ИНФОРМАТИКА

UDC 025.4.03:[004.4:351.852]

S. Anggai, I. S. Blekanov, S. L. Sergeev

INDEX DATA STRUCTURE, FUNCTIONALITY AND MICROSERVICES IN THEMATIC VIRTUAL MUSEUMS

St. Petersburg State University, 7–9, Universitetskaya nab., St. Petersburg,
199034, Russian Federation

Emergence of digital data in the digital age is inevitable, many people rely on search engines for searching information on the Internet. Cultural exhibitions have long ago appeared in virtual museums online as public services which provide a great deal of digital information about collections. In this work we will develop our methods to design and combine services for accessing data in a virtual museum, index data structure with a ranking system and microservices architecture. The digital transformation phenomena are changing institutional innovation and creativity, the museum institution must provide good impact, experiences and values to their visitors, therefore our methodology to develop, organize and integrate a theme in museums institution as an approach to build the concept of Thematic Virtual Museums which serving and providing their visitors powerful relevant information of museum's collections. Refs 17. Figs 5. Table 1.

Keywords: collection, inverted index, thematic virtual museum, information retrieval, microservices.

С. Анггаи, И. С. Блеканов, С. Л. Сергеев

СТРУКТУРА ДАННЫХ ИНДЕКСА, ФУНКЦИОНАЛЬНОСТЬ И МИКРОСЕРВИСЫ В ТЕМАТИЧЕСКИХ ВИРТУАЛЬНЫХ МУЗЕЯХ

Санкт-Петербургский государственный университет, Российская Федерация,
199034, Санкт-Петербург, Университетская наб., 7–9

Появление цифровых данных в эпоху цифровых технологий неизбежно, так как многие люди для поиска информации в сети Интернет полагаются на поисковые системы. Не исключением является и оцифровка данных разных культурных мероприятий (выставок) в виде публичных онлайн-сервисов, получивших название виртуальные музеи, которые уже давно предоставляют в качестве общественных услуг доступ к большому объему цифровой информации. В настоящее время феномен цифровой трансформации оказывает непосредственное влияние на инновационную и творческую деятельность различных государственных и коммерческих учреждений. Это относится и к учреждениям музейно-

Anggai Sajarwo — postgraduate student; sajarwo@gmail.com
Blekanov Ivan Stanislavovich — PhD of technical sciences, associate professor; i.blekanov@gmail.com
Sergeev Sergei Lvovich — PhD of physical and mathematical sciences, associate professor; slsergeev@yandex.ru

Анггаи Сажарво — аспирант; sajarwo@gmail.com
Блеканов Иван Станиславович — кандидат технических наук, доцент; i.blekanov@gmail.com
Сергеев Сергей Львович — кандидат физико-математических наук, доцент; slsergeev@yandex.ru

https://doi.org/10.21638/11701/spbu10.2018.104

го типа, которые для посетителей должны обеспечить удобный доступ к своим ресурсам, опыту, знаниям и культурным ценностям. Потому предложенная методология разработки, организации и интеграции тематическо-ориентированных направлений в учреждениях музейного типа основывается на концепции построения тематических виртуальных музеев, которая обслуживает посетителей и предоставляет им доступ к релевантной информации о коллекциях музея. Рассматриваются разработанные в статье методы проектирования и комбинирования программных сервисов доступа к данным виртуального музея, а также структура данных индекса с системой ранжирования и архитектурой микросервисов. Библиогр. 17 назв. Ил. 5. Табл. 1.

Ключевые слова: коллекция, инвертированный индекс, тематический виртуальный музей, информационный поиск, микросервисы.

1. Introduction. Museum as nonprofit institution which responsible for preserving historical and cultural heritage, exhibit tangible and intangible to public is facing huge problem in digital era as mention in [1]. Museums institution are starting digitizing their collections and describe detail information in order to intensify their activities and present on the Internet as a part of public services. However, digital transformation is inevitable, museums institution must provide good impact, experiences and values to their visitors.

The thematic concept in virtual museum is one of the approaches we take in order to accelerate digital transformation of museums institution. We are combining virtual museum (VM) and information retrieval (IR) concepts in conducting information management, analytics, and focus on users-centric to enhance their experience by providing visitors' information need within the virtual museums system.

In this research work we are preparing, investigating and conducting experiments on data access service, index construction, applying ranking algorithms and development of microservices in thematic virtual museums for speeding up disruptive innovation in museums institution.

2. Data crawling. In section 2 we will describe how this application data access crawling service get the content of web pages. This crawler service can take data from museums institution in text, Hypertext Markup Language (HTML), JavaScript Object Notation (JSON), or Extensible Markup Language (XML) form. We define specific web Uniform Resource Locator (URL) to crawl and run the service to obtain content pages. The crawler service maintains and run a job for specific website and not for general uses, because it ought to crawl based on characteristics of each page in those websites. As in our case we are getting the data from Registration Museum under Ministry of Education and Culture, Republic of Indonesia. There are two models web pages structure in order to get full museum collection pages as the following.

1. List of collections. The crawler service directly accesses to the URL which contains list of museum collections. This URL address show 20 items per page and navigation page to identify how many pages still remain as previous or next page from the first to the end page. All URL retrieved from this page which identified as detail collection URL will be adding to download-queue.

2. Detail of collection. This page contains detail collection information and it can be marked as end of page to be crawled. The information which contain on this page is name of museum institution, collection type, function, age, description and picture with a title.

The crawler services automatically download the content page until all pages have been crawled. In order to prevent Internet network traffic, we are providing delay-time parameter to the crawler services, therefore this engine can be customizing to aggressive or polite crawler modes. The algorithm we are using on this task is Breadth First Search [2], however the page of collection is not deep, therefore to download those pages the formula can be written as shown below:

$$f(p) = \sum_{i=1}^{n}\sum_{j=1}^{m} dl(page).$$

3. Parsing data. In this stage the service will parse data which has been obtained. There are two models we have been using in order to parsing data from crawler service, first we are directly parsing HTML format into Thermal VM (TVM) data schema, and second the parser service follow standard rule for data exchange. Parser service has developed for performing tasks in general, therefore data collections which obtained by crawler can be fully implemented on this service. We are using method as shown in Figure 1.

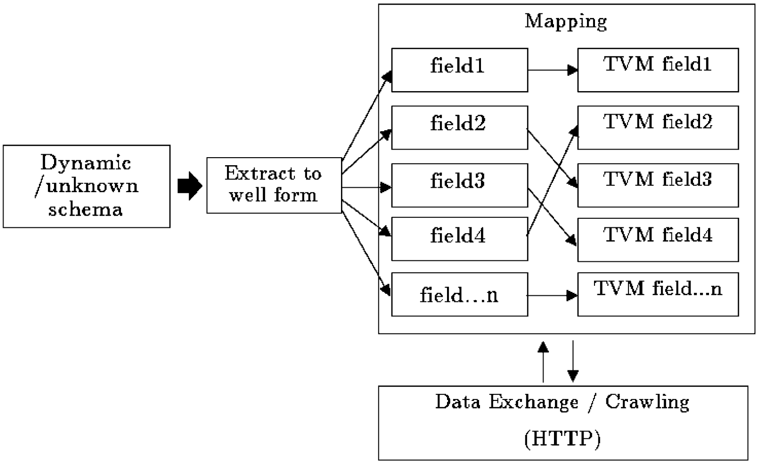

Figure 1. Extracting and mapping dynamic schema to TVM schema

The parser service in Figure 1, show step by step to exchange data where this service can be receiving and parsing XML or JSON format in dynamic schema includes Conceptual Interchange Documentation — Conceptual Reference Model, Lightweight Information Describing Objects (LIDO), and Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), however, we must define dynamic or unknown schema and extracting this schema to well form. Data exchange can be starting when the well form schema is similar with TVM schema, if not then mapping the well form schema to TVM schema should be done, only then data exchange can be performed through Hypertext Transfer Protocol (HTTP) web services.

4. Data store. An important part of data-access service is storing and maintaining data collections which have been crawled and parsed to the standard data structure form. There are many techniques to preserve document collections, one of them is storing into databases. To store data from crawler service, we are using non-traditional database key-value MongoDB for handling structure, semi-structure and unstructured data. There are many advantages of MongoDB as mention in [3–5], this type NoSQL document-oriented database using JSON-like format called Binary JSON (BSON), support for partition and MapReduce. MongoDB is using document store model, allow developer to create free-schema, running on multiplatform and opensource.

In our case we have developed a system to access MongoDB database server using Mgo (mango) driver for Go programming language. The Mgo driver is providing simple application programming interface, easy to use, and fast enough for performing task such as create, read, update, delete operations[1]. The design of TVM database schema for museum collections [6] as shown in table is independent and can be customized depend on the goals of spesific index task.

Table. **TVM database schema for document collections**

N	Key Name	Description
1	_id	Unique object identification
2	iddata	Identification of each data sources
3	idinstitution	Identification of museum institution
4	name	Object name or title
5	regcode	Registration code
6	category	Information about category or type of the object
7	collector	Person who collect the object
8	datefound	Date when the object has founded
9	placesfound	Places when the object has founded
10	period	Period or year of the object event happened
11	age	Age of the object
12	dimensions	Dimensions of the object
13	weight	Weight of the object
14	material	Materials forming or made of the object
15	condition	Describe about past and recent object condition
16	totalcollection	Total object in the museum institution
17	description	Detail object description
18	ref	Reference information about where the data have been taken
19	creator	Creator of the document record in database

5. Forward index. There are several IR techniques for document indexing system, one of them is forward index [7], which is fast to perform task in document indexing[2]. We are taking the museum collections data from MongoDB and managing those collections in TVM forward index. We have modified standard forward index data structure according to TVM data structure needed as shown in Figure 2.

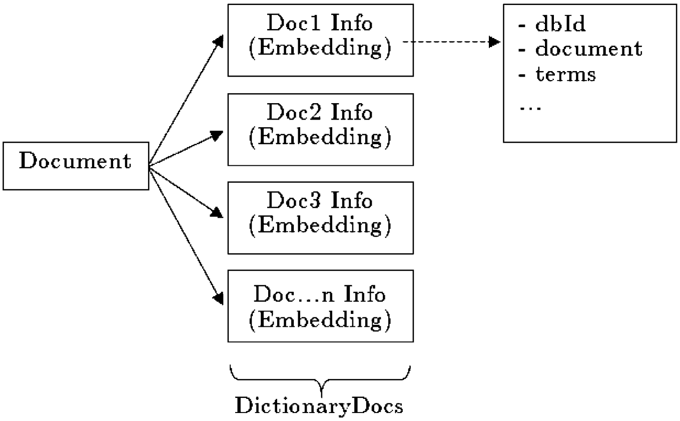

Figure 2. Design TVM forward index

[1] https://labix.org/mgo

[2] https://www.elastic.co/blog/found-indexing-for-beginners-part3

TVM forward index is pair document-payload using unique document identifier (DocId) for each document collections, in order to gain fast access to a document we are using hash map. Payload or embedding information in this forward index can be contains document identifier, list of terms contains in document, a record document information from database, and some necessary information. The development on flexible payload as our approach in TVM forward index give us space to embed something useful in managing and interacting with TVM inverted index.

6. Inverted index. TVM concept which focus on fastest retrieving information need must be designed with good data structure, therefore we have created inverted index as a base for maintaining and searching information from any type application or users queries. There are many standard inverted index models have studied in [8–10], based on those state of the art we are modifying and designing inverted index structure for TVM need [11] as shown in Figure 3.

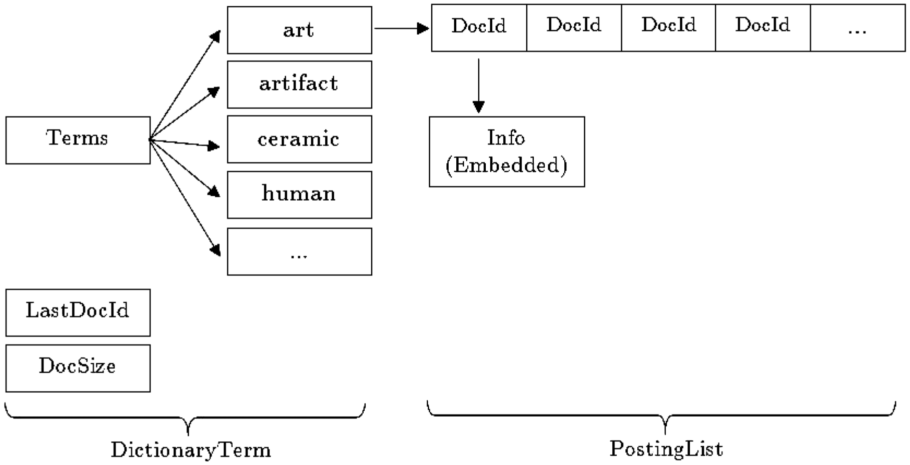

Figure 3. Design TVM inverted index

TVM inverted index maintain terms in dictionary using hash map in order to gain fast lookups, adds and deletes operations. Hash map as theoretically is performing O(1) and the worst case is O(n). Posting list for each term in dictionary also using hash map, it means that our dictionary term design is using double hash map. Access to an embedded information in this TVM inverted index will perform O(1) and the worst case is O(1*n) or O($n + m$) respectively.

TVM inverted index module is providing several ranking functions such as bag of words term-frequency — inverse document frequency (TF-IDF) [9], cosine similarity measure vector space model (VSM) [12], includes data provider for conceptual modeling latent semantic indexing (LSI) [9, 13] and topic modeling latent derelict allocation (LDA) [14], another ranking function can be added as needed as long as it is still matching with TVM inverted index data structure design.

7. Microservices. Index service in TVM are running on server side, therefore we have designed a microservices in order to communicate between applications, and serve requests from front-end to back-end server [6]. There are many benefits of microservice architecture as mentions in [15–17]. TVM microservice in this case only support HTTP POST request with data submission to prevent security issue as shown in algorithm below:

Algorithm. Client Query Request

```
$Auth \leftarrow$ List granted user accounts
while (request from client) do
  if Http.Method = POST then
    if Header.Authorization = $Auth$ then
      $DocId \leftarrow$ decode HTTP.Body payload to JSON format and get document Id
      if $IndexCache$ contain DocId then
        get list-docIds related to current DocId in $IndexCache$
        $info \leftarrow$ get info of list-docIds in Forward Index payload
      else
        calculate ranking using InvertedIndex modules
        get top-K (list-docIds) related to current $DocId$
        $IndexCache \leftarrow$ list-docIds
        $info \leftarrow$ get info of list-docIds in Forward Index payload
      end if
    else
      response : 401 Unauthorized
    end if
  else
    $info \leftarrow$ Request method must be POST
  end if
  response : encode $info$ to JSON format
end while
```

Algorithm pseudocode is describing how TVM microservice listening requests from clients, processing the request and give back a response in appropriate JSON format. HTTP header authentication schema is Basic, which transmit user and password credentials when the connection has established. The client request for searching similarity document is identified by DocId where each request will trigger inverted index ranking function and return list-docIds no more than maximum top-K ranked. The list-docId will be adding to the index-cache and using it again on the similar DocId request for reducing similar calculation process.

8. Experiment and results. TVM system which have provided a design for microservices in this experiment will be used for handling many requests from front-end to back-end server. We have used Intel Xeon Processor 2620v4, memory DDR4 32 Gb, and hard disk Skyhawk Surveillance 2 terabyte in conducting this experiment. The data we have been using as dataset are from “Collection Museum Registration System”, Directorate Cultural Heritage Preservation and Museum, Ministry of Education and Culture, The Republic of Indonesia, which contain 29 362 collections.

In this experiment we have prepared three methods which all outputs will be encoded, written and return to the client side in JSON format, first the service will return only list of docIds form to the client in consequently allowing them to process those list of docIds for another tasks, second the service will get the list of docIds and using them as keys to identify full detail collections information in TVM forward index payload, third the service will get the list of docIds and using them to perform a query to database mongoDB for searching the detail collections information. The query request method to indicate service performance is using document at a time (DAAT) as shown in Figure 4.

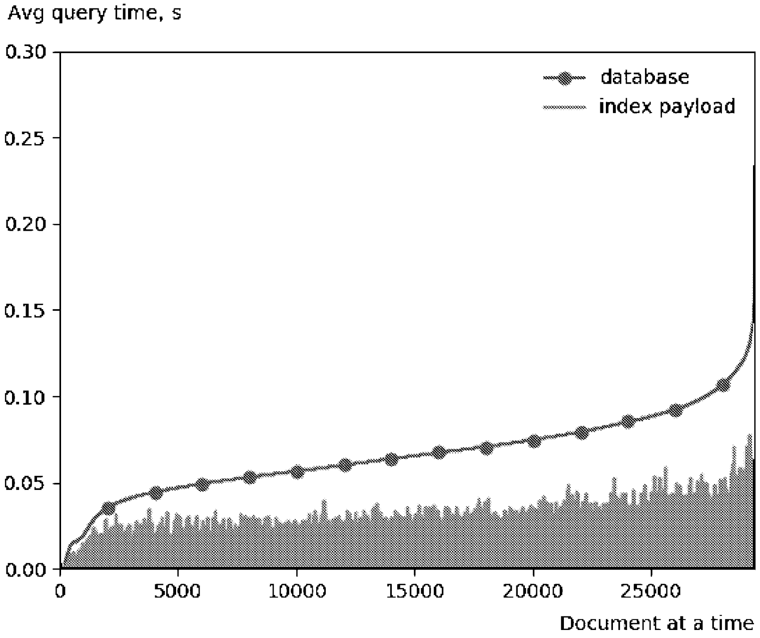

Figure 4. DAAT query requests from front-end to back-end server

DAAT query request from client or front-end using appropriate JSON structure in Figure 4 was handled by TVM independent microservices where after query has performed, then inverted index function directly calculates documents score and the function returned list of top-K ranking in list of docIds form. The query processing time for accessing information to database is 84.5% and access to index payload only need 0.02% from total query request time.

The second experiment, we are using query term at a time (TAAT), where each term in TVM inverted index dictionary have been using for query requests in order to evaluate and calculating documents score which have contained in postings list. The result as shown in Figure 5 is describing query time to database need 99.4% and query to TVM index payload only 0.0005% from total query request time.

The first method is only gives docIds list rather than give a full information, therefore there is no request needed to database or index payload, beside this docIds list can be used as caching for top-K docIds ranking in order to reduce computation time in back-end server. The cache key is docId of recent document which is using as query request, and the value is containing array set of documents identification. The second and third methods give us a significant result, where our method have provided an access to collaborate between TVM inverted index and forward index by given embedded information as payload. Our experiments shown that the methods have been proposed can reduced time to access a detail information of the collection rather than given many direct queries to database system.

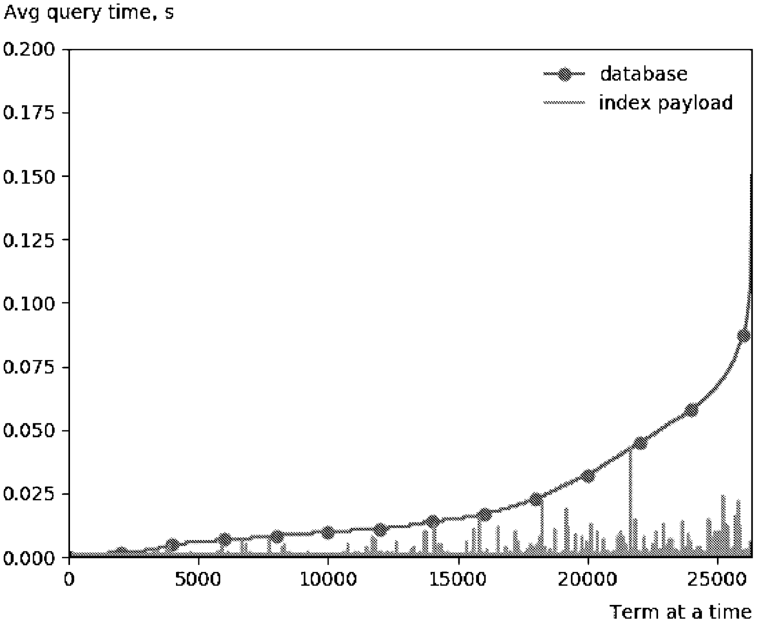

Figure 5. TAAT query requests from front-end to back-end server

9. Conclusion. In this work we have constructed data access service, modified forward and inverted index, and design special microservice architecture for TVM to provide relevant information for the virtual museums visitors. There are several experiments we have conducted, and shown our methodology give significant results when exchanged an information and collaborated between TVM forward and inverted index, query request, process and response through microservices give high performance output. Dynamic or flexible payload in TVM indices structure can be used for multipurpose indexing method in the development of modern information retrieval.

References

1. Anggai S., Blekanov I. S., Sergeev S. L. Management and information processing at virtual museum. *IEEE 4th Intern. conference on System Engineering and Technology* (*ICSET*). Bandung, Indonesia, 2014, pp. 1–5.

2. Hassaan M. A., Burtscher M., Pingali K. Breadth first search on cost-efficient Multi-GPU systems. *2010 19th Intern. conference on Parallel Architectures and Compilation Techniques* (*PACT*). Vienna, 2010, pp. 539–540.

3. Stanescu L., Brezovan M., Burdescu D. D. Automatic mapping of MySQL databases to NoSQL MongoDB. *2016 Federated Conference on Computer Science and Information Systems* (*FedCSIS*). Gdansk, Poland, 2016, pp. 837–840.

4. Jia T., Zhao X., Wang Z., Gong D., Ding G. Model transformation and data migration from Relational Database to MongoDB. *2016 IEEE Intern. Congress on Big Data* (*BigData Congress*). San Francisco, CA, USA, 2016, pp. 60–67.

5. Wu Q., Chen C., Jiang Y. Multi-source heterogeneous Hakka culture heritage data management based on MongoDB. *2016 Fifth Intern. conference on Agro Geoinformatics.* Tianjin, 2016, pp. 1–6.

6. Anggai S., Blekanov I. S., Sergeev S. L. Design Muntoi web-based framework and search engine analytics for thematic virtual museums. *4th Intern. conference on Interactive Digital Media* (*ICIDM*). Bandung, Indonesia, 2015, pp. 1–5.

7. Brin S., Page L. The anatomy of a large-scale hypertextual Web search engine. *Proceedings of the Seventh Intern. conference on World Wide Web 7* (*WWW7*). Amsterdam, The Netherlands, 1998, pp. 107–117.

8. Zobel J., Moffat A. Inverted files for text search engines. *Journal ACM Computing Surveys* (*CSUR*), 2006, vol. 38, no. 2, article 6, pp. 1–56.

9. Manning C. D., Raghavan P., Schutze H. *Introduction to Information Retrieval.* New York, USA, Cambridge University Press, 2008, pp. 61–123.

10. Panev K., Berberich K. Phrase queries with inverted + direct indexes. *Intern. conference on Web Information Systems Engineering.* Thessaloniki, Greece, 2014, pp. 156–169.

11. Anggai S., Blekanov I. S., Sergeev S. L. Construction inverted index for dynamic collections visualization in thematic virtual museums system. *2017 3rd Intern. conference on Science and Technology — Computer* (*ICST*). Jogjakarta, 2017, pp. 186–189.

12. Salton G., Buckley C. *Term Weighting approaches in automatic text retrieval.* Cornell, Cornell University Press, Department of Computer Science, 1987, pp. 323–327.

13. Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., Harshman R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 1990, vol. 41, pp. 391–407.

14. Blei D. M., Ng A. Y., Jordan M. I. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 2003, vol. 3, no. 2, pp. 993–1022.

15. Singleton A. The economics of microservices. *IEEE Cloud Computing*, 2016, vol. 3, no. 5, pp. 16–20.

16. Guo D., Wang W., Zeng G., Wei Z. Microservices architecture based Cloudware Deployment Platform for service computing. *2016 IEEE Symposium on Service-Oriented System Engineering* (*SOSE*). Oxford, UK, 2016, pp. 358–363.

17. Bakshi K. Microservices-based software architecture and approaches. *2017 IEEE Aerospace Conference.* Big Sky, MT, USA, 2017, pp. 1–8.

For citation: Anggai S., Blekanov I. S., Sergeev S. L. Index data structure, functionality and microservices in thematic virtual museums. *Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes*, 2018, vol. 14, iss. 1, pp. 31–39. https://doi.org/10.21638/11701/spbu10.2018.104

Статья рекомендована к печати проф. Л. А. Петросяном.

Статья поступила в редакцию 10 ноября 2017 г.

Статья принята к печати 11 января 2018 г.