Autore: Rosilde Garavoglia

Using Trovi sharing portal on Chameleon

Overview

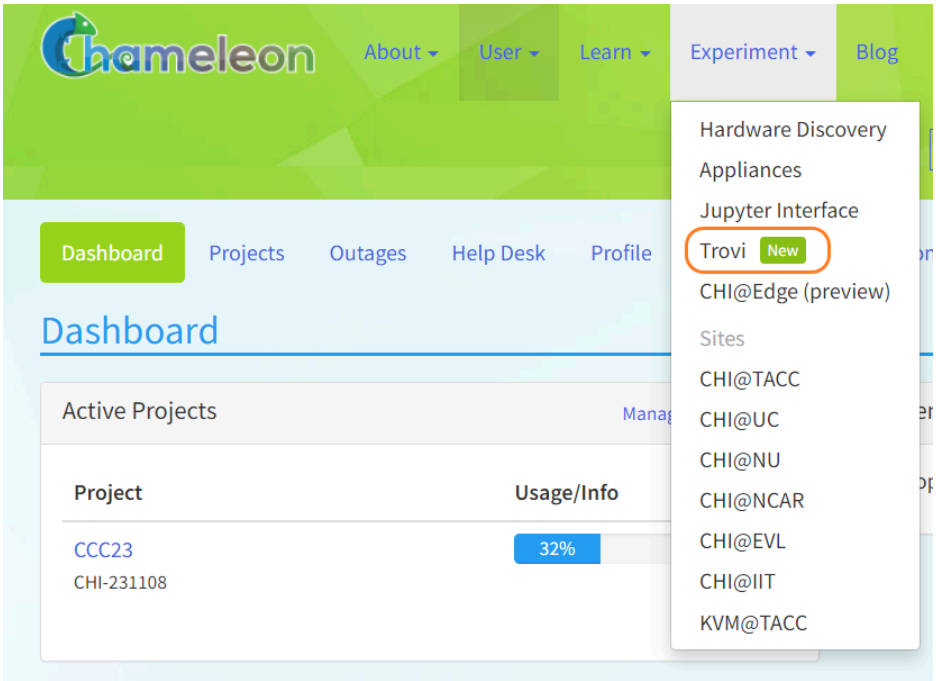
Chameleon Trovi is a sharing portal that allows you to share digital research and education artifacts, such as packaged experiments, workshop tutorials, or class materials. Each research artifact is represented as a remotely accessible folder where a user can put Jupyter notebooks, links to images, orchestration templates, data, software, and other digital representations that together represent a focused contribution that can be run on Chameleon. Users can use these artifacts to recreate and rerun experiments or class exercises on a Jupyter Notebook within Chameleon. They can also create their own artifacts and publish them directly to Trovi from within Chameleon's Jupyter server. In this tutorial, we will use "Bare Metal Experiment Pattern" artifact to experience the potential of Trovi.

Bare Metal Experiment Pattern

The Bare Metal Experiment Pattern artifact will allow us to create a Bare Metal lease, launch a reserved node with an image, associate a Floating Ip to it and configure the instance running the setup script that will download and install all the packages we'll need. Then, we could run the script containing the experiment. It runs stress-ng to run a load on the CPU with 25%, 50% and 100% of the CPU cores and measures the energy and power consumption of each run. Last but not least, we could plot the analysis of the obtained result.

Let's try it

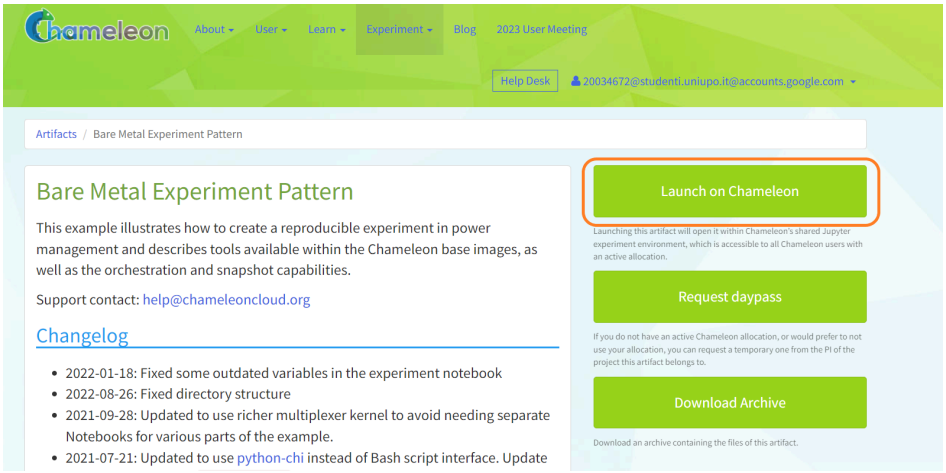
Go to chameleoncloud.org, find the *Experiment* section and select the *Trovi* dropdown option.



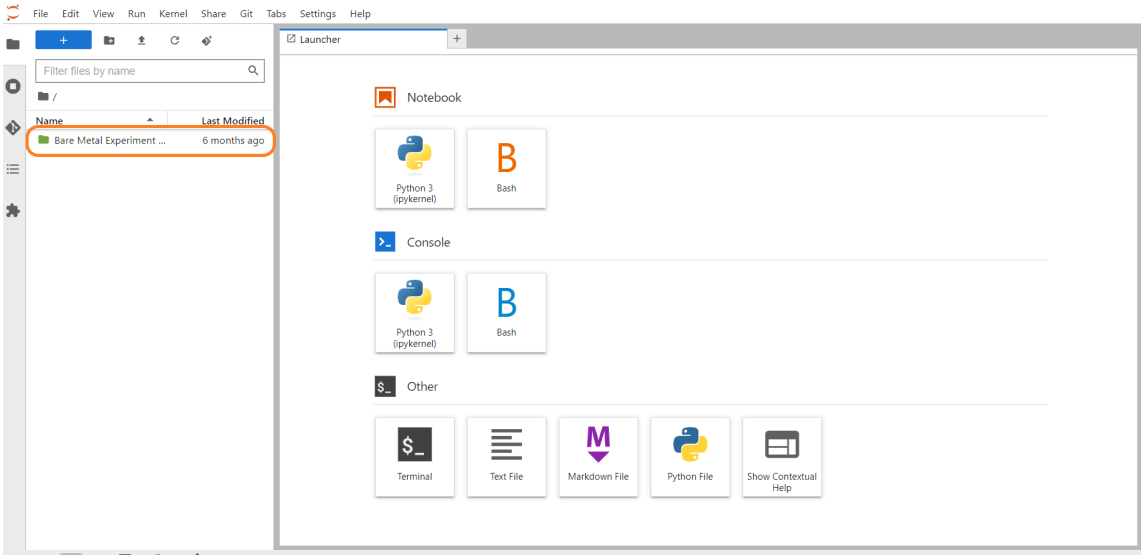
Autore: Rosilde Garavoglia

Once you're on the Trovi homepage, you'll see a list of publicly available experiments and other digital artifacts. You could browse those artifacts or upload your own. Select the *Bare Metal Experiment Pattern* artifact. If you don't see it as one of the first items on the page, search for it in the filter form or just click here.

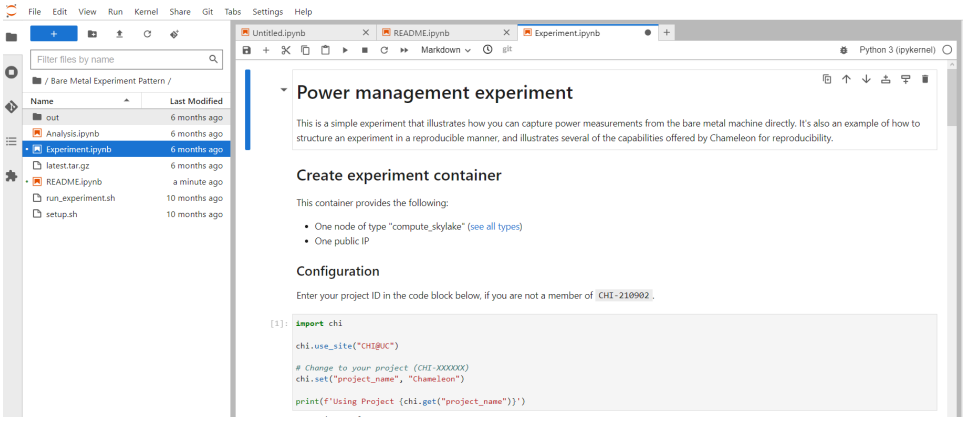
Once you've selected it, you'll see the description of the artifact and the last versions of it. You can also download its archive with the code. If you already have an active Chameleon allocation, select *Launch on Chameleon*. Otherwise, you can request a temporary one from the PI of the project by clicking on *Request daypass*.



You'll be redirected to a Jupyter Hub page. On this page, you can see, on the left, the directory of the artifact. Select and open it.



In the *READMY.ipynb* you can read a brief description of the artifact and all the steps of it. Go to *Experiment.ipynb*.



Create experiment container

The first container (Create experiment container) will allow us to:

- ➢ Create a Bare Metal reservation
- ➢ Instantiate one node of type "*compute_skylake*"
- ➢ Associate a public IP to the instance

If you are not connected to a project, in the first section of the container (*Configuration* section) you can enter your project id and execute the cell (*shift+enter*) to work on that project. If you are, ignore that section and move on.

The *Create Reservation* section will allow you to create a Bare Metal lease. If you want, you can modify the lease duration through the *hours* parameter of the *lease_duration* method.

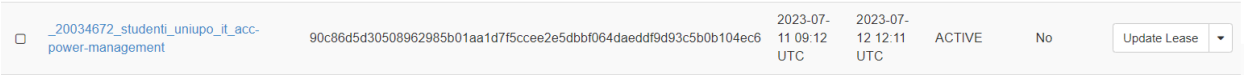
```
import os
import keystoneauth1, blazarclient
from chi import lease

reservations = []
lease_node_type = "compute_cascadelake_r"

try:
    print("Creating lease...")
    lease.add_fip_reservation(reservations, count=1)
    lease.add_node_reservation(reservations, node_type=lease_node_type, count=1)

    start_date, end_date = lease.lease_duration(hours=3)
```

Execute the cell *(ctrl+enter*). You could wait some minutes until it is finished. If you get an error such as "no host available", it may be the case that all of the nodes are reserved. Check the availability calendar to see if this is true.

If everything is alright, you'll see this message:

```
Creating lease...
Waiting for lease to start ...
Lease is now active!
```

You can also check if everything is fine by going to *CHI@UC -> Reservations -> Leases* and seeing if the lease has been created.



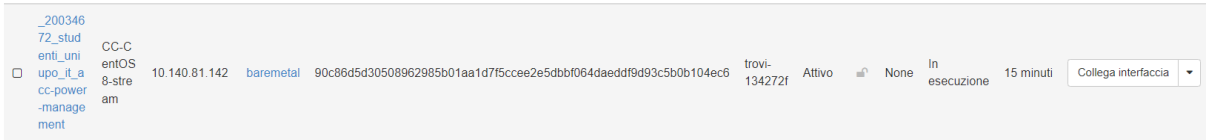
Next, go to *Provision bare metal node* section. The first cell of this section will allow you to launch an instance on the lease you've just created. The image of the instance will be "CC-CentOS8-stream", but you can create it with another instance by modifying the image variable. You can browse the Chameleon images here.

```
from chi import server

image = "CC-CentOS8-stream"

s = server.create_server(
    f"{os.getenv('USER')}-power-management",
    image_name=image,
    reservation_id=lease.get_node_reservation(lease_id)
)

print("Waiting for server to start ...")
server.wait_for_active(s.id)
print("Done")
```

Execute the cell. It will take approximately 10 minutes for the bare metal node to be successfully provisioned. If everything is alright, you'll see this message:

```
Waiting for server to start ...
Done
```

You can also check if everything is fine by going to *CHI@UC -> Compute -> Instances* and seeing if the instance has been created and is correctly active.

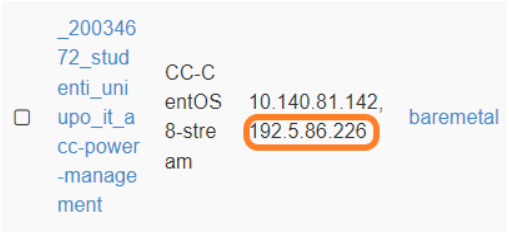


The second and last cell of the *Provision bare metal node* section will allow you to associate a Floating IP Address to your just created instance. In this way, we can reach the instance over the internet or via Jupyter here.

Execute the cell. If everything is alright, you'll see this message:

```
Waiting for SSH connectivity on 192.5.86.238 ...
Connection successful
```

You can also check if everything is fine by going to *CHI@UC -> Compute -> Instances* and seeing if Floating IP has been correctly associated to the instance.



Go to the next section (*Configure the instance*). This cell will allow you to set up the instance through the setup.sh script. It will execute remotely on the instance.

Execute the cell. This warning may appear when running this cell, but that is fine.

```
/opt/conda/lib/python3.10/site-packages/paramiko/client.py:835: UserWarning: Unknown ssh-ed25519 host key for 192.5.86.238: b'16970d4d51
a9d4c97a1b5f92394cac55'
  warnings.warn(
```

If everything is alright, at the end of the verbose you'll see this:

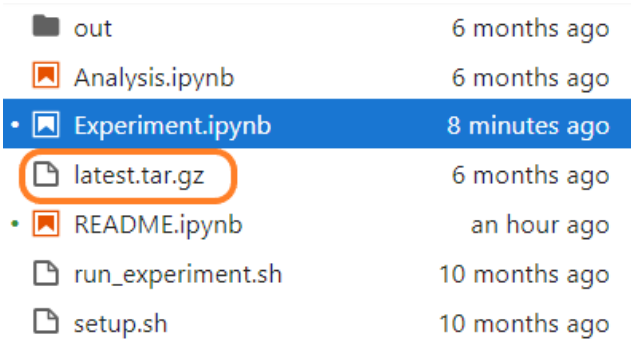
```
Installed:
  Judy-1.0.5-18.module_el8.3.0+757+d382997d.x86_64
  libatomic-8.5.0-20.el8.x86_64
  lksctp-tools-1.0.18-3.el8.x86_64
  stress-ng-0.15.00-1.el8.x86_64

Complete!
```

Autore: Rosilde Garavoglia

Run the experiment

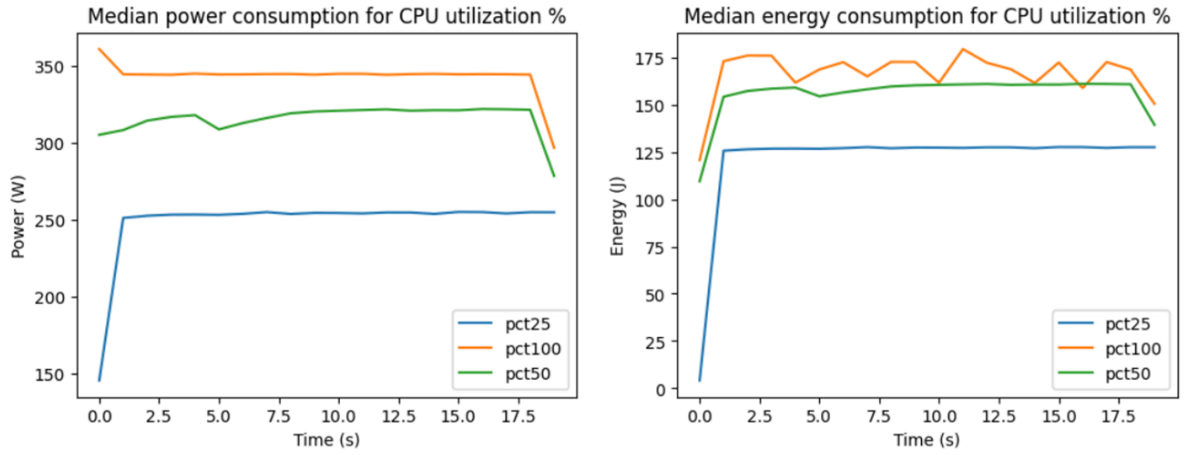
Finally everything is configured and you can run the experiment. Go to *Run the experiment* container. This particular experiment runs stress-ng to run a load on the CPU with 25%, 50% and 100% of the CPU cores, and measures the energy and power consumption of each run. Execute the cell.

After the experiment terminates, you can download the results and extract them to the local output directory for analysis in Jupyter. Execute the last cell of this notebook. You'll get the latest.tar.gz file.



Analyze the data

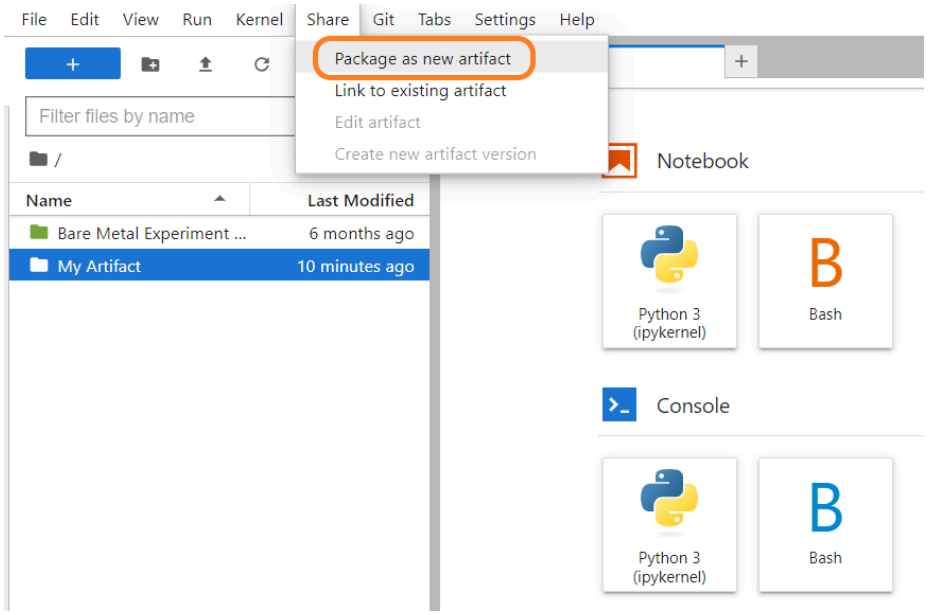
For the last step, you can analyze the obtained data through the *Analysis.ipynb* file.
Execute the cell of the file and visualize the plot of the data.



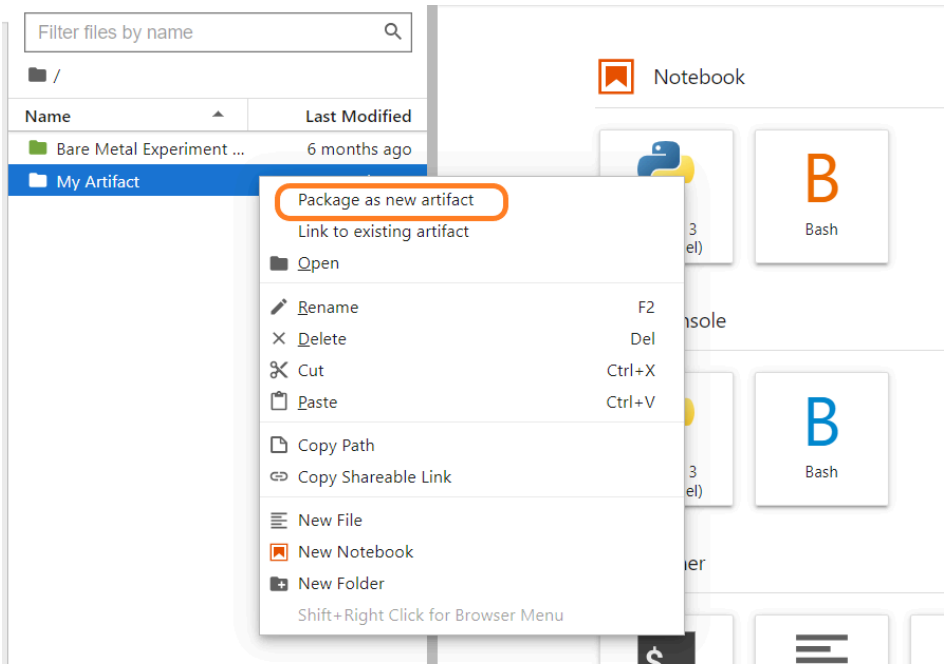
Autore: Rosilde Garavoglia

Create your own artifact

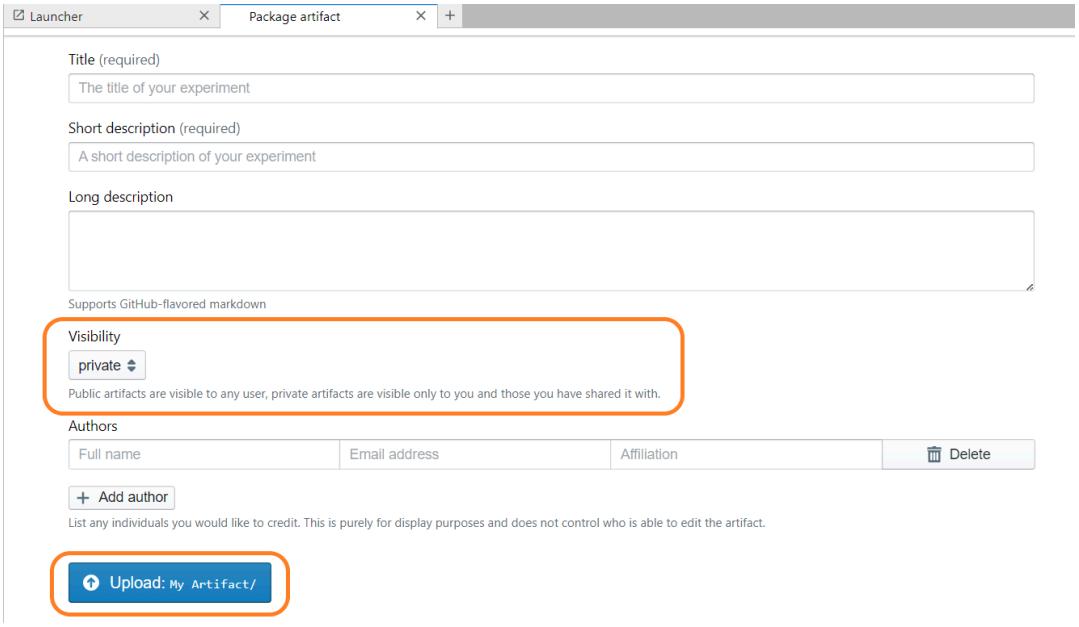
If you want to publish new artifacts to Trovi, you can do it either from your primary Jupyter server or by editing a previously-shared artifact (in this case, you are effectively creating a new “forked” artifact owned by you). When you’ve finished creating or making changes to an experiment, in the Jupyter interface, select the directory (not a single file) you wish to package. Then, click on the *Share* tab and select *Package as a new artifact*.



Otherwise, you can right-click on your artifact directory and select *Package as new artifact*.



You'll be prompted to fill out descriptions about the artifact. To make it accessible to other users you'll need to adjust his sharing settings clicking on the *visibility* section and selecting *public*. When you have finished, click on the *Upload* button at the end of the page.



Your artifact is now packaged and uploaded to Chameleon file storage and, if you want to change it later, you can edit them on the Trovi portal or within Jupyter. For other information about creating and managing your artifacts, click here.