A FRAMEWORK FOR ON-DEMAND INTEGRATION OF ENTERPRISE DATA SOURCES

Matti Heikkurinen[1], Tapio Niemi[2], Marko Niinimäki[3], and Vesa Sivunen[3]

[1] *CERN IT Division, Openlab, CH-1211 Geneva, Switzerland*
[2] *Department of Computer Sciences, FIN-33014 University of Tampere, Finland*
[3] *Helsinki Institute of Physics, CERN Offices, CH-1211 Geneva, Switzerland*

Keywords: Grid, OLAP, Data warehousing

Abstract: Deploying a data warehouse system in a company is usually an expensive and risky investement. Constructing a data warehouse is a large project that can take very long time and a company cannot know in advance exactly what benefits a data warehouse will offer. Thus, in many cases, data warehousing projects have either been abandoned or been shown to be at least partial failures.

We propose a new method by providing a platform to implement business intelligence systems on. The basic idea is to construct the analysis database (i.e. an OLAP cube) on demand and only include the data that is needed for the analysis at hand from the operational databases. In this way the data is always up-to-date, suitable for the current analysis, and some of the biggest risks associated with data warehouse systems can be avoided. In addition, external data can be included in the analysis. The computational costs related to the cube construction are likely to remain at acceptable level, since only the relevant part of the data for the current analysis is needed from operational databases.

We outline the use of Grid techologies in the implementation to offer a cost-effective way to harness enough computing power used on parallel processing and sufficient security infrastructure (GSI). To deal with heterogenous data sources the XML language with XSL transformations is applied.

1 INTRODUCTION

Data warehousing and OLAP are offered as solutions for analysing business data, e.g. (Mohania et al., 1999; Chaudhuri and Dayal, 1997). The motivation is to collect relevant data from different operational databases into a one large database or a collection of smaller databases (often called "data marts"). The collection is usually done periodically, for example once a week. The data warehouse software is used to answer queries about the aggregate data of the business transactions, not to process transactions themselves. Therefore, data warehouse systems are called OLAP (On-Line Analytical Processing) in contrast to OLTP (On-Line Transaction Processing). A common method is to use a multidimensional data model to represent the data warehouse data - in other words the data are represented as a multidimensional hyper cube where the coordinates can be based on variables such as time, department, product sold and so on.

Data warehouse systems are usually large and expensive investments, and tend to be inherently mission critical due to their close ties with the operational data management and top level decision making. Constructing a data warehouse is typically a large project that can easily take even a year, tying up resources from multiple levels of management (tactical and strategic) in addition to the IT personnel. As always in the case of large IT projects, it is difficult to predict in advance what benefits the system will offer to justify the huge cost of the deployment project. Some studies suggest that a third of all the information systems projects are abandoned before completion (Ewusi-Mensah, 1997) and over half of the completed ones are almost 200% over budget further. It is thus not surprising that in many cases data warehouse projects have not really offered what the company has believed. This illustrates the large risks involved in choosing this kind of approach to data management integration.

To mitigate the risks outline above we suggest a new method for business intelligence systems that would enable reaping most of the benefits of a traditional data warehouse without the strategic risks involved. The approach is based on applying the recent advances in the Grid technology development in

Heikkurinen M., Niemi T., Niinimäki M. and Sivunen V. (2004).
A FRAMEWORK FOR ON-DEMAND INTEGRATION OF ENTERPRISE DATA SOURCES.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 604-608
DOI: 10.5220/0002655806040608

a way that would allow efficient and modular integration of various legacy systems into a *"virtual data warehouse"*. This modular integration would also allow replacing these legacy systems independently of each other. This would make it possible to optimise the components in their local environment without limiting their relevance as sources of information for strategic decision making.

Some recent developments in the design and prototyping tools are addressing these problems by making it possible to centralize the information about the corporate data sources and prototype the system that would result from the use of the planned ETL (Extraction, Transformation, Loading) process. The proposed approach will push these ideas even further by making it possible to iterate these prototypes in quick succession with the real transactional data, and then put the final, "accepted" version into production. This is done by leveraging Grid Data access protocols, which make possible hiding the details of the local storage systems from the ETL process.

Another advantage of the use of the Grid technology comes from its ability to create "Virtual Organizations" from groups of entities that belong to different security domains. By mapping the local identities (user accounts, server certificates and other entities managed by the corporate security infrastructure) to Grid certificates, it is possible to integrate data from sources that are not controlled by a single entity. Data access and security features make the proposed approach especially interesting in cases like managing the information systems in recently merged companies - or establishing an extended enterprise type collaboration with strategic partners.

The basic idea behind the virtual data warehouse is to construct the analysis database (i.e. an OLAP cube) *on demand*, and only include the data that are needed for the analysis at hand. Since the data can be gathered directly from the operational data sources, it is always up-to-date and suitable for near-realtime analysis of emerging trends in the company and its environment. The most difficult problem on-demand construction of OLAP cubes is the extraction of data from operational databases. We offer the following three approaches to deal with the problem.

1. In most cases, only a small subset of data stored in OLTP systems is fetched to the OLAP cube. This is possible since we aim at building the OLAP cube to solve some specific problem. In the traditional data warehouse approach the OLAP cube is constructed for general purpose analyses.
2. Grid technologies facilitate dynamic allocation of computing power from larger resource pool for ETL processing. While it is likely that OLTP systems themselves are more efficient now than couple of years ago, the development does not in itself help ETL phase, since the OLTP systems are often legacy systems that will not be updated very often. Furthermore, the amount of data and the users of the OLTP system will usually increase in phase with the system capacity.
3. The selection of the relevant data for the analysis can be done remotely in the operational databases before shipping the data for analysis. Agent technology can partially perform the local data selection and aggregation to decrease the need of the processing power of the local database servers.

Business intelligence systems, like OLAP, have traditionally been limited to the data stored in the data warehouses or some other well-defined, structured databases. There can be cases where this predefined set of data sources is not enough, since the phenomenon under analysis can depend on something outside the scope of the company. For example, the oil price or the weather can have a remarkable effect on business through some complex cause and effect chain. If the rules that are used to test scenarios are limited to the data that is in the corporate data warehouse, the analysis cannot find all the possible explanations for a phenomenon. Virtual data warehouse methodology enables the user to include external data to the OLAP cube through Grid Data access.

The motivation to use Grid technologies in the implementation is related to the capacity of the Grid frameworks to provide enough secure computing and storage capacity on demand to handle much larger datasets than traditional systems with similar costs. This kind of use of parallel processing and shared computing resources requires a strong, universally accepted security infrastructure that is used to access the computing resources of external service provider (this type of Grid can be seen as a advanced "Utility Computing" solution). In addition to Grid technologies, we use XML language with XSL transformations for data source integration (The World Wide Web Consortium, 1999).

2 RELATED WORK

Zurek and Sinnwell (Zurek and Sinnwell, 1999) have studied how changes in companies should reflect to data warehouses. They have noticed that companies tend to change their organization often, necessitating the realignment of data warehouse schema quite frequently. Especially the required dimensions of data and their hierarchies can change even more frequently than the organization itself.

Sypherlink is one of the companies advertising more flexible prototyping and ETL tools for data warehousing systems (Sypherlink, 2003). This solution addresses some of the issues associated with frequent organizational changes.

Integrating data sources for OLAP using XML are studied e.g. Pedersen et al. (Pedersen et al., 2002). Using Grid computing in OLAP cube construction are studied, for example, by Niemi et al. (Niemi et al., 2003b; Niemi et al., 2003a).

3 BACKGROUND

3.1 OLAP

The OLAP cube is a multidimensional database in which the dimension attributes (i.e. co-ordinates) determine the value accessed. For example, the dimensions can be time, location, product, and the measures are sales and profit. The dimensions usually have a hierarchical structure, which enable analysing the data on different levels of details - for example at day- or month-level. In the latter case, the monthly data is aggregated from the daily data. The user analyses data by choosing dimensions to determine the viewpoint to data and "drilling down" or "rolling up" in the dimension hierarchy.

The contents of OLAP databases are typically collected from other data repositories, such as operational databases. For a well-defined and targeted system where the information needs are well known, it may be straightforward to collect the right data at the right time. However, with more and more data generally available also the information needs do develop. Consequently, it gets more difficult to anticipate the needs of the OLAP users. This leads to a situation in which it is increasingly difficult to know in advance what data are required - and when - for the desired analysis tasks.

In a corporate environment the need for up-to-date data means that the OLAP cube needs to be constructed almost in real time (Akinde et al., 2003). Geographical distribution adds challenges related to the management of the user rights, fault tolerance of the computation over the wide area networks and limited amount of available bandwidth.

3.2 Grid

Our design applies Grid technologies in security and parallel processing. Foster and Kesselman describe the Grid as "a software infrastructure that enables flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions and resources" (Foster et al., 2001).

Essential Grid components for our work are the Grid Security Infrastructure and distributed computing based on a Java agent technology. The main benefit for selecting these methods for the data integration is their large user base, indicating a large community of developers and tool providers. Furthermore, the standardization has progressed to a level where most of the components have several implementations, both open source and commercially supported. This offers a flexible way to balance the savings in the license fees in prototyping with the benefits of commercial support structure for production system.

Security, user authentication and user authorization An essential components of our framework is the Grid Security Infrastructure (GSI), which allows secure connections to potentially all computers in the Grid. GSI is based on public key encryption, X.509 certificates, proxy certificates with a limited validity (e.g. 12 hours), and the Secure Sockets Layer (SSL) communication protocol (The Globus project, 2002). Our design uses HTTP/HTTPS protocol to access remote databases, since it is normally allowed to pass through firewalls. The user authentication is based on a certificate, thus separate user IDs or passwords are not needed.

Many aspects of the data access designed here is based on technologies of the European DataGrid, EDG (European DataGrid, 2002). The client's request for EDG data security manager contains the user's proxy certificate, which is used to authenticate the user. The authentication is based on either the subject of the certificate or certificate extension fields described in the next paragraph. As an example of the first case, let us assume that the user has a certificate with subject "/O=Grid/O=NorduGrid/OU=hip.fi/CN=John Doe". The authentication component can assign database update rights to person with this subject. Moreover, it can assign database read rights to anyone whose certificate subject contains "O=NorduGrid/OU=hip.fi". EDG data security scheme can be applied to data accesses independently of the type of the data source.

VOMS In the Grid, data accesses often cross organisational boundaries, making it impractical to bind the access to the user's certificate subject alone. For this reason, virtual organization (VO) extensions to proxy certificate files have been introduced.

The Virtual Organization Membership System (VOMS) presents an extension to a user's X509 proxy certificate that includes their VO membership information. When a VOMS-proxy is generated, a VOMS server is contacted to request a VOMS-extended proxy certificate that follows the standard X509v3 (Housley et al., 1999) certificate format. All the standard fields of the proxy certificate are used to store the user's authentication information. The

X509v3 extension (1.3.6.1.4.1.8005.100.100.1) part is used to include authorization information in the user's proxy certificate. The authorization information is stored in triplets with the following syntax: `GROUP: string, ROLE: string, CAP: string,` where GROUP represents the user's group in the VO (like "OLAP users"), ROLE their role (like "Administrators"), and CAP optional capability specifications (like "10 GB disk space").

Distributed Computing Using Agent Technology Another key component for our system is a mobile Java agent framework (GridBlocks Agent (GB Agent) technology (Karppinen et al., 2003)) that allows distributed ETL processing without shipping all the data over the network. This Java agent technology offers also a way to hide the structure of the stored data from the OLAP system, since all the data sources (different types of databases and also other sources of data) will offer the GB agent interface for the OLAP cube construction. The necessary data transformations for this are encoded in XML, using the XSLT language.

3.3 XML and XML Tools

XML (eXtensible Markup Language) is offered as a solution while operating with data in a heterogeneous environment. XML aims to be "the universal format for structured documents and data on the Web" (The World Wide Web Consortium, 2002a). As such, XML is a meta language – a language for describing other languages – which lets one design customised markup languages for different types of documents, using a declaration syntax defined in recommendation.

In addition to the private data of the company, lots of usefull data for analysis purposes is available on the public Internet. Yet, utilising this data is difficult since the data formats tend to differ from the ones used in the analysis system. XML can be used to solve this problem since the e.g. HTML language of WWW can be seen as an XML sub language.[1] The key issue for our system is that an XML sub language can be transformed to another by using XSLT transformations. XSLT is a programming language, but many tools novel tools for writing XSLT programs are being published, see e.g. (Altova, 2003). These enable the user to generate the XSLT code via graphical user interface operating on more abstract level.

Several XML query languages have been made for retrieving the XML based data, for example XQL, XML-QL, and Quilt. At the moment XQuery developed by the W3C XML Query Working Group - is becoming de facto standard query language for XML databases. The mission of the W3C group is to (W3C, 2003): "provide flexible query facilities to extract data from real and virtual documents on the Web... Ultimately, collections of XML files will be accessed like databases." XQuery has a SQL-like syntax grafted on to XPath, which is a language for addressing parts of an XML document. The expectation is that XQuery will be for XML databases what SQL is for relational databases, i.e. a standard and vendor independent way to query and retrieve data (W3C, 2003).

4 FRAMEWORK ARCHITECTURE

Our work focuses on the ETL process, in which we combine the extraction and transformation processes and include aggregation computing in them. Figure 1 illustrates the system architecture.

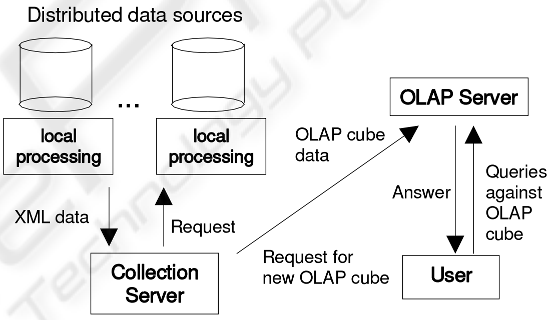

Figure 1: System architecture

The data sources can be legacy OLTP systems with a simple system accepting local processing code added. This interface can be protected using the Grid security framework. The collection server hides the complexity of the underlying system, making it possible to use OLAP tools with no direct Grid support.

Data Collection and Integration We assume that for confidential data, Grid security access mechanism like the one described in Section 3.2 has been applied. Once the data has been extracted, it must be transformed in order to provide a uniform base for our analysis.Writing XSLT instructions is quite complex but, in our case, the situation is easier since the target XML language is always the same, the one used by our ETL system. Therefore, it is quite straightforward to implement a tool to help the user to tell how the source XML (or even HTML) is transformed to the target XML. The aim is that the user works with a graphical interface mainly just clicking the fields of

[1] Strictly speaking, only XHTML is a "proper" XML sub language (The World Wide Web Consortium, 2002b).

the XML document to indicate which field of the document corresponds to the right field of the target document. The method also enables to define calculations in the transformation process.

Data Selection and Aggregation In the data selection we use XML query languages. The data selection is done in the network node where the data is stored. Data aggregation could also be done by using XSLT and XQuery but the current implementations are inefficient for large amounts of data. Thus, we use Java agent technology since Java is platform independent - a crucial feature in a heterogeneous environment.

Data Loading Most of the commercial OLAP servers support uploading of XML data. Failing this, writing a (e.g. XSLT) program to transfer the data into the form understood by the server and/or using the API of the OLAP server is necessary.

5 CONCLUSIONS

We have presented a design that applied Grid components for data access and security to combine data from different sources. In our method, the data for the analysis at hand is collected ’on-line’ from operational databases. In this way, we can avoid some of the issues of monolithic data warehouse projects by deploying the system step by step, utilizing existing legacy systems in the new framework. Grid technologies are applied to distribute the computation for the OLAP cube construction to achieve more performance. Finally, the data is uploaded into an OLAP server to enable the use of common OLAP tools. From the technical standpoint, our design is a promising alternative to existing data warehousing methods and thus a candidate for larger-scale tests, involving industrial usage scenarios and real operational data.

ACKNOWLEDGMENTS

We wish to thank Richard Randell from Imagination Lab, Lausanne. The work was supported by Academy of Finland, project number 203554.

REFERENCES

Akinde, M., Bhlen, M., Johnson, T., and et al. (2003). Efficient OLAP query processing in distributed data warehouses. *Information Systems*, 28.

Altova (2003). Altova corp.
Available on: http://www.altova.com.

Chaudhuri, S. and Dayal, U. (1997). An overview of data warehousing and OLAP technology. *SIGMOD Rec.*, 26(1):65–74.

European DataGrid (2002). European datagrid.
Available on: http://www.eu-datagrid.org.

Ewusi-Mensah, K. (1997). Critical issues in abandoned information systems development projects. *Communications of the ACM*, 9(40):74–80.

Foster, I., Kesselman, C., and Tuecke, S. (2001). The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15(3).

Housley, R., Ford, W., Polk, W., and Solo, D. (1999). RFC 2459, internet X.509 public key infrastructure certificate and CRL profile. Available on http://www.ietf.org/rfc/rfc2459.txt.

Karppinen, J., Niemi, T., and Niinimäki, M. (2003). Mobile analyzer - new concept for next generation of distributed computing. In *CCGrid 2003*. A poster.

Mohania, M., Samtani, S., Roddick, J., and Kambayashi, Y. (1999). Advances and research directions in data warehousing technology. *Australian Journal of Information Systems*.

Niemi, T., Niinimaki, M., Nummenmaa, J., and Thanisch, P. (2003a). Applying grid technologies to XML based OLAP cube constraction.

Niemi, T., Niinimäki, M., and Sivunen, V. (2003b). Integrating distributed heterogeneous databases and distributed grid computing. In *ICEIS 2003*, volume 1, pages 96–103. ICEIS Press.

Pedersen, D., Riis, K., and Pedersen, T. B. (2002). Query optimization for OLAP-XML federations. In *DOLAP 2002*, pages 57–64. ACM Press.

Sypherlink, I. (2003). Data warehousing.
Available on: http://www.sypherlink.com/solutions/dataware.asp.

The Globus project (2002). Commodity Grid Kits. Available on: http://www-unix.globus.org/cog/.

The World Wide Web Consortium (1999). XSL transformations (XSLT). Available on: http://www.w3.org/TR/xslt.

The World Wide Web Consortium (2002a). Extensible markup language (XML). Available on: http://www.w3.org/XML/.

The World Wide Web Consortium (2002b). XHTML, the extensible hypertext markup language. Available on: http://www.w3.org/TR/xhtml1.

W3C (2003). XML query (XQuery).
Available on: http://www.w3.org/XML/Query.

Zurek, T. and Sinnwell, M. (1999). Data warehousing has more colours than just black & white. In *VLDB 1999*. Morgan Kaufmann.