

# 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.

---

## 1 INTRODUCTION

Stride [?] 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 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 [?] as random number generator. The inspected engines are lgc64, lgc64\_shift, mrg2, mrg3, yarn2 and yarn3.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© YYYY ACM. Manuscript submitted to ACM

DOI: 10.1145/nnnnnnnn.nnnnnnn

Manuscript submitted to ACM

## 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<sup>1</sup> 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<sup>2</sup> 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. We can conclude that the parallelisation of the code doesn't have a great influence on the performance. 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.

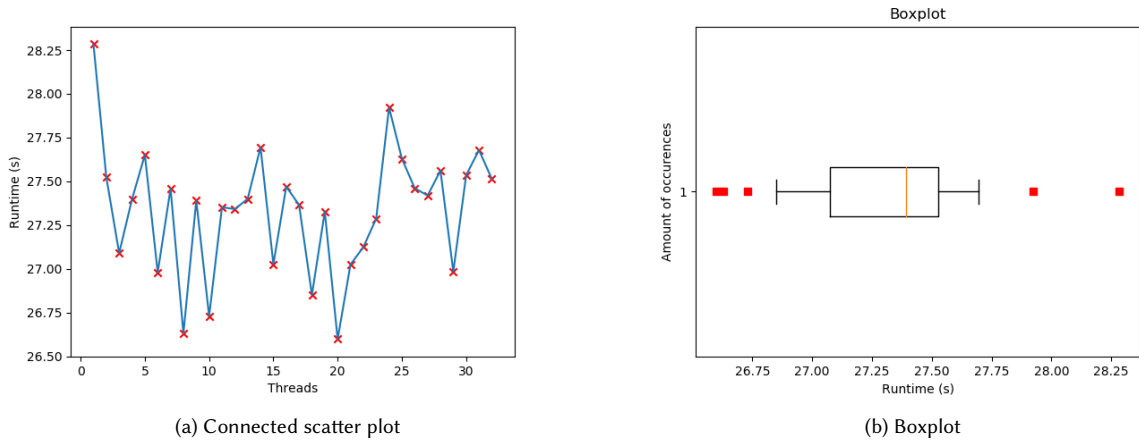


Fig. 1. Running time in function of number of threads

### 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 about 6 seconds. Thus there isn't a big difference. A complete comparison of running times can be found in fig. 2.

<sup>1</sup><https://github.com/LEDfan/Bachelorproef/blob/597e9d14356a40f10c52dbd789ce495f0891720e/assets/src/week3/bench.sh>

<sup>2</sup>[https://github.com/LEDfan/Bachelorproef/blob/b8b67f7083ba8a523bdb16f0c6fd0fdfeea7dacf/assets/src/week3/random\\_engines.py](https://github.com/LEDfan/Bachelorproef/blob/b8b67f7083ba8a523bdb16f0c6fd0fdfeea7dacf/assets/src/week3/random_engines.py)

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

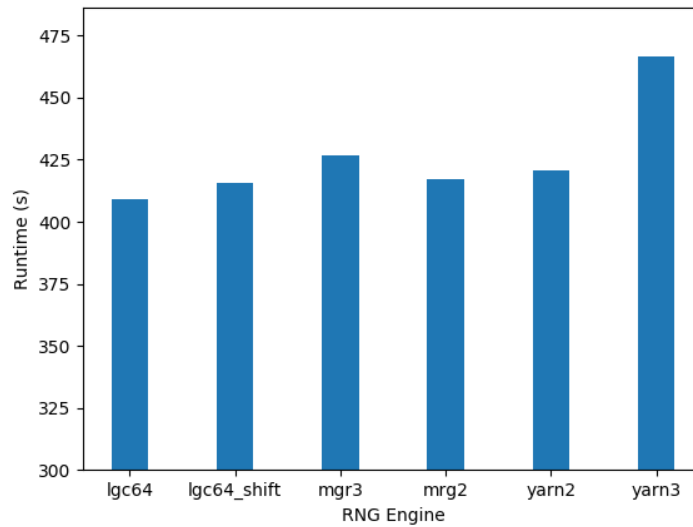


Fig. 2. Running time of the different rng engines for the influenza A test case.

The first test case of Stride (Influenza A<sup>3</sup>) performs a simulation of influenza in Flanders, with an  $R_0$  value<sup>4</sup> of 3.0. For this test case the range of allowed attack rates doesn't change much. The second test case (Influenza B<sup>5</sup>) 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<sup>6</sup>) which uses very low seeding rate and a large immunity factor. The attack vector for every engine is 5. The Measles 16<sup>7</sup> testcase is more interesting for the analysis of the RNG engines. The disease simulated is not influenza 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 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.

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

<sup>3</sup><https://github.com/LEDfan/Bachelorproef/blob/dd1ef48867238d62446813f47d6718908505b7f2/test/cpp/gtester/BatchRuns.cpp#L82>

<sup>4</sup>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.

<sup>5</sup><https://github.com/LEDfan/Bachelorproef/blob/dd1ef48867238d62446813f47d6718908505b7f2/test/cpp/gtester/BatchRuns.cpp#L86>

<sup>6</sup><https://github.com/LEDfan/Bachelorproef/blob/dd1ef48867238d62446813f47d6718908505b7f2/test/cpp/gtester/BatchRuns.cpp#L91>

<sup>7</sup><https://github.com/LEDfan/Bachelorproef/blob/dd1ef48867238d62446813f47d6718908505b7f2/test/cpp/gtester/BatchRuns.cpp#L97>

lgc64	lgc64_shift	mrg2	mrg3	yarn2	yarn3
0.2530	0.8561	0.2717	0.3722	0.4837	0.6319

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 ?? . We can observe that only the lgc64 engine fails the test at  $p = 0.05$ .

lgc64	lgc64_shift	mrg2	mrg3	yarn2	yarn3
$1.039 \times 10^{-7}$	0.5460	0.1423	0.7592	0.2702	0.1516

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

### 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 t-test for the standard deviations and z-test for the means, since it's known that the data is normally distributed.

The hypotheses for the t-test:

$$H_0 : \sigma = \sigma_2$$

$$H_1 : \sigma \neq \sigma_2$$

and for the z-test:

$$H_0 : \mu_1 = \mu_2$$

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

The p-values for these tests are listed in ?? and clearly indicates that the  $H_0$  holds for both tests.

	Influenza A	Measles 16
lgc64	[1855, 2279]	[588591, 591128]
lgc64_shift	[1954, 2226]	[588899, 591513]
mrg2	[1859, 2415]	[589230, 591369]
mrg3	[1985, 2329]	[589401, 591198]
yarn2	[1876, 2331]	[588739, 591488]
yarn3	[1955, 2300]	[589066, 590965]

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

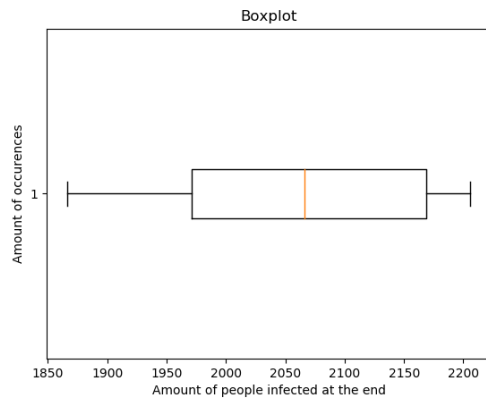
5 An inquiry into the influence of parallelism and the choice of random engines on the runtime and results of Stride

	<b>lgc64_shift</b>	<b>mrg2</b>	<b>mrg3</b>	<b>yarn2</b>
<b>mrg2</b>	$p_\mu = 0.2337$ $p_\sigma = 0.2217$			
<b>mrg3</b>	$p_\mu = 0.01715$ $p_\sigma = 0.0234$	$p_\mu = 0.6365$ $p_\sigma = 0.703$		
<b>yarn2</b>	$p_\mu = 0.6767$ $p_\sigma = 0.6782$	$p_\mu = 0.4729$ $p_\sigma = 0.4849$	$p_\mu = 0.1485$ $p_\sigma = 0.1613$	
<b>yarn3</b>	$p_\mu = 0.1798$ $p_\sigma = 0.3083$	$p_\mu = 0.8226$ $p_\sigma = 0.5178$	$p_\mu = 0.3502$ $p_\sigma = 0.4936$	$p_\mu = 0.5187$ $p_\sigma = 0.5178$

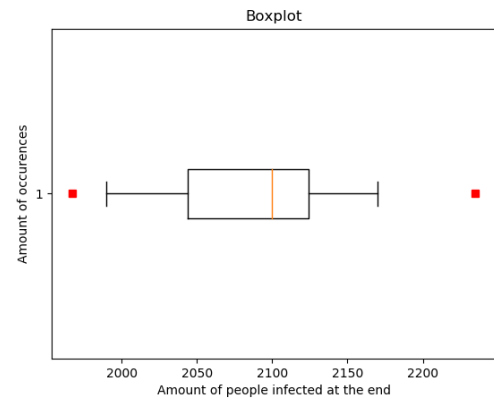
Table 4. P-values of the Influenza A test case

	<b>lgc64_shift</b>	<b>mrg2</b>	<b>mrg3</b>	<b>yarn2</b>
<b>mrg2</b>	$p_\mu = 0.6677$ $p_\sigma = 0.7113$			
<b>mrg3</b>	$p_\mu = 0.6488$ $p_\sigma = 0.6813$	$p_\mu = 0.9985$ $p_\sigma = 0.9983$		
<b>yarn2</b>	$p_\mu = 0.7061$ $p_\sigma = 0.7513$	$p_\mu = 0.4081$ $p_\sigma = 0.3292$	$p_\mu = 0.3812$ $p_\sigma = 0.3917$	
<b>yarn3</b>	$p_\mu = 0.3615$ $p_\sigma = 0.3908$	$p_\mu = 0.124$ $p_\sigma = 0.1112$	$p_\mu = 0.0929$ $p_\sigma = 0.1258$	$p_\mu = 0.6495$ $p_\sigma = 0.65$

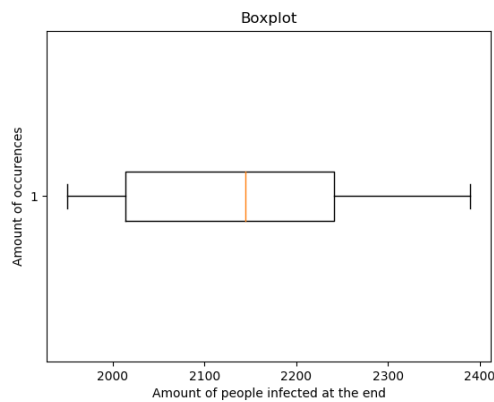
Table 5. P-values of the Measles 16 test case



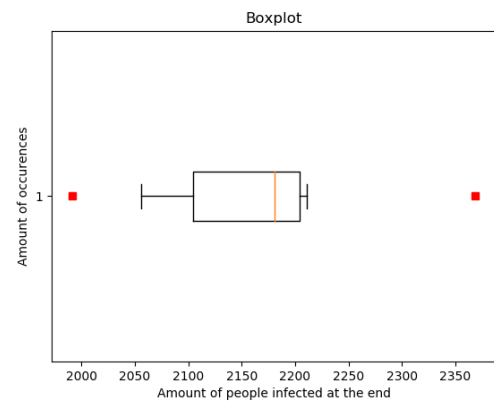
(a) Boxplot of Influenza A run with lgc64



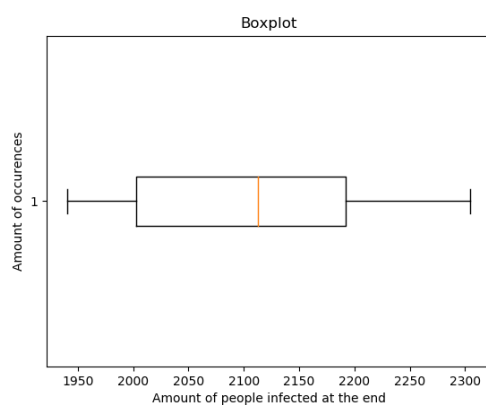
(b) Boxplot of Influenza A run with lgc64.shift



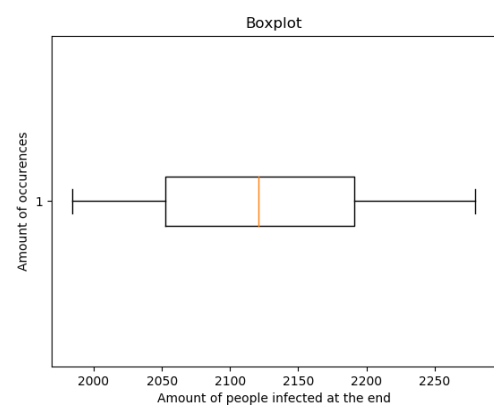
(c) Boxplot of Influenza A run with mrg2



(d) Boxplot of Influenza A run with mrg3



(e) Boxplot of Influenza A run with yarn2



(f) Boxplot of Influenza A run with yarn3

Fig. 3. Running time in function of number of threads