
eosc-perf

Christophe Laures, Moritz Behr, Jonas Blahut, Marc Falkenberg

Sep 01, 2022

CONTENTS:

- 1 Introduction** **1**
- 1.1 EOSC-Perf: A web service to compare computing resources 1
- 1.2 Main features 1
- 1.3 Legal 2
- 1.4 Contact us 2

- 2 Tutorials** **3**
- 2.1 Searching through results on the website 3
- 2.2 Adding a new result 5
- 2.3 Adding a new benchmark 8
- 2.4 How to use the API 9

- 3 Instructions for providers** **13**
- 3.1 Deploying the service 13

- 4 Indices and tables** **15**

INTRODUCTION

1.1 EOSC-Perf: A web service to compare computing resources

EOSC-Perf is a webapp made to host, search, compare and analyze benchmark results from many very diverse computing resources primarily available within the [European Open Science Cloud \(EOSC\)](#).

The main entry point is <https://performance.services.fedcloud.eu/>.

The service is also provided via the [EOSC Marketplace](#)

The landscape of European university servers is very diverse, so it is at times difficult to pick which servers or platform to use. EOSC-Perf is intended to help scientists identify the ideal server for a given task by letting them compare benchmark results that were ran on different servers to identify the server with the best performance for a given task.

The platform itself is intended to gather data from multiple users running benchmarks, so one does not need to re-run benchmarks if matching data already exists. Once data is available for a variety of platforms, the platforms can be easily compared using the existing data.

1.2 Main features

Benchmark results are associated with execution sites, a type of benchmark, an uploader, and optionally a set of user-created tags which may be used for annotation or search filtering.

Users can search through all results and filter results by sites, benchmarks, tags, uploaders, and even specific JSON data-points that are part of the result data. Users can then select multiple results and generate various types of comparison pages, such as speedup graphs.

If results are inaccurate, users may report them to administrators.

It supports submitting new benchmark types by running them in Docker images and submitting their names here. After this, you may submit new benchmark results associated to this kind of benchmark. The benchmark software has to generate a JSON output in order to be integrated in the service.

Users may authenticate themselves through [EGI AAI Check-In](#), which allows them to use single-sign-on with their university.

The service also provides a fully-featured [API](#) allowing to communicate with the service from external applications, e.g to add new results by automated scripts.

Please, check our [Tutorials](#) for more details on how to use the service.

1.3 Legal

By using our service you agree to our [Privacy Policy](#) and the [Terms of Service](#).

1.4 Contact us

In the case of questions, suggestions, or for any feedback, please [contact us](#).

TUTORIALS

Below are a few tutorials to demonstrate how to interact with the service.

2.1 Searching through results on the website

When you first load the website, you will be redirected to the [result search page](#). By default, it will display up the 20 most recently uploaded results. Not every data is useful to everyone, so the user is provided with some tools to filter out irrelevant data.

2.1.1 Filtering

The application offers filtering the results by various criteria, including the benchmark that was ran [1], the site it was ran on [2], and the configuration of virtual machine ('flavor') that was used [3]. Additionally, users may filter by any amount of data points, specifically specific fields from the results' JSON data. Users may refer [4] to a field in the json structure by chaining together the names of the keys with periods, and compare it against given values [6]. For example,

```
{ "machine": { "cpu": { "core_count": 4 } } }
```

would be referred to as 'machine.cpu.core_count'.

Detail

Benchmark: None

Site: None

Flavor: Select a site first!

The screenshot shows a filtering interface with the following elements and callouts:

- 1: Select button for Benchmark
- 2: Select button for Site
- 3: Select button for Flavor
- 4: Input field for JSON Key
- 5: Greater-than symbol (>) for comparison
- 6: Input field for Value
- 7: Close button (X) for the filter
- 8: + Add filter button
- 9: Apply Filters button

1. Select benchmark to filter by
2. Select site to filter by
3. If a site is selected, select a filter to filter by

JSON Filter:

4. Specify JSON-key to filter by (structured as path.to.value)
5. Set comparison mode
6. Specify value you want to compare against
7. Remove filter
8. Add a new filter
9. Update search results using given filtering options

2.1.2 Compare in table

Results matching the search criteria are displayed in a table. This table can be used to interact with specific results (selecting [2], viewing or reporting[5]), and can also be configured to display columns containing specific data fields [4], using the json structure notation explained in the Search overview.

Detail

<input type="checkbox"/> ①	Benchmark	Site ③	Site flavor	Tags	④
<input type="checkbox"/>	thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	<input type="button" value="View"/>
<input checked="" type="checkbox"/> ②	thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	<input type="button" value="View"/>
<input type="checkbox"/>	thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	<input type="button" value="View"/> ⑤
<input type="checkbox"/>	thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	<input type="button" value="View"/>
<input type="checkbox"/>	thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	<input type="button" value="View"/>
<input type="checkbox"/>	thechristophe/openbench-c-ray:latest	CESGA	cor1mem2h10	None	<input type="button" value="View"/>
<input type="checkbox"/>	thechristophe/openbench-c-ray:latest	CESNET-MCC	standard.small	None	<input type="button" value="View"/>
<input type="checkbox"/>	thechristophe/openbench-c-ray:latest	CESNET-MCC	standard.medium	None	<input type="button" value="View"/>
<input type="checkbox"/>	thechristophe/openbench-c-ray:latest	CESNET-MCC	standard.large	None	<input type="button" value="View"/>
<input type="checkbox"/>	thechristophe/openbench-c-ray:latest	CESNET-MCC	hpc.8core-16ram	None	<input type="button" value="View"/>

Results per page: ⑥

« < 1 2 3 > » ⑦

1. Select all results
2. Select one result
3. Click on a column to sort by it
4. Add or remove custom columns using json path syntax
5. View or report result
6. Configure the number of results to display
7. Navigate between pages of results
8. Export selected results' columns' as a .csv file

2.1.3 Compare on the plot

You may generate a plot using the values you have selected. The range of features may vary by which diagram engine you have selected, but the most common one, Apache ECharts, allows you to select one JSON value for each axis and configuring the plot scaling. The diagram will then display all selected results as a 2D graph.

Detail

Diagram Apache ECharts ①

Graph mode ②

X Scale ③ Y Scale ④

Regression All results must be from same site

X Axis: ⑤

Y Axis: ⑥

1. Select diagram engine of your choice
2. Select diagram type
3. Choose between linear and logarithmic scaling for the X axis
4. Choose between linear and logarithmic scaling for the Y axis
5. Enter json key of source data to display on X axis
6. Enter json key of source data to display on Y axis

2.2 Adding a new result

2.2.1 Requirements for results

1. You are able to authenticate via EGI Check-In
2. Results are in the JSON format
3. Results belong to one of the already registered benchmarks (if not, please, see [Adding a new benchmark](#))
4. The Site, Flavor, and time for the benchmark run is known.

2.2.2 Uploading a result via the website

To submit a new result, first navigate to the Result Submission page using the navigation bar:

Diagram

Search Submit Instructions Infrastructure Manager user@example.com

Results
Benchmarks
Sites

Please select a benchmark

Benchmark: None Select
 Site: None Select
 Flavor: Select a site first! Select

Export + Add filter Apply Filters

<input type="checkbox"/>	Benchmark	Site	Site flavor	Tags	
<input type="checkbox"/>	thecristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	View
<input type="checkbox"/>	thecristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	View
<input checked="" type="checkbox"/>	thecristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	View
<input type="checkbox"/>	thecristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	View
<input type="checkbox"/>	thecristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None	View
<input type="checkbox"/>	thecristophe/openbench-c-ray:latest	CESGA	cor1mem2h10	None	View
<input type="checkbox"/>	thecristophe/openbench-c-ray:latest	CESNET-MCC	standard.small	None	View

On the result submission page:

Upload Result

Please select result JSON file

Browse... No file selected. 1

Benchmark: None 2 Select

Site: None 3 Select

Flavor: Select a site first! 4 Select

Execution date: February 6, 2022 20:30 5

Search Keywords...

No tags selected

No tags available 6

tensor 7 Add Tag

8 Submit

[Terms of service](#) [Privacy policy](#) [Email Support](#)

1. Select the JSON containing the benchmark result data
2. Select the benchmark that was used to obtain this data
3. Select the site on which your virtual machine was hosted
4. Select the type of virtual machine you deployed
5. Select the date and time on which you ran the benchmark
6. (optional) Select any filtering tags
7. (optional) Add tags if none match your needs
8. Submit your result

Site Search

Enter your query here, keywords

Site

VirtualboxTestbed

CESGA

CESNET-MCC

IFCA-LCG2

IISAS-FedCloud

NCG-INGRID-PT

« < 1 > »

1. Search through names and descriptions using keywords, separated by spaces
2. Re-run search
3. Select entry
4. Navigate between pages
5. Deselect / select nothing
6. Submit a new entry

2.2.3 Uploading a result via the API

For a detailed example on how to use the API to submit a result, please have a look at the [Jupyter notebook](#).

2.3 Adding a new benchmark

2.3.1 Requirements for benchmark

1. The Benchmark is publicly available from the [Docker Hub](#)
2. It produces a [JSON](#) output file
3. (optional but recommended) [JSON schema](#) for the output file is available

2.3.2 Adding a benchmark via the website

To submit a new benchmark, first navigate to the Benchmark Submission Page:

The screenshot shows the Benchmark Submission Page. At the top, there is a navigation bar with 'Submit' selected, a search bar, and a user profile 'user@example.com'. A dropdown menu is open under 'Submit', with 'Benchmarks' highlighted by a red arrow. The main content area includes a 'Diagram' tab, a 'Please select a benchmark' message, and a table of benchmarks. The table has columns for Benchmark, Site, Site flavor, and Tags. Below the table are buttons for 'Export', '+ Add filter', and 'Apply Filters'.

Benchmark	Site	Site flavor	Tags
<input type="checkbox"/> thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None View
<input checked="" type="checkbox"/> thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None View
<input type="checkbox"/> thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None View
<input type="checkbox"/> thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None View
<input type="checkbox"/> thechristophe/openbench-c-ray:latest	VirtualboxTestbed	virtualbox-arch	None View
<input type="checkbox"/> thechristophe/openbench-c-ray:latest	CESGA	cor1mem2h10	None View
<input type="checkbox"/> thechristophe/openbench-c-ray:latest	CESNET-MCC	standard.small	None View

On the benchmark submission page:

The screenshot shows the 'Add Benchmark' form with the following elements:

- Header:** Search, Submit, Instructions, Infrastructure Manager, user@example.com
- Benchmark:** Input fields for 'user/image' (1) and 'latest' (2).
- Benchmark description (optional):** Text area with placeholder 'Enter a description of the new benchmark here.' (3).
- Benchmark result JSON schema (example here):** Text area containing a JSON schema snippet: `{ "$schema": "https://json-schema.org/draft/2020-12/schema",` (4).
- Submit:** Green button (5).
- Footer:** [Terms of service](#), [Privacy policy](#), [Email Support](#)

1. Enter the docker image name of your benchmark (for example: thechristophe/openbench-c-ray)
2. Enter the tag of your docker image you used
3. (optional) Add a description of your benchmark for other users.
4. Add the JSON schema the results of this benchmark must conform to
5. Submit your result

2.3.3 Adding a benchmark using the API

For a detailed example on how to use the API to submit a benchmark, please have a look at the [Jupyter notebook](#).

2.4 How to use the API

The EOSC Performance API uses REST. JSON is returned by all API responses including errors and HTTP response status codes are to designate success and failure.

All resources are documented following the OpenAPI Version 3 specification. Find the last version under performance.services.fedcloud.eu/api/v1/api-spec.json and access the API endpoint at performance.services.fedcloud.eu/api/v1.

2.4.1 Authentication and authorization

Requests to the EOSC Performance public API are for public and private information. Endpoints availability is marked with with tags on the method description:

- **Public:** Does not require authentication
- **Users:** Requires registration and authentication.
- **Admins:** Requires authentication and administration authorization.

Token

The **Authorization** HTTPS header should be specified using the format **Authorization: Bearer <your-access-token>** to authenticate as a user and have the same permissions as the user itself. You can obtain the access token using `oidc-agent`.

Registration

The usage of POST methods are restricted to authenticated and registered users. Registration in the service means the user accepts the [terms of service](#) exposed at EOSC Performance. You can accept those terms via GUI or by using the POST method `users:register` as an authenticated user using an access token.

```
$ curl -X 'POST' \  
  'https://perf.test.fedcloud.eu/api/v1/users:register' \  
  -H 'Authorization: Bearer <your-access-token>'
```

The registration request does not require additional data as body.

2.4.2 Requests

The API endpoint can only be accessed by **https**, all the **http** attempts would receive **301 Moved Permanently**. A graphical Swagger web interface is available at the root endpoint to provide user friendly documentation, interface and examples.

EOSC Performance API v1 OAS3

/api/v1/api-spec.json

Servers
/api/v1

Authorize

benchmarks

Operations on benchmarks

- GET** /benchmarks (Public) Filters and list benchmarks
- POST** /benchmarks (Users) Uploads a new benchmark
- GET** /benchmarks:search (Public) Filters and list benchmarks
- GET** /benchmarks/{id} (Public) Retrieves benchmark details
- PUT** /benchmarks/{id} (Admins) Implements JSON Put for benchmarks
- DELETE** /benchmarks/{id} (Admins) Deletes an existing benchmark
- POST** /benchmarks/{id}:approve (Admins) Approves a benchmark to include it on default list methods
- POST** /benchmarks/{id}:reject (Admins) Rejects a benchmark to safe delete it.

reports

Operations on reports

results

Operations on results

sites

Operations on sites

flavors

Operations on flavors

tags

Operations on tags

users

Operations on users

Schemas

Examples

Anonymous request example using bash:

```
$ curl -X 'GET' \  
  'https://performance.services.fedcloud.eu/api/v1/benchmarks'
```

Authenticated request example using bash:

```
$ curl -X 'POST' \  
  'https://performance.services.fedcloud.eu/api/v1/users:register' \  
  -H 'Authorization: Bearer <your-access-token>'
```


INSTRUCTIONS FOR PROVIDERS

3.1 Deploying the service

For instructions on how to deploy and run the application, please refer to [README.md](#) at the root of the source tree.

INDICES AND TABLES

- genindex
- modindex
- search