

Package ‘starvz’

September 1, 2020

Title R-Based Visualization Techniques for Task-Based Applications

Version 0.4.0

Description Performance analysis workflow that combines the power of the R language (and the tidyverse realm) and many auxiliary tools to provide a consistent, flexible, extensible, fast, and versatile framework for the performance analysis of task-based applications that run on top of the StarPU runtime (with its MPI (Message Passing Interface) layer for multi-node support). Its goal is to provide a fruitful prototypical environment to conduct performance analysis hypothesis-checking for task-based applications that run on heterogeneous (multi-GPU, multi-core) multi-node HPC (High-performance computing) platforms.

URL <https://github.com/schnorr/starvz>

BugReports <https://github.com/schnorr/starvz/issues>

Depends R (>= 3.6.0)

Imports methods, grDevices, stats, utils, magrittr, dplyr, ggplot2, tibble, rlang, tidyr, patchwork, purrr, readr, stringr, yaml, lpSolve, gtools, data.tree, RColorBrewer, zoo, car, arrow (>= 0.17.0)

License GPL-3

Encoding UTF-8

SystemRequirements C++, bash, StarPU

LazyData true

LinkingTo Rcpp, BH

RoxygenNote 7.1.1

Collate 'RcppExports.R' 'starvz_data.R' 'phase1.R'
'phase1_parse_csv.R' 'phase2.R' 'phase2_aggregation.R'
'phase2_applications.R' 'phase2_atree.R' 'phase2_config.R'
'phase2_gaps.R' 'phase2_imbalance.R' 'phase2_kchart.R'
'phase2_lackready.R' 'phase2_memory.R' 'phase2_metrics.R'
'phase2_mpi.R' 'phase2_node_summary.R' 'phase2_pmtree.R'

'phase2_states_chart.R' 'phase2_themes.R'
 'phase2_time_integration.R' 'phase2_util.R'
 'phase2_var_chart.R' 'phase2_var_panels.R' 'read_functions.R'
 'write_functions.R'

Suggests testthat

NeedsCompilation yes

Author Lucas Mello Schnorr [aut, ths]

(<<https://orcid.org/0000-0003-4828-9942>>),

Vinicius Garcia Pinto [aut] (<<https://orcid.org/0000-0002-6845-9358>>),

Lucas Leandro Nesi [aut, cre] (<<https://orcid.org/0000-0001-8874-1839>>),

Marcelo Cogo Miletto [aut] (<<https://orcid.org/0000-0002-1191-3863>>),

Guilherme Alles [ctb],

Arnaud Legrand [ctb],

Luka Stanisic [ctb],

Rémy Drouilhet [ctb]

Maintainer Lucas Leandro Nesi <lucas.nesi@inf.ufrgs.br>

Repository CRAN

Date/Publication 2020-09-01 16:20:02 UTC

R topics documented:

cholesky_colors	3
data_name_coordinates	4
data_name_handle	4
data_name_tag	5
handles_presence_states	5
lu_colors	6
panel_activenodes	6
panel_atree	7
panel_atree_structure	8
panel_gflops	9
panel_gpubandwidth	10
panel_handles	11
panel_hete_imbalance	11
panel_imbalance	12
panel_kiteration	13
panel_lackready	14
panel_memory_snap	15
panel_memory_state	16
panel_mpibandwidth	17
panel_mpiconcurrent	18
panel_mpiconcurrentout	19
panel_mpistate	20
panel_nodememuse	21
panel_node_summary	21
panel_pmttool_kiteration	22

panel_pmtool_st	23
panel_power_imbalance	24
panel_ready	25
panel_st	26
panel_st_agg_dynamic	26
panel_st_agg_static	27
panel_st_raw	28
panel_st_runtime	29
panel_submitted	30
panel_title	31
panel_usedmemory	31
panel_utilheatmap	32
panel_utiltreedepth	33
panel_utiltreenode	34
pre_handle_gantt	35
qrmumps_colors	35
starvz_assemble	36
starvz_check_data	36
starvz_phase1	37
starvz_plot	38
starvz_plot_list	39
starvz_read	39
starvz_read_config	40
starvz_sample_lu	40
starvz_set_log	41
Index	42

cholesky_colors	<i>Colors for lu</i>
-----------------	----------------------

Description

This will be deprecated

Usage

cholesky_colors()

data_name_coordinates *Handles Name coordinates*

Description

Give handles name by their coordinates

Usage

```
data_name_coordinates(df)
```

Arguments

df data_handle table of Starvz data

Value

data_handle table with new column Value with the name

Examples

```
data_name_coordinates(starvz_sample_lu$Data_handle)
```

data_name_handle *Handles Name address*

Description

Give handles name by their address

Usage

```
data_name_handle(df)
```

Arguments

df data_handle table of Starvz data

Value

data_handle table with new column Value with the name

Examples

```
data_name_handle(starvz_sample_lu$Data_handle)
```

data_name_tag	<i>Handles Name Tag</i>
---------------	-------------------------

Description

Give handles name by their tag

Usage

```
data_name_tag(df)
```

Arguments

df data_handle table of Starvz data

Value

data_handle table with new column Value with the name

Examples

```
data_name_tag(starvz_sample_1u$Data_handle)
```

handles_presence_states	<i>Computes presence of handles over resources</i>
-------------------------	--

Description

Use for precomputation of other memory-related functions

Usage

```
handles_presence_states(data)
```

Arguments

data starvz_data with trace data

Value

Time-Step aggregated handle presences

Examples

```
handles_presence_states(starvz_sample_lu)
```

lu_colors	<i>Colors for lu</i>
-----------	----------------------

Description

This will be deprecated

Usage

```
lu_colors()
```

panel_activenodes	<i>Create the active nodes in memory plot</i>
-------------------	---

Description

Use starvz_data to create a line plot of the number of active nodes per type along the application execution time

Usage

```
panel_activenodes(
  data = NULL,
  step = data$config$activenodes$aggregation$step,
  aggregation = data$config$activenodes$aggregation$active,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  legend = data$config$activenodes$legend
)
```

Arguments

data	starvz_data with trace data
step	size in milliseconds for the time aggregation step
aggregation	enable/disable time aggregation for the plot
x_start	X-axis start value
x_end	X-axis end value
legend	enable/disable plot legends

Value

A ggplot object

Examples

```
## Not run:
panel_activenodes(data=starvz_sample_lu, step=100)

## End(Not run)
```

panel_atree	<i>Create the elimination tree plot with some options in the config file</i>
-------------	--

Description

Use starvz_data to create a representation of the elimination tree structure considering initialization, communication, and computational tasks. These representations can be controlled in the configuration file.

Usage

```
panel_atree(
  data = NULL,
  step = data$config$atree$step,
  legend = data$config$atree$legend,
  zoom = FALSE,
  computation = data$config$atree$computation$active,
  pruned = data$config$atree$computation$pruned$active,
  initialization = data$config$atree$initialization$active,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  communication = data$config$atree$communication$active,
  anomalies = data$config$atree$anomalies$active
)
```

Arguments

data	starvz_data with trace data
step	size in milliseconds for the time aggregation step
legend	enable/disable panel legend
zoom	enable/disable vertical zoom in the tree structure
computation	enable/disable computations representations in the tree
pruned	enable/disable pruned computations representations in the tree
initialization	enable/disable initialization tasks representation
x_start	X-axis start value

x_end	X-axis end value
communication	enable/disable communication tasks representation
anomalies	enable/disable anomalies tasks representation

Value

A ggplot object

Examples

```
## Not run:  
panel_atree(starvz_sample_lu, step=10)  
panel_atree(starvz_sample_lu, step=20, communication=FALSE, initialization=FALSE)  
  
## End(Not run)
```

panel_atree_structure *Create the elimination tree structure plot along time*

Description

Use Atree and Application data to create the elimination tree structure plot in a ggplot object and return it

Usage

```
panel_atree_structure(data = NULL)
```

Arguments

data starvz_data with trace data

Value

A ggplot object

Examples

```
## Not run:  
panel_atree_structure(starvz_sample_lu)  
  
## End(Not run)
```


panel_gflops

*Create a line chart panel with GFlops***Description**

Use the Variable traces to create a line chart panel with GFlops per resource, aggregated by a configurable time step

Usage

```
panel_gflops(
  data,
  legend = data$config$gflops$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$gflops$limit,
  step = data$config$gflops$step
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```
panel_gflops(data=starvz_sample_1u)
```

panel_gpubandwidth *Create a line chart panel with GPU bandwidth*

Description

Use the Variable traces to create a line chart panel with GPU bandwidth per resource, aggregated by a configurable time step

Usage

```
panel_gpubandwidth(
  data,
  legend = data$config$gpubandwidth$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$gpubandwidth$limit,
  step = data$config$gpubandwidth$step,
  total = data$config$gpubandwidth$total
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation
total	show total bandwidth of the execution per resource

Value

A ggplot object

Examples

```
panel_gpubandwidth(data=starvz_sample_lu)
```

panel_handles	<i>Create a space time visualization of data handles</i>
---------------	--

Description

Visualizate data handles movement To accelerate the process:

```
data$handle_states <- handles_presence_states(data)
data$handle_gantt_data <- pre_handle_gantt(data)
To Select time:
handles_gantt(data, JobId=c(jobid))
snap_data <- pre_snap(data, data$handle_states)
memory_snap(snap_data, 1000, tasks_size=200, step=1)
```

Usage

```
panel_handles(data, JobId = NA, lines = NA, lHandle = NA)
```

Arguments

data	starvz_data with trace data
JobId	Select handles of jobid
lines	vertical lines
lHandle	select handles

Value

A ggplot object

Examples

```
panel_handles(data = starvz_sample_lu)
```

panel_hete_imbalance	<i>Create a line chart with heterogeneous resources and tasks imbalance metrics</i>
----------------------	---

Description

This function creates a line chart with imbalance metrics. The function applies the metrics on fixed time-steps defined by the user. The metrics consider that the resources are heterogeneous, and each task has a different performance per resource.

Usage

```

panel_hete_imbalance(
  data,
  legend = data$config$hete_imbalance$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$hete_imbalance$limit,
  step = data$config$hete_imbalance$step
)

```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```

panel_hete_imbalance(data = starvz_sample_lu)

```

panel_imbalance

Create a line chart with homogeneous imbalance metrics.

Description

This function creates a line chart with imbalance metrics. The function applies the metrics on fixed time-steps defined by the user. The metrics consider that the resources are homogeneous.

Usage

```

panel_imbalance(
  data,
  legend = data$config$imbalance$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$imbalance$limit,
  step = data$config$imbalance$step
)

```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```
panel_imbalance(data = starvz_sample_lu)
```

panel_kiteration

Create a special chart for applications with iterations

Description

Plot iterations Y over Time X

Usage

```

panel_kiteration(
  data = NULL,
  middle_lines = data$config$kiteration$middlelines,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  legend = data$config$ready$legend,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  per_node = data$config$kiteration$pernode
)

```

Arguments

data	starvz_data with trace data
middle_lines	plot a middle line
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
legend	enable/disable legends
x_start	X-axis start value
x_end	X-axis end value
per_node	Create node facets

Value

A ggplot object

Examples

```
panel_kiteration(data = starvz_sample_lu)
```

panel_lackready	<i>Shows if the runtimes is lacking ready tasks</i>
-----------------	---

Description

Plot a bar over time that shows when the runtime is lacking ready tasks

Usage

```

panel_lackready(
  data = NULL,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end
)

```

Arguments

data starvz_data with trace data
x_start X-axis start value
x_end X-axis end value

Value

A ggplot object

Examples

```
panel_lackready(data = starvz_sample_lu)
```

panel_memory_snap *Create a snapshot of memory*

Description

Visualize memory in a specific time

Usage

```
panel_memory_snap(data, selected_time, step, tasks_size = 30)
```

Arguments

data starvz_data with trace data
selected_time time
step for discrete events
tasks_size size of tasks in the visualization

Value

A ggplot object

Examples

```
panel_memory_snap(data = starvz_sample_lu, 100, 10)
```

panel_memory_state *Create a memory state space time*

Description

Show memory events

Usage

```
panel_memory_state(  
  data = NULL,  
  combined = data$config$memory$combined,  
  legend = data$config$memory$legend,  
  base_size = data$config$base_size,  
  expand_x = data$config$expand,  
  x_start = data$config$limits$start,  
  x_end = data$config$limits$end  
)
```

Arguments

data	starvz_data with trace data
combined	shows links
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value

Value

A ggplot object

Examples

```
panel_memory_state(data = starvz_sample_lu)
```

panel_mpibandwidth *Create a line chart panel with MPI bandwidth*

Description

Use the Variable traces to create a line chart panel with MPI bandwidth per node, aggregated by a configurable time step

Usage

```
panel_mpibandwidth(
  data,
  legend = data$config$mpibandwidth$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$mpibandwidth$limit,
  step = data$config$mpibandwidth$step
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```
panel_mpibandwidth(data=starvz_sample_lu)
```

panel_mpiconcurrent *Create a line chart panel with MPI concurrent*

Description

Use the Variable traces to create a line chart panel with MPI concurrent per node, aggregated by a configurable time step

Usage

```
panel_mpiconcurrent(
  data,
  legend = data$config$mpiconcurrent$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$mpiconcurrent$limit,
  step = data$config$mpiconcurrent$step
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```
panel_mpiconcurrent(data=starvz_sample_lu)
```

 panel_mpiconcurrentout

Create a line chart panel with MPI concurrent out

Description

Use the Variable traces to create a line chart panel with MPI concurrent out per node, aggregated by a configurable time step

Usage

```
panel_mpiconcurrentout(
  data,
  legend = data$config$mpiconcurrentout$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$mpiconcurrentout$limit,
  step = data$config$mpiconcurrentout$step
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```
panel_mpiconcurrentout(data=starvz_sample_lu)
```

panel_mpistate *Create a space-time view of MPI controlers*

Description

Create a space-time view of MPI controlers

Usage

```
panel_mpistate(  
  data = NULL,  
  legend = data$config$mpibandwidth$legend,  
  base_size = data$config$base_size,  
  expand_x = data$config$expand,  
  x_start = data$config$limits$start,  
  x_end = data$config$limits$end,  
  y_start = 0,  
  y_end = data$config$mpibandwidth$limit  
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value

Value

A ggplot object

Examples

```
panel_mpistate(data=starvz_sample_lu)
```

panel_nodememuse	<i>Create the node memory usage plot</i>
------------------	--

Description

Use starvz_data to create a line plot of the memory usage in MB of active nodes along the application execution time

Usage

```
panel_nodememuse(
  data = NULL,
  step = data$config$activenodes$aggregation$step,
  aggregation = data$config$activenodes$aggregation$active,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  legend = data$config$activenodes$nodememuse$legend
)
```

Arguments

data	starvz_data with trace data
step	size in milliseconds for the time aggregation step
aggregation	enable/disable time aggregation for the plot
x_start	X-axis start value
x_end	X-axis end value
legend	enable/disable plot legends

Examples

```
## Not run:
panel_nodememuse(starvz_sample_lu, step=100)

## End(Not run)
```

panel_node_summary	<i>Create a bar plot with node information</i>
--------------------	--

Description

Bar plot with makespan and abe per node

Usage

```
panel_node_summary(
  data,
  legend = data$config$summary_nodes$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value

Value

A ggplot object

Examples

```
panel_node_summary(data = starvz_sample_1u)
```

```
panel_pmtree_kiteration
```

Create a special chart for applications with iterations with PMtool data

Description

Plot iterations Y over Time X of PMtool data

Usage

```
panel_pmtree_kiteration(
  data = NULL,
  legend = data$config$pmtree$kiteration$legend,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
x_start	X-axis start value
x_end	X-axis end value

Value

A ggplot object

Examples

```
panel_pmtree_kiteration(data = starvz_sample_lu)
```

panel_pmtree_st	<i>Create a space time visualization of pmtree application as a Gantt chart</i>
-----------------	---

Description

Use the PMTOOL Application trace data to plot the task computations by ResourceId over the execution time.

Usage

```
panel_pmtree_st(
  data = NULL,
  legend = data$config$pmtree$state$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value

Value

A ggplot object

Examples

```
panel_pmtool_st(data = starvz_sample_lu)
```

panel_power_imbalance *Create a line chart with heterogeneous imbalance metrics.*

Description

This function creates a line chart with imbalance metrics. The function applies the metrics on fixed time-steps defined by the user. The metrics consider that the resources are heterogeneous and defined by a constant power factor. For the effects of this function, one task is select for computing the relative power between resources.

Usage

```
panel_power_imbalance(
  data,
  legend = data$config$power_imbalance$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$power_imbalance$limit,
  step = data$config$power_imbalance$step,
  task = data$config$power_imbalance$task
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation
task	Task used to computer relative resource power.

Value

A ggplot object

Examples

```
panel_power_imbalance(data = starvz_sample_lu)
```

panel_ready	<i>Create a line chart panel with ready tasks submission</i>
-------------	--

Description

Use the Variable traces to create a line chart panel with ready tasks submission per node, aggregated by a configurable time step

Usage

```
panel_ready(
  data,
  legend = data$config$ready$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$ready$limit,
  step = data$config$ready$step
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```
panel_ready(data = starvz_sample_lu)
```

panel_st	<i>Create a space time visualization of the application as a Gantt chart</i>
----------	--

Description

Use the Application trace data to plot the task computations by ResourceId over the execution time. It will select the aggregation mode if requested.

Usage

```
panel_st(
  data,
  agg = data$config$st$aggregation$active,
  agg_met = data$config$st$aggregation$method
)
```

Arguments

data	starvz_data with trace data
agg	boolean Active or not aggregation
agg_met	Aggregation method, possible: static, dynamic, nodes

Value

A ggplot object

Examples

```
panel_st(data = starvz_sample_lu)
```

panel_st_agg_dynamic	<i>Create a space-time visualization with dynamic aggregation.</i>
----------------------	--

Description

Use any state trace data to plot the task computations by ResourceId over the execution time with Gantt Chart. This function dynamically aggregate states with a dynamic/automatic time-step.

Usage

```
panel_st_agg_dynamic(
  data = NULL,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  expand_x = data$config$st$expand
)
```

Arguments

data	starvz_data with trace data
x_start	X-axis start value
x_end	X-axis end value
expand_x	expand size for scale_x_continuous padding

Value

A ggplot object

Examples

```
panel_st_agg_dynamic(data = starvz_sample_lu)
```

panel_st_agg_static *Create a space-time visualization with static aggregation.*

Description

Use any state trace data to plot the task computations by ResourceId over the execution time with Gantt Chart. This function aggregate states with a static/user-defined time-step.

Usage

```
panel_st_agg_static(
  data = NULL,
  runtime = FALSE,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  step = data$config$st$aggregation$step
)
```

Arguments

data	starvz_data with trace data
runtime	if this is runtime data
x_start	X-axis start value
x_end	X-axis end value
step	time-step

Value

A ggplot object

Examples

```
panel_st_agg_static(data = starvz_sample_lu)
```

```
panel_st_raw
```

Create a space time visualization as a Gantt chart

Description

Use the Application trace data to plot the task computations by ResourceId over the execution time.

Usage

```
panel_st_raw(
  data = NULL,
  ST.Outliers = data$config$st$outliers,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  expand_y = data$config$st$expand,
  selected_nodes = data$config$selected_nodes,
  labels = data$config$st$labels,
  alpha = data$config$st$alpha,
  idleness = data$config$st$idleness,
  taskdeps = data$config$st$tasks$active,
  tasklist = data$config$st$tasks$list,
  levels = data$config$st$tasks$levels,
  makespan = data$config$st$makespan,
  abe = data$config$st$abe$active,
  pmtoolbounds = data$config$pmtool$bounds$active,
  cpb = data$config$st$cpb,
  cpb_mpi = data$config$st$cpb_mpi$active,
  legend = data$config$st$legend,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  runtime = FALSE
)
```

Arguments

data	starvz_data with trace data
ST.Outliers	enable/disable the anomalous task highlighting
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
expand_y	expand size for scale_y_continuous padding

selected_nodes	select only some nodes in some plots
labels	control resources labels: [ALL, 1CPU_per_NODE, 1GPU_per_NODE]
alpha	alpha value for non-anomalous tasks
idleness	enable/disable idleness percentages in the plot
taskdeps	enable/disable task deps path highlighting
tasklist	list of JobIds to highlight the dependencies
levels	number of dependencies to be shown
makespan	enable/disable application makespan at the end of the plot
abe	enable/disable ABE metric
pmtoolbounds	enable/disable pmtool theoretical bounds
cpb	enable/disable critical path bound makespan metric
cpb_mpi	enable/disable critical path bound makespan considering MPI
legend	enable/disable legends
x_start	X-axis start value
x_end	X-axis end value
runtime	TODO I think we should create a separated function for it

Value

A ggplot object

Examples

```
panel_st_raw(data = starvz_sample_lu)
```

panel_st_runtime *Create a space time visualization of the runtime as a Gantt chart*

Description

Use the runtime trace data to plot the task computations by ResourceId over the execution time. It will select the aggregation mode if requested, only static aggregation is available for runtime.

Usage

```
panel_st_runtime(data, agg = data$config$starpu$aggregation$active)
```

Arguments

data	starvz_data with trace data
agg	Active or not static aggregation

Value

A ggplot object

Examples

```
panel_st_runtime(data = starvz_sample_lu)
```

panel_submitted	<i>Create a line chart panel with submitted tasks submission</i>
-----------------	--

Description

Use the Variable traces to create a line chart panel with submitted tasks submission per node, aggregated by a configurable time step

Usage

```
panel_submitted(  
  data,  
  legend = data$config$submitted$legend,  
  base_size = data$config$base_size,  
  expand_x = data$config$expand,  
  x_start = data$config$limits$start,  
  x_end = data$config$limits$end,  
  y_start = 0,  
  y_end = data$config$submitted$limit,  
  step = data$config$submitted$step  
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```
panel_submitted(data = starvz_sample_lu)
```

panel_title	<i>Create the title of StarVZ plot</i>
-------------	--

Description

Use the directory of traces name to create a plot title

Usage

```
panel_title(data)
```

Arguments

data starvz_data with trace data

Value

A ggplot object

Examples

```
panel_title(data = starvz_sample_lu)
```

panel_usedmemory	<i>Create a line chart panel with used memory</i>
------------------	---

Description

Use the Variable traces to create a line chart panel with used memory per resource, aggregated by a configurable time step

Usage

```
panel_usedmemory(
  data,
  legend = data$config$usedmemory$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = data$config$usedmemory$limit,
  step = data$config$usedmemory$step
)
```

Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```
panel_usedmemory(data = starvz_sample_lu)
```

panel_utilheatmap *Create a heatmap of resource utilization*

Description

Similar to the other resource oriented plots but shows the utilization per time step

Usage

```
panel_utilheatmap(
  data,
  legend = data$config$utilheatmap$legend,
  base_size = data$config$base_size,
  expand_x = data$config$expand,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  y_start = 0,
  y_end = NA,
  step = data$config$utilheatmap$step
)
```


Arguments

data	starvz_data with trace data
legend	enable/disable legends
base_size	base_size base font size
expand_x	expand size for scale_x_continuous padding
x_start	X-axis start value
x_end	X-axis end value
y_start	Y-axis start value
y_end	Y-axis end value
step	time step for aggregation

Value

A ggplot object

Examples

```
panel_utilheatmap(data = starvz_sample_lu)
```

panel_utiltreedepth *Create the resource utilization by tree depth plot*

Description

Use starvz_data Application and Atree to create a plot that shows the total resource utilization, painted by tree depth level using geom_ribbon

Usage

```
panel_utiltreedepth(
  data,
  step = data$config$utiltreenode$step,
  x_start = data$config$limits$start,
  x_end = data$config$limits$end,
  legend = data$config$utiltreedepth$legend
)
```

Arguments

data	starvz_data with trace data
step	size in milliseconds for the time aggregation step
x_start	X-axis start value
x_end	X-axis end value
legend	enable/disable plot legends

Value

A ggplot object

Examples

```
## Not run:  
panel_utiltreedepth(starvz_sample_lu, step=100, legend=TRUE)  
  
## End(Not run)
```

panel_utiltreenode *Create the resource utilization by tree node plot*

Description

Use starvz_data Application and Atree to create a plot that shows the total resource utilization, painted by tree node using geom_ribbon. The colors are reused between nodes, not tied to a specific tree node.

Usage

```
panel_utiltreenode(  
  data = NULL,  
  step = data$config$utiltreenode$step,  
  x_start = data$config$limits$start,  
  x_end = data$config$limits$end  
)
```

Arguments

data	starvz_data with trace data
step	size in milliseconds for the time aggregation step
x_start	X-axis start value
x_end	X-axis end value

Value

A ggplot object

Examples

```
## Not run:  
panel_utiltreenode(data=starvz_sample_lu, step=100)  
  
## End(Not run)
```

pre_handle_gantt *Pre-Computation for memory handles panel*

Description

Use for precomputation of memory handles panel

Usage

```
pre_handle_gantt(data, name_func = NULL)
```

Arguments

data	starvz_data with trace data
name_func	function to give names to handles

Value

Pre-Computed data for panel_handles

Examples

```
pre_handle_gantt(data = starvz_sample_lu)
```

qrmumps_colors *Colors for qr mumps*

Description

This will be deprecated

Usage

```
qrmumps_colors()
```

starvz_assemble	<i>Assemble multiple StarVZ panel lists</i>
-----------------	---

Description

Take a panel list, or a list of list of panels and assemble it

Usage

```
starvz_assemble(
  ...,
  config = NULL,
  remove_Y_info = TRUE,
  remove_legends = TRUE
)
```

Arguments

...	Panel list or list of panel lists
config	StarVZ configurations for determining panels heights
remove_Y_info	remove Y labels for a second and subsequent list of panels
remove_legends	remove legends for a second and subsequent list of panels

Value

The ggplot plot

Examples

```
starvz_assemble(starvz_plot_list(starvz_sample_lu), config=starvz_sample_lu$config)
```

starvz_check_data	<i>Check if all required data is available</i>
-------------------	--

Description

The following conditions are checked in order and return FALSE if any fail - If data is not NULL - If data is a StarVZ Class - If data has all tables (given by the names of the list tables) - If each respective table has all columns (given the associated vector) - Execute extra_func on data (that should return true or false)

Usage

```
starvz_check_data(data = NULL, tables = list(), extra_func = NULL)
```

Arguments

data	starvz_data with trace data
tables	A named list (names are tables of data) of vectors (elements are columns), if tables is null continue
extra_func	Extra function to be applied on data to do a last check

Value

Logical, TRUE if data pass all tests

Examples

```
starvz_check_data(starvz_sample_lu, tables = list("Comm_state" = c("Node")))
```

starvz_phase1	<i>Execute StarVZ Phase one.</i>
---------------	----------------------------------

Description

This function calls all CSV-converter inner-functions to pre-process them into StarVZ files. Although this can be directly used in a folder where all CSV compressed (gzip) files reside, we suggest to use the shell tool starvz or phase1-workflow.sh in the tools/ directory.

Usage

```
starvz_phase1(
  directory = ".",
  app_states_fun = lu_colors,
  state_filter = 0,
  whichApplication = "",
  input.parquet = "1"
)
```

Arguments

directory	Directory of CSV files
app_states_fun	Function to determine application
state_filter	Type of filter
whichApplication	Name of Application
input.parquet	Use or not of parquet files

Value

ggplot object with all starvz plots

Examples

```
example_folder <- system.file("extdata", "lu_trace", package = "starvz")
starvz_phase1(directory = example_folder)
```

starvz_plot

Make a StarVZ plot

Description

Create a StarVZ plot considering the data supplied

Usage

```
starvz_plot(  
  data = NULL,  
  name = NULL,  
  save = FALSE,  
  guided = data$config$guided$active  
)
```

Arguments

data	starvz_data class with \$config
name	Path for saved image
save	call ggplot to save the image
guided	compute ideal figure height

Value

ggplot object with all starvz plots

Examples

```
starvz_plot(starvz_sample_lu)
```

starvz_plot_list	<i>Generate the StarVZ Plots</i>
------------------	----------------------------------

Description

Use data to create the list of StarVZ plots

Usage

```
starvz_plot_list(data = NULL)
```

Arguments

data starvz_data with trace data

Value

A list of ggplot plots

Examples

```
starvz_plot_list(starvz_sample_lu)
```

starvz_read	<i>Read starvz trace files</i>
-------------	--------------------------------

Description

Read the directory of trace files (feather or parquet) and the configuration file, and return a starvz_data class used in starvz functions

Usage

```
starvz_read(directory = ".", config_file = NULL, selective = TRUE)
```

Arguments

directory Directory path of trace files
config_file Path for configuration yaml file
selective if True, only read data needed for creating panels activated in config

Value

The starvz_data with all tables

Examples

```
starvz_read("folder_to_parquet_files/")
starvz_read(directory = "folder_to_parquet_files/", config_file = "path_to_config.yaml")
starvz_read() # Read current directory
```

starvz_read_config *Read config files*

Description

Read starvz configuration yaml files. This function is design to replace an already existing configuration on starvz data.

Usage

```
starvz_read_config(file = NULL)
```

Arguments

file path to file

Value

A list containing starvz configuration

Examples

```
example_file <- system.file("extdata", "config.yaml", package = "starvz")
config <- starvz_read_config(example_file)
```

starvz_sample_lu *Small StarVZ data of LU Factorization*

Description

A small StarVZ data object obtained from Chameleon+StarPU LU Factorization Generated by:

```
library(starvz)
pre_phase1 <- starvz_phase1(system.file("extdata", "lu_trace", package = "starvz"), lu_colors, state_f
starvz_sample_lu <- starvz_read(system.file("extdata", "lu_trace", package = "starvz"), system.file("e
usethis::use_data(starvz_sample_lu)
```

Usage

```
starvz_sample_lu
```


Format

An object of class starvz_data of length 24.

Source

starvz_phase1 and starvz_read

starvz_set_log *Active internal debug logs*

Description

Active internal debug logs

Usage

starvz_set_log(state)

Arguments

state Active or not logs

Value

Nothing

Examples

starvz_set_log(FALSE)

Index

- * **datasets**
 - starvz_sample_lu, 40
- * **phase1 functions**
 - starvz_phase1, 37
- cholesky_colors, 3
- data_name_coordinates, 4
- data_name_handle, 4
- data_name_tag, 5
- handles_presence_states, 5
- lu_colors, 6
- panel_activenodes, 6
- panel_atree, 7
- panel_atree_structure, 8
- panel_gflops, 9
- panel_gpubandwidth, 10
- panel_handles, 11
- panel_hete_imbalance, 11
- panel_imbalance, 12
- panel_kiteration, 13
- panel_lackready, 14
- panel_memory_snap, 15
- panel_memory_state, 16
- panel_mpibandwidth, 17
- panel_mpiconcurrent, 18
- panel_mpiconcurrentout, 19
- panel_mpistate, 20
- panel_node_summary, 21
- panel_nodememuse, 21
- panel_pmttool_kiteration, 22
- panel_pmttool_st, 23
- panel_power_imbalance, 24
- panel_ready, 25
- panel_st, 26
- panel_st_agg_dynamic, 26
- panel_st_agg_static, 27
- panel_st_raw, 28
- panel_st_runtime, 29
- panel_submitted, 30
- panel_title, 31
- panel_usedmemory, 31
- panel_utilheatmap, 32
- panel_utiltreedepth, 33
- panel_utiltreenode, 34
- pre_handle_gantt, 35
- qrmumps_colors, 35
- starvz_assemble, 36
- starvz_check_data, 36
- starvz_phase1, 37
- starvz_plot, 38
- starvz_plot_list, 39
- starvz_read, 39
- starvz_read_config, 40
- starvz_sample_lu, 40
- starvz_set_log, 41