Midas - An interactive data catalog for data science teams

Patrick Holl
Technical University of Munich
patrick.holl@tum.de

Kevin Gossling
Technical University of Munich
kevin.gossling@tum.de

ABSTRACT

Midas is a research project following the goal of providing data science teams fast and agile access to data in a processable format. The system combines sophisticated data cataloging features with ad-hoc queryability. It implements a novel, graph-based approach to virtualize datasets across multiple storage technologies.

KEYWORDS

polystore, data management, data virtualization

1 INTRODUCTION

In this extended abstract, we present the on-going research project Midas. Midas is an interactive data catalog system designed for heterogeneous data landscapes and driven by the goal of making data science teams more agile and productive in accessing and preparing datasets.

To provide data consumers, e.g., analysts and data scientists, with the data they need, enterprises create comprehensive data catalogs. These systems crawl data sources for metadata, manage access rights and provide search functionality. Such catalogs are the starting point for almost every analytical task. Once a data scientist has found a potentially interesting dataset in the catalog, he/she has to move to another tool in order to prepare it for analysis. This is because data catalogs often cannot interact with their referenced data sources directly. Instead, engineers have to build ETL pipelines to move and shape data so that it is ready for analysis. This process is time consuming, costly, and can even lead to the insight that the dataset is unsuitable for the intended task because it is hard to asses the data quality based on raw metadata. Even highly sophisticated systems like Goods from Google require such processes [1]. Another challenge for data catalogs is tracking the provenance of derived datasets, specifically when the schema and the location is different from the origin data. In such cases, the datasets need to be registered manually back to the catalog.

Midas tackles the stated problems by providing a large scale data virtualization environment that combines ad-hoc analytical query access with sophisticated metadata management features. In this context, we define interactive as the ability for a data scientist to run large scale ad-hoc queries within the same application that manages the metadata of connected data stores. This approach enables data science teams to share schema details, comments, and other important information in the same place where they access, prepare and analyze the data.

The core concepts of Midas are: Virtualized and sharable datasets, sophisticated metadata management, comprehensible data lineage, interactive performance through adaptive columnar-oriented caches, and ease-of-use.

Figure 1 depicts the overall architecture of the Midas system. The

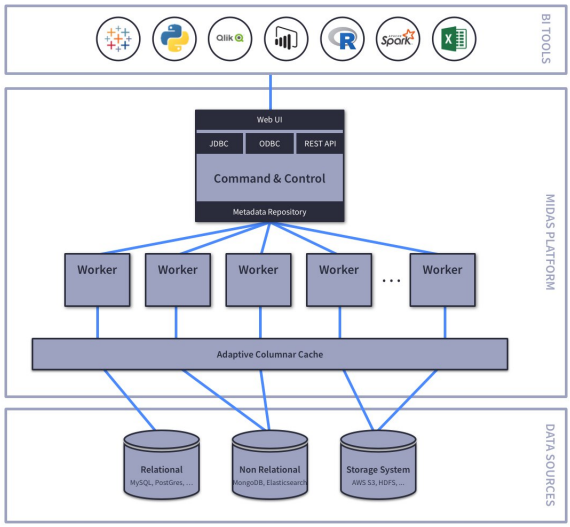

Figure 1: Midas system architecture

architecture is inspired by Dremel and it's open source implementations Apache Drill and Presto [2, 4, 5]. The main components are a distributed, SQL-based query engine that enables users to create virtualizations even on massive scale datasets and the user interface as an interactive data catalog.

2 USE CASE: OPEN DATA ENRICHMENT

The research project is funded by the German government and contributes to the open data movement. Currently, we are evaluating the system in data science teams working in two major German enterprises (>100.000 employees combined). Both data science teams face the challenge that the enterprise's internal data lacks features to build analytical models that potentially lead to new valuable insights. For instance, one team wants to build a model to predict the risk that a particular supplier of this enterprise fails in the delivery process. The risk score should be based on incidents that are happening near the geographical location of this supplier. To build an anlytical model for the risk score, the data science team has to combine an internal dataset that contains supplier information with an openly available dataset like GDELT that contains information about incidents [3]. The internal dataset is provided in JSON format in a document store and GDELT are multiple CSV files. Using a conventional approach requires a data engineer to build an ETL pipeline that loads the internal and the open dataset in a common store and join them by a shared attribute. This process results in a materialized dataset that has several drawbacks in terms of reusability, storage requirements, metadata management, and data lineage. The two latter ones are a crucial factor for the

productivity of a data scientist. For building accurate models and to asses the quality of data, it is necessary to know where the data comes from and how the specific attributes are defined. Especially for open data, attribute names are often cryptic and only known to certain domain experts but not necessarily known by a data scientist. For that reason, data science teams usually maintain codebooks and catalogs describing schema information. However, it is almost impossible to automatically trace the provenance on an attribute level and inherit meta information from the codebooks when an ETL script manually builds the dataset.
Midas facilitates the usage of open data by providing a centrally shared and virtualized data hub to data science teams. Additionally, it implements a novel approach for virtualizing datasets that allows sophisticated data lineage on attribute level and the inheritance of attribute metadata using its graph-based dataset representations.

3 GRAPH-BASED DATA VIRTUALIZATION

A complete definition of the graph representation would go beyond the scope of this abstract. However, in Midas, a virtual dataset is a view on one or multiple datasets defined by a SQL statement. Technically, Midas implements a rooted graph-based approach to represent these views.

Each dataset D consists of a name N, a list of arbitrary metadata objects $META$ and a set of attribute graphs SAG:

$$D := (N, META, SAG)$$

The name N is an arbitrary string which is usually a reference to the name of a table, a file, or a collection. $META$ is a JSON document that contains arbitrary metadata for a dataset like a description or access rights.

The attribute graph $AG \in SAG$ represents the provenance and metadata of a particular attribute $a \in D$ and denotes as follows:

$$AG := (V, E)$$

The vertex V consists of a name N_a and a list of arbitrary metadata objects $META_a$:

$$V := (N_a, META_a)$$

The edges E denote operations on an attribute.

Example. : A data scientist defines a virtual dataset VD by creating a view that joins the two dataset D_1 with the attributes a_{11} and a_{12}, and D_2 with the attributes a_{21}, a_{22}, and a_{23} together. D_1 and D_2 are combined based on their common join key a_{11} and a_{21}, respectively. The SQL statement looks like the following:

```
CREATE VIEW VD AS
(SELECT a_1_1 as join_key, a_1_2, a_2_2, a_2_3
FROM D1, D2 WHERE D1.a_1_1 = D2.a_2_1)
```

Midas takes the incoming query and creates the attribute graphs for VD. Figure 2 depicts the set of attribute graphs for VD.

4 QUERY PERFORMANCE

Performance is a crucial factor in virtualized data environments. Therefore, Midas implements an adaptive columnar cache similar to the column caches in Apache Spark. The implementation is straight forward and the algorithm is as follows:

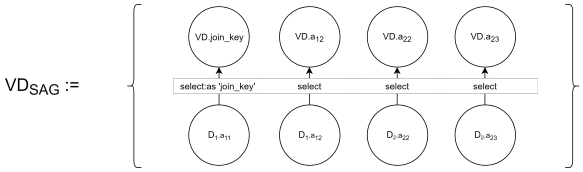

Figure 2: The set of attribute graphs for the virtual dataset VD

(1) Scan the referenced columns in the logical execution plan.
(2) Store the selected columns in a columnar format (Parquet).
(3) For all upcoming queries, do not query the actual source but use the Parquet reference files.

The cache can be used to improve performance or to offload production stores. Currently, the cache is triggered manually by the user depending on the individual use case and data size.

5 DEMO SCENARIO

At the SIGKDD we will demonstrate the current prototype of the Midas system and let the audience interactively use is. Therefore, we show how to add and manage arbitrary metadata to connected data stores and how to query them. Furthermore, we will show how to interact with the open data hub and how to create virtualized and sharable datasets. A demo video of the working system is online available at https://demo.midas.science/kdd

6 CONCLUSION

Currently, the Midas project is at an early research state. However, the results with the data science teams are promising and potentially lead to new workflows of working with virtualized data. However, the system should not be mistaken with a "one size fits all" approach. Midas is currently limited by the expressive power of SQL and the available data store adapters (MySQL, MongoDB, HBase, Hive, Amazon S3, MapR-DB, JSON - File, CSV, and Parquet). Our current focus is to look into methods that allows the system to efficiently leverage the native query capabilities of a connected source system.

REFERENCES

[1] Alon Halevy, Flip Korn, Natalya F. Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. 2016. Goods: Organizing Google's Datasets. *SIGMOD* (2016).
[2] Michael Hausenblas and Jacques Nadeau. 2013. Apache drill: interactive ad-hoc analysis at scale. *Big Data* 1, 2 (2013), 100–104.
[3] Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, Vol. 2. Citeseer, 1–49.
[4] Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, and Theo Vassilakis. 2010. Dremel: interactive analysis of web-scale datasets. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 330–339.
[5] Martin Traverso. 2013. Presto: Interacting with petabytes of data at Facebook. *Retrieved February* 4 (2013), 2014.