



Class MessageLabel

Table of contents

Class Documentation

- Defined in [File `messagelabel.hpp`](#)

Class Documentation

class MessageLabel

Class to define a [MessageLabel](#) that is attached to every GXF Entity being communicated as a message between Operators in Holoscan, for Data Frame Flow Tracking.

A [MessageLabel](#) has a vector of paths, where each path is a vector of [Operator](#) references and their publish and receive timestamps.

Public Types

```
using TimestampedPath = std::vector<OperatorTimestampLabel>
```

```
using PathOperators = std::unordered_set<std::string>
```

Public Functions

```
inline MessageLabel()
```

```
inline MessageLabel(const MessageLabel &m)
```

```
inline MessageLabel &operator=(const MessageLabel &m)
```

```
inline int num_paths()
```

Get the number of paths in a [MessageLabel](#).

Returns

The number of paths in a [MessageLabel](#).

```
std::vector<std::string> get_all_path_names()
```

Get all the names of the path in formatted string, which is comma-separated values of the [Operator](#) names.

Returns

`std::vector<std::string>` The vector of strings, where each string is a path name.

```
inline std::vector<TimestampedPath> paths()
```

```
int64_t get_e2e_latency(int index)
```

Get the current end-to-end latency of a path in microseconds.

MessageLabel and the whole Data Flow Tracking component tracks the latency in microseconds, which is sufficient for our use-cases.

Parameters

index – the index of the path for which to get the latency

Returns

`int64_t` The current end-to-end latency of the index path in microseconds

```
inline double get_e2e_latency_ms(int index)
```

Get the current end-to-end latency of a path in milliseconds.

Parameters

index – The index of the path for which to get the latency.

Returns

`double` The current end-to-end latency of the index path in milliseconds.

```
TimestampedPath get_path(int index)
```

Get the Timestamped path at the given index.

Parameters

index – The index of the path to get

Returns

TimestampedPath& The timestamped path at the given index

```
std::string get_path_name(int index)
```

Get the path name string which is comma-separated values of the operator names.

Parameters

index – The index of the path to get

Returns

The path name string

```
OperatorTimestampLabel &get_operator(int path_index, int op_index)
```

Get the OperatorTimestampLabel at the given path and operator index.

Parameters

- **path_index** – The path index of the OperatorTimestampLabel to get
- **op_index** – The Operator index of the OperatorTimestampLabel to get

Returns

OperatorTimestampLabel& The Operator reference at the given path and operator index

```
void set_operator_pub_timestamp(int path_index, int op_index, int64_t pub_timestamp)
```

Set an Operator's pub_timestamp.

Parameters

- **path_index** – the path index of the Operator
- **op_index** – the index of the Operator in the path
- **pub_timestamp** – the new pub_timestamp to set

```
void set_operator_rec_timestamp(int path_index, int op_index, int64_t
rec_timestamp)
```

Set an Operator's pub_timestamp.

Parameters

- **path_index** – the path index of the Operator
- **op_index** – the index of the Operator in the path
- **rec_timestamp** – the new rec_timestamp to set

```
std::vector<int> has_operator(std::string op_name)
```

Check if an operator is present in the MessageLabel. Returns an empty vector if the operator is not present in any path.

Parameters

op_name – The name of the operator to check

Returns

List of path indexes where the operator is present

```
void add_new_op_timestamp(holoscan::OperatorTimestampLabel o_timestamp)
```

Add a new Operator timestamp to all the paths in a message label.

Parameters

o_timestamp – The new operator timestamp to be added

```
void update_last_op_publish()
```

Update the publish timestamp of the last operator in all the paths in a message label.

```
void add_new_path(TimestampedPath path)
```

Add a new path to the MessageLabel.

Parameters

path – The path to be added.

std::string to_string() const

Convert the MessageLabel to a string.

Returns

std::string The formatted string representing the MessageLabel with all the paths and the Operators with their publish and receive timestamps.

void print_all()

Print the to_string() in the standard output with a heading for the MessageLabel.

Public Static Functions

static inline double get_path_e2e_latency_ms(TimestampedPath path)

Get the end-to-end latency of a TimestampedPath in microseconds. It is a utility function in the class and not dependent on the object of this class. Therefore, it's a static function.

Parameters

path – The path for which to get the latency

Returns

The end-to-end latency of the path in ms

static std::string to_string(TimestampedPath path)

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