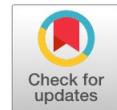


Database Migration using Data Synchronization and Transactional Replication



Winda Sekar Dewi, Ema Utami, Bambang Sudaryatno

Abstract— Database migration is one of the important thing to make the data can be stored optimally and minimize risk of data lost. Data Synchronization and Transactional Replication can be used to do it. Transactional Replication is working on services area to migrate database stored in cloud to another cloud storage. Data Synchronization is used to synchronize migrated database with origin database. Data lost concerns remain as the big consideration to widespread the adoption. Many ways can be done to minimize the risks, one of them is by migrate the database that is already stored in cloud to another database in cloud. This research will conduct database migration using data synchronization and transaction replication. By using Azure and Amazon Web Services as Database Migration Services, these techniques allow SQL Database data to be easily replicated, and allows multiple SQL Databases to isolate mission-critical workloads from analytic queries that run relatively longer. The evaluation is by comparing these two migration services based on the synchronizing time of full loaded database migration. Migration result shows that database migration using transactional replication and data synchronization is influenced by the number of rows as well as size of the table, server location, also uploading and downloading speed in each database migration service.

Keywords— Database Migration, Transactional Replication, Data Synchronization, Azure, AWS.

I. INTRODUCTION

Local data centers or physical data centers are commonly used by both small and big companies to store their data. This physical data center storage room turns out to have many risks, one of them is natural disaster. Natural disasters are unavoidable events when it comes to physical threats. Especially in countries where natural disasters often occur, things like earthquakes, storms and other natural disasters that can damage building infrastructure can threaten the existence of a physical data center. Besides that, the use of physical data centers also requires relatively large costs. In addition to procurement for the physical data center itself, data center storage space must have its own standardization.

Manuscript published on 30 August 2019.

*Correspondence Author(s)

Winda Sekar Dewi, Department of Magister of Informatics Engineering, AMIKOM University

Jl. Ring Road Utara-Condong Catur Sleman, Yogyakarta, 55283, Indonesia

Ema Utami, Department of Magister of Informatics Engineering, AMIKOM University

Jl. Ring Road Utara-Condong Catur Sleman, Yogyakarta, 55283, Indonesia

Bambang Sudaryatno, Department of Magister of Informatics Engineering, AMIKOM University

Jl. Ring Road Utara-Condong Catur Sleman, Yogyakarta, 55283, Indonesia

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Some things that must be considered in developing physical IT infrastructure are power supply, distribution and cooling energy generation, temperature and humidity regulator, data center security, capacity management, and efficiency and energy consumption [1]. In the development of data centers, technology emerged, known as Cloud Computing. Cloud computing offers data storage services known as cloud storage [2]. Cloud storage is a service model where data is managed, maintained, and backed up remotely, which is then available to users inside the network [3]. Cloud storage is an internet-based online service that provides large amounts of storage space and can be adapted to computing power [2]. With this platform, however, at the same time, it shifts the responsibility of the local machine in maintaining and storing data. Cloud Computing allows small and medium enterprises to large companies that require even large load capacity, to fully transfer their data center infrastructure to cloud storage, so they no longer need to build data centers internally [4]. Some cloud service models have started to be studied and illustrate how cloud services are available to clients. The most well-known and most basic service models are a combination of IaaS (infrastructure as a service), PaaS (platform as a service), and SaaS (software as a service). Infrastructure as a service allows small and large companies to move their physical infrastructure to the cloud. By moving their physical infrastructure into a shared cloud, the level of control is similar to what they have in a physical datacenter. IaaS provides close proximity to in-house data centers compared to other types of services. The components of the central data center infrastructure are servers (computing units), storage, networks, and management tools for monitoring and maintaining infrastructure. Infrastructure components such as application servers, cache web server servers, and databases must also be scaled quickly if needed. If a company uses centralized components that are difficult to measure, these components may have to be stored to be compatible with the cloud [5]. As the big companies tried to migrate their data into cloud, supposed that small and medium enterprises (SME) are also preparing to go there. In the process of developing the business, the SMEs also try to grow their services as well as their data using public or private cloud for development [6]. When the business process changes due to several actions like acquisition and merger, data should be moved into several locations of storage [7]. As the business grows, same goes also with the cloud storage. But unfortunately, as the data grows bigger, in the cloud storage also has some risks such as slow internet connection during data migration can cause downtime, unexpected issue in data downloading can affect data security [7].

Database Migration using Data Synchronization and Transactional Replication

One of the case happened in one of the hosting company in Indonesia, as they released their apologize to their customer, around thousands of website in their cloud vanished without any cause in less than 5 minutes. Other problems that might happen during data migration process are data crash or unauthorized access by the third parties [8].

Regarding with adjusting these components, one of the things that is usually be adjusted or moved is the database. The database migration process is usually being done by moving the database from local to cloud, cloud to cloud, or moving to other databases. Database migration is the process of transferring data between two database management systems [9]. The process of data migration is one of the important things in improving and developing the database structure. The size of the database, differences in semantic tables, and syntax differences are examples of some of the main challenges in the field of data migration [10]. In the process of data migration, the thing to do is extract data from the old database and reload it into the new database. The process of exporting, importing and migrating data from various sources is quite time-consuming and complicated, especially if this data source stores data in a different format [11]. Data migration includes two important processes. The first process carried out was the identification, elicitation and collection of entities. In the second process, migration of data from the source entity to the destination entity is carried out. Usually, the data migration system is already in focus on the first process. So the differences and benefits of each of these systems are very dependent on the implementation of the first process. With this view, it can be concluded that data migration systems can be classified into two general categories: traditional data migration systems and semantic data migration systems [10]. There are a lot of research about database migration with different kind of models. Ahmadi et al [10] proposed a new model of data migration which focuses on migrating data between two databases, namely semantic and traditional. In this model, the data migration process is done by creating a middle layer between the two databases. Covering both traditional databases and semantics is one of the features featured in the proposed model. In this method, the semantic database is assumed to be a traditional data and database source as the purpose of data migration. The middle layer that is placed between the two databases consists of two phases, namely creating an entity and then filling it out. Another research discusses about incremental data migration in multi-databases in ETL algorithm approach. Yadav et al [11] discuss in their paper about the challenges of data migration in a dynamic IT environment and what are the main business advantages that have been designed to adapt the traditional tools used for data migration. This paper designs an architecture that is adapted for data migration and allows validated users to select the source and destination database from several options provided. The database migration process is done by migrating the table and then migrating the data in it. Before starting the migration process, a connection must be established in advance and a detailed summary between the source and destination of the selected item. Then the user is asked to choose the desired limit. After that, finally the actual migration process begins. To achieve an effective data migration process, data on the old system is mapped to the new system. This new system has provided designs for data extraction and data loading. This design will later link the old data format with the new system format. This data migration

may involve many processes, including data extraction processes, where the data is read from the old system, and data loading process, where data is written to the new system. The algorithm strategy used in this method is the ETL algorithm where data from the relational database is read and then applied several transformations or business rules before finally being loaded into another database. There is also research that reviews database migration strategies, techniques, and tools. Elamparithi and Anuratha [12] in their paper review and suggest some basic criteria for evaluating data migration tools. To implement good database technology, the software industry, especially those related to cloud computing, faces the challenges of data migration and applications. Therefore, some database migration strategies are needed which are then adjusted to the data migration needs that can provide information about conversions and database applications. This paper is divided into several sections as follows: the first section summarizes the impact of primitive strategies such as reverse engineering and inheritance migration. The next section discusses various possible migration techniques and strategies related to Relational Database Migration (RDBM). In addition, they also discussed various types of Database Migration Toolkit (DBMT) and evaluation criteria. The last section discusses conclusions and future work related to data migration. Another research specifically discusses about data migration in public cloud. Singh and Laxmi [13] examine important issues in the design and development of architectures and scalable techniques for data migration and application migration efficiently and effectively. They explained that data migration and application migration allows management of data storage and computing to be self-managing and autonomous. The unique feature of this automated migration approach is its ability to dynamically adjust schedule data migration that is useful for achieving optimal migration effectiveness. This feature also considers the specific character of an application and I / O profile, as well as paying attention to workload deadlines. Experiments running on the traces of real systems show that basic data migration models appear to be effective in increasing system resource utilization and this adaptive migration model is known to be more efficient in continuously improving and adjusting performance in the scalability of multi-tiered storage systems. From all the statements mentioned above, supposed that neither local nor cloud storage can be defined as the safest place to store data. Local storage is also still needed to do data tracking independently by the name of speed of services. For that, the companies should be calculating and all the risks that might happen and decide which data storage they should choose to store their data. But instead of choosing one of them, companies can these two kinds of data storage by synchronizing them in particular time. This research will conduct about database migration using data synchronization and transaction replication. This technique allows SQL Database data to be easily replicated, and allows multiple SQL Databases to isolate mission-critical workloads from analytic queries that run relatively longer. By using Azure as Infrastructure as a Service, transactional replication will work to migrate database from cloud to cloud, data synchronization will take a part in testing database with queries.

The purpose of this research is to combine these two techniques with expectation of modernizing application in a multilevel approach in order to prepare the development of small and medium enterprises.

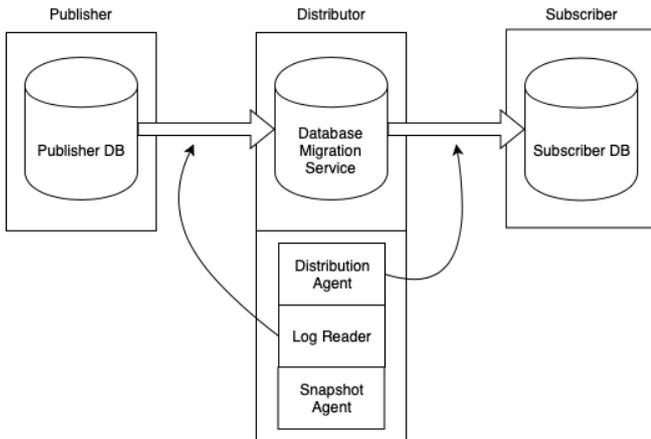


Fig. 1. Transactional Replication Architecture

II. METHOD

The method in this proposed research is by combining transactional replication and data synchronization which suitable on the data migration from cloud to cloud.

A. Transactional Replication

According to Mazilu, replication is a process of sharing information to ensure consistency between redundant resources, such as software or hardware components to increase reliability, fault tolerance and accessibility [14]. In transactional replication, the replication agent or in this case, database migration service will monitor source server and transmit data to target server if there are changes [15]. The architecture of transactional replication is shown in figure 1.

B. Data Synchronization

According to Jindal, data synchronization is the process of building consistency between data from the source to the target data storage and vice versa. Harmonization of data occurs continuously from time to time. Data synchronization technology is designed to synchronize a data set with two or more devices, which then automatically copy changes in two directions. Synchronization is used to control access to good conditions in small-scale multiprocessing systems, in multithread environments, multiprocessor computers and in distributed computers consisting of thousands of units, in banking, database systems, on web servers, and so on [16].

Whereas according to Shabani, replication strategies are interesting in the growth of various settings where guarantees of weak consistency can be received in return for higher availability and the ability to update data when the connection is lost. These uncoordinated updates will have to be synchronized (or reconciled) automatically combining non-conflict updates, while detecting and reporting conflict updates [17].

C. Data Synchronization in Transactional Replication

As said, in order to run the data synchronization process in transactional replication, the first step is to set up the Database Migration Service (DMS) as a distributor, in the same time also set up source database as a publisher and target database as a subscriber. The next step is to assess

database schema on distributor, which is then converted from publisher schema to subscriber schema. After all schema are match, distributor can start to migrate schema and data from publisher to subscriber using transactional replication. After all schema and data are full loaded to subscriber, data synchronization runs every time there is a change in publisher. Data synchronization will stop when the user manually cut over the migration process. This process is explained on figure 2.

III. RESULTS AND DISCUSSION

The main element in this research is database. Database used is from the debt recording application. The source version is Amazon RDS for MySQL 5.7 and the target version is Azure Database for MySQL 5.7. The database has 5 master tables, and 4 transaction tables. Figure 3 shows the relation schema of tables. Tables structure is shown on Table 1.

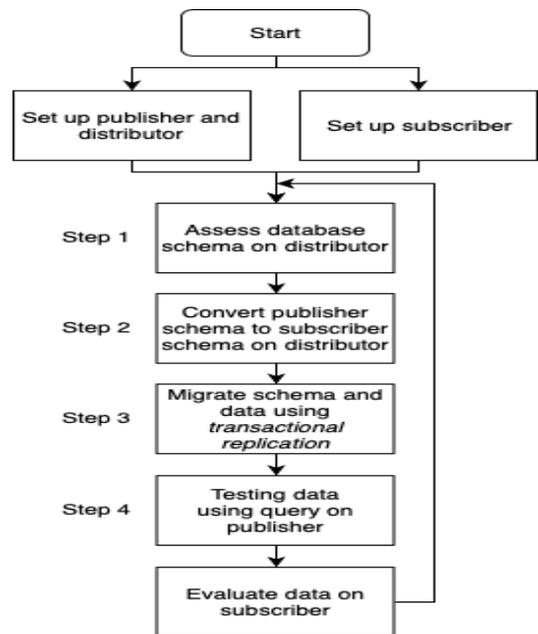


Fig. 2. Migration Process

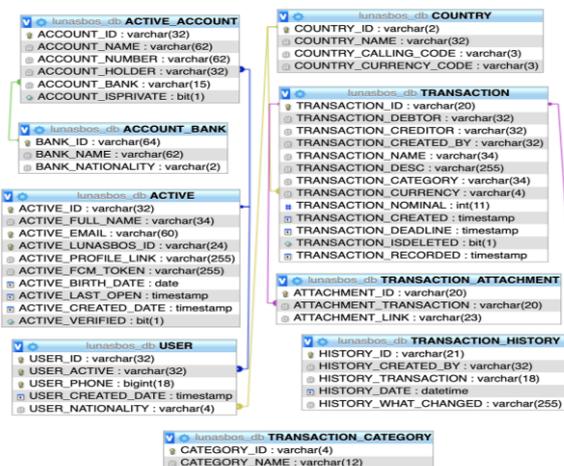


Fig. 3. Relation schema of database

Database Migration using Data Synchronization and Transactional Replication

Table I. Database Structure

Table	Rows	Size
ACCOUNT_BANK	10	48 KiB
ACTIVE	40,524	26.6 MiB
ACTIVE_ACCOUNT	107	48 KiB
COUNTRY	12	96 KiB
TRANSACTION	~292,852	167.2 MiB
TRANSACTION_ATTACHMENT	0	32 KiB
TRANSACTION_CATEGORY	14	208 KiB
TRANSACTION_HISTORY	560	352 KiB
USER	~81,695	32.2 MiB
9 tables	~415,774	226.7 MiB

Table II. Activities Description

Activities	Function
[TASK_MANAGER] I	Start loading table
[SOURCE_UNLOAD] I	Set column origin type
[SOURCE_UNLOAD] II	Unload finished, send rows
[TARGET_LOAD] I	Load finished, receive rows
[TASK_MANAGER] II	Table loading finished

During the migration process, we recorded the duration of the running activities. The running activities are described in the table II.

Activity TASK_MANAGER I recorded the start time of loading table, while TASK_MANAGER II recorded end time of loading table. In activity SOURCE_UNLOAD I, the DMS recorded the start time of unload column and setting column on the target database that is projected from the source database. In activity SOURCE_UNLOAD II, it recorded the end time of unload column and recorded the start time to send rows to target database. In TARGET_LOAD I, it recorded time of receiving rows after loading column finished. The following are logs of time recorded per activity. The sample shown on figure 4 and table III is what was recorded in the TRANSACTION table that has 294,780 rows and size of 167.2 MiB, on database migration on AWS.

The main purpose of this research is to compare between two database migration service which are Azure and AWS by recording the duration time of the full loaded database migration, the experiment results are explained in table IV.

A. Database Migration on Azure

In this process, we use Azure as Database Migration Service (DMS) or that is also called distributor. First, we need to set up the source database as publisher which is done by specifying AWS server on the source section. The next step is to specify the database target as subscriber. In this section, Azure server is filled on the target section. Before doing the migration process, we conducted speed test for Azure server, it showed that Azure server has download speed of 119.66 Mbit/s and upload speed of 335.93 Mbit/s. In the process, we found that user must manually generate and

set up schema target. Then the migration can be processed between two databases.

B. Database Migration on AWS

The second migration is by using Amazon Web Services (AWS) as Database Migration Service (DMS). Before doing the migration process, we conducted speed test for AWS server, it showed that AWS server has download speed of 119.66 Mbit/s and upload speed of 335.93 Mbit/s. As the DMS, AWS is set as distributor. Database migration process on AWS also uses AWS server as the source (publisher) and Azure as the target (subscriber). In this migration process, we found that AWS has a feature of automated generating schema target.

IV. CONCLUSION

In this study, the database migration process is carried out from one cloud to another cloud. Migration is done by equating variables in the form of databases, and base places to carry out migration by the user. The distinguishing variables in this study are the Database Migration Services (DMS), the number of rows and the size of each table. From the data obtained, it is known that the duration of database migration using transactional replication and data synchronization is influenced by the number of rows as well as size of the table, server location, also uploading and downloading speed in each database migration service.

```
09:57:56 [TASK_MANAGER] I : Start loading table 'db_main'. 'TRANSACTION' (Id = 5) by subtask 4.
                          Start load timestamp 00058E05C2B82D31 (replicationtask_util.c:707)
09:57:57 [SOURCE_UNLOAD] I: set_columns_orig_type for db_main.TRANSACTION (mysql_endpoint_metadata.c:576)
09:58:17 [SOURCE_UNLOAD] I: Unload finished for table 'db_main'. 'TRANSACTION' (Id = 5).
                          294780 rows sent. (streamcomponent.c:3392)
09:58:17 [TARGET_LOAD] I: Load finished for table 'db_main'. 'TRANSACTION' (Id = 5).
                          294780 rows received. 0 rows skipped. Volume transferred 276307688.
                          (streamcomponent.c:3663)
09:58:23 [TASK_MANAGER] I: Loading finished for table 'db_main'. 'TRANSACTION'
                          (Id = 5) by subtask 4. 294780 records transferred. (replicationtask.c:2278)
```

Fig. 4. Logs of TRANSACTION Table on AWS

Table III. TRANSACTION Table Activity Logs on AWS

Activity	Start	End	Dur
[TASK_MANAGER]	09:57:56	09:58:23	27 secs
[SOURCE_UNLOAD]	09:57:57	09:58:17	20 secs
[TARGET_LOAD] I	09:58:17	09:58:17	0 sec

Table IV. Azure and AWS Logs

Table	Duration	
	Azure	AWS
ACCOUNT_BANK	0 secs	2 secs
ACTIVE	5 secs	7 secs
ACTIVE_ACCOUNT	0 secs	7 secs
COUNTRY	2 secs	6 secs
TRANSACTION	46 secs	27 secs
TRANSACTION_ATTACHMENT	4 secs	9 secs
TRANSACTION_CATEGORY	2 secs	7 secs
TRANSACTION_HISTORY	0 secs	5 secs
USER	13 secs	9 secs

REFERENCES

1. B. Hanstein, *Rittal White Paper: Security Management for Data Centre Infrastructures*. Freidhelm LOH Group.
2. B. Nagarajan and Dr.J. Suguna, "A Review on Cloud Data Storage in Virtual Perspective" in *International Journal of Computer Science and Information Technologies*, vol. 5, May. 2014, pp. 6027-6031.
3. R. Sudeepa and Dr.H.S Guruprasad, "Effective Secure Storage and Retrieve in Cloud Computing" in *International Journal of Advanced Computing Engineering and Application (IJACEA)*, vol. 3, no. 3, June. 2014, pp. 37-42.
4. W. Dawoud, I. Takouna, C. Meinel, "Infrastructure as a Service Security: Challenges and Solutions," *IEEE Xplore*, in The 7th International Conference on Information and Systems, 2010.
5. E. Gorelik, "Cloud Computing Models", working paper in Massachusetts Institute of Technology, Cambridge, January 2013.
6. B.O. Ogbuokiri, C.N. Udanor, M.N. Agu, "Implementing Big Data Analytics for Small and Medium Enterprises (SME) Regional Growth", in *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 17, issue. 6, ver. IV, Nov-Dec. 2015, pp. 35-43.
7. M. Anshar, M. W. Ashraf, M. Fatima, "Data Migration in Cloud : A Systematic Review" in *American Scientific Research Journal For Engineering Technology and Sciences (ASRJETS)*, vol. 48, no. 1, 2018, pp. 73-89
8. V.S. Kushwah, A. Saxena, "A Security Approach for Data Migration in Cloud Computing", in *International Journal of Scientific and Research Publication*, vol. 5, issue. 3, May. 2103, pp. 1-8.
9. G. Singh, S. Satardekar, Y. Rane, and Prof. A. Srivastava, "Database Migration between Two Heterogeneous Systems" in *International Journal of Engineering Trends and Technology (IJETT)*, vol. 2, no.3, April. 2015, pp. 126-128.
10. R. Ahmadi, B.R. Cami, H. Hassanpour, "Automatic Data Migration between Two Databases with Different Structure", in *International Journal Applied Information Systems (IJ AIS)*, vol. 3, no. 3, July. 2012, pp. 23-28.
11. R. Yadav, P. Patil, U. Nevase, T. Trivedi, Prof.B. Patil, "Incremental Data Migration in Multi-database Systems using ETL Algorithm" in *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 5, no. 2, 2014, pp. 2121-2125
12. M. Elamparithi, V. Anuratha, "A Review on Database Migration Strategies, Techniques, and Tools" in *World Journal of Computer Application and Technology*, vol. 3, no. 3, 2015, pp. 41-48.
13. G. Singh, Dr.V. Laxmi, "A Review on Data Migration in Public Cloud" in *International Journal of Engineering Sciences & Research Technology (IJESRT)*, vol. 6, no. 4, April. 2017, pp. 247-251.
14. M. C. Mazilu, "Database Replication", in *Database Systems Journal*, vol. 1, no. 2, 2010, pp. 33-38.
15. S.A. Moiz, P. Sailaja, G. Venkataswamy, N.P. Supriya, "Database Replication: A Survey of Open Source and Commercial Tools", in *International Journal of Computer Application*, vol. 13, no. 6, January.2011, pp.1-8.
16. R. Jindal, "Review Paper on Database Synchronization between Local and Server" in *International Journal of Engineering Sciences and Research Technology (IJESRT)*, vol. 5, no. 7, July. 2016, pp. 1396-1400.
17. I. Shabani, B. Cico, A. Dika, "Solving Problems in Software Applications through Data Synchronization in Case of Absence of the Network", in *International Journal of Computer Science (IJCSI)*, vol. 9, issue. 1, no. 3, January. 2012, pp. 10-16.



Bambang Sudaryatno finished his Bachelor's Degree in Economy and Master's Degree in Management. Concerning in Database and Management Information Systems, now he is currently a Lecturer and Vice Rector of University of Amikom Yogyakarta.

AUTHORS PROFILE



Winda Sekar Dewi is currently a Master's Student of Informatics Engineering at University of Amikom Yogyakarta. She finished her Bachelor's Degree in Information Systems as Best of The Best Graduate. Her research interests are Database, Social Informatics, Conversion Rate Optimization, and Websites.



Ema Utami finished her Bachelor, Master, and Doctoral Degree at University of Gadjah Mada Yogyakarta. Concerning in Artificial Intelligence and Computer Vision, she is now currently a Professor and Vice Director of Postgraduate Program of University of Amikom Yogyakarta.