

## Shared iterations

SUGGESTED EDITS

The `shared-iterations` executor shares iterations between the number of VUs. The test ends once k6 executes all iterations.

For a shortcut to this executor, use the `vus` and `iterations` options.

### NOTE

Iterations are not guaranteed to be evenly distributed with this executor. VU that executes faster will complete more iterations than slower VUs.

To guarantee that every VU completes a specific, fixed number of iterations, use the `per-VU iterations` executor.

Options

When to use

Example

Observations

## Options

Besides the [common configuration options](#), this executor has the following options:

OPTION	TYPE	DESCRIPTION	DEFAULT
<code>vus</code>	integer	Number of VUs to run concurrently.	1
<code>iterations</code>	integer	Total number of script iterations to execute across all VUs.	1
<code>maxDuration</code>	string	Maximum scenario duration before it's forcibly stopped (excluding <code>gracefulStop</code> ).	"10m"

## When to use

This executor is suitable when you want a specific number of VUs to complete a fixed number of total iterations, and the amount of iterations per VU is unimportant. If the time to complete a number of test iterations is your concern, this executor should perform best.

Ex

An example use case is for quick performance tests in the development build cycle. As developers make changes, they might run the test against the local code to test for performance regressions. Thus the executor works well with a *shift-left* policy, where emphasizes testing performance early in the development cycle, when the cost of a fix is lowest.

## Example

The following example schedules 200 total iterations shared by 10 VUs with a maximum test duration of 30 seconds.

`shared-iters.js`

```
1 import http from 'k6/http';
2 import { sleep } from 'k6';
3
4 export const options = {
5   discardResponseBodies: true,
6   scenarios: {
7     contacts: {
8       executor: 'shared-iterations',
9       vus: 10,
10      iterations: 200,
```

```

1   maxDuration: '30s',
12  },
13  },
14  });
15
16  export default function () {
17    http.get('https://test.k6.io/contacts.php');
18    // Injecting sleep
19    // Sleep time is 500ms. Total iteration time is sleep + time to finish
20    sleep(0.5);
21  }

```

## Observations

The following graph depicts the performance of the [example](#) script:



Based upon our test scenario inputs and results:

- Test is limited to a fixed number of 200 iterations of the `default` function;
- The number of VUs is fixed to 10, and are initialized before the test begins;
- Each *iteration* of the `default` function is expected to be roughly 515ms, or ~2/s;
- Maximum throughput (highest efficiency) is therefore expected to be ~20 items/s,  $= 200 \text{ iters} / 10 \text{ VUs}$   
 $\underline{2 \text{ iters/s} * 10 \text{ VUs;}} \quad = 20 \text{ iters VUs / s}$
- The maximum throughput is maintained for a larger portion of the test;
- The distribution of iterations may be skewed: one VU may have performed 50 iterations, another only 10.

[PREVIOUS](#)  
Executors

[NEXT](#)  
Per VU iterations



Grafana Labs



### PRODUCT

Open Source  
Grafana Cloud k6  
Grafana Cloud k6 Pricing  
Open Source vs Cloud  
Build vs Buy  
Testimonials

### RESOURCES

k6 Docs  
Grafana Cloud k6 Docs  
Extensions  
Integrations  
Modern Load Testing  
Not a developer. Why k6?

### COMMUNITY

Engage♥  
Forum  
Slack  
GitHub  
k6 Champions

### ABOUT

Blog  
Our story  
Our beliefs  
Contact  
Jobs🔗

### Subscribe to our newsletter!

Product developments and news from the k6 community.

SUBSCRIBE NOW