Energy management on PlaFRIM

Supervisor : Brice Goglin (TADaaM)
Intern : Corentin Mercier

Summary

01. Overview of PlaFRIM's consumption

02. Shut down idle nodes with SLURM

03. Reactivity margin and custom timeouts

04. Energy saved thanks to the new system

05. User manual

06. How you can help us

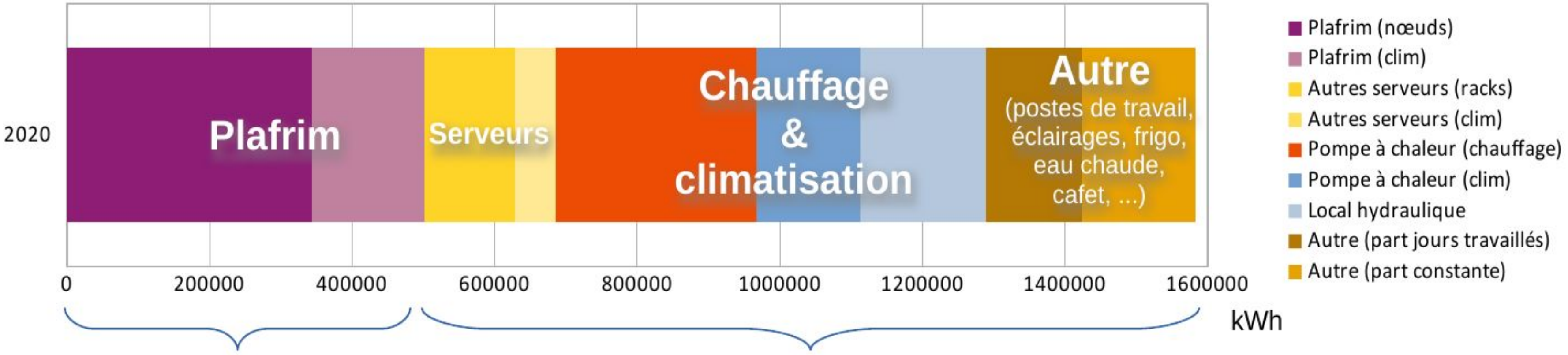
01

Overview of PlaFRIM's consumption

PlaFRIM's share in the building consumption

Overview of Inria BSO electricity consumption

source : cldd-bso@inria.fr



500 MWh / year (50 x 75 m^2 apartment)

PlaFRIM usage overview

Machine utilization per node group

arm	bora	brise	diablo	kona	miriel	mistral	sirocco	souris	visu	zonda
2%	49%	17%	21%	3%	28%	26%	39%	19%	4%	36%

- idle nodes consumption = **128 344 kWh**
- In 2021, **1 kWh = 0.11 €**
  - > money used to power idle nodes = **14 118 €**

Power saving strategies

Non-exhaustive list of strategies and their impact

Strategy	Impact
Shut down idle nodes	High
Reduce CPU frequency during jobs	Moderate
Use the "powersave" governor on idle nodes	Low
Overprovisioning	Low

Power saving strategies

Non-exhaustive list of strategies and their impact

Strategy	Impact
Shut down idle nodes	High
Reduce CPU frequency during jobs	Moderate
Use the "powersave" governor on idle nodes	Low
Overprovisioning	Low

02

Shut down idle nodes with SLURM

SLURM power saving mechanism

1. Identify nodes which have been idle for at least **SuspendTime** seconds.
2. Execute **SuspendProgram** with an argument of the idle node names.
3. Identify the nodes which are in power save mode, <u>but have been allocated to jobs</u>.
4. Execute **ResumeProgram** with an argument of the allocated node names.
5. If the node fails to respond within **SlurmdTimeout**, the node will be marked DOWN and the job <u>requeued</u> if possible

NB : Every name in **bold** *is a variable in slurm.conf*

SLURM's mechanism limits

- New interactive jobs get to wait for nodes to power up
  - > some nodes should remain idle to serve small jobs (reactivity margin)
- **SuspendTime** is the same for every node
  - > special nodes are used for an extended time even if not allocated
  - > arm01, souris, etc

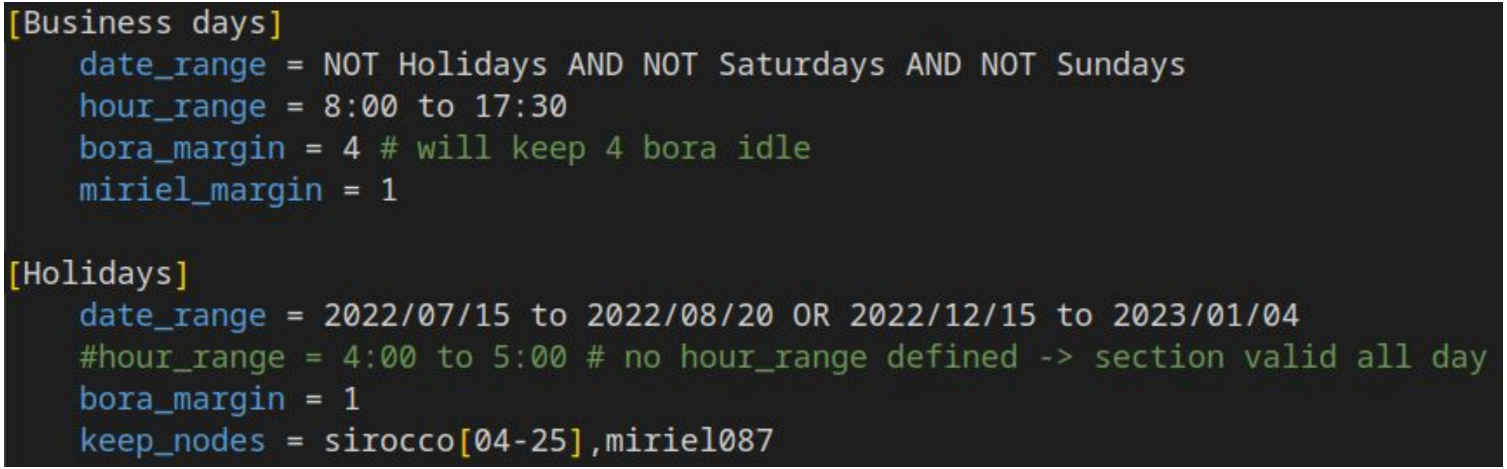
03

Reactivity margin and custom timeouts

Reactivity margin : main difficulty

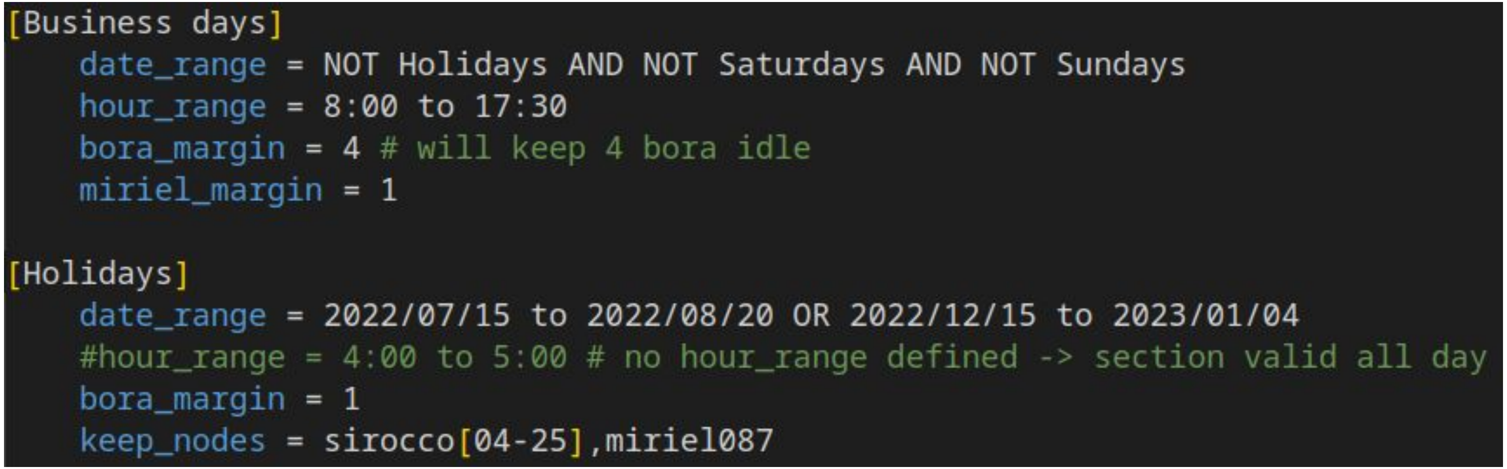
- Only read access to each node idle counter
  - scontrol show node (LastBusyTime)
- One way to write to it
  - make an allocation via salloc / srun

Reactivity margin



```
[Business days]
    date_range = NOT Holidays AND NOT Saturdays AND NOT Sundays
    hour_range = 8:00 to 17:30
    bora_margin = 4 # will keep 4 bora idle
    miriel_margin = 1

[Holidays]
    date_range = 2022/07/15 to 2022/08/20 OR 2022/12/15 to 2023/01/04
    #hour_range = 4:00 to 5:00 # no hour_range defined -> section valid all day
    bora_margin = 1
    keep_nodes = sirocco[04-25],miriel087
```

Reactivity margin



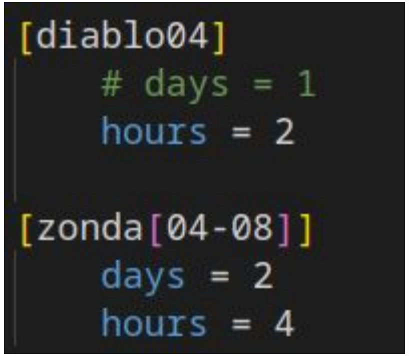
```
[Business days]
    date_range = NOT Holidays AND NOT Saturdays AND NOT Sundays
    hour_range = 8:00 to 17:30
    bora_margin = 4 # will keep 4 bora idle
    miriel_margin = 1

[Holidays]
    date_range = 2022/07/15 to 2022/08/20 OR 2022/12/15 to 2023/01/04
    #hour_range = 4:00 to 5:00 # no hour_range defined -> section valid all day
    bora_margin = 1
    keep_nodes = sirocco[04-25],miriel087
```

- srun -N**4** -C **bora** –job-name=keepIdle true
- srun -N**1** -C **miriel** –job-name=keepIdle true

Custom timeouts

- Based on a "registration" system
  - begins when a wanted node is up
  - sends small jobs until the end of the "registration"
  - delete the "registration" after its end + SuspendTime

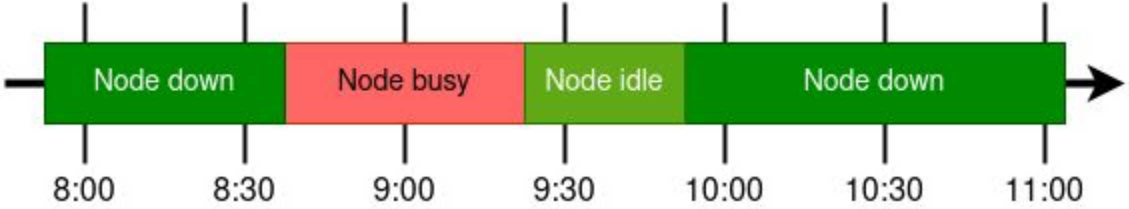
- Each section is a node list
  - define the number of hours and/or days that you want
  - each node of the list will stay up for the time wanted



```
[diablo04]
    # days = 1
    hours = 2

[zonda[04-08]]
    days = 2
    hours = 4
```

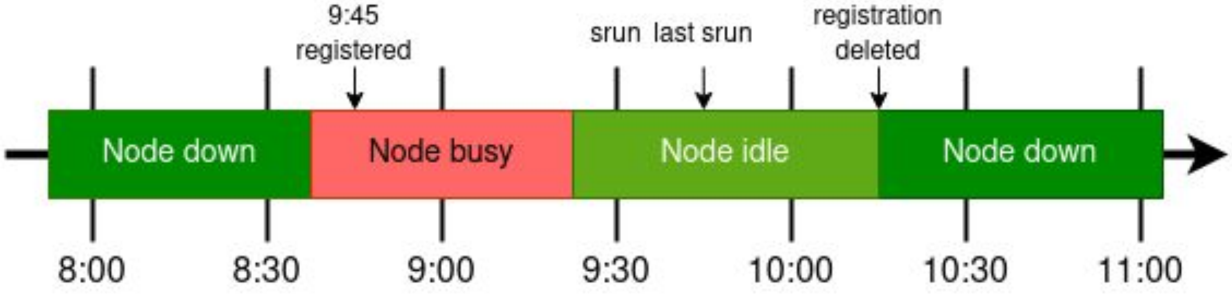
Custom timeouts : example

- SuspendTime = 30 minutes, custom timeout = 1 hour
- Without the system



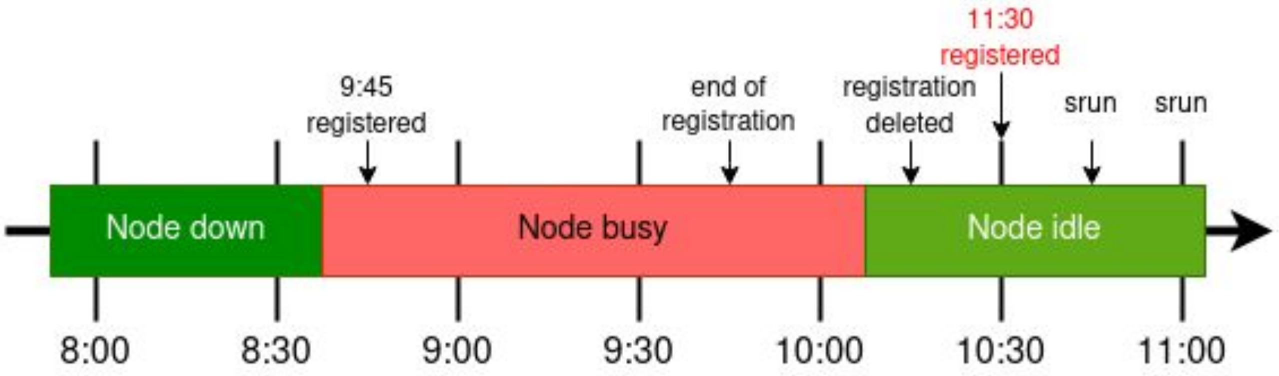
Custom timeouts : example

- SuspendTime = 30 minutes, custom timeout = 1 hour
- The script is called every **15 minutes**



Custom timeouts : example

- The script is called every **15 minutes**
- Node busy **15 minutes** after end of registration -> new registration will occur



Limits of the system

- Based on job allocation
  - quicker increase in job IDs
  - create useless entries in SLURM database
  - may cause a denial of service on very large clusters

04

Energy saved thanks to the new system

Overview of the energy saved

Proportion of powered nodes in their group according to SuspendTime

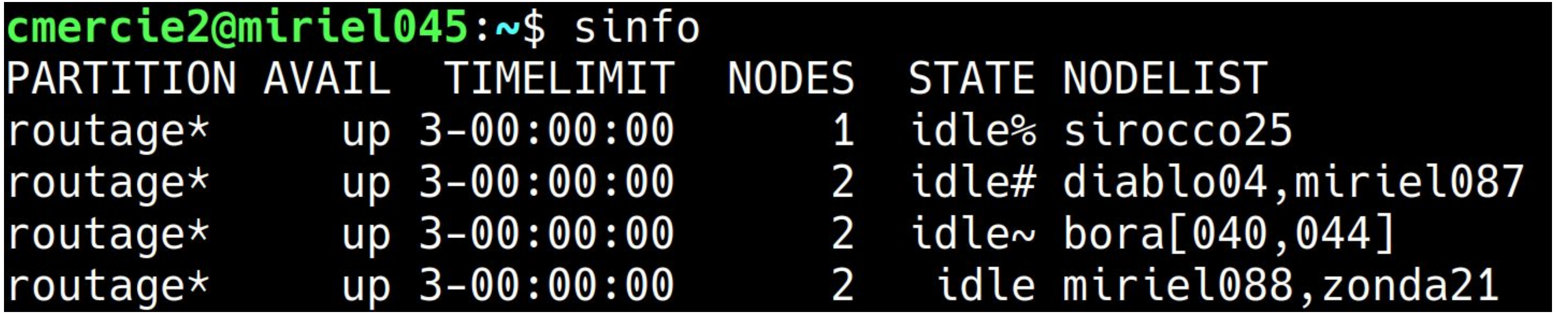
Suspend Time	arm	bora	brise	diablo	kona	miriel	mistral	sirocco	souris	visu	zonda	saved kWh	saved €
0	2%	49%	17%	21%	3%	28%	26%	39%	19%	4%	36%	128 344	14 118
1 hour	3%	55%	18%	22%	3%	30%	28%	42%	19%	5%	38%	122 675	13 494
2 hours	4%	59%	19%	24%	4%	32%	30%	43%	20%	5%	40%	118 123	12 994
4 hours	5%	66%	21%	26%	5%	35%	33%	47%	22%	7%	43%	110 513	12 156

05

User manual

New node states

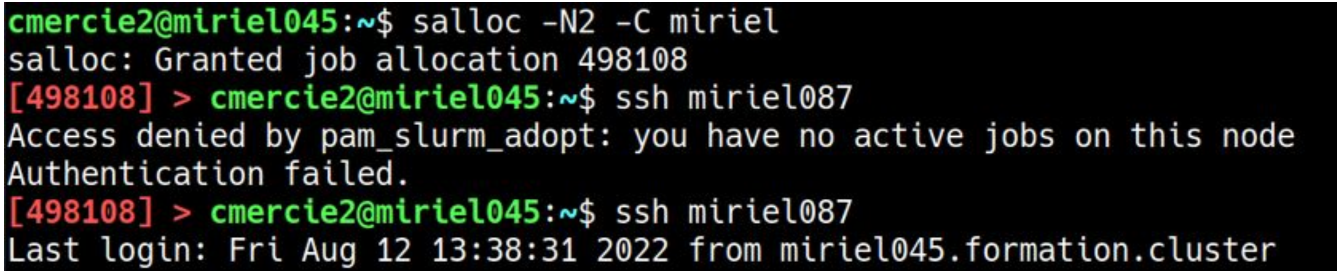
- New symbols will be present when running **sinfo**
  - # : the node is **powering up**
  - % : the node is **powering down**
  - ~ : the node is **down**
- Example



```
cmercie2@miriel045:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
routage*        up 3-00:00:00      1  idle% sirocco25
routage*        up 3-00:00:00      2  idle# diablo04,miriel087
routage*        up 3-00:00:00      2  idle~ bora[040,044]
routage*        up 3-00:00:00      2   idle miriel088,zonda21
```

salloc / srun

- What happens if a node that you requested is down ?
  - srun **blocks** and **wait** for all your nodes then your job begins
  - salloc **returns immediately**
    - you have your allocation !
    - you can't connect until **all your nodes** are ready



```
cmercie2@miriel045:~$ salloc -N2 -C miriel
salloc: Granted job allocation 498108
[498108] > cmercie2@miriel045:~$ ssh miriel087
Access denied by pam_slurm_adopt: you have no active jobs on this node
Authentication failed.
[498108] > cmercie2@miriel045:~$ ssh miriel087
Last login: Fri Aug 12 13:38:31 2022 from miriel045.formation.cluster
```

squeue and node failure

- Jobs sent by the system are called "**keepIdle**"
  - shouldn't last in the queue
  - if you see too many of them, there's a problem
- Node can fail to boot
  - put in **down~** state

- For more information, check the new section **3.10** on the PlaFRIM documentation !
  - https://plafrim-users.gitlabpages.inria.fr/doc/#energy

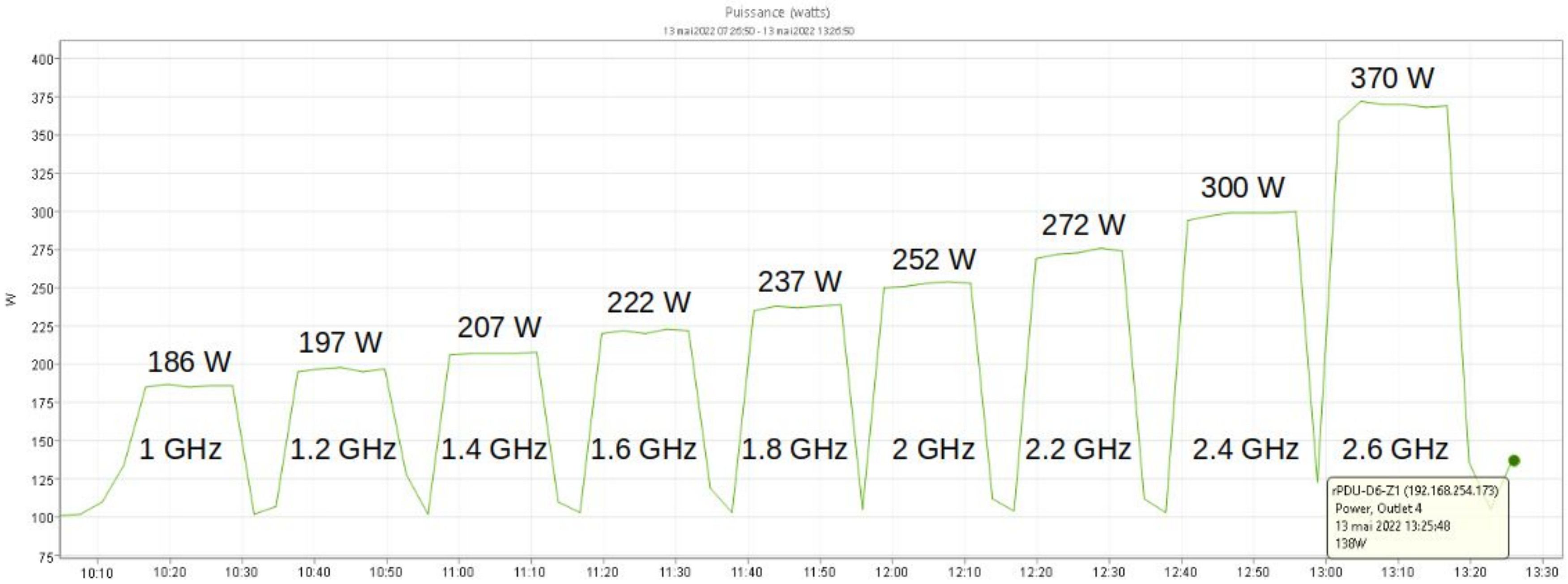
06

How you can help us

Beta-test on Formation

- The system is under beta-testing !
  - on the "Formation" cluster
  - send us an email to get registered
  - we would like any feedback to improve the system

CPU frequency and consumption



rPDU-D6-Z1 (192.168.254.173)
Power, Outlet 4
13 mai 2022 13:25:48
138W

CPU frequency and consumption

- Reducing **a little** the frequency leads to **high savings**
- You can easily reduce the CPU frequency with SLURM
  - salloc -N1 -C bora **–cpu-freq=HighM1**
  - salloc -N1 -C bora **–cpu-freq=2400000** (2.4 GHz)
- The highest the frequency, the highest the savings
  - No real benefit if the max. frequency is low
- /!\ Some machines only accepts specific frequencies
  - /sys/devices/system/cpu/cpuX/cpufreq/scaling_available_frequencies

Thank you !

Feel free to ask any question !