BigDataStack
Holistic stack for big data applications and operations

UNIVERSITY OF PIRAEUS RESEARCH CENTER

Application Dimensioning Workbench

BigDataStack Software Component developed by UPRC

Application Dimensioning Workbench aims to provide insights regarding the required infrastructure resources for the data services and application components (micro-services), linking the used resources with load and expected Quality of Service (QoS) levels.

Input

Application playbook created in Data toolkit and annotated with candidate deployment patterns in Pattern Generation

Output

Annotated playbook with expected QoS levels for each pattern

Initial TRL

- TRL 3 for the model creation part,
- TRL 0 for the benchmarking part, as this did not exist at the beginning of the BigDataStack project.

Final TRL

TRL 5

End Users

Primarily data service and application owners.

BigDataStack
Holistic stack for big data applications and operations

UNIVERSITY OF PIRAEUS RESEARCH CENTER

Application Dimensioning Workbench

Key Features and Benefits

The component has two purposes: - initially benchmark the target service via easily configured and automated parameter sweep tests, thus gathering the necessary performance data; - train prediction models that are able to regress for cases that have not been met before.

Essential Information for Users

• Load injection needs to be adapted to each service that needs to be benchmarked,
• The service structure should follow the format defined by the Data toolkit,
• Deployment and execution of the tests depends on the used deployment platform (e.g. Docker Swarm, Kubernetes etc)

Standards involved in the development of the component

• YML,
• Docker Compose v3

BigDataStack
Holistic stack for big data applications and operations

UNIVERSITY OF PIRAEUS RESEARCH CENTER

Application Dimensioning Workbench

Implementation in BigDataStack Use Cases

The application dimensioning workbench is primarily related to the generic data services offered by the project.

How can the BigDataStack component contribute to Standardization foundations or initiatives?

Not applicable

Differentiators from competitors in the market

• Workload in CBTool is completely designed by user/contributor, thus cannot be easily abstracted and generalized,
• Static setups based on specific stacks, measuring in fact the underlying resources and not the software parameterization.