TABLE OF CONTENTS

MaDaTS API

The MaDaTS API provides several objects and functions to help users manage their workflows and data on multi-tiered storage hierarchy. The API primarily uses three types of objects: Task, VirtualDataObject and VirtualDataSpace.

Task

The Task defines an object that can be executed by MaDaTS.

Task()

Creates a task for MaDaTS.

SYNOPSIS

```
Task(command, type)
```

PARAMETERS

command   (string) an executable command

type   (TaskType) type of the task. A task can either be a COMPUTE or DATA. (default: COMPUTE)

RETURN VALUES

Task   A task object that can be executed by MaDaTS

Attributes

A task object has several attributes that define the execution semantics of the executable command.

- **params** : (type: list) list of parameters to the executable command. The parameters can be any basic data type (string, char, int, float) or a virtual data object.
- **scheduler** : (type: Scheduler.<type>) the type of scheduler that manages the resources and workflow tasks. The default scheduler is *Scheduler.None*, which means the tasks are executed on the local machine. *Scheduler.SLURM* and *Scheduler.PBS* set the respective batch schedulers on HPC systems.
- **scheduler_opts** : (type: dictionary) the different options for the selected scheduler. The options specify the batch job configuration for the scheduler like number of nodes, walltime, queue etc.
- **prerun** : (type: list) list of pre-processing commands that are executed prior to executing the executable task.
- **postrun** : (type: list) list of post-processing commands that are executed after the task is executed.

VirtualDataObject

VirtualDataObect is an abstract data object that represents a directory or a file on the filesystem.

VirtualDataObject()

Creates a virtual data object.

SYNOPSIS

```
VirtualDataObject(datapath)
```

PARAMETERS

datapath   (string) Path to a data directory or file

RETURN VALUES

VirtualDataObject   A virtual data object that represents data directory or file

add_producer()

Adds a task to the virtual data object signifying that the task is one of the producers of the data object.

SYNOPSIS

```
add_producer(task)
```

PARAMETERS

task   (Task) a task that generates the virtual data object

RETURN VALUES

None

add_consumer()

Adds a task to the virtual data object signifying that the task is one of the consumers of the data object.

SYNOPSIS

```
add_consumer(task)
```

PARAMETERS

task   (Task) a task that uses the virtual data object

RETURN VALUES

None

Attributes

A virtual data object also allows users to define certain properties about the data.

- **size** : (type: int) total size of the data in bytes.
- **persist** : (type: boolean) sets the persistence of a data object.
- **replication** : (type: int) defines the replication factor, i.e., the number of copies of the data.
- **non_movable** : (type: boolean) forces the virtual data object to be non-movable, i.e., allows the data to be static on one storage tier independent of the selected data management strategy.

VirtualDataSpace

The VirtualDataSpace object is a data space for workflows that allows users to manage data and workflow.

VirtualDataSpace()

Creates a virtual data space (VDS). The data management functions in MaDaTS simply manage a VDS.

SYNOPSIS

```
VirtualDataSpace()
```

PARAMETERS

None

RETURN VALUES

VirtualDataSpace   An empty virtual data space.

map()

Maps a datapath to a virtual data object on VDS.

SYNOPSIS

```
map(datapath)
```

PARAMETERS

datapath   (string) Path to a data directory or file

RETURN VALUES

VirtualDataObject   A virtual data object that represents the data directory or file

add()
Adds a virtual data object to the VDS.

SYNOPSIS
```
add(vdo)
```

PARAMETERS
vdo   (VirtualDataObject) A virtual data object that needs to be added to the VDS

RETURN VALUES
None

copy()
Copies a virtual data object into another virtual data object on VDS.

SYNOPSIS
```
copy(vdo, storage_id)
```

PARAMETERS
vdo   (VirtualDataObject) a virtual data object that needs to be copied.

storage_id   (string) Name of the storage tier to which the copied virtual data object belongs.

RETURN VALUES
VirtualDataObject   The copied virtual data object that is on the specified storage tier.

replace()
Replaces a virtual data object with another virtual data object on VDS.

SYNOPSIS
```
replace(old_vdo, new_vdo)
```

PARAMETERS
old_vdo   (VirtualDataObject) The virtual data object that would be replaced.

new_vdo   (VirtualDataObject) The virtual data object that replaces the other virtual data object.

delete()

Removes a virtual data object from VDS.

```
delete(vdo)
```

vdo   (VirtualDataObject) The virtual data object that would be deleted.

None

Attributes

A virtual data space allows users to select the data management strategies based on which the workflow and data will be managed by MaDaTS.

- **data_management_policy** : (type: Policy) sets up the policy based on which the data will be managed on multiple storage tiers. By default, MaDaTS does not set any default data management policy as defined by *Policy.NONE*. MaDaTS also has two policies that the users can select from. *Policy.WORKFLOW_AWARE* uses the structure of the workflow to optimize data management, and *Policy.STORAGE_AWARE* uses the properties of the underlying storage tiers to optimize the data management.
- **auto_cleanup** : (type: boolean) enables auto cleaning in MaDaTS for optimizing storage space. If this flag is set, MaDaTS removes copies of the data from the storage tiers if they are no more used by any subsequent tasks.

Data and Workflow Management

MaDaTS provides three interfaces to manage data and workflows on multi-tiered storage systems.

map()

Maps a workflow on to VDS. This transforms a workflow specification defined in terms of tasks and inputs/outputs to a VDS consisting of virtual data objects and tasks as their producers and consumers.

```
map(workflow, lang, policy)
```

workflow	(string/Object) Name of a YAML workflow specification file or a dictionary object describing the workflow. The workflow description must follow the specification supported by MaDaTS (described later).
lang	(string) Describes the language of the workflow description. If the description file is written in Yaml, lang='yaml', and if it's a dictionary object then lang='DictObj'.
policy	(Policy) data management policy as defined by MaDaTS.

RETURN VALUES

VirtualDataSpace	A virtual data space containing virtual data objects

manage()

Manages a VDS. Creates separate tasks to manage data and execute workflow based on the data management policies.

SYNOPSIS

```
manage(vds, execute_mode)
```

PARAMETERS

vds	(VirtualDataSpace) A virtual data space consisting of virtual data objects and associated tasks.
execute_mode	(ExecutionMode) Specifies how the MaDaTS manages the execution of tasks. *ExecutionMode.DAG* executes the workflow with data and compute tasks as a graph, and manages the dependencies accordingly. Each task of the workflow is submitted as a single batch job. *ExecutionMode.BIN* combines several tasks into one job and submits the job to the batch scheduler.

RETURN VALUES

None

query()

Queries a VDS. Allows users to retrieve information about the VDS.

SYNOPSIS

```
query(vds, query)
```

vds   (VirtualDataSpace) A virtual data space consisting of virtual
                             data objects and associated tasks.
query   (list) Describes a query with comma-separated metrics.
                             The metrics are:
                               o *num_vdos*: number of virtual data objects in a VDS
                               o *data_tasks*: number of data tasks created for
                                 managing data
                               o *data_movements*: number of data movements
                                 between the storage tiers
                               o *preparer_tasks*: number of tasks preparing the data
                                 directories prior to moving data
                               o *cleanup_tasks*: number of cleanup tasks for
                                 removing unused data

RETURN VALUES

Dict Object   A dictionary containing the queried metrics and associated
                             values

validate()

Validates the completeness of a VDS. It checks if all the virtual data objects referenced
by the tasks are added to the VDS.

SYNOPSIS

```
validate(vds)
```

PARAMETERS

vds   (VirtualDataSpace) A virtual data space consisting of virtual
                             data objects and associated tasks.

RETURN VALUES

boolean   True, if all the virtual data objects associated with the
                             tasks are added to the VDS. Else, False.

MaDaTS Command Line

In addition to the API, a command-line tool is also provided to execute workflows and
manage data through MaDaTS. The command-line tool can be used as following:

SYNOPSIS

```
madats -w WORKFLOW [-l LANGUAGE] [-m {dag,bin}] [-p {none,wfa,sta}]
```

The options to the command-line tool are:

```
-w WORKFLOW, --workflow WORKFLOW
            workflow description file (default: None)
-l LANGUAGE, --language LANGUAGE
            workflow description language (default: yaml)
-m {dag,bin}, --mode {dag,bin}
            execution mode (default: dag)
-p {none,wfa,sta}, --policy {none,wfa,sta}
            data management policy (default: none)
```

DESCRIPTION

The command-line tool takes a workflow description and maps it to a VDS. The policies can be one of none/wfa/sta, where they imply *Policy.NONE*, *Policy.WORKFLOW_AWARE* and *Policy.STORAGE_AWARE* repectively.

MaDaTS Workflow Description

The workflow specification in MaDaTS requires users to describe the task and it's associated properties in YAML format.

The workflow specification is a dictionary consisting of workflow tasks and their associated properties. The keys in the dictionary are identifiers for the tasks. Each task has a set of properties associated with it. Mentioned below are the elements that are used to define the properties of a task.

1. **command**: describes the command to be executed as part of the workflow task.
            [Type = string]
2. **params**   : lists the set of parameters of the workflow task
            [Type = list]
3. **vin**   : input files to the workflow task to be mapped onto VDS. The input files must be parameters of the command.
            [Type = list]
4. **vout**   : output files of the workflow task to be mapped onto VDS. The output files must be parameters of the command.
            [Type = list]
5. **scheduler**: the name of the batch scheduler through which the tasks will be executed.
            scheduler: slurm, uses the Slurm scheduler to manage the task resources.
            scheduler: pbs, uses the PBS scheduler to manage the resources.
            [Type = string]

6. **scheduler_opts**: specifies the different scheduler options to manage the task. These options are defined in the scheduler's config file. Slurm and PBS has the configuration options defined in config/slurm.cfg and config/pbs.cfg respectively.
[Type = list of dictionaries]