Simulation times, version 2

An M/M/1 queueing system with arrival rate 0.8, service rate 1.0 and measure rate 0.1 was simulated for 123456789 simulated time units. Five runs were made for each programming language.

In all cases event scheduling was used except for Pascal where process scheduling was used. However, the differences in execution times between event scheduling and process scheduling are generally small.

The table shows the execution time in seconds.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Language** | **Run 1** | **Run 2** | **Rung 3** | **Run 4** | **Run 5** |
| Java | 11.9 | 11.7 | 11.9 | 11.9 | 12.0 |
| Python | 523 | 531 | 492 | 525 | 547 |
| Go | 20.3 | 20.3 | 20.2 | 20.2 | 20.8 |
| Julia | 455 | 459 | 441 | 444 | 459 |
| Pascal | 85,2 | 79.4 | 84.6 | 80.8 | 80.9 |
| C | 29.2 | 29.6 | 29.1 | 29.1 | 29.2 |
| JavaScript | 14.4 | 14.3 | 13.5 | 14.0 | 13.6 |

If the mean of the execution times are normed (with time for Java equal to 1.0) one gets:

|  |  |
| --- | --- |
| **Language** | **Normed time** |
| Java | 1.0 |
| Python | 44.1 |
| Go | 1.7 |
| Julia | 38.0 |
| Pascal | 6.9 |
| C | 2.5 |
| JavaScript | 1.2 |

Surprisingly C is just on fourth place.