

An Approach for Virtualization and Integration of Heterogeneous Cloud Databases

Anita Ahuja* Ajay Kumar ** Ramveer Singh***

*(Department of Computer Science, Asst. Professor, Mewar University, Chittorgarh (India)

** (Department of Computer Science, Asst. Professor, Mewar University, Chittorgarh (India))

*** (Department of Computer Science, Professor, R.K.G.I.T.,Mahamaya Technical University, Ghaziabad(India)

ABSTRACT:

Virtualization is the key technology behind cloud computing that allows the creation of an abstraction layer of the underlying cloud Infrastructure. Using virtualization, resources (hardware and software) can be shared and utilized while hiding the complexity from the cloud users. A lot of cloud database are available that managed by different organization such as- Amazon Storage for the Cloud, Google Storage for the Cloud, Hadoop Storage for the Cloud, Yahoo!'s PNUTS, Cassandra, CouchDB etc.

This paper is presented to propose a virtual Database framework that enables the centralized global object oriented database. A virtually integrated huge database that will hide the heterogeneity of various cloud databases. Once they are integrated a consistent access is provide to the end user.

Keywords – OOMDS, Virtualization, Cloud, Databases, cloud computing, Mediator Framework, Peers.

I. INTRODUCTION

Cloud computing is a model for enabling convenient, on- demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction[3]

The different cloud providers adopt different architecture and data models such as Amazon's storage building block Dynamo[6], S3, SimpleDB, and RDS, S3, Google storage building blocks Bigtable, Hadoop's building block HDFS, Hive, HadoopDB, and HBase, Yahoo's PNUTS, Cassandra data model, CouchDB data model.

It is realized that traditional DBMS does not fit well for the cloud computing environment so new data model row oriented, document oriented, widecolumn are widely used in cloud. Different cloud providers use different architecture and data models that best suit their application.

Now A Virtual integrated database management system should be developed that Provides

distribution transparency, Global schema- Common data descriptions & Data placement information, Centralized admin through global catalog, Distributed functions, Query processing, Transaction management, Access control etc[1].

II WHY NOT RDBMS?

RDBMS all have a distributed and parallel version with SQL support for all kinds of data (structured, XML, multimedia, streams, etc.) [1] Standard SQL a major argument for adoption by tool vendors (e.g. analytics, business intelligence), but the "one size fits all" approach has reached the limits result loss of performance.

Now simplicity and flexibility required for applications with specific, tight requirements. New specialized DBMS engines more efficient: column-oriented DBMS for OLAP, DSMS for stream processing, SciDB[11] for scientific analytics, etc. RDBMS provides ACID transactions, complex query language, lots of tuning knobs but it is less suitable for specific optimizations for OLAP, flexible programming model, flexible schema and scalability.

III INTEGRATED DATA MANAGEMENT PROBLEM IN CLOUD

Cloud data are very large (lots of data spaces, very large collections, multimedia etc). They are Complex, unstructured or semi-structured often schema less but metadata (tags,). Different file formats, access protocols and query languages are used. Table decompositions may vary, column names (data labels) may be different (but have the same semantics), and data encoding schemes may vary it also referred as schematic heterogeneity[8]. Cloud users and application developers are in very high numbers with very diverse expertise but very little DBMS expertise.

IV PROPOSED FRAMEWORK

Object Oriented Mediator Database System (OOMDS):

The proposed system is object oriented mediator data base system of various heterogeneous cloud data bases that having object

oriented query language in which object oriented views of data can be specified. In OOMDS has primitive to translate data from different clouds database into object oriented data base. These translated cloud data can be used to build views. This OOMDS supports multiple data base exists on cloud.

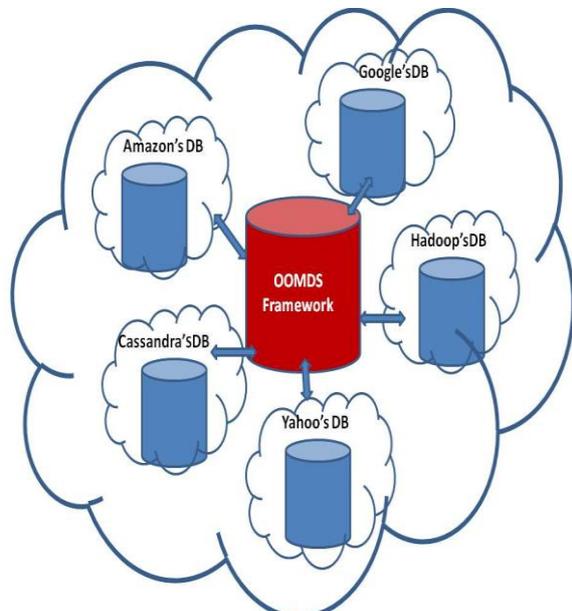


FIGURE : OBJECT ORIENTED MEDIATOR DATABASE SYSTEM

DATA INTEGRATION IN OODMS SYSTEM

OOMDS is a distributed mediator system that uses a object oriented data model and has a relationally complete object oriented query language, OOMDSQL. Through its distributed object oriented multi-database facilities many autonomous and distributed OOMDS peers can interoperate. Object oriented multi-database queries and views can be defined where external data sources of different kinds are translated through OOMDS and reconciled through its functional object oriented mediation primitives. Each mediator peer provides a number of transparent functional views of data reconciled from other mediator peers, wrapped data sources, and data stored in OOMDS itself. The composition of mediator peers in terms of other peers provides a way to scale the data integration process by composing mediation modules. The OOMDS data manager and query processor must be extensible so that new application oriented data types and operators can be added to OODMSQL, implemented in some external programming language (Java, C, C++ or Lisp). The extensibility allows wrapping data representations specialized

for different application areas in mediator peers. The object oriented data model provides very powerful query and data integration primitives which require advanced query optimization.

The mediator/wrapper approach has been used for integrating heterogeneous data in several projects. Most mediator systems integrate data through a central mediator server accessing one or several data sources through a number of "wrapper" interfaces that translate data to a global data model. However, one of the original goals for mediator architectures was that mediators should be relatively simple distributed software modules that transparently encode domain-specific knowledge about data and share abstractions of that data with higher layers of mediators or applications. Larger networks of mediators would then be defined through these primitive mediators by composing new mediators in terms of other mediators and data sources. The core of OOMDS is an open, light-weight, and extensible object oriented database management system with a object oriented data model. Each OOMDS server must contains all the traditional database facilities, such as a storage manager, a recovery manager, a transaction manager, and a functional query language named OOMDSQL. The system can be used as a single-user database or as a multi-user server to applications and to other OOMDS peers.

DISTRIBUTION:

OOMDS is a distributed mediator system where several mediator peers communicate over the Internet. Each mediator peer appears as a virtual functional database layer having data abstractions and a object oriented query language. Object oriented views provide transparent access to data sources from clients and other mediator peers. Conflicts and overlaps between similar real-world entities being modeled differently in different data sources are reconciled through the mediation primitives of the multi-mediator query language OOMBSQL. The mediation services allow transparent access to similar data structures represented differently in different data sources[13]. Applications access data from distributed data sources through queries to views in some mediator peer[9].

Logical composition of mediators is achieved when multi-database views in mediators are defined in terms of views, tables, and functions in other mediators or data sources. The multi-database views make the mediator peers appear to the user as a single virtual database. OOMDS mediators are composable since a mediator peer can regard other mediator peers as data sources[16].

WRAPPING DATA

In order to access data from external data sources OOMDS mediators may contain one or several wrappers which process data from different kinds of external data sources[15], e.g. ODBC-based access to relational databases, access to XML files, CAD systems, or Internet search engines to extract data from heterogeneous cloud data bases. A wrapper is a procedure in OOMDS having specialized facilities for query processing and translation of data from a particular class of external data sources. It contains both interfaces to external data sources and knowledge of how to efficiently translate and process queries involving accesses to different cloud databases. In particular, external OOMDS peers known to a mediator are also regarded as external data sources and there is a special wrapper for accessing other OOMDS peers[18]. However, among the OOMDS peers special query optimization methods are used that take into account the distribution, capabilities, costs, etc., of the different peers[20].

THE CENTRAL NAME SERVER

Every mediator peer must belong to a group of mediator peers. The mediator peers in a group are described through a meta-schema stored in a mediator server called central name server. The mediator peers are autonomous and there is no central schema in the name server [13]. The central name server contains only general meta-information such as the locations and names of the peers in the group while each mediator peer has its own schema describing its local data and data sources. The information in the central name server is managed without explicit operator intervention; its content is managed through messages from the mediator peers. To avoid a bottleneck, mediator peers usually communicate directly without involving the name server; it is normally involved only when a connection to some new mediator peer is established [21].

CONCLUSION:

We have given an overview of the OOMDS mediator system where groups of distributed mediator peers are used to integrate data from different sources. Each mediator in a group has DBMS facilities for query compilation and exchange of data and meta-data with other mediator peers. Derived functions can be defined where data from several mediator peers is abstracted, transformed, and reconciled. Wrappers are defined by interfacing OOMDS systems with external systems through its multi-directional foreign function interface. OOMDS can furthermore be embedded in applications and used as stand-alone databases.

The Object oriented Data Model and query language forming the basis for data integration in OOMDS. The distributed multi-mediator query decomposition strategies used were summarized.

The mediator peers are autonomous without any central schema. A special mediator, the central name server, keeps track of what mediator peers are members of a group. The central name servers can be queried for the location of mediator peers in a group. Meta-queries to each mediator peer can be posed to investigate the structure of its schema.

Some unique features of OOMDS are: A distributed mediator Framework where query plans are distributed over several communicating mediator peers. Using declarative object oriented queries to model reconciled object oriented views spanning multiple mediator peers. Query processing and optimization techniques for queries to reconcile views involving function overloading, late binding, and type-aware query rewrites.

REFERENCES

- [1] S. Aulbach, T. Grust, D. Jacobs, A. Kemper, and J. Rittinger. Multi-tenant databases for software as a service: Schema-mapping techniques. In SIGMOD, 2008.
- [2] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska. Building a database on S3. In SIGMOD, 2008.
- [3] F. Chang, J. Dean, S. Ghemawat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber. Bigtable: A distributed storage system for structured data. In OSDI, 2006.
- [4] B. F. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H.-A. Jacobsen, N. Puz, D. Weaver, and R. Yerneni. PNUTS: Yahoo!'s hosted data serving platform. PVLDB, 1(2), 2008.
- [5] C. Curino, E. Jones, Y. Zhang, and S. Madden. Schism: A Workload-Driven Approach to Database Replication and Partitioning. In VLDB, 2010.
- [6] E. Damiani, S. D. C. di Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Balancing Confidentiality and Efficiency in Untrusted Relational DBMS. CCS, 2003.
- [7] S. Das, D. Agrawal, and A. E. Abbadi. ElasTraS: An elastic transactional data store in the cloud. HotCloud, 2009.
- [8] R. Freeman. Oracle Database 11g New Features. McGraw-Hill, Inc., New York, NY, USA, 2008.
- [9] R. Gennaro, C. Gentry, and B. Parno. Non-Interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers. STOC, 2010.

- [11] H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra. Executing SQL over Encrypted Data in the Database-Service-Provider Model. ACM SIGMOD, 2002.
- [12] "Kernel based virtual machine (KVM)." [Online]. Available:<http://www.linux-kvm.org>
- [13] G. Giunta, R. Montella, G. Agrillo, and G. Coviello, "A GPGPU transparent virtualization component for high performance computing clouds," in Proceedings of the 16th international Euro-Par conference on Parallel processing: Part I, ser. EuroPar'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 379–391. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1887695.1887738>
- [14] L. Shi, H. Chen, and J. Sun, "vCUDA: GPU accelerated high performance computing in virtual machines," in Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–11. [Online]. Available:<http://portal.acm.org/citation.cfm?id=1586640.1587737>
- [15] F. Bellard, "QEMU, a fast and portable dynamic translator," in Proceedings of the annual conference on USENIX Annual Technical Conference, ser. ATEC '05. Berkeley, CA, USA: USENIX Association, 2005, pp. 441. [Online]. Available:<http://portal.acm.org/citation.cfm?id=1247360.1247401>
- [16] Josifovski V. Design, implementation and evaluation of a distributed mediator system for data integration. PhD Thesis, Linköping U., Sweden. <http://www.dis.uu.se/~udbl/publ/vanjaphd.pdf> [1999].
- [17] Bukhres O, Elmagarmid A (eds.). Object-oriented Multidatabase Systems. Prentice Hall, 1996.
- [18] Dayal U, Hwang H-Y. View definition and generalization for database integration in a multidatabase system.IEEE Transactions on Software Engineering 1984; 10(6):628–645.
- [19] A. N. Laboratory. (2010, Jul.) Heckle. [Online]. Available: <http://trac.mcs.anl.gov/projects/Heckle/>
- [20] xCat Open Source Project. (2011,May)xCat extreme cloud administration toolkit. [Online]. Available: <http://xcat.sourceforge.net/>
- [21] P. O. S. Project. (2010, Apr.) Perceus provision enterprise resources and clusters enabling uniform systems. [Online]. Available: <http://www.perceus.org/>

AUTHOR:



ANITA AHUJA is an Asst. Professor in Department of Computer Science and Information Technology at Mewar University, Chittorgarh (Rajasthan). She has completed 'A' level DOEACC Society, M.Sc (IT) from M.C.R.P.V, Bhopal,M.Phil, Rajasthan Vidyapeeth, and Udaipur. And M.Tech.(P) at Mewar University, Chittorgarh . Her research interest is in the fields of Network Security, Cloud Computing , Advance Data Structure and Algorithms.