Journal of ENVIRONMENTAL Treatment Techniques

J. Environ. Treat. Tech.
ISSN: 2309-1185

Journal web link: http://www.jett.dormaj.com

The Development of a New Data Migration Model for NoSQL Databases with Different Schemas in Environment Management System

Lim Fung Ji[1]*, Nurulhuda Firdaus Mohd Azmi[2]

[1]Department of Computer Scence and Embedded Systems. Faculty of Computing and Information Technology, Tunku Abdul Rahman University College
[2]Advanced Informatics Department. Razak Faculty of Technology and Informatics, University Technology Malaysia

Received: 12/06/2019 Accepted: 22/01/2020 Published: 20/02/2020

Abstract

Data migration transfers data from one database to another database. The motivations of data migration are, for example, transferring data from a legacy database to a modern one and maintaining data up-to-date and consistent in a distributed system. Compared to data migration between traditional databases, data migration between heterogeneous NoSQL databases is more challenging due to the characteristics of NoSQL database such as flexible schema, different supporting features, and different storage paradigms. The differences may cause data quality problem after data migration, especially for environment management system where data are required to predict or to convey accurate information. Therefore, the migration of data between heterogeneous NoSQL databases requires not only to overcome the differences of these databases, but also to ensure the quality of the migrated data. In this paper, we proposed a data migration hub, a model that uses a record to record migration style to transfer data between different NoSQL database schemas. The proposed hub is applicable to the environment management system with data validation and fault tolerance in migration process. As confirmed by the pilot study, our method is able to migrate full set of fields to the destinated database in MongoDB.

Keywords: Data migration; NoSQL database; Heterogeneous schema; Document-based NoSQL, MongoDB, Environment management system

1 Introduction

Data migration process transfers data from one database to another database due to several reasons such as transferring data from a legacy system to a new developed system, moving data between distributed data nodes, etc. This is study focuses on data migration between NoSQL databases. In this migration process, different storage paradigms of NoSQL databases need to be taken into consideration in order to avoid any discrepancy of data quality from the original data source. NoSQL database is a type of database that is becoming increasingly popular. It is applied to different areas like environment management systems (EMSs) that using computing technology. Environment management system refers a system that assists in the management of environmental impacts of an organisation and enhances the environmental performance of their services and products [17]. One of the examples of EMS application is the management program of water quality in Bangladesh [18]. Therefore, EMS should provide correct, precise, and up-to-date data for effective management of environmental issues.

2 Data Migration between NoSQL Databases

2.1 Data migration

Data migration is defined as "a tool-supported one-time process that aims at migrating formatted data from a source structure to a target data structure, where the two structures differ on a conceptual and/or physical level" [1]. In addition, data migration not only means to transfer data to a destination database, but also requires to adapt the migrating data to the data schema, model, and types of data in the destinated database [2]. Migration of data involves different types of data stores. These data stores can either be of the same or different types. In case of data migration between NoSQL databases, the characteristics of these databases are required to be considered well.

Corresponding author: Lim Fung Ji, Department of Computer Scence and Embedded Systems. Faculty of Computing and Information Technology, Tunku Abdul Rahman University College. E-mail: limfj@tarc.edu.my

2.2 Characteristics of NoSQL database

NoSQL database is available in four generic types: document-based, column-based, key-value, and graph [3]. NoSQL database has an advantage over relational database due to its "flexi-schema". The "flexi-schema" behaviour allows different structures of records to be stored within the same table [4]. For example, in a document-based NoSQL such as MongoDB, documents (record) within the same collection (table) are allowed to contain different numbers of fields. The "shared nothing architecture" of NoSQL applies local storage pool that allows faster data access by adding number of data nodes. NoSQL database has such a high elasticity that replicates data to newly-added data nodes [5]. Eventually consistency, data can be read from replicas of other data machine if a machine is down [4].

2.3. Challenges of data migration in NoSQL databases

Three challenges commonly arise in data migration [6]: (a) interruption of business operation; (b) loss of data and degradation of data consistency and (c) effort and cost required for data migration. In NoSQL database, data migration process faces some challenges related to the common challenges sated above, which are caused by the characteristics of the NoSQL database.

a) Heterogeneous storage paradigm: Each type of NoSQL database implements different ways to store data. Different storage paradigms have specific rules and format in storing the data. Table 1 summarizes the storage paradigms of NoSQL databases. The table indicates that re-formatting or restructuring of data is required in order to map the data to the targeted database storage structure. Therefore, qualities of data such as completeness, consistency, and correctness are concerned when data are being restructured or reformatted. The degradation of data quality will lead to higher cost of recovery and data quality enhancement process.

Table 1: NoSQL database storage paradigms.

No SQL database	Storage Paradigms
Document-based	Allowing embedded of key value in document; allowing search based on both key and value [7]
Columnar database	Storing data in distributed, multiple dimensional map; having mixed row/column storage [8]
Key-value database	Storing data in byte-array; assessing data through key-value hash table (each key points to a specific datum) [8]
Graph database	Storing data in nodes; connecting data by edge (edge represents the relationships between nodes); using pointer to point to another nodes [7]

b) Flexibility in schema structure: The "flexi-schema" of NoSQL allows more flexibility in storing data [4]. For example, in MongoDB, documents in the same collection may have different numbers of fields. Figures 1 and 2 [9] show the examples of different schema in documents.

```
{
    name: "Midhuna",
    age: 23,
    place: "New York",
    hobbies: ["Singing", "Reading Books"]
}
```

Figure 1: MongoDB document [9]

(c) Supporting features: Even within the same type of NoSQL database, other features such as support on query language and CAP features support are different [11]. The challenges mentioned above are based on the features of NoSQL databases. In the perspective of data quality, data migration may face the challenges in maintaining the quality of data migrated. Table 2 summarizes the challenges of data migration from the perspective of data quality.

Table 2: Data quality challenges of data migration [10].

Challenges	Details
Missing Data	The old and new database may have different fields. Some fields in the new database may not exist in the old one. The NULL value is used to represent the non-existence of data, which is critical for migration of data between different storage formats.
Data Accuracy	When data is migrated through manual approach, especially keyed by human, accuracy of data is not guaranteed.
Legacy System	Some data may lose as the result of system upgrades.
Data Element	Existing data has the problem where same word is used for different definitions. Therefore, it needs further clarification on the value of data transferred. This problem affects data consistency.

According to the challenges discussed in this section, it is clear that data migration requires not only to ensure the data can be migrated, but also concern the quality of migrated data. For NoSQL databases, the challenges discussed in this section have significant effects on data quality. For example, the flexibility of schema in NoSQL may cause missing data in some fields of target database.

3 Related Work

The authors in [2] attempted to overcome the challenge of different data formats between NoSQL databases. To this end, they proposed an approach of migrating data between different types of NoSQL databases by converting the existing data into an intermediate format to be converted later again to the format required by the destinated database. The approach migrated data between column-based NoSQL

database with interface portability issues. The authors further enhanced the approach with fault tolerance features to identify data issues such as missing data and duplication of data [12].

Another framework was proposed in [13], which applied both direct and intermediate conversion of data between NoSQL databases. It covered each type of conversion target on different databases, that is, direct conversion for column-based database to graph database and intermediate conversion for document based databased to column-based database, and vice versa. In [14], to migrate data between NoSQL databases, an approach of migrating data when the data is required or "on-demand" was introduced. This approach is called lazy migration. The lazy migration method has the benefits of reducing the unavailability time duration of data system caused by the process of data migration. In another study [15], a framework was proposed with data migration validation and control features for three types of NoSQL databases, i.e., graph, key-value, and document databases. The approaches discussed above provide different ways of migrating data, allowing transfer of data between different interfaces of databases concerning also the data quality.

```
{
    name: "Midhuna",
    age: 23,
    place: "New York",
    hobbies: ["Singing", "Reading Books"]
    spouse: {
        name: "Akash",
        age: 25
    }
}
```

Figure 2 :MongoDB document with additional fields [9].

Figure 1 and Figure 2 show two sample documents with different structures. In comparison, document in Figure 2 has additional fields named *spouse* which consists of *name* and *age* fields. This difference of field structure raises some issues on data migration:

(a) Inconsistency of field numbers in documents: The inconsistency of fields in documents do not provide the total distinct fields in a collection. Thus, it is challenging when the target database requires a full field list in the documents. In addition, data migration process may not require to migrate all data from the documents in the source database, but it may be required to migrate only selected field(s). This will lead to a problematic condition when some documents in the original database do not have the required field(s).

{b) Handling of value for non-existing field(s): If the targeted documents(s) consists of field that is not the same as the document of original database, in some cases, assigning *null* value to acknowledge the absence of the field is critical for data migration [10]. However, in some cases, *null* value may need extra conversion for numeric field in the target database rather than only showing the non-existence of field in the old database.

4 Proposed Model

In this paper, we proposed a migration hub for heterogeneous schema structure for NoSQL databases. The migration hub is a model that connect different types of NoSQL databases for the purpose of migrating data between these databases. In addition, migration of data is also allowable between different schema structure within the same type of database. The proposed migration hub provides validation of data quality features for consistency, completeness, correctness, and data currency to evaluate the quality of data after migration. Furthermore, the migration hub consists of error detection feature to restore the data in the case error happened during migration process. Figure 3 shows the abstract architecture of the migration hub and Figure 4 depicts the inner components of migration hub.

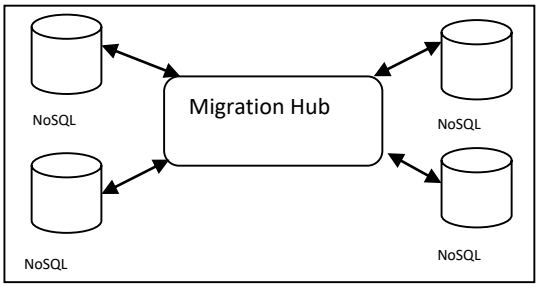

Figure 3: Abstract architecture of migration hub

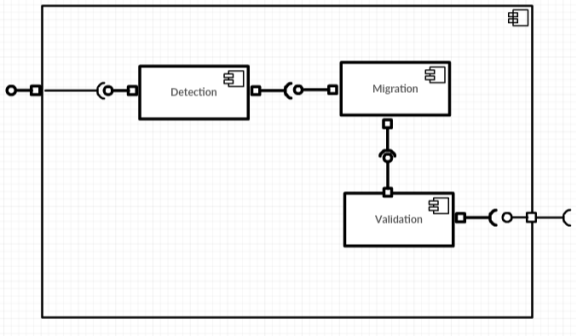

Figure 4: Migration hub's components

The migration hub connects different NoSQL databases for migration. It consists of three main components, i.e., detection, migration, and validation components. The functions of these components are summarized in Table 3. The proposed migration hub was built for migrating data between heterogeneous databases in addition to ensuring the completeness, consistency, correctness, and currency of data. Furthermore, it provides fault tolerance for migration process to recover data.

5. Pilot Study

In the process of developing the migration hub, a pilot study was conducted as a small scale of experiment with limited but important functions to be applied. Table 4 shows the experiment environment for the pilot study.

Table 3: Functions of migration hub components

Component	Functions
Detection	Detecting and analyzing schema of data to be migrated. Generating metadata of the schema and mapping the structure of data to the target database.
Migration	Migrating data to the target database base on metadata created. Detecting and recovering errors for migration process.
Validation	Checking the migration process concerning the aspects of data quality, i.e., completeness, correctness, consistency, and currency.

Table 4: Pilot study setup

Operating System	Windows 10
Database	MongoDB
Database Tool	MongoDB Compass
Programming Language	Java
Integrated Development Environment	Java NetBean, Java SDK

The pilot study was aimed at: (a) migration of data between different data schema of same database type and (b) checking the completeness of data migration process. The pilot study started with the migration of data within the same type of database where the source of data had different schema. The database used was MongoDB that is a document-based database. The data source selected was the curated list shared from GitHub [16]. The dataset contained fourteen different collections, as shown in Table 5.

Table 5: Collection in sample dataset

Collection Name	Description
Books.json	Information regarding different types of books
City_inspection.json	Inspection information of some shops in different systems
Companies.json	Information regarding different companies
Countries-big.json	Abstract information on countries
Countries-small.json	Detailed information on countries, but less records
Covers.json	Book covers information
Grades.json	Student scores in study
Palbum.json	Image lists
People-bson.zip	People information in zip format
Products.json	Information of different products
Profiles.json	Status of client
Restaurant.json	Restaurants addresses information
Students.json	Students examination scores
Tweets.zip	Tweeter data in zip format

We started data migrating from the *products* collection. The *products* collection consists of information related to a product but the documents (records) hold different structures, i.e., different fields. Figure 5 shows part of the discrepancies between documents. As depicted by Figure 5, some documents do not consist of the fields *brand* and *price*. Therefore, when migrating the data within the documents, some issues are needed to be taken into consideration: (a) allowing field selection for migrations; (b) handling variances of fields and (c) ensuring data in selected field(s) are migrated. The above issues are taken into account in the design of the algorithm shown in Figure 6.

	_id Mixed	name String	brand String	type Mixed	price Mixed	rating Mixed
1	"ac3"	"AC3 Phone"	"ACME"	"phone"	200	3.8
2	"ac7"	"AC7 Phone"	"ACME"	"phone"	320	4
3	507d95d5719dbef170f15bf9	"AC3 Series Charger"	No field	[] 2 elements	19	2.8
4	507d95d5719dbef170f15bfa	"AC3 Case Green"	No field	[] 2 elements	12	1
5	507d95d5719dbef170f15bfb	"Phone Extended Warranty"	No field	"warranty"	38	5
6	507d95d5719dbef170f15bfc	"AC3 Case Black"	No field	[] 2 elements	12.5	2
7	507d95d5719dbef170f15bfd	"AC3 Case Red"	No field	[] 2 elements	12	4
8	507d95d5719dbef170f15bfe	"Phone Service Basic Plan"	No field	"service"	No field	3
9	507d95d5719dbef170f15bff	"Phone Service Core Plan"	No field	"service"	No field	3

Figure 5: Different fields in a document

1. Identify destination database.
2. Analyse collection available.
3. Identify collection to migrate.
4. Identify destination collection.
5. Analyse document structure in original collection.
6. Identify field to be migrated.
7. Analyse data type of each field in documents to be migrated.
8. Generate meta data of field to be migrated.
9. Execute migration functions.
10. Check migrated data.

Figure 6: The migration algorithm

The algorithm was implemented using Java language written in NetBean IDE environment. The data in the *products* collection were transferred to a blank collection named *mproducts* in the Target database. For the first run of the pilot study, all fields were migrated to the *mproducts* collection. That is, documents of the *mproducts* would be consisting of all fields of the *products* collection. Therefore, the detection of distinct fields in *products* collection was required. Figure 7 shows the distinct fields detected in *products* collection and indicated for field(s) selection. For the field selection, all fields need to be selected. The result of migration is shown in Figure 8. As can be seen in this figure, the "non-existing" fields in the *mproducts* collection are shown with the value "blank".

```
Selected collection is products
_id
name
brand
type
price
rating
warranty_years
available
for
color
monthly_price
limits
term_years
sales_tax
cancel_penalty
additional_tarriffs
Available field(s) in existing collections
```

Figure 7: Distinct fields detected and indicated for selection

After data migrated was accomplished, it was necessary to check the completeness of data migrated. In the pilot study, we compared the number of documents (records) migrated to the original source of records. The result of comparison is shown in Figure 9. The result shows that the documents (records) were successfully migrated to the target collection. The number of record migrated was the same as the number of record to be migrated in the origin database. This is a simple validation method of counting the number of records to be migrated.

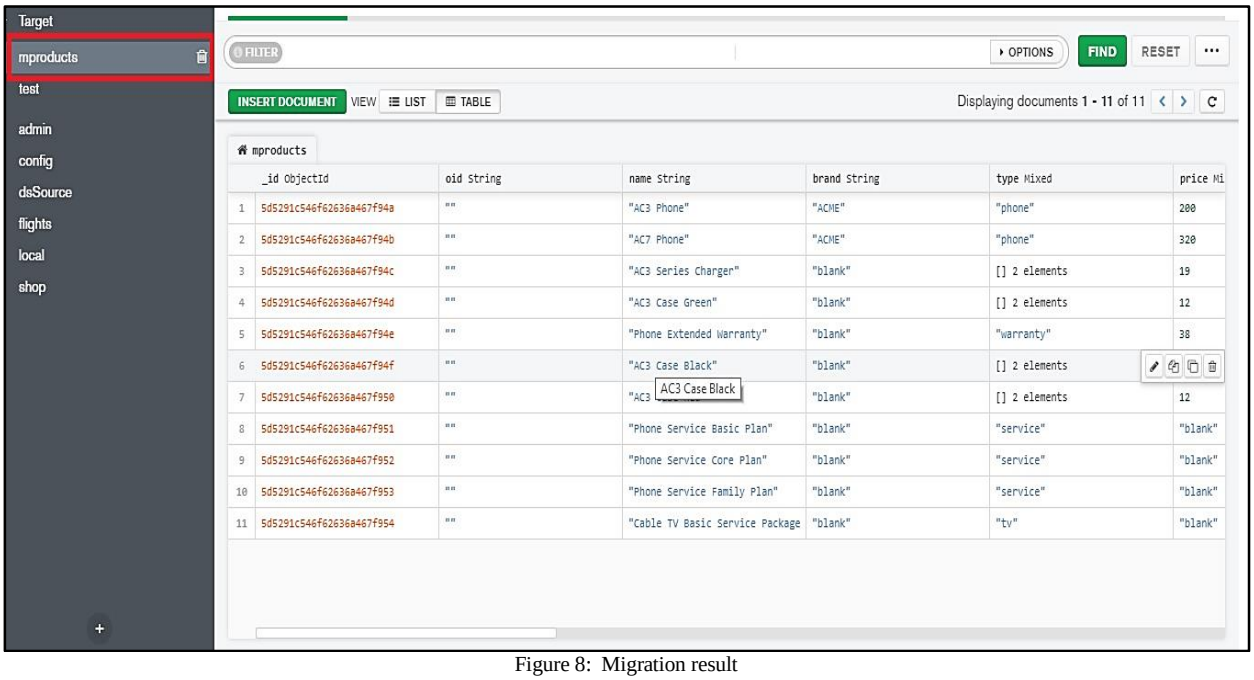

Figure 8: Migration result

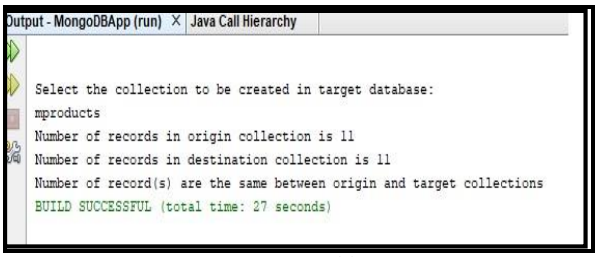

Figure 9: Comparison of field number

6 Further Study

The pilot study reported here lays a foundation model for the development of migration hub. It provides the migration methods between document-based database where the schema is different. However, further study will be performed to enhance the foundation model. The features or functions for the next enhancement are: (a) applying real environmental data; (b) migration between different schema structure; (c) migration between different types of database; and (d) checking data quality concerning consistency, correctness, and completeness.

7 Conclusion

The migration of data between heterogeneous schema of NoSQL databases is not limited to migration across different types of database. Additionally, there is a need to consider different schemas between the same type of NoSQL. The quality of data migrated need to be preserved as to ensure that the target database will not encounter any data violation when it is accessible by the application connected to it.

Aknowledgment

We would like to express our gratitude and sincere to Ministry of Higher Education and Universiti Teknologi Malaysia for funding given through project grant R.K130000.7856.5F229.

Ethical issue

Authors are aware of, and comply with, best practice in publication ethics specifically with regard to authorship (avoidance of guest authorship), dual submission, manipulation of figures, competing interests and compliance with policies on research ethics. Authors adhere to publication requirements that submitted work is original and has not been published elsewhere in any language.

Competing interests

The authors declare that there is no conflict of interest that would prejudice the impartiality of this scientific work.

Authors' contribution

All authors of this study have a complete contribution for data collection, data analyses and manuscript writing.

References

1 F. Matthes and C. Schulz, "Towards an integrated data migration process model-State of the art & literature overview," Technische Universität München, Garching bei München, Germany, Tech. Rep, 2011.

2 M. Scavuzzo, E. D. Nitto, and S. Ceri, "Interoperable Data Migration between NoSQL Columnar Databases," in 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, 2014, pp. 154-162.

3 R. Zafar, E. Yafi, M. F. Zuhairi, and H. Dao, "Big Data: The NoSQL and RDBMS review," in 2016 International Conference on Information and Communication Technology (ICICTM), 2016, pp. 120-126.

4 K. Tsuyuzaki and M. Onizuka. (2012, 13 Jun 2019). NoSQL Database Characteristics and Benchmark System. 10, 5. Available: https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201212fa3.pdf&mode=show_pdf.

5 D. Ganesh Chandra, "BASE analysis of NoSQL database," Future Generation Computer Systems, vol. 52, pp. 13-21, 11// 2015.

6 A. Martens, M. Book, and V. Gruhn, "A data decomposition method for stepwise migration of complex legacy data," presented at the Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice, Gothenburg, Sweden, 2018.

7 P. Swaroop, V. G. , K. R. S. , and S. N. R. , "NoSQL Paradigm and Performance Evaluation," SSARSC International Journal of Geo Science and Geo Informatics, vol. 3, 2016.

8 A. Makris, K. Tserpes, V. Andronikou, and D. Anagnostopoulos, A Classification of NoSQL Data Stores Based on Key Design Characteristics vol. 97, 2016.

9 TutorialKart. (2018, 15/09/2019). MongoDB Document – Structure and Sample Documents. Available: https://www.tutorialkart.com/mongodb/mongodb-document/

10 Z. Ikhlas Fuad, H. A. A. Fatani, and N. A. H. Zammarah, "Data migration challenges: The impact of data quality — Case study of University Putra Malaysia UPM," in 2011 International Conference on Research and Innovation in Information Systems, 2011, pp. 1-5.

11 H. M. L. Dharmasiri and M. D. J. S. Goonetillake, "A federated approach on heterogeneous NoSQL data stores," in 2013 International Conference on Advances in ICT for Emerging Regions (ICTer), 2013, pp. 234-239.

12 M. Scavuzzo, D. A. Tamburri, and E. d. Nitto, "Providing Big Data Applications with Fault-Tolerant Data Migration across Heterogeneous NoSQL Databases," in 2016 IEEE/ACM 2nd International Workshop on Big Data Software Engineering (BIGDSE), 2016, pp. 26-32.

13 A. Bansel, H. González-Vélez, and A. E. Chis, "Cloud-Based NoSQL Data Migration," in 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), 2016, pp. 224-231.

14 M. Klettke, U. Störl, M. Shenavai, and S. Scherzinger, "NoSQL schema evolution and big data migration at scale," in 2016 IEEE International Conference on Big Data (Big Data), 2016, pp. 2764-2774.

15 Y. S. Wijaya and A. AkhmadArman, "A Framework for Data Migration Between Different Datastore of NoSQL Database," in 2018 International Conference on ICT for Smart Society (ICISS), 2018, pp. 1-6.

16 H. Ozler. (2019, 13/7/2019). A curated list of JSON/BSON datasets from the web in order to practice/use in MOngoDB. Available: http://www.github.com/ozlerhakan/mongodb-json-files.

17 Sam MFM, Shuqi AL. The Effects of Environmental Management System towards Company Financial Performance in Southern Region of Peninsular Malaysia. Journal of Environmental Treatment Techniques. 2019;7(4):794-801.

18 Hossain ML, Islam KS. Assessment of Water Quality in Chandpur District of Bangladesh Journal of Environmental Treatment Techniques. 2013;1(2):91-100.

.

Author Profile:

Mr. Lim Fung Ji is a lecturer currently working in Tunku Abdul Rahman University College, Kuala Lumpur, Malaysia. He is now studying PhD in University Teknologi Malaysia, Kuala Lumpur.

Nurulhuda Firdaus Mohd Azmi is Senior Lecturer at Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia. She received BSc in Computer Science at UTM and MSc in Applied Statistics at UPM. Her PhD degree is in Computer Science at University of York, UK with the research focus on Artificial Immune Systems for Adaptive Information Filtering. She currently active in research and development projects in the area of Data Analytics in Operational Research (OR) and Quantitative Management.