TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

RP-STATS QuickStart Guide

Albert Lu (alu@tacc.utexas.eud)

Last modified: 11/4/2023

Start JupyterLab with RP-Stats on Frontera

Step 1: Submit a job to start the Lab server

```
sbatch -A XXXXX /work/projects/remora/job.rp_frt_normal
```

Project account

Step 2: Look for the Jupyter server URL in jupyter.out

Step 3: Use a web browser to open the URL

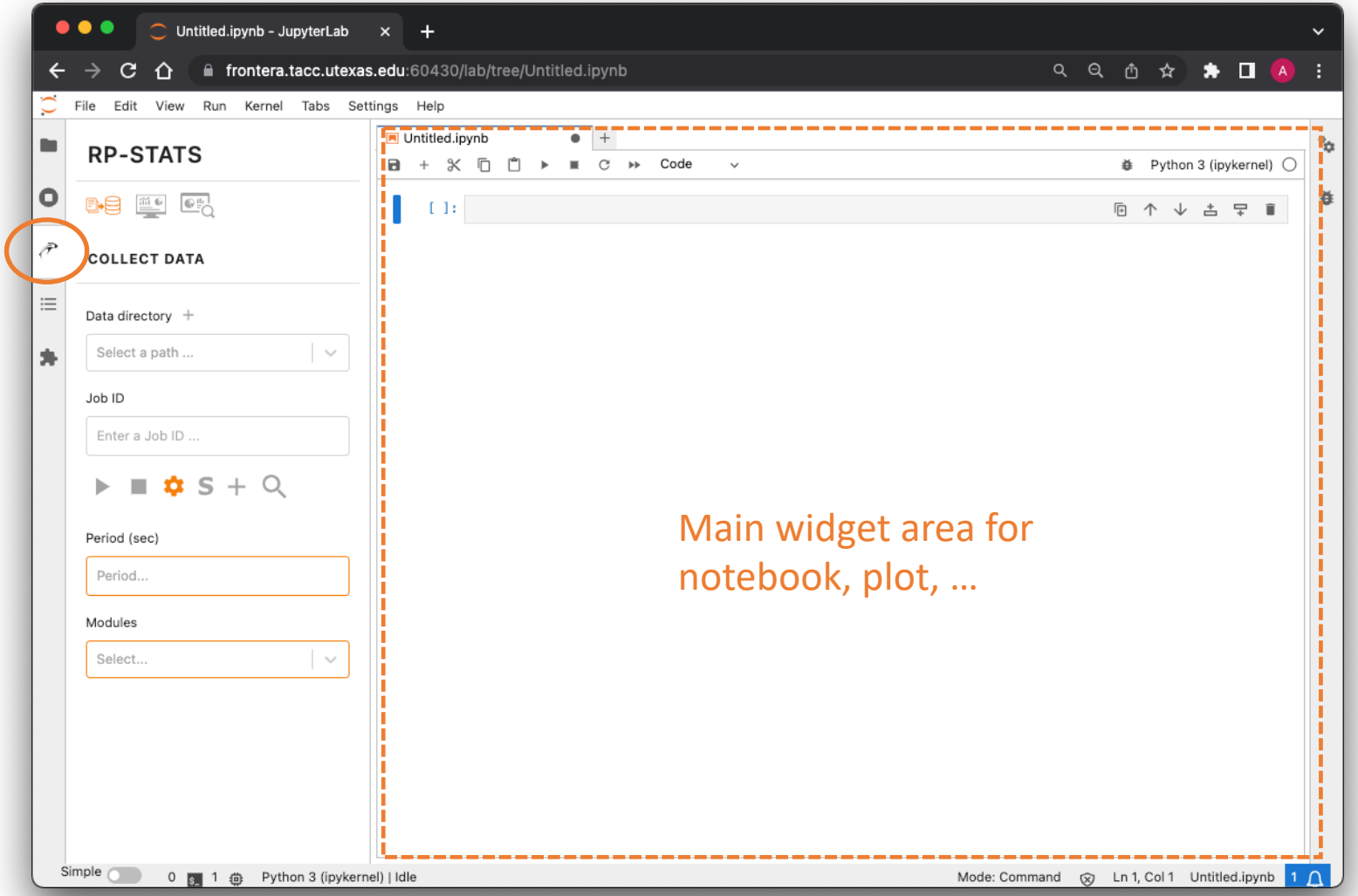
File: jupyter.out

```
TACC: using jupyter binary /work2/projects/remora/rp-frt-v2/bin/jupyter-lab
TACC: WARNING - non-system python detected. Script may not behave as expected
TACC: using token ******************************
TACC: using jupyter command: /work2/projects/remora/rp-frt-v2/bin/jupyter-lab --certfile=/tmp/*** --
config=~/.tap/jupyter_config.py --NotebookApp.token=***
TACC: got login node jupyter port *****
TACC: created reverse ports on Frontera logins
TACC: Your jupyter notebook server is now running at https://frontera.tacc.utexas.edu:*****/?token=***
```

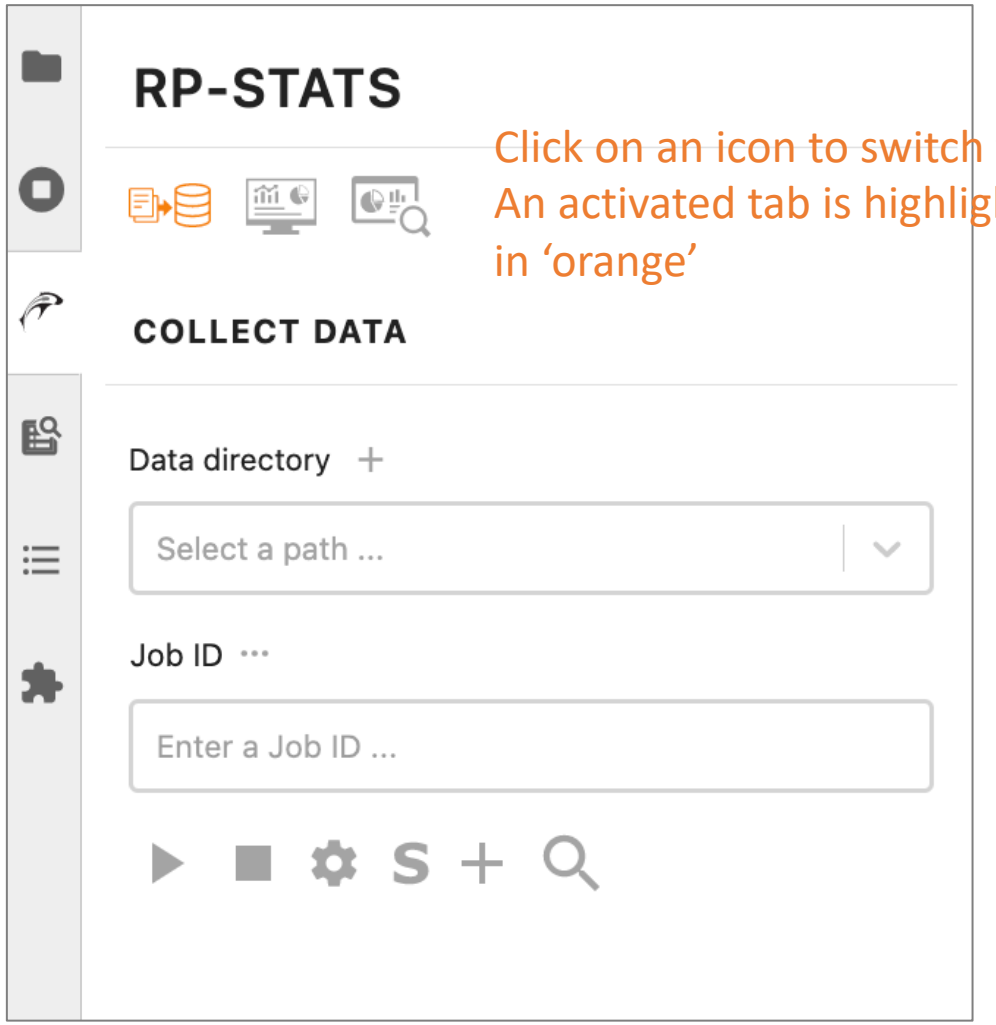
Lab URL

JupyterLab user interface

Select the remora icon to reveal the RP-Stats side panel

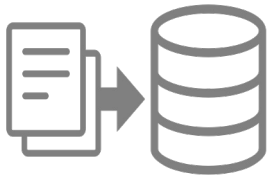


Three basic modes



Click on an icon to switch tabs
An activated tab is highlighted in ‘orange’

COLLECT DATA



Control data collections and view information

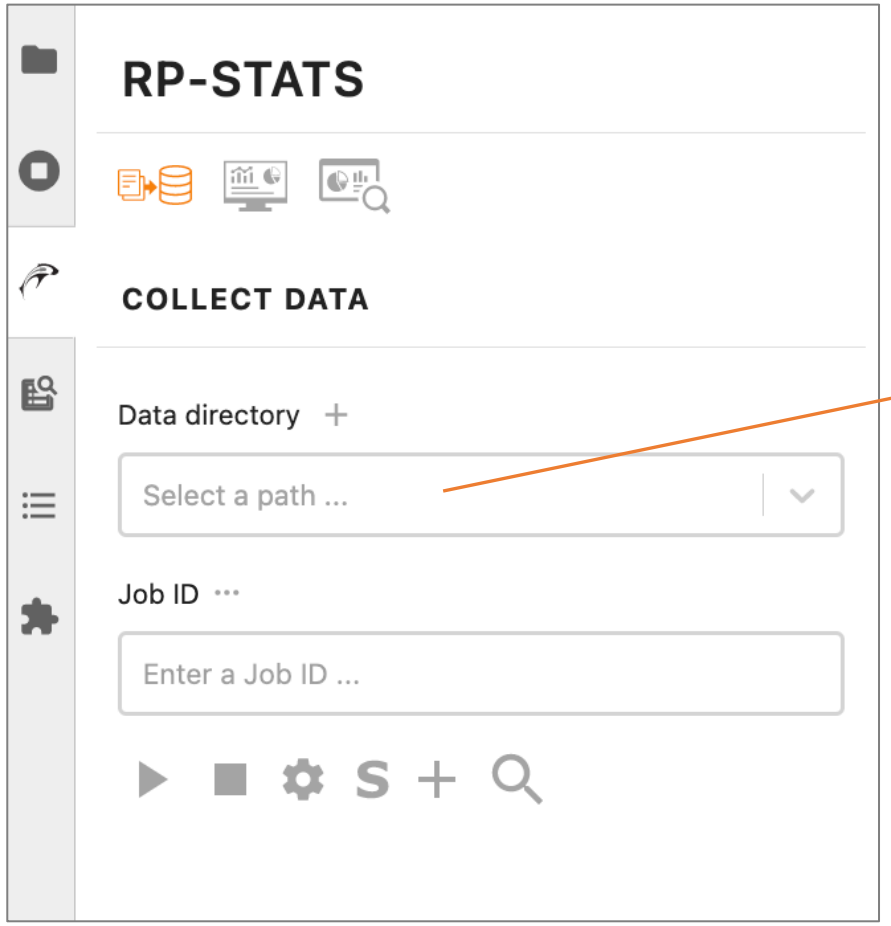
VISUALIZE DATA

Visualize collected data

ANALYZE DATA

Plot and compare job metrics

The *data_dir* directory



In RemoraPy, *data_dir* is a special directory for storing RP-Stats settings and databases

To register a new *data_dir*, first press the "+" button.
Enter the full path to the new directory and then press the return key.
Note that the name of the data directory must be "*remora_data*"

data_dir = _parent_dir_/remora_data

Required

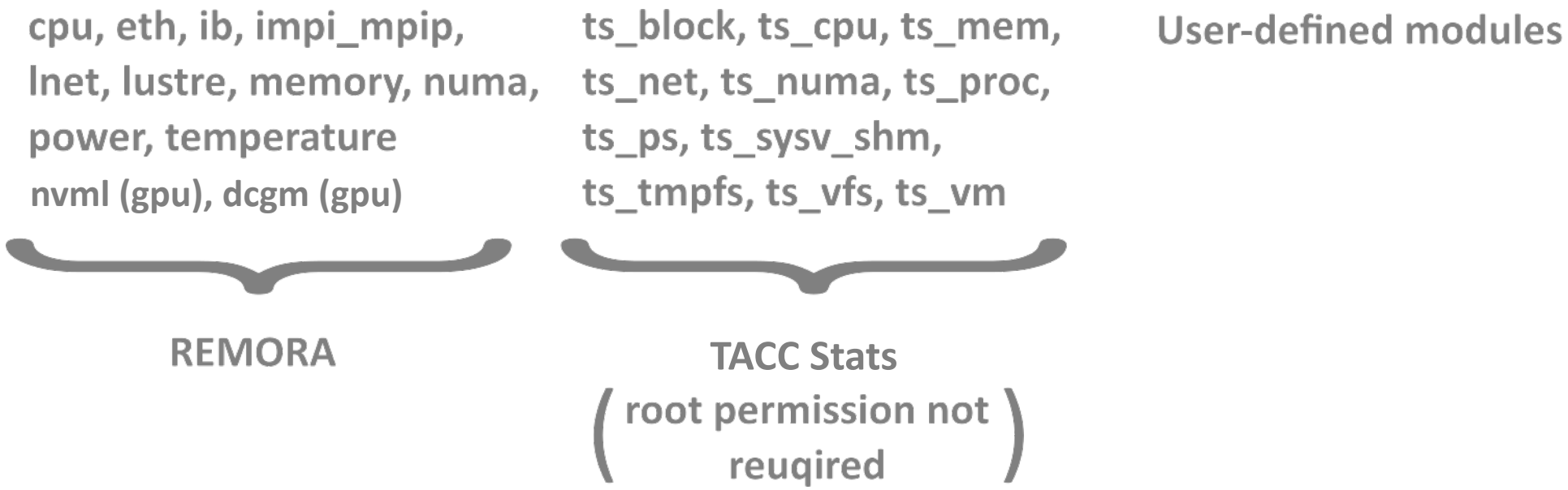
In the above example, the parent directory "_parent_dir_" must be pre-existing . Otherwise, a warning sign ⛔ will show up above the input area.

RP-Stats will create a directory named "*remora_data*" inside _parent_dir_ and copy configuration files into it.

You may have multiple *data_dir.*

Modules

In REMORA, a ***module*** is referred to as a class of functions (defined in the module script file) that is used to collect and process a specific type of performance data, such as CPU and memory utilization. RP-Stats not only supports the original REMORA module types, but can also collect TACC Stats data that does not require root privileges. RP-Stats also allows users to create their own modules.



Job metrics

In RP-Stats, a ***metric*** is referred to as either a number for job characterization (e.g., the number of nodes used) or a measure of some performance-related property (e.g., the average memory utilization). Unless otherwise specified, the reported value of a metric has been *averaged* over all compute nodes allocated for the job. RP-Stats provides three basic types of metrics: the SLURM (with names prefixed with ***sl-***), TACC Stats (***ts-***), and RemoraPy (***rp-***) metrics

sl*-metrics**: Some job information from the SLURM ***sacct command

***ts*-metrics**: The job-level metrics reported by TACC Stats

rp*-metrics**: Three metrics, (postfixed with) ***'max', ***'avg'***, and ***'delta'***, calculated for each type of data collected by REMORA modules (including user-defined modules)

Three *rp*-metrics sub-types

- **'max'** : The maximum value of a type of data seen in a job (all nodes)
- **'avg'** : The mean value calculated by averaging over all data points of a type of data in a job
- **'delta'** : The change in a type of data (end-start) averaged over nodes allocated

sl-metrics	*ts*-metrics	*rp*-metrics
sl-ElapsedRaw	ts-avg_blockbw	rp-cpu_usage_use
sl-NCPUS	ts-avg_cpi	rp-cpu_usage_sys
sl-Nnodes	ts-avg_freq	rp-cpu_usage_idle
sl-Ntasks	ts-avg_cpuusage	rp-eth_rx_bytes
sl-AveCPU	ts-avg_ethbw	rp-eth_rx_packets
sl-MinCPU	ts-avg_fabricbw	rp-eth_tx_bytes
sl-SystemCPU	ts-avg_flops_64b	rp-eth_tx_dropped
sl-UserCPU	ts-avg_flops_32b	rp-eth_tx_errors
⋮	⋮	⋮

Ways to start RemoraPy

Use RP-Stats dashboard

Job ID
XXXXXX

Use *job_id*

Use *path* to a job script

Run *remorapy* in command lines or in a job script

```
module load remorapy
remorapy my_app
```

Load *remorapy* module and then use wrapper script *remorapy* to start the application

Use Python script

```
import RemoraPy
rp=RemoraPy(...)
rp.RunRemora(...)
```

Batch mode

Use *remorapy* wrapper in a job script to start an application

Sample job script

```
ml gromacs/2022.1
ml python3/3.9.2
ml use /work/projects/remora/rp-frt-v2/share/modulefiles
ml remorapy

RP_TEST_DIR=`pwd`
RP_WORK_DIR=${RP_TEST_DIR}
RP_TEST_INPUT_DIR=${RP_TEST_DIR}/input


export OMP_NUM_THREADS=1
export REMORAPY_PERIOD=5
export REMORAPY_MODULE_LIST=full
export REMORAPY_WORK_DIR =${RP_WORK_DIR}

remorapy ibrun gmx_mpi mdrun -s ${RP_TEST_INPUT_DIR}/benchPEP.tpr -deffnm md -g md.log -nsteps 1000
```

Load remorapy module

Set REMORAPY_WORK_DIR so RP-STATS will use ${REMORAPY_WORK_DIR}/remora_data for data_dir

Examples of setting REMORAPY_MODULE_LIST :

REMORAPY_MODULE_LIST="cpu,mem,ib,impi_mpip"

REMORAPY_MODULE_LIST=basic (only cpu and mem)

REMORAPY_MODULE_LIST=full (everything)

Set REMORAPY environment variables

Use the script “remorapy” to launch your application

Python mode – use RemoraPy in a Python script

Minimal RP initialization steps:

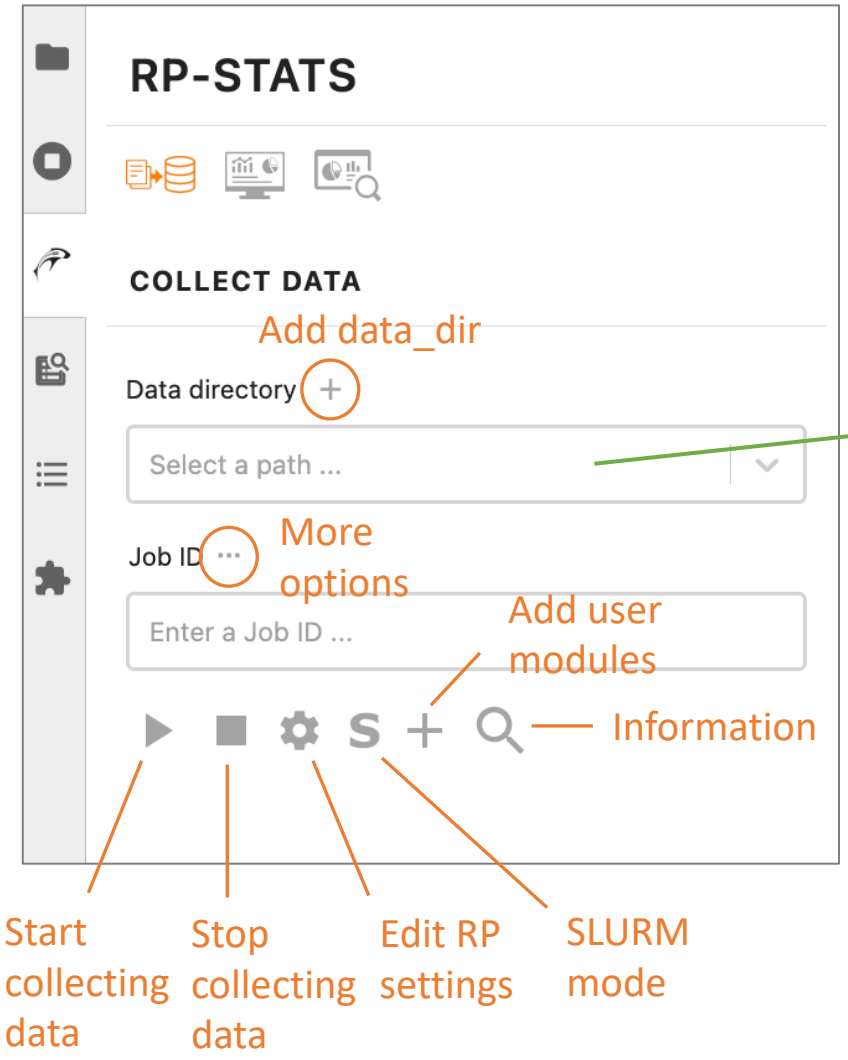
```
from remorapy.remorapy import RemoraPy

work_dir="....."
rp=RemoraPy(remorapy_work_dir=work_dir)
```

Access to all RemoraPy functions
Need a Jupyter notebook interface for visualization

Some top-level functions:

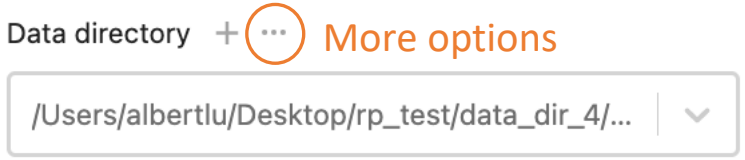
```
rp.RunRemora(job_id, module_list=['cpu','memory'],
                    period=None, batch_mode=False, command=None, debug=False)

rp.StopRemora(job_id, data_dir=None, debug=False)

rp.PostCrash(job_id, data_dir=None, debug=False)

rp.Plot(module_type, plot_mode, plot_type, job_id=None,
            job_remora_dir=None, local_port=8888, is_dashboard=False)

rp.AddRPModule(module_name, commands, n_data)

rp.DeleteRPModule(module_name)
```

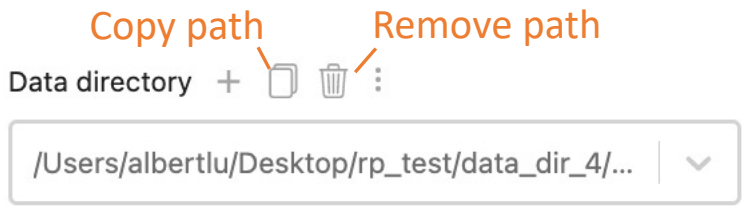
The data collector tab



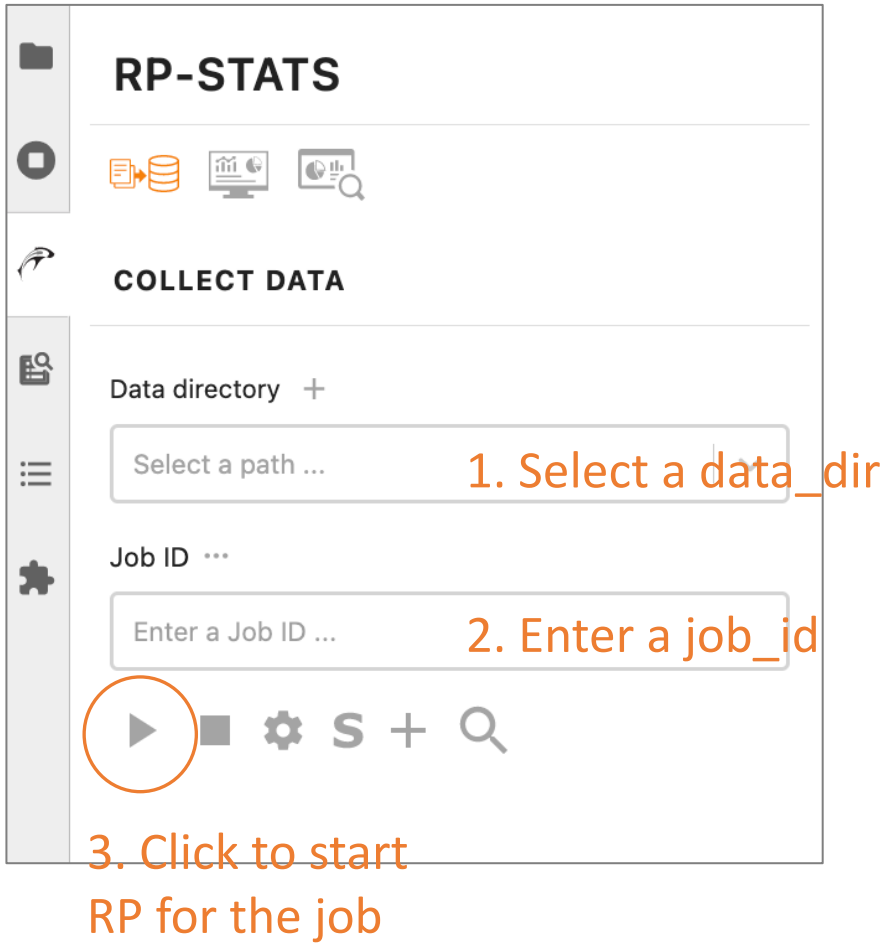
More options button

When a data_dir is selected:



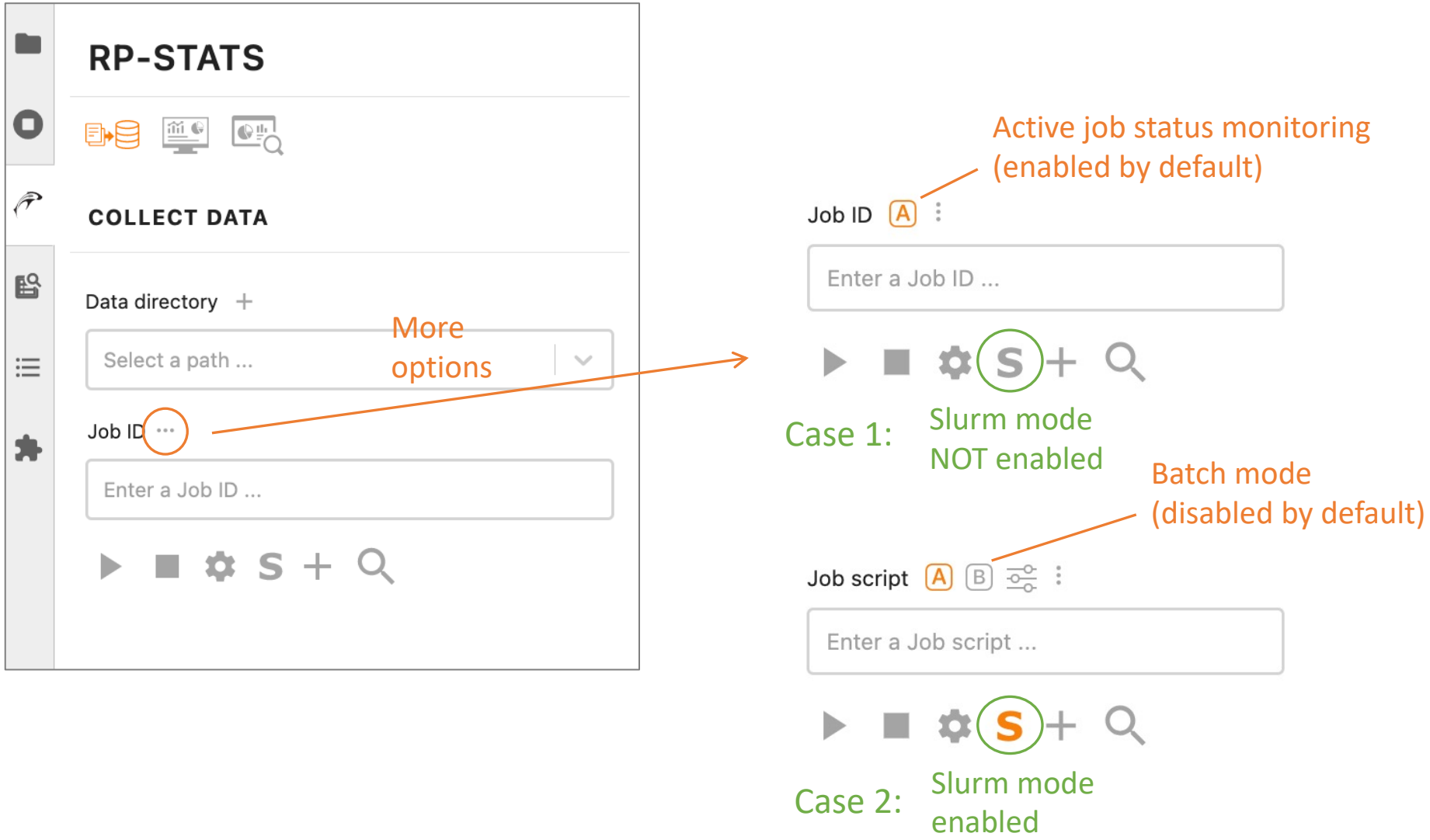


Start data collection on a running job

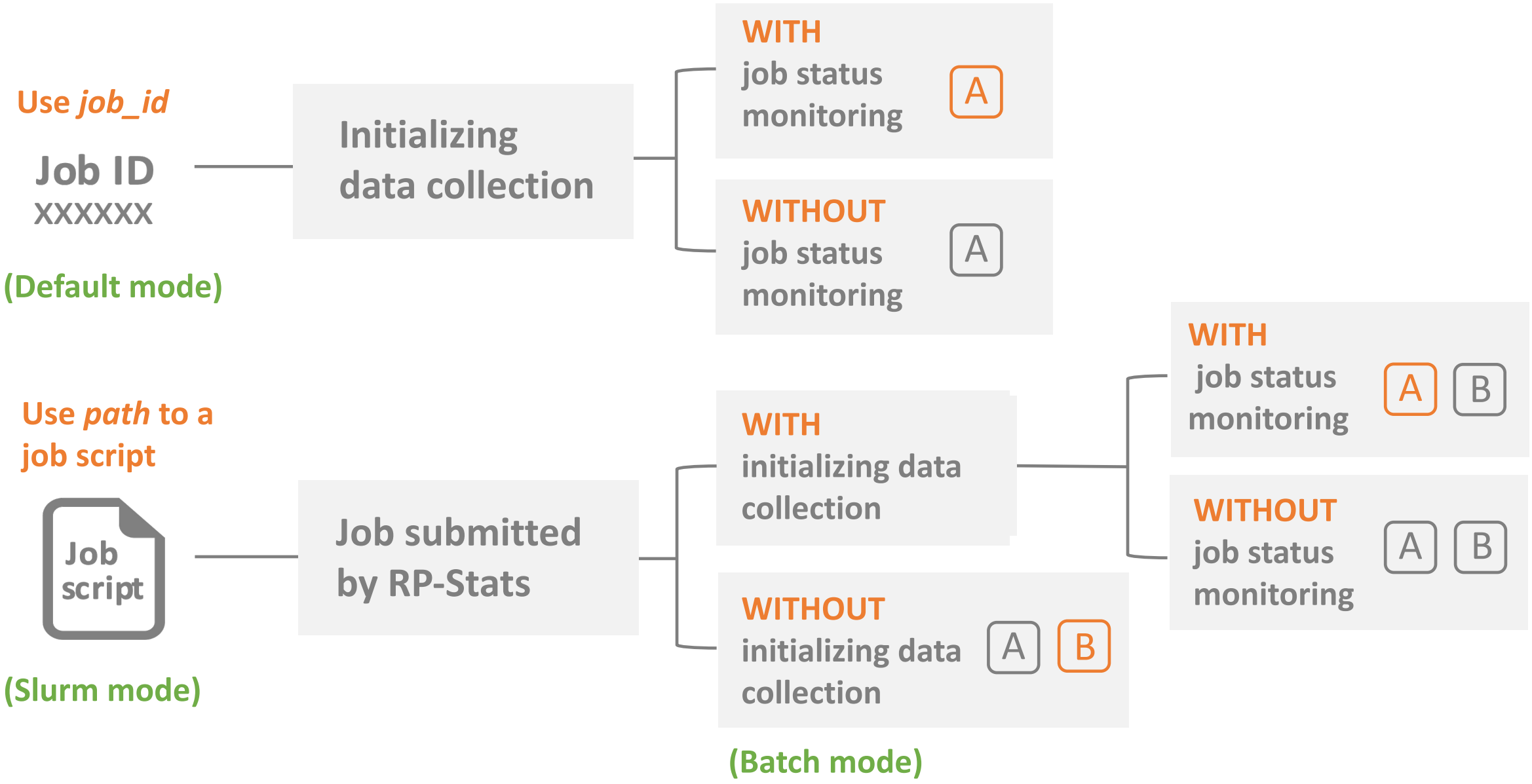


Three steps to start (default)
data collection using Job ID

More control options for data collection



Different approaches to start data collection in RP-Stats



Different approaches to start data collection in RP-Stats

Method	Job status monitoring	Use remorapy wrapper	JLab is required for	Includes post-job data processing	Needs post-crash data processing	Supports LD_PRELOAD	Compiler dependent	Button status
Use Job ID (Default)	ON	NO	Post-job data processing	YES	NO	NO	NO	A
	OFF	NO	Initializing RP	NO	YES	NO	NO	A
Use Job script path (Slurm)	ON	NO	Post-job data processing	YES	NO	YES	NO	A B
	OFF	NO	Initializing RP	NO	YES	YES	NO	A B
Commands in Job script (Batch)	--	YES	NO	YES	NO	YES	YES	A B

The setting button 

Period (sec)

Period...

Modules

Select...

Clear all modules

Select all RP modules

Select all TS modules

SAVE LOAD CLEAR ALL-RP ALL-TS

Save setting (period & modules)

Load a saved setting

Period (sec)

5

Modules

Select...

CPU

ETH

IB

LNET

LUSTRE

Period (sec)

5

Modules

TACCSTATS × CPU × ETH ×

TaccStats Modules

Select...

TS_BLOCK

TS_CPU

TS_MEM

TS_NET

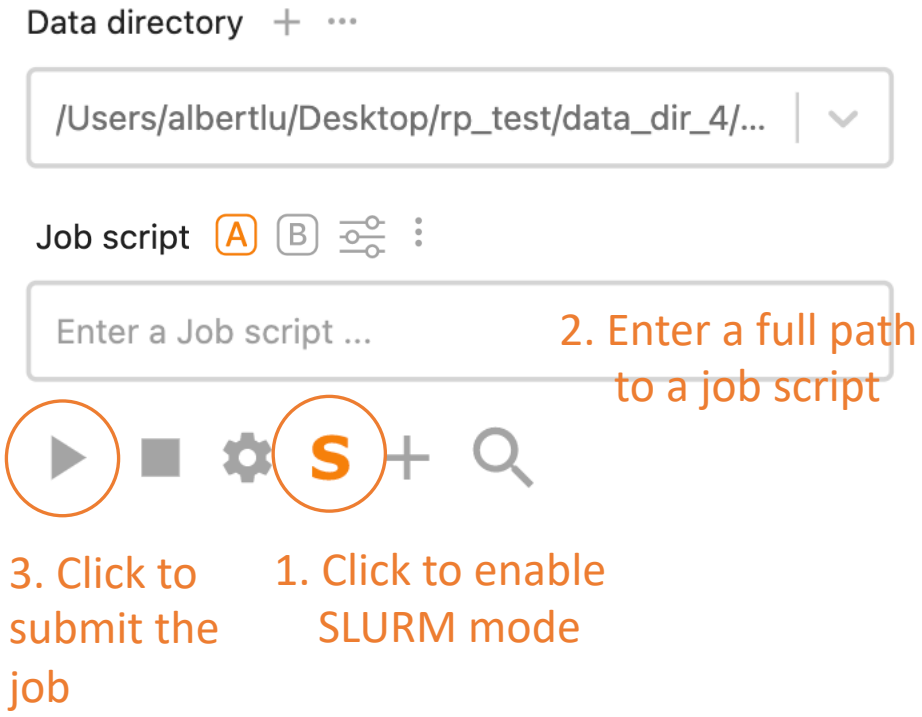
TS_NUMA

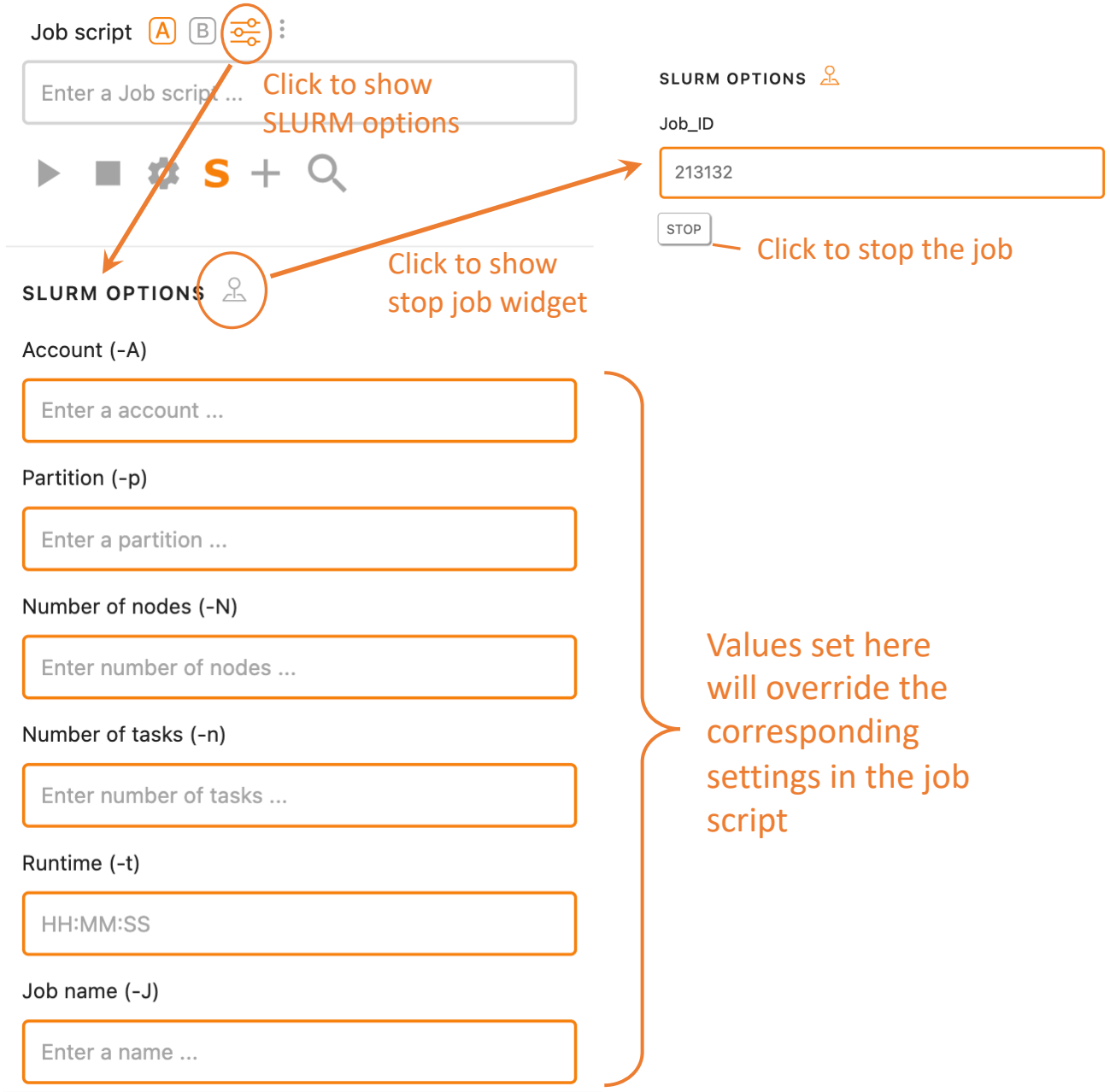
TS_PROC

This list appears if the TACCSTATS module is selected above

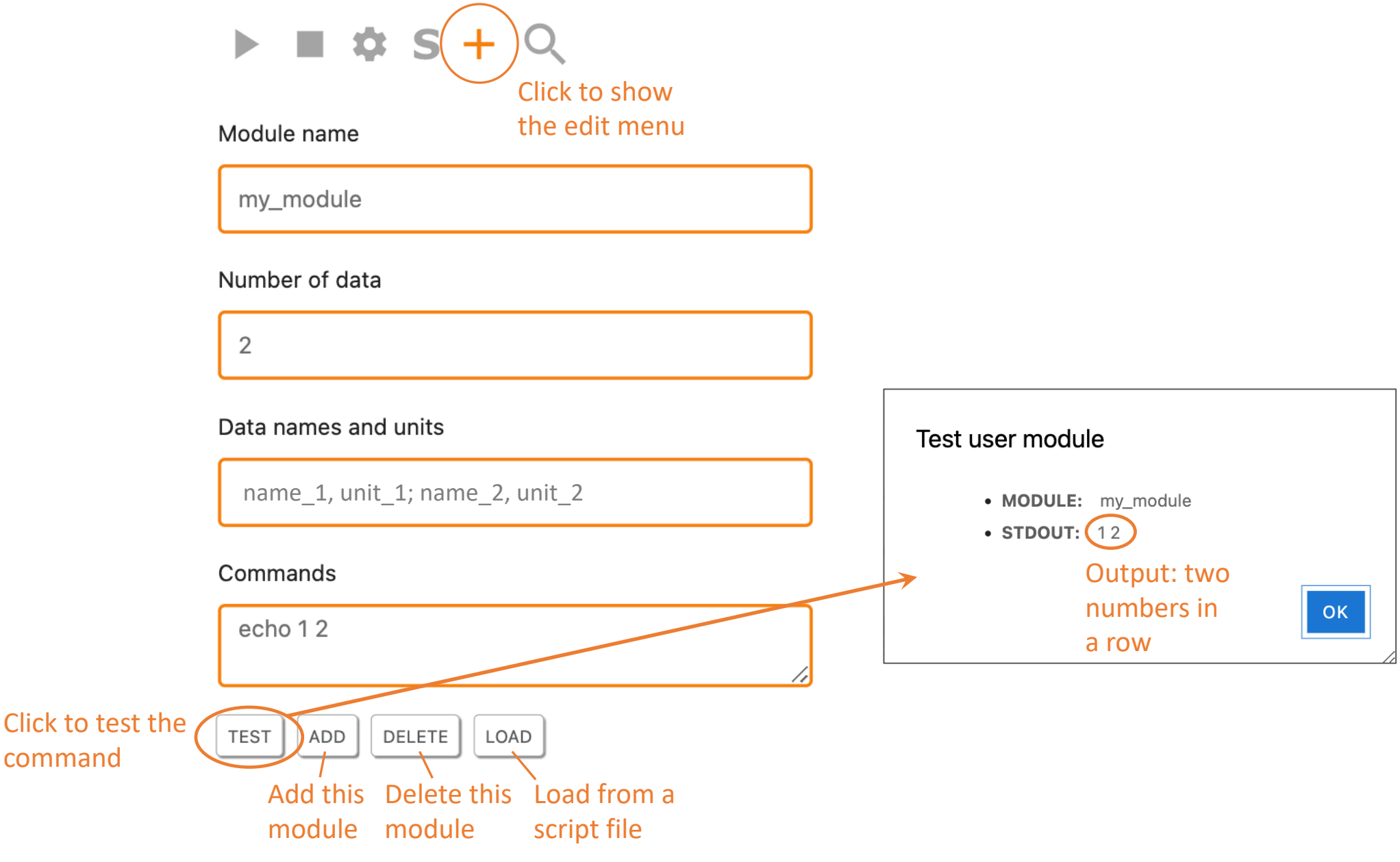
The SLURM mode button S

Three steps to launch a job and collect performance data



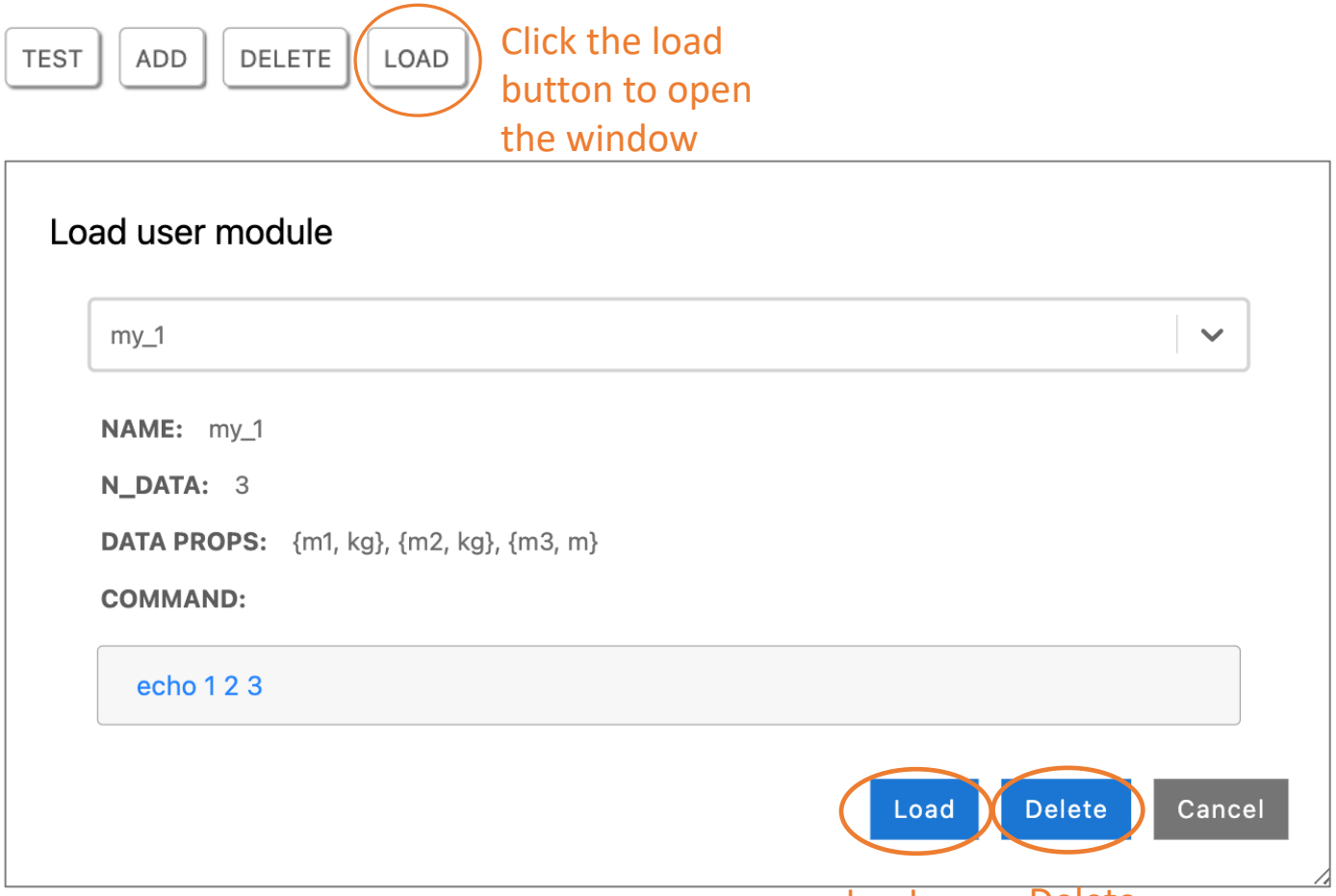


The add user-defined module button +



Load user-defined module file

Load user module from a script file



User scripts saved under the following directory will appear in the drop-down selection list

{data_dir}/modules/user_modules/

Script format

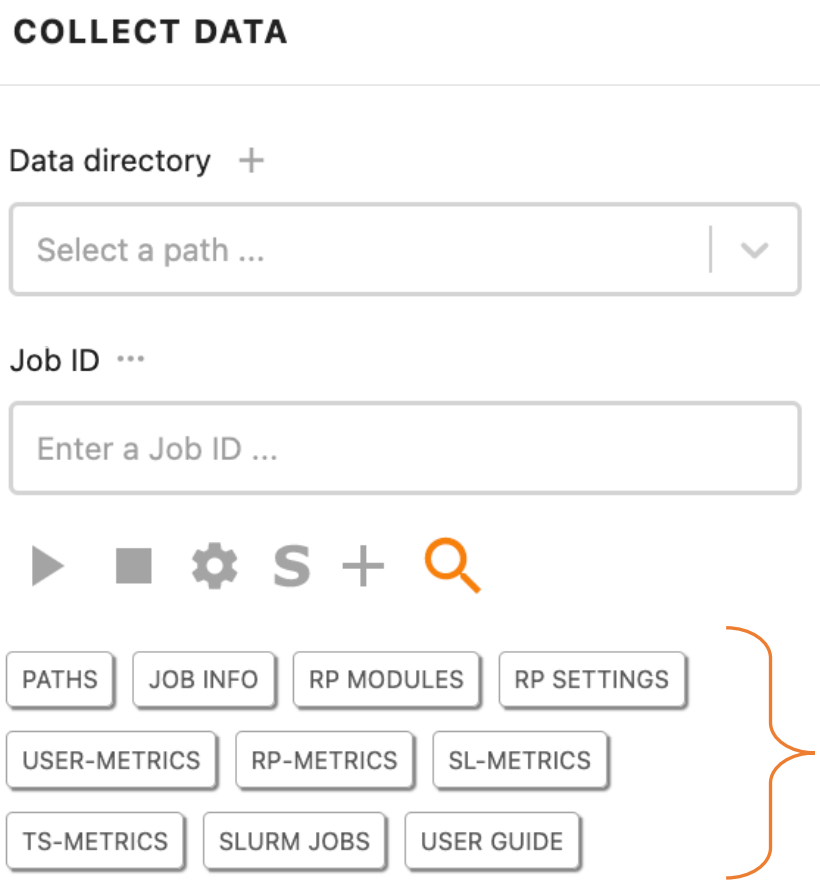
File name: module_name

File syntax:

```
# module_name
# name_1, unit_1; name_2, unit_2 ...
command_1
command_2
command_3 ...
```

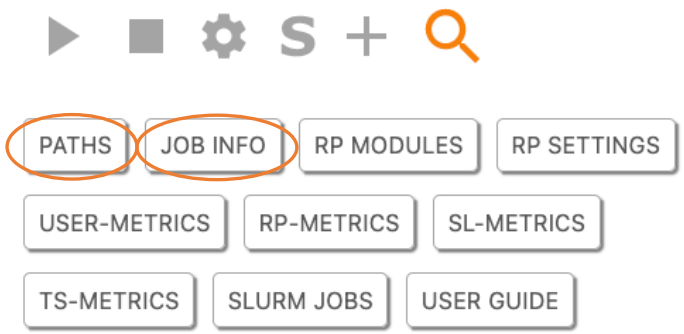
File output: *N* numbers in a row

The Information button



Select a button to access the window containing detailed information about paths, jobs, modules, user settings, metrics, SLURM job status, and to browse the user guide

The paths and job info button



PATHS

Saved RP data_dir

Path	#Jobs	Misc
/Users/albertlu/Desktop/rp_test/data_dir_1/remora_data	4	ⓘ
/Users/albertlu/Desktop/rp_test/data_dir_2/remora_data	3	ⓘ
/Users/albertlu/Desktop/rp_test/data_dir_3/remora_data	39	ⓘ
/Users/albertlu/Desktop/rp_test/data_dir_4/remora_data	6	ⓘ

All registered data_dirs

Info

JOB INFO

Data directory

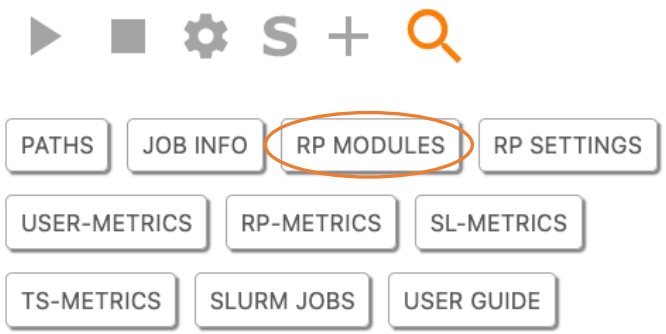
/Users/albertlu/Desktop/rp_test/data_dir_4/remora_data

Job_ID	Status	Misc
10927726	Completed	ⓘ Ⓜ Ⓔ
10927727	Completed	ⓘ Ⓜ Ⓔ
10927728	Completed	ⓘ Ⓜ Ⓔ
10927729	Completed	ⓘ Ⓜ Ⓔ
10927730	Completed	ⓘ Ⓜ Ⓔ
11179319	Completed	ⓘ Ⓜ Ⓔ

All jobs under the selected data_dir

Info Modules Envs

The RP modules info button



RP MODULES

RP data directory

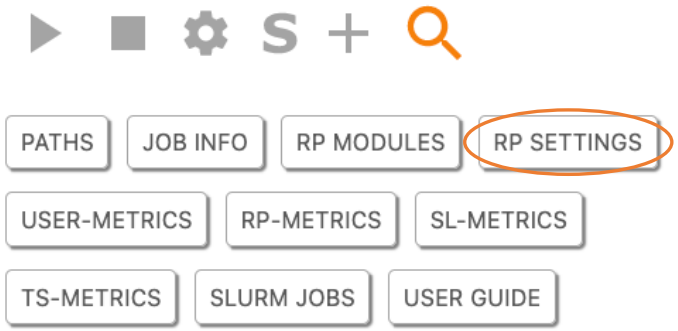
/Users/albertlu/Desktop/rp_test/data_dir_4/remora_data

Name	Category	Misc
cpu	CPU	
eth	NETWORK	
ib	NETWORK	
impi_mpip	MPI	
lnet	IO	
lustre	IO	
memory	MEMORY	
numa	NUMA	
temperature	TEMPERATURE	
taccstats	EXTERNAL	
my_2	USER	
my_module	USER	

All modules (default + user's) under the selected *data_dir*

Info / Source code

The RP settings info button



All user RP settings (period + modules) saved under the selected *data_dir*

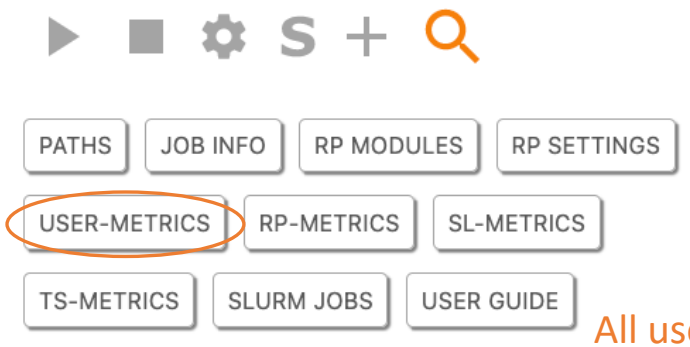
RP SETTINGS

RP data directory

/Users/albertlu/Desktop/rp_test/data_dir_4/remora_data

Name	Period(sec)	RP modules	TS modules
test	3	eth, lustre	
test-a	5	eth, lustre, memory, numa, temperature	
test-b	5	eth, lustre, memory, numa, temperature, lnet, taccstats	ts_numa, ts_proc
test-c	2	eth, impi_mpip, taccstats	ts_block, ts_mem
my_setting	1	cpu, eth, ib, impi_mpip, taccstats	ts_block, ts_sysv_shm, ts_numa

The user-metrics info button



All user-defined metrics saved under the selected *data_dir*

USER-METRICS

RP data directory

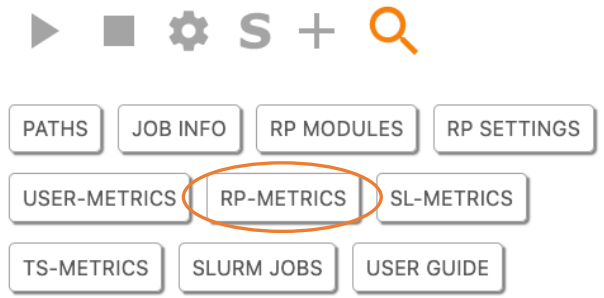
/Users/albertlu/Desktop/rp_test/data_dir_4/remora_data

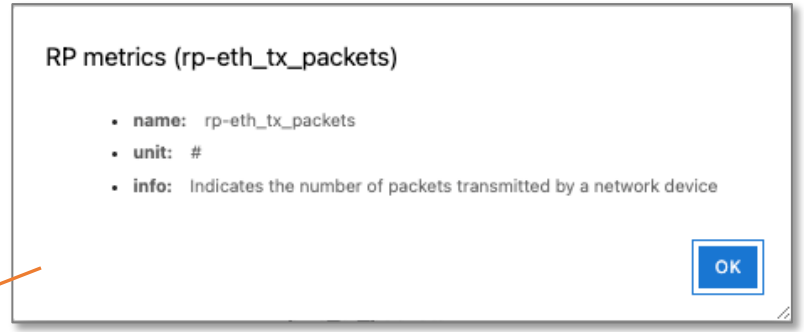
Name	Unit	Equation
plus_one	a.u.	{job_id}+1
plus_two	a.u.	{job_id}+2
plus_3	cm	{job_id}+3
plus_4	m	{job_id}+4

The RP metrics info button

RP-METRICS

Name	Unit	Misc
rp-cpu_usage_usr	%	ⓘ
rp-cpu_usage_sys	%	ⓘ
rp-cpu_usage_idle	%	ⓘ
rp-eth_rx_bytes	bytes	ⓘ
rp-eth_rx_crc_errors	#	ⓘ
rp-eth_rx_frame_errors	#	ⓘ
rp-eth_rx_packets	#	ⓘ
rp-eth_tx_bytes	bytes	ⓘ
rp-eth_tx_dropped	#	ⓘ
rp-eth_tx_errors	#	ⓘ
rp-eth_tx_packets	#	ⓘ
rp-ib_rx_bytes	bytes	ⓘ
rp-ib_rx_packets	#	ⓘ
rp-ib_tx_bytes	bytes	ⓘ
rp-ib_tx_packets	#	ⓘ
rp-lnet_drop_count	#	ⓘ
rp-lnet_drop_length	bytes	ⓘ
rp-lnet_errors	#	ⓘ
rp-lnet_msgs_alloc	#	ⓘ





Select a button to show details

The SLURM metrics info button

PATHS | JOB INFO | RP MODULES | RP SETTINGS

USER-METRICS | RP-METRICS | SL-METRICS

TS-METRICS | SLURM JOBS | USER GUIDE

SL-METRICS

Name	Unit	Misc
sl-ElapsedRaw	seconds	ⓘ
sl-NCPUS	cpus	ⓘ
sl-NNodes	nodes	ⓘ
sl-NTasks	tasks	ⓘ
sl-AveCPU	seconds	ⓘ
sl-MinCPU	seconds	ⓘ
sl-SystemCPU	seconds	ⓘ
sl-UserCPU	seconds	ⓘ
sl-TotalCPU	seconds	ⓘ
sl-CPUTimeRAW	seconds	ⓘ
sl-AveCPUFreq	kHz	ⓘ
sl-AveVMSize	bytes	ⓘ
sl-MaxVMSize	bytes	ⓘ
sl-MaxRSS	bytes	ⓘ
sl-MaxPages	pages	ⓘ
sl-AveDiskRead	bytes	ⓘ
sl-MaxDiskRead	bytes	ⓘ
sl-AveDiskWrite	bytes	ⓘ
sl-MaxDiskWrite	bytes	ⓘ

Slurm metrics (sl-CPUTimeRAW)

- **name:** sl-CPUTimeRAW
- **unit:** seconds
- **info:** Time used (Elapsed time * CPU count) by a job or step in cpu-seconds.

OK

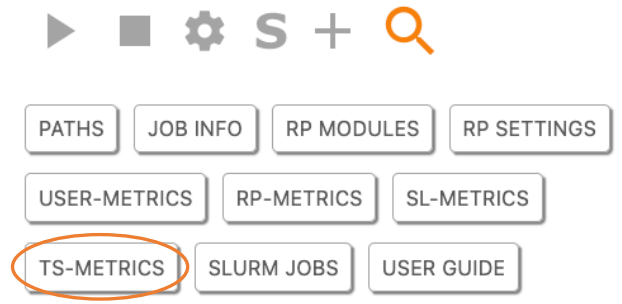
Select a button to show details

The TACC Stats metrics info button

TS-METRICS

Name	Unit	Misc
ts-avg_blockbw	MB/s/node	ⓘ
ts-avg_cpi	cycles/instr	ⓘ
ts-avg_freq	MHz	ⓘ
ts-avg_cpuusage	cores/node	ⓘ
ts-avg_ethbw	MB/s/node	ⓘ
ts-avg_fabricbw	MB/s/node	ⓘ
ts-avg_flops_64b	GFLOPs/s/node	ⓘ
ts-avg_flops_32b	GFLOPs/s/node	ⓘ
ts-avg_l1loadhits	#/s/node	ⓘ
ts-avg_l2loadhits	#/s/node	ⓘ
ts-avg_llcloadhits	#/s/node	ⓘ
ts-avg_lnetbw	MB/s/node	ⓘ
ts-avg_gpuutil	%	ⓘ
ts-avg_lnetmsgs	#/s/node	ⓘ
ts-avg_loads	loads/s/node	ⓘ
ts-avg_mbw	GB/s/node	ⓘ
ts-avg_mcdrambw	GB/s/node	ⓘ
ts-avg_mdcreqs	#/s/node	ⓘ
ts-avg_mdcwait	microseconds	ⓘ
ts-avg_openclose	#/s/node	ⓘ
ts-avg_oscreqs	#/s/node	ⓘ

Select a button to show details



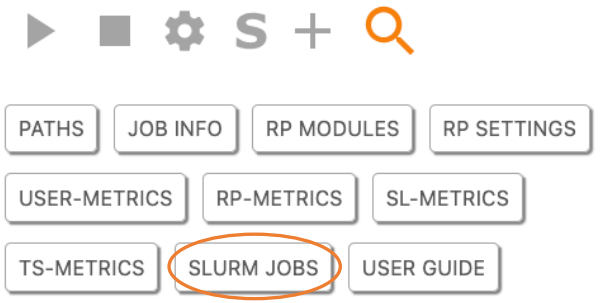
TaccStats metrics (ts-avg_llcloadhits)

- **name:** ts-avg_llcloadhits
- **unit:** #/s/node
- **info:** https://github.com/TACC/tacc_stats/blob/master/tacc_stats/analysis/metrics/metrics.py#L190
- **code:**

```
class avg_llcloadhits():
  def compute_metric(self, u):
    schema, _stats = u.get_type("pmc")
    load_names = ['LOAD_OPS_LLC_HIT', 'MEM_UOPS_RETIRED_LLC_HIT_LOADS']
    loads = 0
    for hostname, stats in _stats.items():
      for eventname in schema:
        if eventname in load_names:
          index = schema[eventname].index
          loads += stats[-1, index] - stats[0, index]
    return loads/(u.dt*u.nhosts)
```

OK

The SLURM jobs status button



SLURM JOBS STATUS

Slurm job status

Host: https://stampede2.tacc.utexas.edu:60102/

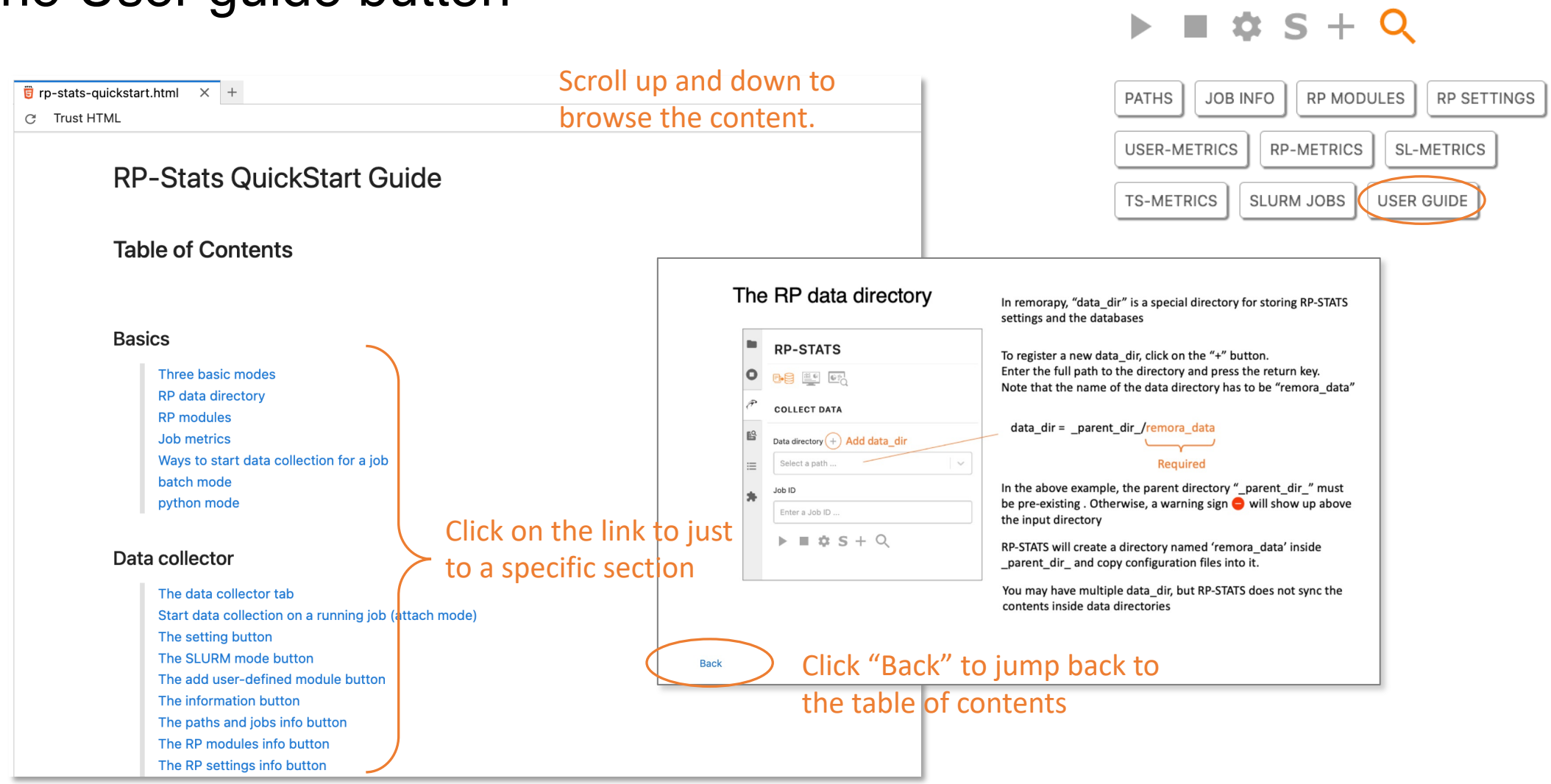
Job_id	Name	Partition	Staus	T_used	Nodes	RP
11199635	tap_jupyter	skx-normal	R	57:39	1	
11200260	test-1	normal	R	0:44	1	

All your jobs are listed in the status table

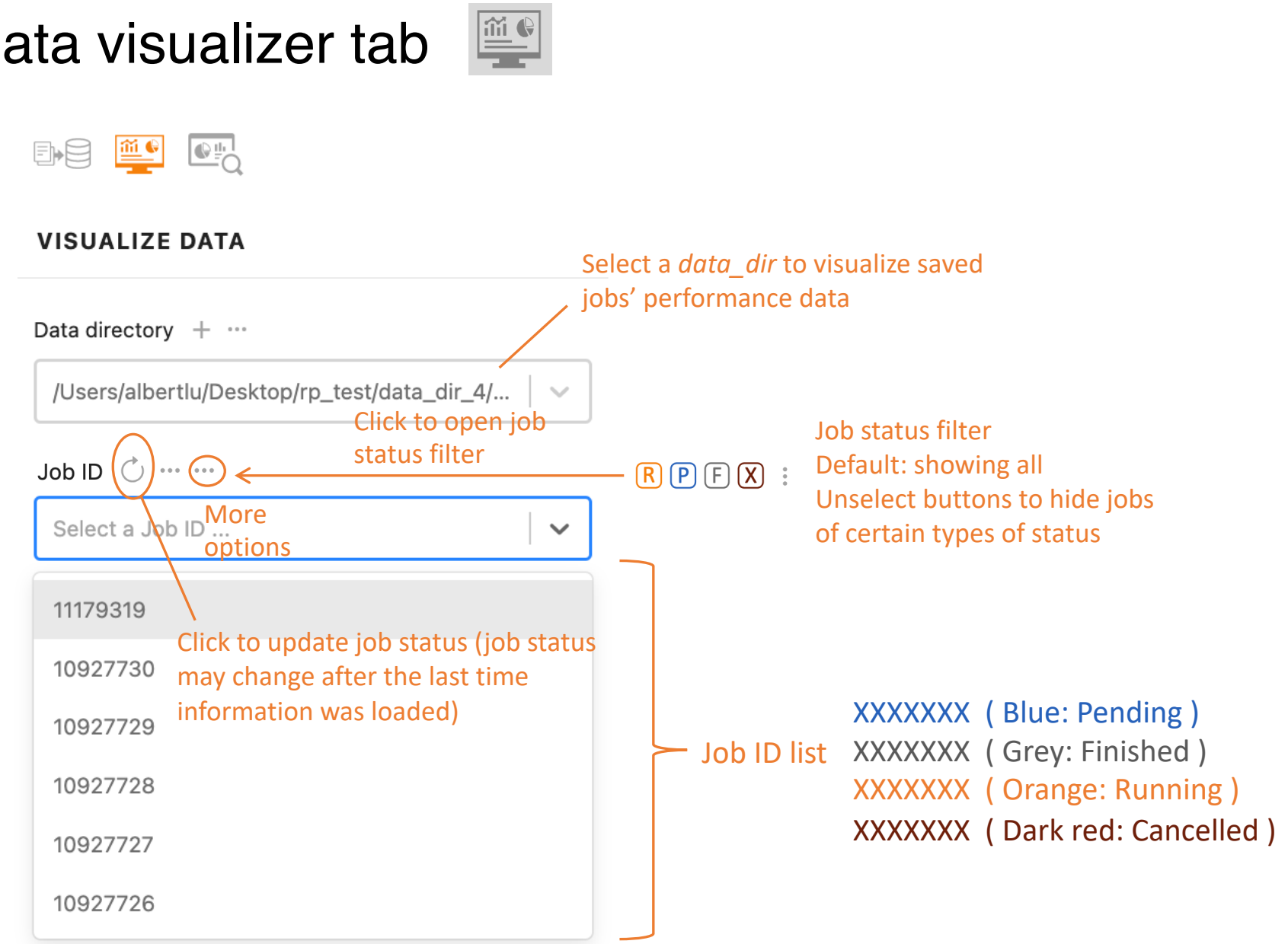
Double-click to show job details

A blinking orange circle indicates RP is running with the job

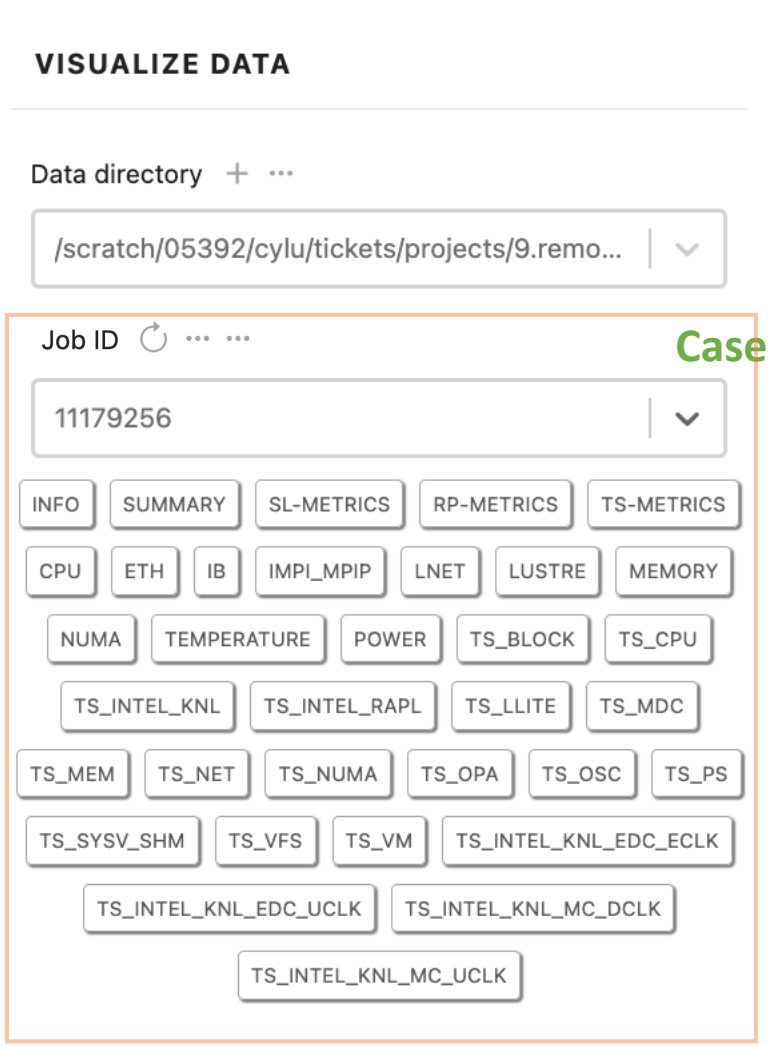
The User guide button



The data visualizer tab



Job info and module buttons



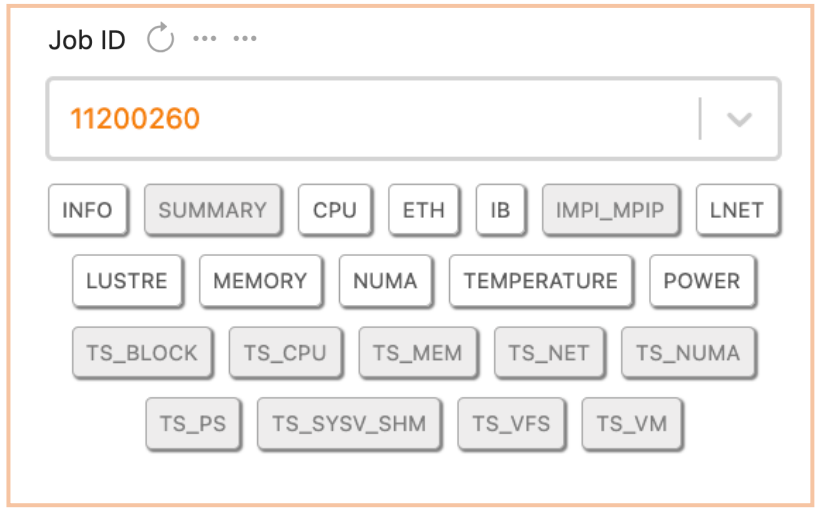
Case 1 : finished job

Information

Plots of REMORA data

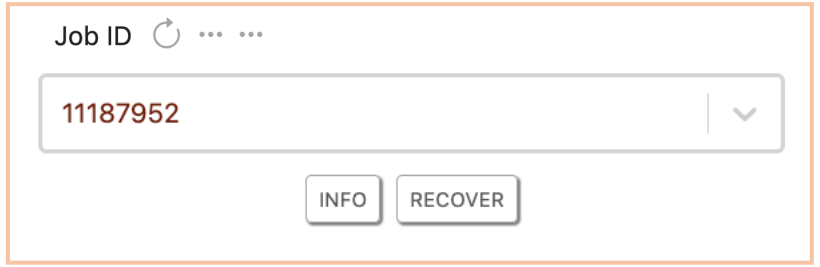
Plots of TACC Stats data (TS_xxx)

Case 2 : running job



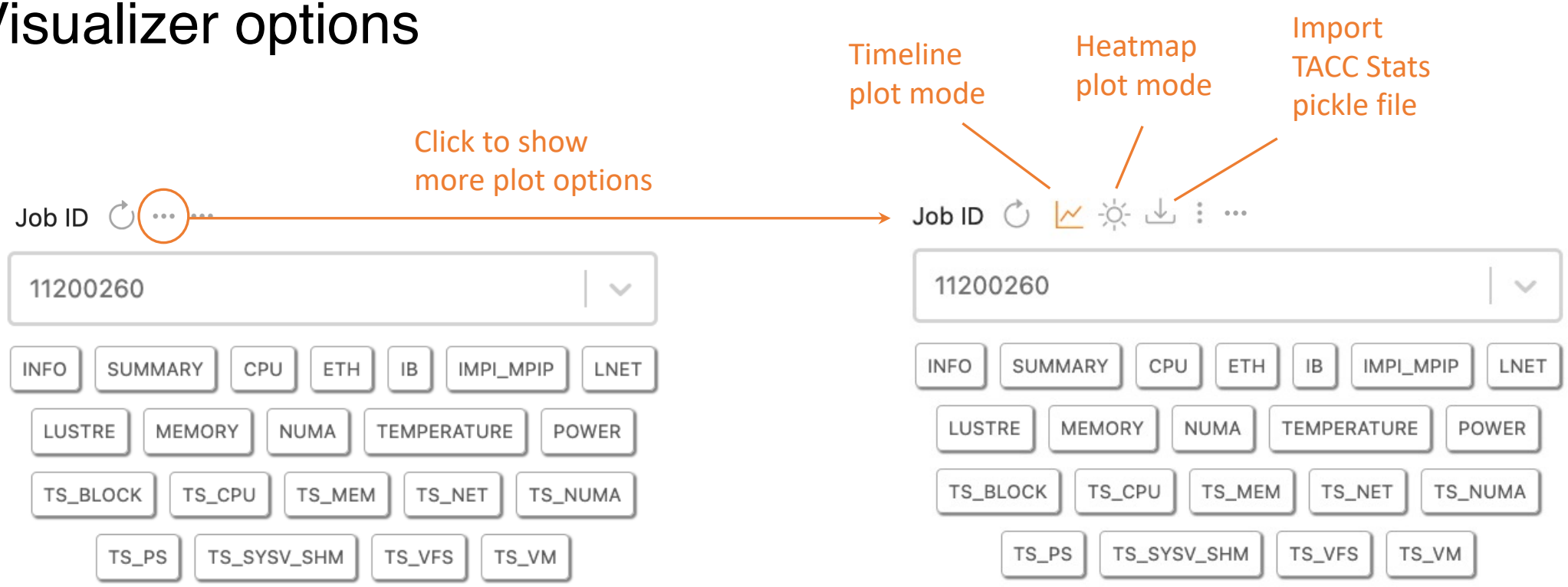
Unavailable functions are shown in grey

Case 3 : cancelled job

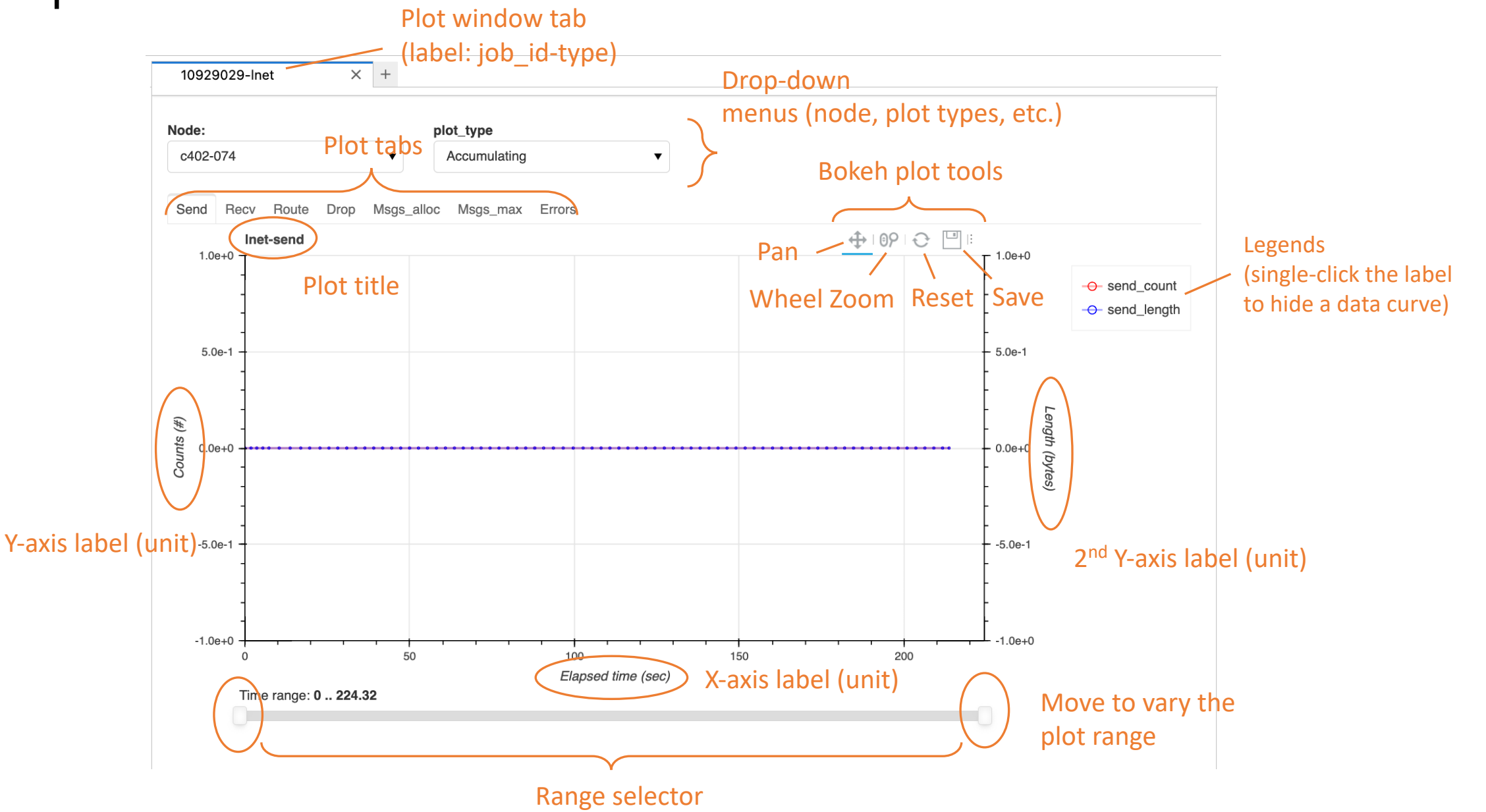


Press the “RECOVER” button and REMORA will try to recover data for visualization

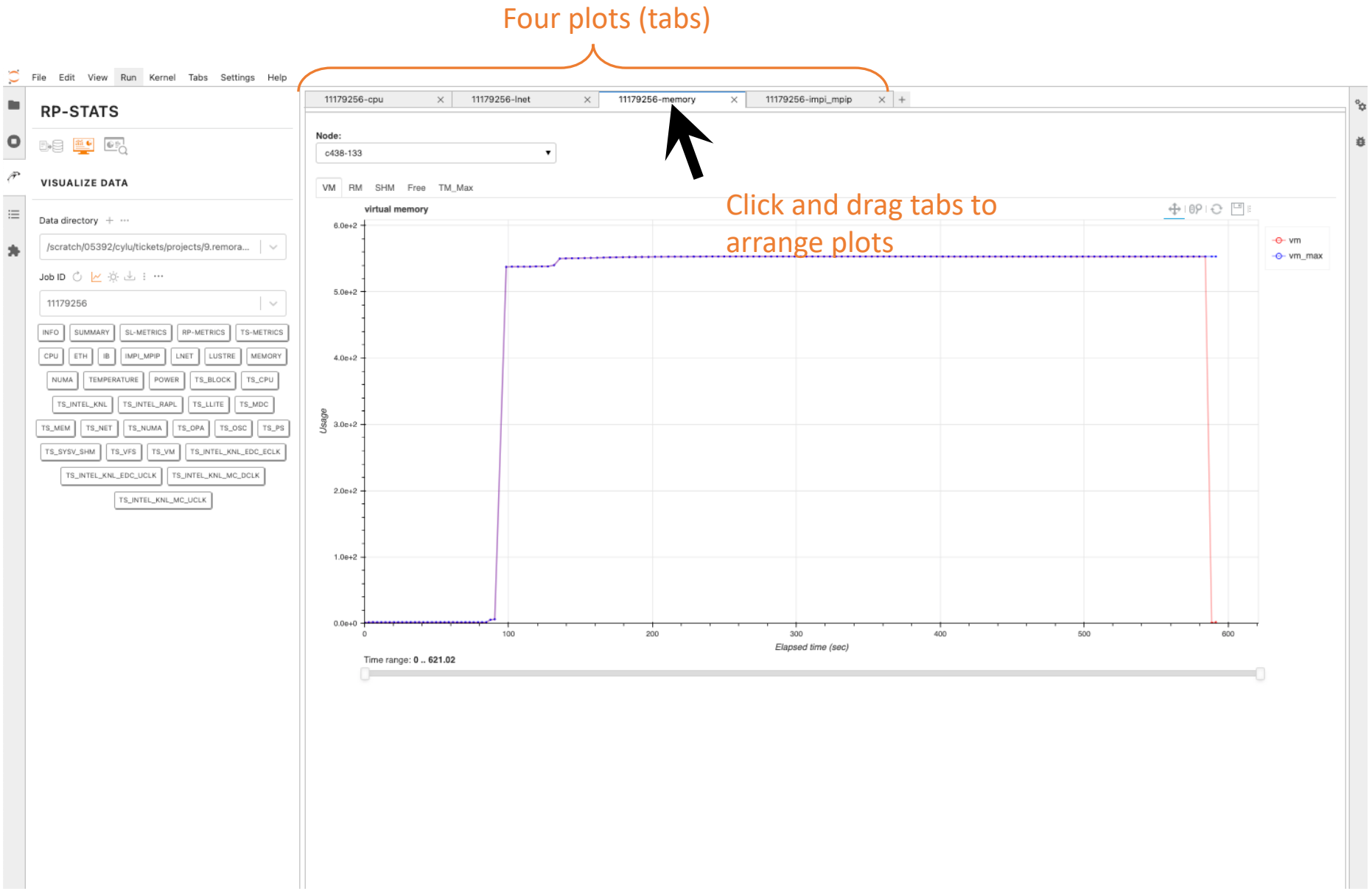
Visualizer options

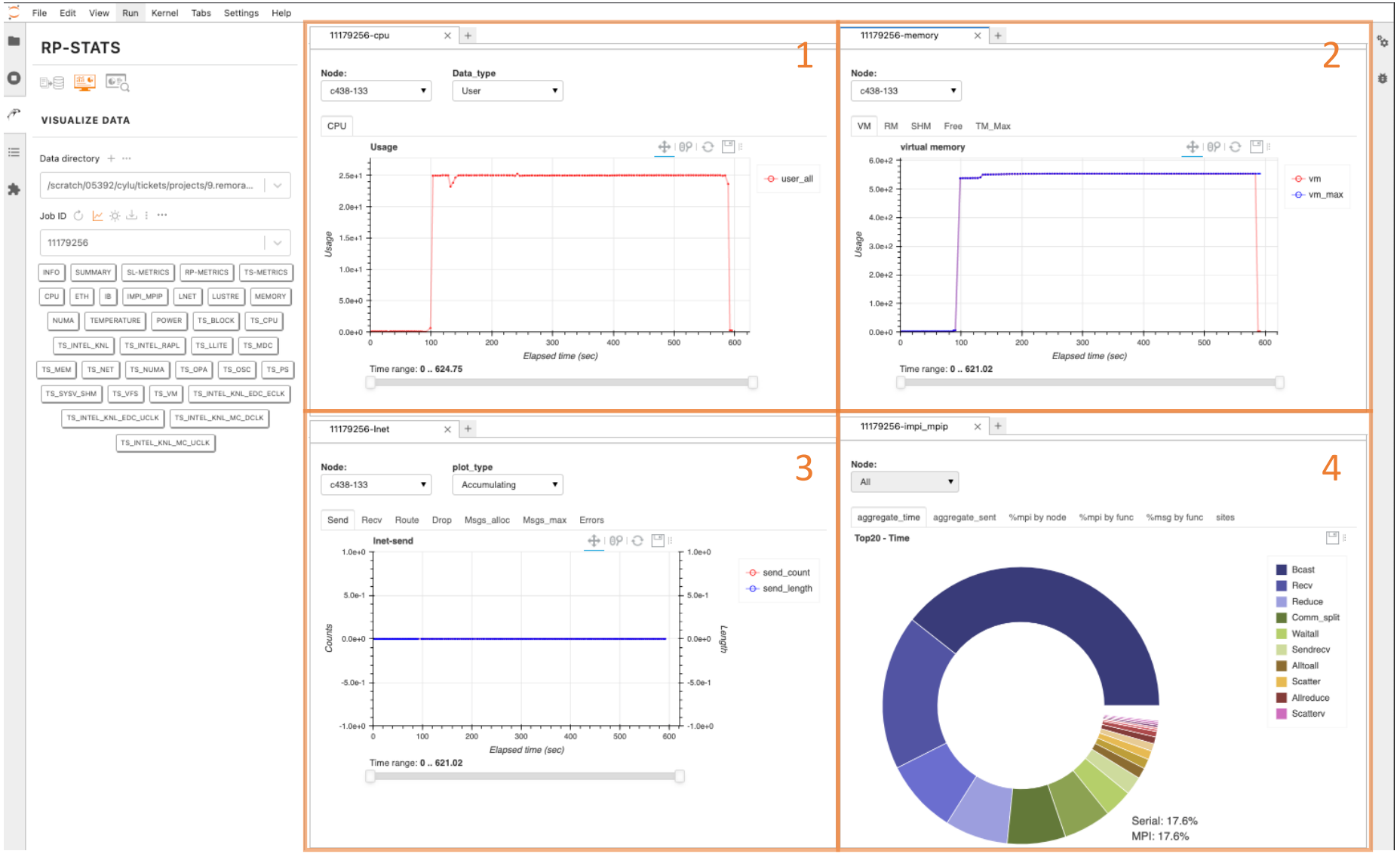


The plot window



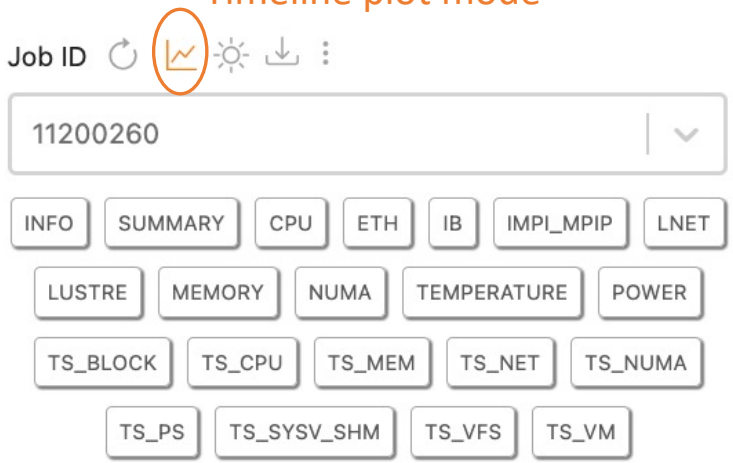
Arrange plots in the main work area



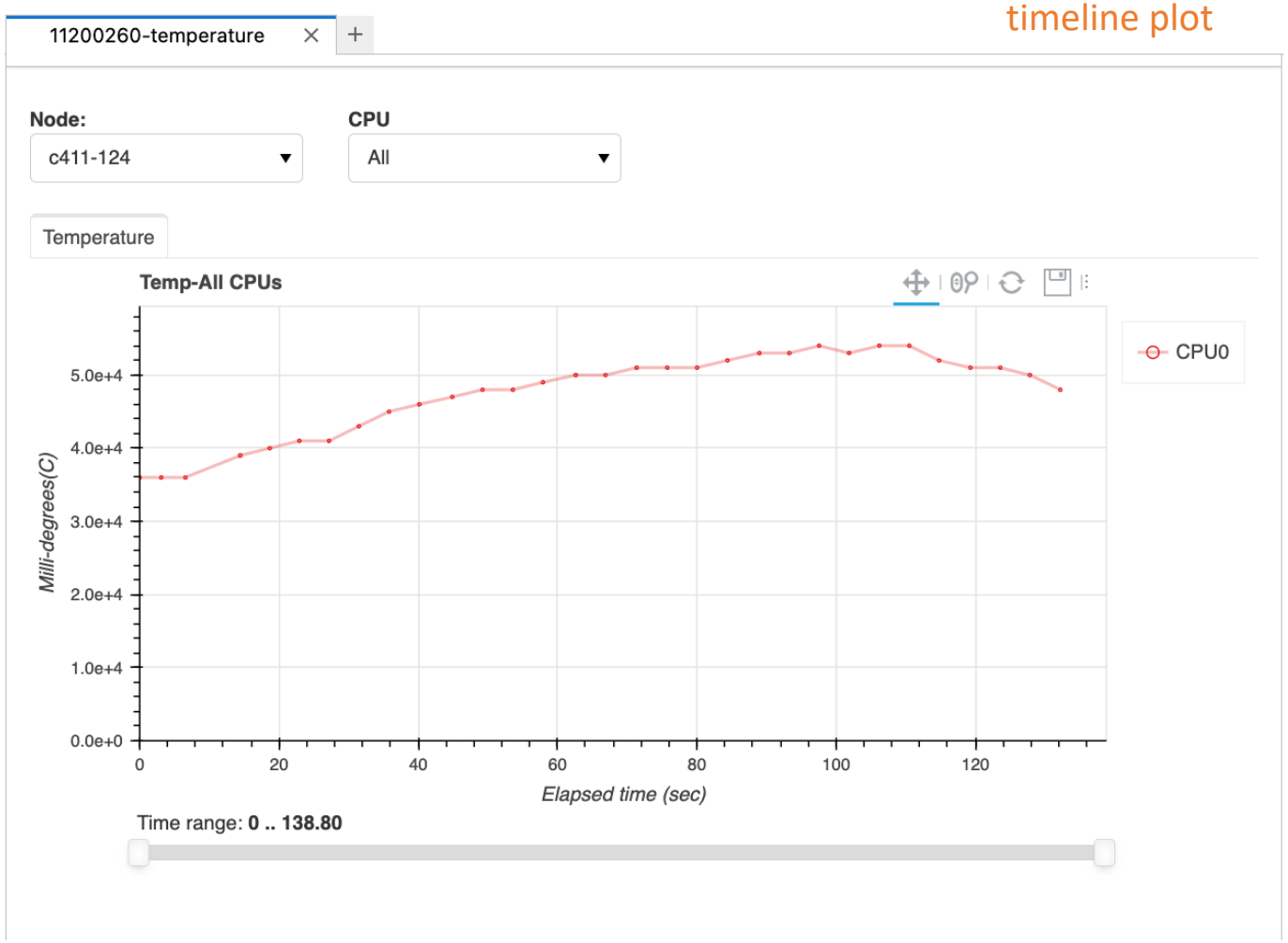

File
Edit
View
Run
Kernel
Tabs
Settings
Help
RP-STATS
VISUALIZE DATA
Data directory
/scratch/05392/cylu/tickets/projects/9.remora...
Job ID
11179256
INFO
SUMMARY
SL-METRICS
RP-METRICS
TS-METRICS
CPU
ETH
IB
IMPI_MPIP
LNET
LUSTRE
MEMORY
NUMA
TEMPERATURE
POWER
TS_BLOCK
TS_CPU
TS_INTEL_KNL
TS_INTEL_RAPL
TS_LLITE
TS_MDC
TS_MEM
TS_NET
TS_NUMA
TS_OPA
TS_OSC
TS_PS
TS_SYSV_SHM
TS_VFS
TS_VM
TS_INTEL_KNL_EDC_ECLK
TS_INTEL_KNL_EDC_UCLK
TS_INTEL_KNL_MC_DCLK
TS_INTEL_KNL_MC_UCLK
11179256-cpu
1
Node:
c438-133
Data_type
User
CPU
Usage
user_all
Usage
Elapsed time (sec)
Time range: 0 .. 624.75
11179256-memory
2
Node:
c438-133
VM
RM
SHM
Free
TM_Max
virtual memory
vm
vm_max
Usage
Elapsed time (sec)
Time range: 0 .. 621.02
11179256-lnet
3
Node:
c438-133
plot_type
Accumulating
Send
Recv
Route
Drop
Msgs_alloc
Msgs_max
Errors
lnet-send
send_count
send_length
Counts
Length
Elapsed time (sec)
Time range: 0 .. 621.02
11179256-impi_mpip
4
Node:
All
aggregate_time
aggregate_sent
%mpi by node
%mpi by func
%msg by func
sites
Top20 - Time
Bcast
Recv
Reduce
Comm_split
Waitall
Sendrecv
Alltoall
Scatter
Allreduce
Scatterv
Serial: 17.6%
MPI: 17.6%

Timeline and heatmap plot mode

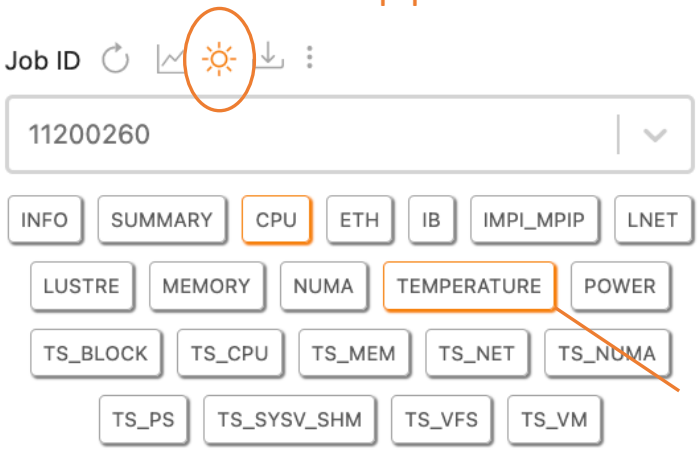
Timeline plot mode



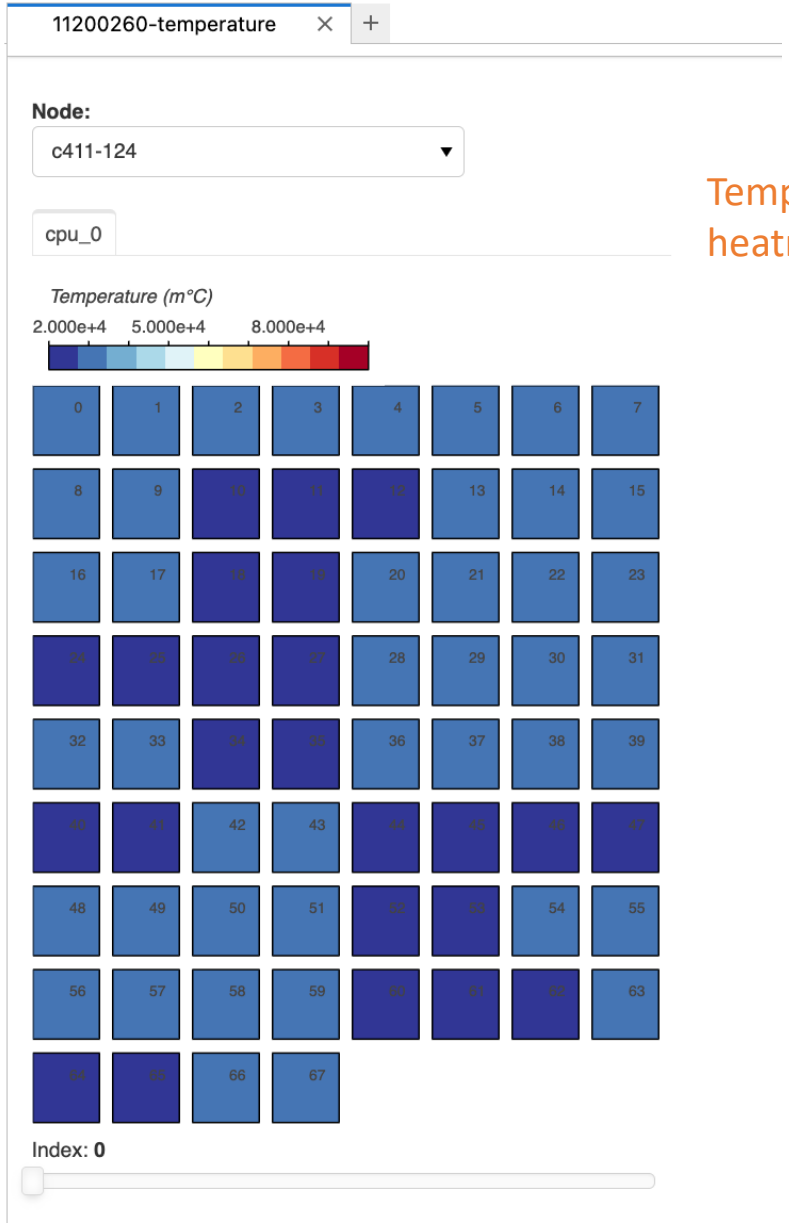
Temperature timeline plot



Heatmap plot mode



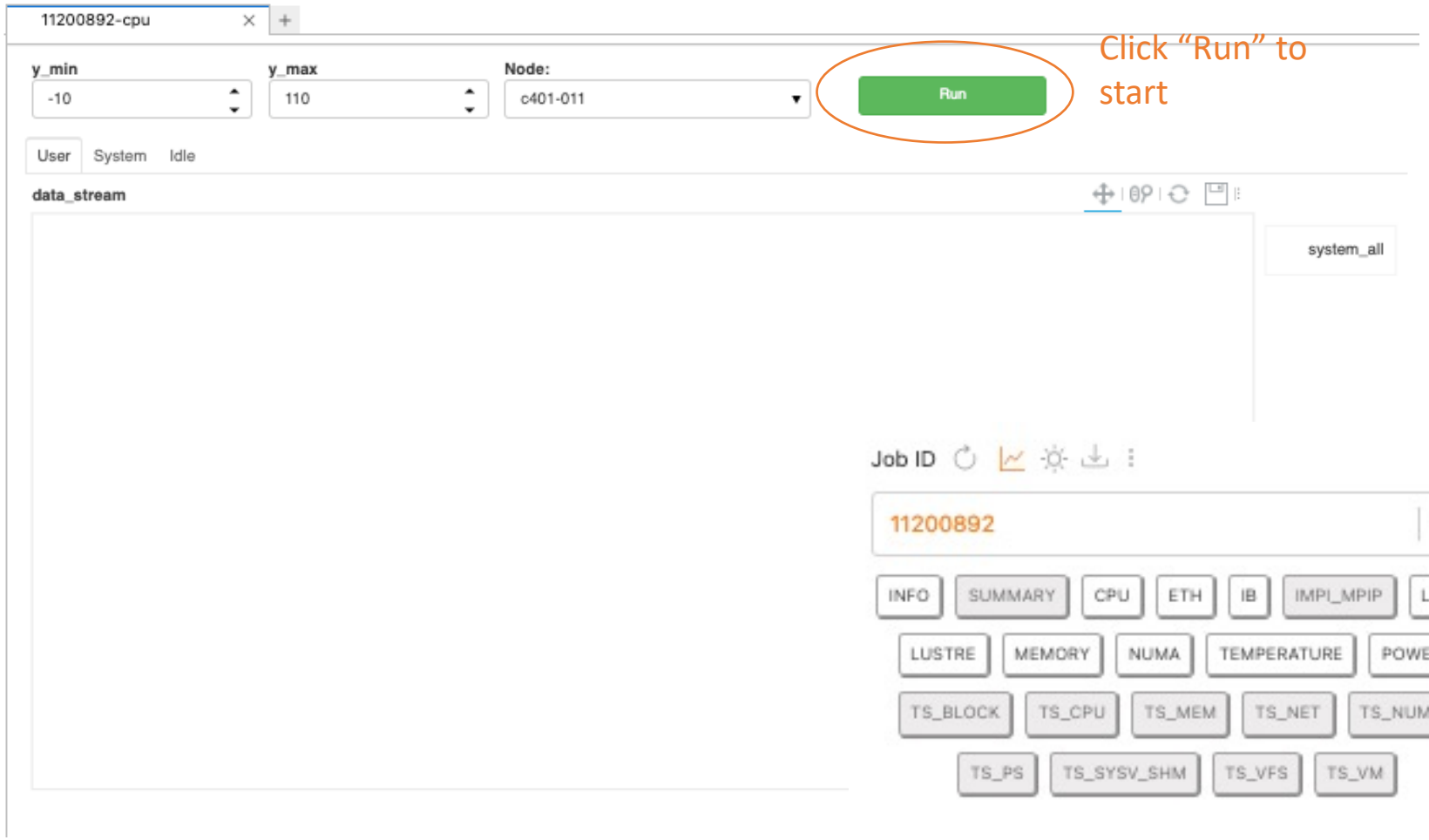
modules highlighted in orange support heatmap plot



Temperature heatmap plot

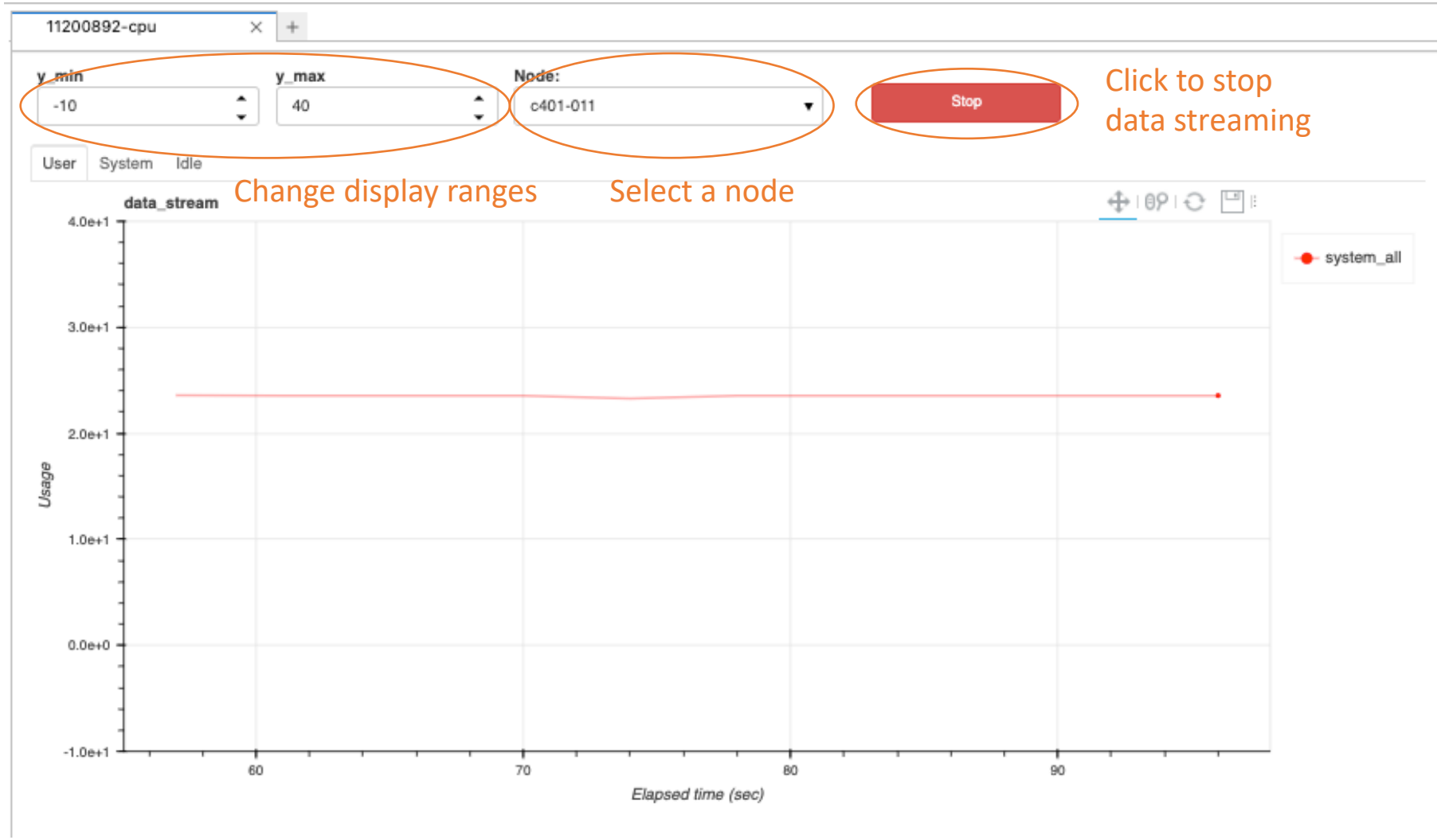
Visualize live streaming data

Interactively visualize data during collection



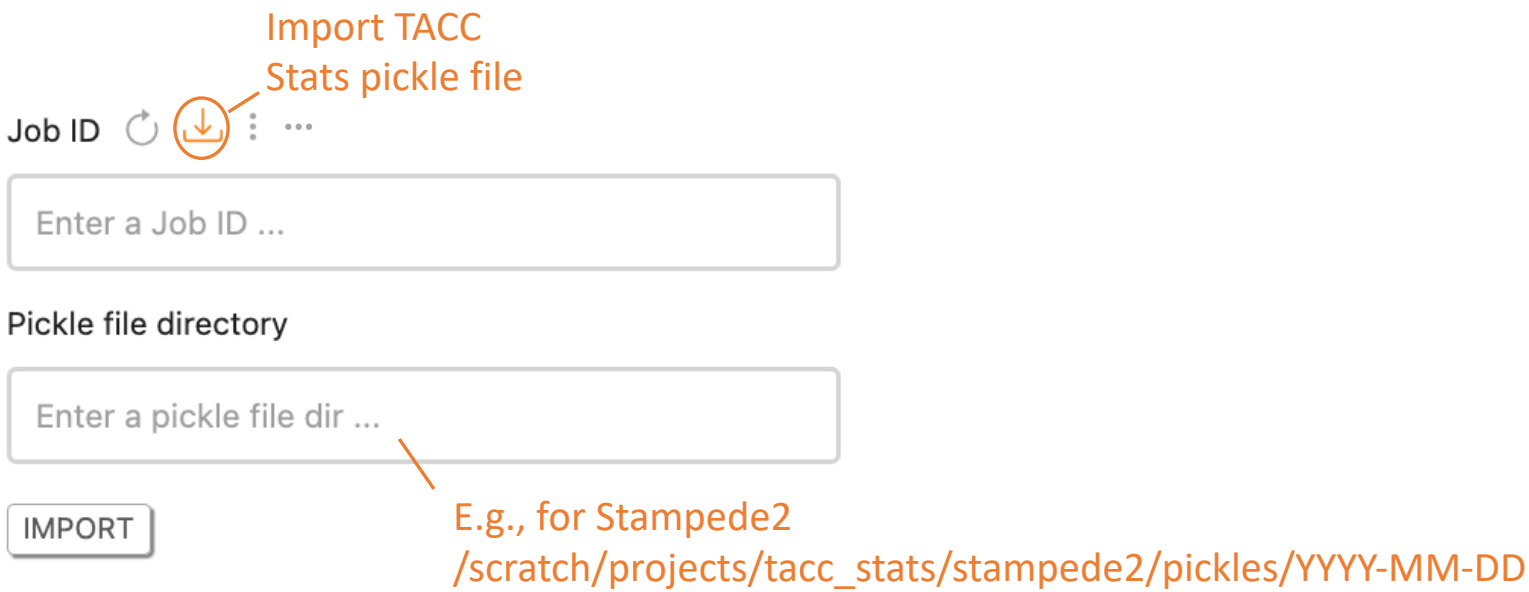
Click “Run” to start

Click on the button of an available module to open the plot. Some module data can also be visualized in a live heatmap plot

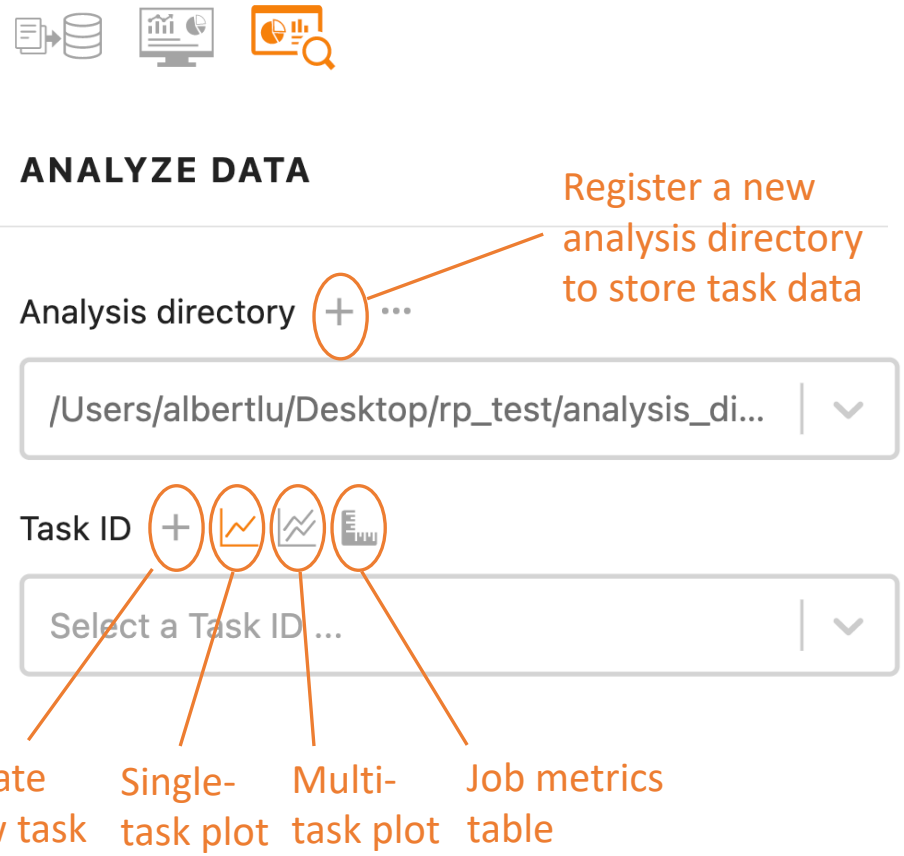

11200892-cpu
y_min
-10
y_max
40
Node:
c401-011
Stop
Click to stop
data streaming
User
System
Idle
Change display ranges
Select a node
data_stream
system_all
4.0e+1
3.0e+1
2.0e+1
1.0e+1
0.0e+0
-1.0e+1
Usage
60
70
80
90
Elapsed time (sec)

The TACC Stats data import button

- Load TACC Stats data for visualization
- Several job-level metrics are calculated from the data (same job-level metrics as found in TACC Stats)

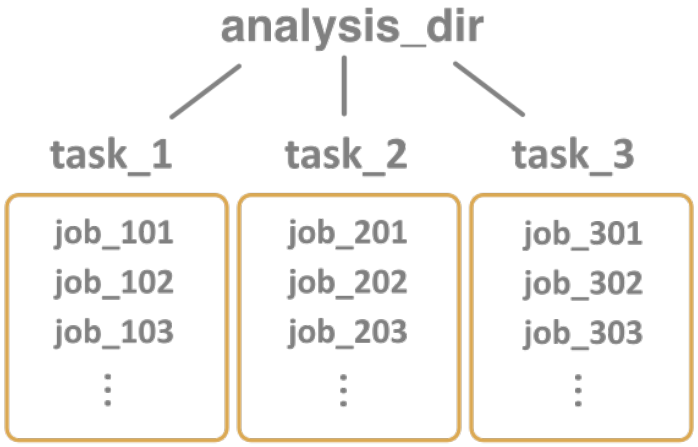


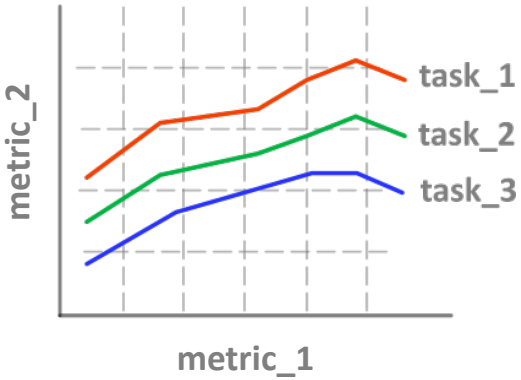
The data analyzer tab



In Analyzer, a “task” is simply defined as a group of “jobs”

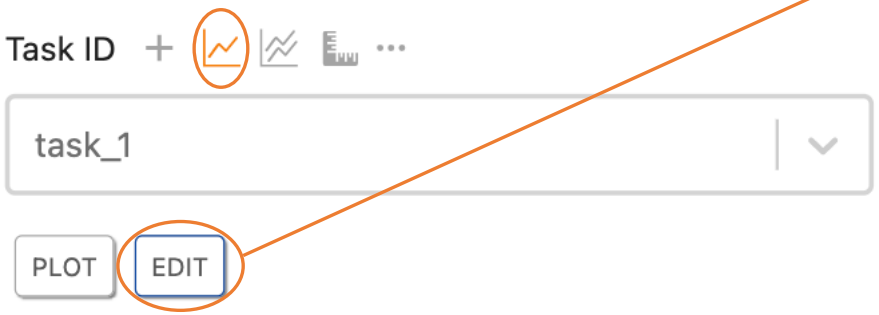
“task-plot” is a 2D line plot of any two job metrics

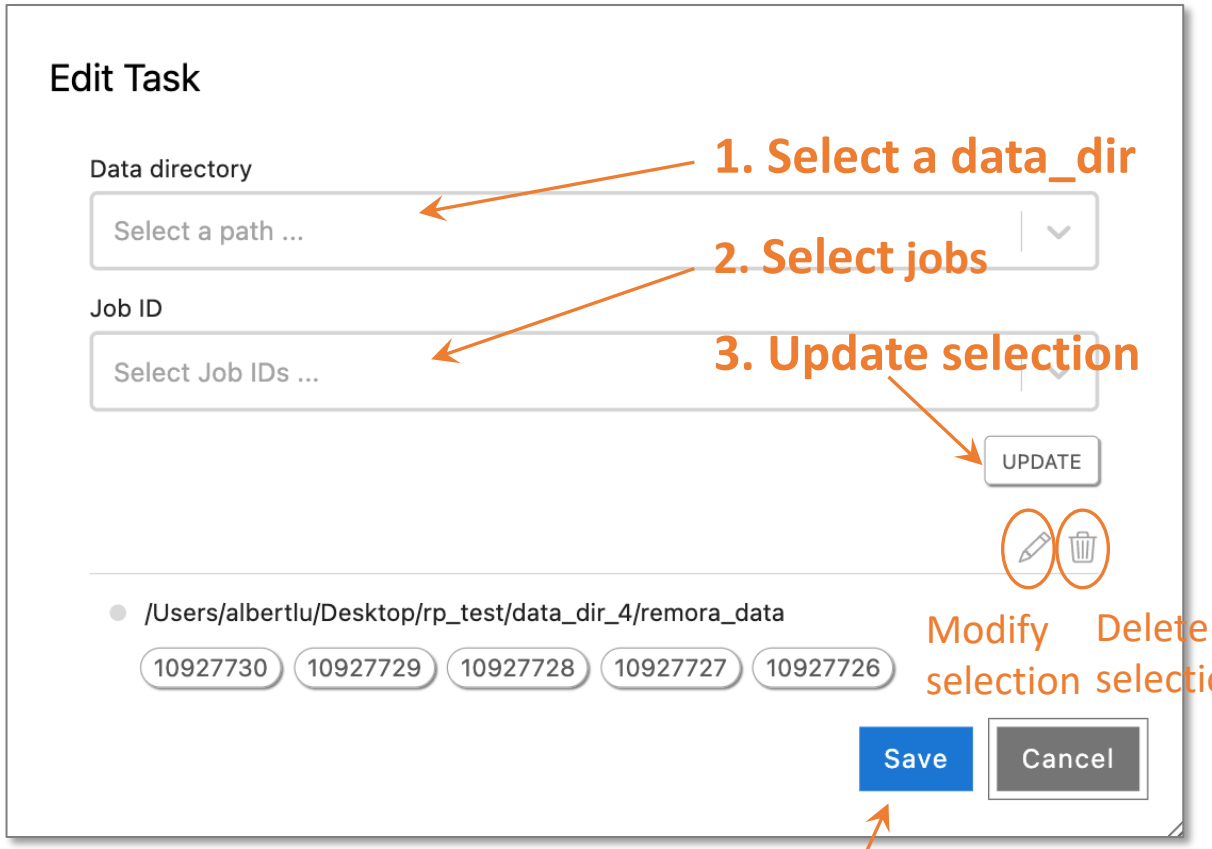




Add jobs to a Task

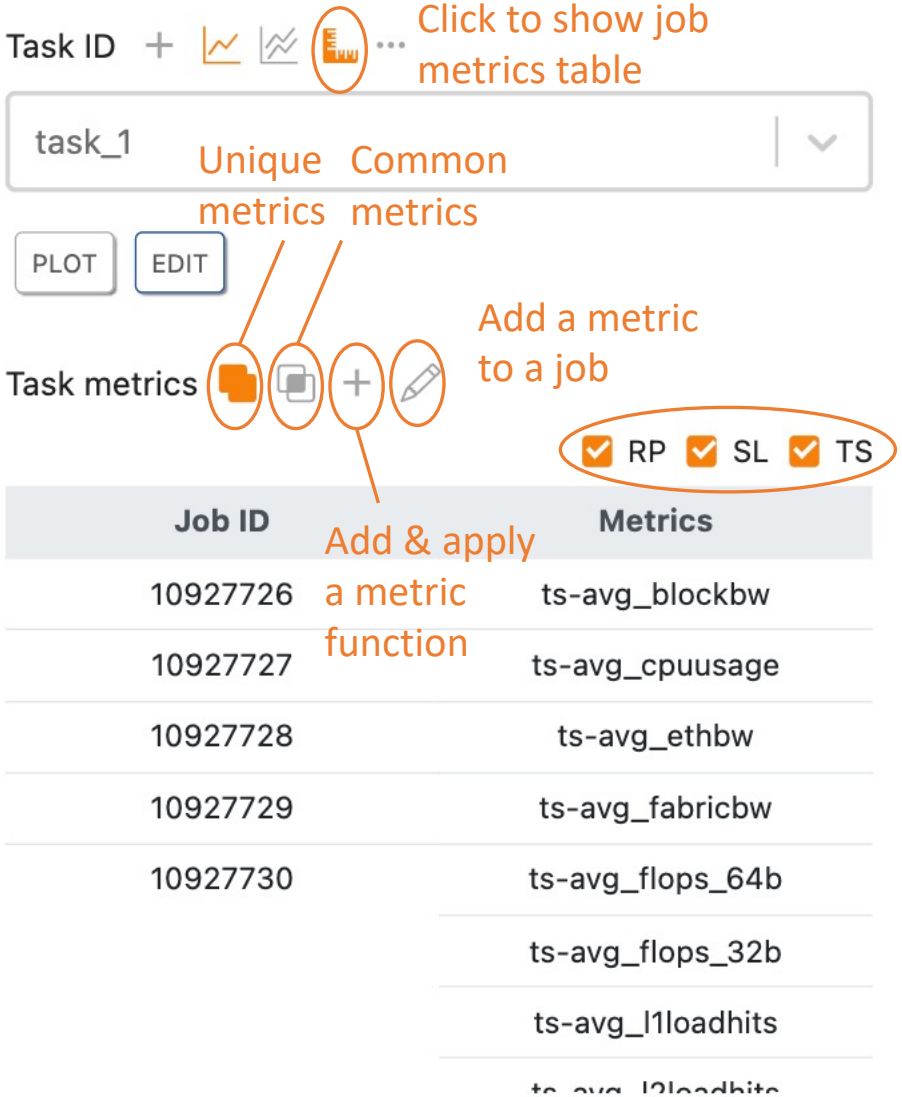
This feature is only available in the single-task plot mode





1. Select a data_dir
2. Select jobs
3. Update selection
Modify selection
Delete selection
4. Save changes

The Job metrics table button



Select/unselect to show/hide one type of metrics

RP: REMORA
SL: SLURM
TS: TACC Stats

The Job metrics table

Task metrics

RP SL TS

Job ID	Metrics
10927726	ts-avg_blockbw
10927727	ts-avg_cpuusage
10927728	ts-avg_ethbw
10927729	ts-avg_fabricbw
10927730	ts-avg_flops_64b

Double-click

Task job 10927729 info

- **ts-avg_blockbw:** 0.6822 (MB/s/node)
- **ts-avg_cpuusage:** 1.7578 (cores/node)
- **ts-avg_ethbw:** 1.1862 x 10^{-2} (MB/s/node)
- **ts-avg_fabricbw:** 0.4336 (MB/s/node)
- **ts-avg_flops_64b:** 3.6650 x 10^{1} (GFLOPs/s/node)
- **ts-avg_flops_32b:** 0 (GFLOPs/s/node)
- **ts-avg_l1loadhits:** 0 (#/s/node)
- **ts-avg_l2loadhits:** 0 (#/s/node)
- **ts-avg_llcloadhits:** 0 (#/s/node)
- **ts-avg_loads:** 0 (loads/s/node)
- **ts-avg_mbw:** 4.4598 (GB/s/node)
- **ts-avg_mdcreqs:** 0.1087 (#/s/node)
- **ts-avg_mdcwait:** 2.6415 x 10^{2} (microseconds)
- **ts-avg_openclose:** 1.0487 x 10^{3} (#/s/node)
- **ts-avg_oscreqs:** 7.3615 (#/s/node)

OK

Task metrics

RP SL TS

Job ID	Metrics
10927726	ts-avg_blockbw
10927727	ts-avg_cpuusage
10927728	ts-avg_ethbw
10927729	ts-avg_fabricbw
10927730	ts-avg_flops_64b
	ts-avg_flops_32b
	ts-avg_l1loadhits

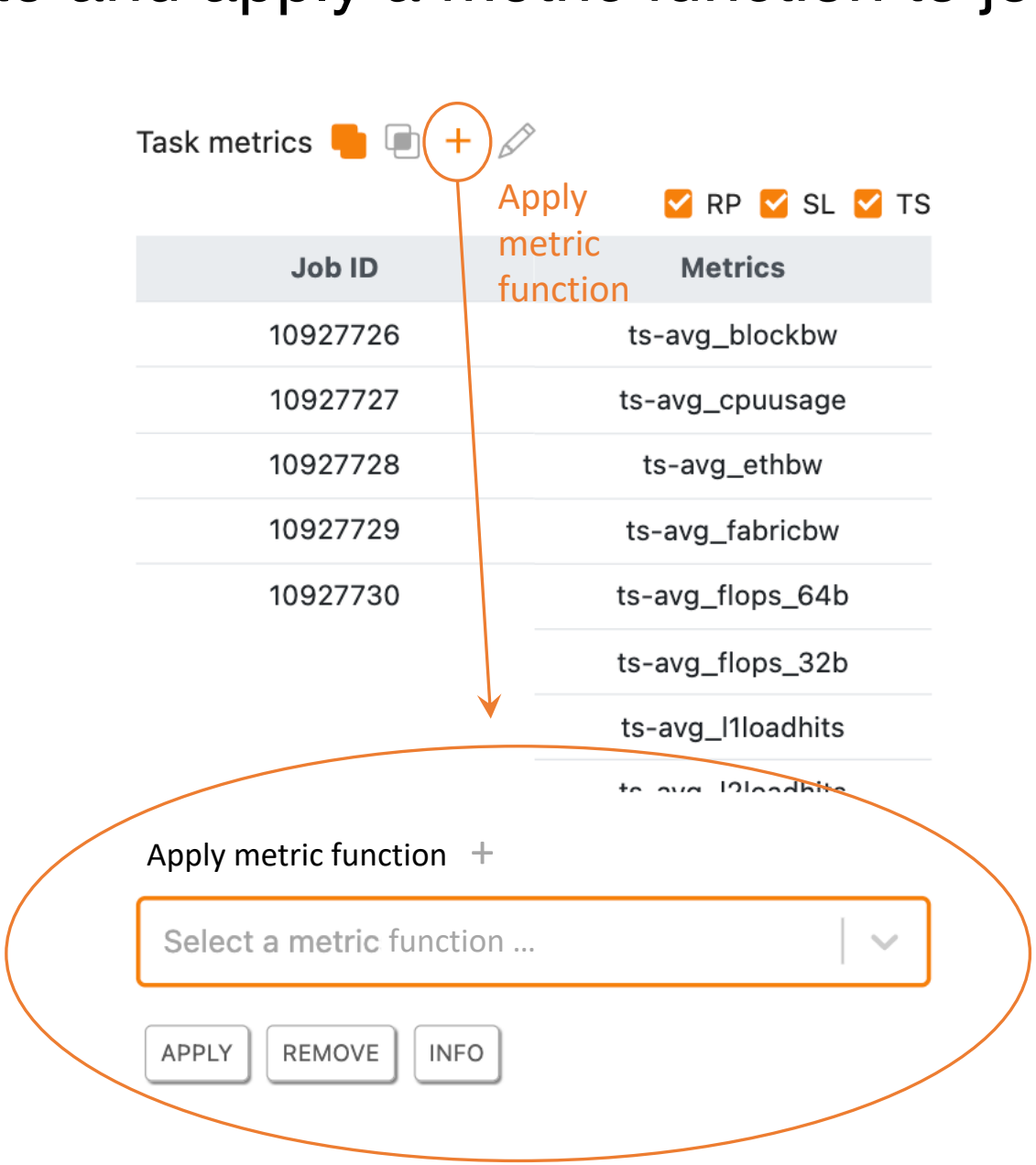
Double-click

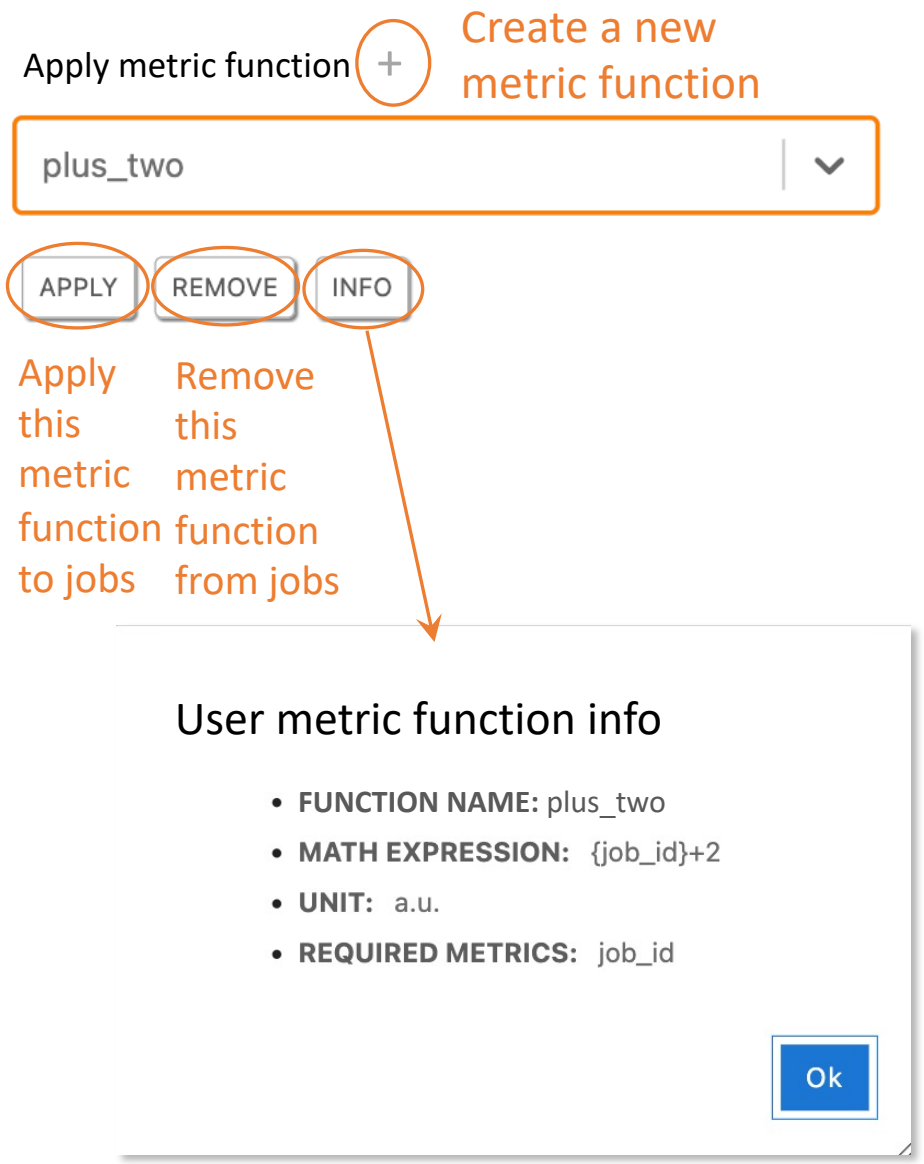
Task metrics ts-avg_flops_64b info

- **10927726:** 0.3598 (GFLOPs/s/node)
- **10927727:** 1.1247 (GFLOPs/s/node)
- **10927728:** 1.1174 (GFLOPs/s/node)
- **10927729:** 3.6650 x 10^{1} (GFLOPs/s/node)
- **10927730:** 1.1221 (GFLOPs/s/node)

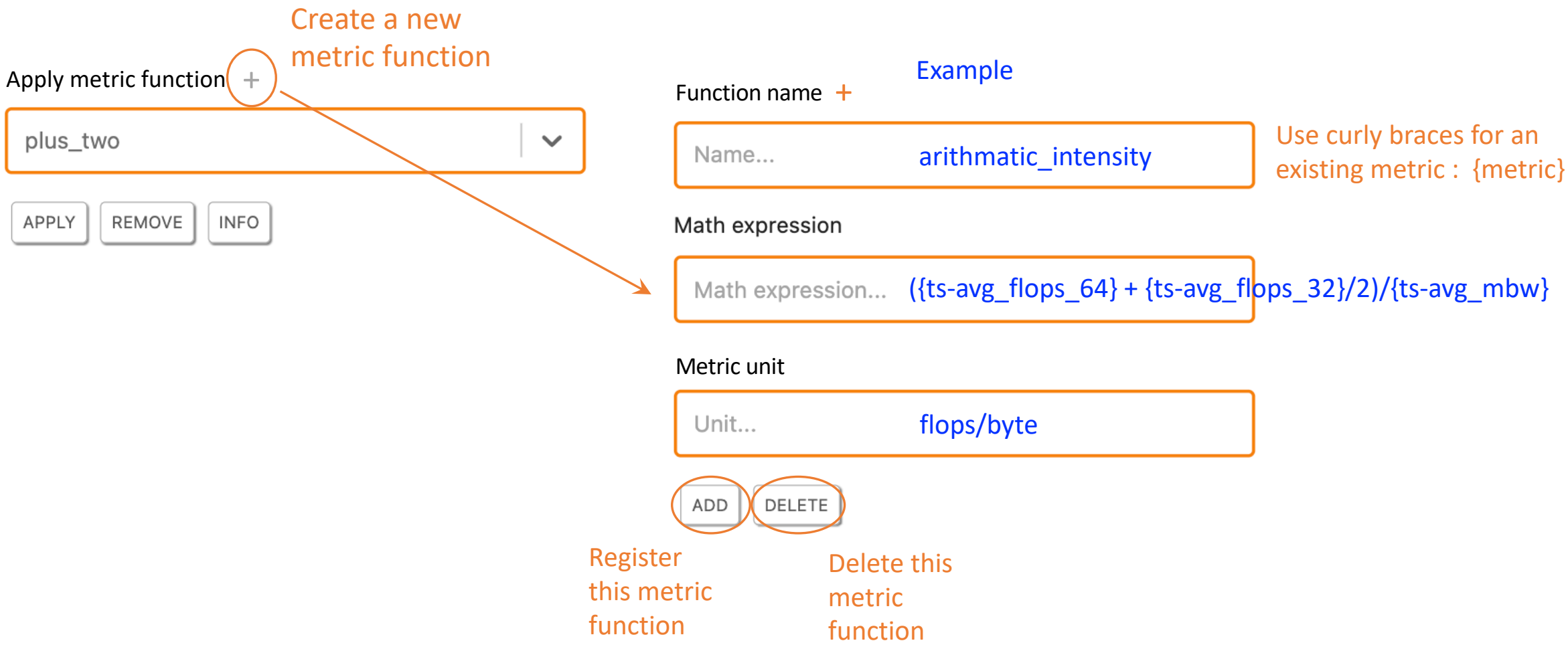
OK

Create and apply a metric function to jobs

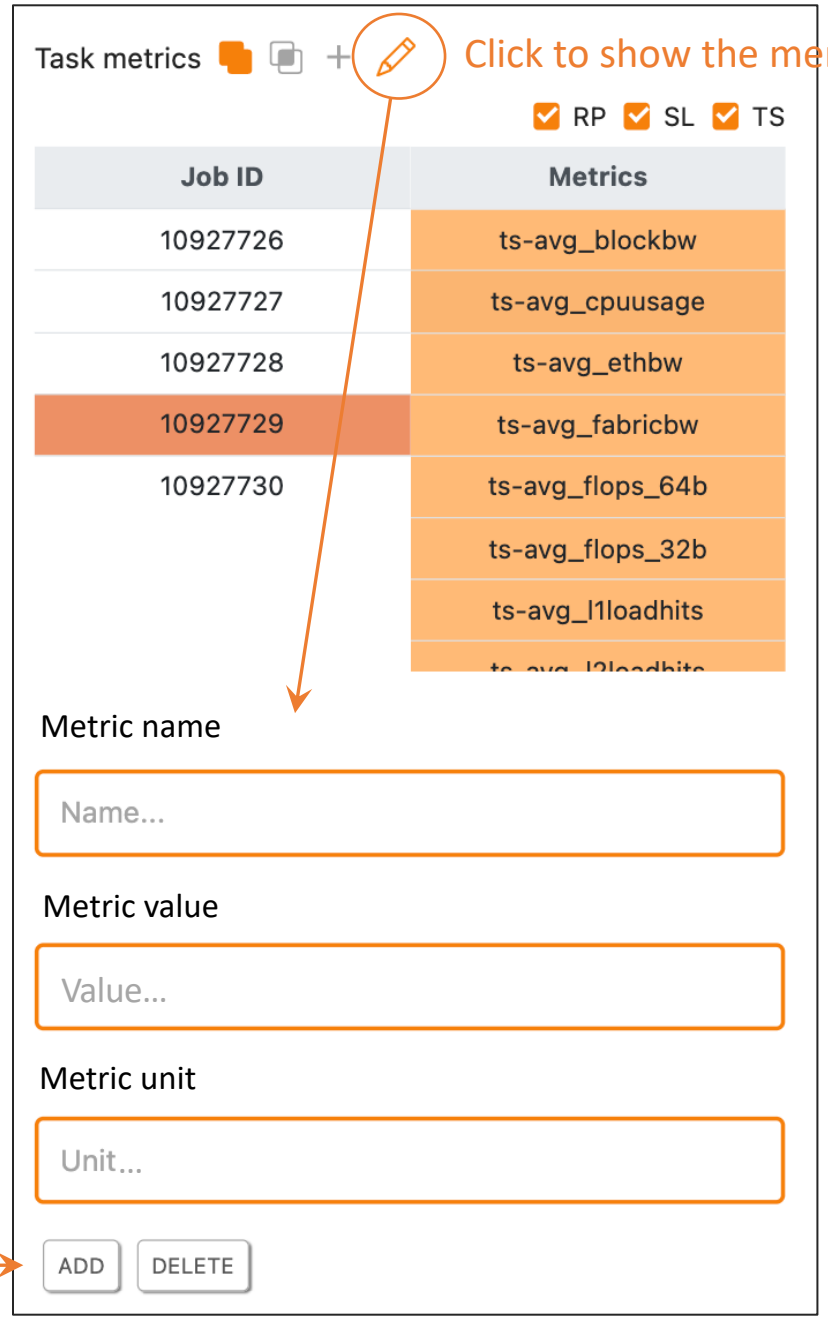




Create a new metric from existing metrics



Add/delete a single job metric

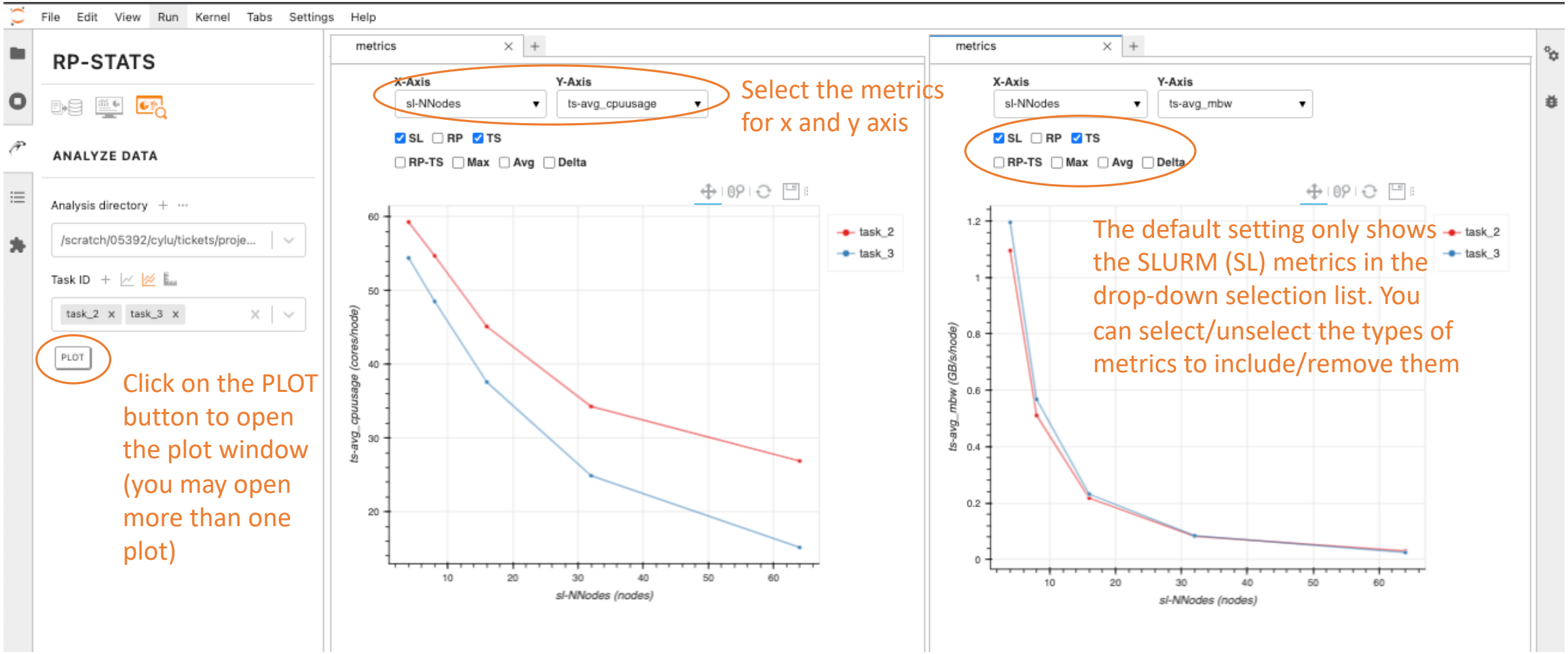


1. Select a job

2. Enter metric info

3. Click to add/delete a metric (only metric name is required to delete a metric)

Task-metric plot



License