Research on Construction Methods of Big Data Semantic Model

Li Kang, Li Yi, LIU Dong

***Abstract*—Big data integration process faced structural heterogeneity and semantic heterogeneity problems. An ontology is an explicit specification of a conceptualization, which is the core of semantic web technologies that can be used to describe the semantics of the metadata. This paper builds a big data model based on ontology by exploiting semantic web technology, and propose an ontology-based semantic model and ontology-based Key/Value storage model, meanwhile giving detailed steps for building the model. The model proposed try to solve the issue of understanding between heterogeneous data systems. Finally, we give an example of the semantic model application, which has been verified on HBase-based prototype system.**

***Index Terms*—big data, ontology, semantic model, RDF, Key/Value**

I. Introduction

Recently, Big Data has been getting increasingly attention and recognition due to its broad research and application prospects [1]. Big data generally refers to large-scale, rapid changes in many types. The data sets are usually integrated data from different sources in a structured, semi-structured, and unstructured data collection. The unstructured and semi-structured data more than 85%. Big Data applications often involve multiple data sources, there are structural heterogeneity and semantic heterogeneity problems between the data sources [2]. Refers to structural heterogeneity of different data stored in the data model could not be directly mapped to each other. Semantic heterogeneous data that describe the terms inconsistent with each other could not understand each other, and unable to reflect the link between the data sets. How to solve the problem of structural heterogeneity and semantic heterogeneity is one of the major challenges facing big data.

There are already some big data applications research will introduce the Semantic Web technology, semantic links, the establishment of a unified data model for big data, such as RDF (Resource Description Framework) model. RDF model is a simple ontology model can provide analysis based on the "knowledge" [4]. Ontology is the core of the Semantic Web technology, which is also an important method to conceptualize domain knowledge and modeling that can be used to describe the semantics of the data. The introduction of ontology to integrate thinking, intelligent retrieval of big data, can provide a new idea for big data research and application. Currently, the research by combining the ontology with the big data application is still rare. This paper aims at the characteristics of big data, building a data model based on ontology, proposed a solution to the structural heterogeneity and semantic heterogeneity of big data.

Manuscript received February 06, 2014; revised February 14, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 60904082.

Li Kang is with the Academy of Equipment, Beijing 101416, China (phone: 86-13520876718; fax: 86-10-66364116; e-mail: xtlikang1987@126.com).

Li Yi is with the Academy of Equipment, Beijing 101416, China (e-mail: ylili@139.com).

Liu Dong is with the Academy of Equipment, Beijing, 101416 China (e-mail: ld4m@139.com).

II. Related Works

The problem of heterogeneity in big data has aroused great concern researchers all over the world. The key to solving the problem lies in the establishment of a unified data model. We need to extract information from the data source and the data associated with the integration and aggregation, and define a unified structure to store [5]. The research of data extraction and integration technologies in the field of traditional database has been relatively mature. In the perspective view of the data model, with a sharp increase in the continuous emergence of new data species, data types, commonly used relational model, extended relational model, object-oriented model, ER model and hierarchical data model and other traditional models are no longer suitable for solving the problem of big data.

Reference [6] based on a relational database, proposed a new management of unstructured data with a structured approach, using the relational model represents descriptive information of unstructured data. Reference [7] adding new fields in the two-dimensional table structure of the relational model, the use of extended relational model to establish a unified data model representing the metadata and data links. These two methods of managing unstructured data to make a meaningful attempt, but the drawback is that not an accurate representation of the complex unstructured data based on the relational model. Reference [8] try to improve against the traditional ER model, proposed a graph data model based on the content support to represent unstructured data. Reference [9] for unstructured data representation, aimed at the problem of manipulation and retrieval, proposed a hierarchical data model. Both models solved the problem of unstructured data some extent, but not they are still not an excellent solution to represent unstructured data relationship between the various components of unstructured data.

Since the W3C (World Wide Web Consortium) has proposed RDF, some scholars have adopted the RDF model to solve the problem in heterogeneous multi-source data

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

integration process. Reference [10] dissertated RDF model in data integration, pointed out the semantic web technologies bringing a new research ideas to the field of data integration. First, the existing data integration system building ontology model to improve the system functions and enhance search efficiency. Second, integrate data sets base on the RDF data model and research mass storage and retrieval technology. Such developments have caused ICDE, DESWeb, VLDB, SIGMOD research institutions and other related high priority, and the research results generated, including RDF storage method and search algorithm based on SPARQL [11].

Reference [12] improved the integration of traditional data extraction, transformation, and loading process. The paper proposed an additional step in the process of extracting data, and transformed data into a unified representation and RDF management model. And the model drawn from different data sources including structured data, semi-structured data and unstructured data. The paper maps the structured data to RDF using R2RML (RDB to RDF Mapping Language) [13] language developed by the W3C's RDB2RDF team. Reference [14] proposed a semantic enhancement algorithm. It use the semantic capabilities from the Interactive Knowledge Stack (IKS) project into existing CMS (Content Management Systems) applications, adding semantic information to data for easy data retrieval and mining. The algorithm is based on the Apache Stanbol [15] framework, combined with LMF (Linked Media Framework) [16] applications to achieve. Wherein, Apache Stanbol is an Apache top-level project, which is evolved from the RDF model provides a semantic content management software stack designed for modular and reusable components. LMF is a server application easy to configure, which uses the core technology to provide advanced intelligent semantic web services.

Most of these studies proposed is to improve the algorithm for their own fields. Although the massive data processing have also been involved, they did not establish a common data model conforms to the characteristics of big data. The main contribution of this paper is to build a big data semantic model base on ontology, and gives data integration method based on the mapping process.

III. Research of big data model Based on ontology

An ontology is an explicit specification of a conceptualization [17], are abstract modeling of the things in the real world such as concepts, constraints, and identity. Due to the big data integrated from heterogeneous data sources, the data could not be shared and understand each other. So this paper proposed ontology-based approach to building big data semantic models, to solve the problem of structural heterogeneity and semantic heterogeneity in big data.

A. Analysis of big data modeling

Data model is a collection of data description, data relationship, data semantics and conceptual tools of consistency constraints. It provide a design approach contain description of the physical layer, the logical layer and the view layer. Traditional data model could be divided into four categories: the relational model, the entity-relationship data model, object-oriented data model, semi-structured data model. Type of data model closely related to the data management system where located. Traditional database management systems mostly used the relational model and the entity-relationship model. In recent years, SimpleDB and BigTable as the representative cloud-based data management system, are used to manage big data. With the corresponding Key/Value data model becoming a new data model to bring researchers new ideas.

Big data generated from the scientific (astronomy, biology, high-energy physics, etc.), computer simulation, Internet applications, e-commerce, and many other areas, and the quantity of data reached PB level. Data sources include sensor data, web clickstream data, mobile device data, radio frequency ID data, etc. The types of data are complex and diverse structures, and its numerical ranges are flexible. Relational data model pursuits the high degree of consistency and correctness, but its drawback is the limitation in extension. In addition, high availability is a necessary condition for big data processing system. Therefore, the database management system based on relational model could not competent to big data analysis.

MapReduce technology is a significant technology to solve big data, and it usually includes three levels [18]: distributed file systems, parallel programming model, and parallel execution engine. MapReduce parallel programming model for the calculation process is divided into two main phases, namely Map phase and the Reduce phase. Map function handles Key/Value pairs, produce a series of intermediate Key/Value pairs, Reduce function is used to merge all intermediate keys have the same Key value pair, and then calculate the final result. This is a simple parallel computation model, which from the system level to solve the scalability, fault tolerance and other issues, by accepting user-written functions and Map Reduce function automatically on scalable parallel execution of large-scale clusters, thereby big data can be processed and analyzed. Therefore, considering the big data modeling, we try to build a big data model on this basis meeting the MapReduce technical framework.

We analysis storage and calculate of big data above. The following analysis is how to combine ontology technology, to build a data model conforms in line with the MapReduce framework, to solve problems of structural heterogeneity and semantic heterogeneity.

B. The method of constructing big data model based on ontology

Formal definitions of ontology are described in this section, combined with the characteristics of big data processing systems, we proposed the semantic model in line with the Key/Value framework.

Semantic model ontology is usually mixed by global ontology and local ontology. This section establishes the semantic model of the entire system by associating the global ontology and local ontology. W3C is given four languages to describe the ontology, namely the RDF, RDFS (RDF Schema), DAML (DARPA Agent Markup Language) + OIL

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(Ontology Inference Layer), OWL (Web Ontology Language). In this paper, we choose OWL, which is a W3C recommended standard ontology description language. Here is an example about IT big data generated by e-commerce site, which gives specific steps to build ontology semantic model.

Step1. Construct class of the model

In the ontology, the most basic class is Thing by default, and all individuals in the model are Thing's subclass. First, build the fundamental class, including products, transactions, and other participants. Set value of the URI (Uniform Resource Identifier) http://www.mmexample.com/saleweb in the example. A URI can identify all system resources uniquely, such resources can be specific or abstract, which can be present or absent. OWL language expressed as follows (for the sake of brevity, we assume that all examples point to the namespace http://www.mmexample.com/saleweb):

<owl:class rdf:about="product" />
<owl:class rdf:about="deal" />
<owl:class rdf:about="participant" />

Base class includes multiple levels of sub-categories, sub-categories belonging to the class of its parent, the concept of a subclass can be further refined. For example, product category, including computers, mobile phones, cameras, office equipment and other sub-categories, including trading in order to return a single, single-service and other sub-categories, class participants including manufacturers, distributors, consumers and other sub-category. Due to space limitations, the following gives OWL expression of computer class. Computer subclasses can be divided into notebook computers, desktop computers, tablet computers and notebooks second son category, expressed as follows:

<owl:class rdf:about="computer" >
<rdfs:subClassOf rdf:resource="product" />
</owl: class>
<owl:class rdf:about="notebook" >
<rdfs:subClassOf rdf:resource="computer" />
</owl: class>

Since the subclass established, the system could recognize "computer" as a product and "notebook" as a computer, and system could infer that "notebook" is also a product.

Step2. Structural properties of the model

An ontology could describe the relationship between individuals and individuals with values by defining attribute. Expression between different entities called ObjectProperty, and the concept of object properties is to reflect external attributes, as we defined hasProduce, hasSale, hasBuy three attributes representing the relationship between product object classes and three sub-classes participants class about manufacturers vendors, consumer. The hasProduce property represents the relationship between product and manufacturer, the system could infer a certain product produced by some manufacturers, we can express it as follows:

<owl:ObjectProperty rdf:about="hasProduce">
<rdfs:domain>
<owl:Restriction>
<owl:onProperty rdf:resource="hasProduce"/>
<owl:someValuesFrom rdf:resource="manufacturer"/>
</ owl: Restriction>
</rdfs: domain>
<rdfs:range>
<owl:Restriction>
<owl:onProperty rdf:resource="hasProduce" />
<owl:someValuesFrom rdf:resource="product" />
</owl: Restriction>
</rdfs: range>
</owl: ObjectProperty>

The attribute expresses the relationship between entity and values called DatatypeProperty (data type attributes). The data type of the attribute reflects the intrinsic properties of the concept, as defined computer_parameter data type properties and its three sub-attributes hasCPU, hasRAM, hasHD, representing computer parameters CPU, memory and hard disk parameters. The domain of hasHD is set to hasHD which is subclass of computer_parameter, its range is limited to a string type, expressed as follows:

<owl:DatatypeProperty rdf:about="hasHD">
<rdfs:subPropertyOf rdf:resource="computer_parameter" />
<rdfs:domain rdf:resource="hasHD" />
<rdfs:range rdf:resource="&xsd;string" />
</owl: DatatypeProperty>

Properties can increase the complexity of the type and constraints to enhance semantic system. Due to space limitations in this case, we only give a summary of this description.

Step3. Construct individual of the model

The individual is an instance of the class in the ontology, the instance can represent any specific things in real-world. In this case, in order to describe the performance parameters T430 laptop , to establish in the sub-class notebook members T430, we add data type properties include hasCPU, hasRAM, hasHD, and set its value to i5-3210M, 8G, 500G. Meanwhile, the introduction T430 pictures, videos, user comments link by defining hasPIC, hasVID, data type properties hasCOM added to the individual, expressed as follows:

<owl: NamedIndividual rdf:about="T430">
<rdf: type rdf:resource="notebook" />
<hasCPU> i5-3210M </ hasCPU>
<hasRAM> 8G </ hasRAM>
<hasHD> 500G </ hasHD>
<HasPIC> T430/pic001.jpg </ hasPIC>
<hasVID> T430/vid001.avi </ hasVID>
<hasCOM> T430/com001.txt </ hasCOM>
</owl: NamedIndividual>

Individual is the main component of the information system, including complicated information about specific things. Due to the huge amount of data of the individual, it is difficult to create each individual by artificial means. The individual could be generated by mapping from relational

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

databases, spreadsheets, documents and other data sources. Since individual mapped to a specific class directly, and individuals are added the attributes and constraints of specified class directly.

The above introduce the basic steps for building ontology semantic model in detail, and we create the ontology by using protégé editor, as shown in Fig 1. Each specific thing in the system corresponds to an only URI through integrating classes, properties and individuals into the ontology framework. So the relationship between things, the information about properties and other information could understand each other, solving the semantic problem between the various data sets.

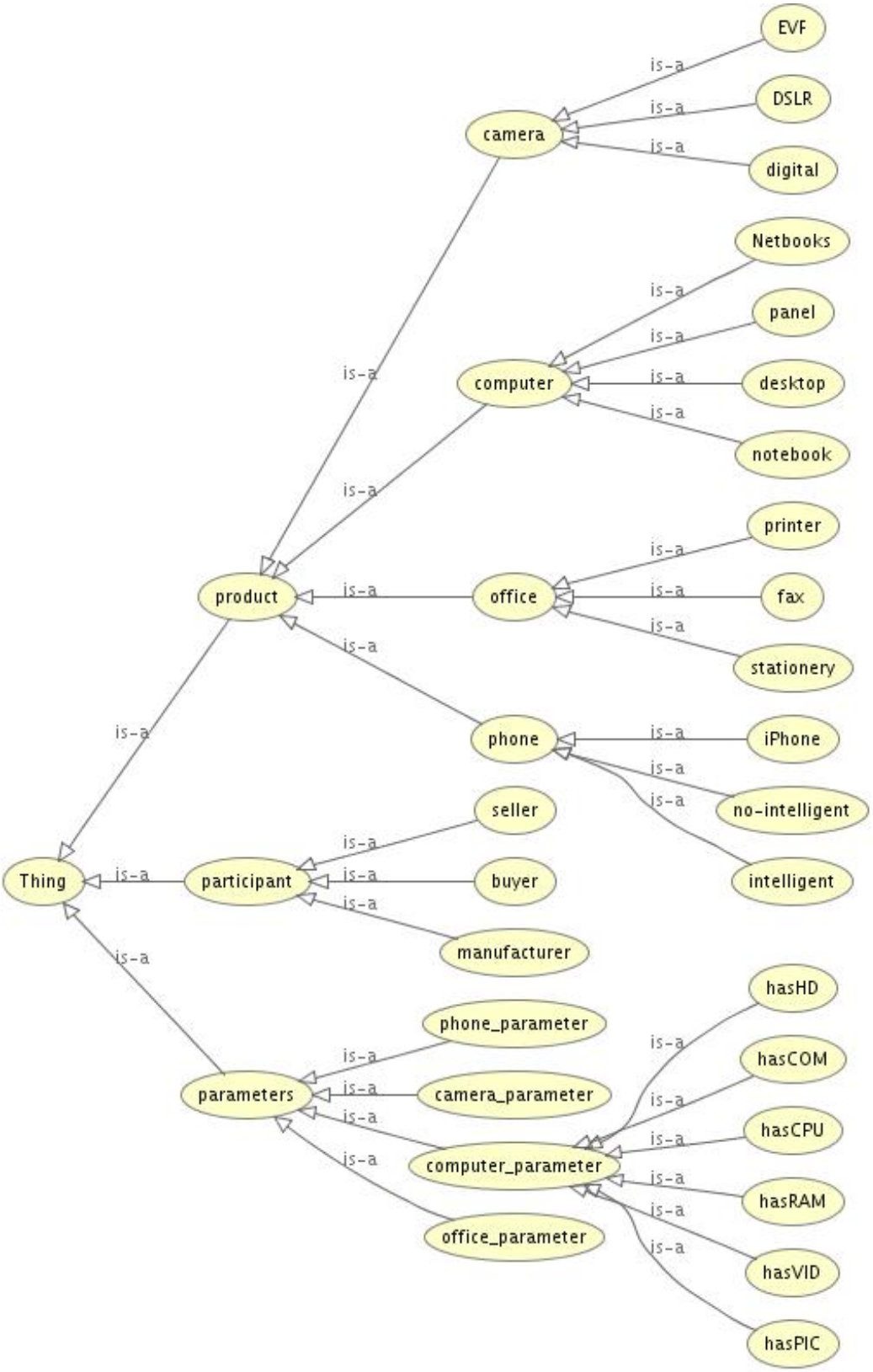

Fig1 SalesWebsite's ontology semantic model

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

C. Ontology-based Key/Value Storage Model

Big data's storage, query, analyze system should have the features of high scalability (to meet the growing needs to the amount of data), high performance (to meet the real-time and high-performance query processing data for reading and writing), fault tolerance (to ensure the availability of distributed systems), scalability (distribution according to need resources) and operational costs as low as possible [20]. In recent years, a new model for big data management, resulting in Google's BigTable [21], based on Hadoop HDFS [22] of HBase [23], Amazon's Dynamo [24], Apache's Cassandra [25] based on a series of cloud computing NoSQL (Not Only SQL, non-relational database) products, and provides a platform for technical support addressing the big data storage, query and analysis. In this paper, we established Key/Value-based data model for big data based on NoSQL platform and NoSQL management system, so in this section we study Key/Value storage model based on ontology.

As the semantic model ontology defined in Section B, we defined all things by URI in system, and each local system's ontology files are named with a unique namespace URI. Local ontology files refined set of individual-level granularity, and the resource file are stored as the Key/Value model in NoSQL. Key/Value data model is actually a map that Key is to find a unique key for each data address, Value is the content of the data is actually stored. This example URI resource (also the name of the body of the document) is set to Key, Value URI that is the corresponding file. Table I choose T430 notebook as an example to describe instances of data which stored in Key/Value model.

As shown in Table I, in this case the Key/Value data model, Key represents a unique identifier for each resource and data address table. Due to space limitations, we use URI represent http://www.mmexample.com/saleweb. Value is a collection of columns, which includes the hasCPU, hasRAM, hasHD, hasPIC, hasVID, hasCOM, and Key's multiple column represent the value of the data within the meaning of different attributes, which hasPIC, hasVID storing unstructured pictures and video format data, they may be stored in the address table or stored by binary code directly. It solved the problem of storing unstructured data. User comments are stored in column hasCOM, we can always add a new comment content to the system. For system upgrades, facing the problem of extended attributes, we could add columns in Key/Value data model to solve the problem.

In this example Key/Value data model is mapped through multi-storage format to simulate the traditional table which shows the concept of relationship model in the relational database and is easy to maintain and query data. It has a simple and flexible scalability, and can solve the problem of heterogeneous data stored in the data stream for meeting the dynamic operation.

IV. System Implementation and Verification

This section is still as an IT e-commerce site for example, we study prototype system for implementation and validation. HBase is a distributed database built on Apache Hadoop, which is an open source implementation of the various functions and features Google's Bigtable, and is currently one of the most popular NoSQL databases. It has been increasingly used in terms of e-commerce, search engines, social networks, etc. Therefore, this section builds a HBase-based prototype system to verify the feasibility and effectiveness of big data model based on ontology.

A prototype system based on cloud -based Hadoop cluster, the cluster set up a master node, 10 working nodes, each node of the system is basically configured as a dual-core CPU@3. 2GHz, 4G RAM, 500G hard disk space. Using Hadoop version is 0.20.2, HBase uses version is 0.92.0.

Accordance with the OWL ontology model previously described, create two HTable tables: HBClass table and HBProperty table, used to store information in this case classes and attributes. HBClass table stored class information, class name as Row-Key, set another two columns family: SubClass, Property sub-classes used to store attribute information and class attributes. Column family data can be added and updated in real time by time stamp, meet data dynamically changing needs. Specific storage structure is shown in Table II. Attribute information HBProperty table storage definition, attribute names as Row-Key, the family consists of two columns: SubProperty, Individual, sub-attributes and attribute values stored property. Specific memory structure as shown in Table III.

Based on the above two HTable table, create an instance of the system tables HBInstance, record specific information for each instance. The URI instance as Row-Key, set two column family HBClass, HBProperty, Deal, Participant, class information is used to store instances, property information, transaction information, evaluating information. As Mobile phone Mi2, camera D80, computer T430, office equipment hp2050 sample for instance, stored in the system, as shown in Table IV. For simplicity, the table gives only a partial list of products subclass information.

Through implementation of the prototype system, all the information the site unified managed by using HBas. The data are consistent with RDF semantic model, solved problem of mutual understanding between data. The data is stored in a Key / Value model, could be handled directly by HBase NoSQL databases which is easy for data updating dynamically, meet the needs of high concurrent data processing in a big data environment.

Table I Instance of data storage in Key/Value model

Key	Value					
	hasCPU	hasRAM	hasHD	hasPIC	hasVID	hasCOM
URI#T430-1/	i5-3210M	8G	500G	URI#T430/001.jpg	URI#T430/001.avi	URI#T430/001.txt
URI#T430-2/	i5-3210M	8G	500G	URI#T430/002.jpg	URI#T430/002.avi	URI#T430/002.txt
URI#T430-3/	i5-3210M	8G	500G	URI#T430/003.jpg	URI#T430/003.avi	URI#T430/003.txt

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table II HBClass storage structure

Row-Key Class	Column Family SubClass	Column Family Property
Product	Phone	Phone_Parameter
	Camera	Camera_Paramete
	Computer	Computer_Parameter
Deal	Manufacturer	hasProduce
	Saler	hasSale
	Buyer	hasBuy
Participant	PIC	hasPIC
	VID	hasVID
	COM	hasCOM

Table III HBProperty storage structure

Row-Key Parameter	Column Family SubProperty	Column Family Individual
Phone_Parameter	hasCPU	CPU GHZ
	hasRAM	RAM G
	HasROM	ROM M
Camera_Parameter	hasPixel	Pixel X
	hasVR	VR N
	hasCMOS	CMOS M*N
Computer_Parameter	hasCPU	CPU GHZ
	hasRAM	RAM G
	hasHD	HD G

Table IV HBInstance Instance table

Row-Key Instance	Column Family HBClass/ Product	Column Family HBProperty	Column Family Deal/ Manufacturer	Column Family Participant/ COM
URI#/Phone/ Mi2/	Phone	Phone_Parameter	China	This is a good phone.
URI#/Camera/D80/	Camera	Camera_Paramete	Japan	D80 is Stop production.
URI#/Computer/T430/	Computer	Computer_ Parameter	China	It's my best friend!
URI#/Office/ hp2050/	Office	Office _Parameter	USA	My office use it.

V. Conclusion

This paper analyzes the problem of structural heterogeneity and semantic heterogeneity in the big data integration process. Combing the Semantic Web technology to solve some of the research results in data integration, we proposed a big data Semantic model based on ontology. By building the ontology between the semantic model to solve the problem of data incomprehensible, we constructed a Key/Value storage model based on ontology to solve the problem of heterogeneous data storage, and build a prototype system based on HBase implementation and verification. The proposed model is the exploration by using semantic web technology to solve the problem of big data, the next step of our research tries to solve the mapping problem in integration from existing database systems to big data management system.

References

[1] Labrinidis, Alexandros, and H. V. Jagadish. Challenges and opportunities with big data. Proceedings of the VLDB Endowment 5 (12), 2012: 2032-2033.
[2] Guo-Jie Li. Scientific value of big data research [J]. Chinese Computer Society Newsletter, 2012, 8(9): 8-15.
[3] Gong Xue-qing, Kim Chul-ching, Wang Xiaoling and other data-intensive science and engineering: needs and challenges [J] Journal of Computers., 2012, (08): 1563-1578.
[4] Liu Wei, Xiacui Juan, Zhang Chunjing big data associated with the data: Data technology revolution is coming [J] Library and Information Technology，2013(4): 2-9.
[5] X. Meng, kind of large data management: concepts, technologies and challenges [J] Computer Research and Development, 2013, 50(01): 146-169.
[6] Doan A, Naughton J, Baid A, et al. The case for a structured approach to managing unstructured data[J]. arXiv preprint arXiv:09091783, 2009.
[7] Srivastava, Divesh, and Yannis Velegrakis. Intensional associations between data and metadata. Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM, 2007.
[8] Siadat, Mohammad-Reza, et al. Data modeling for content-based support environment (C-BASE): application on Epilepsy Data Mining. Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on. IEEE, 2007.
[9] Amato, Giuseppe, Giovanni Mainetto, and Pasquale Savino. An approach to a content-based retrieval of multimedia data. Multimedia Tools and Applications 7.1-2 (1998): 9-36.
[10] Hassanzadeh, Oktie, Anastasios Kementsietsidis, and Yannis Velegrakis. Data management issues on the semantic web. Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, 2012.
[11] SPARQL. [EB/OL]. [2013-06-23]. http://www.w3.org/TR/rdf-sparql-query/.
[12] Lopes, Nuno, et al. On the Semantics of Heterogeneous Querying of Relational, XML and RDF Data with XSPARQL. EPIA. 2011.
[13] R2RML: RDB to RDF Mapping Language. [EB/OL]. [2013-06-22]. http://www.w3.org/TR/r2rml/
[14] Damjanovic, Violeta, et al. Semantic enhancement: the key to massive and heterogeneous data pools. Proceeding of the 20th international IEEE ERK (electrotechnical and computer science) conference. 2011.
[15] Apache Stanbol. [EB/OL]. [2013-06-23]. http://stanbol.apache.org/index.html
[16] Linked Media Framework. [EB/OL]. [2013-06-25]. https://code.google.com/p/lmf/
[17] Gruber TR. A translation approach to portable ontology specifications[J]. Knowledge acquisition, 1993, 5(2): 199-220.
[18] Qin Xiong faction, the king will be held, DU Xiao-Yong and other large data analysis--RDBMS and MapReduce competition and symbiosis [J] Journal of Software, 2012, 23(1), 32-45.
[19] Gómez-Pérez, Asunción, and Richard Benjamins. Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. IJCAI and the Scandinavian AI Societies. CEUR Workshop Proceedings, 1999.
[20] Rys M. Scalable SQL. Communications of the ACM, 2011,54(6):48−53.
[21] Chang F, Dean J, Ghemawat S, et al. Bigtable: A distributed storage system for structured data. In: Proc. of the OSDI. New York: ACM Press, 2006.
[22] Borthaku D. The hadoop distributed file system: Architecture and design. 2009. http://hadoop.apache.org/common/docs/r0.18.0/ hdfs_design.pdf
[23] Hbase Development Team. Hbase: Bigtable-like structured storage for hadoop HDFS. 2009. http://wiki.apache.org/hadoop/Hbase
[24] De Candia G, Hastorun D, Jampani M, et al. Dynamo: Amazon's highly available key-value store. In: Proc. of the SOSP. New York: ACM Press, 2001. 205−220.
[25] Lakshman A, Malik P. Cassandra—A decentralized structured storage system. 2009. http://www.cs.cornell.edu/projects/ladis2009/ papers/lakshman-ladis2009.pdf

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)