Jean-Marc Pierson (Ed.)

LNCS 3836

Data Management in Grids

First VLDB Workshop, DMG 2005
Trondheim, Norway, September 2005
Revised Selected Papers

Springer

Jean-Marc Pierson (Ed.)

Data Management in Grids

First VLDB Workshop, DMG 2005
Trondheim, Norway, September 2-3, 2005
Revised Selected Papers

Springer

Volume Editor

Jean-Marc Pierson
Lyon Research Center for Images and Intelligent Information Systems (LIRIS)
National Institute of Applied Science (INSA de Lyon)
69631 Villeurbanne, France
E-mail: jean-marc.pierson@liris.cnrs.fr

Library of Congress Control Number: 2005938219

CR Subject Classification (1998): H.2, H.4, H.3, H.2.4, E.2, H.5, C.2

ISSN 0302-9743
ISBN-10 3-540-31212-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-31212-3 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11611950 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3836

Message from the Program Chair

It was my great pleasure to welcome you in Trondheim for the first "Data Management in Grids" workshop, associated to the VLDB conference. Since the mid-1990s and the emergence of Grids, many research activities have been initiated in relation to data management in these dynamic, heterogeneous and cross-organizational environments. The database community can offer its unique expertise in the management of very large, widely distributed databases. Conversely, Grids offer a novel and very exciting field of research for database scientists both in terms of application domains and fundamental research.

This workshop was intended to bring together these two communities, and thus to offer a unique workspace for researchers to discuss and exchange ideas about the emerging challenges and opportunities offered by Data Grids. The co-location with the VLDB conference attracted researchers from both fields and launched interesting discussions.

The call for papers attracted 24 submissions. From the submissions, the Program Committee selected nine regular papers for the one-day workshop. The international flavor (seven countries represented for the final program) produced a very enriching and interactive workshop. In addition to the paper presentations, the program also included two invited talks : "Globally Distributed Data" by Reagan Moore, San Diego Supercomputing Center, USA, and "An Outline of the Global Grid Forum Data Access and Integration Service Specifications," by Mario Antonioletti, University of Edinburgh, UK.

I would like to thank all those who submitted papers for consideration, the participants of the conference, and the members of the Program Committee for their hard and conscientious work, their time and their careful effort in the reviewing process.

2005 Jean-Marc Pierson

Organization

Program Chair
Jean-Marc Pierson, LIRIS, INSA-Lyon, France

Program Committee
Lionel Brunie, LIRIS, France
Rajkumar Buyya, University of Melbourne, Australia
Neil P. Chue Hong, EPCC, The University of Edinburgh, UK
Alvaro A.A. Fernandes, The University of Manchester, UK
Ian Foster, Argone National Laboratory, USA
Dieter Gawlick, Oracle, USA
Peter Kacsuk, Mta Sztaki Research Institute, Hungary
Peter Kunszt, CERN, Switzerland
Stephen A. Langella, The Ohio State University, USA
Susan Malaika, IBM, USA
Johan Montagnat, CNRS, France
Reagan Moore, SDSC, USA
Gianluca Moro, DEIS - University of Bologna, Italy
Brigitte Plateau, ID-IMAG, France
Thierry Priol, IRISA, France
Heinz Stockinger, University of Vienna, Austria

External Reviewers
Gabriel Antoniu
Yves Denneulin
Sushant Goel
Yvon Jégou
Fabrice Jouanot
Stefano Lodi
Gabriele Monti
Ludwig Seitz
Olivier Valentin
Christine Verdier

Lecture Notes in Computer Science

For information about Vols. 1–3754

please contact your bookseller or Springer

Vol. 3860: D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006. XI, 365 pages. 2006.

Vol. 3855: E. A. Emerson, K.S. Namjoshi (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 443 pages. 2006.

Vol. 3850: R. Freund, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 371 pages. 2006.

Vol. 3842: H.T. Shen, J. Li, M. Li, J. Ni, W. Wang (Eds.), Advanced Web and Network Technologies, and Applications. XXVII, 1057 pages. 2005.

Vol. 3840: M. Li, B. Boehm, L.J. Osterweil (Eds.), Unifying the Software Process Spectrum. XVI, 522 pages. 2006.

Vol. 3838: A. Middeldorp, V. van Oostrom, F. van Raamsdonk, R. de Vrijer (Eds.), Processes, Terms and Cycles: Steps on the Road to Infinity. XVIII, 639 pages. 2005.

Vol. 3837: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks. IX, 307 pages. 2005.

Vol. 3836: J.-M. Pierson (Ed.), Data Management in Grids. X, 143 pages. 2005.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005. (Sublibrary LNAI).

Vol. 3834: D. Feitelson, E. Frachtenberg, L. Rudolph, U. Schwiegelshohn (Eds.), Job Scheduling Strategies for Parallel Processing. VIII, 283 pages. 2005.

Vol. 3833: K.-J. Li, C. Vangenot (Eds.), Web and Wireless Geographical Information Systems. XI, 309 pages. 2005.

Vol. 3832: D. Zhang, A.K. Jain (Eds.), Advances in Biometrics. XX, 796 pages. 2005.

Vol. 3831: J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, J. Štuller (Eds.), SOFSEM 2006: Theory and Practice of Computer Science. XV, 576 pages. 2006.

Vol. 3829: P. Pettersson, W. Yi (Eds.), Formal Modeling and Analysis of Timed Systems. IX, 305 pages. 2005.

Vol. 3828: X. Deng, Y. Ye (Eds.), Internet and Network Economics. XVII, 1106 pages. 2005.

Vol. 3827: X. Deng, D. Du (Eds.), Algorithms and Computation. XX, 1190 pages. 2005.

Vol. 3826: B. Benatallah, F. Casati, P. Traverso (Eds.), Service-Oriented Computing - ICSOC 2005. XVIII, 597 pages. 2005.

Vol. 3824: L.T. Yang, M. Amamiya, Z. Liu, M. Guo, F.J. Rammig (Eds.), Embedded and Ubiquitous Computing – EUC 2005. XXIII, 1204 pages. 2005.

Vol. 3823: T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, L.T. Yang (Eds.), Embedded and Ubiquitous Computing – EUC 2005 Workshops. XXXII, 1317 pages. 2005.

Vol. 3822: D. Feng, D. Lin, M. Yung (Eds.), Information Security and Cryptology. XII, 420 pages. 2005.

Vol. 3821: R. Ramanujam, S. Sen (Eds.), FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science. XIV, 566 pages. 2005.

Vol. 3820: L.T. Yang, X. Zhou, W. Zhao, Z. Wu, Y. Zhu, M. Lin (Eds.), Embedded Software and Systems. XXVIII, 779 pages. 2005.

Vol. 3819: P. Van Hentenryck (Ed.), Practical Aspects of Declarative Languages. X, 231 pages. 2005.

Vol. 3818: S. Grumbach, L. Sui, V. Vianu (Eds.), Advances in Computer Science – ASIAN 2005. XIII, 294 pages. 2005.

Vol. 3816: G. Chakraborty (Ed.), Distributed Computing and Internet Technology. XXI, 606 pages. 2005.

Vol. 3815: E.A. Fox, E.J. Neuhold, P. Premsmit, V. Wuwongse (Eds.), Digital Libraries: Implementing Strategies and Sharing Experiences. XVII, 529 pages. 2005.

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005. (Sublibrary LNAI).

Vol. 3813: R. Molva, G. Tsudik, D. Westhoff (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. VIII, 219 pages. 2005.

Vol. 3810: Y.G. Desmedt, H. Wang, Y. Mu, Y. Li (Eds.), Cryptology and Network Security. XI, 349 pages. 2005.

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005. (Sublibrary LNAI).

Vol. 3808: C. Bento, A. Cardoso, G. Dias (Eds.), Progress in Artificial Intelligence. XVIII, 704 pages. 2005. (Sublibrary LNAI).

Vol. 3807: M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005 Workshops. XV, 275 pages. 2005.

Vol. 3806: A.H. H. Ngu, M. Kitsuregawa, E.J. Neuhold, J.-Y. Chung, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005. XXI, 771 pages. 2005.

Vol. 3805: G. Subsol (Ed.), Virtual Storytelling. XII, 289 pages. 2005.

Vol. 3804: G. Bebis, R. Boyle, D. Koracin, B. Parvin (Eds.), Advances in Visual Computing. XX, 755 pages. 2005.

Vol. 3803: S. Jajodia, C. Mazumdar (Eds.), Information Systems Security. XI, 342 pages. 2005.

Vol. 3802: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part II. XLII, 1166 pages. 2005. (Sublibrary LNAI).

Vol. 3801: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part I. XLI, 1122 pages. 2005. (Sublibrary LNAI).

Vol. 3799: M. A. Rodríguez, I.F. Cruz, S. Levashkin, M.J. Egenhofer (Eds.), GeoSpatial Semantics. X, 259 pages. 2005.

Vol. 3798: A. Dearle, S. Eisenbach (Eds.), Component Deployment. X, 197 pages. 2005.

Vol. 3797: S. Maitra, C. E. V. Madhavan, R. Venkatesan (Eds.), Progress in Cryptology - INDOCRYPT 2005. XIV, 417 pages. 2005.

Vol. 3796: N.P. Smart (Ed.), Cryptography and Coding. XI, 461 pages. 2005.

Vol. 3795: H. Zhuge, G.C. Fox (Eds.), Grid and Cooperative Computing - GCC 2005. XXI, 1203 pages. 2005.

Vol. 3794: X. Jia, J. Wu, Y. He (Eds.), Mobile Ad-hoc and Sensor Networks. XX, 1136 pages. 2005.

Vol. 3793: T. Conte, N. Navarro, W.-m.W. Hwu, M. Valero, T. Ungerer (Eds.), High Performance Embedded Architectures and Compilers. XIII, 317 pages. 2005.

Vol. 3792: I. Richardson, P. Abrahamsson, R. Messnarz (Eds.), Software Process Improvement. VIII, 215 pages. 2005.

Vol. 3791: A. Adi, S. Stoutenburg, S. Tabet (Eds.), Rules and Rule Markup Languages for the Semantic Web. X, 225 pages. 2005.

Vol. 3790: G. Alonso (Ed.), Middleware 2005. XIII, 443 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005. (Sublibrary LNAI).

Vol. 3788: B. Roy (Ed.), Advances in Cryptology - ASIACRYPT 2005. XIV, 703 pages. 2005.

Vol. 3787: D. Kratsch (Ed.), Graph-Theoretic Concepts in Computer Science. XIV, 470 pages. 2005.

Vol. 3785: K.-K. Lau, R. Banach (Eds.), Formal Methods and Software Engineering. XIV, 496 pages. 2005.

Vol. 3784: J. Tao, T. Tan, R.W. Picard (Eds.), Affective Computing and Intelligent Interaction. XIX, 1008 pages. 2005.

Vol. 3783: S. Qing, W. Mao, J. Lopez, G. Wang (Eds.), Information and Communications Security. XIV, 492 pages. 2005.

Vol. 3782: K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, T. Roth-Berghofer (Eds.), Professional Knowledge Management. XXIII, 739 pages. 2005. (Sublibrary LNAI).

Vol. 3781: S.Z. Li, Z. Sun, T. Tan, S. Pankanti, G. Chollet, D. Zhang (Eds.), Advances in Biometric Person Authentication. XI, 250 pages. 2005.

Vol. 3780: K. Yi (Ed.), Programming Languages and Systems. XI, 435 pages. 2005.

Vol. 3779: H. Jin, D. Reed, W. Jiang (Eds.), Network and Parallel Computing. XV, 513 pages. 2005.

Vol. 3778: C. Atkinson, C. Bunse, H.-G. Gross, C. Peper (Eds.), Component-Based Software Development for Embedded Systems. VIII, 345 pages. 2005.

Vol. 3777: O.B. Lupanov, O.M. Kasim-Zade, A.V. Chaskin, K. Steinhöfel (Eds.), Stochastic Algorithms: Foundations and Applications. VIII, 239 pages. 2005.

Vol. 3776: S.K. Pal, S. Bandyopadhyay, S. Biswas (Eds.), Pattern Recognition and Machine Intelligence. XXIV, 808 pages. 2005.

Vol. 3775: J. Schönwälder, J. Serrat (Eds.), Ambient Networks. XIII, 281 pages. 2005.

Vol. 3774: G. Bierman, C. Koch (Eds.), Database Programming Languages. X, 295 pages. 2005.

Vol. 3773: A. Sanfeliu, M.L. Cortés (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XX, 1094 pages. 2005.

Vol. 3772: M.P. Consens, G. Navarro (Eds.), String Processing and Information Retrieval. XIV, 406 pages. 2005.

Vol. 3771: J.M.T. Romijn, G.P. Smith, J. van de Pol (Eds.), Integrated Formal Methods. XI, 407 pages. 2005.

Vol. 3770: J. Akoka, S.W. Liddle, I.-Y. Song, M. Bertolotto, I. Comyn-Wattiau, W.-J. van den Heuvel, M. Kolp, J. Trujillo, C. Kop, H.C. Mayr (Eds.), Perspectives in Conceptual Modeling. XXII, 476 pages. 2005.

Vol. 3769: D.A. Bader, M. Parashar, V. Sridhar, V.K. Prasanna (Eds.), High Performance Computing – HiPC 2005. XXVIII, 550 pages. 2005.

Vol. 3768: Y.-S. Ho, H.J. Kim (Eds.), Advances in Multimedia Information Processing - PCM 2005, Part II. XXVIII, 1088 pages. 2005.

Vol. 3767: Y.-S. Ho, H.J. Kim (Eds.), Advances in Multimedia Information Processing - PCM 2005, Part I. XXVIII, 1022 pages. 2005.

Vol. 3766: N. Sebe, M.S. Lew, T.S. Huang (Eds.), Computer Vision in Human-Computer Interaction. X, 231 pages. 2005.

Vol. 3765: Y. Liu, T. Jiang, C. Zhang (Eds.), Computer Vision for Biomedical Image Applications. X, 563 pages. 2005.

Vol. 3764: S. Tixeuil, T. Herman (Eds.), Self-Stabilizing Systems. VIII, 229 pages. 2005.

Vol. 3762: R. Meersman, Z. Tari, P. Herrero (Eds.), On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. XXXI, 1228 pages. 2005.

Vol. 3761: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part II. XXVII, 653 pages. 2005.

Vol. 3760: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Part I. XXVII, 921 pages. 2005.

Vol. 3759: G. Chen, Y. Pan, M. Guo, J. Lu (Eds.), Parallel and Distributed Processing and Applications - ISPA 2005 Workshops. XIII, 669 pages. 2005.

Vol. 3758: Y. Pan, D.-x. Chen, M. Guo, J. Cao, J.J. Dongarra (Eds.), Parallel and Distributed Processing and Applications. XXIII, 1162 pages. 2005.

Vol. 3757: A. Rangarajan, B. Vemuri, A.L. Yuille (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. XII, 666 pages. 2005.

Vol. 3756: J. Cao, W. Nejdl, M. Xu (Eds.), Advanced Parallel Processing Technologies. XIV, 526 pages. 2005.

Table of Contents

Globally Distributed Data

Reagan W. Moore

San Diego Supercomputer Center,
San Diego, CA
moore@sdsc.edu

The management of globally distributed data is simplified through the use of data grids which enable data sharing environments. Data grids provide both the interoperability mechanisms needed to interact with legacy storage systems and legacy applications, as well as the logical name spaces needed to identify files, resources, and users. Data grids also provide support for consistent management of state information about each file within the distributed environment. The state information includes access controls, descriptive metadata, and administration metadata. These capabilities enable data virtualization, the ability to manage data independently of the chosen storage repositories. Applications that manage globally distributed data include data grid federations, distributed digital libraries, and distributed persistent archives.

The Storage Resource Broker (SRB), developed at the San Diego Supercomputer Center, is an example of generic data grid software infrastructure that uses database technology to manage state information for globally shared collections. The software has been under development since 1995, initially funded by the Defense Advanced Research Project Agency to support Massive Data Analysis Systems. Subsequent projects focused on the application of the technology in support of distributed data management. Across most of the projects that use data grid technology, the goal of distributed data management was either:

- sharing of data. Scientists share original data sets while pursuing research.
- publication of data. Scientists register data sets used in the research into a digital library for future discovery and access.
- preservation of data. This corresponds to the creation of standard digital reference sets that are then used as the intellectual capital of the discipline. New theory or observational results are compared against these reference data. The technology that manages infrastructure independence (ability to migrate collections onto new hardware and software systems) is called a persistent archive.

The application areas included a patent digital library for the US Patent and Trademark Office; Department of Energy high energy physics data grids; prototype research persistent archives for the National Archives and Records Administration; National Library of Medicine digital library for digital embryo images; National Science Foundation persistent archive for the National Science Digital Library; National Institute of Health Biomedical Informatics Research Network data grid for neuroscience data; NSF Real-time Observatories, Applications, and Data Management network for sensor data; and NSF National Virtual Observatory for astronomy sky surveys.

J.-M. Pierson (Ed.): VLDB DMG 2005, LNCS 3836, pp. 1 – 3, 2005.

Each project defined standard descriptive metadata, standard data encoding formats, and standard services that would be used to manipulate the data. The descriptive metadata were organized in a collection hierarchy along with administrative metadata that represented state information managed by the data grid. Data grids separated the management of the state information from the storage of the data. The result of each operation on the material within the data grid was tracked and the associated state information was associated with the files that were manipulated. An example is the creation of a replica of a file. The location of the replica and the date it was created were registered into a metadata catalog that is maintained in a relational database.

A typical project assembled a shared collection that contained 1-million to 2-million files, with the largest shared collection holding over 27 million files. The sizes of the collections ranged from 1-2 terabytes to collections of simulation data that exceeded 150 terabytes. The collections were organized by groups of researchers that contained 20 persons up to several hundred persons.

SDSC is approaching the issue of global data management from three different perspectives:

1. Data virtualization: Create a shared collection that manages the name spaces for resources, files, users, metadata, and access controls independently from the storage system.
2. Access virtualization: Integrate distributed data management into major applications such as digital libraries, persistent archives, and real-time sensor data management systems.
3. State information virtualization: Define the consistency constraints that are applied in the update of state information, and provide the ability to change the constraints when federating data grids, modifying views of collections, managing data placement, or asserting global consistency properties.

The data virtualization efforts are exemplified through the use of the SRB to manage logical name spaces to provide persistent global naming. In addition, the SRB differentiates between the access methods required by preferred interfaces, and the storage repository protocols required to interact with legacy storage systems. The SRB maps from the access interface protocol to a standard set of operations for manipulating data and metadata. The SRB then maps from a standard set of operations that will be performed at a storage repository to the particular protocol required by that system. The result is the ability to manage data that is distributed across multiple types of storage systems, while providing a uniform access interface.

The access virtualization builds upon the data virtualization through the integration of advanced user interfaces. A major example is the integration of digital library technology with data grid technology. The goal is the ability to provide digital library services on collections that are distributed across multiple storage systems, and authentication domains.

An example is the integration of the DSpace digital library, developed at MIT and Hewelett Packard, with the Storage Resource Broker data grid. The resulting system is able to support:

- creation of DSpace collections whose size is greater than the local disk capacity
- replication of digital entities between sites for disaster recovery
- access to digital entities that reside in another DSpace instance

A similar effort is being done to integrate the Fedora digital library, developed at University of Virginia and Cornell University, on top of the Storage Resource Broker data grid.

The state information virtualization is a new research effort with the goal of producing the next generation of data management technology. When federating data grids that manage different consistency constraints, the ability to characterize the chosen constraints becomes important. Building a shared collection that crosses multiple environments with different consistency requirements can be accomplished through the direct association of the governing constraint with each metadata attribute. Two types of constraints are required: procedural rules that are followed when state information is updated, and global consistency constraints that can be verified as properties of a sub-collection. This latter example is one of the major challenges of distributed data management systems, namely how to create consistent state from possibly inconsistent state.

For further information, see http://www.sdsc.edu/srb.

XML Data Integration in OGSA Grids

Carmela Comito and Domenico Talia

DEIS, University of Calabria,
Via P. Bucci 41 c,
87036 Rende, Italy
{ccomito, talia}@deis.unical.it
http://www.deis.unical.it/

Abstract. Data integration is the flexible and managed federation, analysis, and processing of data from different distributed sources. Data integration is becoming as important as data mining for exploiting the value of large and distributed data sets that are available today. Distributed processing infrastructures such as Grids can be used for data integration on geographically distributed sites. This paper presents a framework for integrating heterogeneous XML data sources distributed among the nodes of a Grid. We propose a query reformulation algorithm to combine and query XML documents through a decentralized point-to-point mediation process among the different data sources based on schema mappings. The above cited XML integration formalism is exposed as a Grid Service within the GDIS architecture. GDIS is a service-based architecture for providing data integration in Grids using a decentralized approach. The underlying model of such architecture is discussed and we show how it fits the XMAP formalism/algorithm.

1 Introduction

The Grid offers new opportunities and raises new challenges in data management arising from large scale, dynamic, autonomous, and distributed nature of data sources. A Grid can include related data resources maintained in different syntaxes, managed by different software systems, and accessible through different protocols and interfaces. Due to this diversity in data resources, one of the most demanding issue in managing data on Grids is reconciliation of data heterogeneity. Therefore, in order to provide facilities for addressing requests over multiple heterogeneous data sources, it is necessary to provide data integration models and mechanisms.

Data integration is the flexible and managed federation, analysis, and processing of data from different distributed sources. In particular, the rise in availability of web-based data sources has led new challenges in data integration systems for obtaining decentralized, wide-scale sharing of data, preserving semantics. These new needs in data integration systems are also felt in Grid settings. In a Grid it is not suitable to refer to a centralized structure for coordinating all the nodes because it can become a bottleneck and, most of all, it doesn't benefit from the dynamic and distributed nature of Grid resources.

The Grid community is devoting great attention toward the management of structured and semi-structured data such as databases and XML data. The most

J.-M. Pierson (Ed.): VLDB DMG 2005, LNCS 3836, pp. 4–15, 2005.

significant examples of such efforts are the *OGSA Data Access and Integration* (OGSA-DAI) [1] and the *OGSA Distributed Query Processor* (OGSA-DQP) [2] projects. However, till today only few of those projects [3,4] actually meet schema-integration issues necessary for establishing semantic connections among heterogeneous data sources.

For these reasons, we propose a framework for integrating heterogeneous XML data sources distributed over a Grid. By designing this framework, we aim at developing a decentralized network of semantically related schemas that enables the formulation of distributed queries over heterogeneous data sources. We designed a method to combine and query XML documents through a decentralized point-to-point mediation process among the different data sources based on schema mappings. We offer a decentralized service-based architecture that exposes this XML integration formalism as a Grid Service [5]. We refer to this architecture as the *Grid Data Integration System* (GDIS). The GDIS infrastructure exploits the middleware provided by OGSA-DQP, OGSA-DAI, and Globus Toolkit 3 [6], building on top of them schema-integration services.

The remainder of the paper is organized as follows. Section 2 presents a short analysis of data integration systems focusing on specific issues related to Grids. Section 3 presents the XMAP integration framework; the underlying integration model and the XMAP query reformulation algorithm are described. Section 4 illustrates the deployment of the XMAP framework on a service-based Grid architecture. Finally, Section 5 outlines future work and draws some conclusions.

2 Data Integration and Grids

The goal of a data integration system is to combine heterogeneous data residing at different sites by providing a unified view of this data. The two main approaches to data integration are federated database management systems (FDBMSs) and traditional mediator/wrapper-based integration systems.

A federated database management system (FDBMS) [7] is a collection of cooperating but autonomous component database systems (DBSs). The DBMS of a component DBS, or component DBMS, can be a centralized or distributed DBMS or another FDBMS. The component DBMSs can differ in different aspects such as data models, query languages, and transaction management capabilities.

Traditional data integration systems [8] are characterized by an architecture based on one or more mediated schemas and a set of sources. The sources contain the real data, while every mediated schema provides a reconciled, integrated, and virtual view of the underlying sources. Moreover, the system includes a set of source descriptions that provide semantic mappings between the relations in the source schemas and the relations in the mediated schemas [9].

Data integration on Grids presents a twofold characterization:

1. data integration is a key issue for exploiting the availability of large, heterogeneous, distributed and highly dynamic data volumes on Grids;
2. integration formalisms can benefit from an OGSA-based Grid infrastructure, since it facilitates dynamic discovery, allocation, access, and use of both data

sources and computational resources, as required to support computationally demanding database operations such as query reformulation, compilation and evaluation.

Data integration on Grids has to deal with unpredictable, highly dynamic data volumes provided by unpredictable membership of nodes that happen to be participating at any given time. So, traditional approaches to data integration, such as FDBMS [7] and the use of mediator/wrapper middleware [9], are not suitable in Grid settings. The federation approach is a rather rigid configuration where resources allocation is static and optimization cannot take advantage of evolving circumstances in the execution environment. The design of mediator/wrapper integration systems must be done globally and the coordination of mediators has to be done centrally, which is an obstacle to the exploitation of evolving characteristics of dynamic environments. As a consequence, data sources cannot change often and significantly, otherwise they may violate the mappings to the mediated schema.

The rise in availability of web-based data sources has led to new challenges in data integration systems in order to obtain decentralized, wide-scale sharing of semantically-related data. Recently, several works on data management in peer-to-peer (P2P) systems are moving along this direction [10, 11, 12, 13]. All these systems focus on an integration approach not based on a global schema: each peer represents an autonomous information system, and data integration is achieved by establishing mappings among the various peers.

To the best of our knowledge, there are only few works designed to provide schema-integration in Grids. The most notable ones are *Hyper* [3] and *GDMS* [4]. Both systems are based on the same approach that we have used ourselves: building data integration services by extending the reference implementation of OGSA-DAI. The *Grid Data Mediation Service* (GDMS) uses a wrapper/mediator approach based on a global schema. GDMS presents heterogeneous, distributed data sources as one logical virtual data source in the form of an OGSA-DAI service. This work is essentially different from ours as it uses a global schema. For its part, *Hyper* is a framework that integrates relational data in P2P systems built on Grid infrastructures. As in other P2P integration systems, the integration is achieved without using any hierarchical structure for establishing mappings among the autonomous peers. In that framework, the authors use a simple relational language for expressing both the schemas and the mappings. By comparison, our integration model follows as Hyper an approach not based on a hierarchical structure, however differently from Hyper it focuses on XML data sources and is based on schema-mappings that associate paths in different schemas.

3 A Decentralized XML Data Integration Framework

In this section, we describe a framework meant to integrate heterogeneous XML data sources distributed among nodes of a Grid. The primary design goal of this framework is to develop a decentralized network of semantically related schemas that enables the formulation of queries over heterogeneous, distributed data sources.

The environment is modeled as a system composed of a number of Grid nodes, where each node can hold one or more XML databases. These nodes are connected to each other through declarative mappings rules. The framework implements then a method to combine and query XML documents through a decentralized point-to-point mediation process among the different data sources. Moreover, the interface it exposes to access and query the XML data sources is completely uniform, regardless of the intrinsic complexity of the underlying system.

3.1 Integration Model

Our integration model is based on schema mappings to translate queries between different schemas. The goal of a schema mapping is to capture structural as well as terminological correspondences between schemas.

As mentioned before, traditional centralized architecture of data integration systems is not suitable for highly dynamic and distributed environments such as the Grid. Thus, we propose an approach inspired from [13] where the mapping rules are established directly among source schemas without relying on a central mediator or a hierarchy of mediators. In consequence, in our integration model, there is no global schema representing all data sources in a unique data model but a collection of local schemas (the native schema of each data source). This way, the coordination of the various nodes is completely decentralized. Each node is free to establish the semantic connections with the source schemas it considers more appropriate. Therefore, to integrate a source in the system, one needs only to provide a set of mapping rules that describes the relationships between its schema and the other schemas it is related to.

The specification of mappings is thus flexible and scalable. Regardless of the total number of nodes composing the system, each source schema is directly connected to only a small number of other schemas. However, it remains reachable from all other schemas that belong to its "transitive closure". For any mapping M, its closure is defined as the set of rules that can be derived from M by repeated composition of schema paths. In other words, the system supports two different kinds of mapping to connect schemas semantically: *point-to-point* mappings and *transitive* mappings. In transitive mappings, data sources are related through one or more "mediator schemas". For example, if we have a source A directly connected to a source B and B connected to C, A is connected to *both* B and C. Establishing the mappings this way creates a graph of semantically related sources where each of the sources knows its direct semantic neighbors (point-to-point mapping) and can learn about the mappings of its neighbors (transitive mapping). Therefore, in our integration model all nodes are equal: there is no distinction between data sources and mediators. Each node acts both as a data source contributing data and as a local mediator providing an uniform view over the data provided by other nodes.

We address structural heterogeneity among XML data sources by associating paths in different schemas. Mappings are specified as path expressions that relate a specific element or attribute (together with its path) in the source schema to related elements or attributes in the destination schema. The data integration