different variables to see how it behave, and check if it improves or not. The variable that we are improving is the *nr\_migrate* and in the Kernel its default value is 32, and we played with the numbers to see if the system improves using a benchmark. The thing is that the kernel is in part in charge of creating and destroying processes, so we are trying to improve that, that variable what does is that it "regulates" how many processes it can do at the time.

### 5 Justification

By improving the kernel of any OS, you can get a better performance in any task that your computer needs to run, and this advantage could be taken lightly, but in terms of the industrial competition this could mean, a big improvement in time and money. For example, if Google could make that the OS of his servers run faster, this could take them to another level in the industry.

With the *nr\_migrate* we could only tried a small part of it, because we need to play with the variables to see a real improvement,

but we can at least give us an idea. We tried from *nr\_migrate* =7 to *nr\_migrate* =52 and we noticed really irregular answers.

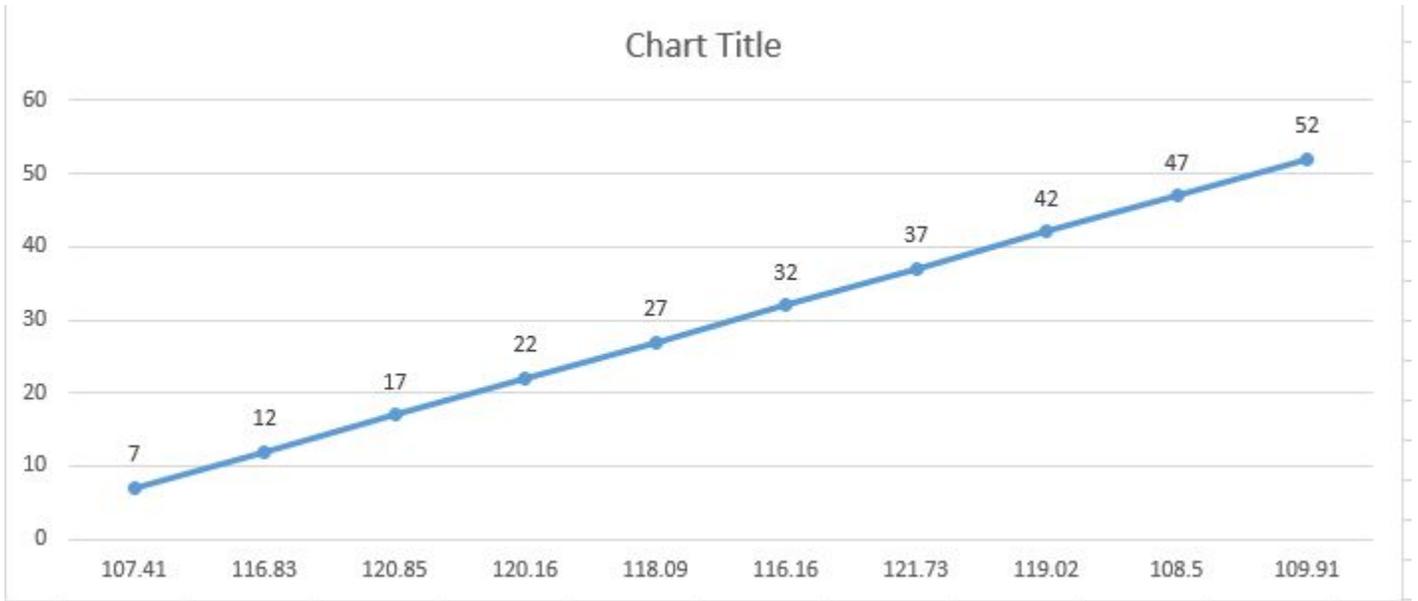


Figure 1: Line graph of the results of the AIO Stress test with different values.

## 6 Development

After we select the variable that we want to play with, we proceed to establish the range and the scale in which we want to modify the *nr\_migrate* parameter. The default value was 32, so we decide to establish a scale of five, and also we set the range from 7 to 52. In the phoronix benchmark we use an a-synchronous I/O test, and the parameters that the results gave us were the AIO-Stress of the cpu, the Standard Error, and the Standard deviation of the test. In the next table we show all the results that we get from the phoronix benchmark.

NR Migrate	AIO-Stress	Standard Error	Standard Deviation
7	107.41	1.83	3.41%
12	116.83	1.90	3.26%
17	120.85	1.01	1.46%
22	120.16	0.73	1.06%
27	118.08	0.45	0.66%
32	116.16	1.87	2.79%
37	121.73	2.41	4.84%
42	119	2.92	6.01%
47	108.5	2.79	6.30%
52	109.91	1.68	2.65%

## 7 Results

Well we notice that it has no patron. It does not matter if the variables were successive, our numbers were: 7, 12, 17, 22, 17.. and so on until 52, and the AIO-Stress at first were increasing but then at the number 22 it started to go down, and then at a 37 it started to go up again, and it repeated the patron.

## 8 Conclusion

We think that we need to modified more variables than just the *nr\_migrate* so we can notice an improvement. Because we need to play with the variables and numbers to make a difference and an improvement. With that variable we could not see any patron.

## 9 References

Kernel. *Chenyihe*. <http://dev.blankonlinux.or.id/browser/tambora/linux/debian/patches/features/all/rt/sche>  
limit-nr-migrate.patch  
Definicion de Kernel. *Definicionesabc*. <http://www.definicionabc.com/tecnologia/kernel.php>  
Linux Kernel Boot Parameters. *Redsymbol*. <http://redsymbol.net/linux-kernel-boot-parameters/2.6.21/>