



holoscan.schedulers

This module provides a Python API to underlying C++ API Schedulers.

holoscan.schedulers.EventBasedScheduler	Event-based multi-thread scheduler
holoscan.schedulers.GreedyScheduler	Greedy scheduler
holoscan.schedulers.MultiThreadScheduler	Multi-thread scheduler

class holoscan.schedulers.EventBasedScheduler

Bases: holoscan.gxf._gxf.GXFScheduler, holoscan.core._core.Component, holoscan.gxf._gxf.GXFComponent

Event-based multi-thread scheduler

Parameters

fragment

The fragment the condition will be associated with

clock

The clock used by the scheduler to define the flow of time. If None, a default-constructed *holoscan.resources.RealtimeClock* will be used.

worker_thread_number

The number of worker threads.

stop_on_deadlock

If enabled the scheduler will stop when all entities are in a waiting state, but no periodic entity exists to break the dead end. Should be disabled when scheduling conditions can be changed by external actors, for example by clearing queues manually.

max_duration_ms

The maximum duration for which the scheduler will execute (in ms). If not specified (or if a negative value is provided), the scheduler will run until all work is done. If periodic terms are present, this means the application will run indefinitely.

stop_on_deadlock_timeout

The scheduler will wait this amount of time before determining that it is in deadlock and should stop. It will reset if a job comes in during the wait. A negative value means not stop on deadlock. This parameter only applies when *stop_on_deadlock=true*,

name

The name of the scheduler.

Attributes

args	The list of arguments associated with the component.
------	--

description	YAML formatted string describing the component.
fragment	Fragment that the scheduler belongs to.
gxf_cid	The GXF component ID.
gxf_name	The name of the component.
gxf_context	The GXF context of the component.
gxf_entity_id	The GXF entity ID.
gxf_type_name	The GXF type name of the scheduler.
id	The identifier of the component.
name	The name of the scheduler.

clock	
max_duration_ms	
spec	
stop_on_deadlock	
stop_on_deadlock_timeout	
worker_thread_number	

Methods

<pre>add_ arg (*args, **kwa rgs)</pre>	Overloaded function.
<pre>gxf_i nitiali ze (self)</pre>	Initialize the component.
<pre>initial ize (self)</pre>	Initialize the scheduler.
<pre>setu p (self, a rg0)</pre>	setup method for the scheduler.

```
__init__(self: holoscan.schedulers.\_schedulers.EventBasedScheduler, fragment:
holoscan.core.\_core.Fragment, *, clock: holoscan.resources.\_resources.Clock = None,
worker_thread_number: int = 1, stop_on_deadlock: bool = True, max_duration_ms: int = -
1, stop_on_deadlock_timeout: int = 0, name: str = 'multithread_scheduler') None
```

Event-based multi-thread scheduler

Parameters

fragment

The fragment the condition will be associated with

clock

The clock used by the scheduler to define the flow of time. If None, a default-constructed *holoscan.resources.RealtimeClock* will be used.

worker_thread_number

The number of worker threads.

stop_on_deadlock

If enabled the scheduler will stop when all entities are in a waiting state, but no periodic entity exists to break the dead end. Should be disabled when scheduling conditions can be changed by external actors, for example by clearing queues manually.

max_duration_ms

The maximum duration for which the scheduler will execute (in ms). If not specified (or if a negative value is provided), the scheduler will run until all work is done. If periodic terms are present, this means the application will run indefinitely.

stop_on_deadlock_timeout

The scheduler will wait this amount of time before determining that it is in deadlock and should stop. It will reset if a job comes in during the wait. A negative value means not stop on deadlock. This parameter only applies when *stop_on_deadlock=true*,

name

The name of the scheduler.

`add_arg(*args, **kwargs)`

Overloaded function.

1. `add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None`

Add an argument to the component.

2. `add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None`

Add a list of arguments to the component.

property `args`

The list of arguments associated with the component.

Returns

arglist

property clock

property description

YAML formatted string describing the component.

property fragment

Fragment that the scheduler belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(self: [holoscan.gxf_gxf.GXFComponent](#)) None

Initialize the component.

property gxf_typename

The GXF type name of the scheduler.

Returns

str

The GXF type name of the scheduler

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

`initialize(self: holoscan.gxf._gxf.GXFScheduler)` None

Initialize the scheduler.

property max_duration_ms

property name

The name of the scheduler.

Returns

name

`setup(self: holoscan.core._core.Scheduler, arg0: holoscan.core._core.ComponentSpec)`
None

setup method for the scheduler.

property spec

property stop_on_deadlock

property stop_on_deadlock_timeout

property worker_thread_number

`class holoscan.schedulers.GreedyScheduler`

Bases: `holoscan.gxf._gxf.GXFScheduler`, `holoscan.core._core.Component`,
`holoscan.gxf._gxf.GXFComponent`

Greedy scheduler

Parameters

fragment

The fragment the condition will be associated with

clock

The clock used by the scheduler to define the flow of time. If None, a default-constructed `holoscan.resources.RealtimeClock` will be used.

stop_on_deadlock

If enabled the scheduler will stop when all entities are in a waiting state, but no periodic entity exists to break the dead end. Should be disabled when scheduling conditions can be changed by external actors, for example by clearing queues manually.

max_duration_ms

The maximum duration for which the scheduler will execute (in ms). If not specified (or if a negative value is provided), the scheduler will run until all work is done. If periodic terms are present, this means the application will run indefinitely.

check_recession_period_ms

The maximum duration for which the scheduler would wait (in ms) when all operators are not ready to run in the current iteration.

stop_on_deadlock_timeout

The scheduler will wait this amount of time before determining that it is in deadlock and should stop. It will reset if a job comes in during the wait. A

negative value means not stop on deadlock. This parameter only applies when *stop_on_deadlock=true*,

name

The name of the scheduler.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the component.
fragment	Fragment that the scheduler belongs to.
gxf_cid	The GXF component ID.
gxf_name	The name of the component.
gxf_context	The GXF context of the component.
gxf_entity_id	The GXF entity ID.
gxf_type_name	The GXF type name of the scheduler.
id	The identifier of the component.
name	The name of the scheduler.

check_recession_period_ms	
clock	

max_duration_ms	
spec	
stop_on_deadlock	
stop_on_deadlock_timeout	

Methods

add_arg (*args, **kwargs) rgs)	Overloaded function.
gxf_initialize (self)	Initialize the component.
initialize (self)	Initialize the scheduler.
setup (self, arg0)	setup method for the scheduler.

*`__init__(self: holoscan.schedulers._schedulers.GreedyScheduler, fragment: holoscan.core._core.Fragment, *, clock: holoscan.resources._resources.Clock = None, stop_on_deadlock: bool = True, max_duration_ms: int = - 1, check_recession_period_ms: float = 0.0, stop_on_deadlock_timeout: int = 0, name: str = 'greedy_scheduler') None`*

Greedy scheduler

Parameters

fragment

The fragment the condition will be associated with

clock

The clock used by the scheduler to define the flow of time. If None, a default-constructed *holoscan.resources.RealtimeClock* will be used.

stop_on_deadlock

If enabled the scheduler will stop when all entities are in a waiting state, but no periodic entity exists to break the dead end. Should be disabled when scheduling conditions can be changed by external actors, for example by clearing queues manually.

max_duration_ms

The maximum duration for which the scheduler will execute (in ms). If not specified (or if a negative value is provided), the scheduler will run until all work is done. If periodic terms are present, this means the application will run indefinitely.

check_recession_period_ms

The maximum duration for which the scheduler would wait (in ms) when all operators are not ready to run in the current iteration.

stop_on_deadlock_timeout

The scheduler will wait this amount of time before determining that it is in deadlock and should stop. It will reset if a job comes in during the wait. A negative value means not stop on deadlock. This parameter only applies when *stop_on_deadlock=true*,

name

The name of the scheduler.

`add_arg(*args, **kwargs)`

Overloaded function.

1. `add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None`

Add an argument to the component.

2. `add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None`

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property check_recession_period_ms

property clock

property description

YAML formatted string describing the component.

property fragment

Fragment that the scheduler belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

`gxf_initialize(self: holoscan.gxf._gxf.GXFComponent)` None

Initialize the component.

property `gxf_typename`

The GXF type name of the scheduler.

Returns

str

The GXF type name of the scheduler

property `id`

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (`holoscan.gxf.GXFExecutor`), the identifier is set to the GXF component ID.

Returns

id

`initialize(self: holoscan.gxf._gxf.GXFScheduler)` None

Initialize the scheduler.

property `max_duration_ms`

property `name`

The name of the scheduler.

Returns

name

setup(self: [holoscan.core._core.Scheduler](#), arg0: [holoscan.core._core.ComponentSpec](#))
None

setup method for the scheduler.

property spec

property stop_on_deadlock

property stop_on_deadlock_timeout

class holoscan.schedulers.MultiThreadScheduler

Bases: [holoscan.gxf._gxf.GXFScheduler](#), [holoscan.core._core.Component](#),
[holoscan.gxf._gxf.GXFComponent](#)

Multi-thread scheduler

Parameters

fragment

The fragment the condition will be associated with

clock

The clock used by the scheduler to define the flow of time. If None, a default-constructed *holoscan.resources.RealtimeClock* will be used.

worker_thread_number

The number of worker threads.

stop_on_deadlock

If enabled the scheduler will stop when all entities are in a waiting state, but no periodic entity exists to break the dead end. Should be disabled when scheduling conditions can be changed by external actors, for example by clearing queues manually.

check_recession_period_ms

The maximum duration for which the scheduler would wait (in ms) when an operator is not ready to run yet.

max_duration_ms

The maximum duration for which the scheduler will execute (in ms). If not specified (or if a negative value is provided), the scheduler will run until all work is done. If periodic terms are present, this means the application will run indefinitely.

stop_on_deadlock_timeout

The scheduler will wait this amount of time before determining that it is in deadlock and should stop. It will reset if a job comes in during the wait. A negative value means not stop on deadlock. This parameter only applies when *stop_on_deadlock=true*,

name

The name of the scheduler.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the component.
fragment	Fragment that the scheduler belongs to.
gxf_cid	The GXF component ID.
gxf_name	The name of the component.
gxf_context	The GXF context of the component.

gxf_e id	The GXF entity ID.
gxf_t ypen ame	The GXF type name of the scheduler.
id	The identifier of the component.
nam e	The name of the scheduler.

check_recession_period_ms	
clock	
max_duration_ms	
spec	
stop_on_deadlock	
stop_on_deadlock_timeout	
worker_thread_number	

Methods

add_ arg (*args, **kwa rgs)	Overloaded function.
gxf_i nitiali ze (self)	Initialize the component.
initial ize (self)	Initialize the scheduler.

```
setu  
p  
(self, a  
rg0)
```

setup method for the scheduler.

```
__init__(self: holoscan.schedulers.\_schedulers.MultiThreadScheduler, fragment:  
holoscan.core.\_core.Fragment, *, clock: holoscan.resources.\_resources.Clock = None,  
worker_thread_number: int = 1, stop_on_deadlock: bool = True,  
check_recession_period_ms: float = 5.0, max_duration_ms: int = - 1,  
stop_on_deadlock_timeout: int = 0, name: str = 'multithread_scheduler') None
```

Multi-thread scheduler

Parameters

fragment

The fragment the condition will be associated with

clock

The clock used by the scheduler to define the flow of time. If None, a default-constructed *holoscan.resources.RealtimeClock* will be used.

worker_thread_number

The number of worker threads.

stop_on_deadlock

If enabled the scheduler will stop when all entities are in a waiting state, but no periodic entity exists to break the dead end. Should be disabled when scheduling conditions can be changed by external actors, for example by clearing queues manually.

check_recession_period_ms

The maximum duration for which the scheduler would wait (in ms) when an operator is not ready to run yet.

max_duration_ms

The maximum duration for which the scheduler will execute (in ms). If not specified (or if a negative value is provided), the scheduler will run until all work is done. If periodic terms are present, this means the application will run indefinitely.

stop_on_deadlock_timeout

The scheduler will wait this amount of time before determining that it is in deadlock and should stop. It will reset if a job comes in during the wait. A negative value means not stop on deadlock. This parameter only applies when *stop_on_deadlock=true*,

name

The name of the scheduler.

`add_arg(*args, **kwargs)`

Overloaded function.

1. `add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None`

Add an argument to the component.

2. `add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None`

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property check_recession_period_ms

property clock

property description

YAML formatted string describing the component.

property fragment

Fragment that the scheduler belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(self: [holoscan.gxf.gxf.GXFComponent](#)) None

Initialize the component.

property gxf_typename

The GXF type name of the scheduler.

Returns

str

The GXF type name of the scheduler

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

`initialize(self: holoscan.gxf.gxf.GXFScheduler)` None

Initialize the scheduler.

property `max_duration_ms`

property `name`

The name of the scheduler.

Returns

name

`setup(self: holoscan.core_core.Scheduler, arg0: holoscan.core_core.ComponentSpec)`
None

setup method for the scheduler.

property `spec`

property `stop_on_deadlock`

property `stop_on_deadlock_timeout`

property `worker_thread_number`

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024