



WWW.TACC.UTEXAS.EDU



TEXAS
The University of Texas at Austin

RP-STATS QuickStart Guide

Albert Lu (alu@tacc.utexas.edu)

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

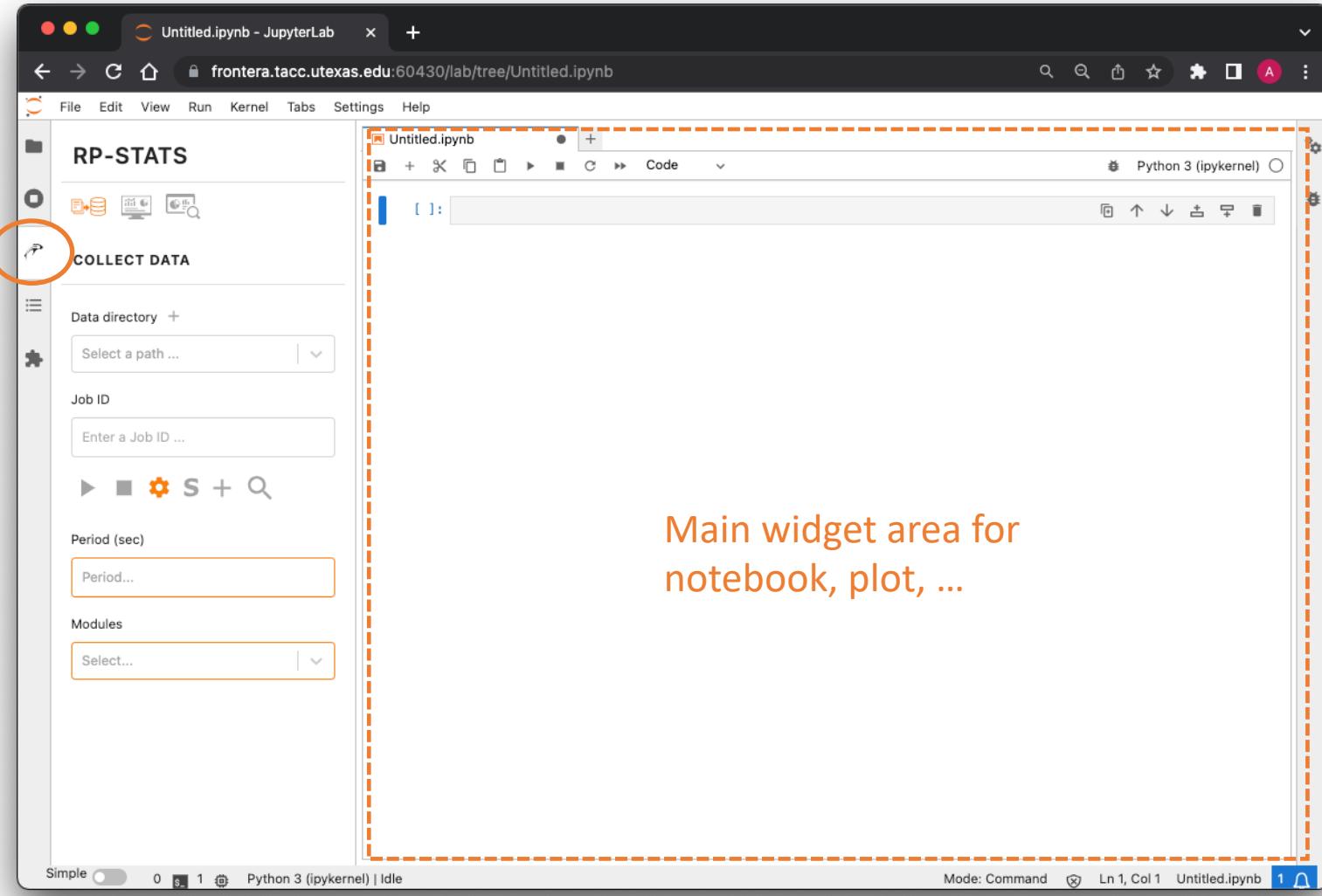
File: jupyter.out

```
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

RP-STATS

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

COLLECT DATA

Data directory +
Select a path ...

Job ID ...
Enter a Job ID ...

▶ ■ ⚙ S + 🔎

COLLECT DATA



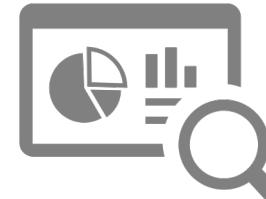
Control data collections and view information

VISUALIZE DATA



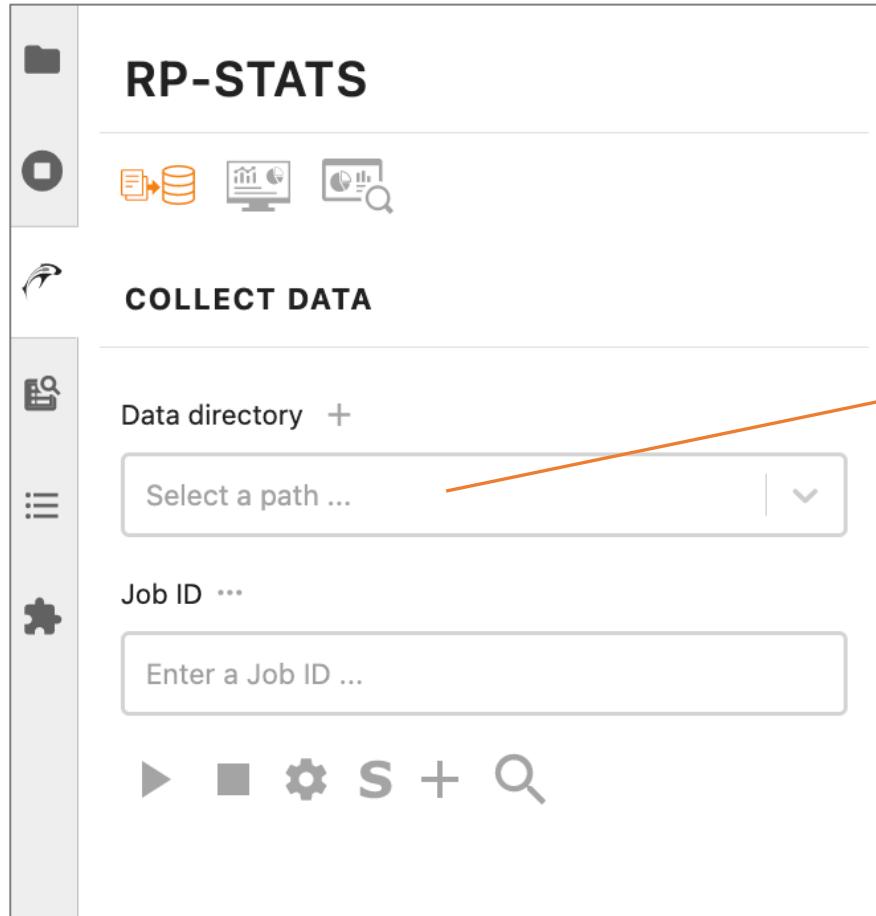
Visualize collected data

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.

cpu, eth, ib, impi_mpip,
Inet, lustre, memory, numa,
power, temperature
nvml (gpu), dcgm (gpu)

ts_block, ts_cpu, ts_mem,
ts_net, ts_numa, ts_proc,
ts_ps, ts_sysv_shm,
ts_tmpfs, ts_vfs, ts_vm

User-defined modules



REMORA



TACC Stats
root permission not
required

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

sl-ElapsedRaw
sl-NCPUS
sl-Nnodes
sl-Ntasks
sl-AveCPU
sl-MinCPU
sl-SystemCPU
sl-UserCPU

:

ts-metrics

ts-avg_blockbw
ts-avg_cpi
ts-avg_freq
ts-avg_cpuusage
ts-avg_ethbw
ts-avg_fabricbw
ts-avg_flops_64b
ts-avg_flops_32b

:

rp-metrics

rp-cpu_usage_use
rp-cpu_usage_sys
rp-cpu_usage_idle
rp-eth_rx_bytes
rp-eth_rx_packets
rp-eth_tx_bytes
rp-eth_tx_dropped
rp-eth_tx_errors

:

Ways to start RemoraPy

Use RP-Stats dashboard

Job ID
xxxxxx



Use *job_id*

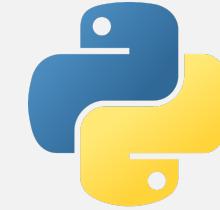
Use *path* to a job script

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 ————— Load remorapy
                         module

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
```

Use the script “remorapy” to launch
your application

} Set REMORAPY
environment
variables

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)

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)
```

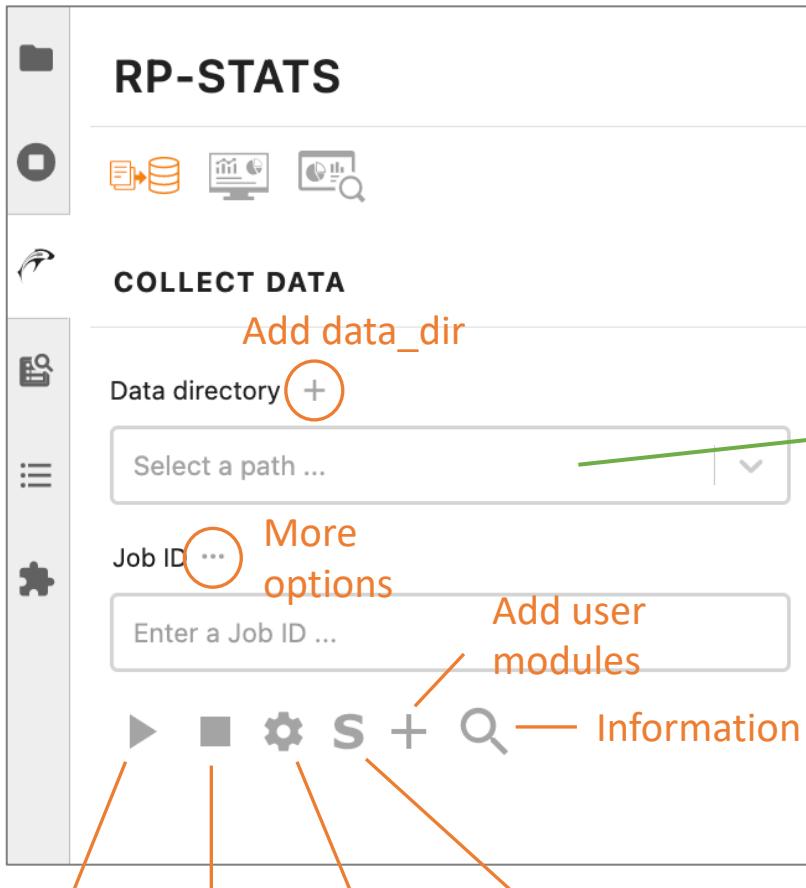
```
rp.PostCrash(job_id, data_dir=None, debug=False)
```

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

```
rp.AddRPMModule(module_name, commands, n_data)
```

```
rp.DeleteRPMModule(module_name)
```

The data collector tab

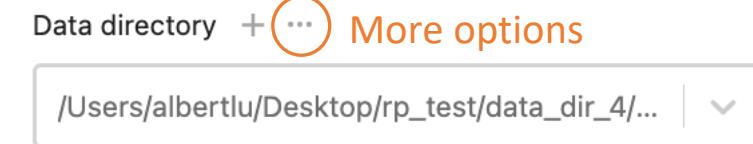


Start collecting data Stop collecting data Edit RP settings SLURM mode

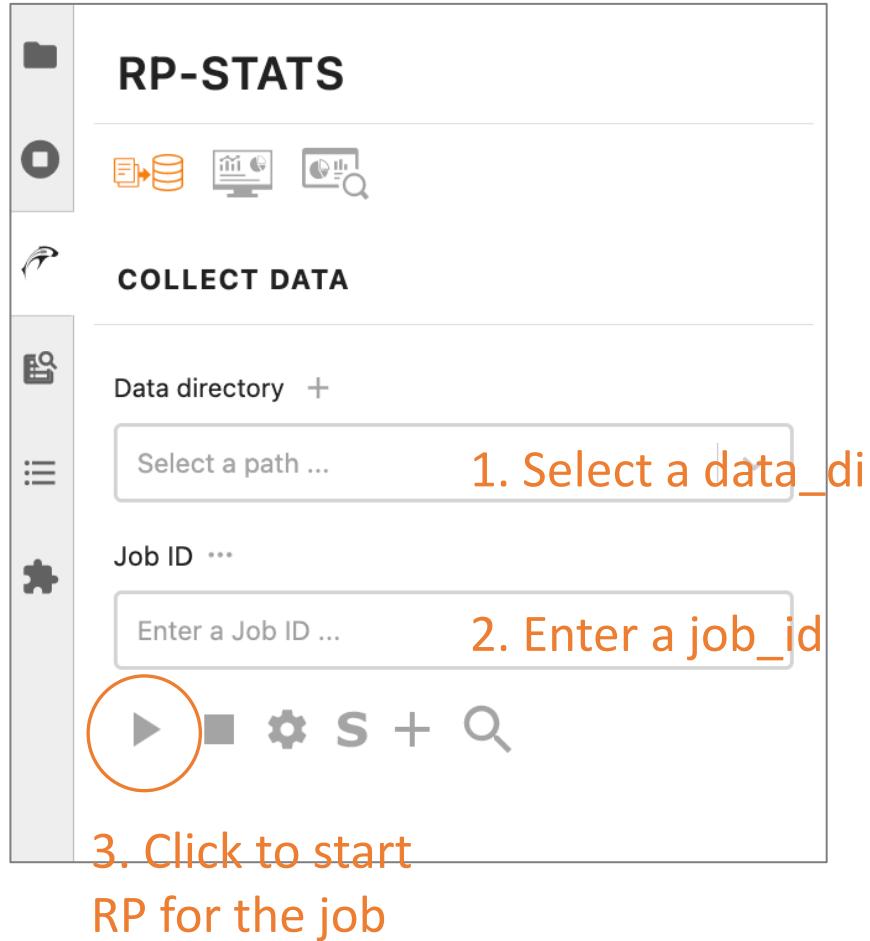
More options button



When a data_dir is selected:

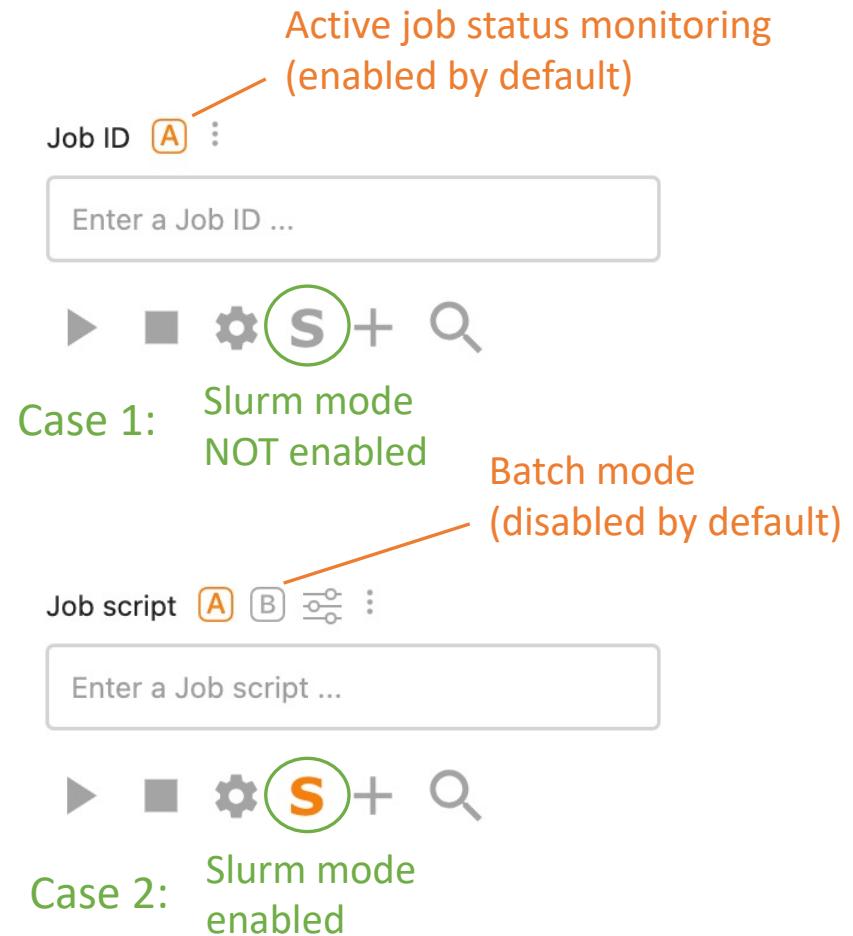
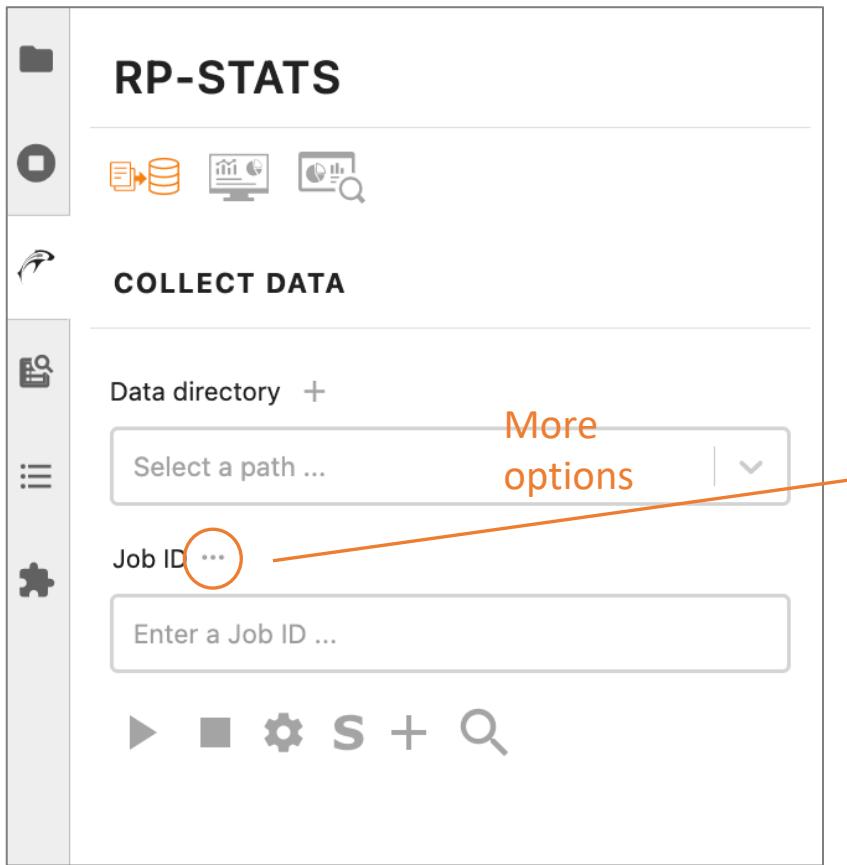


Start data collection on a running job

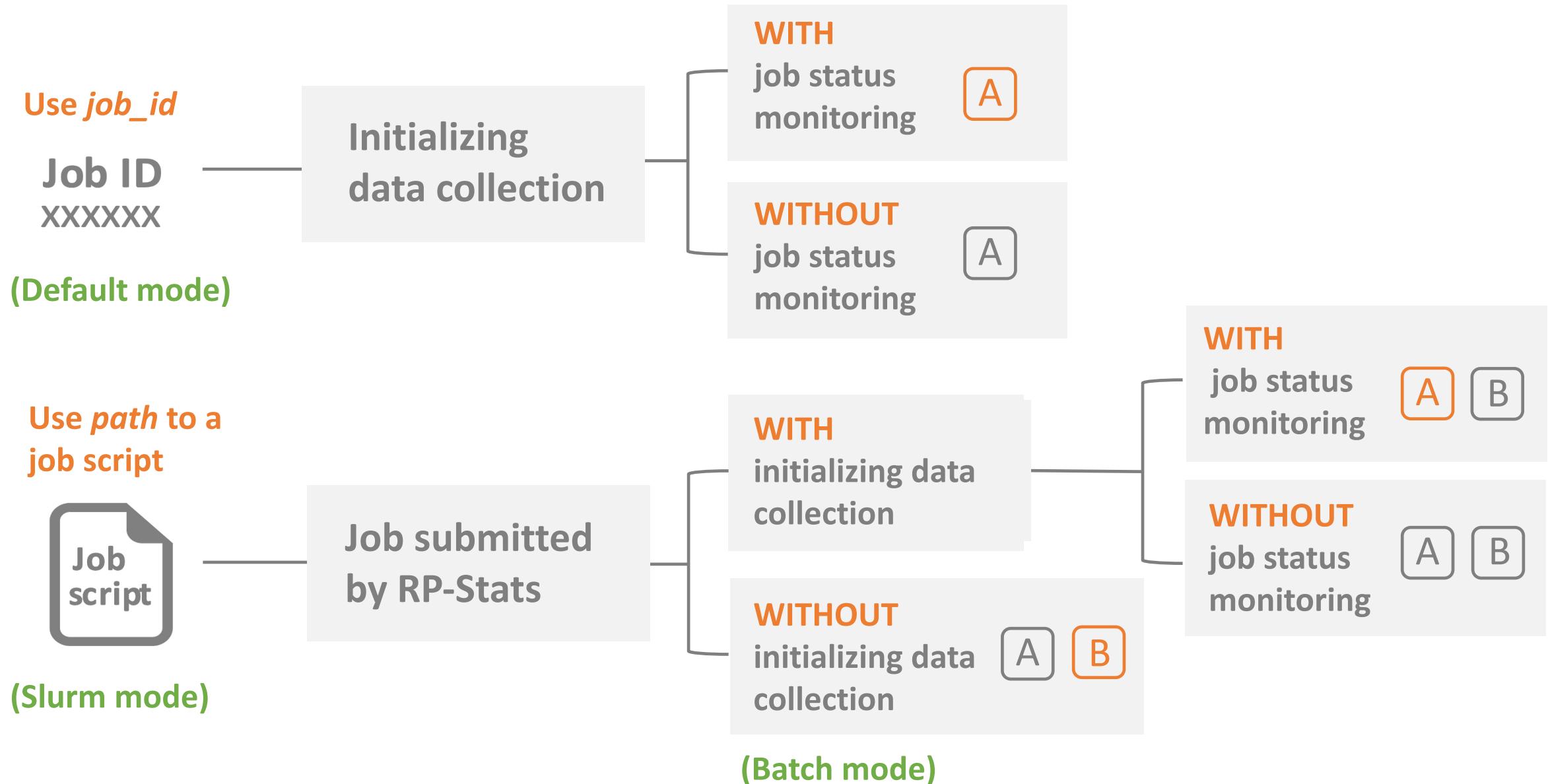


Three steps to start (default) data collection using Job ID

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)

Modules

Clear all **Select all** **Select all TS**

Select... **modules** **RP modules** **modules** | ▾

SAVE **LOAD** **CLEAR** **ALL-RP** **ALL-TS**

Save setting
(period & modules) Load a
saved setting

Period (sec)

Modules

 | ▾

- CPU
- ETH
- IB
- LNET
- LUSTRE

Period (sec)

Modules

TACCSTATS ✕ CPU ✕ ETH ✕ ✕ | ▾

TaccStats Modules

 | ▾

- 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

Three steps to launch a job and collect performance data

Data directory + ...

/Users/albertlu/Desktop/rp_test/data_dir_4/... | ▾

Job script    :

Enter a Job script ...     

2. Enter a full path to a job script

3. Click to submit the job 

Job script   :

Enter a Job script ... 

▶ ■ ⚙  + 🔎

SLURM OPTIONS 

Account (-A) 

Partition (-p) 

Number of nodes (-N) 

Number of tasks (-n) 

Runtime (-t) 

Job name (-J) 

SLURM OPTIONS 

Job_ID 

STOP 

Values set here will override the corresponding settings in the job script

The add user-defined module button



Click to show
the edit menu

Module name

my_module

Number of data

2

Data names and units

name_1, unit_1; name_2, unit_2

Commands

echo 1 2

Click to test the
command



Add this
module

Delete this
module

Load from a
script file

Test user module

- MODULE: my_module
- STDOUT: 1 2

Output: two
numbers in
a row



Load user-defined module file

Load user module from a script file

TEST ADD DELETE LOAD

Click the load button to open the window

Load user module

my_1

NAME: my_1

N_DATA: 3

DATA PROPS: {m1, kg}, {m2, kg}, {m3, m}

COMMAND:

echo 1 2 3

Load Delete Cancel

load script file Delete 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



COLLECT DATA

Data directory +

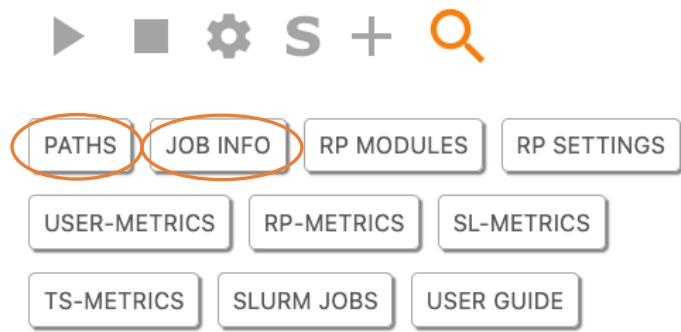
A text input field with a dropdown arrow on the right.

Job ID ...

A text input field.

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

JOB INFO

Data directory

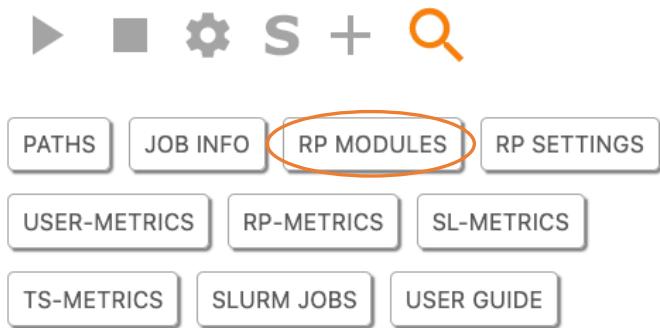
/Users/albertlu/Desktop/rp_test/data_dir_4/remora_data

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	
Inet	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

The screenshot shows the 'RP SETTINGS' page of a monitoring application. At the top, there are navigation icons: back, forward, settings, search, and a plus sign. Below them are tabs: PATHS, JOB INFO, RP MODULES, and RP SETTINGS, with RP SETTINGS highlighted and circled in orange. Further down are USER-METRICS, RP-METRICS, SL-METRICS, TS-METRICS, SLURM JOBS, and USER GUIDE tabs.

The main area is titled 'RP SETTINGS' and contains a table of monitoring configurations. The table has columns: Name, Period(sec), RP modules, and TS modules. The data is as follows:

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, Inet, taccstats	ts_numa, ts_proc
test-c	2	eth, impi_mpi, taccstats	ts_block, ts_mem
my_setting	1	cpu, eth, ib, impi_mpi, taccstats	ts_block, ts_sysv_shm, ts_numa

An orange bracket on the left side of the table groups the first five rows under the heading: "All user RP settings (period + modules) saved under the selected *data_dir*".

The user-metrics info button

The screenshot shows the 'USER-METRICS' tab selected in a navigation bar. The navigation bar also includes PATHS, JOB INFO, RP MODULES, RP SETTINGS, TS-METRICS, SLURM JOBS, and USER GUIDE. A search icon is located at the top right. The main area is titled 'USER-METRICS' and displays the following information:

- RP data directory: /Users/albertlu/Desktop/rp_test/data_dir_4/remora_data
- A table listing four user-defined metrics:

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

An orange bracket on the left side of the table groups the four rows and points to the text: "All user-defined metrics saved under the selected *data_dir*".

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

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	%	(i)
rp-cpu_usage_sys	%	(i)
rp-cpu_usage_idle	%	(i)
rp-eth_rx_bytes	bytes	(i)
rp-eth_rx_crc_errors	#	(i)
rp-eth_rx_frame_errors	#	(i)
rp-eth_rx_packets	#	(i)
rp-eth_tx_bytes	bytes	(i)
rp-eth_tx_dropped	#	(i)
rp-eth_tx_errors	#	(i)
rp-eth_tx_packets	#	(i)
rp-ib_rx_bytes	bytes	(i)
rp-ib_rx_packets	#	(i)
rp-ib_tx_bytes	bytes	(i)
rp-ib_tx_packets	#	(i)
rp-Inet_drop_count	#	(i)
rp-Inet_drop_length	bytes	(i)
rp-Inet_errors	#	(i)
rp-Inet_msgs_alloc	#	(i)



RP metrics (rp-eth_tx_packets)

- name: rp-eth_tx_packets
- unit: #
- info: Indicates the number of packets transmitted by a network device

OK

Select a button to show details

The SLURM metrics info button



SL-METRICS

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



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	(i)
ts-avg_cpi	cycles/instr	(i)
ts-avg_freq	MHz	(i)
ts-avg_cpuusage	cores/node	(i)
ts-avg_ethbw	MB/s/node	(i)
ts-avg_fabricbw	MB/s/node	(i)
ts-avg_flops_64b	GFLOPs/s/node	(i)
ts-avg_flops_32b	GFLOPs/s/node	(i)
ts-avg_llloadhits	#/s/node	(i)
ts-avg_l2loadhits	#/s/node	(i)
ts-avg_llcloadhits	#/s/node	(i)
ts-avg_inetbw	MB/s/node	(i)
ts-avg_gputil	%	(i)
ts-avg_inetmsgs	#/s/node	(i)
ts-avg_loads	loads/s/node	(i)
ts-avg_mbw	GB/s/node	(i)
ts-avg_mcdrambw	GB/s/node	(i)
ts-avg_mdcreqs	#/s/node	(i)
ts-avg_mdcwait	microseconds	(i)
ts-avg_openclose	#/s/node	(i)
ts-avg_oscreqs	#/s/node	(i)

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

Double-click to show job details

A blinking orange circle indicates RP is running with the job

All your jobs are listed in the status table

The User guide button



rp-stats-quickstart.html X +
Trust HTML

RP-Stats QuickStart Guide

Table of Contents

- Basics**
 - Three basic modes
 - RP data directory
 - RP modules
 - Job metrics
 - Ways to start data collection for a job
 - batch mode
 - python mode
- Data collector**
 - The data collector tab
 - Start data collection on a running job (attach mode)
 - The setting button
 - The SLURM mode button
 - The add user-defined module button
 - The information button
 - The paths and jobs info button
 - The RP modules info button
 - The RP settings info button

Scroll up and down to browse the content.

Click on the link to just to a specific section

The RP data directory

In remorapy, "data_dir" is a special directory for storing RP-STATS settings and the databases

To register a new data_dir, click on the "+" button. Enter the full path to the directory and press the return key. Note that the name of the data directory has to be "remora_data"

`data_dir = _parent_dir/remora_data`

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

RP-STATS will create a directory named 'remora_data' inside `_parent_dir` and copy configuration files into it.

You may have multiple data_dir, but RP-STATS does not sync the contents inside data directories

Back

Click "Back" to jump back to the table of contents

PATHS JOB INFO RP MODULES RP SETTINGS
USER-METRICS RP-METRICS SL-METRICS
TS-METRICS SLURM JOBS **USER GUIDE**

The data visualizer tab



VISUALIZE DATA

Data directory + ...

/Users/albertlu/Desktop/rp_test/data_dir_4/... | ▾

Click to open job
status filter

Job ID

Select a Job ID ...
More options

11179319

10927730

10927729

10927728

10927727

10927726

Select a *data_dir* to visualize saved
jobs' performance data

R P F X :

Job status filter

Default: showing all
Unselect buttons to hide jobs
of certain types of status

XXXXXX (Blue: Pending)
XXXXXX (Grey: Finished)
XXXXXX (Orange: Running)
XXXXXX (Dark red: Cancelled)

Click to update job status (job status
may change after the last time
information was loaded)

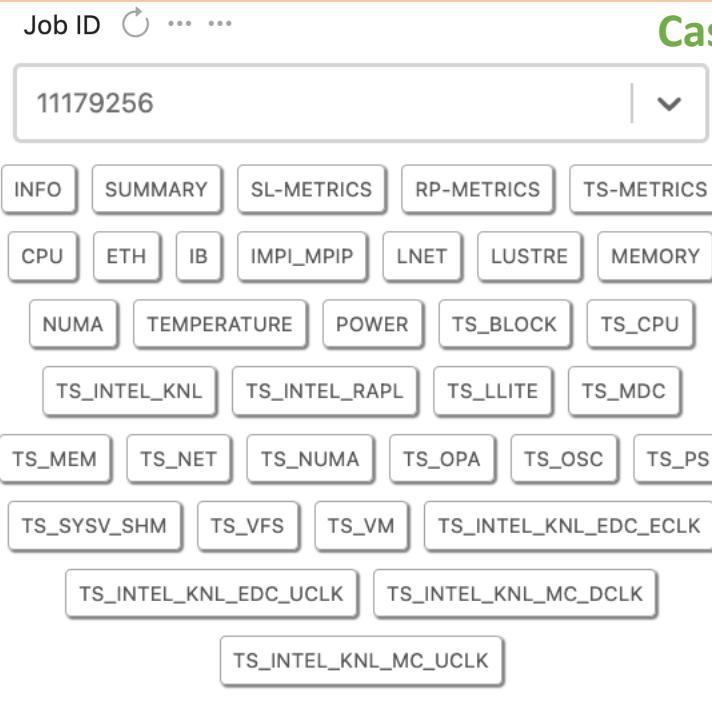
Job ID list

Job info and module buttons

VISUALIZE DATA

Data directory + ...

/scratch/05392/cylu/tickets/projects/9.remo...

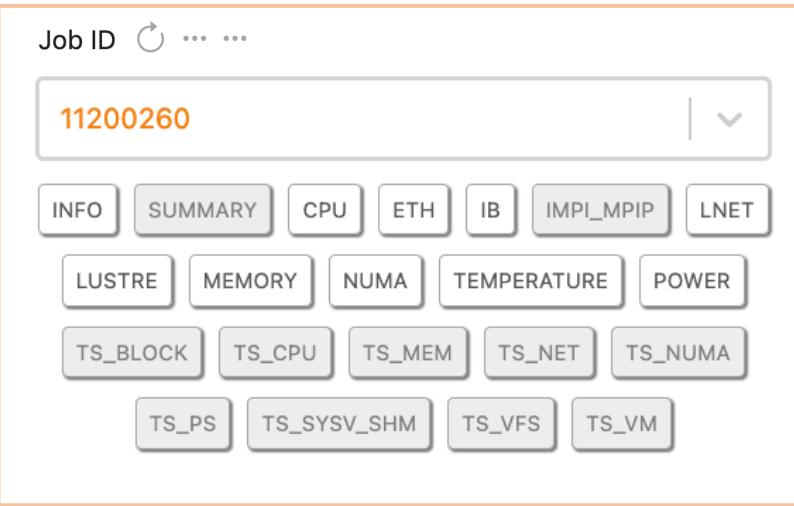


Case 1 : finished job

Information
Plots of
REMORA data

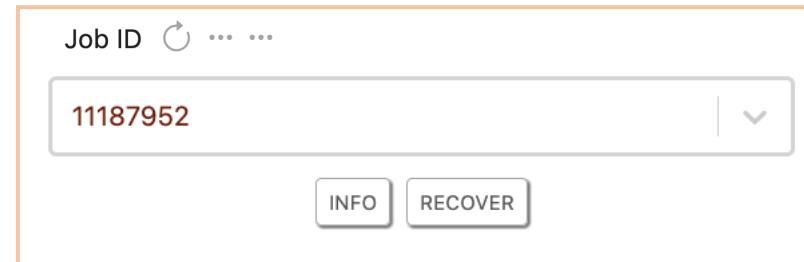
Plots of TACC
Stats data
(TS_xxx)

Case 2 : running job



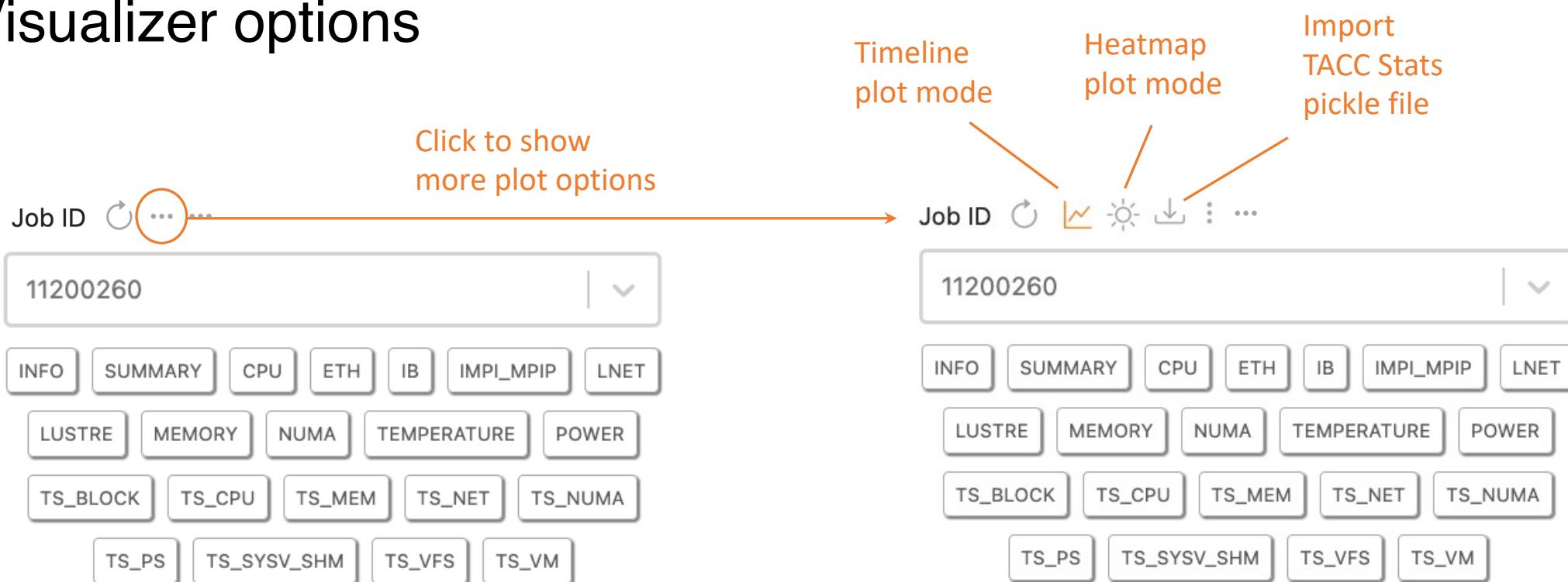
Unavailable
functions are
shown in grey

Case 3 : cancelled job

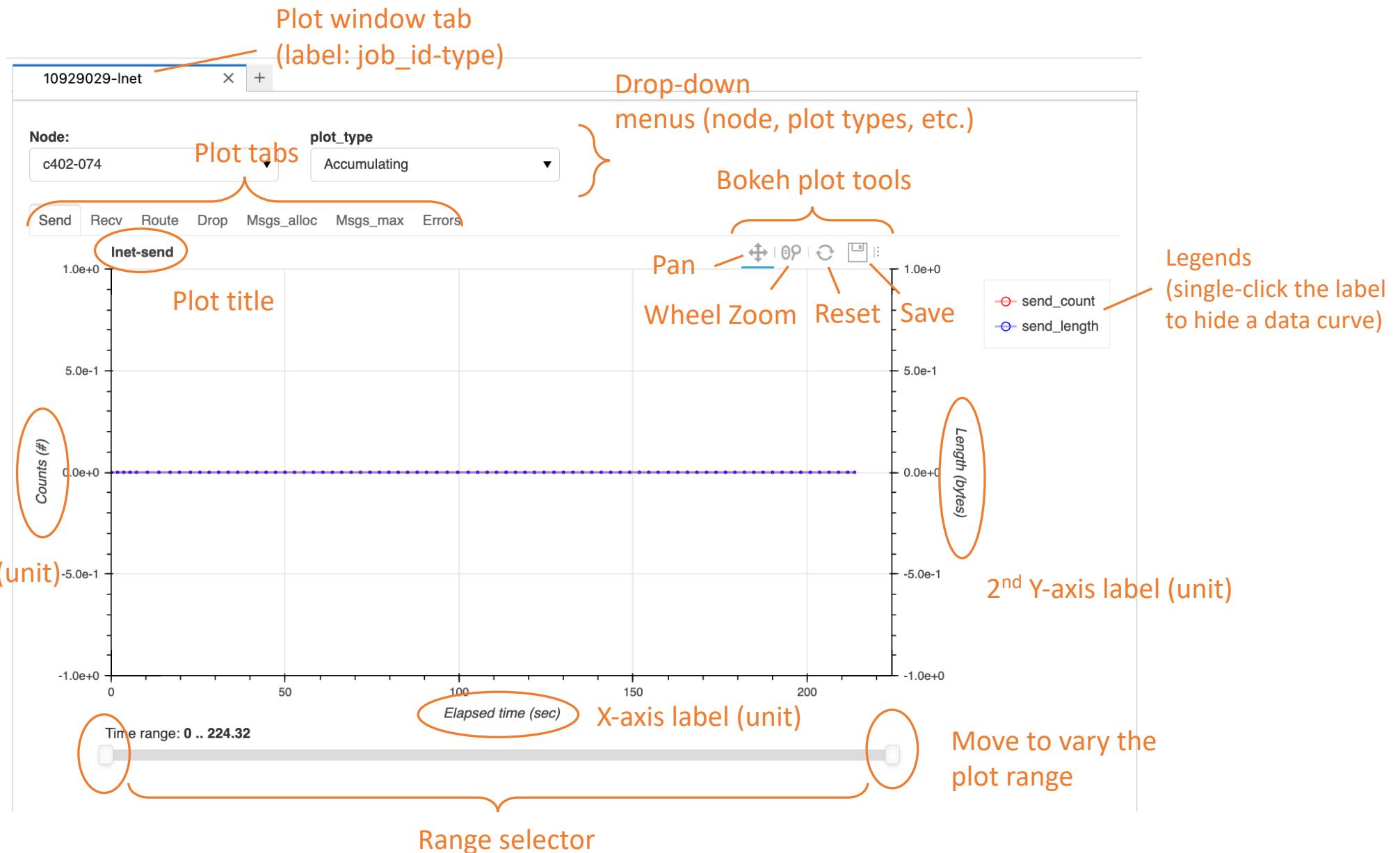


Press the "RECOVER"
button and REMORA
will try to recover
data for visualization

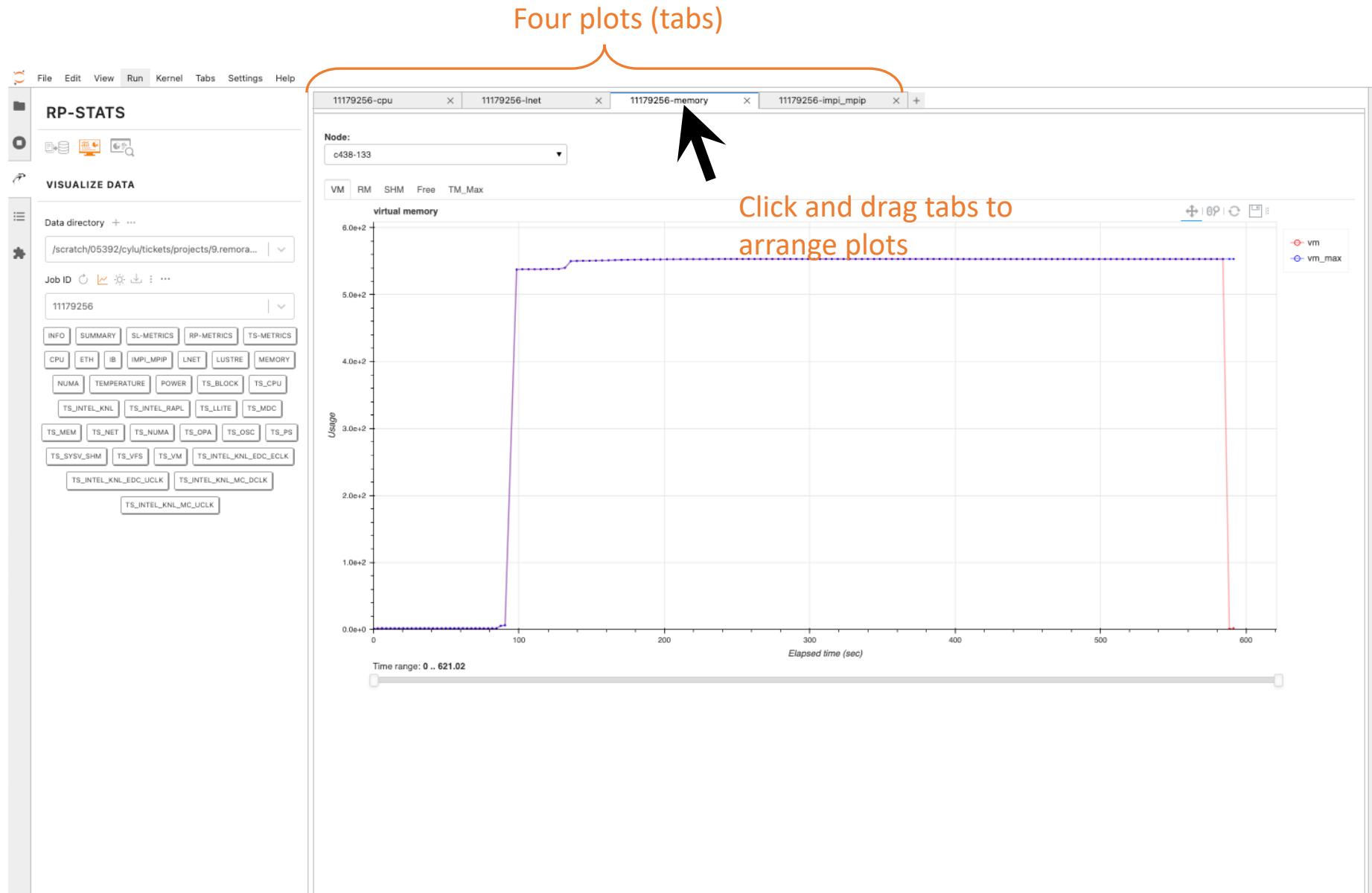
Visualizer options

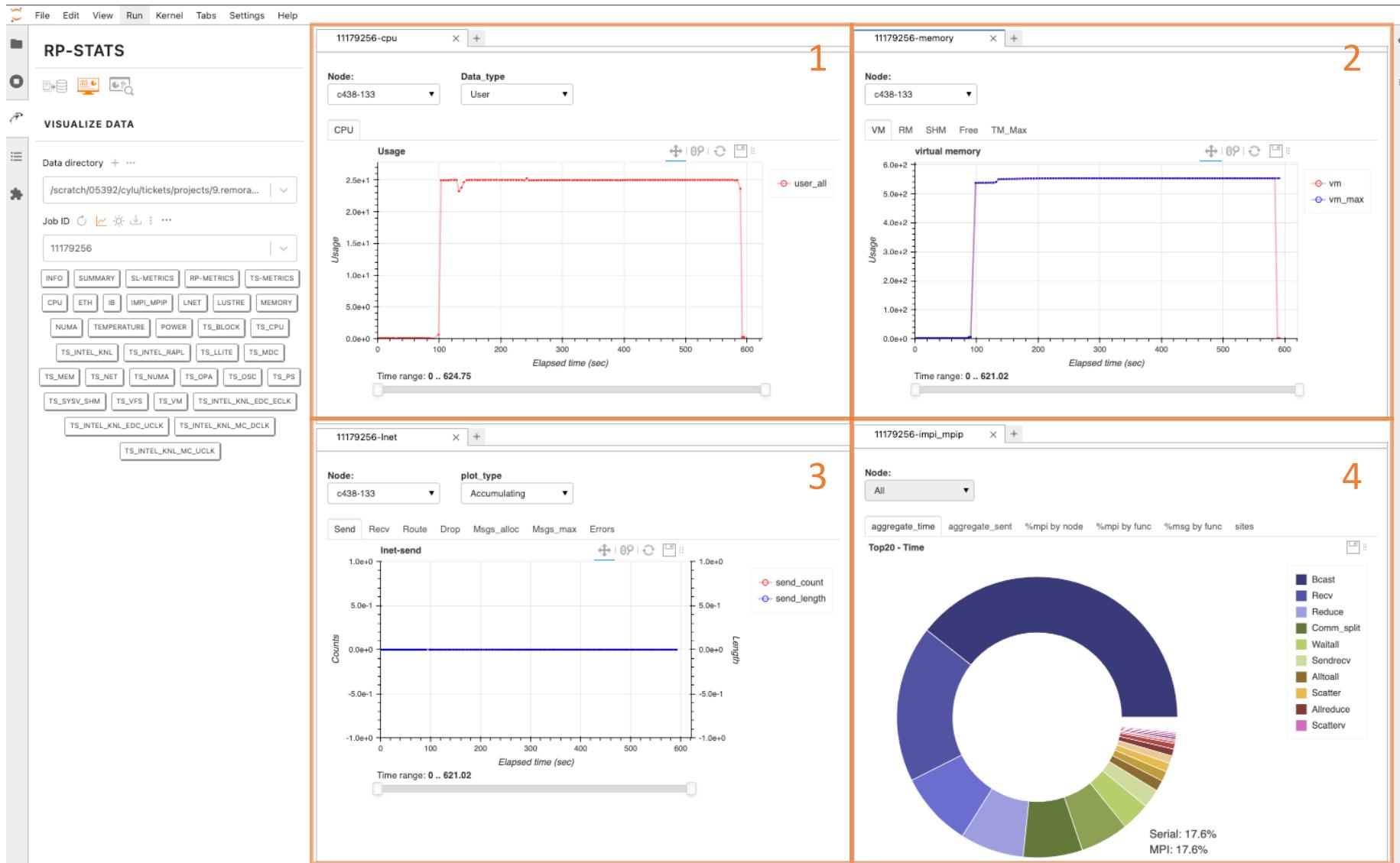


The plot window

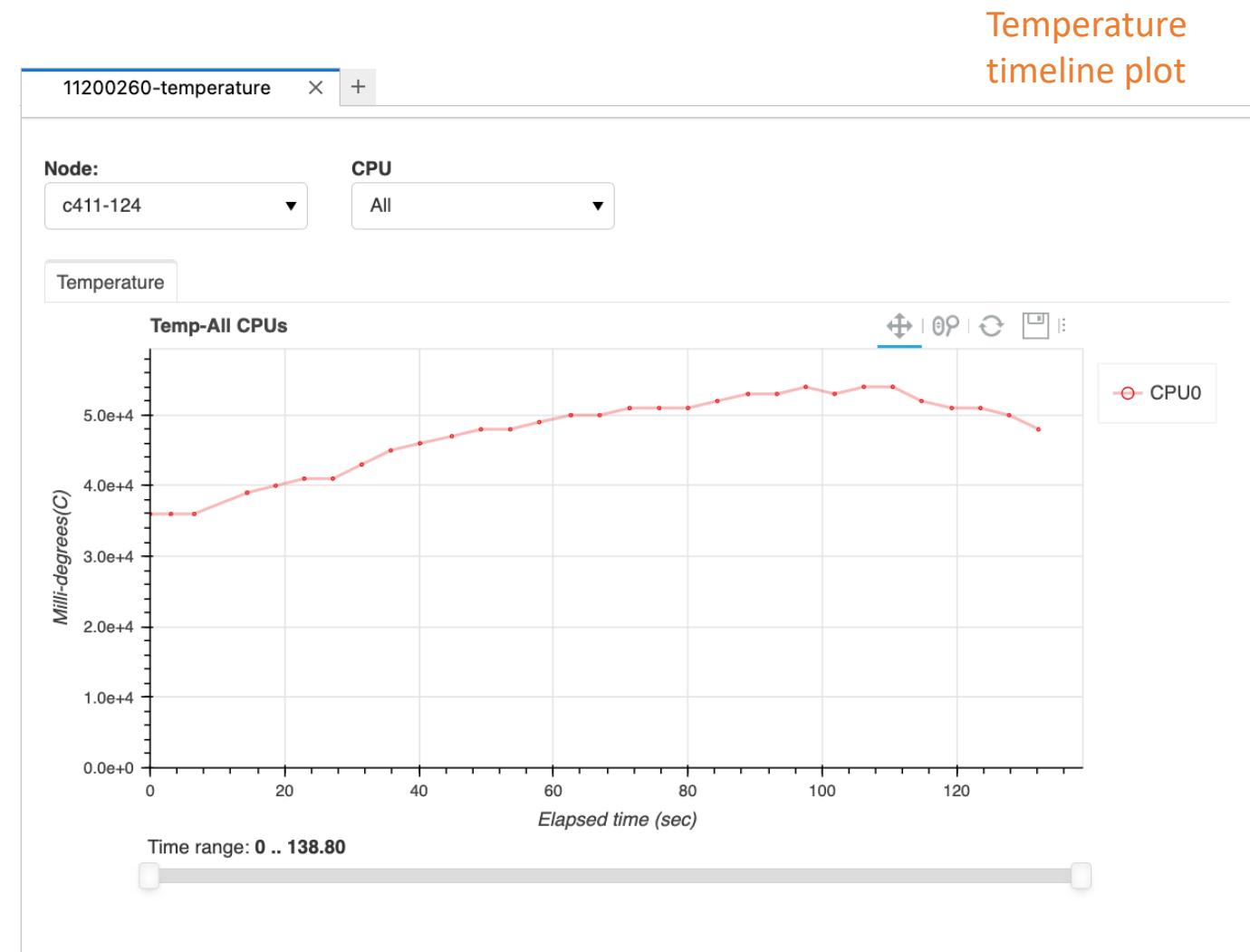
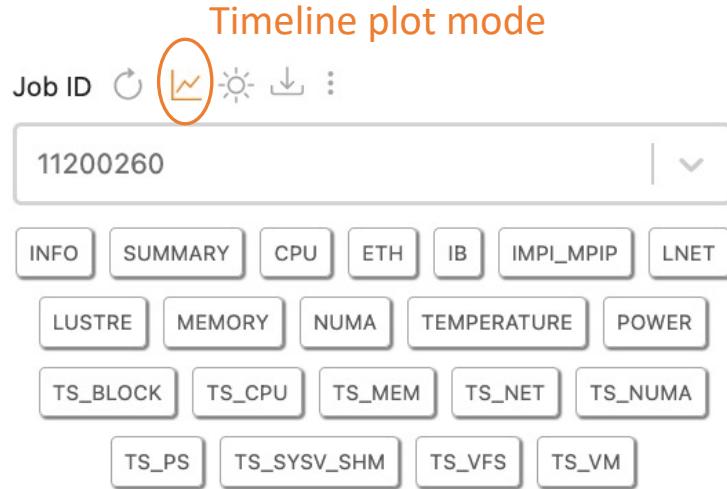


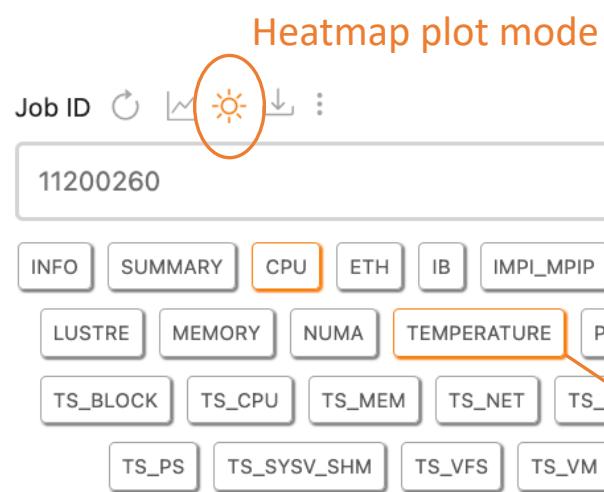
Arrange plots in the main work area



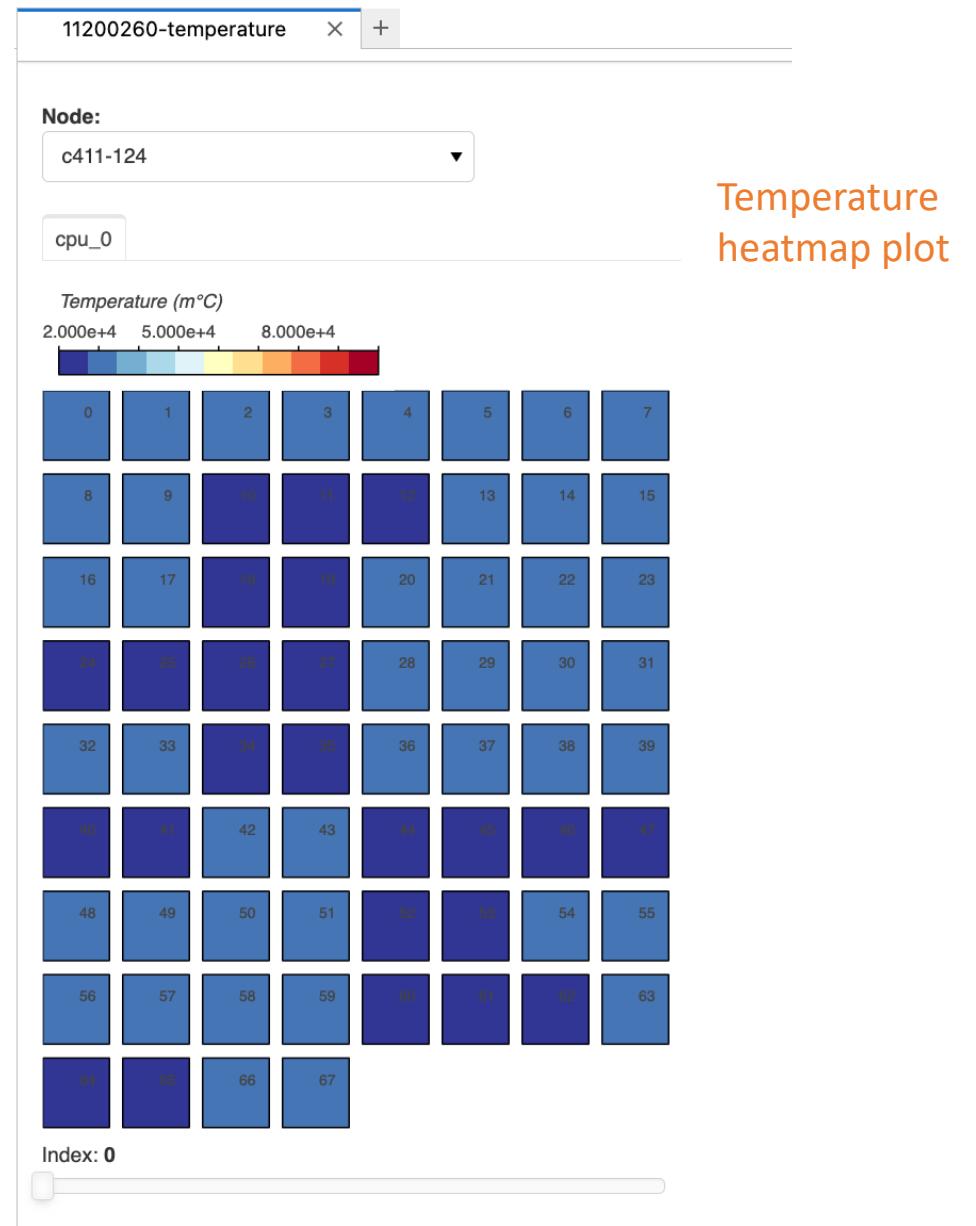


Timeline and heatmap plot mode



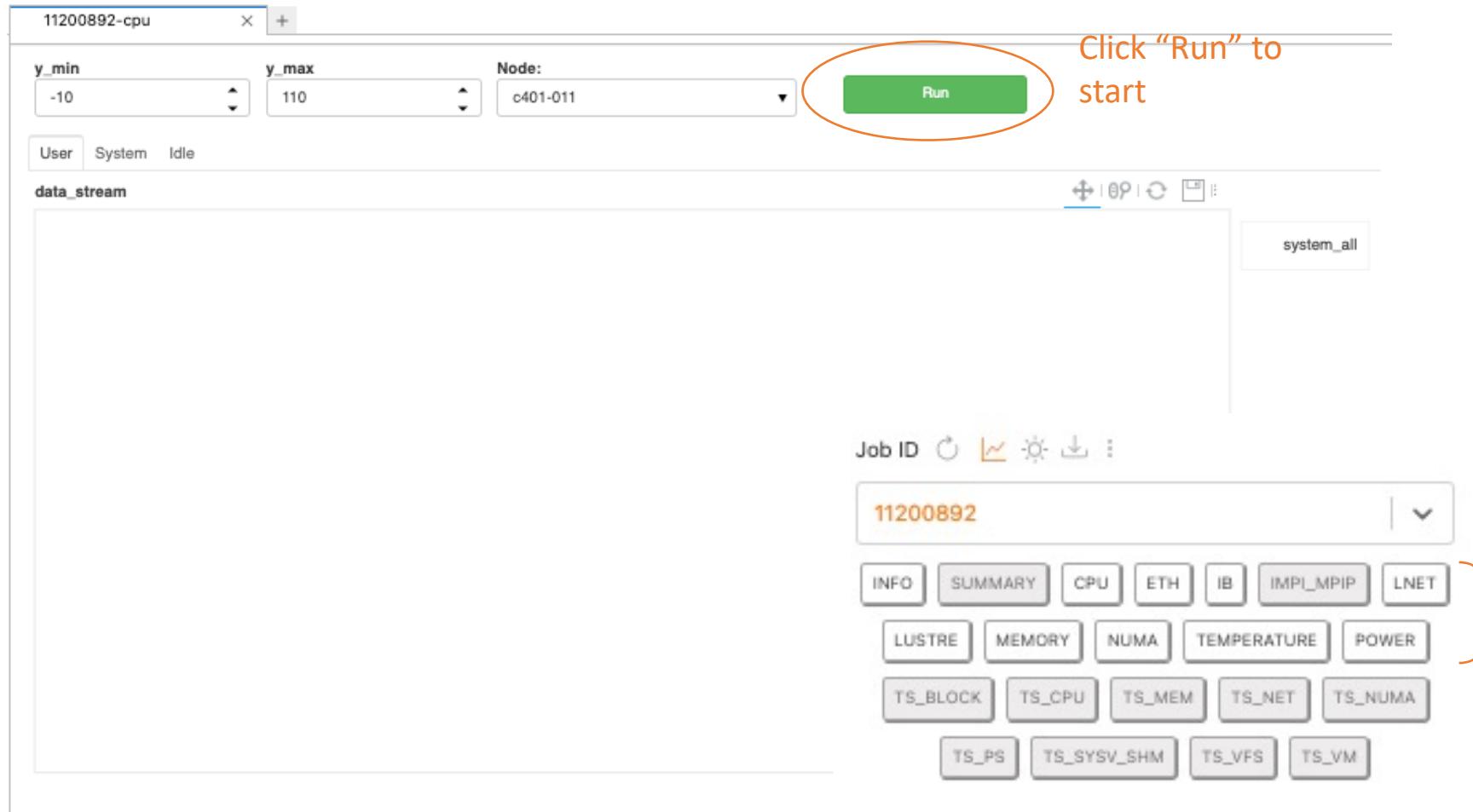


modules highlighted
in orange support
heatmap plot



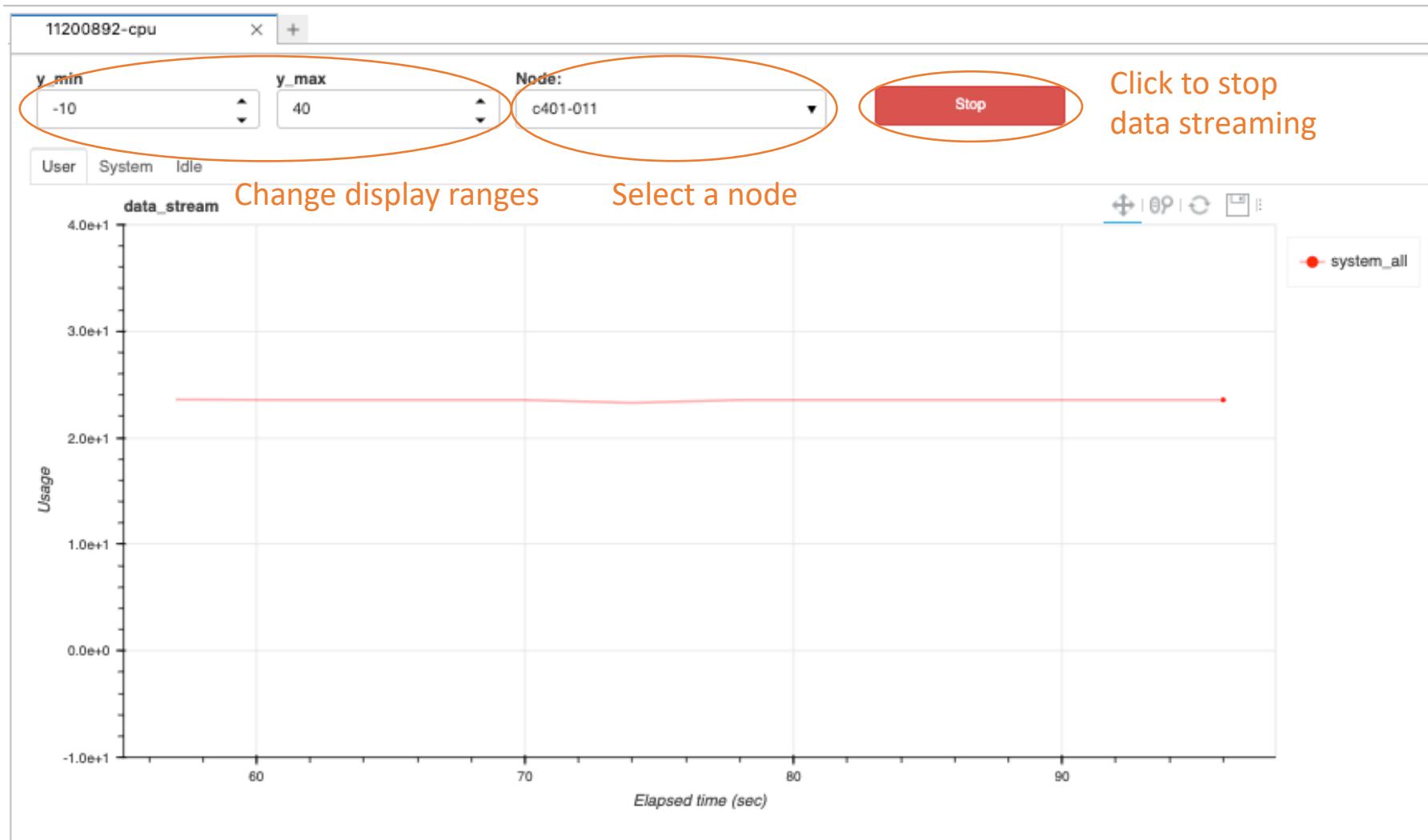
Visualize live streaming data

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



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)

Import TACC
Stats pickle file

Job ID  ...

Enter a Job ID ...

Pickle file directory

Enter a pickle file dir ...

IMPORT

E.g., for Stampede2
`/scratch/projects/tacc_stats/stampede2/pickles/YYYY-MM-DD`

The data analyzer tab



ANALYZE DATA

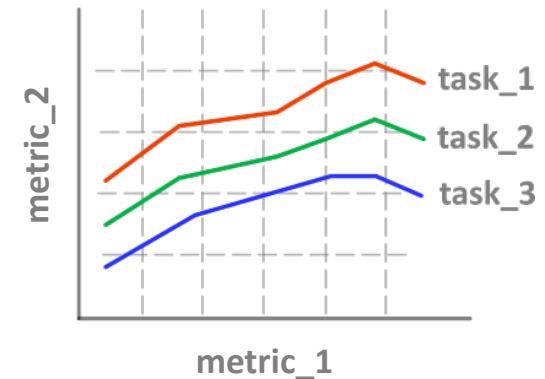
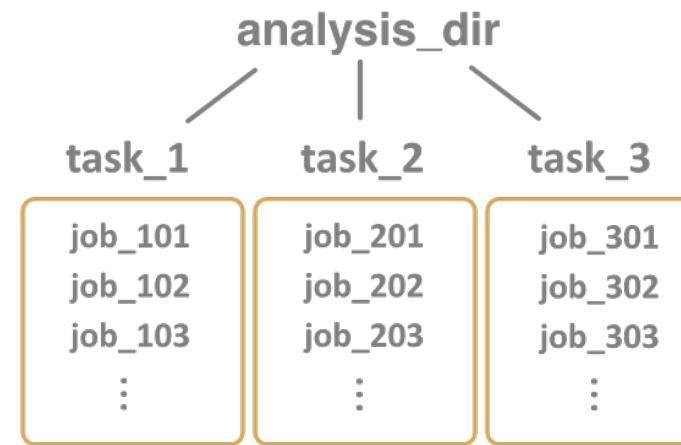
Analysis directory + ...

Task ID +

Create new task Single-task plot Multi-task plot Job metrics table

Register a new analysis directory to store task data

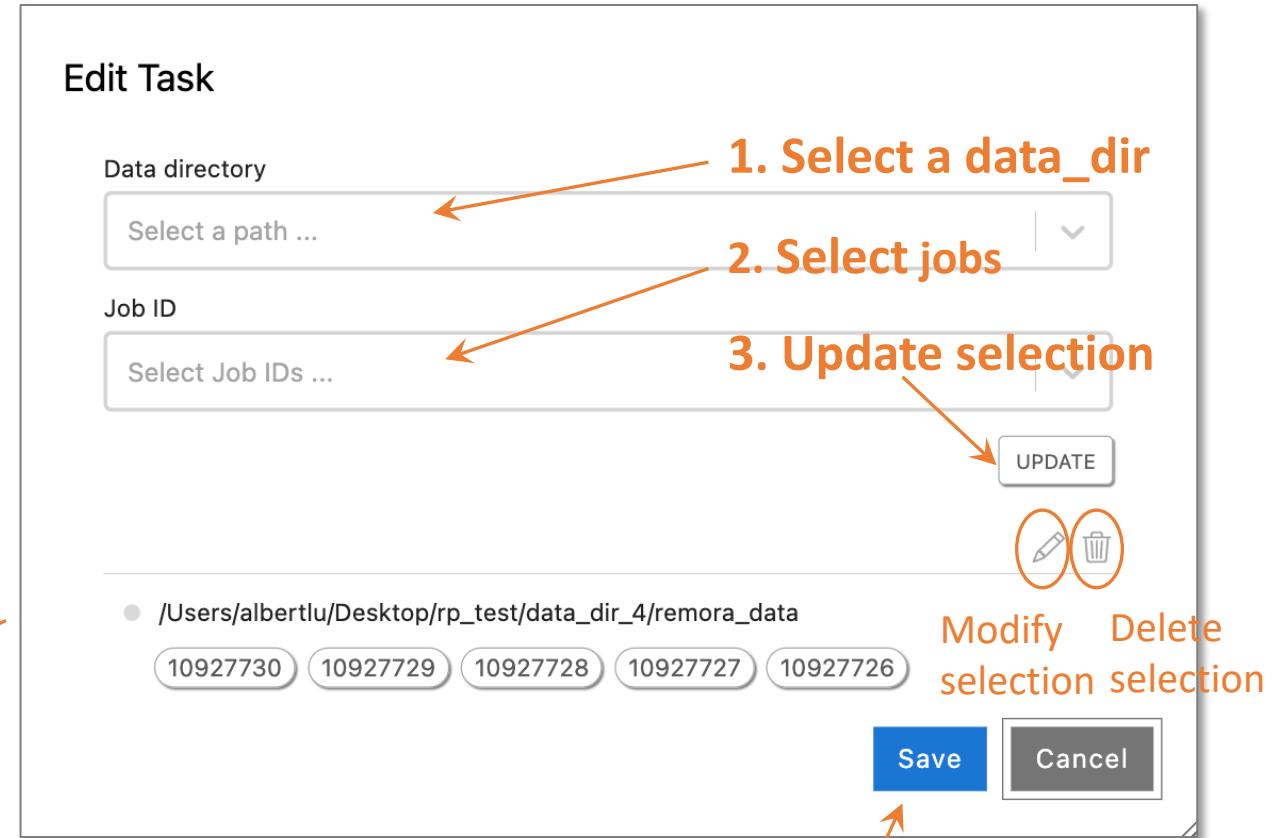
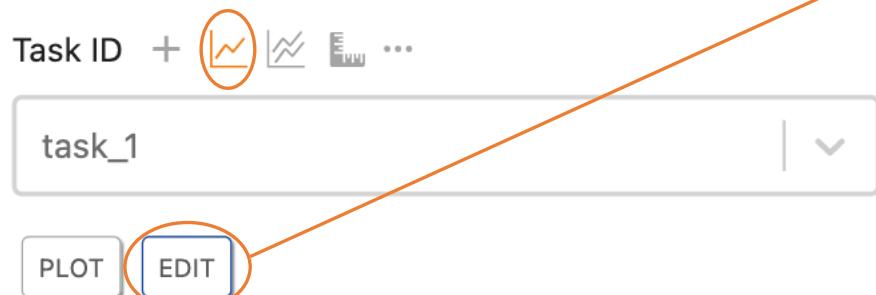
In Analyzer, a “task” is simply defined as a group of “jobs”
“task-plot” is a 2D line plot of any two job metrics



Add jobs to a Task

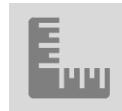


This feature is only available in the single-task plot mode



4. Save changes

The Job metrics table button



Task ID + ... Click to show job metrics table

task_1 Unique Common
metrics metrics | ▾

PLOT EDIT

Task metrics

Add a metric to a job RP SL TS

Job ID	Add & apply a metric function	Metrics
10927726		ts-avg_blockbw
10927727		ts-avg_cputusage
10927728		ts-avg_ethbw
10927729		ts-avg_fabricbw
10927730		ts-avg_flops_64b
		ts-avg_flops_32b
		ts-avg_l1loadhits
		ts-avg_l2loadhits

Select/unselect to show/hide one type of metrics

RP: REMORA

SL: SLURM

TS: TACC Stats

The Job metrics table

Task metrics +

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 metrics +

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: 1.1862×10^{-2} (MB/s/node)
- ts-avg_fabricbw: 0.4336 (MB/s/node)
- ts-avg_flops_64b: 3.6650×10^1 (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_llloadhits: 0 (#/s/node)
- ts-avg_loads: 0 (loads/s/node)
- ts-avg_mbw: 4.4598 (GB/s/node)
- ts-avg_mdreqs: 0.1087 (#/s/node)
- ts-avg_mdcwait: 2.6415×10^2 (microseconds)
- ts-avg_openclose: 1.0487×10^3 (#/s/node)
- ts-avg_oscreas: 7.3615 (#/s/node)

OK

RP SL TS

Job ID

Metrics

ts-avg_flops_32b

ts-avg_l1loadhits

ts-avg_l2loadhits

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: 3.6650×10^1 (GFLOPs/s/node)
- 10927730: 1.1221 (GFLOPs/s/node)

OK

Create and apply a metric function to jobs

Task metrics

Apply metric function RP SL TS

Job ID	Metrics
10927726	ts-avg_blockbw
10927727	ts-avg_cpusage
10927728	ts-avg_ethbw
10927729	ts-avg_fabricbw
10927730	ts-avg_flops_64b ts-avg_flops_32b ts-avg_l1loadhits ts-avg_l2loadhits

Apply metric function +

Select a metric function ...

APPLY **REMOVE** **INFO**

Apply metric function

Create a new metric function

plus_two

APPLY **REMOVE** **INFO**

Apply this metric function to jobs

Remove this metric function from jobs

User metric function info

- **FUNCTION NAME:** plus_two
- **MATH EXPRESSION:** {job_id}+2
- **UNIT:** a.u.
- **REQUIRED METRICS:** job_id

Ok

Create a new metric from existing metrics

The diagram illustrates the process of creating a new metric function from existing metrics. It shows two main interfaces: a left interface for applying metric functions and a right interface for creating new metric functions.

Left Interface (Apply metric function):

- Header: Create a new metric function
- Text input: plus_two
- Buttons: APPLY, REMOVE, INFO

Right Interface (Create new metric function):

- Header: Example
- Text input: Function name: arithmetic_intensity
- Text input: Math expression: Math expression... $\{{\text{ts-avg_flops_64}} + {\text{ts-avg_flops_32}}\}/2\} / \{{\text{ts-avg_mbw}}\}$
- Text input: Metric unit: Unit... flops/byte
- Buttons: ADD, DELETE

Annotations highlight specific elements:

- An orange circle highlights the '+' button in the Apply interface.
- An orange arrow points from the 'plus_two' input in the Apply interface to the 'arithmetic_intensity' input in the Create interface.
- An orange circle highlights the 'ADD' button in the Create interface.
- An orange circle highlights the 'DELETE' button in the Create interface.
- Text on the right side of the Create interface states: "Use curly braces for an existing metric : {metric}"
- Text below the Create interface says: "Register this metric function" and "Delete this metric function".

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 metrics   + 

Click to show the menu

RP SL TS

Job ID	Metrics
10927726	ts-avg_blockbw
10927727	ts-avg_cpusage
10927728	ts-avg_ethbw
10927729	ts-avg_fabricbw
10927730	ts-avg_flops_64b
	ts-avg_flops_32b
	ts-avg_l1loadhits
	ts-avg_l2loadhits

Metric name

Name...

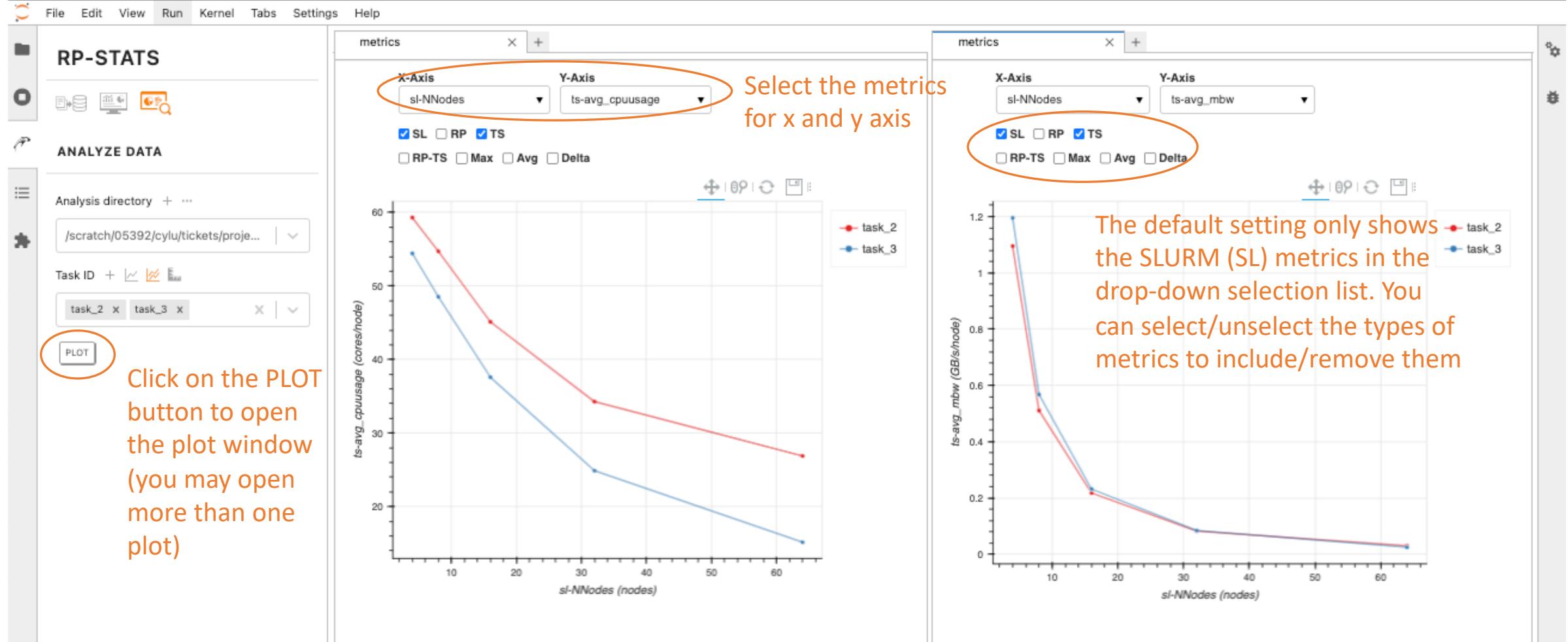
Metric value

Value...

Metric unit

Unit...

Task-metric plot



License

©The University of Texas at Austin, 2023

This work is licensed under the Creative Commons Attribution Non-Commercial 3.0 Unported License. To view a copy of this license, visit
<http://creativecommons.org/licenses/by-nc/3.0/>

When attributing this work, please use the following text: “RP-STATS QuickStart Guide”, Texas Advanced Computing Center, 2023. Available under a Creative Commons Attribution Non-Commercial 3.0 Unported License.

