DataBench

Evidence Based Big Data Benchmarking to Improve Business Performance

D3.2 DataBench Toolbox - Alpha including support for reusing of existing benchmarks

Abstract

This is a supporting document of the demonstrator of the Alpha version of the DataBench Toolbox The Toolbox intends to offer a framework for big data benchmarking based on existing efforts in the community. The document provides an overview of the features developed for the Alpha version, including the generic framework to support existing benchmarks and the integration of several benchmarking tools into it. Moreover the document presents the advances, development and findings in the back-end of the Toolbox, but also the initial minimal development of the front-end, initially foreseen for the final release.

This document is the second deliverable related to the DataBench Toolbox after D3.1. It provides an update of the architecture presented in D3.1 in the light of the advances done in the Alpha version. More updates will be provided as part of the two upcoming releases of the Toolbox scheduled in the DataBench WP3 lifecycle.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780966

Deliverable D3.1	DataBench architecture
Work package	WP3
Task	3.2
Due date	30/06/2019
Submission date	XX/06/2018
Deliverable lead	ATOS
Version	1.0
Authors	ATOS (Tomás Pariente, Iván Martínez and Ricardo Ruiz) GUF (Todor Ivanov)
Reviewers	IDC (Philip Carnelley, David Wells), POLIMI (Chiara Francalanci)

Keywords

Benchmarking, big data, big data technologies, architecture, business performance, performance metrics, toolbox, use cases

Disclaimer

Copyright Notice

Table of Contents

Table of Figures

Table of Tables

Executive Summary

This document is a demonstrator of the first release of the DataBench Toolbox, also known as Alpha version. As such, this is an accompanying document that intends to shed some light on the current status of the Toolbox in terms of architecture, coverage of the functions and future work.

Therefore, the document provides an overview of the implementation of the architecture behind the Alpha version of the Toolbox, as well as the status of the implementation of its main components. There is a description of the main architectural choices and tools used for the implementation (Bootstrap and Play! for the Graphical User Interface and its access to the back-end; MySQL as initial Toolbox repository; and Ansible and AWX to automate the configuration, deployment and execution of the benchmarks).

The document also provides a walkthrough the functions developed by midway into the project (M18 – June 2019). To this extent the document provides screen-shots of the design mock-ups done in the first year of the project as well as of the current web implementation of the Dashboard associated to the Toolbox.

As a "demonstrator" deliverable, the document is the reflection of the software and demo which provides web access to the DataBench Toolbox Dashboard. The URL of the Alpha version is not available yet to the public, but can be accessed on request.

The Toolbox will have two more releases separated six months apart. Therefore, the document provides indications for the roadmap of the Toolbox in the coming year. It is worth mentioning that the Alpha version only covers aspects related to the registration, search, execution and retrieval of results of existing benchmarks. No work on business insights or integration with the knowledge graph that will include ways to derive non-technical aspects related to big data benchmarking has been attempted in this version, and will be the main subject of improvements in the releases to come.

1. Introduction

This document presents the DataBench Toolbox initial architecture. The DataBench Toolbox will include formal mechanisms to ease the process of reusing existing or new big data benchmarks into a common framework that will help stakeholders to find, select, download, execute and get a set of homogenized metrics. The DataBench Toolbox will be an integral part of the DataBench framework, which ultimately will deliver recommendations and business insights out of big data benchmarks.

The present document therefore starts by putting the DataBench Toolbox in context with the rest of the DataBench framework to later on dive into the details of the envisaged architecture. It is important to notice that the Toolbox will be based on existing efforts in big data benchmarking, rather than proposing new benchmarks. The DataBench Toolbox therefore aims to be an umbrella framework for big data benchmarking. The idea behind it is to provide ways to declare new benchmarks into the Toolbox and provide a set of automatisms and recommendations to allow the usage of these tools to become part of the ecosystem. Due to the different nature and technical scope of the existing tools, the degree of automation may vary from one tool to another. The baseline will be the possibility to download the selected benchmarking tools from the Toolbox web user interface, and to provide adapters to actually get the results of the benchmarking into the DataBench Toolbox in order to get homogenized technical metrics (i.e. throughput) that make comparable results from several benchmarks. Other tools may be subject to tighter integration and even automation of the deployment in the benchmarking system.

The document is structured as follow:

- Section 1 provides the introduction to the objectives of the deliverable.
- Section 2 is a summary of the essential features of the DataBench Toolbox Alpha version.
- Section 3 dives into the back-end processes developed in the Alpha version.
- Section 4 provides an overview of the front-end of the Alpha version of the Toolbox along with screenshots of the main parts of it.
- Finally, Section 5 provides the conclusions of the document as well as outlining the future work on the DataBench Toolbox.

2. DataBench Toolbox Alpha version essentials

The Alpha version of the Toolbox is a preliminary release providing limited functionality mostly focused on the initial steps of the addition, listing and execution of an initial set of technical big data benchmarks. As with any Alpha version, the main goal is to provide a preview of some of the main functions and processes that will be further developed and enhanced in successive releases. Therefore, the Alpha version covers only a limited set of the processes and functions envisaged for the DataBench Toolbox.

This section aims to explain briefly the main aspects covered in the Alpha version of the DataBench Toolbox.

2.1 Alpha version - Toolbox processes covered

Deliverable D3.1 [1] provided a detailed list of the processes that will take part in the life-cycle of the DataBench Toolbox. This section aims to briefly summarize the processes covered in the Alpha version as well as the degree of fulfilment of those processes, as shown in Table 1. The main processes described in D3.1 are: Accessing the Toolbox, Analytics and Metrics management, User Intentions, Setup and Runtime, Visualization and Reporting, and Benchmark Management. For more details in the processes, please refer to D3.1. section 3.

Process	Covered in Alpha version	Remaining functionality
Accessing	Define User Profile (partial): Creation of a user profile following the typology of the actors of the system: Admin, Technical User, Business User or Benchmark Provider. Sign in: Users are able to Create their accounts Permissions: The DataBench Administrator is able to Grant Permissions to technical users (admin acting as benchmark provider so far) Access to DataBench Toolbox: Users are able to access either anonymously (guests) or log in to the Toolbox.	Better management of access control for guest users Improve the GUI for sign-in, log-in and administrative management Improve and complete the definition for Business users and Benchmark providers, as well as the user grants by the administrator.
User Intentions	Not covered in the Alpha version (only initial search not goal-oriented)	All
Setup and Runtime	Configuration of benchmarks to be able to automate their deployment and execution from the web via Ansible recipes	Integrate the configuration of more benchmarks (so far only 4 have been integrated) for deployment and execution

Process	Covered in Alpha version	Remaining functionality
	Deployment: - Sandbox installation: Running benchmarks on AWX from the web – done and tested - In-house deployment and running of benchmarks "offline" in a separate infrastructure (done and tested, but no functionality available for users to download the Ansible Playbook, but for the Admin so far) Execution: - Initial injection of result files of the execution of selected benchmarks from the web to an Apache Kafka listener - Send results to Kafka from AWX - Results processing from Kafka to the internal DB	Implement functionality for Technical users to download the Ansible Playbook for "offline" execution of benchmarks Improve the configuration process Improve the result injection Implement software to homogenize technical metrics as much as possible
Analytics and metrics management		No functionality covered in the Alpha version related to analytics or homogenization of technical metrics or business insights
Visualization and Reporting	Inventory of benchmarks Metadata-based Google-like search box Filtered advanced search by an initial set of metadata	The visualization functionality is so far minimal, but sufficient for showcasing the Alpha version. Most improvements will be dealing with search functionality (advanced and guided search), visualization of technical metrics and visualization of business-related aspects and knowledge
Benchmark Management	Web Inventory management (Creation and Deletion of inventory) Creation of benchmarks and characterization via rich metadata Initial setting of technical metrics, recipes for deployment and execution of benchmarks catalogues	Improve management functions Develop technical metrics Logging functionality partial (more in collaboration with WP5)

Table 1 – Processes and functions covered in the Alpha version

It is worth mentioning that the Alpha and Beta versions of the Toolbox were not supposed to have an advanced Graphical User Interface (GUI). The web GUI was scheduled for the final version. However, in order to offer a clear user experience and be able to engage with the benchmarking community from earlier stages, the Alpha version already provides an initial web GUI offering support for all the processes listed above. We expect that this will help with the uptake of the Toolbox and bootstrap the engagement process.

From a conceptual perspective WP4 and WP3 came up with a visual representation of the main processes that is shown in Figure 1.

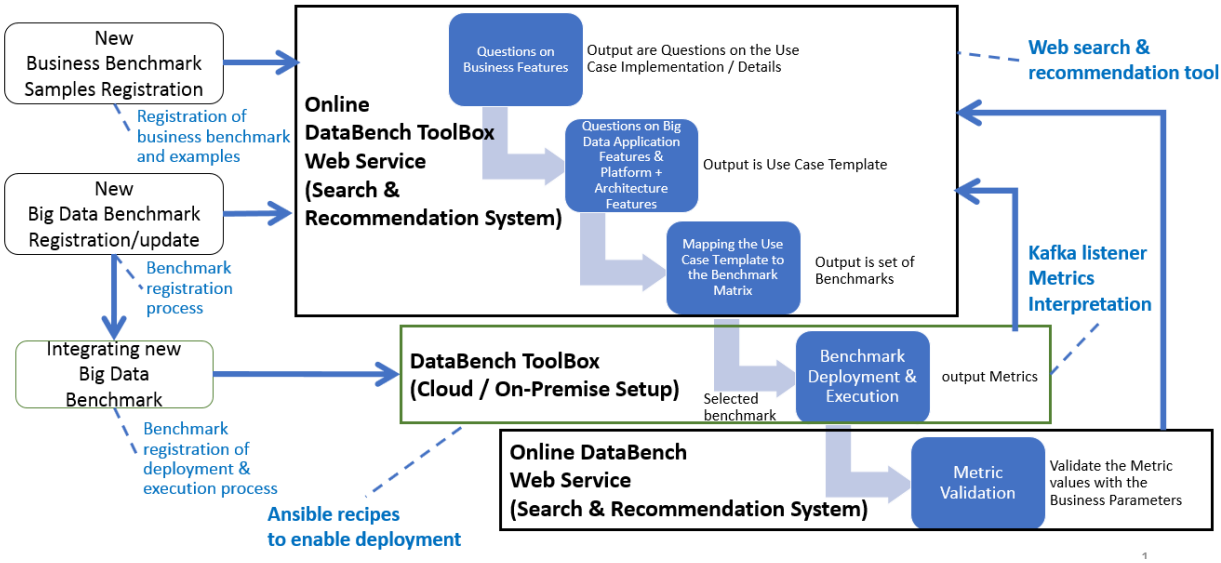

Figure 1. Conceptual overview of the Toolbox processes

The Alpha version covers mostly the following processes:

- Registration ("New Big Data Benchmark Registration/Update") and configuration ("Integrating new Big Data Benchmark") of benchmarks.
- The initial user interface and logic for the "Online DataBench Toolbox Web Service", which allows searching and selection of existing benchmarks registered in the tool.
- The execution of Ansible recipes to enable deployment and execution of benchmarks through the "DataBench Toolbox (Cloud / On-Premise Setup).
- The insertion of the technical results of the executions via a Kafka listener.

Therefore, the processes related to registering, searching and visualizing business insights are not part of the Alpha version.

2.2 Alpha version - Toolbox architecture update

Deliverable D3.1 provided a first version of the architecture of the DataBench Toolbox. This section provides an update of the architecture focusing on the elements delivered in the Alpha version of the architecture.

Figure 2 was shown in D3.1 and depicts the functional architecture of the DataBench Toolbox.

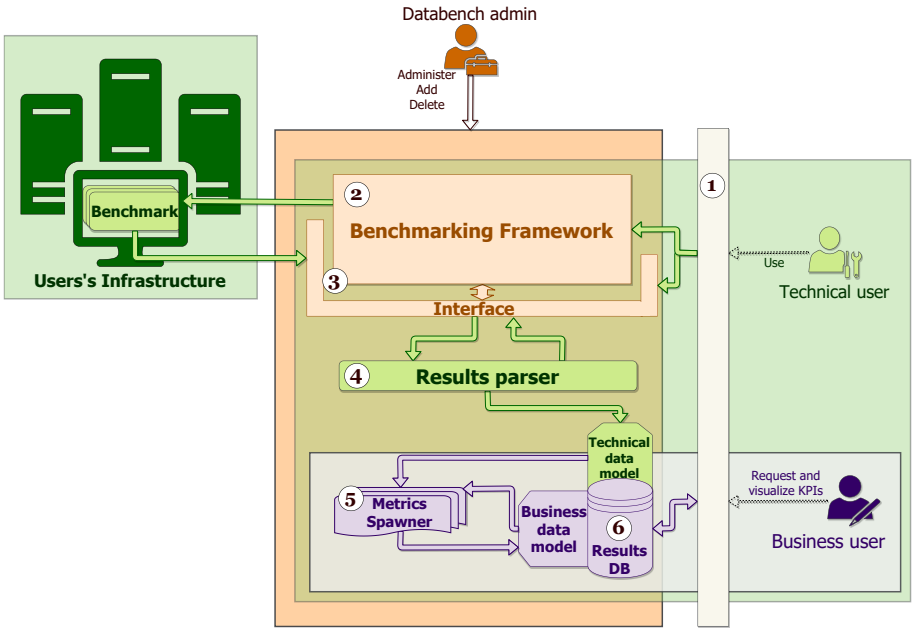

Figure 2. Functional overview of the framework architecture (from D3.1 [1])

This general overview still holds and shows the main functional blocks of the DataBench Toolbox. The implementation of functions related to these building blocks for the Alpha version of the Toolbox is still partial, and will be complemented and completed in the successive versions of the tool.

From a more technical perspective, the implementation of the Alpha version of the Toolbox therefore relies on a set of pre-existing frameworks and tools, thus avoiding starting from scratch to develop the desired functionality and that can be integrated with the Ansible approach for deployment. Figure 3 shows the main frameworks and their interaction that serve as baseline for the different elements of the architecture.

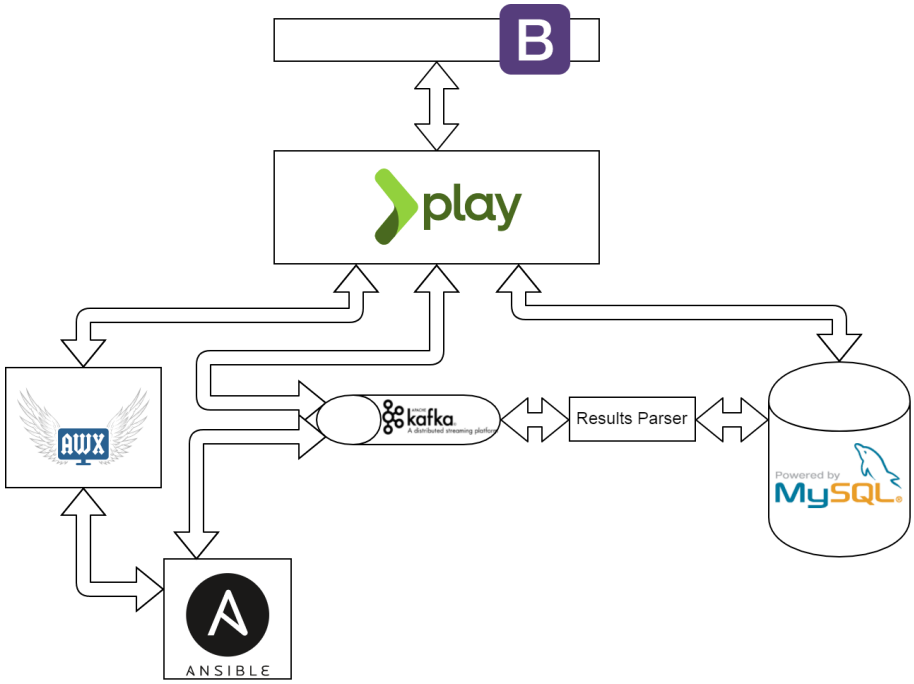

Figure 3. Overview of the technical implementation of the Alpha version

The main core of the Toolbox is hidden behind the scenes. The back-end is what provides the full range of options available to run the different integrated benchmarks. It is comprised of several modules described below:

- Bootstrap [2]: The GUI of the Alpha version has been developed using the Bootstrap framework. Bootstrap is a toolkit that allows the creation of powerful web applications based on HTML, CSS and JavaScript. It provides many open templates that facilitate the development of web sites as well as many out-of-the-box graphical elements to ease the process of web development.
- Play!-Framework [5]: It is the back-end framework used to implement the web functionality. Using Java or Scala as programming language, it provides a MVC (Model-view-controller) development pattern. It also supports easy integration with Bootstrap and different databases through JDBC making it suitable for the project.
- AWX project [6]: AWX is the upstream open source project of Ansible Tower, which allows the control of the automation of deployment of software and tools. AWX provides a web dashboard, a REST API and a task engine on top of Ansible.
- Kafka [9]: It is a well-known distributed streaming platform based on the publish-and-subscribe paradigm. It is used to act as an interface between Ansible and the Results database. Kafka is used in the Alpha version as the middleware to get the results from the execution of the benchmarks (publisher) and the Results parser (subscriber).
- MySQL [10]: Widely used open source relational database management system. The Alpha version is using MySQL as back-end because the data generated and stored in the platform is mostly of relational structure, thus making it easier to store it in a relational database. Future versions of the Toolbox will interact with the Knowledge Graph developed in the scope of WP5.

Apart from these commercial tools, the Alpha version of the Toolbox uses some custom made modules to be able to homogenize the results from the different benchmarks into a common data structure. This will allow the platform to analyse the data generated by the benchmarks to extract useful information from it to be presented to the users.

2.3 Alpha version - Toolbox back-end and repository

The automation of the configuration and deployment of benchmarks of the Alpha version is based on Ansible [8] for the back-end. As explained in D3.1: “Ansible is an orchestration, configuration and deployment tool, based on templates called playbooks that simplify the process of deploying and configuring applications in different hosts”. Therefore, the Alpha version is using playbooks to configure and enable downloading and deployment of several big data benchmarks integrated so far. The playbooks developed in the Alpha version have been stored in a playbook/configuration repository hosted in a server dedicated for DataBench. This repository of playbooks allow the configuration, selection and execution of the requested playbooks, allowing the deployment of the big data benchmarks selected in a pre-existing infrastructure. Once the desired playbook is selected, the system provides the requested playbooks to the user to configure them according to their infrastructure and software requirements.

Figure 4 is inherited from D3.1. It shows the way the orchestration of the configuration, deployment and execution of benchmarks is done in the Alpha version based on Ansible.

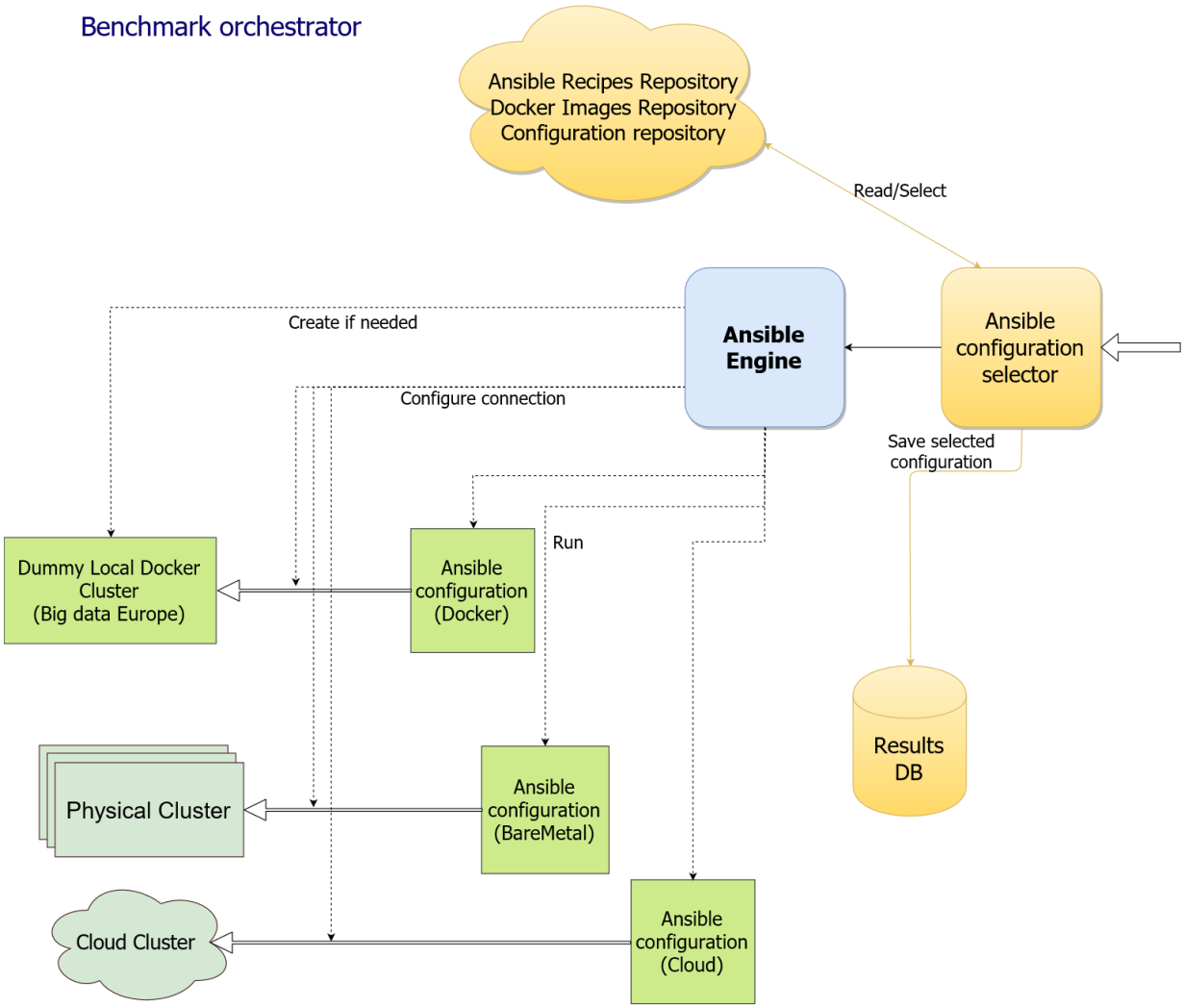

Figure 4 . Benchmark orchestrator detail (source D3.1 [1])

The Benchmark orchestrator allows the interaction with the playbooks repository and the front-end, as well as providing the way to run the playbooks, deploy, execute and retrieve the results to be stored in the MySQL implementation of the Results DB. Ansible should be installed in the master node (our DataBench server where the Alpha version of the Toolbox is running), and the connection to the hosts where the benchmarks will be deployed and configured should be enabled and configured properly to allow the installation and interaction.

The results of the execution of the benchmarks can be informed to the Toolbox by using our Kafka pub-sub connector. The results, different in format and meaning in each type of benchmark, is passed via Kafka to the MySQL Results DB, and homogenized. The Alpha version provides only a dummy implementation of the Results Parser and a visualization of the results retrieved directly.

Figure 5 shows the current data model implemented in MySQL for the Results DB used in the Alpha version.

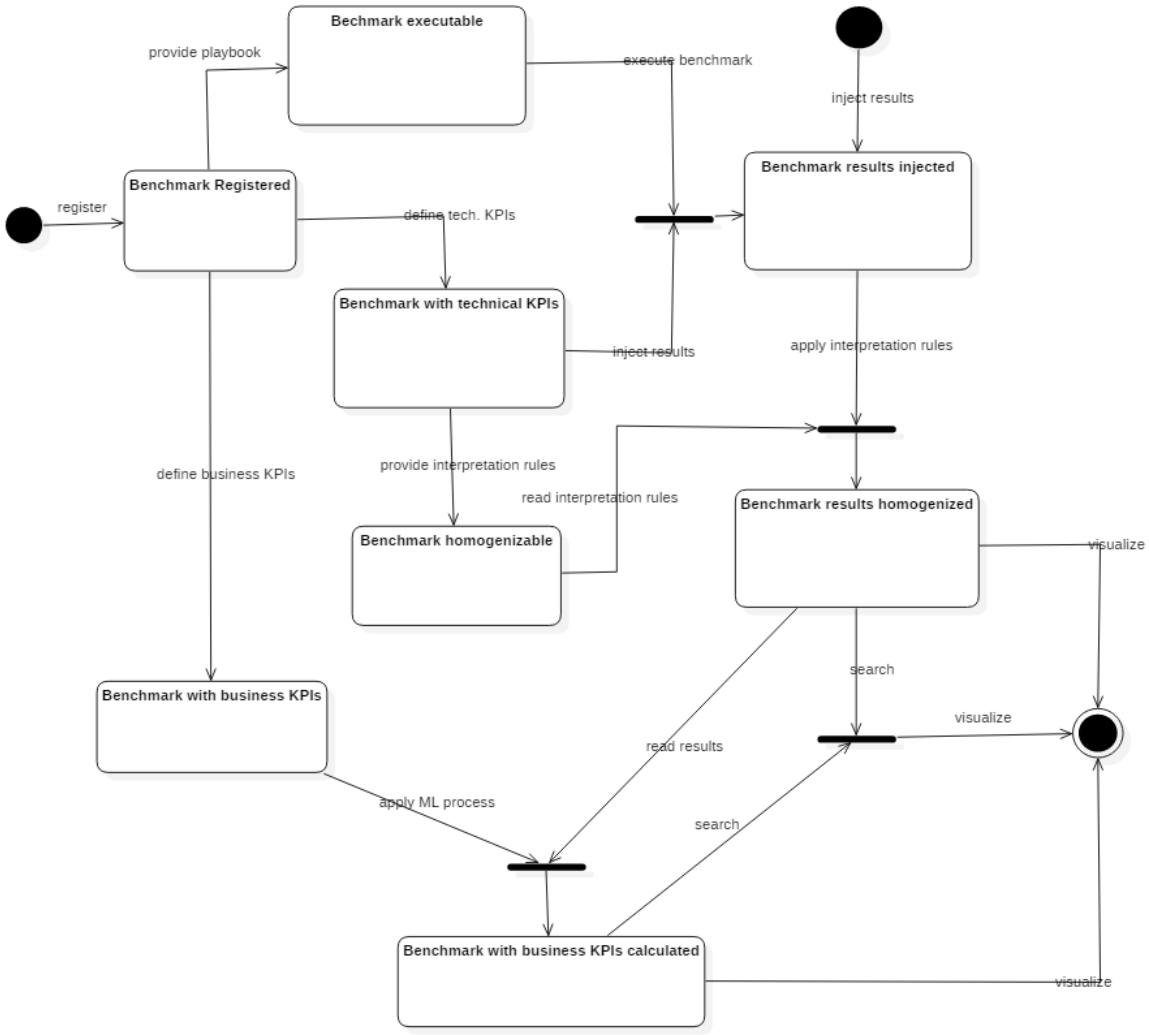

Figure 5. Alpha version data model

As far as the results database is concerned, the volume expectation is not large enough to explore highly scalable technologies and the proposed technical data model is quite relational. We need to relate the benchmark information, the information of the run and the results obtained which looks like a perfect data model for any relational database. Moreover, we aim to connect directly the dashboard/ web front-end with the results database so we need a database technology that allows connectivity from a web front-end and provides all the capabilities to be open to the public as well as to be used internally to store the results.

As anticipated, the subsequent releases of the Toolbox, expected for M26 and M30, will perform interaction with the Knowledge Graph developed in WP5. This interaction is still to be defined, but it will consist of the addition of data from the benchmark configuration and results to the knowledge graph, as well as the introduction of new functionality for end users such as searching, navigation and querying.

2.4 Alpha version - Toolbox front-end

Even though the front-end GUI of the DataBench Toolbox was not planned for the Alpha version (it was planned for the final version in M30), as was explained earlier in this document we decided to develop some Web GUI functionality to provide an easy interface for users to interact with the Alpha version and at the same time to be able to showcase the Toolbox to external communities from this very early stage.

As a result of this, the front-end is work in progress and, for some of the use cases, only the basic actions are covered. However, it is usable and sufficient to start presenting it through demonstrations.

To create the front-end of the toolbox, a well-known toolkit for front-end development called Bootstrap has been used. It is a library of modules developed to ease the prototyping using HTML, JavaScript and CSS allowing the creation of attractive and responsive web sites.

The modules already implemented in the front-end are the following:

- Related to the definition and searching of big data benchmarks
  - Support for adding new benchmarks
  - Search of benchmarks by their features
  - Guided search
  - Benchmark configuration and launch
- Related to gathering and showing results from the execution of the benchmarks
  - Inject results
  - Show results per user
- Related to User Management:
  - User creation
  - User information, login and logout
  - User credentials
  - User inventories

Section 4.2 provides an overview of the current GUI of the Alpha version and screenshots.

2.5 Alpha version – Benchmarks integrated so far

So far, the Alpha version delivers the integration of the automatic execution of the first three benchmarks, which are some of the most widely used big data benchmarks. This selection was made to test and probe the approach for integration of benchmarks into the Toolbox using Ansible, as it is by no means exhaustive. Future releases of the Toolbox will see the integration of more benchmarks as well as guidelines and best practices to make the process of integration extensible for benchmark providers.

Besides these 3 benchmarks, we have also started the integration with a performance tool developed in the scope of the EU project CLASS.

From that list of five benchmarks, we have accomplished the following:

- HiBench [17] :
  - HiBench has been integrated with Spark (2.1) and Hadoop (given a correct installation of those systems in the target host).
  - All the workloads integrated.

  - It gives the possibility to download the benchmark and compile it from the project git or run an already installed version
- Yahoo streaming benchmark [15]
  - It gives the possibility to download the benchmark and compile it from the project git or run an already installed version
  - Allows to define the versions of the software required to run
  - Working for Spark and Flink with default configuration (pending parameterization of the configurations)
- YCSB (Yahoo Cloud serving benchmark)[14]:
  - It gives the possibility to download the benchmark and compile it from the project git or run an already installed version
  - Implemented for ArangoDB, OrientDB, MongoDB and Redis
  - Using the default workloads included with the benchmark
- CLASS [11]: The Benchmarking performance tool for OpenWisk developed by IBM in the scope of the project CLASS has been initially integrated for testing purposes.

For all of them, as described in section 4.2, the front-end gives the possibility to fill in the variables used to configure each of the runs. These variables are provided from AWX so any update to them will be automatically reflected in the front-end for the user to change them according to their use case.

More than 20 other benchmarks provided by WP1 have been registered into the Toolbox, although only for searching purposes (not integrated with Ansible playbooks so far). It is not the aim of DataBench to fully integrate all possible benchmarks, but rather provide the methodology and tools to enable it for benchmark providers.

2.6 Alpha version – Features included to describe benchmarks

Starting from the previous work done in DataBench deliverable D1.1 [3], we have implemented a categorization of features that define the current benchmarks. This categorization splits the features into 3 big groups called: Benchmark Specific, Big Data Application and Platform and Architecture. This categorization is extracted form Deliverable 4.1 [2] and can be seen in the following figures (Figure 6, Figure 7 and Figure 8). Note that WP1 categorization also includes the group “business features”. These group of features will be addressed in future version of the Toolbox.

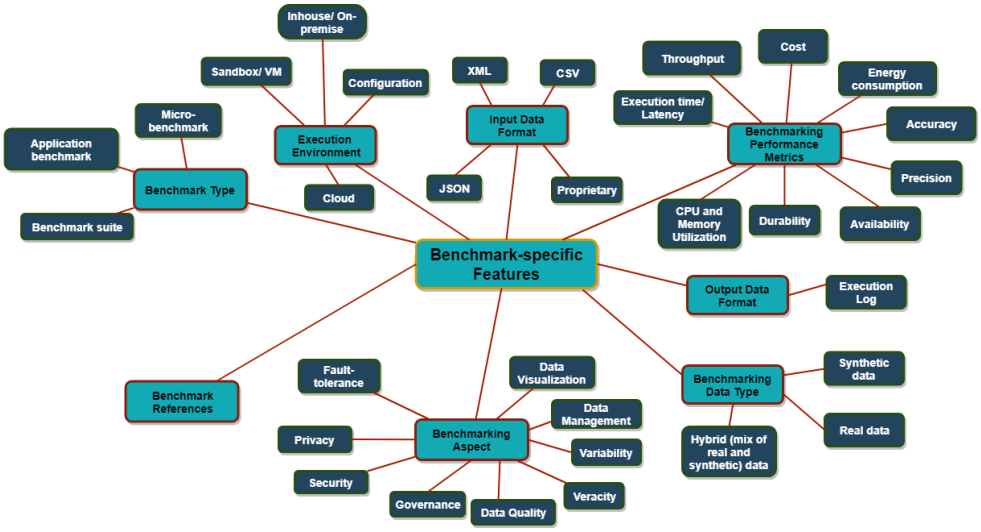

Figure 6. Benchmark specific features (source D4.1 [2])

Figure 7. Big Data Application features (source D4.1 [2])

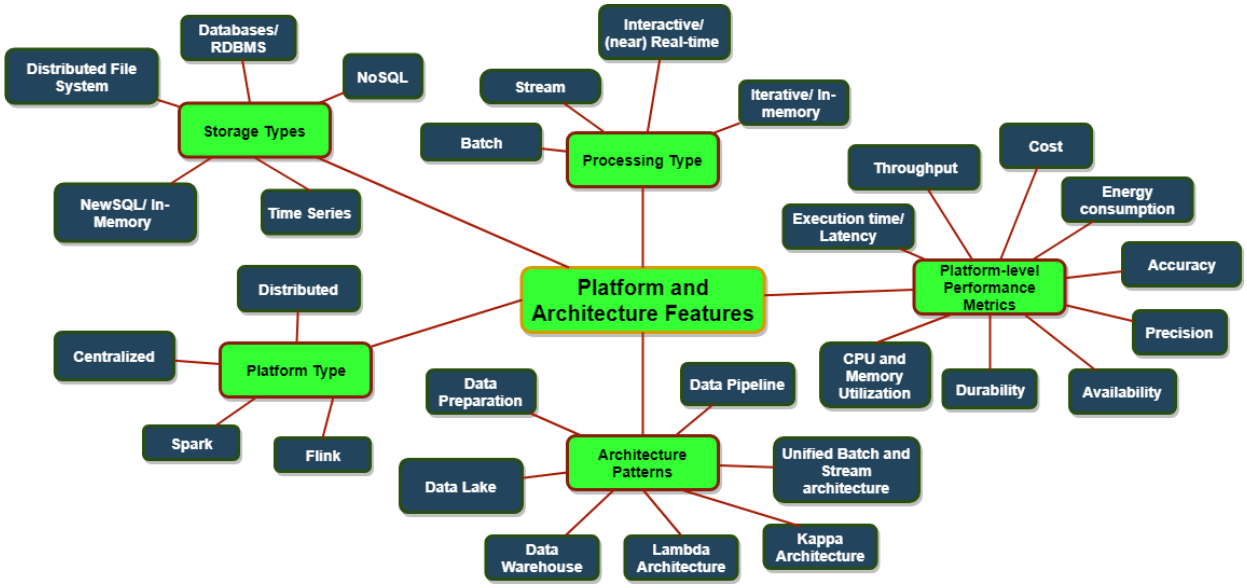

Figure 8. Platform and Architecture features (source D4.1 [2])

This features are the ones prototyped in the Alpha version, as explained in section 4, and are being used to categorize benchmarks as explained in Figure 18, Figure 19 and Figure 20.

On top of that categorization, within the Toolbox we have extended it to allow adding some extra features in any of those categories, This allows the benchmark providers to better specify the characteristics of their benchmarks by not restricting them to a set of closed ones.

3. DataBench Toolbox Alpha version back-end

3.1 Support for adding and configuring benchmarks

One of the core functions of the DataBench platform is the support for adding, configuring and running the integrated benchmarks from the web platform. To allow the platform implements the features categorization and naming extracted from WP4 to generate a creation formulary to add new benchmarks to the platform.

These new benchmarks can be further integrated with the platform by providing an ansible playbook to allow running then on any desired target host. To help on the creation of these Ansible playbooks by any of the benchmarks provider in future versions of the Toolbox we will create a set of guidelines and examples on how they can be integrated.

For those benchmarks already integrated, the platform allows some degree of configuration and personalization for each run. As can be seen in Figure 18 there is a page dedicated to describe the configuration required by each benchmark where the user can fill in the information needed to run the benchmark in their own premises. To be able to run the benchmarks in their own premises, the user will need to provide the host address in the inventory and the required credentials to log in that machine. This information, given its sensitive content, is encrypted when stored, being only shown to the logged user and not by any other, including administrators.

3.2 Support for deployment of benchmarks

DataBench is built on top of Ansible and AWX so we can use all its functions within the platform. Any task involving the deployment or configuration of any benchmark is done using these tools.

To help the user in a task that is non-trivial, like the deployment and configuration of the benchmarks, we have developed a set of Ansible Playbooks that allow a configuration- based run of the benchmarks in the infrastructure defined by the users.

For the offline option of downloading the playbooks, we have a variables file that can be configured by experienced users to run the benchmark on their premises in an easy way.

For the web platform, simply filling the variables described in the "Extra_vars", as can be seen in Figure 21, is enough for the platform's Ansible playbooks to download, compile, install, configure and run the benchmark. Moreover, we have a set of tools that also allow them to, if required, allow Ansible to send the results back to the platform without any extra interaction by the user.

In future versions of the platform, our aim is to further extend this deployment model into a fully configurable web tool that allows, not only the configuration and execution of predefined runs of the benchmarks, but also give the user the possibility to configure the integrated benchmarks to be run with any configuration that they support. This alone will already transform the DataBench Toolbox in a powerful tool to allow integrating and running a big set of benchmarks in any desired configuration, providing the users with an easy to use tool to run any benchmark they want in any environment they need.

3.3 Support for retrieving results

In order to populate the results database, DataBench will not only use the results generated by the benchmark runs on the web application. We have created a way to inject external results into the platform as can be seen in Figure 26 to be used in the knowledge graph in future versions of the Toolbox.

To do so, as we have a Kafka instance working as an API between the benchmarks and the results database, we have developed a set of connectors from the web application to that Kafka instance.

These connectors share the code with the modules in charge of sending the results embedded in the ansible recipes, but in this case they are executed from the web application.

Once the raw results are in the database, since the benchmarks are heterogeneous, a different module to parse the raw results into a more homogeneous result structure has been developed. For each of the benchmarks there is a parser submodule handling the data and converting it into the structure of results making easier the post-hoc analysis and usage of these results.

For the Alpha version of the Toolbox, a simple page for visualizing the results has been developed in order to provide the users a first glimpse of what is to be expected from the benchmark results already integrated in the platform as can be seen in Figure 27.

Future versions of the Toolbox will create a more user-friendly rendering of the results as well as implementing a Result Parser component to allow the homogenization of technical metrics.

4. DataBench Toolbox Alpha version front-end

4.1 DataBench Toolbox mock-ups

In order to bootstrap the process of generation of the GUI of the Toolbox, we started by creating an initial mock-up of the Toolbox. This mock-up GUI was initially shown to participants in a session on Benchmarking Big Data at the EBDVF 2018 conference in Vienna [12]. The mock-up GUI was developed using Balsamiq [13], which allows certain degree of user interaction and therefore gives the possibility to show the different processes in sequence.

All figures in this section are screen-shots of the mock-ups shown at the EBDVF that served for initial discussions on the processes and visualization aspects with the partners. The figures show at their left hand side a reference to the conceptual overview (Figure 1) with a red circle marking the part of the process covered by each of the mock-up figure (see Figure 1 for a more legible view of the highlighted area).

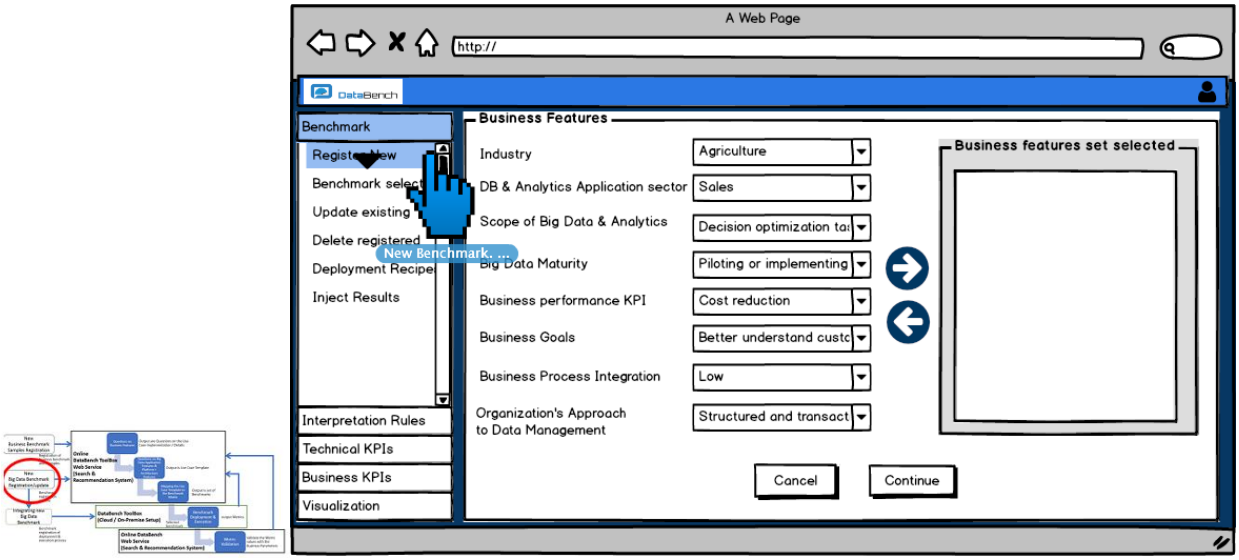

Figure 9. Mock-up - Benchmark registration process – Business Features

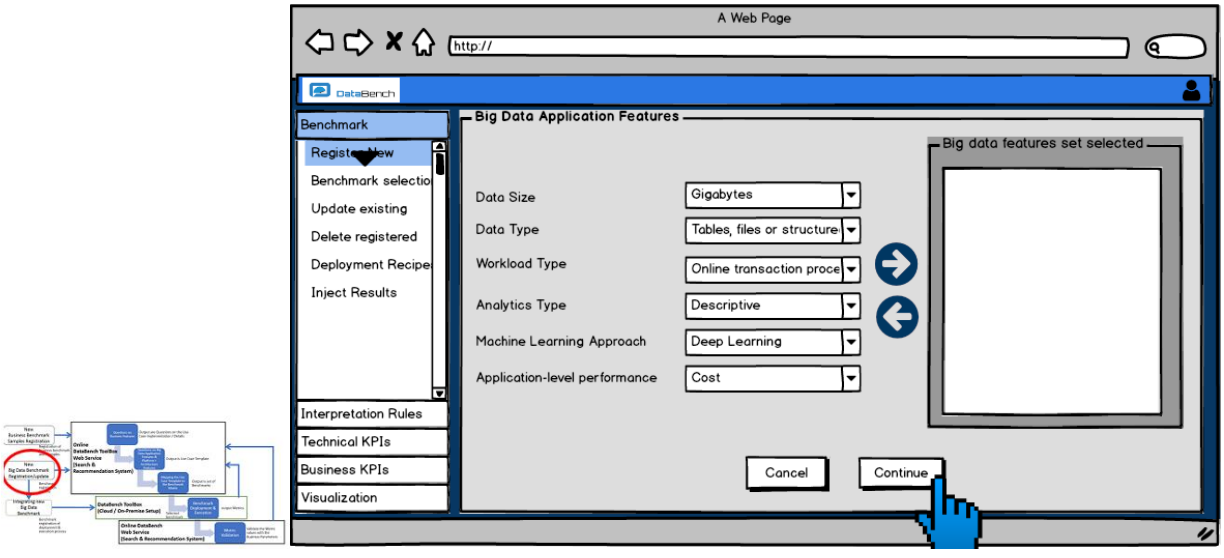

Figure 10. Mock-up - Benchmark registration process – Big Data Application Features

Registering a new benchmark includes adding business, technical, platform and specific features. The Benchmark Providers should fill-in these forms in order to categorize properly each benchmark and enable its search within the DataBench Toolbox.

Similar to what is shown for these two screenshots (Figure 9 and Figure 10), the process continues by defining other set of metadata to register properly the benchmark (Platform, Technical and Specific features). Once the new benchmark is properly defined, it is saved and ready for search, but still not ready for integration and automation via Ansible. It is expected that only a few benchmarks will be completely automated, although the Toolbox will be extensible to allow automation of as many benchmarks as possible.

If automation is possible, the users can continue the process right after finishing the initial metadata categorization, as shown in Figure 11.

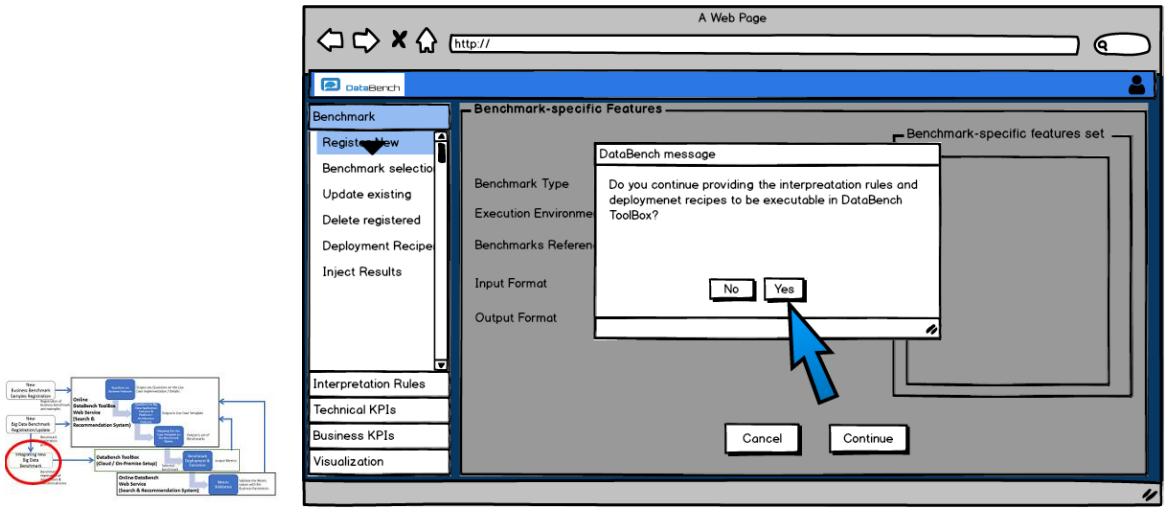

Figure 11. Mock-up - Benchmark registration process – Adding configuration for deployment

If the user decides to continue defining the way to integrate the benchmark, they have to provide some interpretation rules (to interpret the results of the benchmark), as well as the definition of the Ansible recipes for deployment and execution. Figure 12 shows the form to define the interpretation rules of a benchmark.

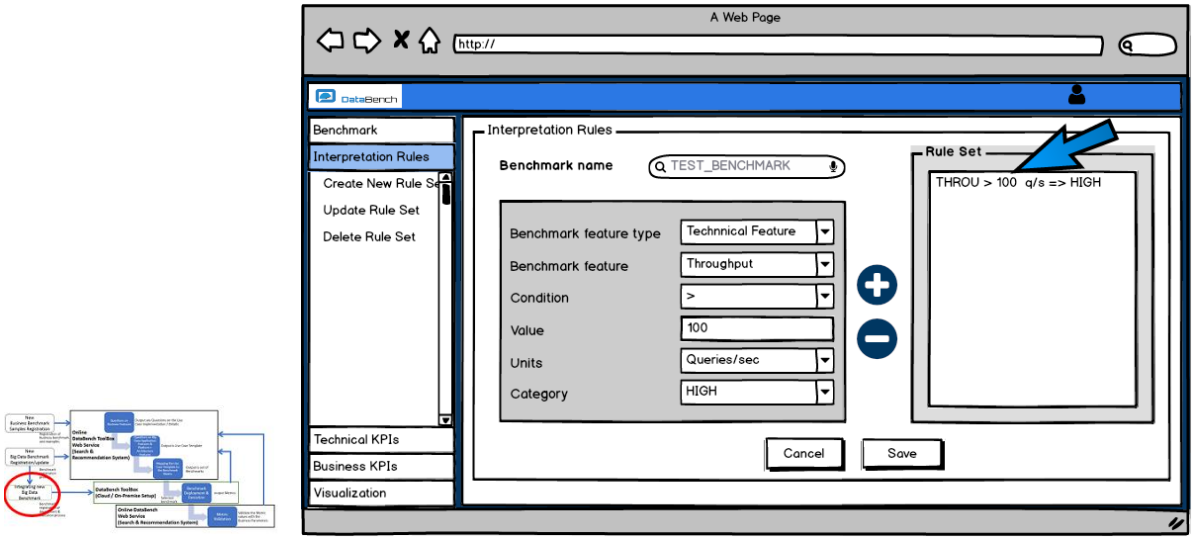

Figure 12. Mock-up - Benchmark registration process – Adding configuration for deployment - Rules

Once the benchmark has been properly defined it will be available for search and selection. One possible search functionality is shown in Figure 13.

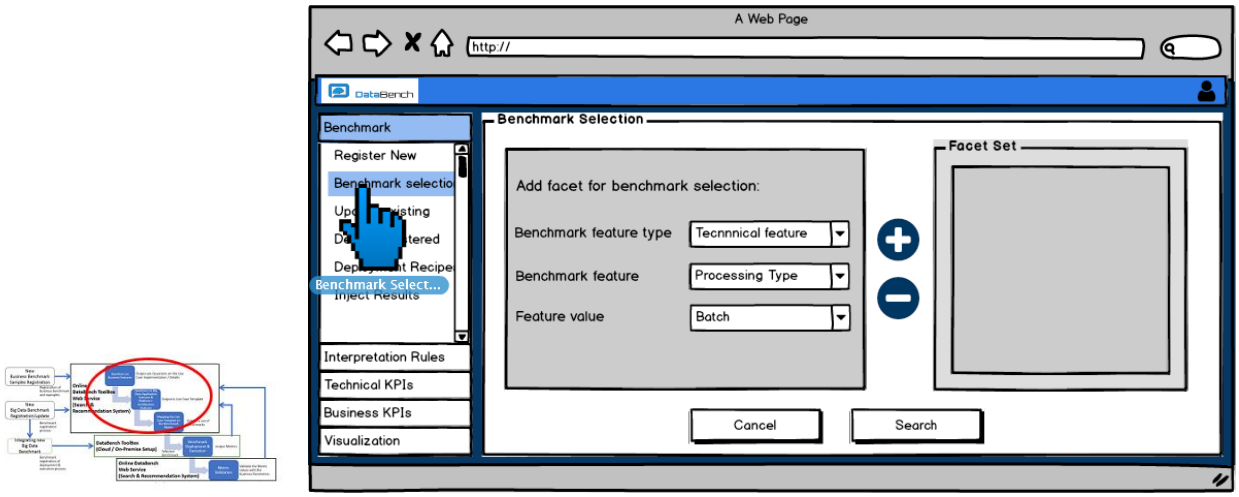

Figure 13. Mock-up - Benchmark selection process – Guided Search

If a Technical user decides to deploy one of the benchmarks, Figure 14 shows how they are able to select the appropriate benchmark and initiate the deployment, either in cloud or on-premise.

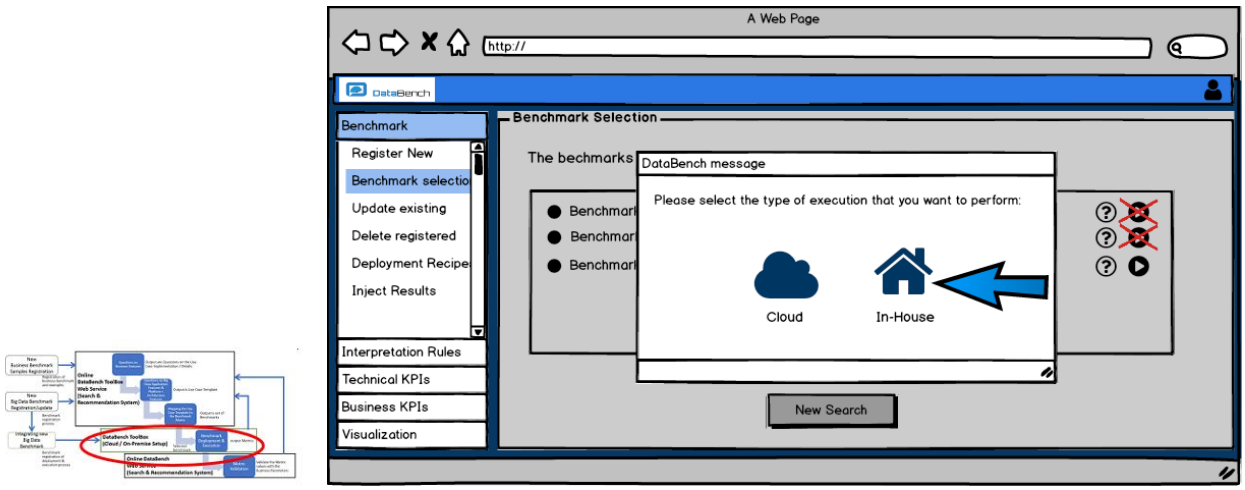

Figure 14. Mock-up - Benchmark deployment process

Injection of results back from the execution of the benchmark. The injection could be done in several ways: via a Kafka listener, direct file injection, etc. Figure 15 shows the file injection.

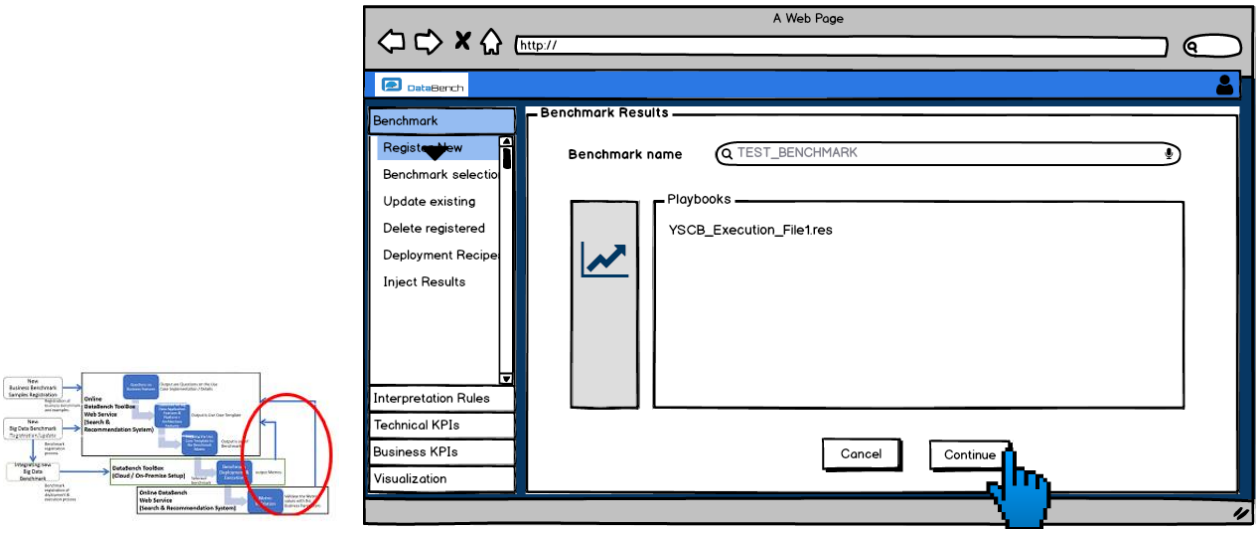

Figure 15. Mock-up - Benchmark injection process – Injection of results

And finally, the mock-up showings the results after the injection, as visualized in Figure 16.

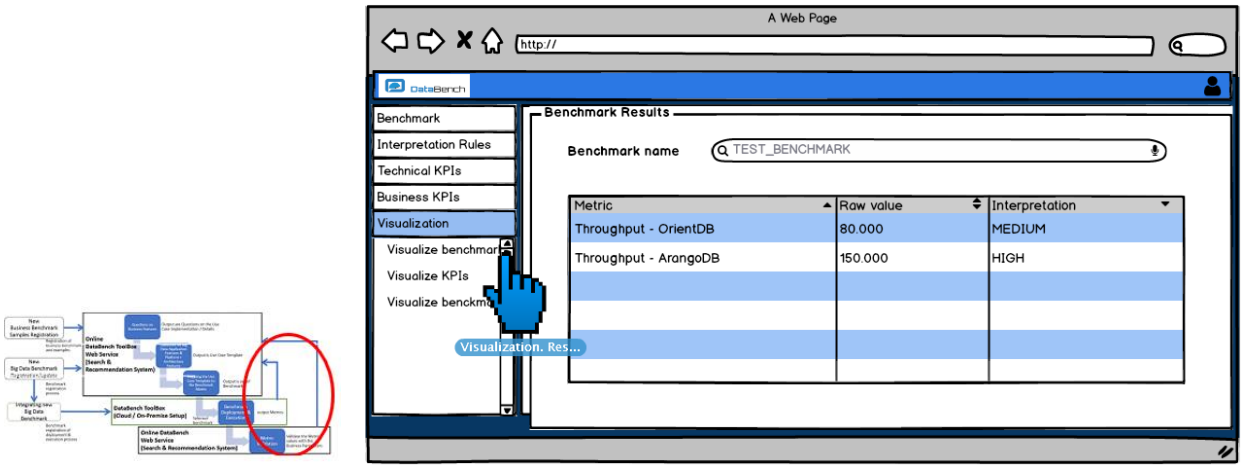

Figure 16. Benchmark visualization of results

4.2 Alpha version front-end

The mock-ups showed in section 4.1 evolved into a fully-fledged entirely functional GUI developed in Bootstrap. This is the front-end of the Alpha version of the DataBench Toolbox. The current version follows the principles shown in the mock-ups, but also shows an evolution (i.e. a revisited set of features to categorize a benchmark) and a different look and feel. The functionality to sign-in has been incorporated to allow registered users to interact with the Toolbox (Figure 1). In the Alpha version the functionality to register new users is still in draft, as it not required for testing the platforms. New users are requested to the administrator externally to the web application. Guest users can access to the Toolbox as well with limited functionality, mostly related to search and browse.

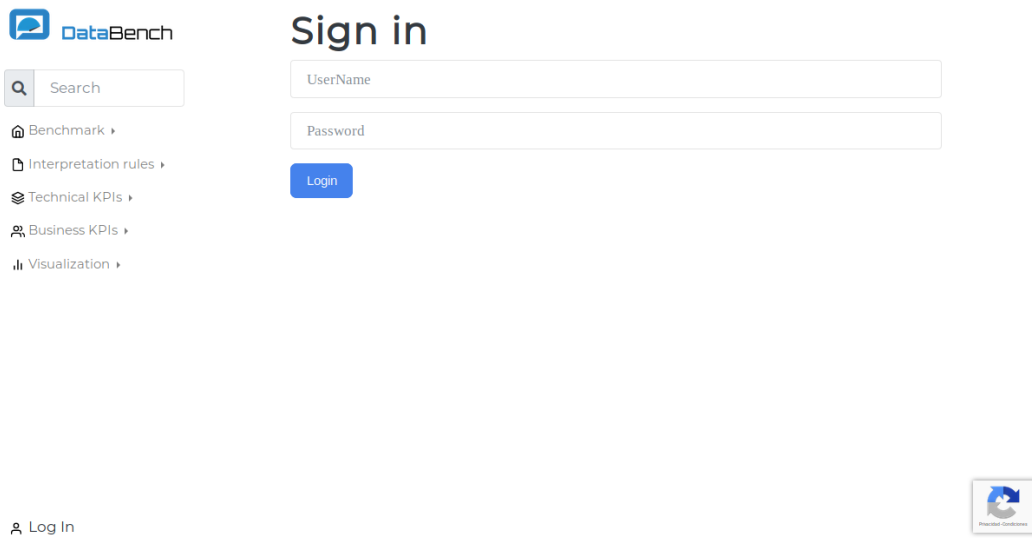

Figure 17. Sign-in / log-in

Once signed-in, registered users are able to add new benchmarks to the Toolbox by selecting the option "Benchmark – Register New" in the left menu.

Once selected, the registration process takes in total three forms to get all the metadata needed as mentioned in section 2.6. Figure 18 shows the first form that the user should fill in to add the appropriate metadata for registration, in this case the specific features of the benchmark to be registered.

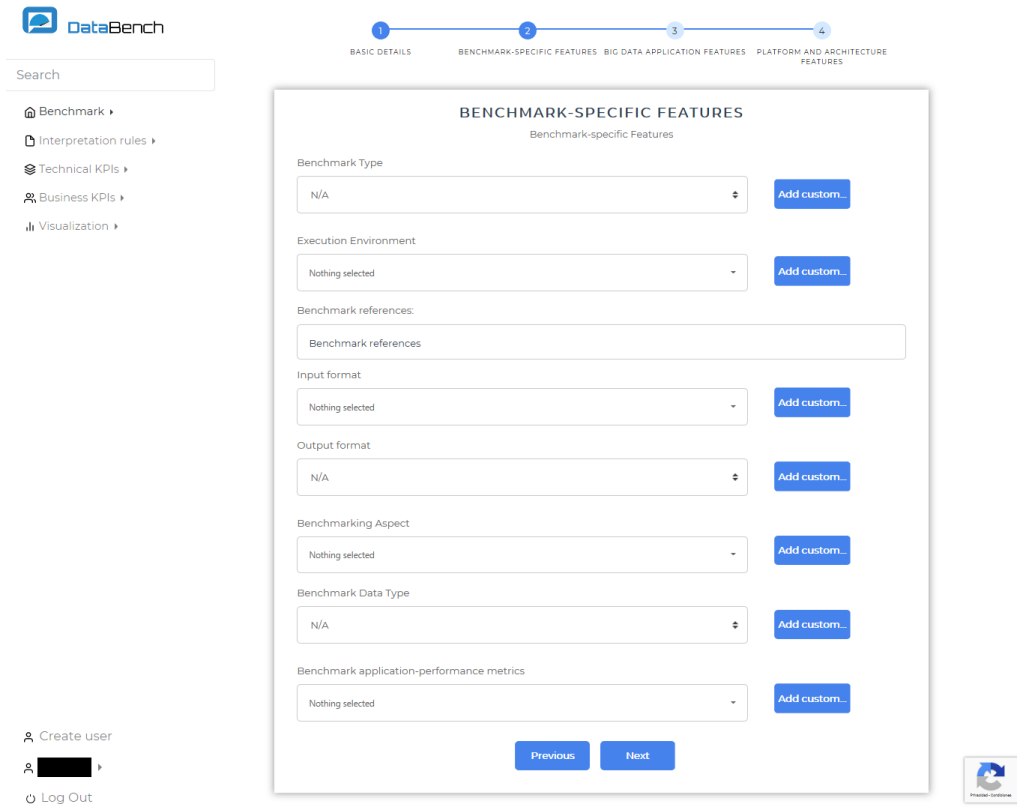

Figure 18. Benchmark registration process – Benchmark-Specific Features

Once filled-in, the second registration form depicted in Figure 19 gets the big data application features.

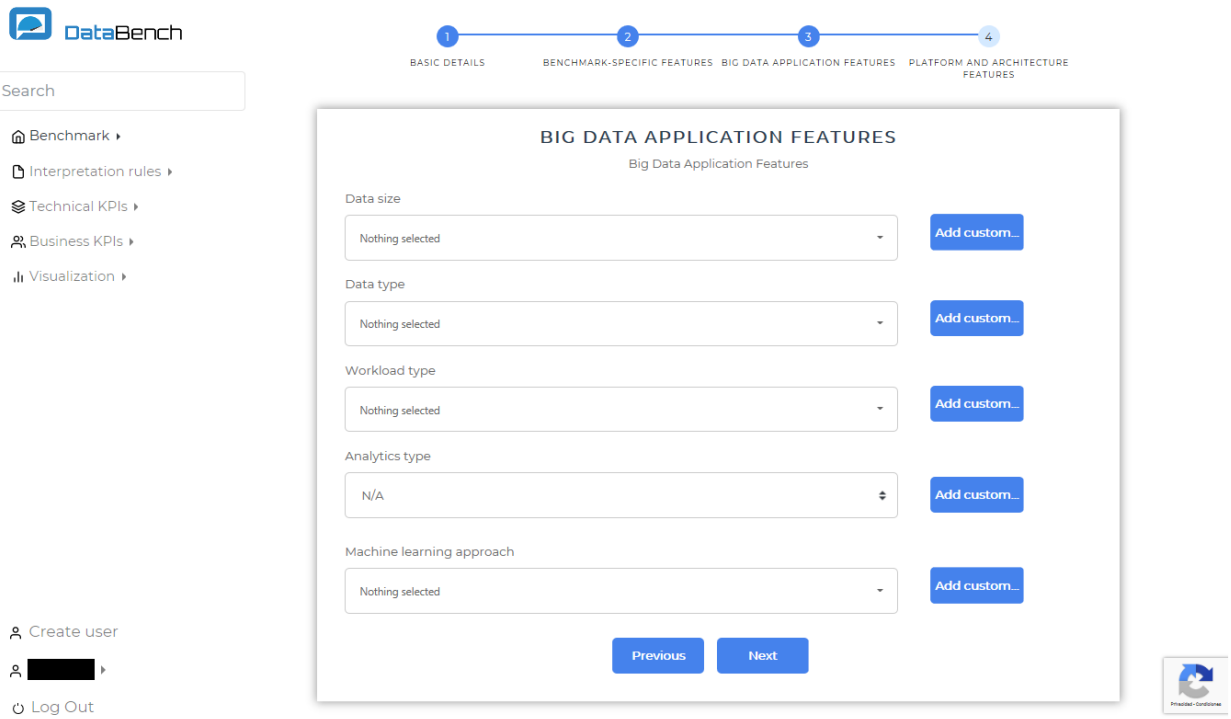

Figure 19. Benchmark registration process – Big Data Application Features

And finally the third form (Figure 20) ends the registration process by providing the platform and architecture metadata associated to the new benchmark.

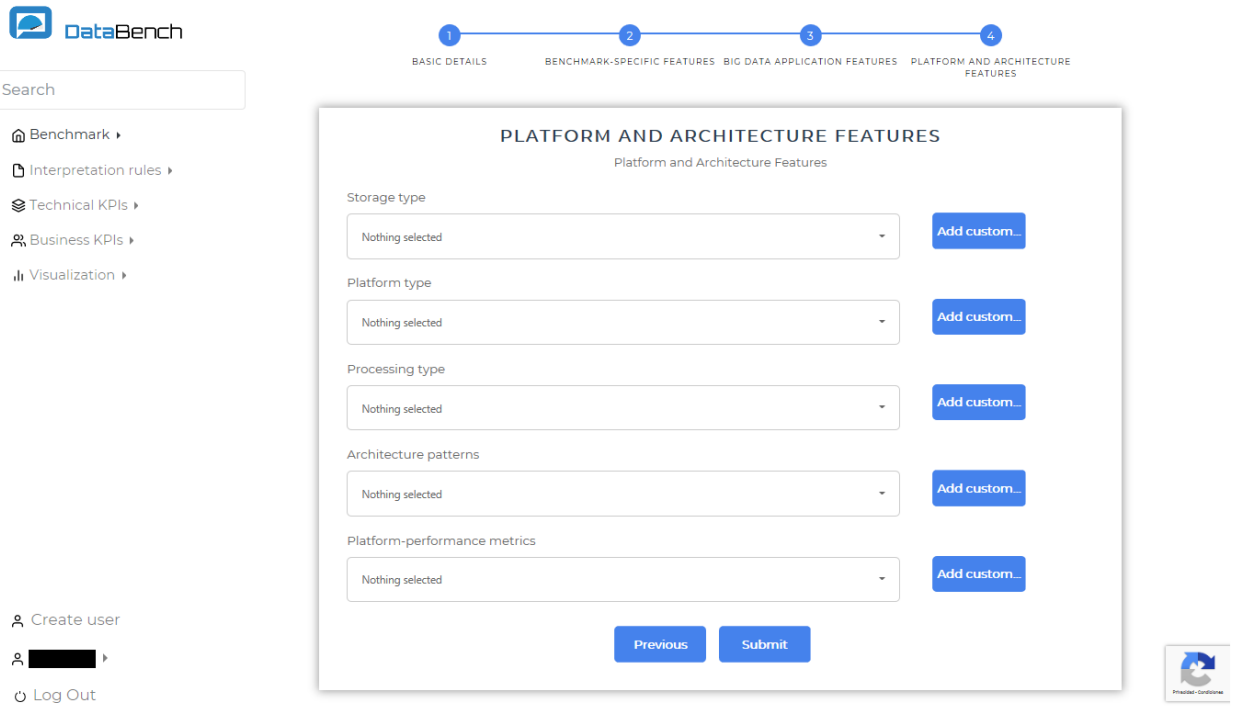

Figure 20. Benchmark registration process – Platform and Architecture Features

Once the data from the three forms are validated, the new benchmark is registered and ready for searching in the Toolbox. Note that this process is only done to enable the search of benchmarks in the DataBench Toolbox catalogue. If the user would like to go beyond mere search and enable more automation, then some other steps must be taken to properly configure the benchmark. This is called deployment configuration as shown in Figure 21.

This process requires expert users that know how the deployment and running of the benchmark can be integrated in the tool. It can be done by IT people belonging to the benchmark providers, or simply by IT personnel who know a bit about the benchmarks. For example, in the case of the Alpha version of the Toolbox, this configuration has been done for 4 benchmarks by IT people that didn't know the 4 tools in advance, but knew about Ansible and configuring automation. In future versions of the Toolbox this process will be further enhanced to allow more web-based automation.

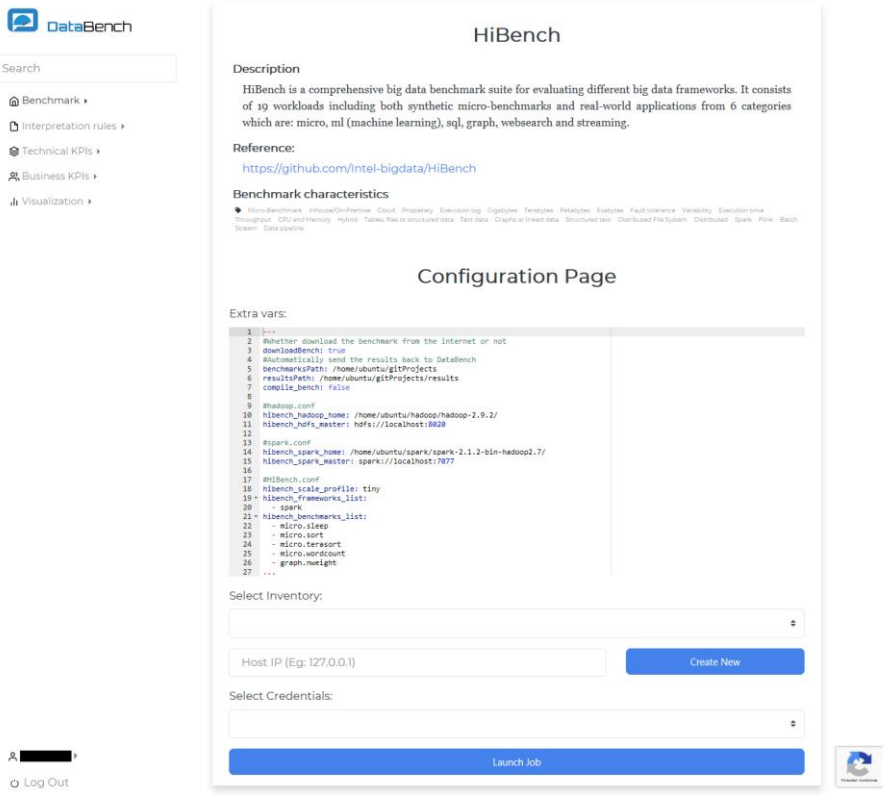

Figure 21. Benchmark registration process – Adding configuration for deployment

The Toolbox provides both for guest and registered users functionality related to search for registered benchmarks. Figure 22 shows the two possibilities:

- Search box: On top of the left menu, the search box allows for a quick search based on all the metadata features values added in the previous steps. This provides a shortcut to search for specific elements (e.g. by asking for "%Hadoop%", the system will respond with all the benchmark registered that were annotated with that metadata).
- Guided Search: By selecting the option "Benchmarks – Guided benchmark search" from the left menu, the system prompts in the Alpha version with four major metadata features to allow filtering the results. This functionality will be enlarged in future versions to be transformed in a faceted search by most of the metadata fields associated to the benchmarks.

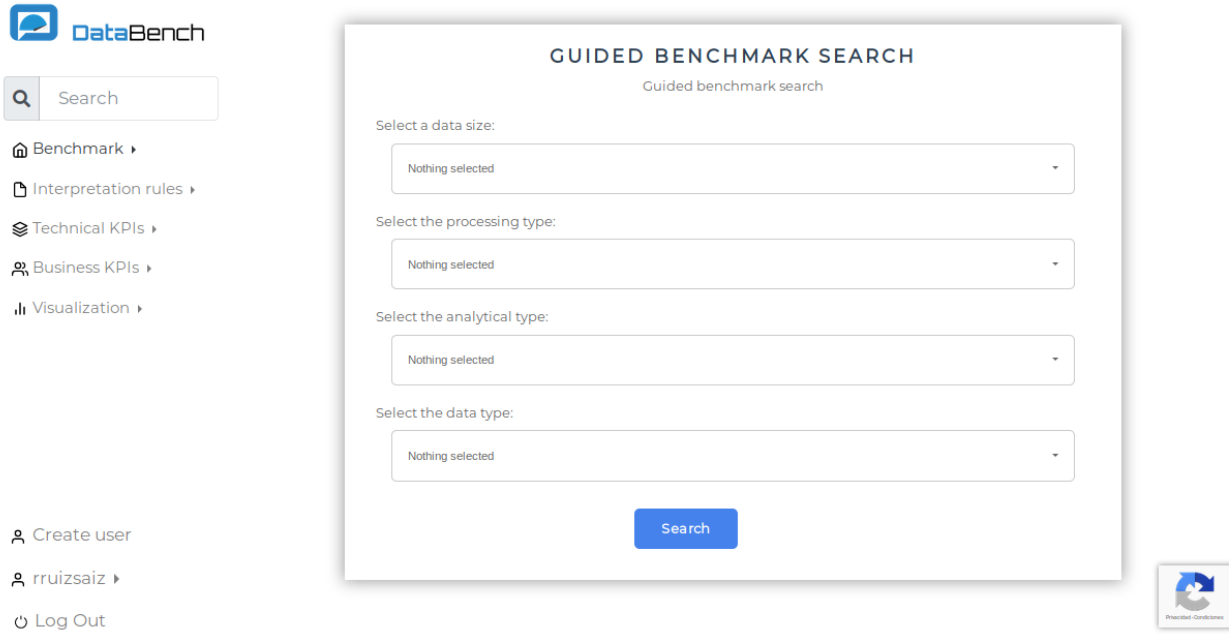

Figure 22. Benchmark selection process – Guided Search

The results of the search are shown in Figure 23. This is work in progress, but it shows the results that matches the search. In blue are the results that are benchmarks that are fully automated for deployment and running, while in grey are the ones from what we have only metadata but no further configuration. The look and feel will be changed in future versions of the Toolbox.

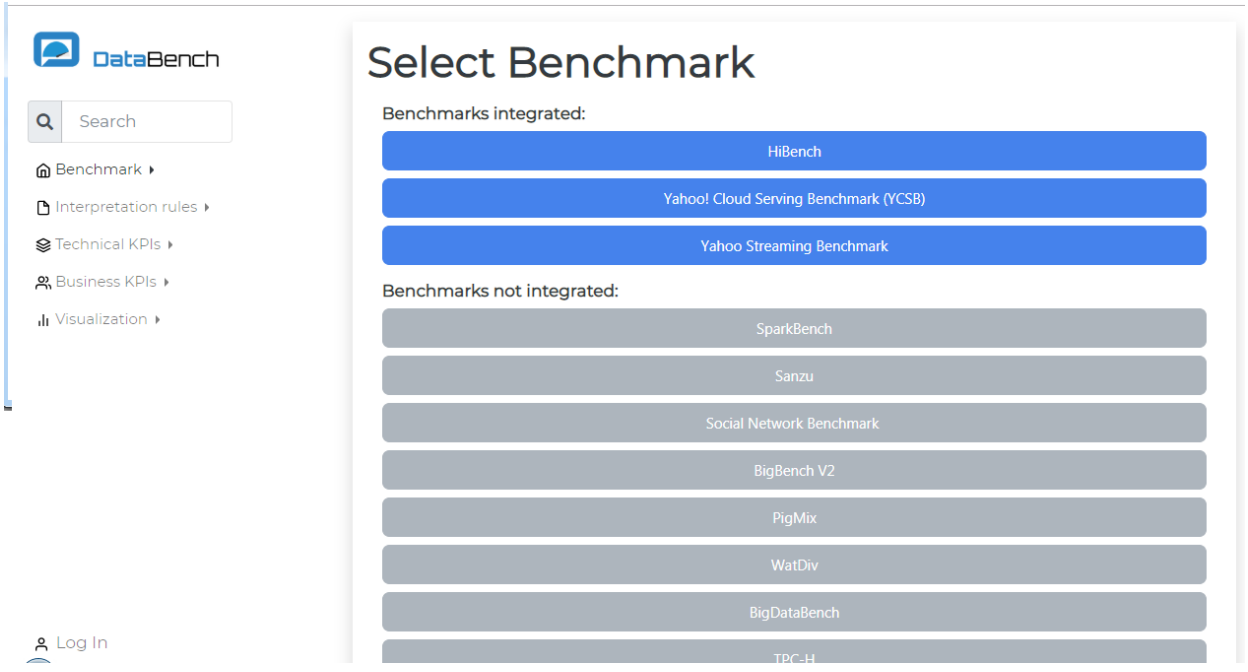

Figure 23. Benchmark selection process – Results of the search

From the previous figure it is possible to select one of the benchmarks to visualize a form with the metadata and information available in the system, as shown in Figure 24.

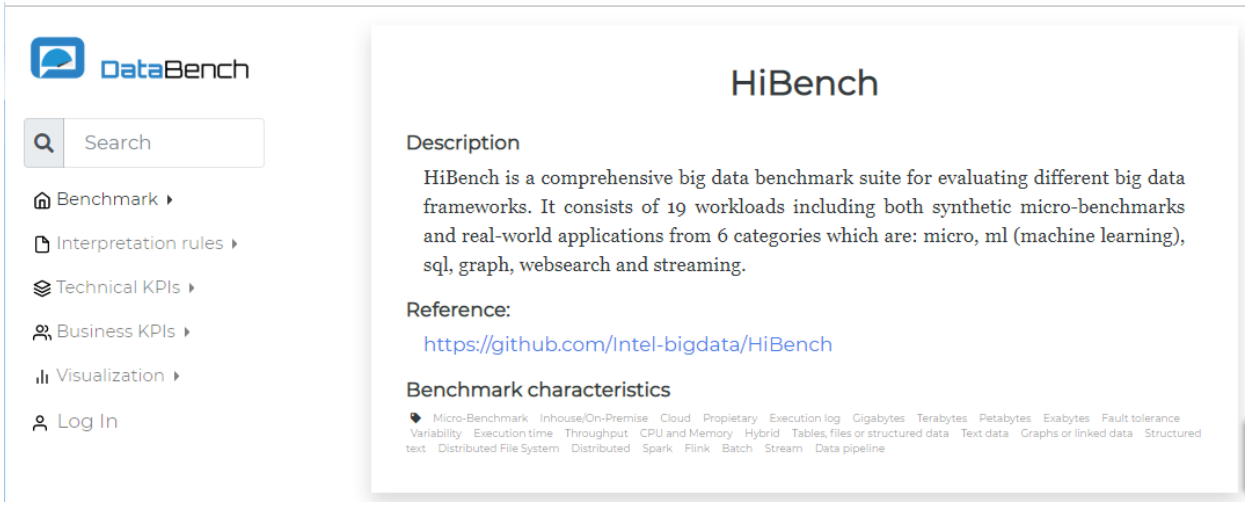

Figure 24. Benchmark selection process – Results of the search

The selection and configuration of the benchmark is performed in the form shown in Figure 21 and then, when pressing Launch button it will be sent to AWX to be deployed and run on the configured server. Figure 25 shows the execution steps, in real time, of launched benchmark as can be seen in the AWX dashboard.

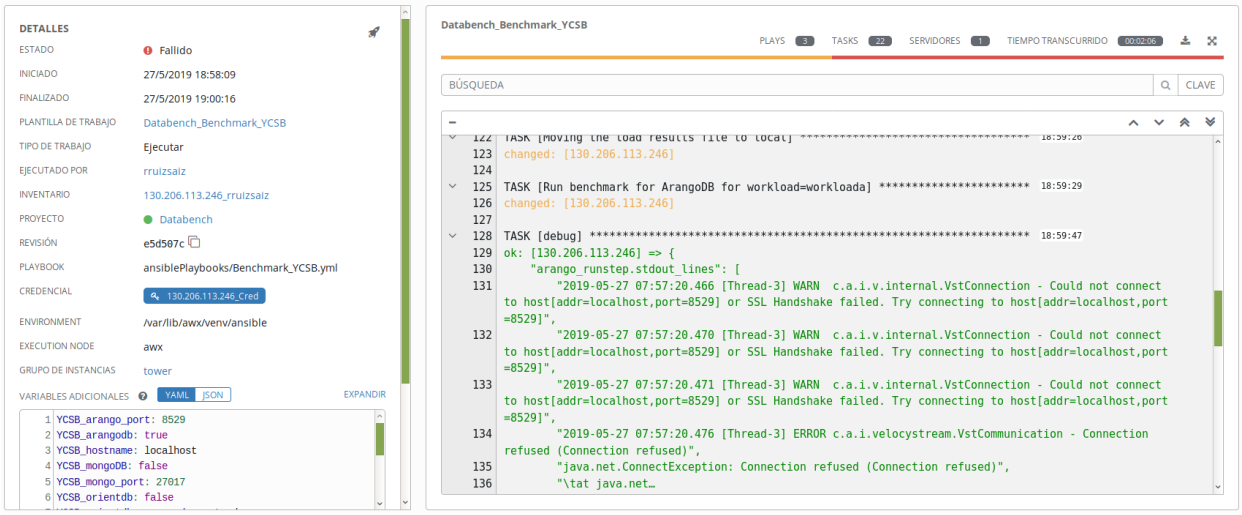

Figure 25. Benchmark deployment process

The Alpha version of the Toolbox already offers two possibilities to get the results back from the execution of the benchmark:

- Automatic injection of results: This is done automatically by Ansible and AWX based on the configuration done during the registration process, and therefore there is no need for user interaction.
- Manual upload of results: In cases where the result of the benchmark is a file (i.e. in YSCB) or the users would like to remain in control of what they upload to the Toolbox, the manual uploading is also possible. Figure 26 shows the upload of results accessible from the left menu via the option “Benchmark – Inject results”.

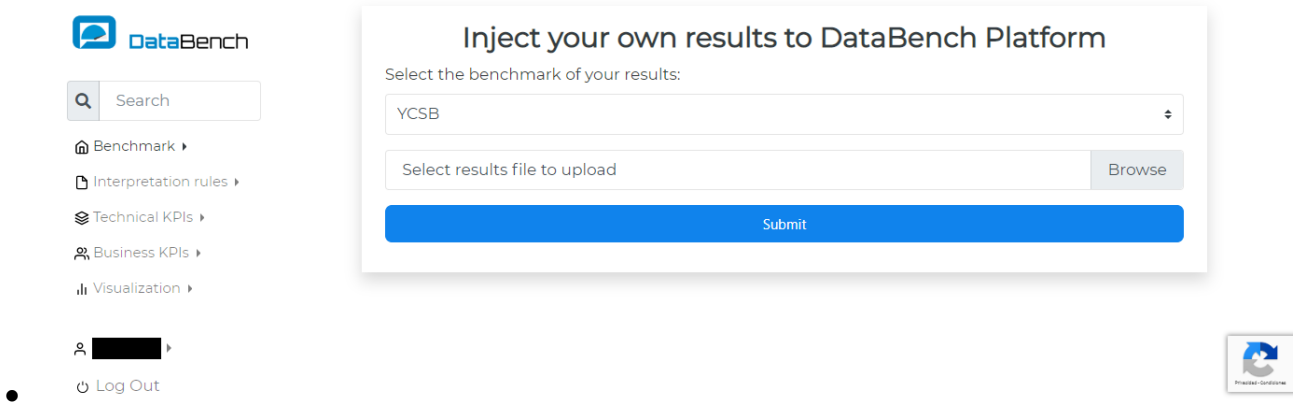

Figure 26. Benchmark injection process – Injection of results

Either way, users are able to visualize their results as shown in Figure 27. This visualization is still very preliminary and it is not formatting the results given by a tool. Future versions will provide not only formatting, but also homogenization and relation to specific technical metrics to be defined in DataBench.

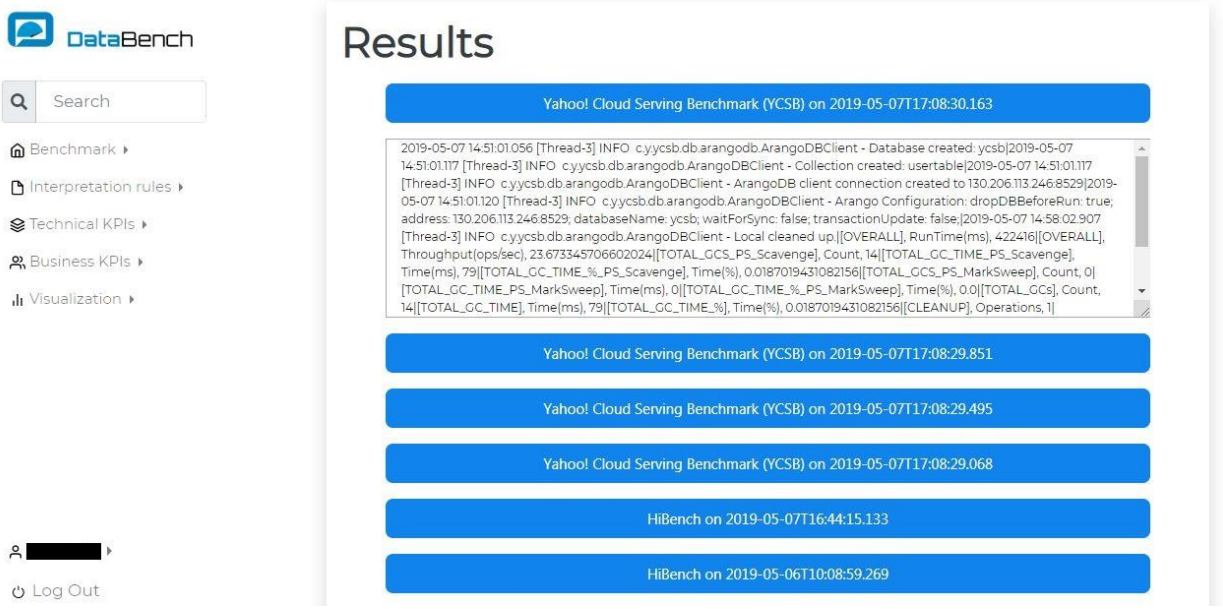

Figure 27. Benchmark visualization of results

These are the main web visualization tools developed so far for the Alpha version.

The front-end of the Alpha version dashboard can be accessed through the following URL: http://83.149.125.78:9000/

Note that this is a temporal URL to be replaced in the future when the final release of the Toolbox will be accessible to the public.

5. Conclusions and future work

The Alpha version of the DataBench Toolbox has been released as a first attempt to showcase the main functions related to big data technical benchmarking. Therefore, this document is accompanied by a web dashboard that give access to the main functionality developed so far. The document is in consequence mirroring the demo and software developed for this Alpha version, whose URL is available on request as it is not yet intended to the public, but for internal evaluation and showcasing.

This document provides a summary and an update of the architecture of the Toolbox with respect to what was reported in D3.1. From that summary, the document explains the current implementation of the architecture behind the Alpha version of the Toolbox. A description of the choices made to implement the different architectural building blocks as well as the tools used in its implementation has been also provided.

It is important to point out that the Alpha version provides a far more advanced web dashboard than originally foreseen. This has been done on purpose to be able to promote the results of the project to wider communities by having a comprehensive tool to show, and not only a good coverage of back-end functions. However, the coverage has been kept as intended and is showing a complete life-cycle of management of benchmarks since the registration process to the execution and retrieval of technical metrics.

In this sense, the document describes the functions covered in the Alpha version by the existing prototype and web interface. Mock-ups and screenshots of the main graphical artefacts has been included in the document to better understand the coverage.

But this is nevertheless an Alpha version of the Toolbox, and therefore there is a lot of room for improvement and many functions that have yet to be covered. The Alpha release covers mainly the registration, search, execution and retrieval of technical results of existing benchmarks. In this respect, besides improvements and enhancements to the current functionality, future work will be mostly focused on the connection with the knowledge graph under development in WP5, the inclusion of homogenization of technical metrics and surfacing business insights using visualization techniques with the collaboration of other work packages.

This document will be followed by two more official releases: D3.3 (Beta version) in M24 (December 2019) and D3.4 (release version) in M30 (June 2020).

6. References

[1]. D3.1. DataBench Deliverable D3.1. DataBench Architecture.

[2]. D4.1. DataBench Deliverable D4.1. Data collection plan.

[3]. D1.1. DataBench Deliverable D1.1. Industry Requirement with benchmark metrics and KPIs.

[4]. Bootstrap documentation - https://getbootstrap.com/

[5]. Play ! - https://www.playframework.com/

[6]. AWX project – https://www.ansible.com/products/awx-project

[7]. Ansible Tower - https://www.ansible.com/products/tower

[8]. Ansible - https://www.ansible.com/

[9]. Apache Kafka - https://kafka.apache.org/

[10]. MySQL - https://www.mysql.com/

[11]. CLASS EU project - https://class-project.eu/

[12]. EBDVF 2018 – Vienna – Benchmarking Big Data workshop - https://www.european-big-data-value-forum.eu/program/benchmarking/

[13]. Balsamiq tool - https://balsamiq.com/

[14]. YCSB, https://github.com/brianfrankcooper/YCSB, 2018

[15]. YSB, https://github.com/yahoo/streaming-benchmarks, 2018

[16]. BigBench - https://github.com/intel-hadoop/Big-Data-Benchmark-for-Big-Bench, 2018

[17]. HiBench Suite - https://github.com/intel-hadoop/HiBench, 2018