1 2 9 0

UNIVERSIDADE Ð COIMBRA

Process Scheduling Algorithms' Workload Design, Simulation and Analysis with Inferential Statistics

Experimental Methods in Computers Science
Department of Informatics Engineering
College of Science and Technology
Universidade de Coimbra

Oleksandr Yakovlyev 2015231448
Achilles S. Do Nascimento 2017206098
Pedro F. Durão B. L. Félix 2017276005

Method

In terms of methodology for this work, having already gotten some insight into the construction of workloads and the different implications that these might infer to the different schedulers, we started by reading the theoretical slides made available about the topic in question. Having the slides read and with the help of some extra bibliography online we managed to get the bases of the libraries and functions needed to fulfill the different tests.

Following this section we will mention the way that we issued our hypothesis while taking some basis of our past work as well as some feedback made from the professor.

Hypothesis Taken

For this assignment we reformulated our hypothesis, according to the received feedback. Firstly, we changed the performance criteria: instead of focusing solely on tat and ready_wait times, we used

- turnaround_time per cpu_bursts_time: turnaround time normalized for cpu_burst time.
- ready_wait_time per number of bursts : this measure is normalised in respect to the number of bursts that a given process has, and lessens the impact of randomness. We then compare the averages for each process. A lower value corresponds to a more responsive system.
- cpu_usage: this metric is calculated for the whole simulation (per scheduler). A higher cpu usage means that the workload is completed faster and the scheduler is more efficient.
-

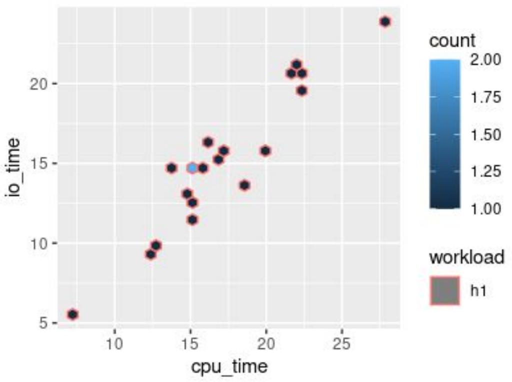
Hypothesis 1:

The average turnaround_time/cpu_burst_time (tat_per_cpu) will be:

- Different between SRTF and FCFS
- Similar between SRTF and SJF

The workload used is "h1" (all data files can be found in the github repository and in the drive folder linked at the end). This is a small but cpu intensive workload. We generated 35 seeds, using the same bash scripts used in the first part of the assignment.

A quick overview of the workload is displayed below:



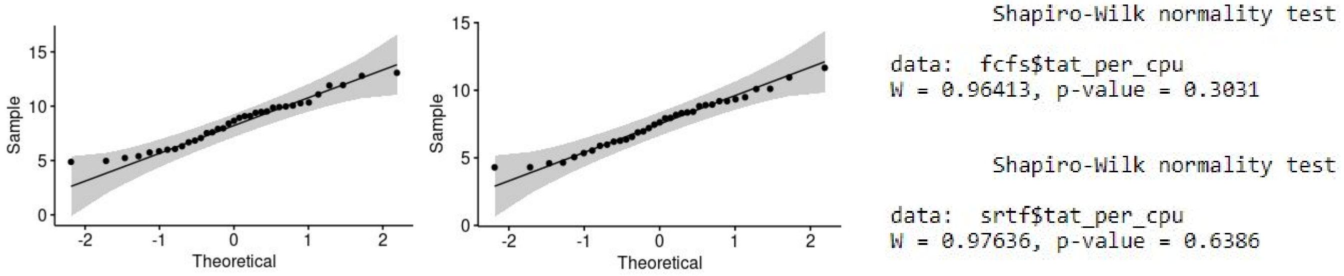
scheduler	usage %	total_cpu_burst	time
FCFS	0.981	341	347
RR	0.98	341	348
SJF	0.956	341	356
SRTF	0.956	341	356

Figure 1 and Table 1. Visualization and description of the "h1" workload, seed = 1

To test our hypothesis we will make two paired t-tests. The first one can be written as:

$$\begin{aligned}
&u_0 = SRTF(AVG(tat\_per\_cpu)) \\
&u_1 = FCFS(AVG(tat\_per\_cpu)) \\
&H_0 : u_0 = u_1 \\
&H_1 : u_0 \neq u_1
\end{aligned}$$

In our data, each sample will correspond to a simulation with a specific seed and scheduler combination. Even though the number of samples is > 30 and, by the Central Limit Theorem we can assume a normal distribution, we still tested normality with the Shapiro-Wilk test:



```
          Shapiro-Wilk normality test

data:  fcfs$tat_per_cpu
W = 0.96413, p-value = 0.3031

          Shapiro-Wilk normality test

data:  srtf$tat_per_cpu
W = 0.97636, p-value = 0.6386
```

Figure 2,3,4. QQ-plot visualization and R console results for the FCFS and SRTF tat_per_cpu distribution, respectively

The t-test gave us a p-value = 1.1618e-07, which is significantly lower that 0.05 leading us to reject H0 and accept H1, and conclude that SRTF and FCFS do have meaningful tat_per_cpu differences (mean of the differences as reported by the R t.test function is approximately -0.93).

Our second part of the hypothesis is formulated as follows:

$$\begin{aligned}
&u_0 = SRTF(AVG(tat\_per\_cpu)) \\
&u_1 = SJF(AVG(tat\_per\_cpu)) \\
&H_0 : u_0 = u_1 \\
&H_1 : u_0 \neq u_1
\end{aligned}$$

And the paired t-test tests yielded a p-value = 0.4928 > 0.05 meaning that we retain H0 and reject H1, or by other words, that SRTF and SJF do not have significantly different tat_per_cpu times, at a 95% confidence level.
This is in accordance with our initial exploratory data analysis findings, maybe even obvious and not a particularly interesting find.

FCFS will often have high overall turnaround times because many small processes will often have to wait at least as long as the longest process that was scheduled before.

By definition SRTF and SJF will prioritise the completion of short processes reducing their turnaround time. Under this workload SJF and SRTF will not display significant differences because most of the time the order in which the processes are run will be the same.

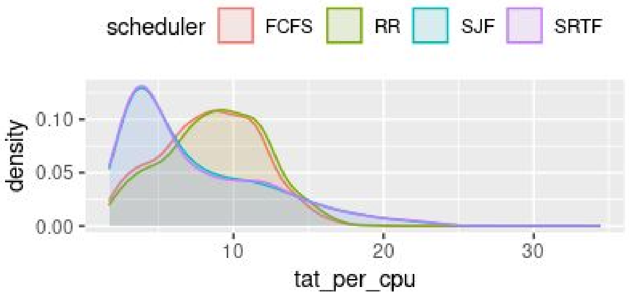


Figure 5. Density graph of tat_per_cpu across the different schedulers

Hypothesis 2:

We think that, for a mixed workload with multiple short bursts processes and some heavy processes, the scheduling algorithms that prioritize short processes will be much more responsive. To measure responsiveness we will use the average of ready_wait_time/burst.

Formally, we have:

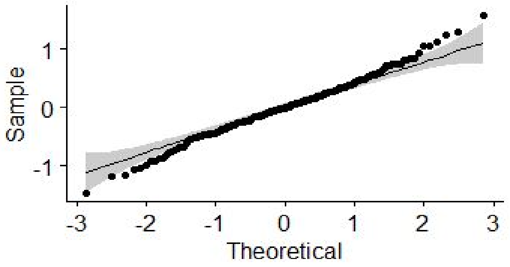
$$u1 = AVG(FCFS(readywaittime/burst))$$
$$u2 = AVG(RR(readywaittime/burst))$$
$$u3 = AVG(SJF(readywaittime/burst))$$
$$u4 = AVG(SRTF(readywaittime/burst))$$
$$H_0 = u1 = u2 = u3 = u4$$
$$H_1 : \exists i, j : ui \neq uj$$

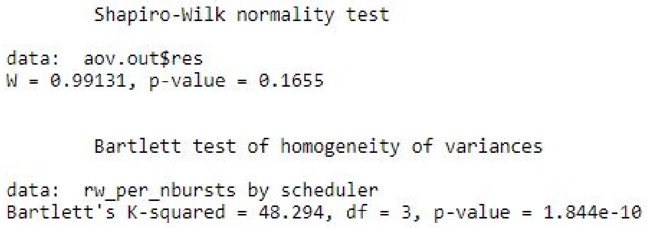
To test our hypothesis we will be making an One-Way ANOVA test.

The workload used was “mixed_h2_240”. In this dataset we merged two different workloads, “long_solo” and “short_burst”, in order to create a workload where there would be a great amount of short burst processes intertwined with heavy processes.

Files	num_procs	mean_io_bursts	mean_iat	min_CPU	max_CPU	min_IO	max_IO
long_solo	5	5	10	10	20	10	20
short_burst	100	100	10	0.1	1	1	2

Table 2. Description of “long_solo” and “short_burst” workloads





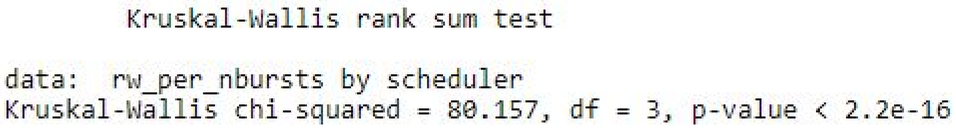
```
         Shapiro-Wilk normality test

data:  aov.out$res
W = 0.99131, p-value = 0.1655


         Bartlett test of homogeneity of variances

data:  rw_per_nbursts by scheduler
Bartlett's K-squared = 48.294, df = 3, p-value = 1.844e-10
```

Figure 6,7,8. QQ-plot visualization and R console results for the ANOVA residuals, normality test and homogeneity test, respectively

For our case each sample per scheduler will be a different seed from a different simulation. We utilized 120 seeds being that each scheduler has 30 to his name different from the others. Being that this is an One-Way ANOVA we started by testing the normality of the samples with Shapiro-Wilk test and the homogeneity of the same with the Bartlett test.

As we can see despite the workload having passed the normality test with a p-value of 0.1655 we cannot proceed with the ANOVA test due to the fact that the samples do not pass the Bartlett homogeneity test with a p-value of 1.844e-10.

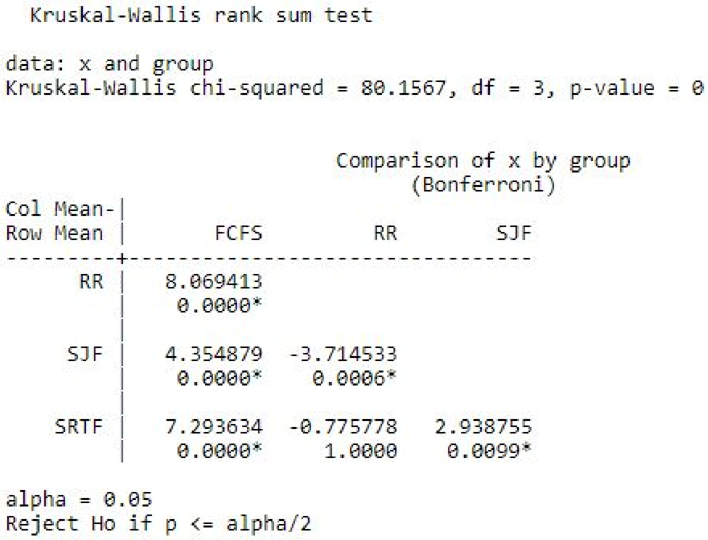
Having this conclusion taken we parted to verify the non-parametric non-parametric Kruskal-Wallis rank sum test:



```
        Kruskal-Wallis rank sum test

data:  rw_per_nbursts by scheduler
Kruskal-Wallis chi-squared = 80.157, df = 3, p-value < 2.2e-16
```

Figure 9. Console results of Kruskal-Wallis rank sum test

As the P-value is lesser than 0.05, we reject the null hypothesis, meaning that at least two schedulers have different ready wait time/ number of bursts. Passing then to the Post-hoc analysis done with the Dunn's test:



```
  Kruskal-Wallis rank sum test

data: x and group
Kruskal-Wallis chi-squared = 80.1567, df = 3, p-value = 0


                           Comparison of x by group
                                (Bonferroni)
Col Mean-|
Row Mean |       FCFS         RR        SJF
---------+---------------------------------
      RR |   8.069413
         |    0.0000*
         |
     SJF |   4.354879  -3.714533
         |    0.0000*    0.0006*
         |
    SRTF |   7.293634  -0.775778   2.938755
         |    0.0000*     1.0000    0.0099*

alpha = 0.05
Reject Ho if p <= alpha/2
```

Figure 10. Console results of Dunn's test

From these tests we conclude that there are significant differences between all the schedulers except RR and SRTF.

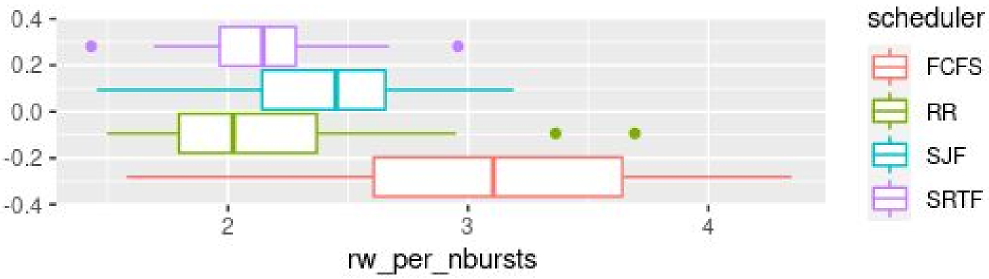


Figure 11. R output when testing the data for outliers

Hypothesis 3:

We believe that, for a system with mixed short and long processes, SRTF will have an overall higher CPU usage. On the other hand, schedulers like FCFS will take longer to complete the workload, having less cpu usage, because the processes will accumulate in the ready wait queue while the io queue is empty.

To test this hypothesis we will use a paired T-test. Formally, we have:

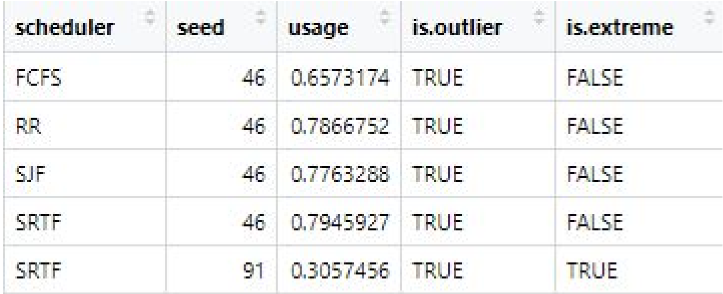
$$H_0 = usage(SRTF) = usage(FCFS)$$
$$H_1 = usage(SRTF) > usage(FCFS)$$

On the right we can see a quick overview of the workload:

As we can see, in this workload we too have two distinct types of processes. We ran this workload for 120 different seeds, so our sample size is > 30 and we can assume normality. But still tested normality and noticed something unexpected. The Shapiro-Wilk test for the SRTF gave us a p-value < 2.2e-16 (contrasting with the FCFS p-value = 0.4708). Upon visual inspection of the qq plots we found the cause: an outlier.

This is also visible on the boxplots, not shown in this report, but can be easily generated by running the R notebooks present in the submission. After further inspection, we found out that this low value was caused by a crash of the simulator. This could have been because of our workload merging script, but due to time constraints we did not investigate further and simply removed the problematic sample and ensured that the remaining samples were correct.

After the correction both schedulers passed the Shapiro-Wilk test.



scheduler	seed	usage	is.outlier	is.extreme
FCFS	46	0.6573174	TRUE	FALSE
RR	46	0.7866752	TRUE	FALSE
SJF	46	0.7763288	TRUE	FALSE
SRTF	46	0.7945927	TRUE	FALSE
SRTF	91	0.3057456	TRUE	TRUE

Figure 12. R output when testing the data for outliers.

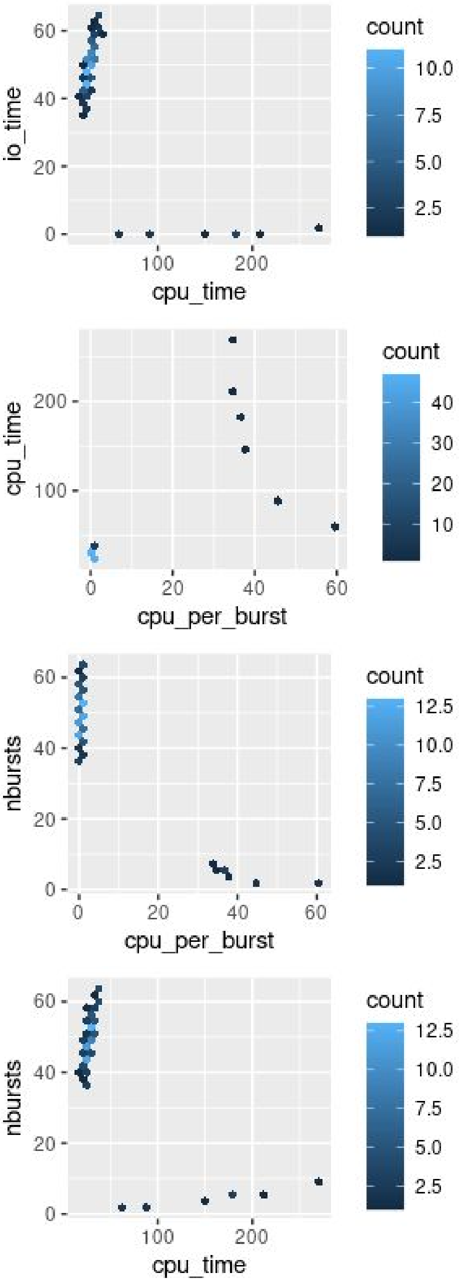


Figure 13. Four different visualizations for the “h3” workloads

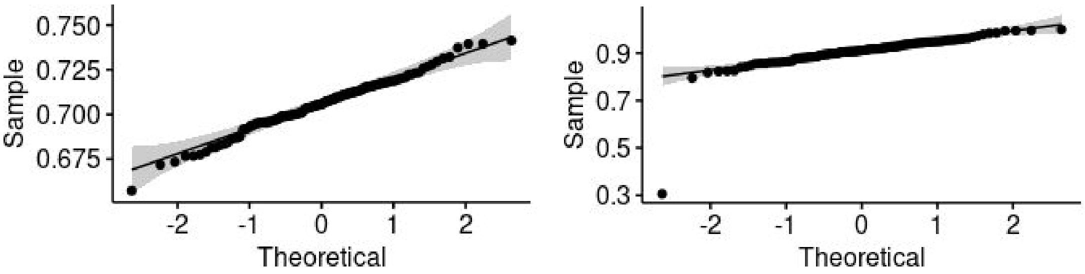


Figure 14. R qq plots with the detected outlier (left FCFS, right SRTF)

As expected, the p-value of the paired t-test yielded 2.2e-16 , significantly lower than a significance level of 0.05, so we reject the null hypothesis, meaning the average of used cpu is significantly different between the FCFS and SRTF.

To expand on this, we decided to test for differences on the remaining schedulers. We thought that the FCFS would be the only one that's significantly different and the others would have similar CPU usage, as it appeared from the single tests that we ran whilst exploring different hypotheses.

$$H_0 : usage(SRTF) = usage(SJF) = usage(RR)$$
$$H_1 : \exists i, j : usage_i \neq usage_j$$

For this test we decided to run a Repeated Measurements ANOVA. The

scheduler	p-value
RR	0.6498467
SJF	0.6060078
SRTF	0.6539960

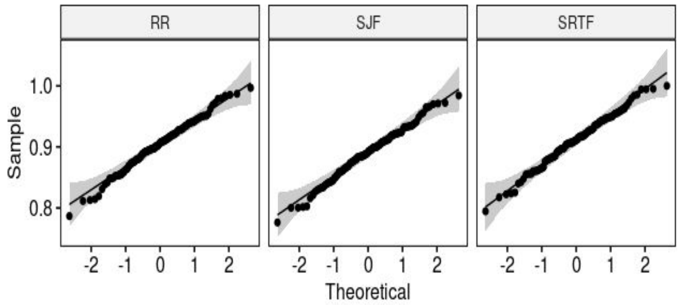
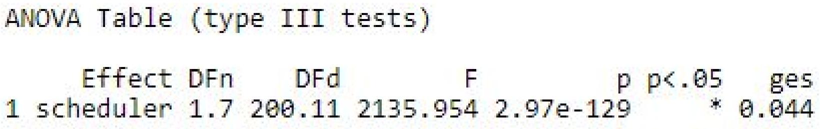


Table 3. Results of the Shapiro test

Aside from normality, another assumption of the Repeated Measurements ANOVA is the sphericity (all intents and purposes, it is equivalent to the homogeneity of variances).
The function anova_test() that we used (rstatix package) already checks this for us (with the Mauchly's test). The function get_anova_table() (also rstatix package) that extracts the anova table automatically applies Greenhouse-Geisser sphericity correction to the factors violating the sphericity assumption.

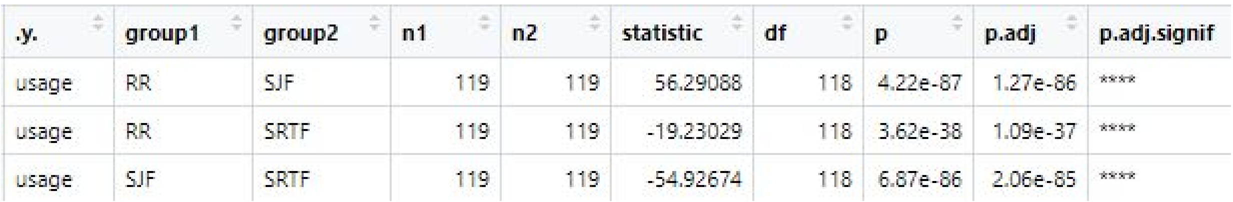
The result is as follows:



```
ANOVA Table (type III tests)

     Effect DFn    DFd        F         p p<.05   ges
1 scheduler 1.7 200.11 2135.954 2.97e-129     * 0.044
```

Figure 16. Output of the Repeated Measures ANOVA

As we rejected the null hypothesis, we ran pairwise t-tests as a post-hoc analysis.



.y.	group1	group2	n1	n2	statistic	df	p	p.adj	p.adj.signif
usage	RR	SJF	119	119	56.29088	118	4.22e-87	1.27e-86	****
usage	RR	SRTF	119	119	-19.23029	118	3.62e-38	1.09e-37	****
usage	SJF	SRTF	119	119	-54.92674	118	6.87e-86	2.06e-85	****

Figure 17. Results of pairwise t-tests

The results were surprising, but nevertheless stated that the differences are significant.
This turn of events may be explained by our preconceptions because we didn't notice any "significant differences" while manually looking at the data.

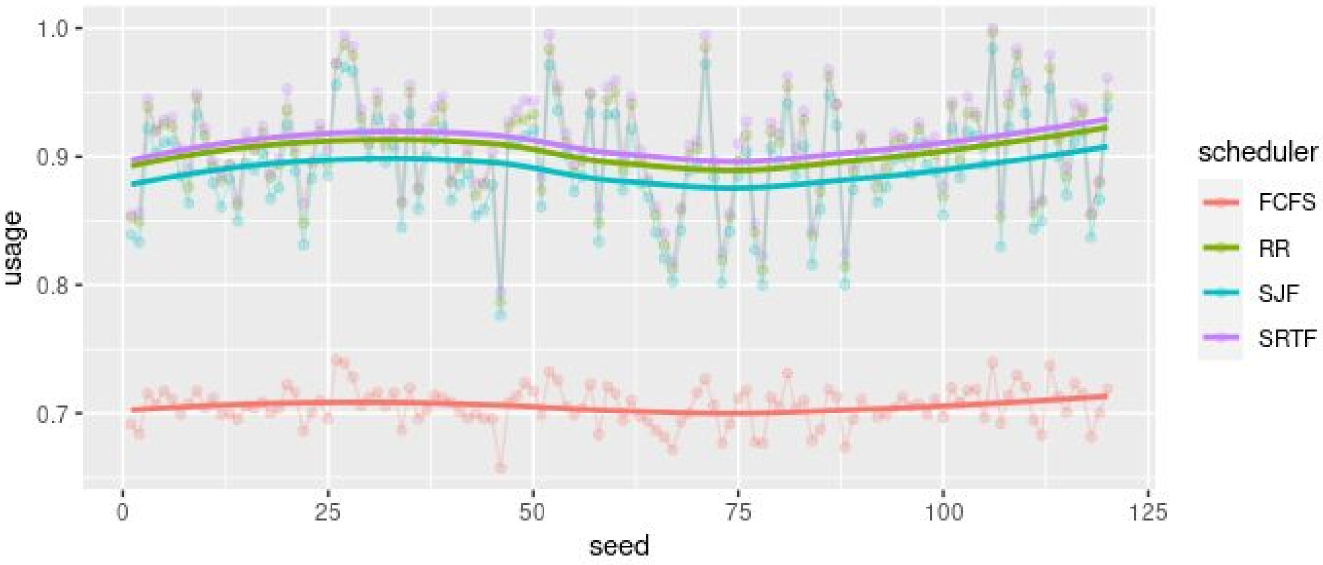


Figure 18. Average CPU usage by seed for each scheduler

References

Theoretical Lessons Slides, from 2020 by Carlos Fonseca
https://www.slideshare.net/rosariocacao/testes-parametricos-e-nao-parametricos-3396639
https://www.technologynetworks.com/informatics/articles/paired-vs-unpaired-t-test-differences-assumptions-and-hypotheses-330826#:~:text=What%20is%20an%20unpaired%20t,significant%20difference%20between%20the%20two.
https://statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide.php
https://statistics.laerd.com/statistical-guides/repeated-measures-anova-statistical-guide.php
http://www.sthda.com/english/wiki/kruskal-wallis-test-in-r

Google Drive with the tests and code:
https://drive.google.com/drive/folders/1YIwvXtvtpNFAHonANOv7AYonkD-isTRG?usp=sharing