

Feuerhake, Jörg

Working Paper

XQS/MD*Crypt as a means of education and computation

SFB 373 Discussion Paper, No. 2002,46

Provided in Cooperation with:

Collaborative Research Center 373: Quantification and Simulation of Economic Processes,
Humboldt University Berlin

Suggested Citation: Feuerhake, Jörg (2002) : XQS/MD*Crypt as a means of education and computation, SFB 373 Discussion Paper, No. 2002,46, Humboldt University of Berlin, Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes, Berlin, <https://nbn-resolving.de/urn:nbn:de:kobv:11-10049046>

This Version is available at:

<https://hdl.handle.net/10419/65342>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

XQS/MD*Crypt as a Means of Education and Computation

Jörg Feuerhake

Institut für Statistik und Ökonometrie, wirtschaftswissenschaftliche Fakultät der Humboldt–Universität zu Berlin, Spandauerstr. 1, 10178 Berlin, Deutschland

Abstract. This paper introduces MD*Crypt as a means of distributing statistical methods and computing power. This paper explores MD*Crypt regarding topics of education. The role of MD*Crypt in projects associated with teaching statistics is examined. In an outlook, the potential of the MD*Serv/MD*Crypt architecture in frameworks of distributed computing and method providing is mentioned.

Keywords. education, XQS, MD*Crypt, distributed computing

1 Introduction

Nowadays statistics always goes with processing large amounts of data. The use of strong computers is absolutely necessary to handle this data. All considerations need to take this into account. When teaching statistics one faces particular demands. In the attempt to match those demands, it is necessary to consider how they can be met. This paper seeks to describe how computational power and a variety of methods are made available to students via the Internet. Distributing methods and computing power will be referred to as "distributed computing".

Since we are concerned with teaching statistics at the graduate level, we needed to provide students with a strong tool for computational statistics. Or more precisely, we need to provide computing power, as well as a wide selection of implemented statistical methods. The software to use in this case is XploRe, because of its well selected pool of statistical methods. This software has grown to be a strong standalone statistical environment. XploRe's developed language is highly applicable to meet student's needs.

With the emergence of Web technologies it became possible to acquire more students in a convenient way. It will be shown in this paper how attracting more students is achieved and the relationship to MD*Crypt will be discussed in further detail. Furthermore, particular features of the chosen architecture are discussed and evaluated.

Later, other approaches to deploy statistical methods and computing power are evaluated. The Rweb project is reviewed in more depth showing its quite remarkable approach to solving statistical problems.

Starting from the status quo of development and redefining the term "distributed computing", ways of improvement available to the described architecture will then be discussed. If the MD*Crypt architecture is reviewed middleware centered, one can see these middlewares as core of a peer2peer web. The implications of this type of architecture will be described at the end of this paper.

2 MD*Crypt as an example

2.1 The Goal: Accessibility

As accessibility of information is a crucial question in teaching, students must be provided with the latest information about the issues they are concerned with. With 'paper-based' information, that task was not trivial. When the Internet emerged it

became possible to publish new knowledge in almost an instant. Nonetheless, in teaching statistics, there is more than just published information. Furthermore, it is important to provide computing power and statistical algorithms, as well as the possibility to explore taught material and topics interactively in the classroom. To reach this goal, it becomes important to go beyond the means of pure Hyper-Text-Transportation-Protocol. In order to perform statistical computations, it is necessary to ensure that the computing machine is capable of actually computing at a certain level. One could achieve this by making sure that computing is done on a particular server machine. Client/Server computing appears to be relevant for these goals to be achieved.

In a subsequent step, it becomes crucial to provide an easy way to access server capacity. This has to be done in a way that can easily be carried out on a wide range of platforms. This requirement leads to the architecture described below.

2.2 Resulting Architecture

XploRe provides a large and well developed pool of statistical methods, as well as the ability to solve complex numerical problems. Hence XploRe becomes a promising candidate for serving as a statistical engine in teaching. Further, the methods implemented are well documented and the documentation is easy to obtain, or so called "browser enabled". The XploRe server implementation communicates on the stdio stream of it's underlying platform. Since this stdio communication is not appropriate for TCP/IP, it is important to have a middleware, providing the methods to connect via TCP/IP.

Using XploRe as a server for method providing and computational tasks, it is important to choose a suitable environment to implement the client. What technology guarantees most widespread platform access? Since Java's main argument was "write once, run anywhere", that seemed to be the language to implement the XQS communication package. Once using Java, there was only a short step made in sticking to using Java proposed approaches to programming.

Returning to the XQS, it suffices to state that MD*Serv does implement the communication between XQS and XQC. Since XQS is communicating on a stdio stream, it additionally needs to read from the TCP/IP stream and then write the same information to the stdio stream of the listening XQS. For the server to client communication, the same mechanism applies in reverse. Within the XQS/XQC architecture, MD*Serv captures a central role just as Figure 1 suggests.

The future client needs to implement the XQS Listener. By doing this the client is forced to implement the method *handleServerReply()*. This method is meant to handle incoming results from the XQS. It is crucial, that the handling of the result is left to the programmer of the future client, who has the knowledge of how to implement the representation of the result, such as graphic representation, most efficiently. In conclusion, we finally face an architecture as seen in Figure 1 according to the graphic. While MD*Serv and MD*Crypt handle all issues of TCP/IP communication, the XQS provides methods, data and computing power. The actual client needs to implement the user interface and handles the XQS results according to the technical environment on the client's machine.

The entire architecture relies on the ability of the future client programmer to interpret the XQS results properly and efficiently. That leads directly to the need for proper documentation and support. Another point that pushes Java to be the language of choice for MD*Crypt is the Javadoc tool which enables MD*Crypt to provide class documentation in an instant. Though this would not be sufficient to explain MD*Crypt to a third party developer it is a big step towards third party support.

For some projects, the whole architecture was brought to a single machine. So, with stronger processors in widespread use, the XQS/XQC-architecture can be de-

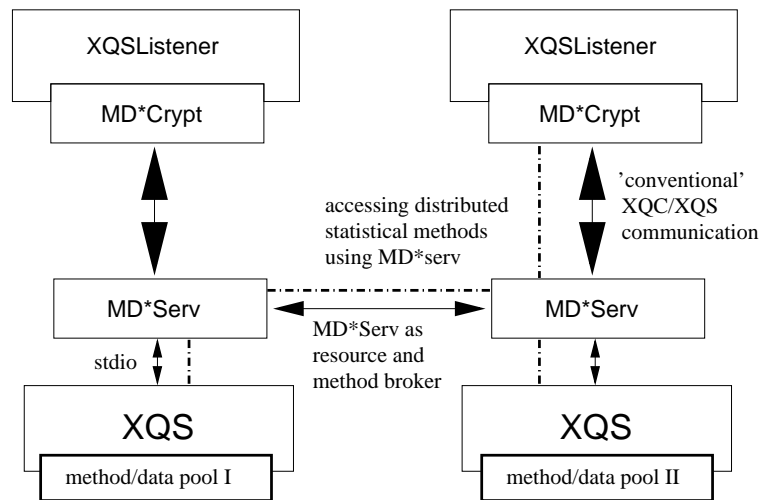


Fig. 1. the role of MD*Serv

ployed on closed single media, such as an electronic book. Creating media for teaching statistics on a XQS/XQC basis means writing net-based means of teaching as well as media ready for single machine use in the same step.

3 Related Projects

MD*Crypt is, as we have seen, a package to deploy computing power and statistical methods platform independent. Projects utilizing MD*Crypt to either use XploRe's statistical methods or reach accessibility via the Internet will now be introduced. This will be explained with regards to issues of education.

3.1 ReX

Recognizing the fact that Microsoft Excel is the most common software to analyze and process data, it is obvious to bring the methods one wants to deploy into the Excel interface. This can be done by offering an Excel add-in. Since it is possible to generate a com dll from Java code, MD*Crypt can be utilized to act as an Excel add-in. The Excel user is then able to use methods provided by XploRe.

Knowledgeable students will very likely be able to work with the Excel look and feel. Furthermore, students can concentrate on the contents taught if they do not need to familiarize themselves with an unknown user interface, ReX is a promising candidate as a means of teaching statistics. With the possibility to setup the XQS/XQC architecture on a single machine, it is possible to use the XQS/XQC architecture even in environments with impermanent Internet access. In situations, such as reviewing lectures at home, this becomes a major advantage.

3.2 XQC

XploRe's XQC provides a user interface very similar to XploRe's standalone user interface. Because one can run it as a Java applet, it becomes available on almost any machine connected to the Web. Students who are used to working within XploRe's

user interface find it quite simple to use the XQC. Note that this is possible without having XploRe installed on the client machine. Therefore one can teach or study in environments poorly arranged for statistical teaching purposes.

3.3 e-books

A point that has not been emphasized so far is that teaching statistics should always come with the possibility to explore certain issues taught interactively. When using new formats of publishing contents, the teacher should go beyond presenting just the functionality of a book. One needs to find ways to extend the features of electronically enhanced books towards interactive examples. Here with MD*Crypt and the features of XploRe's XQC it is then possible to present interactive examples for statistical content just incorporated into an electronic book. There are already quite a developed amount of e-books using this feature.

These e-books present their content as a browser enabled e-book and provide the possibility to calculate examples, change their parameters or graphic representation all via the XQC.

4 Known Problems

There are of course some challenges to overcome with the MD*Crypt approach. Since MD*Crypt is a Java package, it relies on Java runtime on the client machine. If it is utilized in an applet, as it is done within XploRe's XQC, it is likely that on the client machine an installation effort has to be carried out. Within some environments this becomes a major drawback.

Another drawback is related to Java's applet technology. Since applets are regarded as potential security problems some crucial functionality is not present. For example confined server connecting and disabled copy/paste functionality are problems as long as applets are not signed by a trusted institution. Not every third-party programmer has access to assets necessary to get a trusted signature for his client. Since third party programmers are crucial for MD*Crypt this remains to be a problem. Pure html interfaces do not face these boundaries in functionality. See the Rweb section for further consideration.

5 Layout and Data-processing

As it was shown, with MD*Crypt it becomes possible to access all implemented algorithms provided by XploRe and handle the provided results via TCP/IP. This is a major improvement relative to XploRe's standalone version. The next section will show that other approaches can achieve this goal as well. They may even do this with less effort, like client programming, for the user. However, a main achievement is MD*Crypt's usability, adjusted to the use-case. With changing environments and working scenarios different clients with different user interfaces may connect XploRe Servers via MD*Crypt. This division of layout from data-processing does provide a lot of scalability. One may use XploRe's algorithms in a lot more occasions than XploRe's standalone could cover. Furthermore the clients may adjust to almost any use-case.

6 Different Approaches: Rweb

One remarkable project in deploying statistical methods via the Internet is Rweb. Rweb implements information transport with a common gateway interface. One great advantage to Rweb is its pure html interface. Since it does not rely on Java or a different runtime as its core environment, it runs without the installation of a plugin. Rweb pays a large price for this convenience. Rweb cannot change its interface according to the underlying environment or as a reaction to different tasks.

Rweb's second advantage compared with the MD*Crypt approach is that it does not need further programming to implement a client. The browser runs again, but it is not adjustable to perform certain tasks.

Other Web-based data analysis approaches do not have the general approach which MD*Crypt or Rweb try to implement. In this context, one should mention Data Analysis Lab, WebStats and Statlet. The statistical methods in these programs are implemented as applets and run in Java enabled browsers. Unfortunately, they implement a very small range of methods and avoid Client/Server approaches. Client/Server architectures are most important to make sure that large scale problems can be solved as well.

7 Development Paths

The term distributed computing encompasses just more than making available methods and capacities geographically independent. Distributed computing means further on sharing capacities and methods throughout the Internet. Please see Figure 2 as an outline of a network of MD*Serv/XQS running on different machines:

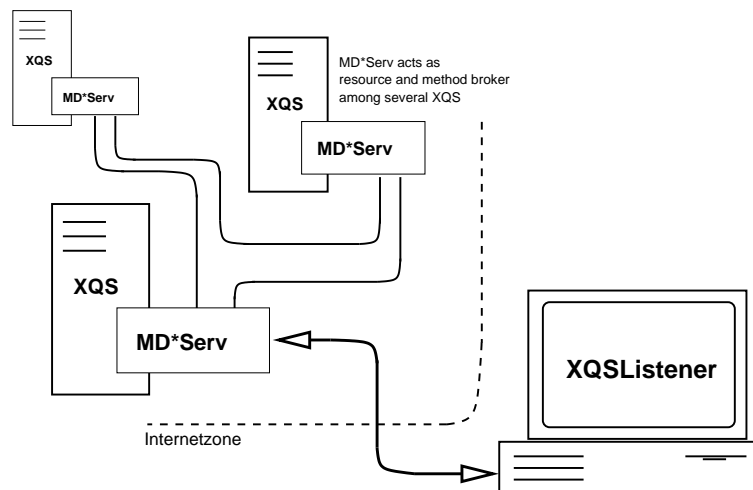


Fig. 2. XQS/MD*Serv systems connected

In returning to the described architecture, one can begin with the stated central role of MD*Serv. MD*Serv then, is easy to recognize as the pivotal point of a system of XQS's. MD*Serv would then be capable of providing functionality often referred to as distributed computing. MD*Serv is by now prepared to contact other instances of MD*Serv. When enabled to hold and pass information concerning its running servers, the servers' computing capacity and the methods available on their system it is possible to extend MD*Serv to serve as a resource and method broker. A XQC would then be able to contact one MD*Serv on a particular URL:Port and MD*Serv would dispatch this client's request. The result could be a grid of homogeneously working XQS and even more importantly, statistical methods available on one server

could be deployed to the client even if it connected to a different machine. With pure applet technology and in the absence of a middleware this architecture remains impossible. Please refer to Figure 2 to evaluate the impact.

The main task to be accomplished by MD*Serv will be resource broking. Resource broking would mean passing providing and evaluating information about XQS's availability and about the methods these XQS's could provide to solve problems. Further, MD*Crypt/MD*Serv now provide communication patterns, such as graphic contexts and complex numeric arrays, ready for the use in statistical environments. This does not force MD*Serv to only connect to XploRe's servers. Provided, MD*Serv would map command structures of other languages and the engines could be connected, any of the mapped statistical systems could act as a XQS. Refer to the $\hat{\Omega}$ project's Java interface to find Java classes that provide methods to contact R. MD*Serv is not only able to dispatch requests among XploRe's servers, but could also access methods implemented within other statistical environments.

The improvements of distributed computing for students and teachers are straightforward. A grid is not as prone to failure as much as single servers are. Methods meant for distribution do not need to be stored on every server system. Method libraries can be shared as well as computing capacities.

8 Concluding Remarks

With MD*Crypt, a package is provided to deploy statistical methods and computing power to solve statistical problems. Programming skills provided, MD*Crypt is highly adjustable to a wide range of environments. Through this package a general approach in providing statistics can be utilized.

In developing MD*Crypt, it will be important to improve the underlying protocol. Furthermore, potentials of the achieved architecture need to be pushed forward to achieve new functionality and foolproof the recent architecture.

A different approach to the deployment of a large amount of statistical methods is Rweb, which follows a cgi approach in request handling. It is a project worth noticing for its achievements in accessibility via the Internet.

The possibility of solving large scale statistical problems is one goal when teaching statistics. First steps are made to achieve wide accessibility to computing resources and statistical methods.

Resources on the Web

MD*Crypt, <http://www.md-crypt.com>

$\hat{\Omega}$ project's Java interface, <http://www.omegahat.org/RSJava/>

ReX MS Excel add-in, <http://www.md-rex.com>

Rweb html-interface, <http://www.math.montana.edu/Rweb/>

XQC XploRe's Java-interface, <http://www.xploRe-stat.de/java/java.html>

References

- Aydinli ,G. (2002) 'Netbased Spreadsheets in Quantitative Finance'. In: "*Applied Quantitative Finance*", Berlin: Springer Verlag.
- Aydinli ,G. , Härdle, W. , Kleinow, T. and Sofyan, H. (forthcoming 2002) 'Linking Office Applications with XploRe'. In: *Computational Statistics; special edition*, Heidelberg: Physica-Verlag.
- Kleinow, T. and Lehmann, H. (forthcoming 2002) 'Client/Server based Statistical Computing'. In: *Computational Statistics*, Heidelberg: Physica-Verlag.
- Manzur, M.R. 'GridSim: A Toolkit for the Modeling and Simulation of Global Grids'. available on the web:
<http://www.csse.monash.edu.au/rajkumar/papers/gridsim.pdf>;