**Team members**

Anton Razzhigaev

Maria Begicheva

Ekaterina Zharkova

**Time of implementation**

10.12.19 – 17.12.19

**Project description**

We have chosen the program of massive loading data from different web-links. There is a problem that it takes a long time to load a huge number of images during working with Machine Learning problems.

We have found a free dataset of images. It represents a list consisted of thousands (4295) of links. It takes more than 30 minutes to download each image one by one. To decrease time which this process takes, we decided to divide a list with links to several equal parts and process each of them separately.

For doing it a Python script “download\_images.py” was created. It takes three external variables: name of file with links, number of parts this list to be divided into and its id to decide which part of files to process. Result of this script’s working is a directory “downloaded\_images” in which all images are located.

To make processes of separate downloading work in parallel, bash-script “multiprocess.sh” was created. It directs each process to a separate thread.

To test a parallel downloading with different numbers of processes one more bash-script “d\_p.sh” was created. Changing the number of parts of list of links in a loop, it measures the time which parallelizing script actually takes for downloading the whole dataset.

The whole project is located in “PROJECT.zip” folder. There is a full list of links in it. Another one “PROJECT\_laptop.zip” contains a reduces list of links.

**Testing processes**

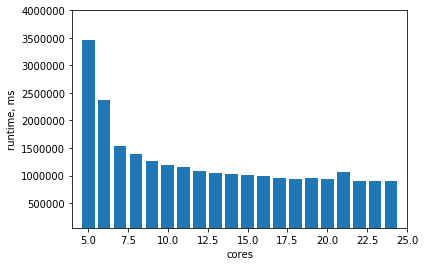
As it was already mentioned, there is a bash-script which runs the parallelizing script for several times. Its input variables are: name of file with list of links and number of parts which this list should be divided into. For running the script, doing ‘sbatch’ command, we used 24 CPU cores on cluster. The table of time consuming for downloading through different numbers of threads (consequently different numbers of links in one thread) was obtained. Starting from 5 parts then dividing ratio was increased to 24 by 1. As

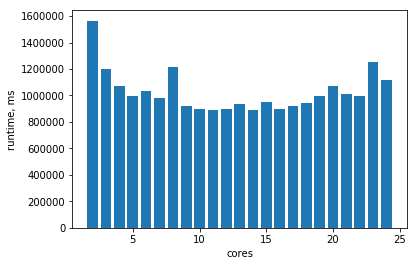
The program was also tested on two laptops. File with links of images was reduced by about 10 times and the script ran with the numbers of division varying from 2 to 24. All measurements were done for 3 times to get an average value. Tables and plots were obtained for both cases

**Results**

Here are plots which were obtained:

1. For testing on Zhores



1. For testing on laptop

To build these plots we have obtained several data tables which are located in “res\_zhores.zip” archive for Zhores and in “res\_laptop.zip” archive for laptop.