# PHYSICAL MACHINE vs VIRTUAL MACHINE for compiling kernel

By
Kartik Gupta

***CONTENTS***

## OBTAINED DATA

## MACHINE 1 : PHYSICAL MACHINE

Specifications:

OS : ubuntu 14.04(updated till date 12/08/16)
RAM : 7.7 GB
SIZE : 20GB
DISK TYPE: SATA
CPUs : 8
File System : ext4

The output obtained on running the script “test.py”

### Test1

```
Line #    Mem usage    Increment   Line Contents
================================================
    16   16.848 MiB    0.000 MiB       @profile
    17                                 def compile_kernel():
    18   16.852 MiB    0.004 MiB         subprocess.call('sudo make clean && make
mrproper',shell=True)
    19   16.781 MiB   -0.070 MiB         subprocess.call('sudo make defconfig',shell=True)
    20   14.477 MiB   -2.305 MiB         subprocess.call('sudo make',shell=True)
```

```
559.29user     // this
79.38system
13:45.16elapsed
77%CPU (0avgtext+0avgdata 208820maxresident)k
412944inputs+720488outputs (904major+34544498minor)
pagefaults 0swaps
```

**Test2**

```
Line #    Mem usage    Increment   Line Contents
================================================
    16   16.840 MiB    0.000 MiB   @profile
    17                             def compile_kernel():
    18   16.844 MiB    0.004 MiB         subprocess.call('sudo make clean && make
mrproper',shell=True)
    19   16.844 MiB    0.000 MiB         subprocess.call('sudo make defconfig',shell=True)
    20   16.844 MiB    0.000 MiB         subprocess.call('sudo make',shell=True)
```

```
481.14user
 68.65system
 9:08.07elapsed
 100%CPU (0avgtext+0avgdata 208368maxresident)k
55584inputs+712064outputs (190major+34218689minor)
pagefaults 0swaps
```

**Test3**

```
Line #    Mem usage    Increment   Line Contents
================================================
    16   16.855 MiB    0.000 MiB   @profile
    17                             def compile_kernel():
    18   16.859 MiB    0.004 MiB         subprocess.call('sudo make clean && make
mrproper',shell=True)
    19   16.859 MiB    0.000 MiB         subprocess.call('sudo make defconfig',shell=True)
    20   16.859 MiB    0.000 MiB         subprocess.call('sudo make',shell=True)
```

```
479.33user
 65.22system
 8:57.26elapsed
 101%CPU (0avgtext+0avgdata 209136maxresident)k
376inputs+708528outputs (0major+34218752minor)
pagefaults 0swaps
```

**Therefore:**
**Average memory consumption is between 16-17MiB**
**Average System time is approximately 70 seconds**

**MACHINE 2: VIRTUAL MACHINE (hosted on open-nebulla)**

Specifications:
OS-Debian
RAM : 4GB
SIZE:4.5GB
DISK TYPE :unknown
CPUs : 4
File System : ext4

Due to time constraints and some difficulty with the virtual machine , I could conduct only 1 test on the virtual machine

Test1

```
Line #  Mem usage     Increment   Line Contents
================================================
     16      9.574 MiB      0.000 MiB   @profile
     17                            def compile_kernel():
     18      9.578 MiB      0.004 MiB       subprocess.call(' make clean && make
mrproper',shell=True)
     19      9.578 MiB      0.000 MiB       subprocess.call(' make defconfig',shell=True)
     20      9.578 MiB      0.000 MiB       subprocess.call(' make',shell=True)
```

```
763.80user
103.47system
15:04.86elapsed
 95%CPU (0avgtext+0avgdata 162432maxresident)k
1464inputs+708424outputs (16major+31244672minor)
pagefaults 0swaps
```

**Therefore:**
**Average memory consumption is between 9-10MiB**
**Average System time is approximately 100 seconds**

**OBSERVATIONS:**

*****It is quite clear that on the virtual machine , the cpu/system spends more time but here the specs of machine in terms of RAM differ by a factor of 2 , so we cannot infer much from it.

*One observations which very clearly visible is that there is no memory increment ocurring when the final make command is being called to compile the kernel . Instead , logic says that memory usage at this point should increase as this is actual command where compiles kernel.Rest everything till now was just a setup for this statement to exectute properly.But memor-profiler is not showing any increase in memory usage.This observation is hard for me to explain as in fact , I myself , am not understanding it. It even suggests that probably the memory profiler might be a broken tool.

*Also , in case of virtual machine , memory usage is coming out to be lesser than that in case of physical machine.This again differs from my logic as it should be more according to me.This leads me to a possibility that the hypervisor might be even abstracting the actual memory for a virtual OS and so the memory usage that we are getting might not be the actual memory usage.

**CONCLUSION**

This experiment has led to more questions than answering some of the earlier ones .
It has led to following conclusions for me:-
1)Memory profiler tool might be broken
2)Hypervisor might be abstracting actual memory usage
3)We can’t conclude that a physical machine is better than a virtual machine