Collaborative Integration and Management of Community Information in the Cloud

Wang Ning, Xu De, Xu Baomin
School of Computer and Information Technology
Beijing Jiaotong University
Beijing, China
{nwang, dxu, bmxu}@bjtu.edu.cn

***Abstract*—The cloud computing paradigm has been receiving much attention recently, and data management applications such as e-business are potential candidates for deployment in the cloud. In this paper, we introduce CloudCDI, which is a platform for collaborative integration and management of community information in the cloud. We present the architecture for CloudCDI, which is based on cloud computing framework to facilitate collaborative management and pay-as-you-go integration of data in data centers. Furthermore, a unified data model, Resource Net Model, is put forward to represent unstructured, semi-structured and structured data inside a single model. We design a hierarchical resource view model to enable best-effort cooperation for community members. In order to access data in CloudCDI conveniently and efficiently, a scaleable query model is also proposed to facilitate users to move seamlessly from one query mode to another.**

Keywords-data integration;cooperation;community information management;cloud computing

I. INTRODUCTION

Recently, the cloud computing paradigm has been receiving much attention not only in industry but also in institute. Cloud computing can make a great shift of computer processing, storage, and software delivery away from the desktop and local servers into next generation data centers hosted by large infrastructure companies such as Amazon, Google, Yahoo, Microsoft or Sun. Data management applications are potential candidates for deployment in the cloud[1]. Some large companies have begun to establish new data centers for hosting cloud computing applications such as social network, business applications, media content delivery, and scientific workflows.

A. Challenges for Community Information managemwnt in the Cloud

There are many communities on the web, each with a specific set of topics such as sciences, medicine, academia, government, and even business. Community members want to upload, share and query data. In general, the data from various sources is heterogeneous, from unstructured through semi-structured to structured. With more and more software applications and hardware infrastructures are moved from private environment to third data centers, it is a challenging problem with respect to how to deploy Community Information Management (CIM) systems in the cloud and enable community members share, collaborate, and query information easily and efficiently through the internet.

In fact, a CIM system based on cloud computing has some fundamental features that distinguish it from paradigms in classical data integration.

First of all, a CIM system should be towards pay-as-you-go data integration to avoid high cost on creating mediated schema and mappings. With increasingly large volumes of data in data centers and especially some data such as biomedical information with very complex structure, it is cumbersome and costly to create a semantically integrated view and precise mappings before community members can utilize the data. A pay-as-you-go data integration and management system takes a data-coexistence approach: it does not require any investments in semantic integration before querying service on the data are provided. Rather, it can be gradually enhanced over time by defining relationships among the data.

Next, because data from various sources may be incomplete, a CIM system should enable best-effort cooperation among community members with respect to maintenance of the data and metadata. Community members can cooperate to complete data and reconcile the conflicts.

Finally, a CIM system should enable users to access and query the system easily, and move seamlessly from one query mode to another. Users can always begin searching what they want using keywords. With time going, the pay-as-you-go system can be enhanced by getting relationships among data by collaborating. At that time, structural search constraints can be allowed to help users to locate what they want more efficiently.

B. Related Work

There is a growing body of work which adapts classical data integration ideas to the community setting. Systems like Orchestra [2] focus on maintaining data utility despite significant disagreement, they do not provide facilities for users to cooperate on managing their data and metadata. Cimple [3] extracts data from pages, provides user services over the data, and then supports mass collaboration. It focuses on uncovering and exploiting the structure "hidden" in unstructured data. Wikipedia and related systems are highly cooperative, however, they only work for unstructured or mildly-structured data, and do not fully support integration. Youtopia[4] is a platform for collaborative management and integration of relational data,

978-0-7695-3997-3/10 $26.00 © 2010 IEEE
DOI 10.1109/ICEE.2010.357

however, it can not include arbitrary data formats and manage the data in its full heterogeneity.

To sum up, all the systems above have not considered how to realize community information integration and management in the cloud computing framework. Furthermore, none of them can include arbitrary data formats and manage the data in its full heterogeneity.

C. Contribution Summary

In this paper, we will introduce CloudCDI, a platform for collaborative integration and management of community information with arbitrary data formats in the cloud. Our main contributions are summarized as below:

- The system architecture for CloudCDI is first presented to enable collaborative integration and management of community information based on cloud computing framework.
- A unified data model named Resource Net Model is put forward, which can represent information with arbitrary data formats in CloudCDI.
- A hierarchical resource view model is presented to enable best-effort cooperation for community members. Every member can only edit his own user resource view, and system will merge changes into group resource view when he decides to check in.
- A scaleable query model is presented to enable users to access data in CloudCDI easily and to move seamlessly from one query mode to another.

The rest of the paper is organized as follows. We start in section 2 by giving a solution framework for CloudCDI. Section 3 introduces a unified data model used throughout this paper. A hierarchical resource view model is put forward in section 4 for best-effort cooperation. We present a scaleable query model in section 5 and conclude the paper in section 6.

II. CLOUDCDI ARCITECTURE

CloudCDI is a platform that allows community members to register, upload, update and share data with arbitrary formats in the cloud in a collaborative fashion. As shown in Fig.1, the architecture of CloudCDI contains 4 layers: data storage layer, communication layer, data exchange and share layer, user interface layer.

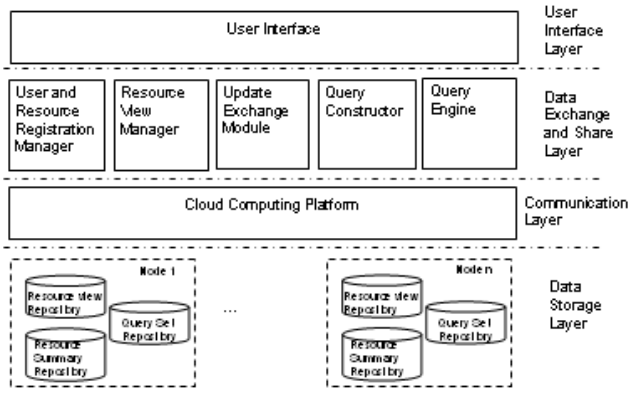

Figure 1. The architecture of CloudCDI

A. Data Storage Layer

Data storage layer provides a logical abstraction of 3 important repositories.

1) Resource view repository: Every community member can upload and share data with others. A user resource view is used to merge user own data with others. Every member can only edit his own user resource view, and merge changes into group resource view only when he checks in his new version. All these resource views are stored in resource view repository.

2) Resource summary repository: To facilitate community members to locate data more easily and efficiently, cloudCDI presents a scaleable query model, which helps user move seamlessly from simple keyword search mode to complex structural query one. To realize this flexible query model, some metadata is needed to help user construct complex query and provide the basis for query optimization. CloudCDI extracts metadata from user and group resource views, then build resource summaries [5] respectively in pay-as-you-go fashion, and store them in resource summary repository.

3) Query set repository: As scientists and others in communities store and share increasingly large volumes of data in data centers, they need the ability to analyze the data by issuing exploratory queries. In this new setting, powerful query management capabilities are necessary for helping users both in query formulation and in deciding which queries to ask. Using query set repository, CloudCDI can store and manage the queries users have issued in order to provide powerful query management capabilities.

B. Communication Layer

Communication layer in CloudCDI takes Hadoop as the task coordinator and network communication layer. Queries are parallelized across nodes using the Map-Reduce framework.

C. Data Exchange and Share Layer

Data exchange and share layer is composed of 5 components.

1) User and resource registration manager: There are many groups in a community, in which members have common interests. Every community member can register as a group member in CloudCDI. User and resource registration manager is used to help users register and upload data in it.

2) Resource view manager: Every community member works on his own user resource view, and a group resource view maintains a consistent version of all users' resource views in the group. Resource view manager help users create and maintain their user resource views in CloudCDI.

3) Update exchange Module: Every member can only edit his own user resource view, and he can merge his changes into the group resource view when checking in. As the basis for best-effort cooperation, update exchange module manages users' updates and takes charge for merging changes from various users into their group resource view.

4) Query constructor: In order to realize automatic query recommendation in CloudCDI, query constructor is needed for

helping users to construct queries according to query condition, query output and other query features.

5) Query engine: CloudCDI provides a scaleable query model for users. Given a query which can be a simple keyword search or a complex structural query, query engine is responsible for selecting optimal logical query plan and then translating it into Map-Reduce execution plan so that the query can be parallelized across nodes in the cloud.

D. User Interface Layer

CloudCDI has convenient user interface to help community members upload, share and query data. They can create user resource views, make editions in the views, merge their changes, submit queries and analyze the results.

III. RESOURCE NET MODEL

CloudCDI is a platform which enables integration and management of community information with arbitrary formats in the cloud. Either structured data as tables, semi-structured data as XML documents, or unstructured data as images can be uploaded by community members in CloudCDI. In order to merge and share various heterogeneous data, a unified data model is needed for representing unstructured, semi-structured and structured data inside a single model.

There were some researches about unified data model for a highly heterogeneous mix of data. iDM is a data Model for iMeMex personal dataspace management system[6]. In iDM, all personal information available on a user's desktop is exposed through a set of resource views. Because the aim of iDM is to remove the boundary between inside and outside a file, relationships, which exist in structured data, such as primary-foreign-key relationship, are neglected in the model. R-radius Steiner Graph, which can model unstructured, semi-structured and structured data as graphs, is proposed in EASE [7] for efficient and adaptive keyword search. However, the granularity of nodes in EASE's graph model is coarse. Especially for unstructured data, the model can not describe semantics but only contents. EASE can support keyword search efficiently by means of its extended inverted index, but it is helpless for complex structural query.

In order to support integration and management of community information in pay-as-you-go fashion, CloudCDI provides a scaleable query model which is based on Resource Net Model.

In Resource Net Model, we use concept resource item to represent any data item uploaded by users in CloudCDI, which can be a tuple in a table, an element in an XML document, a document or an image. Resource items in CloudCDI are linked to each other with resource associations in arbitrary directed graph structures to construct a resource net. We will formally define them as follows.

Definition 1 (Resource Item) A resource item RI is a 5-tuple (ID, n, s, c, l), where ID is an identifier component, n is a name component, s is a structure component, c is a content component, and l is a lineage component. We define each component of RI as follows:

ID: ID is the identifier of RI.

n: n is the name of RI

s: s is a 2-tuple (T,V), where T is defined as a sequence of attributes and V is a sequence of corresponding values for T.

c: c is the content of RI.

l: l is the lineage of RI, which identifies where it is from.

In fact, structure component and content component of a resource item are optional in CloudCDI. For structured resource items such as tuples in a table, content component is unnecessary. However, for unstructured resource items such as documents or images, content component is absolutely necessary, and sometimes structure component is needed for describing structure information which can be extracted from documents or annotated for images.

Definition 2 (Resource Association) A resource association RA is a 2-tuple (RIDm, RIDn), where RIDm, RIDn are identifiers for two resource items.

A resource association can be primary-foreign-key relationships between tuples, parent-child relationships between elements in XML documents, hyperlinks between web pages. Apart from those native relationships existed, CloudCDI allows users to designate resource associations when they cooperate to share and merge data.

Definition 3 (Resource Net) A resource net RN is a 2-tuple (R, A), where R is a set of resource items, and A is a set of resource associations between them.

A resource net is a basic unit for users to upload, merge and query data in CloudCDI. It can uniformly represent heterogeneous resources and associations between them.

IV. RESOURCE VIEW MANAGEMENT

A. Group Resource View and User Resource View

In CloudCDI, each user can register as a member for one or more groups, but only has one user resource view in a group.

A group resource view is a common consistent view for all group members, which can not be edited directly. At the beginning, it is a resource net composed of all resource items uploaded by group members. Each user can create his user resource view for editing, and merge changes into the group resource view only when he checks in his new version.

B. Collaboraion in CloudCDI

A user resource view is the basis for collaboration in CloudCDI. By means of user resource views, users can collect information he cares for from other group members. From the resource net of a group resource view, user can build his own view through operations below.

1) Union: Given two resource nets RN_1 and RN_2, a union operation based on RN_1 and RN_2 can be represented as

$$\Sigma (RN_1, RN_2)$$

The result of union operation is a new resource net, in which resource items and resource associations are the union of resource items and resource associations respectively from RN_1 and RN_2.

2) Selection: Given a resource net RN and a topic K which can be a keyword expression, a predicate expression, or the combination of them, selection operation can be represented as

σ (RN, K)

A resource item n_1 of RN is called concrete resource item if its content component contains some keywords in K and its structure component meets the predicate expression in K, and a resource item n_2 of RN is called steiner resource item if there exist two concrete resource items u and v which can be connected through resource associations, and n_2 is on the path between u and v.

The result of selection operation is a new resource net, which is composed of all concrete resource items, steiner resource items, and associated resource associations in RN.

3) Link: Given two resource nets RN_1, RN_2, and a topic K, a link operation based on RN_1, RN_2, K can be represented as

Ł (RN_1, RN_2, K)

The result of link operation is a new resource net RN' which contains all resource items and resource associations from RN_1 and RN_2. Furthermore, if a resource item n_1 of RN_1 and a resource item n_2 of RN_2 are related to the topic K, a new resource association will be added to RN'.

4) Neighbor: Given a resource net RN and one of its resource item identified by rid, and also the length n of a resource association path, a neighbor operation based on RN, rid and n can be represented as

Φ (RN, rid, n)

A resource item u is the n-radius neighbor of a resource item v if u can be reached from v through no more than n resource associations. Of course, the n-radius neighbors of a resource item include itself.

The result of neighbor operation is a new resource net, which is composed of n-radius neighbors of the given resource item and resource associations between them.

Users collect data they care for by building user resource views. Based on it, they can make editions such as correcting errors, completing structure component for unstructured data, merging resource items, appending resource associations, etc. The changes can be seen by others only when the user decides to check in his new version. In CouldCDI, uncertainty and conflicts can be reconciled by this kind of cooperation.

V. SCALEABLE QUERY MODEL

As a pay-as-you-go system in which semantics can be enhanced with system running, a scaleable query model is needed in CloudCDI for enabling users to move seamlessly from simple keyword search mode to complex structural query one. We will use some examples to explain how query capability can be enhanced gradually in CloudCDI.

(1) keyword query

"cloud computing": for this query, CloudCDI return the result of a selection operation, which is based on current resource net and topic "cloud computing".

(2) keyword & structural query

author="Stephen" and "cloud computing": for this query, CloudCDI return the result of a selection operation, in which concrete resource items are those with "cloud computing " in their content components and having the value "Stephen" for attribute "author" in their structure components.

(3) path query

books/*[author="Stephen" and "cloud computing"]: for this query, CloudCDI will first get the result of neighbor operation based on current resource net , the resource item named "books", and the path length 1. Then, the same selection operation as example (2) will be executed on the result of above neighbor operation.

(4) extended path query

books/{3}/*[author="Stephen" and "cloud computing"]: for this query, CloudCDI will first get the result of neighbor operation based on current resource net , the resource item named "books", and the path length 3. Then, the same selection operation as example (2) will be executed on the result of above neighbor operation.

The query capability in CloudCDI is scaleable. Users can use keyword query in any case. However, as long as semantics is enhanced in the system, users can move gradually to issue more complex structural queries.

VI. CONCLUSIONS

In this paper, we introduce the architecture for CloudCDI, which is based on cloud computing framework to facilitate pay-as-you-go integration and collaborative management of data in data centers. A unified data model, Resource Net Model, is put forward to represent unstructured, semi-structured and structured data. We also design a hierarchical resource view model to enable best-effort cooperation for community members. Finally, a scaleable query model is proposed to help users to access data easily and efficiently in CloudCDI. In the future, we will continue the work on query optimization for the scaleable query model, and will study powerful query management strategy to facilitate query recommendation in CloudCDI.

REFERENCES

[1] Daniel J. Abadi, "Data management in the cloud: limitations and opportunities," IEEE Data Eng. Bull., vol.32, no.1, pp.3–12, 2009.

[2] N. E. Taylor and Z. G. Ives, "Reconciling while tolerating disagreement in collaborative data sharing," in SIGMOD, pp.13–24, 2006.

[3] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R.McCann, M.Sayyadian and W. Shen, "Community information management," IEEE Data Eng. Bull., vol.29, no.1, pp.64–72, 2006.

[4] Łucja Kot and Christoph Koch, "Cooperative update exchange in the Youtopia system," in VLDB, August, 2009.

[5] Wang Ning and Xu De, "Resource summary for pay-as-you-go dataspace systems," in ICSP2008, pp.2842–2845, October, 2008.

[6] J. P. Dittrich and M. A. V. Salles, "iDM: a unified and versatile data model for personal dataspace management," in VLDB, pp.367–378, September, 2006.

[7] G. Li, B.C. Ooi, J. Feng, J. Wang, and L. Zhou, "Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data," in SIGMOD, June, 2008.