Raspberry Pi benchmark

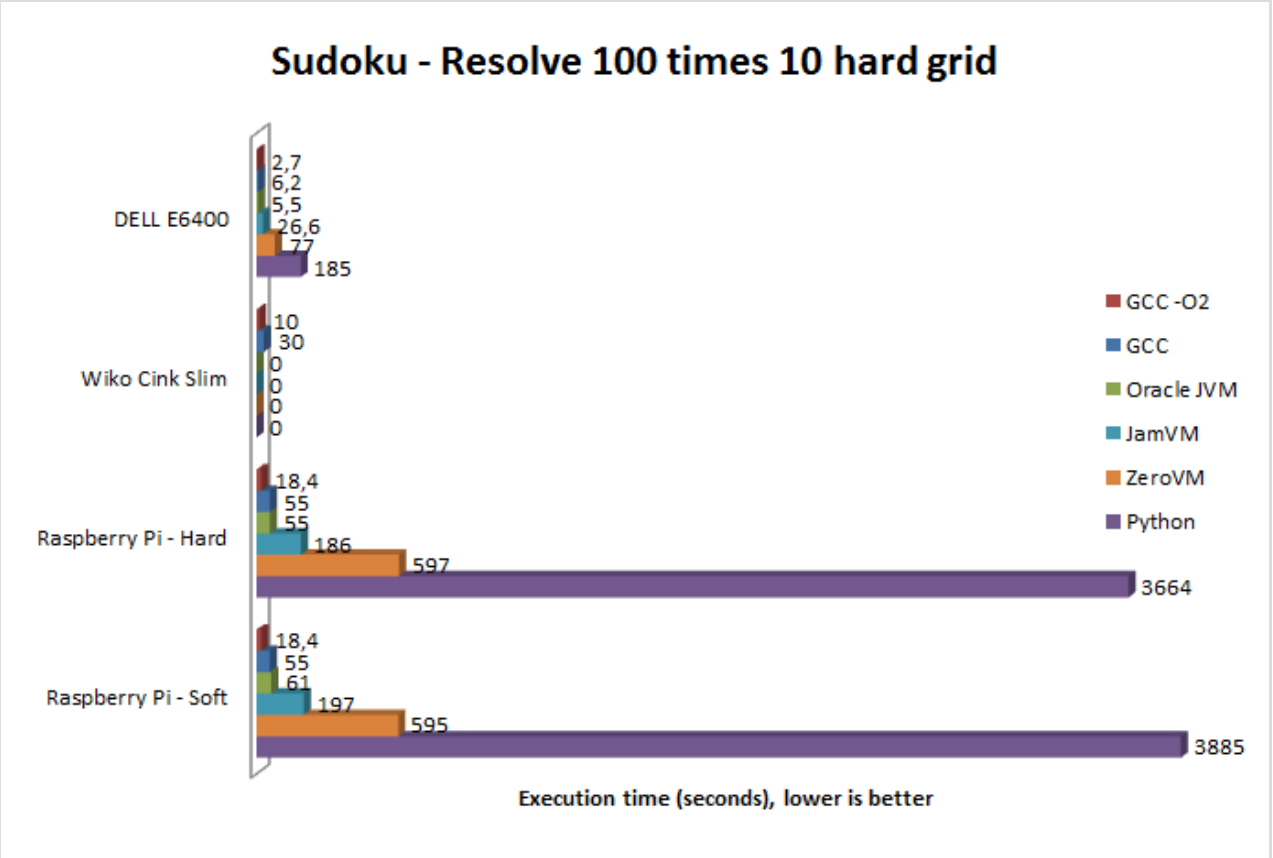
Posted on **12 January 2013**

Last week, I made few bench with Raspberry Pi at work. I wanted to compare hard float and soft float, technologies available, and quickly compare to few other hardware.

With sudoku solving and matrix multiplication code available at http://attractivechaos.github.com/plb/, I made a little modification in the code which is available here. I then tried GCC, Oracle JVM, OpenJDK 7 ZeroVM and JamVM, then Python on these platforms :

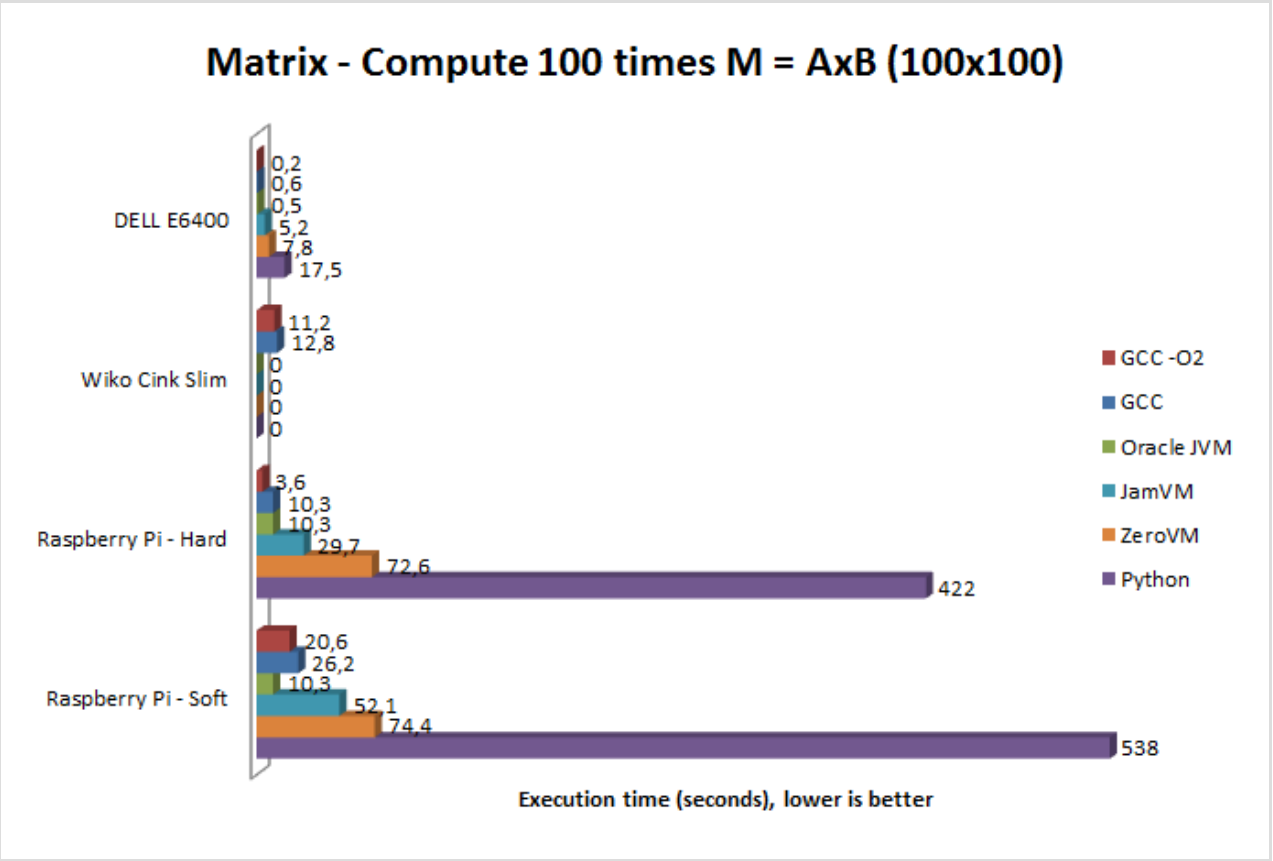
- Raspberry Pi – Single Core ARM 700Mhz – 512Mb
  - Debian wheezy (soft-float) with JDK7 for Oracle JVM
  - Raspbian wheezy (hard-float) with JDK8 for Oracle JVM
- Phone – Wiko Cink Slim – Dual Core ARM 1Ghz – 512Mb
  - Android 4.0
- Laptop – Dell E6400 – Dual Core X86 2.26Ghz – 2Gb
  - Ubuntu 12.10

Technologies comparison



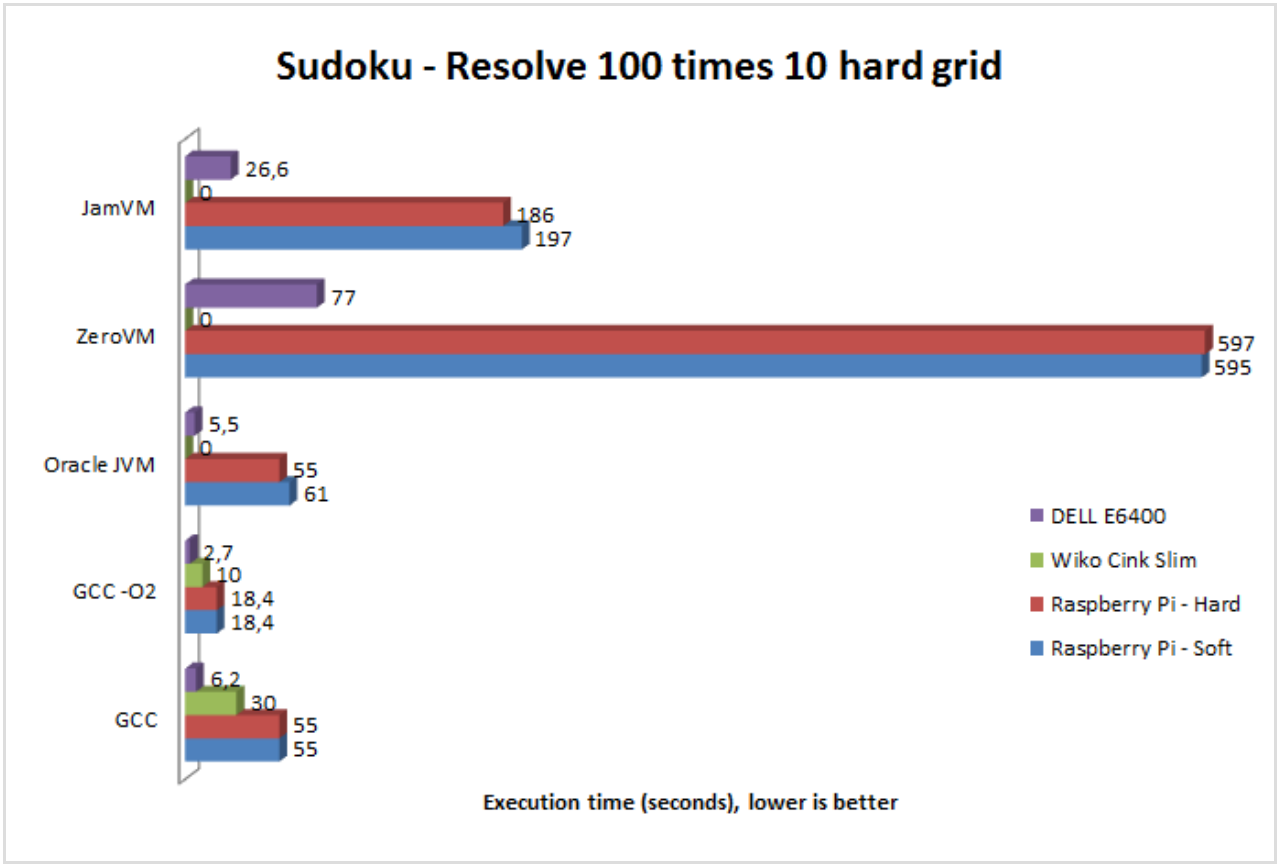
Here we can see that GCC optimized code is faster on Raspberry as expected. Surprising, Oracle JVM is really fast, with the same time as GCC without optimization. Python explodes times. On the laptop, the order is the same but the time increase with Python is not so important. On the laptop, Python is 30x slower than GCC, whereas it's 60x slower on the Raspberry Pi. Is there a Raspberry issue with Python or is it a generally ARM issue ? I was not able to try Python on the phone to get an idea. Phone

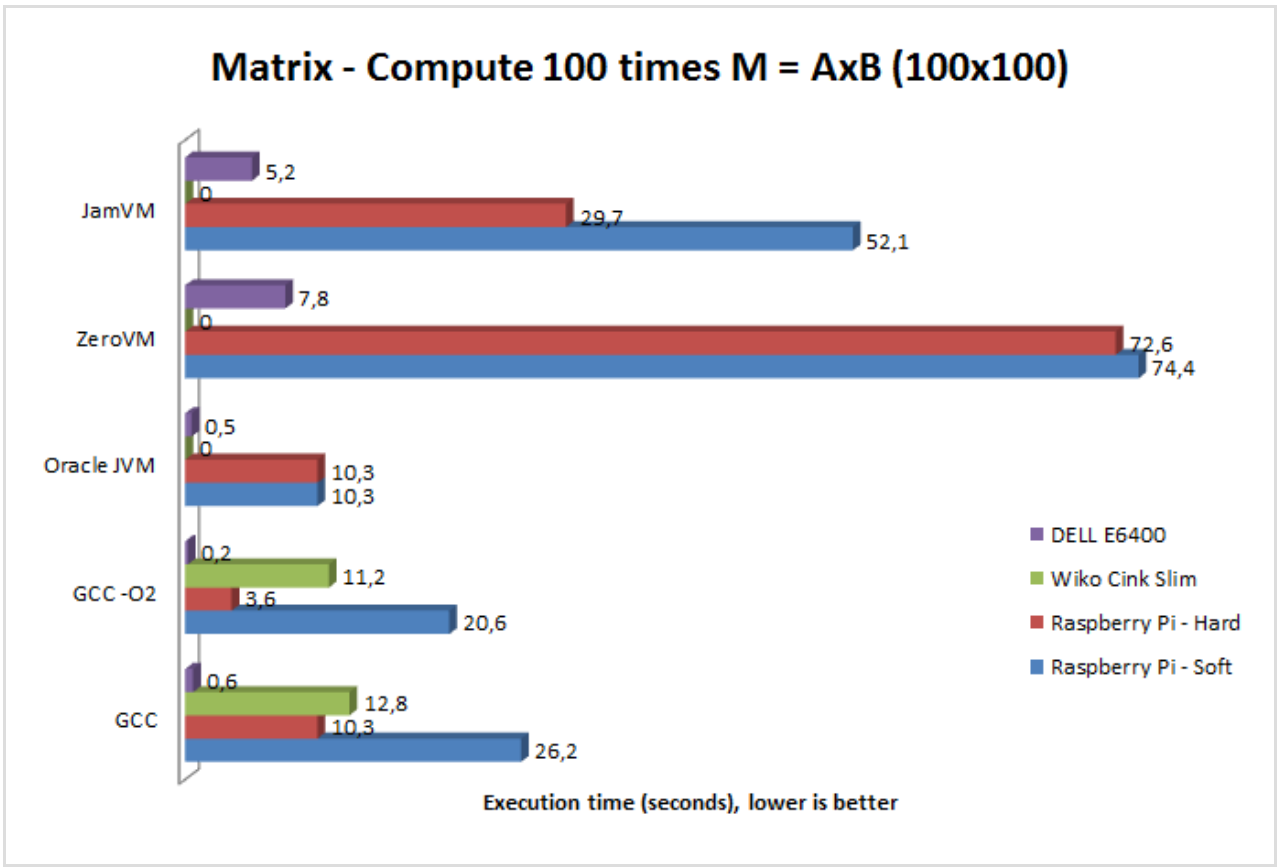
results are not useful here.



With float computation, the orders does not change, except on soft Debian, where the Oracle JVM is faster than GCC. Congrats to Oracle !

Platform comparison





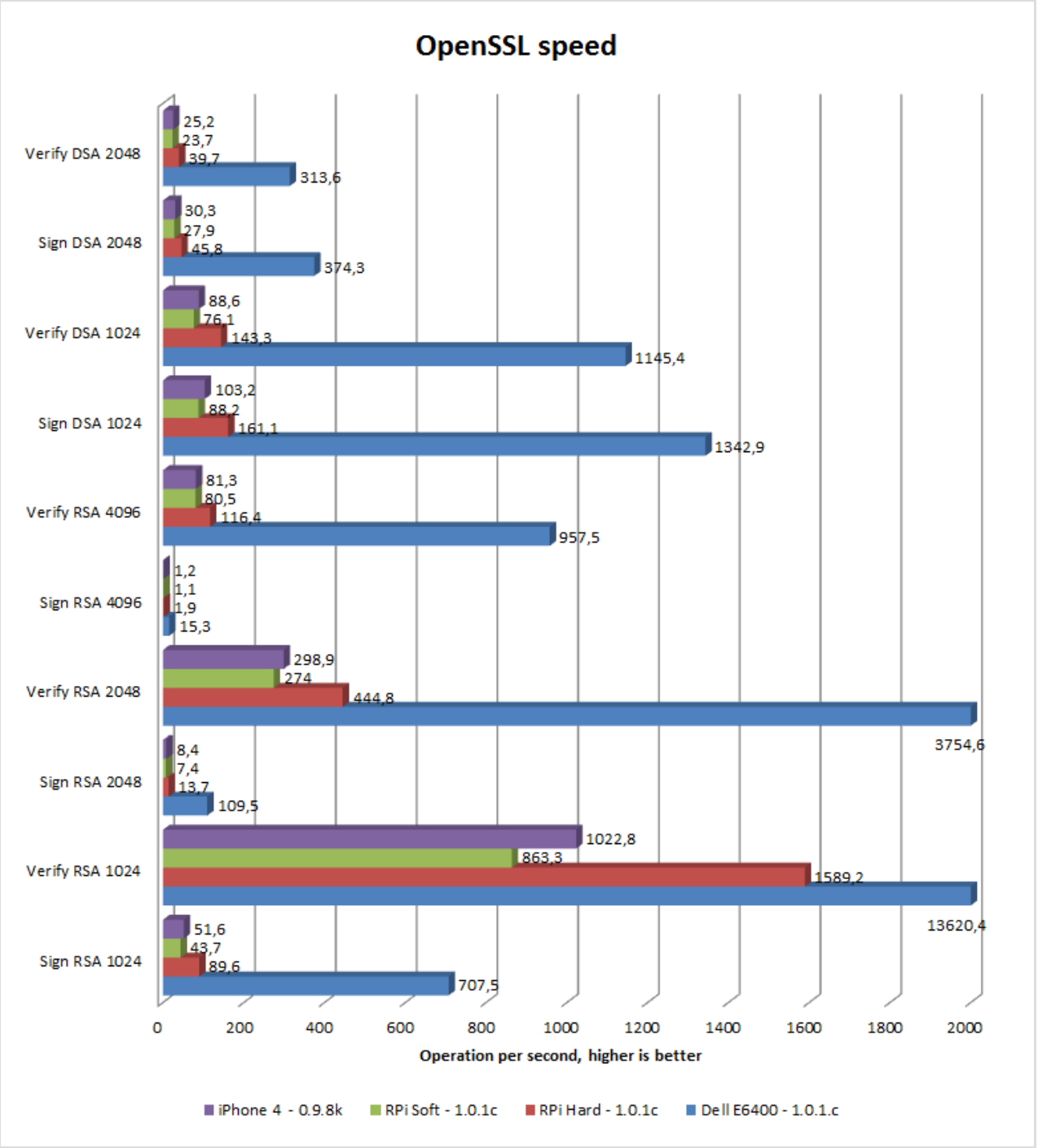
Times given here are the same as before, but grouped by technology to compare each platforms. As we can see, Oracle JVM has the same performance on both hard and soft float Pi.

With pure integer computation, we can see on GCC that time reduction on ARM platforms is proportional to frequency increase. Core count does not matter in this test. But with float computation,

the Rasperry Pi can be faster than the phone, thanks to hardware float support, even with a slower frequency. On the same Raspberry, hard float support increases performance by 2 !

We can also see the difference between x86 and ARM architecture. The laptop CPU has a frequency multiplied by 2.26 compared to phone's one, but the execution is 20 times faster.

OpenSLL speed test



Please take note I have set the graph maximum value to 2000 operations per second, but Verifying with RSA 1024 or 2048 exceed this maximum. As we can see, OpenSSL is two time faster on a hard-float Pi than a soft-float Pi.

Conclusion

First, I have to say that even if Python is a great language and technology, as the syntax is quite simple and it allows various programming paradigm: sequential, functional, object-oriented. But the ARM port is really slow. As Oracle JVM is now available, even on Raspbian with the JDK8 Early Access, and regarding the result, I think Java is the best choice to get best programming ease/performance ratio. Then, the problem is that Raspberry Pi is not enough powered to run a Java IDE like Eclipse. Another computer, a network sharing and a SSH connection on the Pi will help developing in Java for the Pi.

Comparing ARM platforms shown that hard float support is a great thing as the performance increase is really significative and float computation is used almost everywhere, in data compression for instance.

Finally, comparing ARM to x86 architecture is a real proof that frequency does not always matter. I may try with my old 400Mhz Pentium 2, I'm pretty sure the result will be the same as the Raspberry Pi. As no one should use a Pentium 2 to do intensive web surfing and desktop tasks, you may understand why it's a bad idea to use the Raspberry as a desktop computer replacement.

On my side, I decided to port WebIOPi to Java before the 1.0 final release.

External links :

1. a quick performance comparison of java on raspbian
2. Java GPIO Frequency Benchmarks
3. Benchmarking Raspberry Pi GPIO Speed
4. Speed Comparison

This entry was posted in **Non classé** by **trouch**. Bookmark the **permalink [http://trouch.com/2013/01/12/raspberry-pi-benchmark/]** .