

Table 5-1: Numeric Computation Time for 5,000,000 Iterations of the *n*-Body Algorithm on Raspbian (Jessie Minimal Image)

VALUE	TYPE	RPi 3 at 1.2 GHz ¹	RPi 2 at 1 GHz ²	RPi B+ at 1 GHz ³	64-BIT i7 PC ⁴
C/C++	Compiled	1.00 × (6.5s)	1.00 × (9.3s)	1.00 × (10.0s)	1.00 × (0.61s)
C++11	Compiled	1.06 × (6.9s)	0.69 × (6.4s)	0.70 × (7.03s)	0.95 × (0.58s)
Haskell	Compiled	1.16 × (7.6s)	1.17 × (10.8s)	1.07 × (10.8s)	1.15 × (0.70s)
Java ⁵	JIT	1.52 × (9.94s)	1.45 × (13.4s)	2.29 × (23.0s)	1.36 × (0.83s)
Mono C#	JIT	2.72 × (17.8s)	2.47 × (22.9s)	3.62 × (36.4s)	2.16 × (1.32s)
Cython ⁶	Compiled	2.74 × (17.9s)	2.67 × (24.8s)	2.80 × (28.0s)	1.26 × (0.77s)
Node.js ⁷	JIT	2.76 × (18.1s)	6.23 × (57.7s)	50.1 × (503s)	6.54 × (3.99s)
Lua	Interpreted	20.2 × (132s)	21.2 × (197s)	25.7 × (258s)	34.3 × (20.9s)
Cython	Compiled	64.2 × (420s)	66.6 × (618s)	163 × (1633s)	58.0 × (34.4s)
Perl	Interpreted	92.6 × (601s)	81.5 × (756s)	171 × (1716s)	82.0 × (50.0s)
Python	Interpreted	94.1 × (616s)	89.9 × (834s)	198 × (1992s)	89.7 × (54.7s)
Ruby	Interpreted	147 × (962s)	140 × (1298s)	265 × (2662s)	47.4 × (28.9s)

¹ RPi 3 running at 1.2 GHz, quad core (only one core utilized), ARMv7 (rev 4 with a 32-bit Linux distribution: Linux 4.1.19-v7+) supports: half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpaee evtstrm crc32. Please ensure that you use a high-quality power supply that is capable of delivering at least 1.5 A.

² RPi 2 overclocked at 1 GHz, quad core (only one core utilized), ARMv7 (rev 5) supports: half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpaee evtstrm. Note: Overclocking your RPi may reduce its lifespan.

³ RPi B+ overclocked at 1 GHz, single core, ARMv6 (rev 7 v6) supports: half thumb fastmult vfp edsp java tls.

⁴ Windows 8.1 PC running a 64-bit Debian Jessie VirtualBox VM that was allocated 3 threads (of 12) on an Intel i7-5820K @ 3.3 GHz, with the VM allocated 16 GB of RAM. Only one thread is used.

⁵ You can use `sudo apt install oracle-java8-jdk` to install the Oracle JDK on the Raspberry Pi platform.

⁶ This Cython test involved modifying the Python source code to optimize it. It is not simply the compilation of raw Python code. The second Cython test represents the simple compilation of raw Python source code.

⁷ Node.js (`node -v`) is version v5.10.1 and it supports the ARM NEON accelerator processor. NEON is available on the RPi 2/3 (ARMv7) but not on the RPi B+ (ARMv6), which contributes to the poor performance of Node.js on the RPi B+ of 50.1× the baseline. See the feature titled “LAMP and MEAN” in Chapter 12 for instructions on how to install the latest version of Node.js on the RPi.

The code examples have not been optimized for multicore processors, so for example, the C/C++ code only uses a single core of the RPi 3 processor. Albeit, regular Linux threads are automatically offloaded to other cores and the full memory bandwidth is available to the one core. Multicore programming is discussed in the next chapter, where you can see that the performance of the RPi 2/3 can be further improved relative to the RPi B+, which has a single-core processor. All the programs use between 98% and 99% of the CPU while they are executing.

The second column in Table 5-1 indicates the language type, where *compiled* refers to natively compiled languages, *JIT* refers to just-in-time compiled languages,