An inquiry into the influence of parallelism and the choice of random engines on the runtime and results of Stride

NIELS AERENS, THOMAS AVÉ, TOBIA DE KONINCK, and ROBIN JADOUL

In this article, we investigate a few properties of the Stride project. In particular, we look at the influence of the amount of parallelization with respect to the running time of a simulation and the difference in simulation outcomes when varying the used random number generator.

ACM Reference format:
Niels Aerens, Thomas Avé, Tobia De Koninck, and Robin Jadoul. 2018. An inquiry into the influence of parallelism and the choice of random engines on the runtime and results of Stride. 1, 1, Article 1 (March 2018), 7 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Stride [2] is an individual-based Simulator for the Transmission of Infectious Diseases with focus on model flexibility and performance. The performance of such a simulator is important for the researcher to have fast feedback in order to build a reasonable model of the disease. The Stride program currently has scenario tests to prevent regressions in the k simulation output. To provide fast feedback to the developers working on Stride the test should run sufficiently quickly. The running time of the tests are a good indication of the running time of the simulator itself. Hence we perform a performance analysis of the running time of tests and the influence of parallelism.

The scenario tests are using the Attack Rate as parameter to assert the simulator outcome. Since Stride is a stochastic system and thus relies on randomness, the output of its tests are variable. Therefore the tests use an acceptability range for the test outcome. We determined these ranges by running the tested simulations multiple times (100+). We noticed using a QQ-Plot and hypothesis tests that the Attack Rate can be considered to follow a normal distribution. For the accepted ranges, we decided to allow a distance of 2 standard deviations from the observed mean.

The second topic of this paper is to analyze the influence of the random number generator engine on the Attack Rate. Stride uses the Trng library [1] as random number generator. The inspected engines are `lgc64`, `lgc64_shift`, `mrg2`, `mrg3`, `yarn2` and `yarn3`.

2 METHODS

2.1 Influence of multi-threading

The analysis described in this section was performed on a 32 core AMD machine. Therefore the tests were executed with a maximum of 32 threads, which is a reasonable maximum for any workstation at time of writing. By using a Bash-script[1] the Stride tests were run with an increasing amount of threads, starting at 1 up to 32.

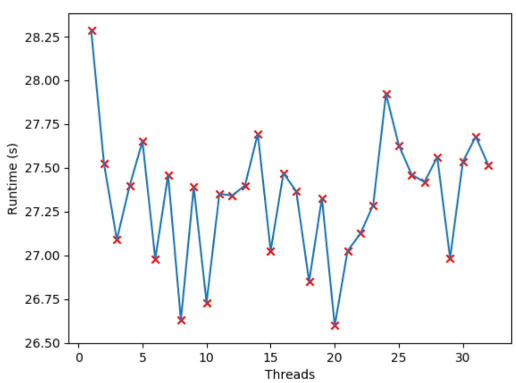
2.2 Influence of random engine

Using a Python3 script[2] the simulation was run using the different random engines, at the same time the running time was measured.

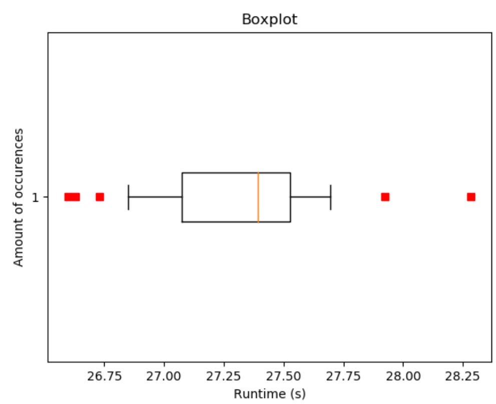
3 RESULTS

3.1 Influence of multi-threading

The running time in function of the number of threads are shown in fig. 1. The difference between the first and third quartile is only 0.45 seconds. This difference is negligible compared to the 27 seconds running time. Even the furthest outliers are less than 2 seconds apart. We can safely conclude that the parallelisation of the code doesn't have a great influence on the performance. There should hopefully be some room for improvement here for the performance of Stride. Note that for each number of threads, the values are obtained by running the Influenza A simulation 15 times and taking the average running time.



(a) Connected scatter plot



(b) Boxplot

Fig. 1. Running time in function of number of threads

3.2 Influence of random engine

The engine which needs the least amount of time is the `lgc64` engine, the `yarn3` engine needs the most time. The difference in time is 60 seconds when running the tests 15 times, which means that for one run the difference is only

[1]https://github.com/LEDfan/Bachelorproef/blob/597e9d14356a40f10c52dbd789ce495f0891720e/assets/src/week3/bench.sh
[2]https://github.com/LEDfan/Bachelorproef/blob/b8b67f7083ba8a523bdb16f0c6fd0fdfeea7dacf/assets/src/week3/random_engines.py

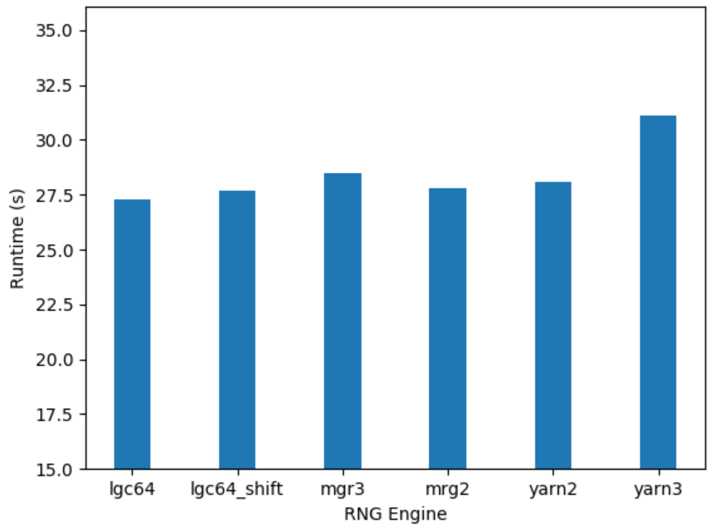


Fig. 2. Running time of the different rng engines for the influenza A test case.

about 6 seconds. Thus there isn't a big difference. A complete comparison of running times can be found in fig. 2. The distribution analysis of the results with different random engines was done with 100 data points for each random engine. While this is still a relatively sample size, much larger sample sizes were not feasible due to the time requirements to run them all.

The first test case of Stride (Influenza A[3]) performs a simulation of influenza in Flanders, with an R_0 value[4] of 3.0. For this test case the range of allowed attack rates doesn't change much. The second test case (Influenza B[5]) uses a seeding rate of 0, which results in a constant attack rate of 0. As expected the RNG engine doesn't have influence on this outcome. The same holds for the third test case (Influenza C[6]) which uses very low seeding rate and a large immunity factor. The attack vector for every engine is 5. The Measles 16[7] testcase is more interesting for the analysis of the RNG engines. The disease simulated is not influeanza but the measles. The R_0 value is set to 16, which results in a very large part of the population becoming infected. The last testcase (Measles 60) sets the R_0 value to 60. It infects the whole population, so the attack rate is for all the engines the same.

3.3 Normality of the output

To validate our assumption that the distribution of the attack rate was normal for our tests, we both looked at QQ-plots and histograms of the outputs of different runs against a reference normal distribution with $\mu = \bar{X}, \sigma = S_X$, verified that histograms had a more or less normal look to them, and verified with the Shapiro-Wilkes test.

[3]https://github.com/LEDfan/Bachelorproef/blob/dd1ef48867238d62446813f47d6718908505b7f2/test/cpp/gtester/BatchRuns.cpp#L82
[4]Basic reproduction number, a measure for the infectiousness of a disease: the amount of people will 1 infected person infect directly in a completely susceptible population.
[5]https://github.com/LEDfan/Bachelorproef/blob/dd1ef48867238d62446813f47d6718908505b7f2/test/cpp/gtester/BatchRuns.cpp#L86
[6]https://github.com/LEDfan/Bachelorproef/blob/dd1ef48867238d62446813f47d6718908505b7f2/test/cpp/gtester/BatchRuns.cpp#L91
[7]https://github.com/LEDfan/Bachelorproef/blob/dd1ef48867238d62446813f47d6718908505b7f2/test/cpp/gtester/BatchRuns.cpp#L97

Looking at the Influenza A testcase, we find the resulting P-values for Shapiro-Wilkes test in table 1. Here, all random engines result in an (at least approximately) normal distribution for the attack rate.

lgc64	**lgc64_shift**	**mrg2**	**mrg3**	**yarn2**	**yarn3**
0.8719	0.4609	0.9385	0.6556	0.7254	0.7089

Table 1. P-values of the Influenza A test case for the Shapiro-Wilkes normality test

For the Measles 16 testcase, resulting P-values for the Shapiro-Wilkes test can be found in table 2.

lgc64	**lgc64_shift**	**mrg2**	**mrg3**	**yarn2**	**yarn3**
0.8231	0.4216	0.9115	0.4199	0.1473	0.4431

Table 2. P-values of the Measles 16 test case for the Shapiro-Wilkes normality test

We can consider the attack rate to be normally distributed in all presented cases.

A few of the plots can be seen in section 3.3, and for those who would like to see more of them, they can be generated with the script and datafiles found at TODOTODOTODO.

3.4 Equality of acceptable ranges for the test cases

In this section the equality of the different value ranges which are accepted by the tests for each random engine is verified. Only the Influenza A and Measles 16 test cases are studied since they provide have a variable range. The tests are done using a Levene test for the standard deviations and a t-test for related samples for the means.

The hypotheses for the Levene test:

$$H_0 : \sigma_1 = \sigma_2$$

$$H_1 : \sigma_1 \neq \sigma_2$$

and for the t-test:

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2.$$

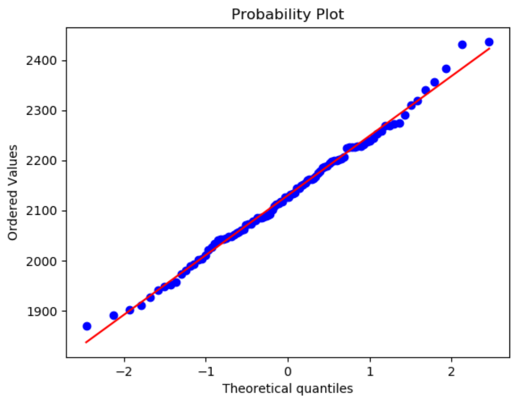
The p-values for these tests are listed in table 4 and table 5. They clearly indicates that the H_0 should not be rejected for both tests on a confidence level of $\alpha = 0.05$. Consequently, we can fairly confidently conclude that the distribution of the attack rate does not differ significantly when using a different random engine.

4 FUTURE WORK

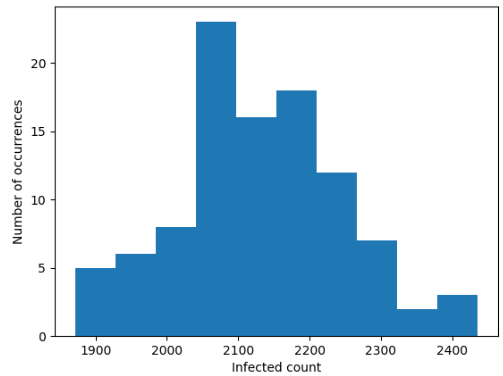
With respect to the analysis performed in this paper, it would be interesting to see an investigation of the influence of threading on the attack rate, or another representative metric of the output of the Stride simulator. It is clear that we hope and expect this effect to be minimal to nonexistent, but it is a good sanity check of the Stride codebase that this would be verified.

5 POSTSCRIPTUM

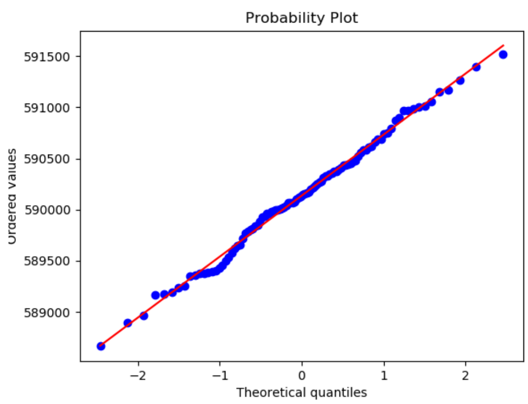
This analysis was done using commit `5563ebd337ae75ed35b739307afe03bdb5219325` of the LEDfan/Bachelorproef repository, corresponding to the `fde6c507e8596bde6d632f6759c9e65f4b878a68` commit in the upstream broeckho/stride repository.



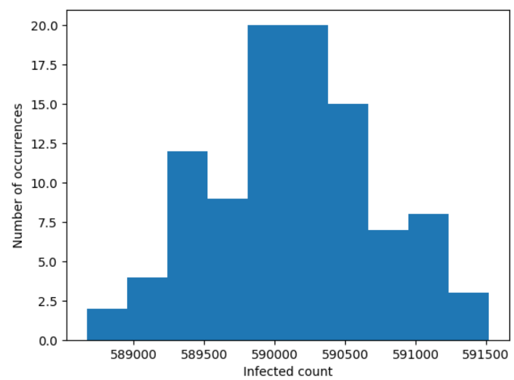
(a) QQ plot of Influenza A results with `lgc64`



(b) Histogram of Influenza A results with `lgc64`



(c) QQ plot of Measles 16 results with `mrg2`



(d) Histogram of Measles 16 results with `mrg2`

Fig. 3. Visually evaluating the normality of the attack rate

	Influenza A	**Measles 16**
`lgc64`	[1896, 2364]	[588959, 591317]
`lgc64_shift`	[1893, 2372]	[588992, 591360]
`mrg2`	[1899, 2334]	[588967, 591307]
`mrg3`	[1891, 2345]	[589012, 591268]
`yarn2`	[1879, 2331]	[588836, 591615]
`yarn3`	[1902, 2331]	[588751, 591443]

Table 3. The acceptable attack rates for the Influenza A en Measles 16 test case for every random number engine

	lgc64		**lgc64_shift**		**mrg2**		**mrg3**		**yarn2**	
	p_μ	p_σ	p_μ	p_σ	p_μ	p_σ	p_μ	p_σ	p_μ	p_σ
lgc64_shift	0.8819	0.7593								
mrg2	0.2797	0.3763	0.3359	0.2392						
mrg3	0.3942	0.7113	0.3958	0.5026	0.8842	0.6079				
yarn2	0.1507	0.5990	0.1432	0.4116	0.4590	0.7271	0.4339	0.8737		
yarn3	0.3578	0.4295	0.3351	0.2750	0.9843	0.9042	0.9017	0.6833	0.4517	0.8102

Table 4. P-values of the Influenza A test case

	lgc64		**lgc64_shift**		**mrg2**		**mrg3**		**yarn2**	
	p_μ	p_σ	p_μ	p_σ	p_μ	p_σ	p_μ	p_σ	p_μ	p_σ
lgc64_shift	0.2164	0.8855								
mrg2	0.9754	0.8673	0.2264	0.9838						
mrg3	0.9630	0.9900	0.2671	0.8882	0.9373	0.8687				
yarn2	0.3326	0.2670	0.5949	0.2232	0.3163	0.2107	0.3081	0.2400		
yarn3	0.6517	0.5366	0.3611	0.4624	0.6508	0.4457	0.6282	0.5097	0.2195	0.6542

Table 5. P-values of the Measles 16 test case

REFERENCES

[1] Heiko Bauke. 2015. Tina's Random Number Generator Library. (2015).

[2] Elise Kuylen, Sean Stijven, Jan Broeckhove, and Lander Willem. 2017. Social Contact Patterns in an Individual-based Simulator for the Transmission of Infectious Diseases (Stride). *Procedia Computer Science* 108 (2017), 2438 – 2442. https://doi.org/10.1016/j.procs.2017.05.086 International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.