KOBE: Cloud-native Open Benchmarking Engine for Federated Query Processors

Charalampos Kostopoulos[1][2], Giannis Mouchakis[1],
Nefeli Prokopaki-Kostopoulou[1], Antonis Troumpoukis[1],
Angelos Charalambidis[1], and Stasinos Konstantopoulos[1]

[1] Institute and Informatics and Telecommunications, NCSR "Demokritos", Greece
{gmouchakis,nefelipk,antru,acharal,konstant}@iit.demokritos.gr
[2] ECE, National and Technical University of Athens, Greece
el09161@mail.ntua.gr

Abstract. KOBE is a benchmarking system that leverages modern containerization and Cloud technologies. Data sources are formally described in more detail than what is conventionally provided, covering not only the data served but also the specific software that serves it and its configurations. KOBE provides a specification formalism and a command-line interface that completely hides from the user the details of provisioning and orchestrating the benchmarking process. Finally, KOBE automates collecting and comprehending logs, and extracting and visualizing evaluation metrics from these logs.

Keywords: Benchmarking, Federated querying, Cloud-native

1 Introduction and Motivation

In the federated query processing community releasing a benchmark amounts to releasing datasets, query workloads, and, at most, a benchmark-specific evaluation engine for executing the query load [2, 4, 3]. Note, however, that federated query processors do not manage the data they serve, but instead provide a data-integration abstraction over the actual query processors that are in direct contact with the data. As a consequence, benchmark results can be greatly affected by the performance and characteristics of the underlying data services; and even more so under realistic conditions, where internet latency and throughput between the federator and the federated data sources is a key factor.

Research articles need to specify what software has been used to implement the SPARQL endpoints, how it has been configured and distributed among hardware nodes, and the characteristics of these nodes and of the network that connects them to the federation system. Reproducing an experiment from such a description is a tedious task; Based on our own experience with federated query processing research we strive to minimize the effort required and uncertainty involved in replicating experimental setups from the federated querying literature.

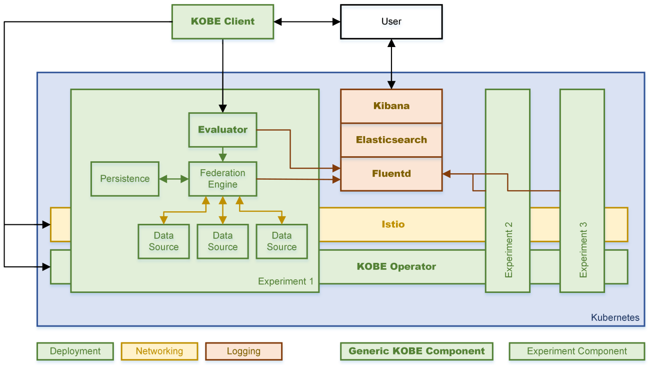


Fig. 1. Information flow through a KOBE deployment.

Our first step in that direction was to complement our own benchmark with Docker images of the populated triple stores and of the federation systems [6].

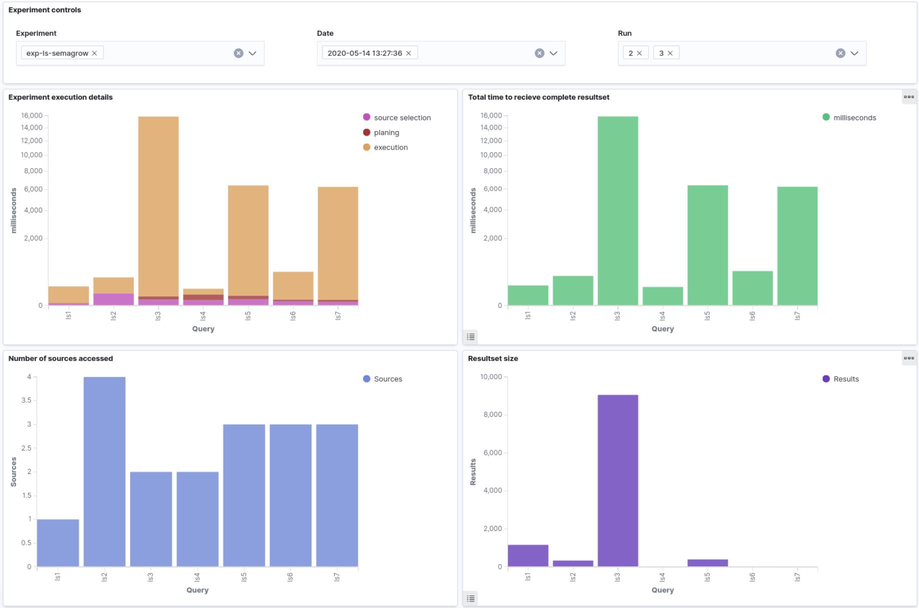
Following up, we further developed the KOBE Open Benchmarking Engine into a framework that leverages modern containerization and Cloud computing technologies in order to reproduce benchmarking experiments. Data sources are formally described in more detail, covering not only the data served but also the specific software that serves it and its configuration. KOBE provides a specification formalism and a command-line interface that hides from the user the mechanics of provisioning and orchestrating the benchmarking process on Kubernetes-based infrastructures; and of simulating network latency. Finally, KOBE automates the process of collecting and comprehending logs, and extracting and visualizing evaluation metrics from them.

2 The KOBE open benchmarking engine

KOBE comprises three subsystems (Figure 1):

1. The *deployment subsystem* uses Kubernetes to deploy containerized images of the experiment components, and deploys the *operator* that orchestrates benchmarking and monitors the progress of each experiment in the cluster.
2. The *networking subsystem* uses Istio, a Cloud-native controller that tightly integrates with Kubernetes. The KOBE operator utilizes Istio to setup the network connections between the data sources and the federation engine, explicitly controlling their quality to simulate a specified behaviour.
3. The *logging subsystem* is implemented as an Elasticsearch/Fluentd/Kibana (EFK) stack. It collects and manages the logs produced by the various experiment components (the data sources, federators and evaluators) and produces meaningful diagrams and graphs about the benchmarking process.

The users interact with the operator via a command-line interface to submit experiment descriptions. These descriptions are YAML files that specify the



Experiment controls

Experiment: exp-ls-semagrow

Date: 2020-05-14 13:27:36

Run: 2, 3

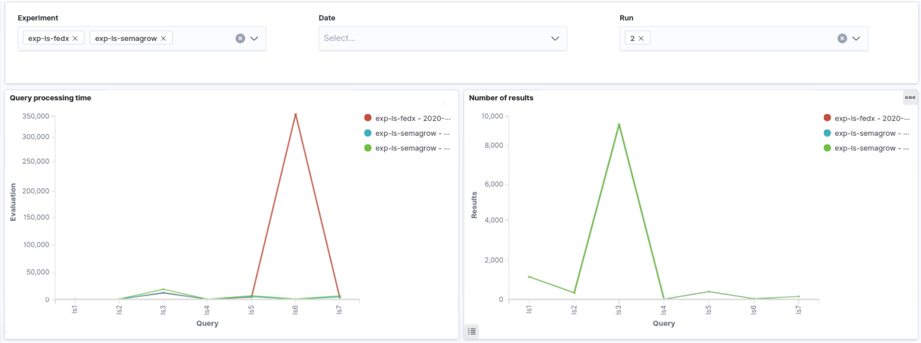
Fig. 2. Details of a specific experiment execution (average of multiple runs)

query load, the federator that is being benchmarks, and the data sources it federates. A collection of federators and benchmarks are included in the KOBE distribution, but the user is able to also provide their own as Docker images.

The operator uses the *evaluator*, provided as part of KOBE, to apply the query load to the federator. Istio manages the network connections between the data sources and the federation engine, allowing control over the virtual network characteristics. Finally, the EFK stack collects logs, extracts metrics, and provides visualizations of the these benchmark metrics. A set of Kibana panels relevant to benchmarking federated query processing are included in the KOBE distribution, but, naturally more can be defined by the user.

3 Description of the Demonstration

A benchmarking experiment is expressed as YAML files that specify data sources, queries, and parameters of the experiment and the evaluator. The operator uses this information to deploy the necessary containers and execute the experiment. We will show the contents of these YAML files and demonstrate how their content impacts experiment outcomes. We will then show how to submit an experiment using the command-line application. We will inspect the deployed containers, showing how, for example, for metadata-based federation engines [e.g., Semagrow, 1] the relevant metadata has been extracted and made available whereas for engines where this is not required [e.g., FedX, 5] this step has been skipped.



Experiment: exp-ls-fedx ×, exp-ls-semagrow ×

Date: Select...

Run: 2 ×

Fig. 3. Comparison of three experiment executions

We will also demonstrate the ability to separate the logs produced by multiple executions of the same query, e.g., to get an average or to try different parameters in different experiments. In order to differentiate between executions and comprehend logs from a specific execution of a specific experiment, each log contains a unique id generated by the KOBE operator. The id is passed to the federator via a *SPARQL comment*. Federators that have been adapted to KOBE read this id and include it in the logs; those that have not been modified for KOBE are unaffected as they ignore the comment in the SPARQL query.

Naturally, detailed metrics about each phase of query processing cannot be obtained from such engines, and results are only reported at the level of complete query runs. As an example, consider Figure 2 reporting an experiment execution for the life-science (ls) query set of the FedBench benchmark for the Semagrow federation engine (adapted to KOBE), and compare it to the coarser graph in Figure 3 that compare Semagrow and FedX (not adapted).

4 Conclusion

KOBE is a benchmarking system that allows federated querying researchers to publish more complete and reproducible descriptions of their experiments, and that automates the tedious tasks of deploying the data services needed to execute an experiment, collecting logs, and extracting evaluation metrics.

KOBE is developed as an open source project and the repository includes instructions and example experiments: https://github.com/semagrow/kobe

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825258. Please see http://earthanalytics.eu for more details.

Bibliography

[1] Charalambidis, A., Troumpoukis, A., Konstantopoulos, S.: SemaGrow: Optimizing federated SPARQL queries. In: Proc. 11th Intl Conference on Semantic Systems (SEMANTiCS 2015), Vienna, Austria, 16–17 Sep 2015 (2015)

[2] Guo, Y., Pan, Z., Heflin, J.: LUBM: a benchmark for OWL knowledge base systems. Web Semantics 3(2) (Jul 2005)

[3] Saleem, M., Hasnain, A., Ngomo, A.N.: Largerdfbench: A billion triples benchmark for SPARQL endpoint federation. J. Web Semant. 48. (2018)

[4] Schmidt, M., Görlitz, O., Haase, P., Ladwig, G., Schwarte, A., Tran, T.: FedBench: A benchmark suite for federated semantic data query processing. In: Proc. ISWC 2011, Bonn, Germany, October 2011. LNCS vol. 7031. (2011)

[5] Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: FedX: A federation layer for distributed query processing on Linked Open Data. In: Proc. ESWC 2011, Heraklion, Greece, May 2011. LNCS vol. 6644. Springer (2011)

[6] Troumpoukis, A., Charalambidis, A., Mouchakis, G., Konstantopoulos, S., Siebes, R., de Boer, V., Soiland-Reyes, S., Digles, D.: Developing a benchmark suite for Semantic Web data from existing workflows. In: Proc Benchmarking Linked Data Workshop (BLINK), ISWC 2016, Kobe, Japan. (2016)