

tf.profiler.ProfileOptionBuilder

Contents

Class ProfileOptionBuilder

Methods

`__init__`

`account_displayed_op_only`

Class ProfileOptionBuilder

Defined in [tensorflow/python/profiler/option_builder.py](#).

Option Builder for Profiling API.

For tutorial on the options, see

<https://github.com/tensorflow/tensorflow/tree/master/tensorflow/core/profiler/g3doc/options.md>

```
# Users can use pre-built options:
opts = (
    tf.profiler.ProfileOptionBuilder.trainable_variables_parameter())

# Or, build your own options:
opts = (tf.profiler.ProfileOptionBuilder()
        .with_max_depth(10)
        .with_min_micros(1000)
        .select(['accelerator_micros'])
        .with_stdout_output()
        .build())

# Or customize the pre-built options:
opts = (tf.profiler.ProfileOptionBuilder(
    tf.profiler.ProfileOptionBuilder.time_and_memory())
        .with_displaying_options(show_name_regexes=['.*rnn.*'])
        .build())

# Finally, profiling with the options:
_ = tf.profiler.profile(tf.get_default_graph(),
                       run_meta=run_meta,
                       cmd='scope',
                       options=opts)
```

Methods

`__init__`

```
__init__(options=None)
```

Constructor.

Args:

- `options`: Optional initial option dict to start with.

account_displayed_op_only

```
account_displayed_op_only(is_true)
```

Whether only account the statistics of displayed profiler nodes.

Args:

- `is_true`: If true, only account statistics of nodes eventually displayed by the outputs. Otherwise, a node's statistics are accounted by its parents as long as it's types match 'account_type_regexes', even if it is hidden from the output, say, by `hide_name_regexes`.

Returns:

self

build

```
build()
```

Build a profiling option.

Returns:

A dict of profiling options.

float_operation

```
@staticmethod  
float_operation()
```

Options used to profile float operations.

Please see

https://github.com/tensorflow/tensorflow/tree/master/tensorflow/core/profiler/g3doc/profile_model_architecture.md on the caveats of calculating float operations.

Returns:

A dict of profiling options.

order_by

```
order_by(attribute)
```

Order the displayed profiler nodes based on a attribute.

Supported attribute includes micros, bytes, occurrence, params, etc.

<https://github.com/tensorflow/tensorflow/tree/master/tensorflow/core/profiler/g3doc/options.md>

Args:

- `attribute` : An attribute the profiler node has.

Returns:

self

select

```
select(attributes)
```

Select the attributes to display.

See <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/core/profiler/g3doc/options.md> for supported attributes.

Args:

- `attributes` : A list of attribute the profiler node has.

Returns:

self

time_and_memory

```
@staticmethod
time_and_memory(
    min_micros=1,
    min_bytes=1,
    min_accelerator_micros=0,
    min_cpu_micros=0,
    min_peak_bytes=0,
    min_residual_bytes=0,
    min_output_bytes=0
)
```

Show operation time and memory consumptions.

Args:

- `min_micros` : Only show profiler nodes with execution time no less than this. It sums accelerator and cpu times.
- `min_bytes` : Only show profiler nodes requested to allocate no less bytes than this.
- `min_accelerator_micros` : Only show profiler nodes spend no less than this time on accelerator (e.g. GPU).
- `min_cpu_micros` : Only show profiler nodes spend no less than this time on cpu.
- `min_peak_bytes` : Only show profiler nodes using no less than this bytes at peak (high watermark). For profiler nodes consist of multiple graph nodes, it sums the graph nodes' peak_bytes.
- `min_residual_bytes` : Only show profiler nodes have no less than this bytes not being de-allocated after Compute() ends. For profiler nodes consist of multiple graph nodes, it sums the graph nodes' residual_bytes.

- `min_output_bytes` : Only show profiler nodes have no less than this bytes output. The output are not necessarily allocated by this profiler nodes.

Returns:

A dict of profiling options.

trainable_variables_parameter

```
@staticmethod
trainable_variables_parameter()
```

Options used to profile trainable variable parameters.

Normally used together with 'scope' view.

Returns:

A dict of profiling options.

with_accounted_types

```
with_accounted_types(account_type_regexes)
```

Selectively counting statistics based on node types.

Here, 'types' means the profiler nodes' properties. Profiler by default consider device name (e.g. /job:xx/.../device:GPU:0) and operation type (e.g. MatMul) as profiler nodes' properties. User can also associate customized 'types' to profiler nodes through OpLogProto proto.

For example, user can select profiler nodes placed on gpu:0 with: `account_type_regexes=['.*gpu:0.*']`

If none of a node's properties match the specified regexes, the node is not displayed nor accounted.

Args:

- `account_type_regexes` : A list of regexes specifying the types.

Returns:

self.

with_empty_output

```
with_empty_output()
```

Do not generate side-effect outputs.

with_file_output

```
with_file_output(outfile)
```

Print the result to a file.

with_max_depth

```
with_max_depth(max_depth)
```

Set the maximum depth of display.

The depth depends on profiling view. For 'scope' view, it's the depth of name scope hierarchy (tree), for 'op' view, it's the number of operation types (list), etc.

Args:

- `max_depth` : Maximum depth of the data structure to display.

Returns:

self

with_min_execution_time

```
with_min_execution_time(  
    min_micros=0,  
    min_accelerator_micros=0,  
    min_cpu_micros=0  
)
```

Only show profiler nodes consuming no less than 'min_micros'.

Args:

- `min_micros` : Only show profiler nodes with execution time no less than this. It sums accelerator and cpu times.
- `min_accelerator_micros` : Only show profiler nodes spend no less than this time on accelerator (e.g. GPU).
- `min_cpu_micros` : Only show profiler nodes spend no less than this time on cpu.

Returns:

self

with_min_float_operations

```
with_min_float_operations(min_float_ops)
```

Only show profiler nodes consuming no less than 'min_float_ops'.

Please see

https://github.com/tensorflow/tensorflow/tree/master/tensorflow/core/profilerg3doc/profile_model_architecture.md on the caveats of calculating float operations.

Args:

- `min_float_ops` : Only show profiler nodes with float operations no less than this.

Returns:

self

with_min_memory

```
with_min_memory(  
    min_bytes=0,  
    min_peak_bytes=0,  
    min_residual_bytes=0,  
    min_output_bytes=0  
)
```

Only show profiler nodes consuming no less than 'min_bytes'.

Args:

- `min_bytes` : Only show profiler nodes requested to allocate no less bytes than this.
- `min_peak_bytes` : Only show profiler nodes using no less than this bytes at peak (high watermark). For profiler nodes consist of multiple graph nodes, it sums the graph nodes' `peak_bytes`.
- `min_residual_bytes` : Only show profiler nodes have no less than this bytes not being de-allocated after `Compute()` ends. For profiler nodes consist of multiple graph nodes, it sums the graph nodes' `residual_bytes`.
- `min_output_bytes` : Only show profiler nodes have no less than this bytes output. The output are not necessarily allocated by this profiler nodes.

Returns:

self

with_min_occurrence

```
with_min_occurrence(min_occurrence)
```

Only show profiler nodes including no less than 'min_occurrence' graph nodes.

A "node" means a profiler output node, which can be a python line (code view), an operation type (op view), or a graph node (graph/scope view). A python line includes all graph nodes created by that line, while an operation type includes all graph nodes of that type.

Args:

- `min_occurrence` : Only show nodes including no less than this.

Returns:

self

with_min_parameters

```
with_min_parameters(min_params)
```

Only show profiler nodes holding no less than 'min_params' parameters.

'Parameters' normally refers the weights of in TensorFlow variables. It reflects the 'capacity' of models.

Args:

- `min_params` : Only show profiler nodes holding number parameters no less than this.

Returns:

self

with_node_names

```
with_node_names(  
    start_name_regexes=None,  
    show_name_regexes=None,  
    hide_name_regexes=None,  
    trim_name_regexes=None  
)
```

Regular expressions used to select profiler nodes to display.

After 'with_accounted_types' is evaluated, 'with_node_names' are evaluated as follows:

For a profile data structure, profiler first finds the profiler nodes matching 'start_name_regexes', and starts displaying profiler nodes from there. Then, if a node matches 'show_name_regexes' and doesn't match 'hide_name_regexes', it's displayed. If a node matches 'trim_name_regexes', profiler stops further searching that branch.

Args:

- `start_name_regexes` : list of node name regexes to start displaying.
- `show_name_regexes` : list of node names regexes to display.
- `hide_name_regexes` : list of node_names regexes that should be hidden.
- `trim_name_regexes` : list of node name regexes from where to stop.

Returns:

self

with_pprof_output

```
with_pprof_output(pprof_file)
```

Generate a pprof profile gzip file.

To use the pprof file: `pprof -png --nodecount=100 --sample_index=1`

Args:

- `pprof_file` : filename for output, usually suffixed with .pb.gz.

Returns:

self.

with_stdout_output

```
with_stdout_output()
```

Print the result to stdout.

with_step

```
with_step(step)
```

Which profile step to use for profiling.

The 'step' here refers to the step defined by `Profiler.add_step()` API.

Args:

- `step` : When multiple steps of profiles are available, select which step's profile to use. If -1, use average of all available steps.

Returns:

self

with_timeline_output

```
with_timeline_output(timeline_file)
```

Generate a timeline json file.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

Support

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

