AUDR: An Advanced Unstructured Data Repository

Xianglong Liu, Bo Lang, Wei Yu, Junwu Luo, Lei Huang
State Key Laboratory of Software Development Environment
Beihang University, P.R.China
{xlliu, langbo, yu_wei, luojunwu, huanglei}@nlsde.buaa.edu.cn

Abstract

Unstructured database serves as a new database targeted mainly at unstructured data management, which addresses the limitations of relational database. In this paper, an unstructured database management system named Advanced Unstructured Data Repository (AUDR) is introduced. AUDR is designed to manage massive and various types of unstructured data including text, image, audio and video. Based on a uniform data model named the Tetrahedral Data Model proposed recently, a scalable architecture is designed to provide storage, process and mining functions for massive complex unstructured data. To support content-based retrieval, intelligent retrieval and associated retrieval, it defines and implements intelligent query language of unstructured data by extending XQuery language. As a typical unstructured data, image management including storage, retrieval and clustering is described in detail. Finally the system evaluations prove that AUDR serves as a novel and efficient unstructured data management system.

1. Introduction

As the information age is coming, the amount of information increases at a terrific rate. In addition to web pages, there exists a large amount of images, audios, videos, and data from social networks, deep web and mobile Internet. Also, the rate of data growth is accelerated and it follows the Moore's Law: doubling every 18 months. According to a study of IDC and EMC [1], 1,800 EB (1EB = 1,000 PB) digital information will be produced in 2011, and the amount of information will increase tenfold from 2005 to 2011. Traditional relational database [2], by introducing relation model, relational algebra and relational calculus of mathematics, after dozens of years of application and development, has laid its own advantages on structured data management. However, according to Gartner Group statistics, 80% of today's data is unstructured data, which are from rich sources, contain complex content and have different structures. The traditional relational database for these complex types of unstructured data has been powerless.

Take image data, the typical unstructured data, as an example. The traditional image management method is based on textual retrieval, namely keywords tagging. Although this method is simple, the textual description of the image is difficult to fully express the rich content. In addition, due to continuing expansion of data amount, the manual annotation consumes too much time and labor. In 1990s, Content-Based Image Retrieval (CBIR) emerged [3, 4]. The basic idea is to search image by analyzing its visual content like color, shape and texture. Google, Baidu and other search engine companies have been gradually trying to establish CBIR to provide efficient and accurate image retrieval. DB2 of IBM also proposed Extender mechanisms to support unstructured data management, which serves as an unstructured data processing extensions and plug-ins of structured database management systems. Some content management companies including Autonomy begin to enter markets and play roles in the content-based management of unstructured data.

Unstructured database is targeted mainly at unstructured data management. Compared with the current relational database, the biggest difference is that it broke the limitations of relational database which is not easy to define the structure and to change the data length [5]. And unstructured database supports duplicate fields, child fields and variable length fields. When handling the sequence information such as full-text information and unstructured information of duplicate data and variable-length data, unstructured database has incomparable advantages than the traditional relational database.

However, due to heterogeneous and complex structures of different types of unstructured data, state-of-art unstructured database systems cannot manage all types very well. A uniform data model is essential to unify descriptions of all types of unstructured data. Also as the data amount reaches 10^{18} bytes level, it is impossible for these systems to provide satisfying service. To handle these massive complex data, a scalable architecture with distributed storage and processing should be established and support a variety of data management functions.

In this paper, based on analysis of data model, an unstructured database named Advanced Unstructured Data Repository (AUDR) is designed and implemented. First a uniform data model named Tetrahedral model [6] for unstructured data is proposed to unify the description of various types of unstructured data and

978-1-4577-0208-2/11/$26.00 ©2011 IEEE

intrinsic link between the components. Based on the Tetrahedral model, we propose a scalable architecture, which not only adapts to the emerging unstructured data, but also supports the distributed massive unstructured data storage and computing under multiple operating system platforms [7]. In this architecture, "Three-dimensional form" [8, 9] is applied to storage management in three dimensions: the data objects, attributes and time. The data is stored as an object to support any format of unstructured data. Meanwhile, in order to meet requirements of massive data storage and processing, AUDR implements distributed file storage system, and distributed file indexing and retrieval [10, 11, 12].

AUDR also achieves data mining including unstructured data automatic association, classification, clustering and summarization [13, 14]. It defines and implements intelligent query language of unstructured data [15, 16]. This language supports query by semantic and low-level features independently, associated retrieval between them, and intelligent operations like clustering. In addition, a series of data management including storage management, model management, tool management, security management, version management and metadata management are integrated in AUDR.

The rest of the paper is organized as follows: In Section 2, the scalable architecture of AUDR is presented. Section 3 introduces the uniform data model named Tetrahedral model. An intelligent query language is designed and its functions will be briefly described in Section 4. Then Section 5 will give details of one typical module: image retrieval. Finally, we evaluate the system in Section 6 and conclude in Section 7.

2. Scalable architecture of AUDR

Traditional relational database is expert in management and operation of structured data. For unstructured data, specialized data management techniques need be studied. The aim of Unstructured data management system is not only to deal with structured data (such as numbers, symbols and other information) but also handle unstructured data (text, image, sound, video, hypermedia and other information.) Compared with the popular relational database, the biggest difference is that it breaks the limitations in relational databases that the definition of database structure is not easy to alter and data length is fixed. Also it has the advantage in dealing with continuous information and unstructured information. Our system, Advanced Unstructured Data Repository (AUDR), is a database management system, which is applied to management of unstructured data, namely text, image, audio and video. The design of AUDR follows several principles:

- First the system needs to uniformly describe, integrate and manage massive unstructured data and can support multiple development platforms and technologies.
- The system should have high-performance to quickly response user request and capability of massive data storage and high-speed processing.
- The system should provide intelligent operations and language on unstructured data, and support various types of user-friendly interfaces.

The architecture of AUDR is shown in Figure 1, composed of physical layer, data accessing layer, processing layer, and user interface layer. Data model and query language are the logical basis for unstructured data management in AUDR, based on which, all AUDR modules are designed. Data model is a formal representation based on data abstract, and query language is the formal representation based on data operation abstract. The underlying data model of AUDR named Tetrahedral Data Model was proposed in [6] by Wei Li and Bo Lang. Unstructured data can be described by a tetrahedral, with facets standing for basic attribute, semantic feature, low-level feature and raw data respectively.

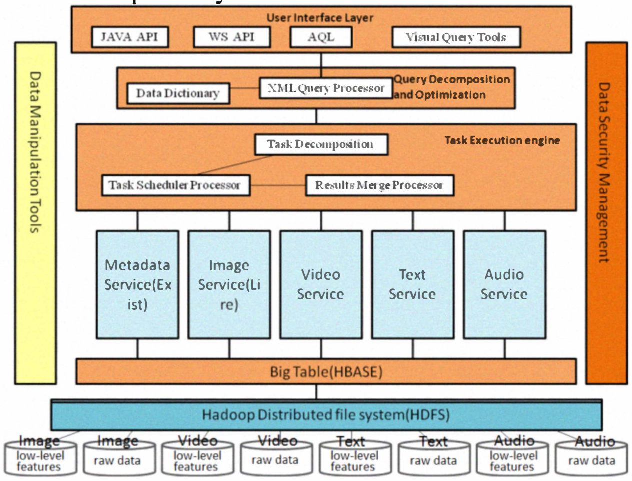


Figure 1. Scalable architecture of AUDR

Unstructured data management faces storage and processing of rich types of unstructured data, therefore in physical layer a loosely coupled storage system should be built to support distributed storage [7, 8, 9] and distributed computing [10] and thus to achieve user-oriented distributed query.

Data accessing layer is responsible for data storage and retrieval. There are four independent unstructured data search engines in this layer mainly for a variety of different types of unstructured data including image, text, audio and video. Unstructured data search engines achieve various retrieval methods like content based [3, 4], basic properties based and semantic based retrieval. In order to speed up the content-based retrieval efficiency, the engine employs both Map-reduce [10] mechanisms and hashing based index [11] to achieve distributed parallel and sublinear retrieval of massive unstructured data. For basic properties based and semantic based retrieval, meta data is used to describe the management and semantic information. It can be automatically extracted when data import or labeled by users. Moreover, a serials of operations including data classification, clustering and automatic association are realized [13, 14].

In the processing layer, the control server takes the responsibility of parsing the user requests, and calls the interfaces provided by the search engines in data accessing layer to carry out data operations.

The top layer named user layer provides interfaces easy to use, including AQL intelligent query interface, programming interface and Web Service interface.

AUDR also provides typical functions of data management, namely meta data management, storage management, model management and security management, and a range of user-friendly tools: AUDR management tool, unstructured data migration tools, backup and recovery tools, and query analysis tools, through which users and administrators can manage unstructured data.

3. Data model of AUDR

Unstructured data, such as text, image, audio and video, has a non-uniform structure, and is stored as raw data. Therefore, it cannot be understood and processed directly by computers. In order to manage unstructured data, the fundamental approach is to describe the data and then to use the descriptive information to implement data operations. Keywords based on semantic descriptions, descriptions of low-level features, or concept based semantic descriptions are presently used to describe unstructured data. Unstructured data is composed of basic attributes, semantic features, low-level features and raw data, and there are relationships between the elements of these components:

- Basic attributes: All kinds of unstructured data have attributes such as name, type, author, and time of creation. However, it should be noted here that basic attributes do not include the semantics of the data.
- Semantic features: Special semantic properties are expressed using text, including the intention of the author, subject explanation, and the meaning of low-level features.
- Low-level features: Low-level features include properties of unstructured data acquired by using special data processing techniques, such as color, texture, and shape for images.
- Raw data: Raw data refers to the stored files of unstructured data.

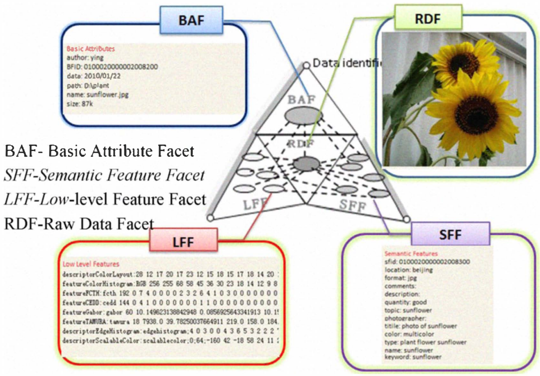

BAF
Basic Attributes
author: ying
BFID: 0100020000002008200
data: 2010/01/22
path: D:\plant
name: sunflower.jpg
size: 87k
RDF
Data identifi
BAF
RDF
LFF
SFF
BAF- Basic Attribute Facet
SFF-Semantic Feature Facet
LFF-Low-level Feature Facet
RDF-Raw Data Facet
LFF
Low Level Features
descriptorColorLayout:28 12 17 20 17 23 12 15 18 15 17 18 14 20
featureColorHistogram:RGB 256 255 68 58 45 36 30 23 18 14 12 9 8
featureFCTH:fcth 192 0 7 4 0 0 0 0 2 3 2 6 4 1 0 3 0 0 0 0 0 0 0
featureCEDD:cedd 144 0 4 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
featureGabor:gabor 60 10.149623138842948 0.0856925643341913 10.1
featureTAMURA:tamura 18 7938.0 39.78250037664911 219.0 158.0 184.
descriptorEdgeHistogram:edgehistogram;4 0 3 0 0 4 3 6 5 3 2 2 2
descriptorScalableColor:scalablecolor;0;64;-160 42 -18 58 24 11
SFF
Semantic Features
sfid: 0100020000002008300
location: beijing
format: jpg
comments:
description:
quantity: good
topic: sunflower
photographer:
titile: photo of sunflower
color: multicolor
type: plant flower sunflower
name: sunflower
keyword: sunflower

Figure 2. Tetrahedral model for image

Based on above analysis of unstructured data, the uniform data model named Tetrahedral Data Model was recently proposed by Wei Li and Bo Lang [6]. The tetrahedral data model presented in this paper characterizes unstructured data based on these four aspects. The tetrahedral model is composed of a vertex, four facets and the lines between the facets, as shown in Figure 2. The vertex represents the unique identifier for the unstructured data; the bottom facet represents the raw data; the three side facets represent the basic attributes, the semantic features and the low-level features separately, and the lines connecting facets represent the associations between the elements on each facet.

4. Unstructured data query language: AQL

Since the basic attribute and semantic feature facets in the tetrahedral model are stored in the XML format [15], it is natural to use the XQuery language [16] to retrieve the related information. However, additional functions such as low-level query and intelligent query cannot be implemented by XQuery, it is necessary to extend it. Therefore, we develop an XQuery based query language AQL, which is short for Advanced Query Language, in order to accelerate the query in AUDR.

4.1. Extended functions based on XQuery

As mentioned above, the common XQuery can be used to retrieve files in XML format, but could not realize the content-based video and audio query. Besides basic query on basic attribute or semantic feature, AQL extends functions based on XQuery language shown in Figure 3.

(1) Low-level Query: In the tetrahedral data model, the facets of basic attribute, semantic feature and low-level feature, the former two are described in XML files, while the latter one not. Therefore, the query by basic

attribute and semantic feature can be realized by XQuery language, but the query by low-level feature is what to be extended in AQL. Input a sample data, such as an image, a piece of music, a short video and so on, and low-level query will return the most matches data according to the basic feature.

(2) Intelligent Query: It contains functions like query results clustering, classification and analysis. They operated on semantic feature and low-level feature and thus could not be realized by XQuery. Intelligent Query can combine the basic attribute, semantic feature and low-level feature to realize the associated query to make data retrieval both quickly and precisely.

(3) Multi-facet Query: The multi-facet query is the associated query on basic attribute, semantic feature and low-level feature. According to the tetrahedral data model, the four facets of unstructured data are stored by four files respectively. Besides, there is a file storing the associated relationship among these facets. Reults of the associated query should be merged from results of query on each facet.

In XQuery language, there are two ways to deal with this problem. One is to use multiple XQuery clauses. Every XQuery clause operates a file, and then merge the results in an external function. The other is to use nesting query in a single clause, which makes multi-facet query too complicated, therefore it is necessary to extend XQuery to support the multi-facet query function. Such query synthetically utilizes basic attributes, semantic features and low-level features to locate data fast.

(4) Multi-tetrahedron Query: It means to carry out associated query on different types of unstructured data, which involves the operation on multiple file collections. Using the semantic feature, multi-type unstructured data can be associated, and then multi-tetrahedron query can be extended in AQL.

(5) Composite Query: By combining or nesting the functions mentioned above, the composite query is implemented.

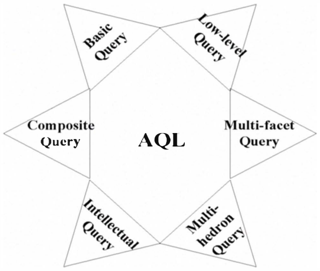


Figure 3. AQL extended functions

4.2. Grammar definition of AQL

AQL grammar is formed by extending and simplifying XQuery grammar according to functional requirements in Section 4.1. Here are definitions of some key clauses in the grammar:

(1) outermost FLWOR expression

FLWORExpr ::= ForClause LetClause WhereClause
OrderByClause ReturnClause (FilePathClause)

The structure of the outermost layer in AQL is based on the FLWOR expression of XQuery language, but appends a clause that specifies the input file path.

(2) inner nested FLWOR expression

FLWORExprInner::= ForClause (LetClause) WhereClause
OrderByClause ReturnClauseInner (FilePathClause)

Compared with outer FLWOR expression, inner nested one only returns identifiers of the retrieved raw data, not the specified data itself.

(3) “for” clause

ForClause ::= "for" "$" VarName "in"
TypePath(“(”FLWORExprChanged”)”)

“for” clause specifies the range of the data type involved in multi-facet retrieval and can nest FLWOR clause to achieve multi-hedron retrieval.

(4) “let” clause

LetClause ::= "let" "$" VarName ":=" LocalPart (",$"
VarName ":=" LocalPart)*

“let” clause defines the path variables involved in multi-facet retrieval and these variables can improve the XML database query efficiency.

(5) “where” clause

WhereClause::="where" (“BA{”condition”}”
(“SF{”condition”}”) (“LF{”lfcondition”}”)) |
(“SF{”condition”}”(“LF{”lfcondition”}”)) |
(”LF{”lfcondition”}”}

“where” clause defines the constraints in multi-facet retrieval. Corresponding to facet types, there are three constraints: basic attributes constraints, semantic features constraints and low-level feature constraints.

(6) “order by” clause

OrderByClause ::= (“ba””{”(("order" "by") | ("stable"
"order" "by")) TypePath OrderSpecList”}”)
(“sf””{”(("order" "by") | ("stable" "order" "by")) TypePath
OrderSpecList”}”)

“order by” clause defines the ranking order of the return results. There are two types of ranking order: ranking by basic attributes and by semantic features.

(7) “return" clause

ReturnClause ::= "$" VarName LocalPart
(IntelligenceOperation(“,” IntelligenceOperation)*)

“return” clause specifies the content of return value and intelligent retrieval is extended in return grammar.

5. Image retrieval in AUDR

In AUDR, based on Tetrahedral data model and distributed storage, the design and implementation of text, image, audio and video retrieval are similar. Due to the limitation of space, in this paper we only give a comprehensive illustration of image retrieval, omitting details of the rest.

The image module consists of data storage, retrieval and analysis. The storage module extracts the visual feature and semantic feature in parallel [12], imports both data and features into distributed file system and builds file index for retrieval. The retrieval module provides semantic retrieval, instance retrieval and associated retrieval and accelerates query response using index. The analyzing module plays a role in clustering and classifying data.

5.1. Storage

The visual features like shape, color and texture are suitable for describing the visual contents and thus demonstrate a good representation capability. Visual features used in the AUDR are extracted by using LIRe library [17], including Color Histograms in RGB and HSV color space and MPEG-7 descriptors [18].

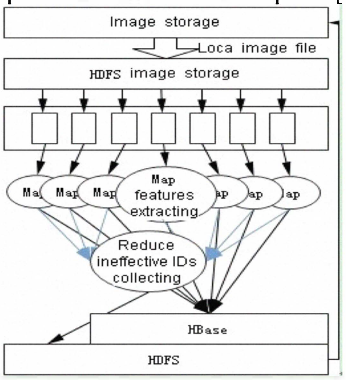


Figure 4. Parallel image storage

Visual feature extraction is time-consuming. What AUDR deals with is massive images, and together with their visual features. AUDR stores massive raw image data in a distributed file system and the extracted information in three dimensional table named HBase [8]. Data is logically organized as table, row, family and column in HBase. HBase sparsely stores data rows in labeled tables.

AUDR adopts MapReduce computing model [10] to extract the visual features of images and then write the features and image files all into HBase parallelly. The implementation method of distributed image storage is shown in Figure 4.

After extracting features and importing data, for each visual feature AUDR constructs a hashing based index. We build the nearest neighbor training dataset using a part of imported data, then learn the hashing functions on the dataset. With these hashing functions, each data in AUDR can be encodes as a binary string. For each binary string, a bucket is used to record data having the similar codes. The hashing based index reduces the storage space of massive data, while guarantees the retrieval efficiency and accuracy [11].

5.2. Retrieval

For semantic retrieval, AUDR adopts popular TF-IDF based textual retrieval methods like BM25, which satisfy the time and precision requirements.

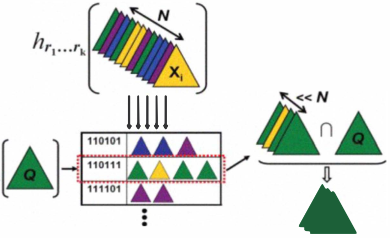


Figure 5. Image retrieval using hashing

For instance retrieval, the system should calculate the similarity between the query image and the target images, and then returns those images matching the sample image most closely. Because the target database contains large datasets, it will spend long time on whole-table search. Two solutions are provided in AUDR.

First a MapReduce computing model is taken to execute map tasks and reduce tasks parallelly to reduce running time and increase efficiency. In each task, feature matching is completed and top ranked results are transmitted to the reduce task. Then in reduce task, map results are merged with reranking and returned.

Another solution is to retrieve images using hashing based index [11]. When given the query image, the module encodes it using the pre-trained hashing functions. Then the query image can be represented by a binary codes and images falling in the bucket of index corresponding to the codes are the results candidates as shown in Figure 5. A fully similarity reranking can be used to reorder these candidates, where the top ranked are the final results.

5.3. Image clustering

Because of the ambiguity of natural language, results of semantic retrieval are often not the satisfying, while due to semantic gap, low-level retrieval returns a number of visually similar but semantically irrelevant images. Therefore search results of different themes shown together takes users a lot of time to locate images they need. For this problem, AUDR establishes a multi-view image-stage clustering framework. With full use of the image multi-modals including semantics and visual features, the framework employs an improved two-stage partitioning and hierarchical clustering algorithm to dig deep relationships between the images, to re-organize search results and thus improve the retrieval efficiency.

Two-Phase Clustering Method (TPCM) is proposed and implemented in this system which is essentially a combination of the division and hierarchical methods. Tanimoto coefficients and cosine distances measure low-level and semantic similarities individually. It consists of two phrases. In the first phrase, one applies the k-means algorithm [13] to the input data to get m classes. In the second phrase, we implement an improved CURE [14] algorithm on the m classes until we finally obtain k classes. TPCM leverages the high efficiency of k-means and the high quality of CURE, and thus it gives a better clustering result in a relatively short time, compared to the two methods. The procedure of TPCM is illustrated in Figure 6. Since CURE uses several data points to represent one class and modifies the status of the classes by the shrinkage factor, thus it is able to handle the data obeying the non-ball distribution, yielding a higher quality of clustering.

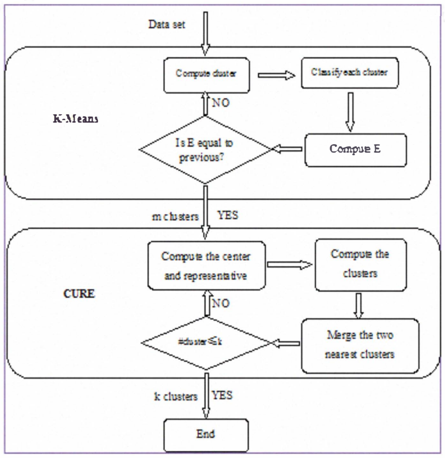


Figure 6. Two-Phase clustering method

6. System evaluation

In this section, we would like to give an overall evaluation of the system. However due to the limit space and similar design, we still take image management as an example to test the performance of both the retrieval and AQL. Besides, we also test the performance of image clustering and classification.

The Hadoop [7] cluster in which AUDR is deployed on consists of 9 nodes. Each node has 4G memory, Core 2 CPU @3GHz and 500G hard disk.

6.1. Image retrieval

6.1.1. Image import. Figure 7 shows the total time consumed to store images with respect to their number. Although MapReduce framework should spend time on initializing the job and exchanging data among clusters, as the number of images increases, the advantage of MapReduce in processing large data is becoming obvious. That is because image processing time takes more percentage in the total time with the increasing number of images. Image storage in centralized way tends to grow linearly, while storage based on MapReduce tends to grow much slower.

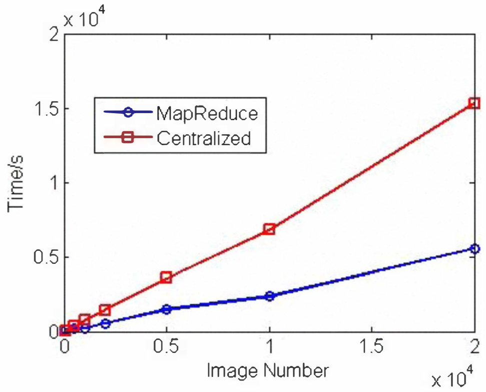


Figure 7. Image storage evaluation

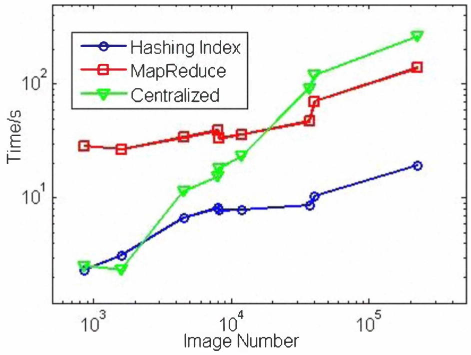


Figure 8. Image retrieval evaluation

6.1.2. Image retrieval. Retrieval performance comparison is conducted among retrieval using index, parallel image retrieval and centralized retrieval. We made experiments on 100 query images. As Figure 8 shows, both parallel retrieval using MapReduce ("MapReduce" for short) and retrieval using hashing index ("Hashing Index" for short) outperform centralized way, and "Hashing Index" performs better than "MapReduce". The time consumed by "Hashing Index" increases very slightly with respect to image number and stays below 20s for even 20,000 images, while that of centralized retrieval ("Centralized") tends to grow linearly. For "MapReduce", when the scale of the data enlarges, it gradually shows superiority but is inferior to "Hashing Index". On average, the speedup of "MapReduce" to "Centralized" reaches 2 when image number increases, while "Hashing Index" can exceed 10. Therefore, it can be concluded that two proposed solutions: "Hashing Index" and "MapReduce" in AUDR accelerate the efficiency greatly.

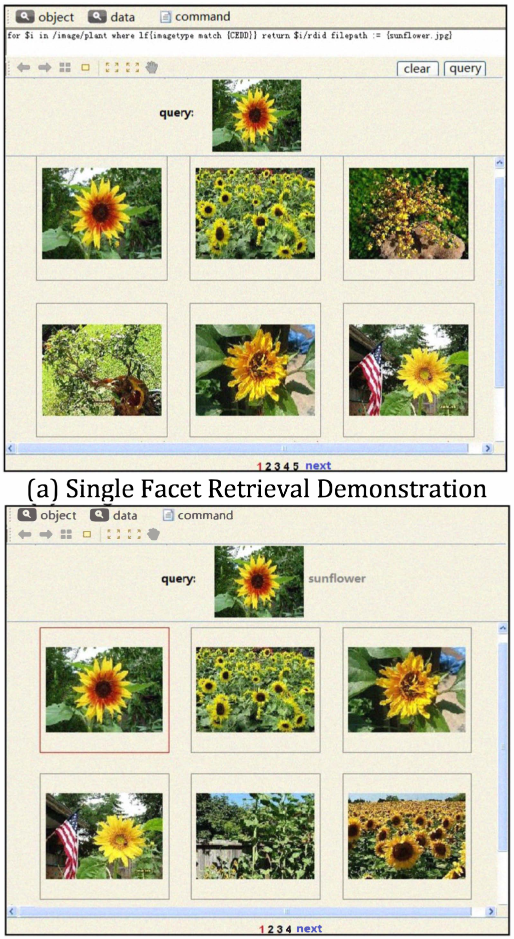
6.1.3. AQL functions. The following cases illustrate the specific AQL clauses and their corresponding result.

(1) Single Facet Retrieval

Single Facet Retrieval is retrieval based on the single facet of the tetrahedral, such as the low-level feature facet. In AQL both basic query and low-level query are single facet retrieval. We choose the semantic feature as the query facet, and the related test AQL clause is :

for $i in /image/plant where lf{imagetype match {CEDD}} return $i/rdid filepath := {./sunflower.jpg}

The clause means that we retrieve images in the plant category that are similar to the input instance "./sunflower.jpg" using the low-level feature CEDD. The results are shown in Figure 9(a), where the returned pictures are ranked according to the similarity to the instance in a descending order. However, the result contains other plants visually similar to sunflower. In the following experiment, we will show that this problem can be solved by Associated Query in AUDR.



```
for $i in /image/plant where lf{imagetype match {CEDD}} return $i/rdid filepath := {sunflower.jpg}
```

(a) Single Facet Retrieval Demonstration

(b) Associated Retrieval Demonstration

Figure 9. AQL functions evaluation

(2) Associated Retrieval

We carry out experiments on AQL functions of associated query on the different facets of a single tetrahedron. The semantic feature and low-level feature are associated in this experiments, and the related AQL clause is

for $i in /image/plant where sf{./keywords = "sunflower"} lf{imagetype match {CEDD}} return $i/rdid filepath := {./sunflower.jpg}

The above clause means that the system should return images labeled keywords "sunflower" in plant category and similar to the query image "./sunflower.jpg" using the low-level feature CEDD. As seen from Figure 9(b), the returned images are all sunflowers, which means the multi-facet associated query can provides more accurate results than the single facet query.

6.2. Clustering

6.2.1. Image clustering. To compare the performance of the proposed TPCM with both K-Means and CURE, we conduct experiments on CMU Content-based image retrieval database and use F1 and time as the measurement. The image data set including a total of 273 samples divided into eight categories.

On the same dataset, we run each of the three methods 10 times with the same specified K, and take the average as the final result. Clustering results of these methods are shown in Table 1, according to which TPCM achieves the highest F1 value while the time is slightly more than K-Means. This indicates that TPCM utilizes the advantages of both K-Means and CURE and can obtain better clustering results in a short time.

Table 1. Comparison with fixed K

Methods	*F1-measure*	*Time(s)*
K-Means	0.678	0.357
CURE	0.711	6.391
TPCM	0.716	1.415

Without specified K, we compare three methods again. The number of categories obtained by these methods, F1 values and time are shown in Table 2. TPCM clusters images into 7 categories which is closest to the true number of categories among the three methods. Moreover, it again reaches the highest F1-measure while the execution time is very low.

Table 2. Comparison without specified K

Methods	Categories	F1-measure	Time(s)
K-Means	10	0.678	0.357
CURE	3	0.711	6.391
TPCM	7	0.716	1.415

6.2.2. AQL functions. Here we evaluate the clustering AQL function on results of the multi-facet associated query. The related AQL clause is:

for $i in /image/plant where lf{imagetype match {CEDD}} return $i/rdid cluster by lowfeature filepath := {./sunflower.jpg}

The meaning of the above clause is that the system first retrieves images similar to query image "./sunflower.jpg" using low-level feature CEDD in plant category, and then clusters the results according to the low-level feature. The clustered result is shown in Figure 10, where all the retrieved images are divided into six classes. We can see that images in the same class are very similar.

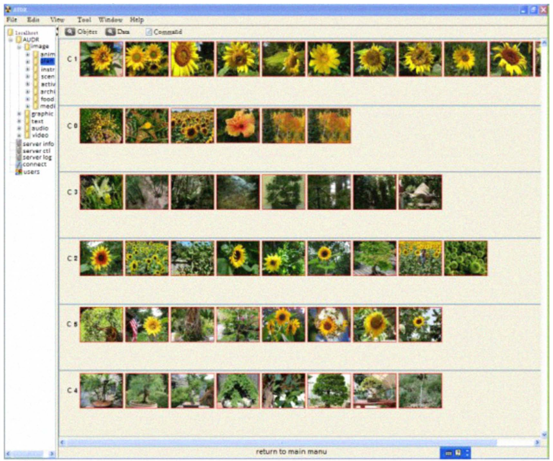


Figure 10. Image clustering demonstration

7. Conclusion

In this paper, an unstructured database named AUDR is introduced, which is a platform of the massive and heterogeneous unstructured data management. In AUDR, firstly a uniform data model named Tetrahedral model is proposed to unify the description of various types of unstructured data and their intrinsic relationships. Based on Tetrahedral model, we propose a scalable architecture scalable architecture to handle storage and process of massive complex unstructured data. AUDR not only enables efficient storage, indexing and access of unstructured data, but also achieves data mining including automatic association and clustering. It defines and implements intelligent query language of unstructured data. This language supports query by semantic characteristics and low-level features independently, associated retrieval between low-level and semantic features, and intelligent operations like clustering. In addition, a series of data management including storage management, model management, tool management, security management, version management and metadata management are integrated in AUDR. The system evaluations prove that AUDR serves as a novel and efficient unstructured data management system.

8. Acknowledgement

This work was supported by the National Major Research Plan of Infrastructure Software under Grant No.2010ZX01042-002-001-00 and the Foundation of the State Key Laboratory of Software Development Environment under Grant No. SKLSDE-2011ZX-01.

9. References

[1] John F. Gantz. "The Expanding Digital Universe". IDC, 2007.

[2] Oracal, "A Comparison of Oracle Berkeley DB and Relational Database Management Systems", An Oracle Technical White Paper, March 2009.

[3] Hirata K., Kato T., "Query by Visual Example – Content-Based Image Retrieval", EDBT, 1992.

[4] Flickner M., "Query by image and video content-The QBIC System", *IEEE Computer*, 1995, 28 (9):23-32.

[5] Doan A, Naughton J F, Baid A, et al., "The case for a structured approach to managing unstructured data", Conference on Innovative Data Systems Research, Asilomar, 2009.

[6] W. Li and B. Lang, "A tetrahedral data model for unstructured data management," *SCIENCE CHINA INFORMATION SCIENCES*, 2010, 1497-1510

[7] http://hadoop.apache.org/hdfs/.

[8] http://hbase.apache.org/.

[9] Chang F., Dean J., et al., "Bigtable: A Distributed Storage System for Structured Data", OSDI, 2006.

[10] Dean J., Ghemawat S., "Map Reduce: Simplified Data Processing on Large Cluster", OSDI, 2004.

[11] Qin Lv, William Josephson, at al., "Multi-probe LSH: efficient indexing for high-dimensional similarity search", VLDB, 2007. 950-961.

[12] Jing Zhang, Xianglong Liu, Junwu Luo, Bo Lang, "DIRS: Distributed Image Retrieval System Based on MapReduce", ICPCA, 2010. 93-98.

[13] Hartigan,J.A., "A k-means clustering algorithm", *Applied Statistics*,1979,p100-108.

[14] Guha, Sudipto; Rastogi, Rajeev; Shim, Kyuseok. "CURE: An Efficient Clustering Algorithm for Large Databases", *Information Systems* , 2001, 35–58.

[15] Wolfgang Meier.eXist, "An Open Source Native XML Database", 2002.

[16] Chamberlin D., "XQuery A query language for XML W3C working draft", 2005.

[17] Mathias Lux, Savvas A. Chatzichristofis, "LIRe: Lucene Image Retrieval - An Extensible Java CBIR Library", ACM, 2008.

[18] S.-F. Chang, T. Sikora, and A. Puri., "Overview of the mpeg-7 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 2001, 11(6):688--695.