RAM performance with modifications on the kernel

José Francisco Robles Sapién & Daniel Alejandro Jiménez Gómez

October 18th 2016

Abstract

In this paper, we measure the memory performance throughout the Phoronix test "RAMspeed SMP". We decide to test this specific benchmark because we know how important is the memory for the system performance. This document shows how much the memory performance could change if we modify some variables in the linux kernel.

1 Introduction

For instance, the RAM memory does not directly affect the processor performance. The RAM cannot make that the processor's clock rate increases and work faster; consequently, the memory do not increase the performance. Then, how the RAM is related to the performance? The processor need several instructions to work, the processor search them in the RAM, if the instructions are not stored there, they will have to be transferred from the HDD or SSD to the RAM. Having more amount of RAM memory imply that more instructions can fit in the memory, and bigger programs can be loaded at once. If the memory in the RAM is not enough to load a program, the processor creates a file in the HDD called Swap File, it is used to store the RAM data. The problem is that requesting data from the hard disk is way more expensive than order it from the RAM. For example, the processor communicates with the RAM at a transfer rate of 800 MB/s and with the fastest hard disk at 100 MB/s (Computer Memory Upgrade, 2003).

RAM not only permits that the CPU access files faster, it also helps to run more processes simultaneously. The RAM capacity, and the faster its cycles (MHz) are, is proportional to the number of processes the processor can run at the same time (Williams, n.d).

2 Theoretical Framework

RAM (Random-Access Memory) is the main place for storing instructions and data while a program is being executed. Before the CPU can run the program, the data is copied into the RAM. Each memory location in the RAM allocates one word of data. The words have a size, and it is how many bits the CPU can manipulate in one go (Bitesize, n.d).

RAM has two attributes that are directly related to the system's performance:

- Capacity: The bytes that the memory has to store instructions and data.
- Speed: The time that takes to the RAM to receive a request from the CPU and then read or write data (Dell, 2016).

Variables:

- Migrate (sched_nr_migrate) controls how many task are moved across processors through migration software interrupts (Occurs when a program ends or requests certain services from the operating system.) If many tasks are created by SCHED_OTHER, they will all run on the same processor. The default value is 32, If the value increases, the performance of the SCHED_OTHER threads will be boosted, having the disadvantage of increasing the latency for real-time tasks.
- Runtime (sched_rt_runtime_us), by default real-time tasks will consume 95% of the time (CPU/sec), and the other 5% is used by SCHED_OTHER tasks (Suse, n.d).

RamSpeed is a benchmark to test the performance of the RAM with different operations like Copy, simply moves data from one memory location to another; Scale, it is like Copy, but it multiplies the value by a constant before writing to the new location; Add, read a value from a memory location, then read other value from another place, adds them together and writes the result on a new location; Triad, combines the Scale and Add function, it is like the Add function with the exception that the first value is multiplied by a constant before it is added to the second value. All the operations are realized with 2 types of variables (integer and floating point). The results are presented in MB per second (Hopper, 2015).

3 Objective

The objective of this project is to see how modifications in the kernel improves or deteriorates the RAM memory performance of the execute process, in this case the process of "Add" with integer variables.

4 Justification

Nowadays the application programs are more complex, having the necessity of more RAM memory to run adequately, providing an outstanding user experience. Also a real technology trend is the multitasking; to have multiple programs running simultaneously it is necessary to have a larger amount of memory in order to have data and instructions closer to the processor.

Advantages:

- More space to store data and instructions closer to the CPU.
- The ability to open several programs simultaneously without crashing.
- Multitask
- Increase of performance because swapping is not needed.

Disadvantages:

- At some point, there will be no difference between 16Gb or 32 Gb of RAM.
- Power Consumption
- Expensive

5 Development

For this report, we used a Laptop with the next specification for software and hardware. The hardware was a processor Intel Core i5-2410M @ 2.90GHz (4 Cores), a motherboard LENOVO KL6, a chipset Intel 2nd Generation Core Family DRAM, a memory (RAM) of 4096MB, and a disk 750GB Seagate ST9750420AS.

The type of RAM Memory is a DDR3 DRAM with a speed of 1334 Mhz and a total and data width of 64 bits. This means that the memory is Double Data Rate 3 Dynamic Random-Access Memory. It can transfer data at a clock rate of 800-2600 MHz

Advantages:

- The transfer of data is faster
- Less consumption of energy due to the improvements in the performance at low energy.

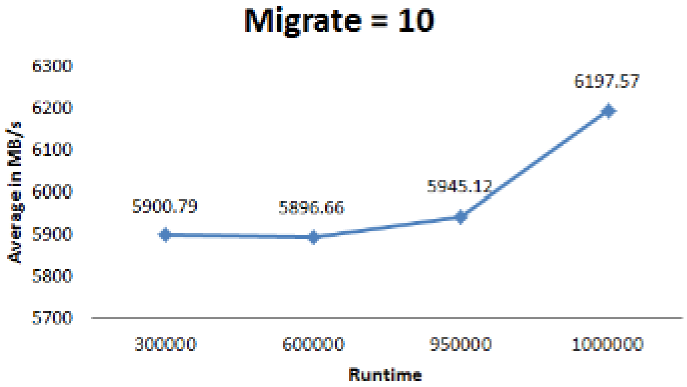
Disadvantages:

- There is no reduction in latency.

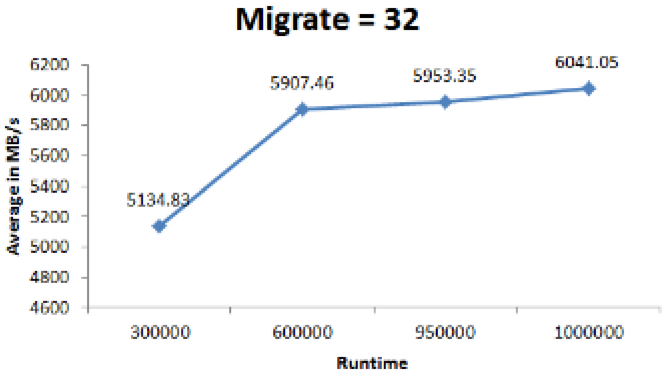
And the software was an Operating System Ubuntu 16.04, with a kernel 4.4.0-31-generic (x86_64).

First, we decided to do the test of the RAM performance doing the “ADD” operation in integers and with different scenarios of the variables, one without changes, one with greater migrate but less runtime, other with greater migrate and greater runtime, etc. We took note of this results to create the different graphs. One thing that really caught our attention was that, if we repeat some of the variables, the result changed quite a bit, and the reason we think this happened was because sometimes, during the tests, we were doing other projects/homeworks in the computer so it may slowed the test, that's why in the result, sometimes the data looks a little bit odd. For the migration, we did the test with 10, 32, 64 and 120; and for the runtime we did the test with 300000, 600000, 950000 and 1000000.

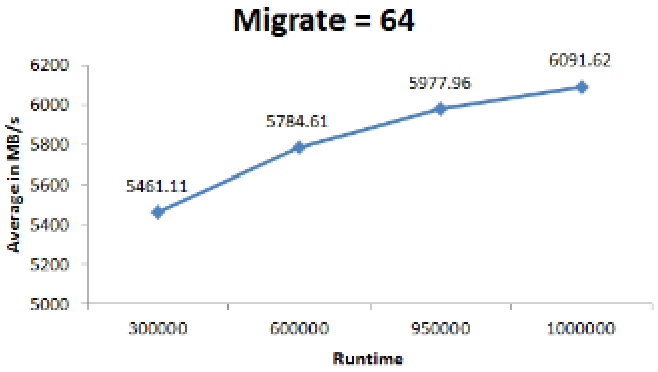
6 Results



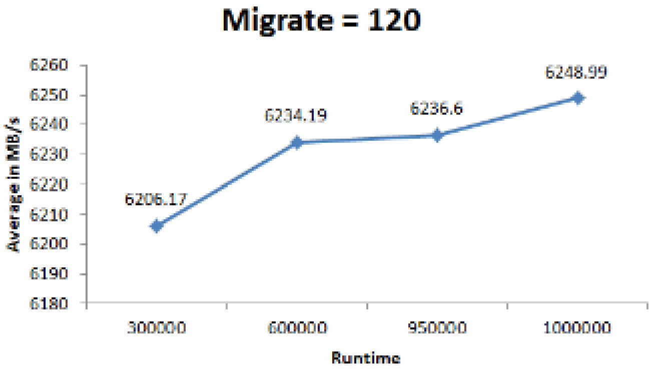
Here we reduce the initial value of the migrate from 32 to 10 and expected that the average time will be reduce.



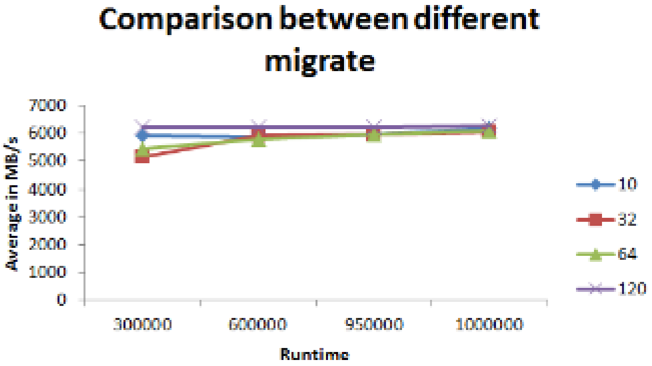
In this one we implemented the initial value of migrate, 32, and comparing it with the first one, the 10, the first one was more consistent that this one, may be because of what we explained in the development section.



Here, the graph looks more similar to the second one than the first one, but has more consistent results because there isn't a BIG jump from one point to another.



This final test was with almost the quadruple of the migrate and what we expected was an increase on the performance since the beginning, and it happened. Even though the test didn't increase a lot with the changes in the runtime, it started at a high performance that the other tests.



Finally, we can say that, if we want to be even more precise with the results of each test, we should do them without doing anything else in the computer because it can affect, not by a lot, but it can. Also, we could observe that the increase in the migrate is the one that affects the most in our tests because it moves more the average time while the runtime affects a little bit.

7 Conclusion

Finally, after doing the test, we can conclude that the performance of the RAM can be affected by the migrate variable in the kernel. The runtime variable didn't affect as much as we thought it would. Migrate as it says in the theoretical framework, boosted the SCHED OTHER threads, but the real-time tasks got worse, having slowdowns in the performance of the computer, this happened a lot during the test with values like 64 and 120. You can't have both at maximum at the same time, there has to be a balanced between or sacrifice the performance of the SCHED OTHER threads for real-time tasks and vice versa.

8 References

Bitesize. (n.d). CPU and Memory. BBC. Retrieved from http://www.bbc.co.uk/education

Computer Memory Upgrade. (2003). Memory and its influence on performance. Computer Memory Upgrade. Retrieved from http://www.computermemoryupgrade.net/

Dell. (2016). How Random Access Memory affects performance. Dell. Retrieved from http://www.dell.com/support/

Hopper, Jenifer. (2015). Untangling memory access measurements. IBM Developer Works. Retrieved from https://www.ibm.com/developerworks/

Suse. (n.d). Completely Fair Scheduler. Suse. Retrieved from https://www.suse.com/documentation/

Williams, Jane. (n.d). How Does Memory Affect Processors?. Chron. Retrieved from http://smallbusiness.chron.com/