colca$_{d}ocsDocumentation$

Release latest

Aug 02, 2021

CONTENTS:

Everything you need to know about COLCA.

COLCA (Cost Optimizator and Low Carbon Accountability) is a Python App for optimizing operation costs of industrial processes powered with a mixture of renewable / non renewable energy. The App also performs an accountability of *Carbon Emissions* for each project and forecasting its cost.

CHAPTER

ONE

HOW THE DOCUMENTATION IS ORGANIZED

- *What is COLCA?* shows what is the final purpose of COLCA.
- *Tutorial* take you by the hand through a series of steps to run a simulation of COLCA App. So, if you're new to COLCA, please start here.
- *How to guide* are recipes. They guide you through the steps involved in addressing key problems and use-cases.
- *Reference guide* contain technical reference for web scraping, APIs, features. Describe how they work and how to use it .

CHAPTER

TWO

CONTENTS

2.1 What is COLCA?

Imagine that you have a machine, for instance a water pump. But this machine has something different, it is powered through a solar panel. So, you have a solar water pump. Your goal with this solar system is to have running water at a very low cost and with almost no CO2 emissions. Great!

Let's imagine now that this solar water pump needs to fill in a big tank with water in a day. The water of this tank will be used later.

In order to have a very low energy cost and minimize CO2 emissions, you need to check the weather forecast daily to switch between using the solar energy or power the system with the national grid. Not only that, you have to check if the oil levels of the pump are correct and ckeck out for oil at good pricess evary time it must be changed.

Ok, you have only one machine, you can do it yourself.

But, if you have dozens of different machines, spread out there, different locations, different weather conditions, different renewable power sources…different needs…how to handle that? How to perform a cost control, or monitorize CO2 emissions and the price for that emissions?

There is a lot of work to do for going green!

2.1.1 What is the final purpose of this App?

COLCA manages all these things for you. It analyzes the weather forecasting for each 'machine', energy prices, CO2 right prices and many other parameters to set up the optimal time window for each 'machine' to work. For each 'machine', it automatically optimizes the renewal energy source use and minimizes the CO2 emissions (and associated costs).

But not only that. It manages the purchase process of elements needed by the 'machine'. For instance, if the 'machine' needs industrial oil for working properly, it searches the web for a good price and perform the purchase (this feature is not available yet).

Everything is under control. Saves money and time.

2.1.2 Why use COLCA?

It is time for going GREEN.

Optimize renewal sources is the best way to increase your profits. Avoiding CO2 emissions is another good way to increase your profits and make a great contribution to our future.

2.1.3 What is a 'process'?

COLCA doesn`t use 'machine', we prefer to call it 'process'. Why? Because is more general and adds abstraction about the things we are working with.

A 'process' could be our solar water pump, a wind-powered tool... anything that makes use of a renewal power source.

2.1.4 What kind of information has a 'process'?

A 'process' can contain any kind of information. For example dates and time for working, lists of elements that it needs for working properly, temperatures, pressures, any pyshical measurement you want to manage.

An example could be the following:

```
'city_name': 'My City',
'id_number': 1,
'name': 'pump station 12',
'last_updated': '2021-06-04T06:20:37.185000',
'new_needs': True,
'needs_items': ['oil', 'rhodium'],
'new_orders': True,
'start_work': '2021-06-04T06:20:37.184000',
'time_span': 1200
'purchase_orders_made': True,
'purchased_items': ['gasoline', 'aluminium'],
'co2_penalty': 23.44
```

2.1.5 Tell me about how it works...

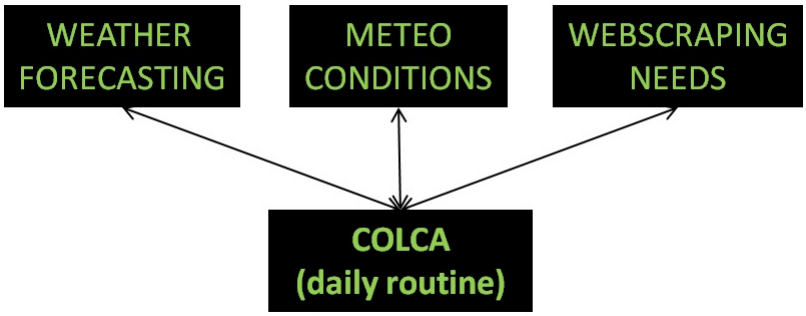
You are the owner of a company and you want to apply COLCA to some 'processes' that are now powered througn windmills installed nearby your production plant.

You pick, for instance one 'process' with this information profile:

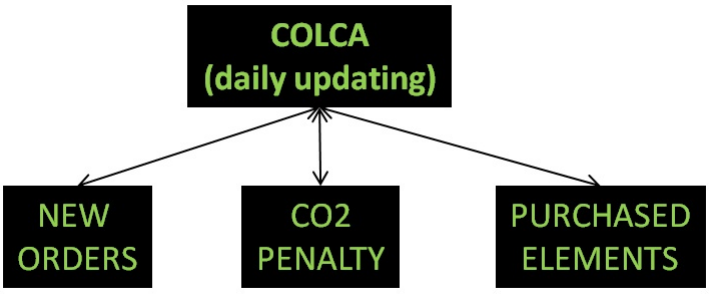
'city_name': 'Other City',
'id_number': 198,
'name': 'compressor 46',
'last_updated': '2021-02-03T05:15:37.174000',
'new_needs': True,
'needs_items': ['oil', 'diesel'],
'new_orders': True,
'start_work': '2021-06-04T06:20:37.184000',
'time_span': 1200
'purchase_orders_made': False,
'purchased_items': [],
'co2_penalty': 0

This 'process' is inside COLCA. Everyday COLCA is executed automatically.

For each 'process' COLCA checks the weather forecast, meteo conditions, inspect websites for purchasing elements (if it is needed) at the best possible price, and evaluates CO2 emissions.



After all this work, COLCA updates the status of each 'process', giving new orders to the 'process' and with calculation of the CO2 emissions and its costs (penalty).



Later any process can make a consult at any moment to COLCA to update its status

COMPLETAR ESTA PARTE CON LA INFORMACION DE LOS RESULTADOS

2.2 Tutorial

This tutorial guides you in running a simulation with COLCA App, step by step.

2.2.1 Installation of COLCA App

To use COLCA, first clone COLCA from the GitHub repo:

```
(.venv) $ git clone https://github.com/alberto-sastre/Project2_AS.git
```

2.2.2 Installation of skforecast

You have to install `skforecast` package from the github repository

```
(.venv) $ pip install git+https://github.com/JoaquinAmatRodrigo/skforecast@v0.1.9
```

2.2.3 Initialization of mongoDB

You have to install mongoDB and should have it running on your system.

If you have installed the MongoDB 4.4 Community Edition, you have to run MongoDB as a background service in terminal:

```
(.venv) $ brew services start mongodb-community@4.4
```

If you are working under Windows OS, by installing MongoDB in Windows you can install it as a background service (mongoDB installer will ask you for this option) and it will be available at running time.

2.2.4 Initialization of uvicorn

First, install uvicorn in your virtual enviroment using pip:

```
(.venv) $ pip install uvicorn
```

Second, execute uvicorn in terminal:

```
(.venv) sascr/colca_services$ uvicorn colca:app --reload
```

Third, do not close this terminal until the end of the simulation.

2.2.5 Run the simulation

Now you are ready to run the simulation, to do this:

```
(.venv) sascr/colca_tutorial$ python3 -m tutorial
```

It takes a while to complete, not much ;)

First, you have to setup 2 processes in order to perform a simulation. For this, you have to follow the instructions on screen, step by step. We recommend you to use the 'default' values, it is easier.

1. You have to define a city name for a process:

   ```
   SETTING UP FIRST PROCESS:
   Select city name: [Oviedo, Aviles]; default: Oviedo >>
   ```

2. You have to give a name to the process (to get it correctly idetified):

   ```
   Select process name: [Pump Station 1, Pump Station 2]; default: Pump Station 1 >>
   ```

3. You have to select what does your process need (for instance, oil):

   ```
   Select need items: [oil, rhodium, aluminium]; default: oil >>
   ```

4. You have to do the same with the second process:

   ```
   SETTING UP SECOND PROCESS:
   Select city name: [Oviedo, Aviles]; default: Aviles >>
   ```

Second, once you have your processes defined, you will see the description of then on screen. Now, you can press any button to start the simulation:

```
THESE ARE THE TWO PROCESSES WITH FULL DATA:


city_name: Oviedo                                    city_name: Aviles
co2_penalty: 23.44                                   co2_penalty: 23.44
id_number: 1                                         id_number: 2
last_updated: 2021-06-04T06:20:37.185000             last_updated: 2021-06-04T06:20:37.
↪185000
name: Pump Station 1                                 name: Pump Station 2
needs_items: ['oil']                                 needs_items: ['aluminium']
new_needs: True                                      new_needs: True
new_orders: True                                     new_orders: True
purchase_orders_made: True                           purchase_orders_made: True
purchased_items: ['gasoline', 'aluminium']           purchased_items: ['gasoline',
↪'aluminium']
start_work: 2021-06-04T06:20:37.184000               start_work: 2021-06-04T06:20:37.184000
time_span: 1200                                      time_span: 1200
```

Finally you will be presented with the **simulation results** on screen. You can also access to this data in a `.log` file where all the same information is available:

```
RESULTS OF THE SIMULATION, IN GREEN COLOR YOU CAN SEE THE CHANGES:


city_name: Oviedo                                    city_name: Oviedo
```

```
co2_penalty: 0
id_number: 1
last_updated: 2021-06-04T06:20:37.185000
↪321000
name: pump station 21
needs_items: ['oil', 'rhodium']
new_needs: True
new_orders: True
purchase_orders_made: True
purchased_items: ['gasoline', 'aluminium']
start_work: 2021-06-04T06:20:37.184000
time_span: 1200
```

```
co2_penalty: 23.44
id_number: 1
last_updated: 2021-06-14T07:37:42.
↪321000
name: pump station 21
needs_items: ['oil', 'rhodium']
new_needs: True
new_orders: True
purchase_orders_made: True
purchased_items: ['oil', 'rhodium']
start_work: 2021-06-14T15:00:00.000000
time_span: 2465
```

2.2.6 Interpreting the results

Once you have the results of the simulation, either in the `.log` file or in screen, we can make an interpretation of the results with you, in order to extract useful information.

1. co2_penalty

   `co2_penalty: 0` `co2_penalty: 23.44`

   Indicates de c02_penalty, expressed in the currency of the country. It applies when the process has to get power from a non-renewal source.

2. last_updated

   `last_updated: 2021-06-04T06:20:37.185000` `last_updated: 2021-06-14T07:37:42.`
   `↪321000`

   When this process was updated for the last time.

3. new_orders

   `new_orders: True` `new_orders: True`

   If COLCA determines that there are new orders, it uses this flag, setting it to `True`.

4. purchase_orders_made

   `purchase_orders_made: True` `purchase_orders_made: True`

   If COLCA made a purchase of any element needed by the process, this flag is set to `True`.

5. purchased_items

   `purchased_items: ['gasoline', 'aluminium']` `purchased_items: ['oil',␣`
   `↪'rhodium']`

   This is a list of elements that COLCA has purchasd automatically for the process in order to cover the process's needs.

6. start_work

   `start_work: 2021-06-04T06:20:37.184000` `start_work: 2021-06-14T15:00:00.`
   `↪000000`

   This is the optimal date and time when the process has to start working.

7. time-span

   `time_span: 1200` `time_span: 2465`

This is the time window for optimizing the energy used by the process, in order to avoid co2 emissions and get the better energy price (if needed a non-renewable source).

2.2.7 End of the simulation

COLCA's simulation has come to its end. With this simulation you have seen how COLCA works very fast, analyzing the processes and making the best decisions for them. Keeps the information of each process updated, and an accountability oc co2 emissions and side costs could be easily performed.

In production, this is a fully automated process. Just add a process with the setup information and let COLCA help you going Green!

Note: You can also pip install `requirements.txt` to get all dependencies in your environment at once. You have to do this from the same folder where `requierements.txt` is stored:

```
(.venv) $ pip install -r requirements.txt
```

2.3 How to guide

This guide tries to give you solutions for the topics you may face, to have COLCA running suscessfully for you.

2.3.1 How to install COLCA

For Colca installation just follow these steps: *install_colca*

2.3.2 How to install skforecast

For skforecast installation, just follow this step: *install_skforecast*

2.3.3 How to install and configure mongoDB

For installating and configure MongoDB, just follow these steps: *install_mongodb*

2.3.4 How to install and configure uvicorn

For uvicorn installation and setup, just follow these steps: *install_uvicorn*

2.3.5 How to add 'processes' to the App

For adding processes to the database:

```
(.venv) sascr/colca_services$ python3 -m add_processes
```

Now you have to complete the initial information for each process, one at a time:

```
SETTING UP FIRST PROCESS:
Select city name: >> Madrid
Select process name: >> Pump_Station_21
Select need items: >> aluminium, oil
```

Or you can pass a list of objects to add to the system:

```
(.venv) sascr/colca_services$ python3 -m add_processes <list_of_processes>
```

Each element of <*list_of_processes*> must be a python dictionary with the following key/value pattern:

```
city_name: str
co2_penalty: float
id_number: int
last_updated: datetime ISO format
name: str
needs_items: list of str
new_needs: bool
new_orders: bool
purchase_orders_made: bool
purchased_items: list of str
start_work: datetime ISO format
time_span: int
```

2.3.6 How to remove 'processes' from the App

For removing processes to the database:

```
(.venv) sascr/colca_services$ python3 -m remove_processes <id_of_process>
```

Where <id_of_process> is the 'id-number' that you can find for each process info.

You can remove all processes from the database:

```
(.venv) sascr/colca_services$ python3 -m remove_processes -all
```

By specifying the `all` option, all processes in the database will be removed.

To avoid accidental remove of processes, you need to confirm this operation:

```
YOU ARE GOING TO REMOVE ALL INFORMATION IN DATABASE
DO YOU WANT TO CONTINUE? PRESS 'Y' TO CONFIRM, ANY OTHER KEY TO CANCEL.

WARNING!! THIS OPERATION CANNOT BE UNDONE!
```

2.3.7 How to configure and run COLCA

For setting up and run COLCA App, follow the next steps in order:

1. First run the `config` file:

```
(.venv) sascr/colca_services$ python3 config_colca
```

   You have to set the daily time for running COLCA App, press `enter`:

```
Select daily time execution for COLCA: >> 9:00
```

2.3.8 How to access the results

For viewing data from the database, you can use the `data-view`:

You have to select an `id_number` of a process:

```
Select a process 'id_number' to access information: >> 3
```

You will be presented with the information on secreen as well as in XXX file:

```
EXTRACT OF PROCESS INFORMATION TAKEN FROM THE DATABASE (NOT AVAILABLE YET).
```

2.3.9 How to perform testing of the App

To perform the suite test for the source the code, from the root run `pytest`:

```
(.venv) sascr/pytest
```

2.3.10 How to update COLCA

Updating COLCA...

2.4 Reference guide

ESTA ES LA PARTE EN LA QUE SE DETALLAN TODAS LAS FUNCIONES