High Performance Computing

Iulia Enache,
Ogoke Felicity

6-6-2022

—

Distributed Systems

Libraries & Modules

- NumPy
- pandas
- json
- requests
- Mpi4py
- timeseries

Process

File: Financial_calculations_service.py

- We created the communicator using the mpi4py library
- We generated 100 time series with 300 days
- We included a while loop for getting the job descriptions from the jobs queue.
- And for each job, we got the list of assets and put each asset as a task
- For each unit containing the asset we calculated:

→Firstly, we got the time series using the asset as an index

→Then we used it for fitting a linear regression model

→With each model, we predicted the value for the next step

- After getting all predicted values, we computed their average and sent it to the results queue

File: docker-compose.yml

- At the root of the app project, create a file named docker-compose.yml.
- In the compose file, we'll started off by defining the schema version.
- Next, defined the list of services (or containers) we want to run as part of our application.
- The servers are;
  - rest_server authentication.py
  - rest_server_master_data.py
  - Rest_server_message_queues.py
  - Rest_server_Calculations.py
- Finally we started up our applications by running docker-compose up.