An SQL-like Query Tool for Data Anonymization and Outsourcing

Mohamad Nassar, Adel Al Rahal Al Orabi,
Marwan Doha, Bechara Al Bouna

Cyber SA 2015

Overview

- Data Anonymization and Outsourcing
  - Issues of interest
- Anonymization Techniques
- Related Work
- SQL Like Anonymization
  - Query Language
  - Software Architecture
- Experiments
- Conclusion

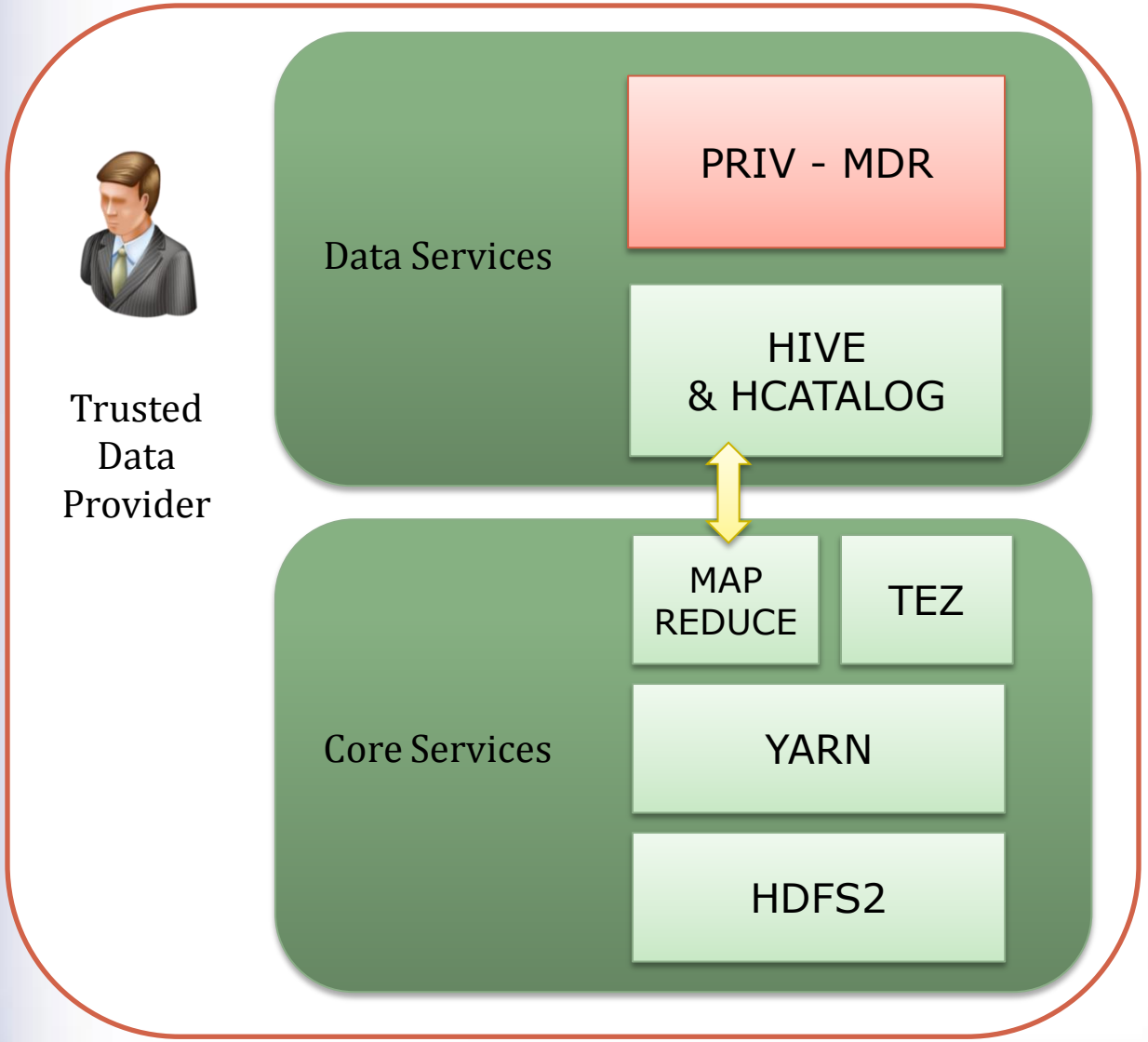
Introduction

- Cloud computing is on the rise
  - Financial and operational benefits
  - Risk mitigation
  - Speed of delivery
  - Etc.
- Hadoop is a key technology related to big data
  - Handles large datasets
  - Open source
  - Fast and reliable
  - Etc.

Issues of interest

- Hadoop lacks privacy preserving constraints for data outsourcing
  - Anonymity techniques not implemented
  - Limited SQL syntax
- Objectives
  - Design and develop a privacy-oriented module for data retrieval from Hadoop built on top of Hive
  - Extend SQL query language to include privacy preserving methods and parameters

High Level Architecture



Anonymization Techniques

- Consists of a set of groups, where each group is a set of tuples linked with a multi-set of sensitive values
- Based on the anonymization mechanism, each QI-group may correspond to either a set of quasi-identifer (QI) values or a single generalized QI value
- *Anatomy (Xiao et al., VLDB '06)*
  - separates QI-values from sensitive values into a quasi-identifier table (QIT) and a sensitive table (ST) that adhere to l-diversity

Anonymization Techniques

- *t*-closeness (Li et al., ICDE'07)
  - Distance between the distribution of sensitive attribute in a QI group and the distribution of the sensitive attribute in the overall table should be no more than a threshold t
- *Slicing (Li et al., TKDE'12)*
  - Preserves better data utility then generalization, protects against membership disclosure, by partitioning the dataset both vertically and horizontally
- *Safe Grouping (AL Bouna et al., DBSec 2013)*
  - Ensures that individual tuples are grouped in one and only one QI-group that is at the same time
    - l-diverse
    - Respects a minimum diversity for identity attribute values
    - Every subset T_u in QI_j are of equal number of tuples

Related Work

- Airavat Provides strong security and privacy guarantees for distributed computations on sensitive data
  - Based on Mandatory Access Control (as implemented by SELinux) and differential privacy
- Privacy Integrated Queries (PINQ)
  - Enforces differential privacy.
  - Provides analysts with a programming interface to uncurbed data through a SQL-like language

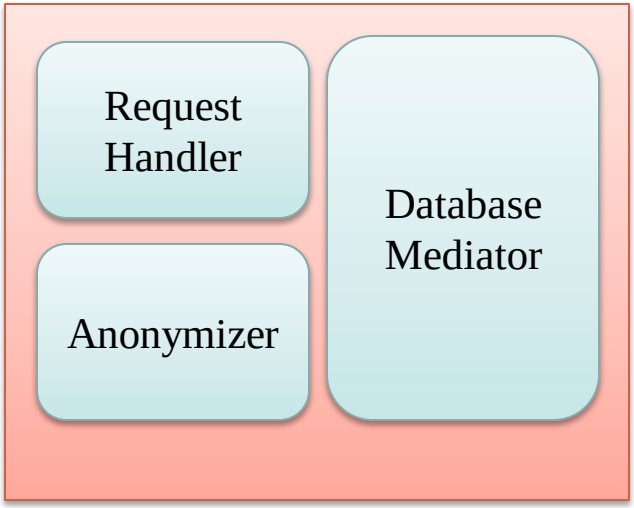
SQL Like Anonymization Query

- Extension of Hive Query Language
  - Anonymization methods
  - Privacy parameters
  - Quasi-identifying and sensitive values

SELECT <*expression*$_1$>, [<*expression*$_2$>]
FROM *table*$_1$, [*table*$_2$]
[**WHERE** <*condition*$_1$>[[AND |OR] <*condition*$_2$>...]]
[**ORDER BY** <*column_name*>[ASC |DESC]]
USING <*anonymization_technique*$_1$> **ON** <*sensitive_attribute*$_1$>,
[<*sensitive_attribute*$_2$>]
WITH <*param*$_1$ =*value*$_1$> [**AND** <*param*$_2$ =*value*$_2$ >]

Privacy Mediator Software Architecture

- Request Handler
- Anonymizer
- Database Mediator



Basic Components

- Request Handler
  - Interprets user's request
    - Retrieve database metadata
    - User's sql query
- Database Mediator
  - Parses the query retrieved from the Request handler to load the proper DBMS connector (SQL, MYSQL, and HIVE)
  - Communicates with the DBMS to retrieve the data and send it back to the anonymization component

Basic Components

- Anonymizer
  - The Anonymization component has three main responsibilities:
    - Retrieves the data that needs to be anonymized from the Data Base Mediator
    - Applies the appropriate Anonymization technique on the sensitive attributes defined in the query parsed by the Request Handler
    - Sends back the anonymized data to the Request Handler.

Experiments

- To study the running time of our proposed query we generate from 1 to 1000 of two types of queries.

```
1) SELECT COUNT (*) FROM adult WHERE attribute1 = value1
and attribute2= value2;
2) SELECT COUNT (*) FROM adult WHERE attribute1 = value1
and attribute2= value2 anonymize using anatomy on
<sensitive_arrtibute> with l=5;
```

- Comparing the running time for the set of queries
  - The results of our preliminary evaluation suggests a reasonable overhead of approximately 68 percent

Conclusion

- Multi purpose anonymization and data outsourcing technique
  - Compatible query language for data anonymization
  - Handles several anonymization techniques
  - Extensible for many databases
- Perspectives and future work
  - Include differential privacy
  - Enable query history evaluation
  - Handle privacy breach in sequential releases

THANK YOU