

Лабораторная работа 6

1. Эксперименты с последовательным и параллельным выполнением вычислительно сложных задач

Первый эксперимент. (1 процессор)

Последовательный запуск

```
./experiment.sh 1ex_1processor_sequentially.log sequentially.sh  
python3 make_gr.py 1ex_1processor_sequentially.log
```



Видим почти линейный график. Скачки можно объяснить запуском различных легких процессов (открытие текстового редактора и тп). Так как exp.sh не зависит от аргумента (в плане времени) и запускается n раз (n in [1, 20]), график линейный.

Параллельный запуск

```
./experiment.sh 1ex_1processor_parallel.log parallel.sh  
python3 make_gr.py 1ex_1processor_parallel.log
```

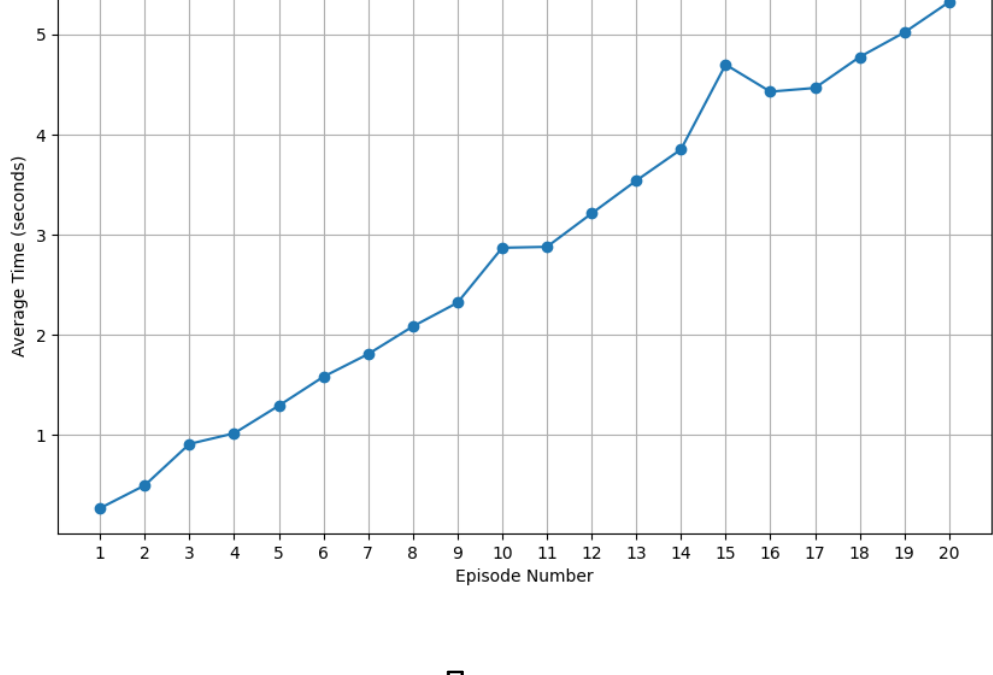


График не отличается от первого. Процессы выполняются на одном процессоре

➔ Не параллельно, а асинхронно. Поэтому графики схожи.

Второй эксперимент. (2 процессора)

Последовательный запуск

```
./experiment.sh 2ex_2processor_sequentially.log sequentially.sh  
python3 make_gr.py 2ex_2processor_sequentially.log
```

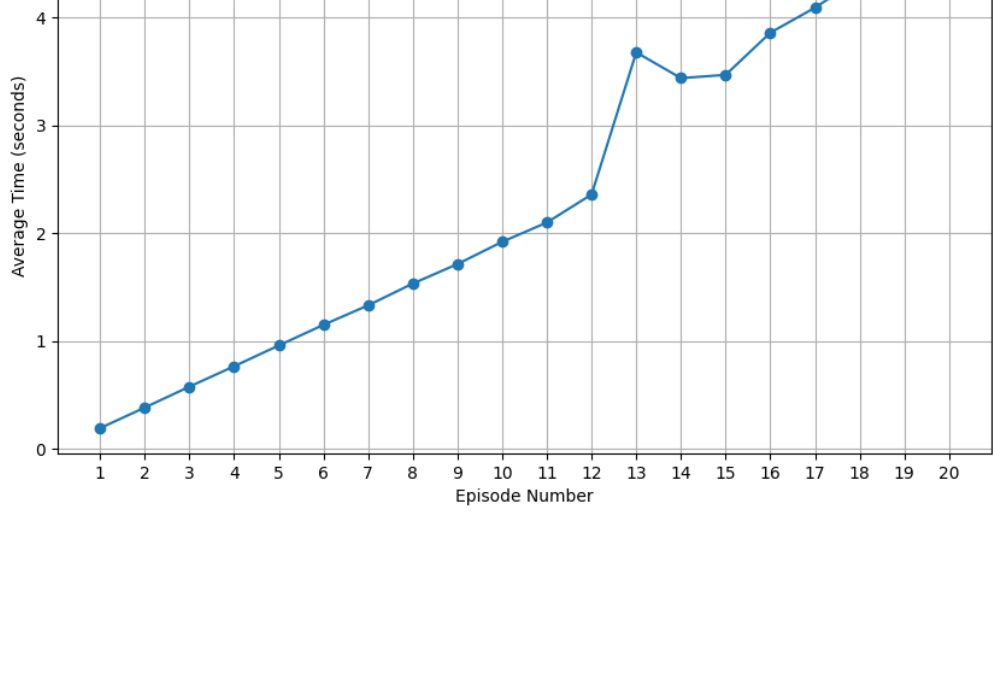


График такой же как предыдущие, так как процессы выполняются последовательно. Нет прироста в скорости работы, так как ничего не параллелится.

Параллельный запуск

```
./experiment.sh 2ex_2processor_parallel.log parallel.sh  
python3 make_gr.py 2ex_2processor_parallel.log
```

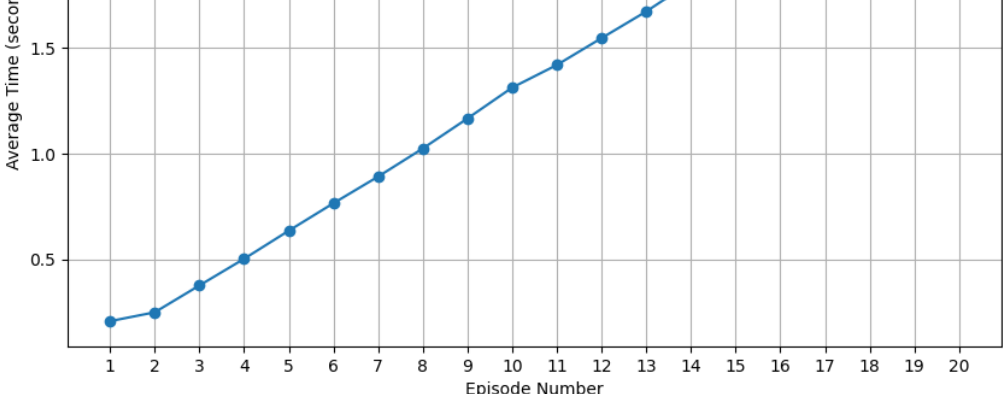


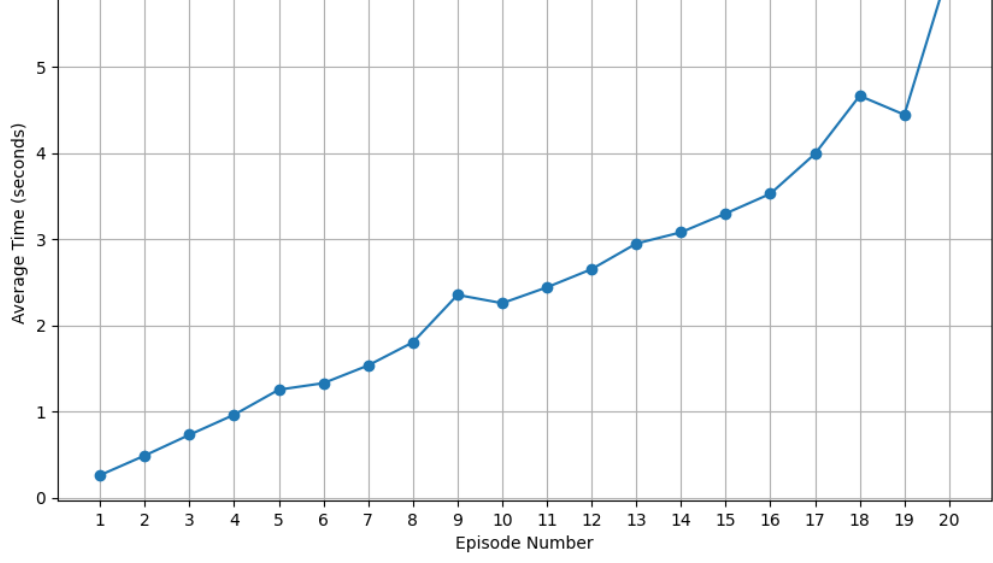
График так же линейный, но виден прирост в скорости (примерно в два раза быстрее, так как одновременно запускается два скрипта). В данном случае скрипты действительно работают параллельно.

2. Эксперимент с параллельным и последовательным выполнением задач с большими объемами считываемых и сохраняемых данных

Первый эксперимент. (1 процессор)

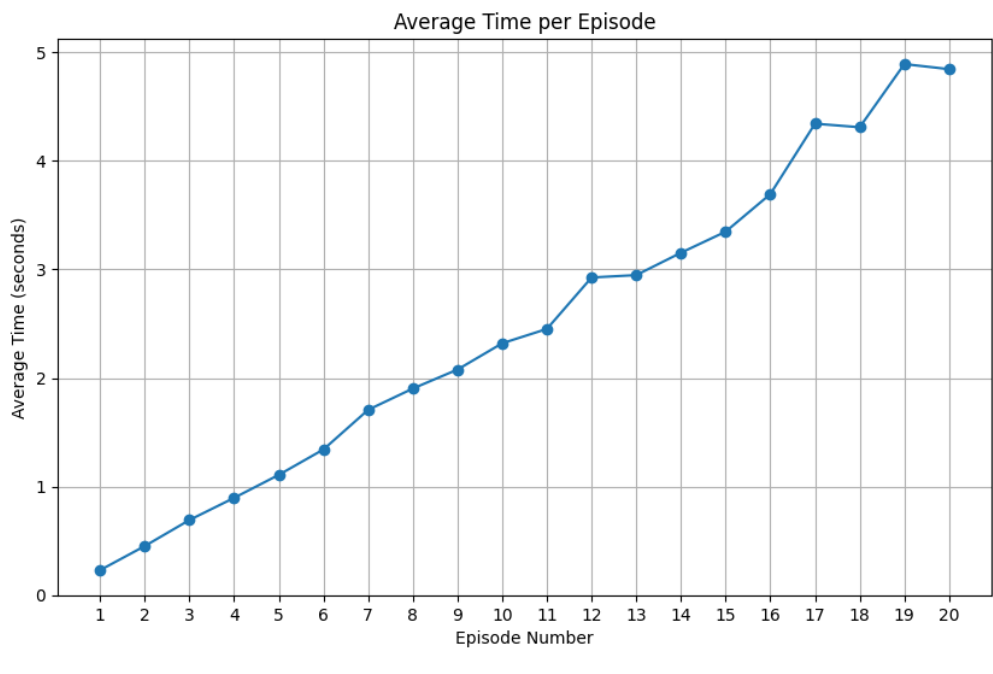
Последовательный запуск

```
./experiment2.sh 1ex_1processor_sequentially_handler.log  
sequentially2.sh  
python3 make_gr.py 1ex_1processor_sequentially_handler.log
```



Параллельный запуск

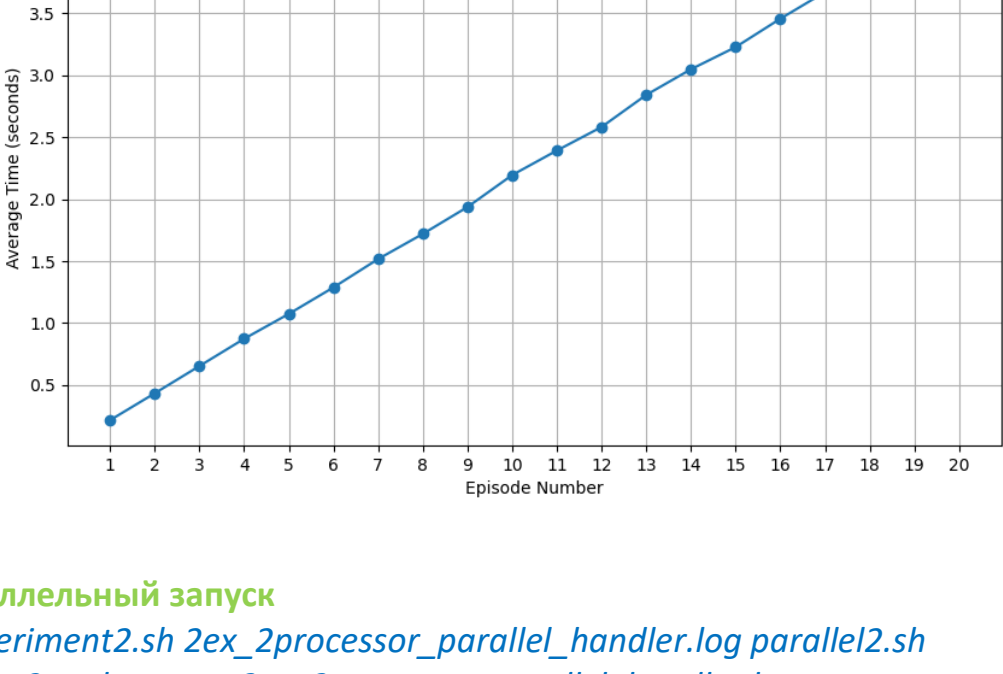
```
./experiment2.sh 1ex_1processor_parallel_handler.log parallel2.sh  
python3 make_gr.py 1ex_1processor_parallel_handler.log
```



Второй эксперимент. (2 процессора)

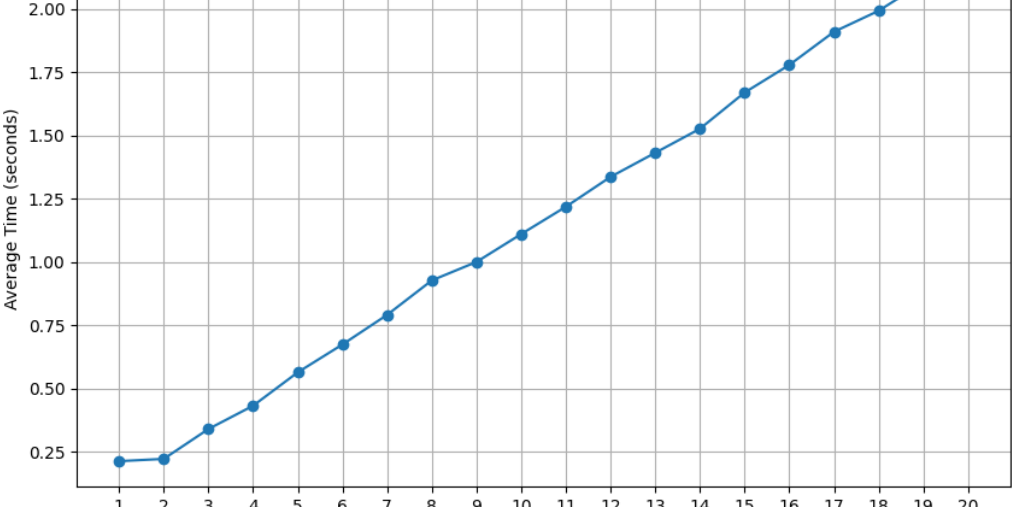
Последовательный запуск

```
./experiment2.sh 2ex_2processor_sequentially_handler.log  
sequentially2.sh  
python3 make_gr.py 2ex_2processor_sequentially_handler.log
```



Параллельный запуск

```
./experiment2.sh 2ex_2processor_parallel_handler.log parallel2.sh  
python3 make_gr.py 2ex_2processor_parallel_handler.log
```



Все эти графики можно объяснить аналогично первым четырем