

Assignment 4: High Performance Computing Message Passing Interface (MPI)

In this exercise, we will implement the financial calculations service using a parallel distributed approach. The financial calculations service will read a request from the queue that was implemented in Assignment 3, and perform the following calculations:

- Let $\{a_1, a_2, \dots, a_n\}$ be the list of assets that was received with the request, and $a_i \in [1, 100]$
- For each a_i , a time series T_i with stock prices is retrieved (a_i is just the index of T_i)
- For each T_i , a linear regression model

$$y_i(t) = \beta t + \alpha$$

will be calculated using the whole time series as training data (let T be the length of the time series and therefore $t \in (0, T]$).

- The resulting linear model will be used to predict the value of the asset for the next time step, $y_i(T+1)$.
- The result of the simulation will be the average of the predicted values of the assets in the request

$$y = \frac{1}{n} \sum_{i=1}^n y_i$$

To speed up the calculations, the calculation service has access to a number P of processors (which can be in the same machine or distributed among several machines). To set up this processing cluster, we will use the *Message Passing Interface* and its implementation OpenMPI. The service will satisfy the following requirements:

- Upon starting, the service will generate 100 random time series with 300 days (sample code from `DS_Examples/mpi/create_timeseries.py`)
- Then, the service will pull the first message from the jobs queue. If the queue is empty, the service will block until a message is available.
- For each asset i in the request, it will pass a message to a worker processor to calculate the linear model for the time series T_i .
- The service will gather the results of its workers to compute the final result, which will be sent to the results queue.
- Now the service will take the next message from the queue.
- The number of processors will be configurable and default to 5.

We will see MPI examples in the exercises and sample code will be available in the `DS_Examples/mpi` folder.

Optional: Create a *docker-compose.yml* file that starts the service, creates a local Docker network and starts 5 worker nodes.

Deadline: 11/06/2023, 23:59 CET.

Additional requirements

- All files shall be submitted in a single zip file.
- A README.MD file will also be included in the submission with a short description of the submitted files.

Assessment

Total: 15 points.

- All requirements are satisfied: 10 points.
- The documentation is concise and technically correct: 5 points.
- **Optional exercise:** additional 5 points.