

Stackdriver Profiler lets you visually compare two profiles of the same type, taken from the same service within a project. You can compare profiles that differ by:

- **Ending times:** Does the service run faster at certain times? Has there been a gradual improvement or decline over time?
- **Zones:** Does the service run faster in some zones than in others?
- **Service versions:** Do newer versions differ markedly from older ones? How?
- **Weight:** How do profiles captured during high consumption periods differ from average profiles?

When you set up a comparison, you specify the parameters for an original profile and a compared profile. These two profiles share all configuration parameters except the one defined by the **Compare to** field.

To configure a comparison, do the following:

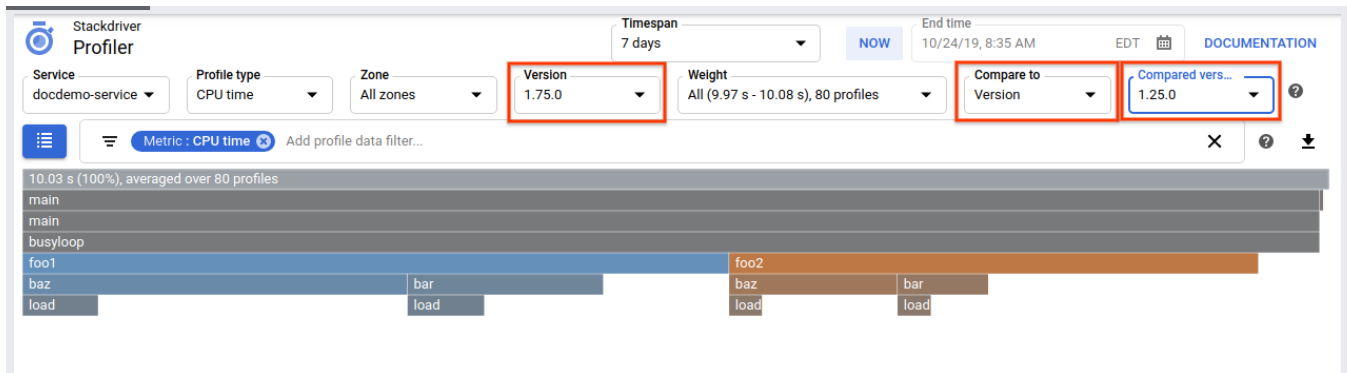
1. Set the comparison type by selecting an option from the **Compare to** menu:

- **End date/time** (/profiler/docs/selecting-profiles#time-config)
- **Zone** (/profiler/docs/selecting-profiles#zone-config)
- **Version** (/profiler/docs/selecting-profiles#version-config)
- **Weight** (/profiler/docs/selecting-profiles#weight-filter)

Selecting one of these options results in a new configuration field. The field name begins with **Compared** and ends with the comparison type. For example, if you select **Version**, a **Compared version** field is displayed next to the **Compare to** field.

2. Specify the compared profile by selecting an option from the **Compared** menu.

For example, the screenshot illustrates the settings to compare two versions of `docdemo-service`:



The settings for the following fields apply to the original profile and the compared profile:

- **Timespan:** 7 days
- **End time:** 10/24/19, 8:35 AM
- **Service:** docdemo-service
- **Profile type:** CPU time
- **Zone:** All zones
- **Weight:** All

For the original profile, the **Version** is 1.75.0. For the compared profile, the **Version** is 1.25.0.

The graph that results from a comparison differs from the standard graph in the following ways:

- Meaning of colors
- Meaning of function-block size
- Metric information displayed by the tooltip
- Contents of the focus list

In a standard flame graph, you can specify the color mode of the graph. For more details, go to [Color mode \(/profiler/docs/filtering-profiles#color-filter\)](/profiler/docs/filtering-profiles#color-filter).

In a comparison flame graph, the colors represent the difference between the total metric consumption of the function in the original profile and in the compared profile:

- Gray: There is little or no difference.
- Red: The difference is positive.
- Blue: The difference is negative.

The larger the difference in consumption values between the two profiles being compared, the more saturated the color.

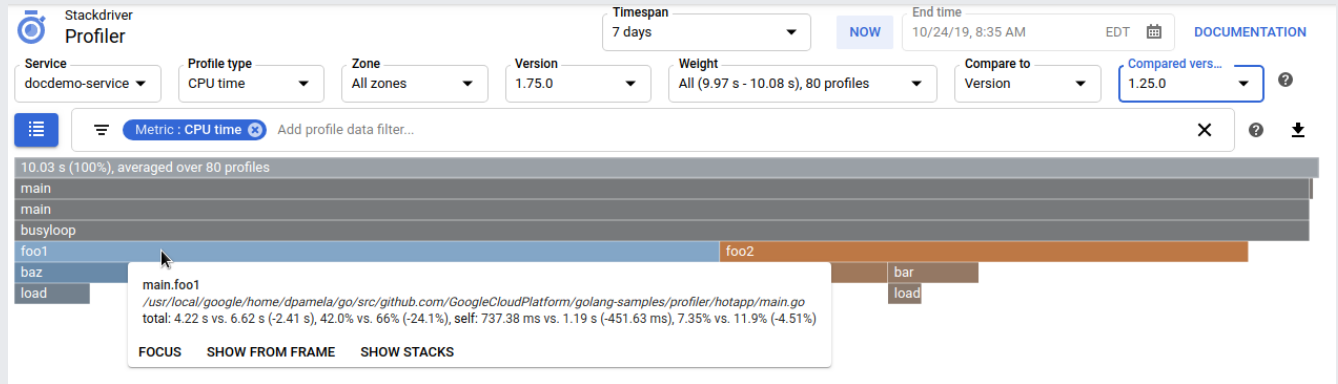
For example, in the previous figure, the `foo1` call stack is blue, indicating that the original profile consumed *less* CPU than the compared profile. The frame `foo1` is a brighter blue than its child frame `bar`, indicating that there is a larger difference in metric consumption for the function `foo1` than for the function `bar`. Lastly, the `foo2` call stack is red, indicating that the original profile consumed *more* CPU than the compared profile.

In a standard flame graph, the size of the function blocks illustrates the relative consumption of the metric being analyzed.

In a comparison flame graph, the size of the function blocks illustrates the relative *average* consumption of the metric being analyzed. By averaging the values for the two profiles, a block that exists only in one profile still appears in the comparison graph, at half its pre-comparison size.

In a standard flame graph, the metrics displayed by the tooltip are composed of two clauses, total and self. In both clauses, the absolute value is displayed along with a percentage:

In a comparison flame graph, because there are two profiles being compared, the tooltip displays the metric values for both profiles:



As shown in the figure, the total, and self clauses have the following form:

In this expression, `ABS_` means the absolute value and `REL_` means the relative percentage. Therefore, `[ABS_ORIGINAL]` is the absolute value for the original profile and `[ABS_COMPARED]` is the absolute value for the compared profile. The field `[ABS_DIFF]`, defined as `[ABS_ORIGINAL] - [ABS_COMPARED]`, is the absolute difference between the two profiles. When the absolute difference is zero, **no change** is displayed.

The percentage is computed for each profile as the ratio of a function's raw value to root's raw value. For example, in the original profile, `foo1` for took 4.22 seconds and `root` took 10.03 seconds. Therefore `[REL_ORIGINAL]` is about 42.0%. A similar computation is performed for the compared profile. Lastly, the field `[REL_DIFF]`, defined as the difference of these percentages, is the relative difference between the two profiles. When this difference is zero, **no change** is displayed.

For the preceding example, the total clause shows that the function `foo1` in the original profile executes in 4.22 seconds. The absolute difference of -2.41 seconds indicates that `foo1`

executes about 2.41 seconds faster in the original profile than in the compared profile:

In a standard flame graph, each row in the [Focus list](#)

(/profiler/docs/focusing-profiles#by_using_the_focus_list) contains a function name, the self and total metric consumption values, and the count of call stacks into that function.

In the comparison flame graph, each row in the [Focus list](#)

(/profiler/docs/focusing-profiles#by_using_the_focus_list) contains a function name, the self and total metric consumption values, and the count of call stacks into that function *for the original profile*. For a comparison flame graph, the focus list also includes two columns labeled **Diff**, a button labeled **Relative Diff**, and a button labeled **Absolute Diff**. For information on the relative difference and absolute difference, go to [Displayed metric values \(#diff-values\)](#).

By default, the color of the comparison flame graph is determined by the absolute differences between the original and compared profiles. In contrast, by default, the focus list displays the relative differences between the two profiles being compared:

Select focus function							
				RELATIVE DIFF	ABSOLUTE DIFF		?
Name	↓ Self	%	Diff	Total	%	Diff	Count
baz	3.22 s	32.2%	-3.88%	4.03 s	40.2%	-4.70%	2
foo2	2.74 s	27.3%	+13.2%	5.36 s	53%	+25.7%	1
load	1.6 s	15.9%	-1.85%	1.6 s	15.9%	-1.85%	4
bar	1.28 s	12.8%	-1.49%	2.08 s	20.7%	-2.51%	2
foo1	734.5 ms	7.33%	-4.05%	4.21 s	42.0%	-23.8%	1
busyloop	374.25 ms	3.73%	-2.04%	9.95 s	99%	-0.132%	1
procyield	27.13 ms	0.271%	+0.101%	27.13 ms	0.271%	+0.101%	4
futex	12.38 ms	0.123%	-0.007%	12.38 ms	0.123%	-0.007%	16

The table in the example shows that the function `foo1` takes about 4.21 seconds to run in the original profile. The time 4.21 seconds corresponds to about 42.0% of the runtime. The **Diff** column, which displays the relative difference, contains the value of **-23.8%**.

To view the absolute difference between metric values, click **Absolute Diff**.

Whether you have a standard flame graph or a comparison flame graph, you can sort the table rows in ascending \uparrow or descending \downarrow order by selecting a table header element.

To turn off comparison mode, set the **Compare to** field to **None**.

- To learn how to download your profile data, see [Downloading profiles](/profiler/docs/downloading-profiles) (/profiler/docs/downloading-profiles).
- For information on using the Profiler agent to collect profiling data for your services, see:
 - [Profiling Go applications](/profiler/docs/profiling-go) (/profiler/docs/profiling-go)
 - [Profiling Java applications](/profiler/docs/profiling-java) (/profiler/docs/profiling-java)
 - [Profiling Node.js applications](/profiler/docs/profiling-nodejs) (/profiler/docs/profiling-nodejs)
 - [Profiling Python applications](/profiler/docs/profiling-python) (/profiler/docs/profiling-python)
 - [Profiling applications running outside Google Cloud](/profiler/docs/profiling-external) (/profiler/docs/profiling-external)