Improving Security by Using a Database Management System for Integrated Statistical Data Analysis

Vadym Khatsanovskyy[+], Jan-Eric Litton, Ruslan Fomkin[*]
Department of Medical Epidemiology and Biostatistics, Karolinska Institutet
Box 281, SE-171 77 Stockholm, Sweden

Vadym.Khatsanovskyy@gmail.com, Jan-Eric.Litton@ki.se, Ruslan.Fomkin@gmail.com

ABSTRACT

International research collaborations access and integrate data collected in different countries. For different reasons, e.g., legislation, data owners need to control who has access to and how their data are analyzed. The analysis of data is performed in statistical software, which is usually called on top of a data management system, e.g., a database management system (DBMS). Therefore access to data is controlled by the DBMS, while statistical analyses are usually controlled by another system. To improve security we propose a novel architecture for executing statistical analysis on data stored in a DBMS. In the proposed architecture the statistical software is called from a DBMS. The architecture allows control of both data retrieval and statistical data analysis from one system, i.e., DBMS. We implemented a prototype for executing analysis programs by calling statistical software SAS from a relational DBMS IBM DB2 over data stored in DB2 database. This paper describes the proposed architecture and the implemented prototype.

1. INTRODUCTION

For research in Medical Epidemiology the ability to integrate data from different sources is very important. During data integration data from different independent data owners are transferred to one place, where the integration is performed. The data owners might be obligated to keep control by whom and how their data are analyzed. This requires analyzing integrated data remotely in a controlled way on a remote analysis server, since controlling analyses of integrated data transferred to researchers' computers is unrealistic. Furthermore, disease-specific projects have limited resources for building their remote infrastructures over integrated data.

A number of established infrastructures, e.g., [1-4], provide remote protected statistical analyses of data from databases. All of them are based on architectures, in which accesses to statistical software and the execution of statistical analysis are controlled separately from access to data in the databases. The main drawback of the infrastructures is that it is expensive to maintain them: both build and run.

We propose a novel architecture for executing statistical programs over data integrated by and managed by a database management system (DBMS). In the proposed architecture the statistical software is called from the DBMS, which manages integrated data. The built-in privacy protection techniques of the DBMS are used to control both accesses to integrated data and calls to statistical software performing analysis of the data.

We implemented a first prototype SAQeL, Statistical Analysis from SQL. SAQeL helps to understand if it is difficult to interface the statistical software from a Relational DBMS (RDBMS). The prototype implements an interface to call the statistical software SAS [5] from RDBMS IBM DB2 [6]. It was tested on a study from epidemiological research on cervical cancer.

The main lesson that we learned from the prototype is that it is easy to implement a calling interface from an RDBMS, e.g., IBM DB2, to statistical software, e.g., SAS. Our next step is to implement a user-friendly interface to return the analysis results, which are often several tables and pictures, to researchers.

The rest of the paper is organized as following. Section 2 discusses related work. The proposed architecture is presented in Section 3. Section 4 describes the prototype: its design considerations, architecture, user interface and on-going work. Section 5 concludes the paper with summary and future work.

2. RELATED WORK

There exist a number of infrastructures, which allow remote analyses of data. Examples of these are LISSY from Luxembourg Income Study [1], the Danish system at Statistics Denmark [2], BioGrid Australia at Melbourne Health [3], MONA at Statistics Sweden [4], and PPA at CSIRO, Australia [7]. Each one implements a remote analysis server, where statistical analyses of data are performed on users' requests. On some of them [2-4] researchers access the remote analysis servers directly and work with statistical software interactively. The remote analysis servers implement authentication of researchers, authorize researchers to access statistical software and other tools on the server, and audit researchers' activities. Other systems [1] require researchers to submit their analysis programs through submission systems and do not provide direct access to the remote analysis servers. Then the executions of analysis programs are performed in batch.

Most of the systems [1, 2, 4, 7] analyze data, which are extracted from original databases and stored in files. Therefore, file systems of the remote analysis servers perform authorization of accesses to data files. Some systems [3] allow statistical programs to access DBMSs and request extraction of the data directly from the

PAIS 2011, March 25, 2011, Uppsala, Sweden.
Copyright 2011 ACM 978-1-4503-0611-9…$10.00.

[+] Current affiliation of Vadym Khatsanovskyy is Klarna AB.

[*] Current affiliation of Ruslan Fomkin is Starcounter AB.

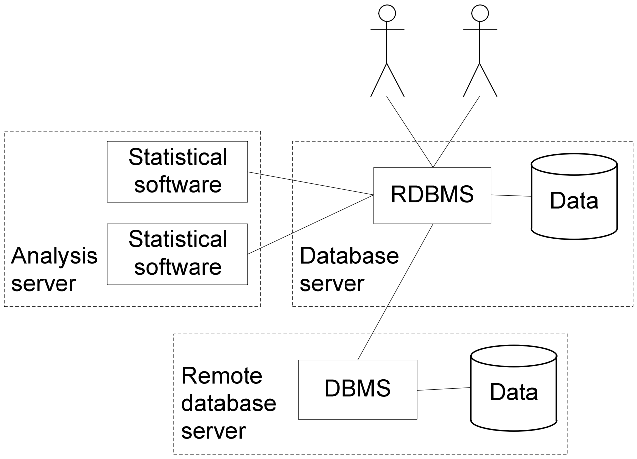

Figure 1. Proposed architecture.

databases. Thus, strong privacy protection mechanisms of DBMSs are utilized. Still BioGrid Australia [3] provides ability to store extracted data locally in the remote analysis server, thus file system is involved in authorization process.

Building infrastructures based on remote analysis servers [1-4, 7] requires significant resources.

A project exists which implements an interface to call analysis programs from a DBMS. LeSelect [9] extends a database query language SQL with ability to call image analysis programs. SQL is used in LeSelect to provide transparent and efficient ways to execute different image analysis algorithms over images in a distributed collaborative environment. The result of an analysis is a single table. In contrast, the focus of our work is on statistical analysis, which results in several tables and graphs.

A number of projects implement interfaces to call certain external statistical functions, e.g., MECHAMOS [10]. Such solutions require enormous effort in order to implement the interface for the many functions of the statistical software. Furthermore, scientists will miss the preprocessing abilities of the statistical software.

3. ARCHITECTURE PROPOSAL

We propose to call statistical software from an RDBMS, which manages the data. Our proposed architecture is presented in Figure 1. Researchers use the same RDBMS interface to perform aggregated queries, create views, or execute statistical analysis. In the figure, data integration of local data with remote data is performed by the RDBMS on the database server. For example, federated databases are used for data integration in project CODIR [11] and in BioGrid Australia [3].

Before researchers can use the infrastructure, they and their projects have to be approved by a committee. Then their permissions are implemented in the infrastructure by database administrators (DBAs) through giving necessary privileges to database roles and users.

In the architecture researchers first submit queries to the RBDMS on the database server through a client interface. The queries either perform simple analyses, which produce aggregated data as a result of processing data from tables or views of the integrated database, or the queries create views, which are defined in terms of other views or tables of the integrated database. The queries are executed only if the researchers are authorized to run them. Then the researchers create their statistical programs, which are registered in the database server. The statistical programs are defined in term of views, which were created earlier. The researchers request executions of their statistical programs. If the researchers are authorized to call the statistical programs, the RDBMS submits the programs to the corresponding statistical software for the execution on the analysis server. During the execution the statistical software access data from the views defined in the RDBMS. The data are transferred to the statistical software if the researchers are authorized. After the statistical programs are executed the results of statistical analyses are transferred back to the researchers. All authorizations are build-in in RDBMSs and performed automatically.

4. THE PROTOTYPE

We investigated our proposed architecture by implementing a prototype, SAQeL. SAQeL extends DB2 [6] with the ability to call SAS [5] and execute SAS programs over the data stored in DB2. This section describes the current implementation of SAQeL: its design considerations, its architecture, and an example of running statistical analysis in it.

4.1 Design Considerations

SAQeL design is based on simplicity and extensibility. One question for consideration is in which way to call SAS from DB2. One way is to use SAS Integration Technologies to establish a SAS server and then communicate with the SAS Server from DB2. The server approach requires SAS modules from SAS business solution in addition to SAS Foundation modules such as SAS Base. This increases the cost and complicates the construction of the prototype. Another drawback of the SAS server approach is the inability to extend this solution to other statistical software, which do not implement their own server solutions, e.g., Stata [12]. Another way to call SAS is through system calls to a SAS client, SAS Base, and to execute SAS programs in batch mode. This requires preparing the SAS programs in files together with the configuration parameters. This solution is cheaper and simpler than using the SAS server. Furthermore, other statistical software, e.g., R [13] and Stata, can be called in batch modes. Therefore, SAQeL calls a SAS client in a batch mode.

Another question for consideration is how to transfer data from DB2 to SAS. There are mainly two ways: by extracting data into files and then accessing the files from a SAS program, or by establishing a direct connection from SAS to DB2 through, e.g., ODBC interface [14]. Extracting data into files is less secure and has worse performance than connecting DB2 directly from SAS. Furthermore, other statistical software, e.g., Stata and R, also implement interfaces to connect to RDBMSs directly. Therefore, in SAQeL data are transferred to SAS for execution by establishing a direct connection from SAS to DB2 at run time.

A standard way of extending RDBMSs functionalities, e.g., of DB2, is by developing and deploying external routines. DB2 puts the following restrictions on external routines: external routines cannot create new threads and processes, and new connections cannot be established from processes running external routines. Therefore, SAQeL implements a listening process, *SAQeL Analysis Service*, to receive requests from an external routine and to call SAS in a batch mode. Thus SAS is able to create new connections to DB2 to retrieve data.

The final design decision is to develop an external stored procedure in Java for calling SAQeL Analysis Service from DB2,

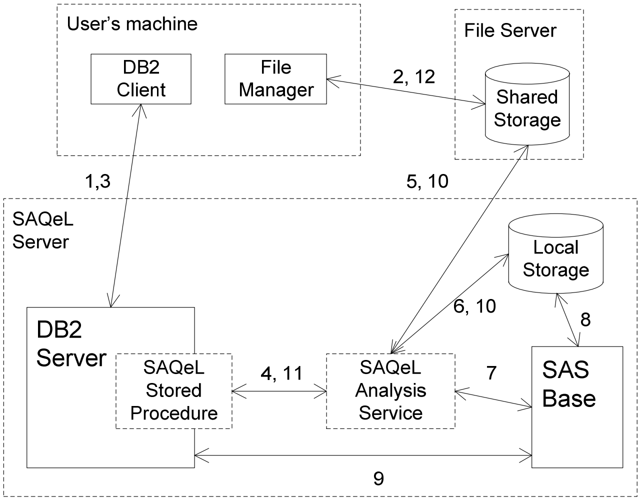

Figure 2. SAQeL architecture.

since Java stored procedures suit us the best among other possible external routines and implementation languages.

4.2 SAQeL Architecture

SAQeL architecture is presented in Figure 2. Both *DB2 Server* and *SAS Base* run on the same server, *SAQeL Server*. *DB2 Server* is extended with an external stored procedure, *SAQeL Stored Procedure*, implemented in Java. *SAQeL Stored Procedure* is deployed on *DB2 Server* and can be invoked using SQL commands. The primary goal of *SAQeL Stored Procedure* is transmitting parameters of a statistical analysis execution from a user to *SAQeL Analysis Service* and returning the status of the statistical analysis execution back to the user. *SAQeL Analysis Service* is a Java process that runs permanently. It executes a SAS program in a batch mode, i.e. without the user's direct interaction with *SAS Base*. For this purpose, *SAQeL Analysis Service* prepares configuration parameters for the batch execution and manages the statistical analysis results. *SAQeL Analysis Service* receives statistical analysis requests from *SAQeL Stored Procedure* via a TCP/IP connection socket.

Users communicate with *DB2 Client* and *File Manager* to submit their analysis for execution on *SAQeL Server*. *File Manager* is used to store SAS programs on a storage shared between *User's Machine* and *SAQeL Server*, and to access the analysis results from there. *DB2 Client* is used to issue SQL queries, which perform data analysis, and to access information about the successful execution of the analysis or the occurrence of an error.

The process of an analysis is as follows. Through *DB2 Client* a user (1) issues a query to *DB2 server* in order to create views to be used in the SAS programs. *DB2 Server* authorizes the user's query and creates the views. Then a SAS program is created by the user and (2) saved on *File Server*. The user makes a request (3) to perform a statistical analysis by calling *SAQeL Stored Procedure*, and *DB2 Server* authorizes the call. As input parameters of this procedure, the user specifies the name and location of the SAS program, credentials and desired location for the storage of the analysis results. *SAQeL Stored Procedure* (4) transforms the input parameters into a message, initiates a socket connection with *SAQeL Analysis Service* and transfers the message. *SAQeL Analysis Service* determines the authenticity of the request, by checking its compliance with the internal protocol. In the case of a positive result, it extracts information from the received message, (5) reads the user's original SAS program from *File Server* and generates a SAS program for execution by adding supplementary configuration code to the original SAS program. The generated SAS program (6) is saved in a temporary file at *Local Storage*. After that, *SAQeL Analysis Service* (7) makes a system call to the operating system to run *SAS Base* in a batch mode. *SAS Base* (8) reads the file with the SAS program from *Local Storage* and executes it. During the execution, it (9) creates a connection to *DB2 Server* and accesses the data if the user is authorized. When the statistical analysis is completed, control is returned back to *SAQeL Analysis Service*. *SAQeL Analysis Service* (10) moves the analysis results from *Local Storage* to *File Server* if the execution was successful. Then *SAQeL Analysis Service* generates an output message with the general information about the execution results or the error, and (11) sends the message to *SAQeL Stored Procedure*.

SAQeL Stored Procedure presents the message about completion to the user in *DB2 client*. Finally, the user (12) accesses the analysis results using *File Manager*.

4.3 An Example of Running Analysis

This section presents an example of doing survival analysis for a study from epidemiological research on cervical cancer. First a researcher creates views, which are going to be accessed in a SAS program. For example, view *survmig.pc_cohort* is created by:

```
CREATE VIEW survmig.pc_cohort AS
SELECT lopnr, diagyr
FROM (SELECT lopnr,
               MIN(diag_cancer_yr) AS diagyr
      FROM cerv_db.cancer
      WHERE icd_7='171' AND
            malign_benign IS NULL
      GROUP BY lopnr)
WHERE diagyr BETWEEN 1960 AND 2005
```

Then the user writes a SAS program in terms of a view *survmig.pc_cohort_duration* as following:

```
PROC LIFETEST DATA=survmig.pc_cohort_duration
     METHOD=km PLOTS=(s) NOCENS;
  TIME years*censor(1);
  STRATA birth_place;
RUN;
```

The SAS program is stored in *Shared Storage*. Finally, the user calls *SAQeL Stored Procedure* with following parameters: user name and password for DB2 connection form SAS, name of the SAS program, and input and output paths:

```
CALL SAQeL (‘john’, ‘abc123’, ‘mysasprogram’,
            ‘S:\john\analysis\Programs\’,
            ‘S:\john\analysis\Results\’);
```

After the analysis execution the user receives a message in *DB2 client*, which notifies that the analysis has been completed and the results are available in the specified folder.

4.4 On-going Work

In our approach we do not restrict researchers on which SAS procedures they use. Therefore, we continue testing SAQeL with different examples of SAS programs. We are also going to incorporate other statistical packages such as Stata and R.

Another on-going task is to improve the user interface. First of all we are improving the interface of *SAQeL Stored Procedure* to eliminate the need in the user's name and password and to report more information about execution results. Furthermore, we are investigating how to return execution results directly through DB2 client and how to avoid using a shared storage for this. Similarly we are looking into deploying SAS programs through DB2 client.

5. SUMMARY AND FUTURE WORK

We proposed novel architecture for executing statistical analyses on integrated data in a secure way. Our proposal utilizes strong privacy protection of RDBMSs. This should enable development of secure integrated infrastructures at a reasonable cost. Our first prototype SAQeL successfully demonstrates interface between DB2 and SAS. It was implemented in 10 weeks by a Bachelor student in computer science. SAQeL was demonstrated to our colleagues, who are doing research in epidemiology. They would be glad to use the system for remote analysis of research data.

Our future work is to combine SAQeL with our work on federation solution for integrating data from national population registers in Sweden as it is proposed in the project on Cross-Organizational Infrastructure for register-based Research (CODIR) [11]. CODIR aims to eliminate the disclosure of any personal information. Therefore, we will implement monitoring of SQL queries, analysis program codes and execution results for disclosing only aggregated results.

6. ACKNOWLEDGEMENTS

The work has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 200754 (the GEN2PHEN project).

We thank Tore Risch, Magnus Stenbeck and Pär Sparén, for developing the proposed architecture and for the discussions. We are grateful to Pär Sparén and his research group, in particular Fatima Azerkan and Jonas Hällgren, for providing their epidemiological study as the use case.

7. REFERENCES

[1] Barry, S., Marc, C.: Remote access systems for statistical analysis of microdata. Statistics and Computing 13 (2003) 381-389. See also http://www.lisproject.org.

[2] Borchsenius, L.: New developments in the Danish system for access to micro data. Monographs of official statistics (2005) 13-20.

[3] Hibbert, M., Gibbs, P., O'Brien, T., Colman, P., Merriel, R., Rafael, N., Georgeff, M.: The Molecular Medicine Informatics Model (MMIM). Stud Health Technol Inform 126 (2007) 77-86. See also http://www.biogrid.org.au.

[4] Hjelm, C.G.: MONA-Microdata ON-Line access at Statistics Sweden. Monographs of official statistics (2005) 21-28. See also http://www.scb.se.

[5] SAS Software, http://www.sas.com.

[6] IBM DB2 Software, http://www.ibm.com/software/data/db2/.

[7] Sparks, R., Carter, C., Donnelly, J.B., O'Keefe, C.M., Duncan, J., Keighley, T., McAullay, D.: Remote access methods for exploratory data analysis and statistical modelling: Privacy-Preserving Analytics. Comput Methods Programs Biomed 91 (2008) 208-222.

[8] Haas, L.M., Lin, E.T., Roth, M.A.: Data integration through database federation. IBM Syst. J. 41 (2002) 578-596.

[9] Luc, B., Fran, oise, F., Fabio, P., Patrick, V.: Processing Queries with Expensive Functions and Large Objects in Distributed Mediator Systems. Proceedings of the 17th International Conference on Data Engineering. IEEE Computer Society (2001) 91-98.

[10] Tisell, C., Orsborn, K.: A system for multibody analysis based on object-relational database technology. Advances in Engineering Software 31 (2000) 971-984.

[11] Ruslan, F., Magnus, S., Jan-Eric, L.: Federated Databases as a Basis for Infrastructure Supporting Epidemiological Research. Proceedings of the 2009 20th International Workshop on Database and Expert Systems Application. IEEE Computer Society (2009) 313-317.

[12] Stata: Data Analysis and Statistical Software, http://www.stata.com.

[13] The R Project for Statistical Computing, http://www.r-project.org.

[14] Open Database Connectivity Overview, http://support.microsoft.com/kb/110093.