International Journal of Pure and Applied Mathematics
Volume 117 No. 21 2017, 531-550
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

Migration Middleware for Multiple Servers of Augmented Reality Service

[1]Hui-hwan Park, [2]Jeong-jin Lee and [3]Hoon Choi

[1]Department of Computer Science &Engineering,

Chungnam National University,

Daejeon, Korea.

knight713@cnu.ac.kr

[2]Department of Computer Science &Engineering,

Chungnam National University,

Daejeon, Korea.

likejj@cnu.ac.kr

[3]Department of Computer Science &Engineering,

Chungnam National University,

Daejeon, Korea.

hc@cnu.ac.kr

Abstract

A typical user device of Augmented Reality (AR) services, the Head Mounted Display (HMD) usually has low computing power. Hence, it is required for an HMD to utilize a server that has powerful computing resources and stores a large amount of information.When a server that receives the AR services request from the HMD is under heavy workload, the processing of the request may be delayed. Also, if the server does not have information on the image requested from the HMD, the processed result can be inaccurate. In such situation, it is better that the server passes the client's request to another server, which helps reduce the server's workload and increase the accuracy of the processed result. In this paper, we propose a migration middleware which provides the transaction migration functionality for the accuracy improvement and connection migration functionality for load balancing. This paper describes the migration middleware that provides communication between user devices and servers of AR services. It contains design ofthe middleware protocols, and experiments which show that transaction migration reduces response time of the service.The number of recommended migrationsis also

suggested based on analysis of advantages and disadvantages of transaction migration.

This middleware may be applied to any communication service with multiple servers including the AR service, toimprove accuracy and achieve load balancing.

Key Words: Augmented reality, communication middleware, multiple servers, transaction migration, connection migration.

1. Introduction

Augmented reality (AR) is a field of virtual reality and is computer graphics technique that synthesizes virtual object or information in a real world and displays them to the user[1,2]. By projecting virtual graphics on top of perceived real-life views, AR services allow users to perceive the consensus reality and the additional information simultaneously. Head Mounted Display (HMD), such as Google Glass, is a representative device for providing AR services[3]. The HMD displays information related to the perceived objects by using attached cameras to recognize objects or motions[4,5,6].

In order to provide such AR services, a high-performance computing resource is required to store and process real images and information on various objects[7,8]. However, there are limits to the storage capacities and computing resources in devices which use AR services.

Hence, it is effective to store the information required to provide the AR services in a server with sufficient storage space, and sends the user request to the server with powerful processor and sufficient storage[9,10]. In order to provide AR services in such way, communication between the device and the server is required, and communication middleware Migrim(Migration enhanced Grid Middleware) plays role in this process.

The HMD manages multiple server list, and selects on or more appropriate servers to establish a connection.

A user requests information related to the objects and motions to be recognized by sending a transaction to a connected server. The server analyzes the request received from the HMD and returns the result information to HMD.

However, since the HMD does not know the status of the server in this process, when the requests are concentrated on one server, the load is concentrated, which causes a delay in processing time and a reduction in the efficiency of resource utilization.

In addition, because HMD cannot know what information each server handles, it is difficult to guarantee the accuracy of the processing result or the information of the requested object is not in the server.

In this paper, we propose a migration middleware based on multiple servers for the AR services and describe the design and implementation of the migration middleware.

2. Proposed Model

This chapter describes the operation and functionality of migration middleware.

Migration Middleware Operation

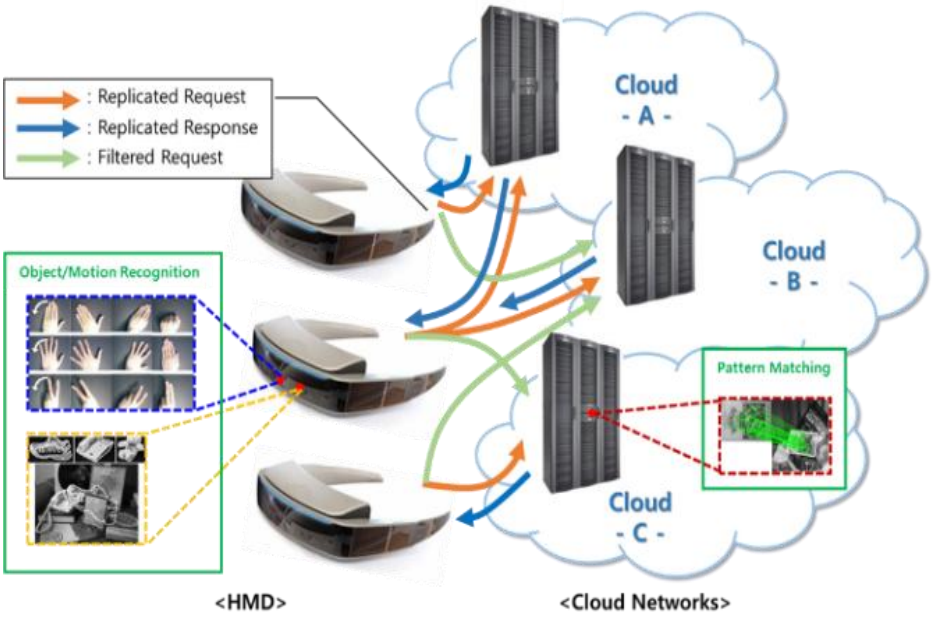

Figure 1: Operation of Migration Middleware

Figure 1 shows three users who use a HMD for AR services by collaborating with multiple servers through migration middleware.

When receiving the AR service request from the application program on the device, the middleware sends the request to the server middleware. The server middleware forwards the received request to the server application and the request is analyzed. When the analysis is completed, the server application returns processing result information about the AR services request to the device through the middleware. When the device middleware receives the result information, the device middleware forwards the result information to the application program, and finally, the result information is displayed from application program to the user.

In order to provide the AR services in the HMD, the migration middleware can be used in the operation illustrated aboveFigure 1.

Migration Middleware Functionality

We proposed a migration middleware to improve accuracy of processing results and load balancing by providing the transaction migration and connection migration functions in the AR services.

- **Transaction Migration**

Transaction migration is a function that migrates transaction requests to another server for improving the accuracy of the processing result.The procedure of transaction migration is shown in Figure 2.

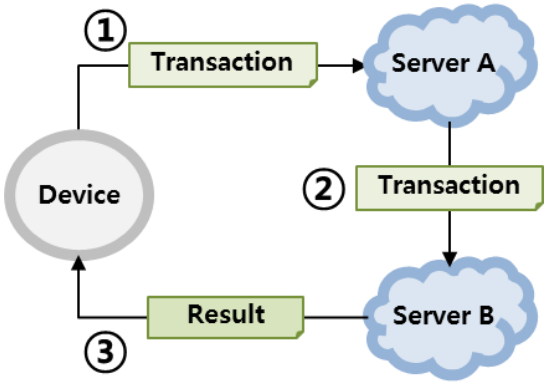

Figure 2: Procedure of Transaction Migration

- The server A receives and processes a transaction requested from a device.
- If there is no information on the transaction in the server A, or the accuracy of the processing resultof transaction is low, the transaction is migrated to the server B by sending a transaction migration message.
- Server B processes the migrated transaction and returns the result information to the device.

- **Connection Migration**

Connection migration is a function to migrate the connection of a device to another server in order to solve the load concentration problem occurring in the server. The operation procedure of the connection migration is shown in Figure 3.

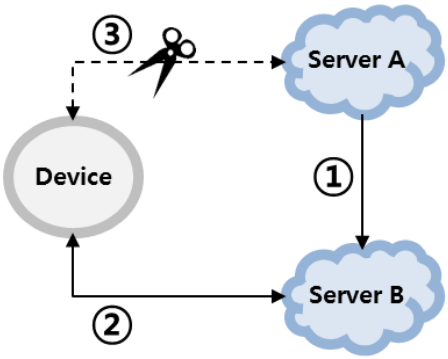

Figure 3: Procedure of Connection Migration

- If the load of server A is concentrated, a connection migration message is sent to server B.
- Server B, which has received the connection migration message, establishes a connection with the target device.
- When connection establishment between the device and server B is completed, server A disconnects the existing connection with the device.

3. Migration Middleware Design

This chapter describes the design contents of the migration middleware proposed in the paper. We define requirements of migration middleware based on the scenarios required in the environment that provides the AR services and the quality of service(QoS) for communication management. In addition, the structure and operation of migration middleware are designed using class diagram, sequence diagram and state transition diagram.

User Scenario Definition

We define the device and server side scenarios that occur when the user uses the migration middleware by using the use case diagram.

- **User scenario of Device**

We defined six user scenarios of device as shown in Figure 4 using the usecase diagram.

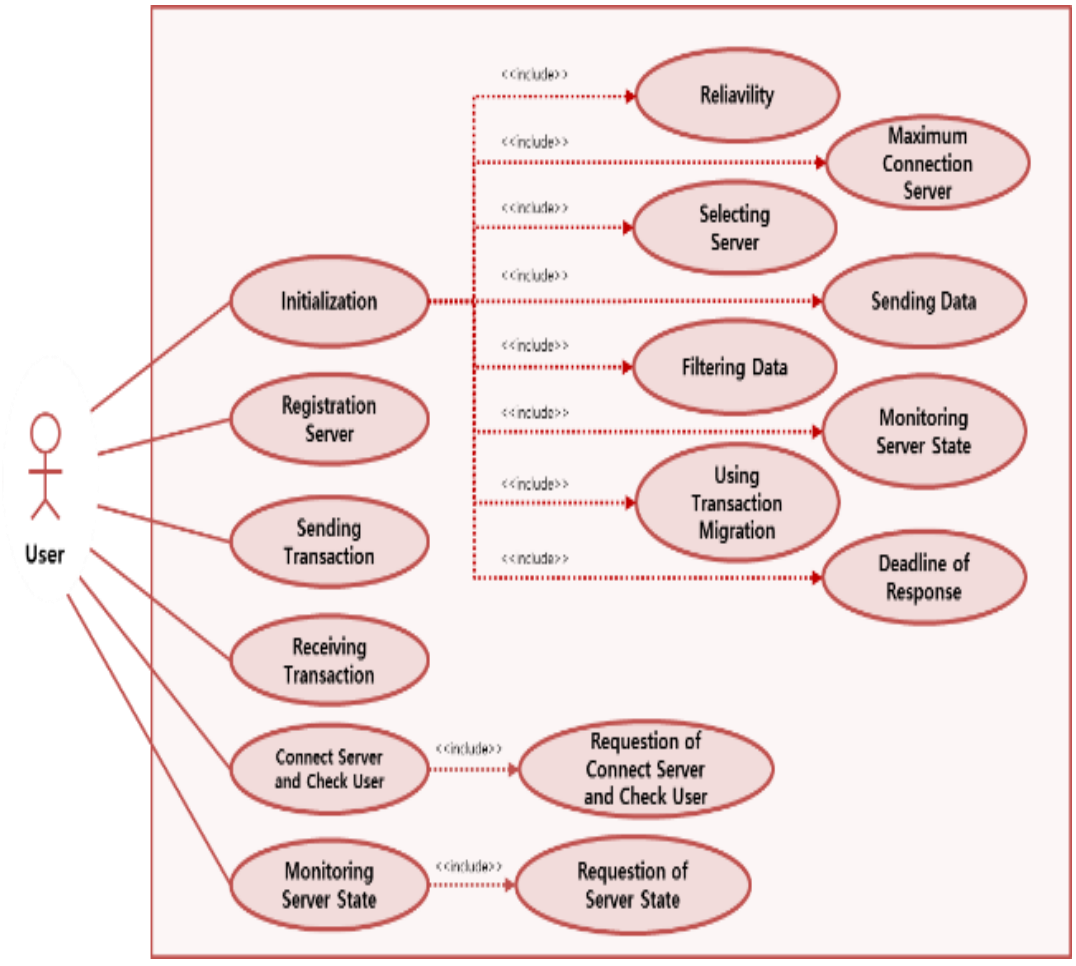

Figure 4: User Scenario in Device

- **User Scenario of Server**

We defined eight user scenarios of server as shown in Figure 5 using the use case diagram.

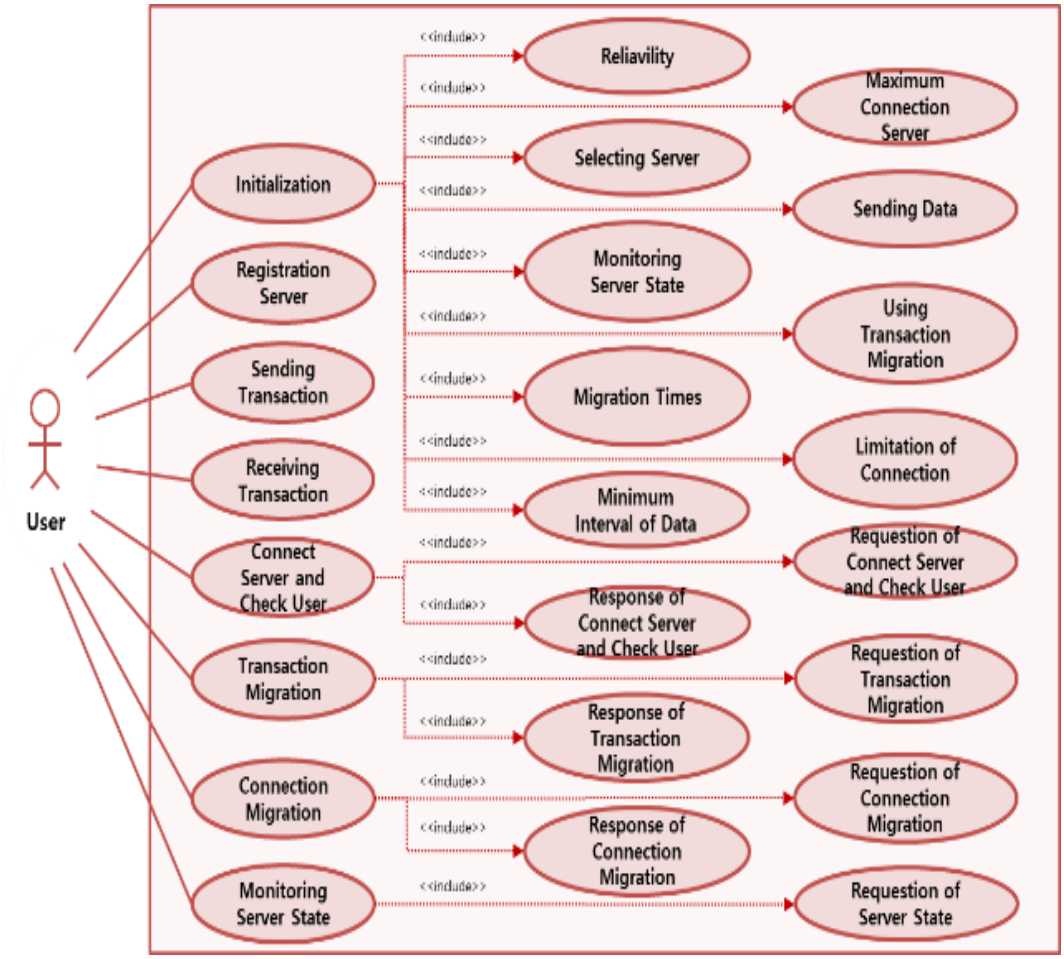

Figure 5: User Scenario in Server

Requirement Definition

We defined the device and server requirements based on the user scenario defined above.

Table 1: Definition of Requirement

	Device	Server
User Functional Requirement	28	37
System Functional Requirement	44	59
Quality of Service	13	15

- **User Functional Requirement**

We defined the user functional requirements for the middleware to operate

according to the user scenario. Table 1 shows 28 device user function requirements and 37 server user functional requirements.

- **System Functional Requirement**

We defined the user functional requirements for the middleware to operate according to the user scenario. There are 28 device user function requirements and 37 server user functional requirements.

- **Quality of Service (QoS)**

In addition, we defined QoS to efficiently provide communication functions according to middleware requirements. There are 13 device QoS and 15 server QoS.

QoS Definition

In order to provide efficient communication service, communication function that can be controlled by middleware is defined as QoS. Each QoS is an additional function that changes the middleware operation. QoS is classified into server management QoS, transaction management QoS, migration management QoS and communication management QoS.

- **Server Management QoS**

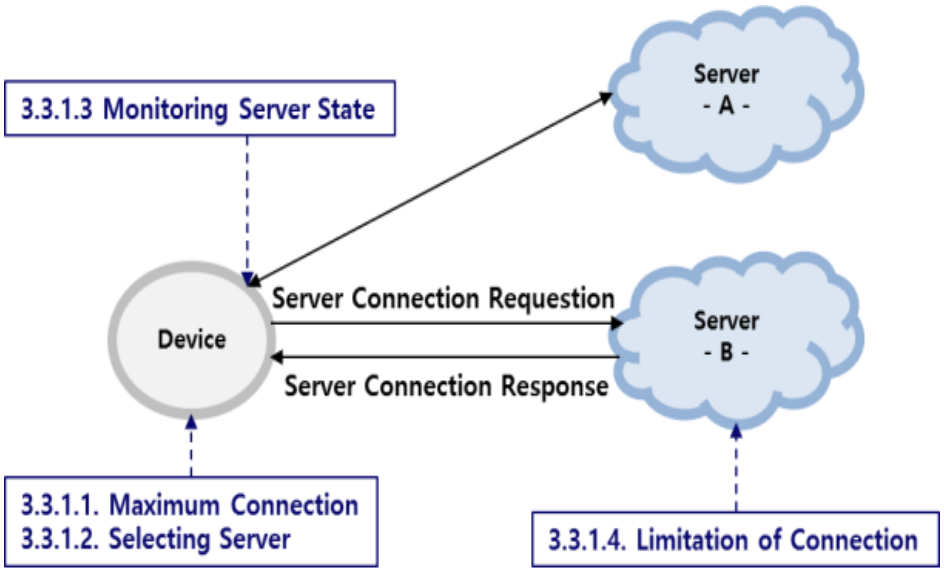

Figure 6: Server Management QoS

Figure 6 shows Server management QoS which is related to server status information and connection between device and server. It is used in the process of establishing the connection and maintaining the state information of the server after establishing the connection.

- **Transaction Management QoS**

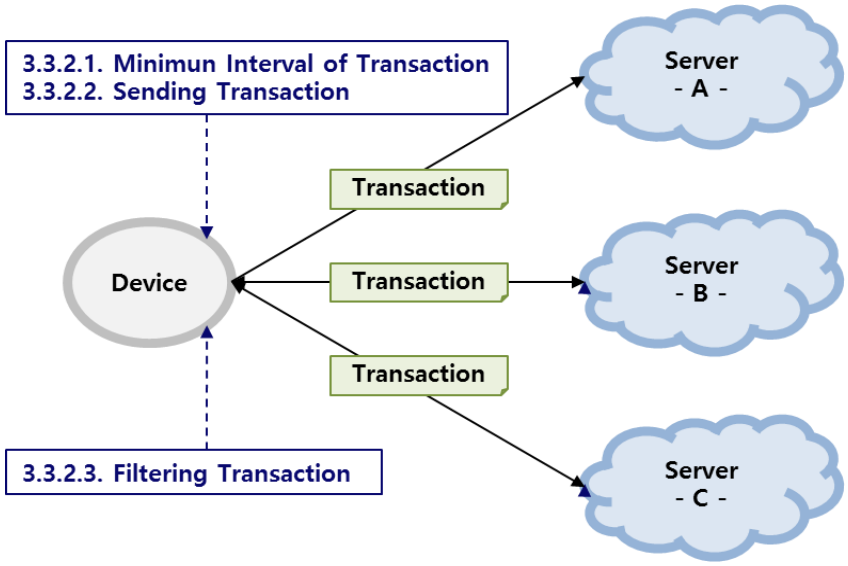

Figure 7: Transaction Management QoS

Figure 7 shows Transaction management QoS which is related to sending and receiving transactions. It is used when a device sends a transaction to a server and receives a transaction result from server.

- **Migration management QoS**

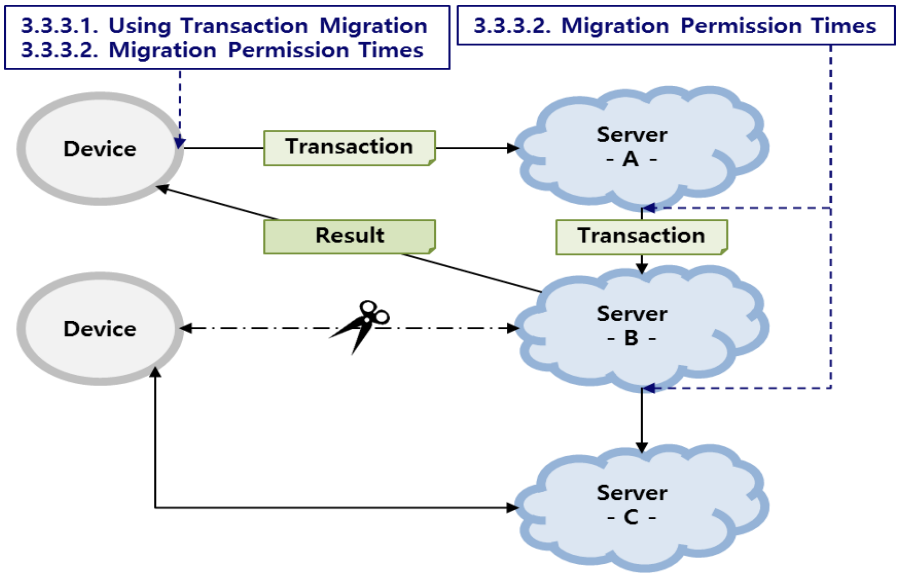

Figure 8: Migration Management QoS

Figure 8 showsMigration management QoS which is related to the transfer of the server to another server. It is used to process connection migration and transaction migration.

- **Communication Management QoS**

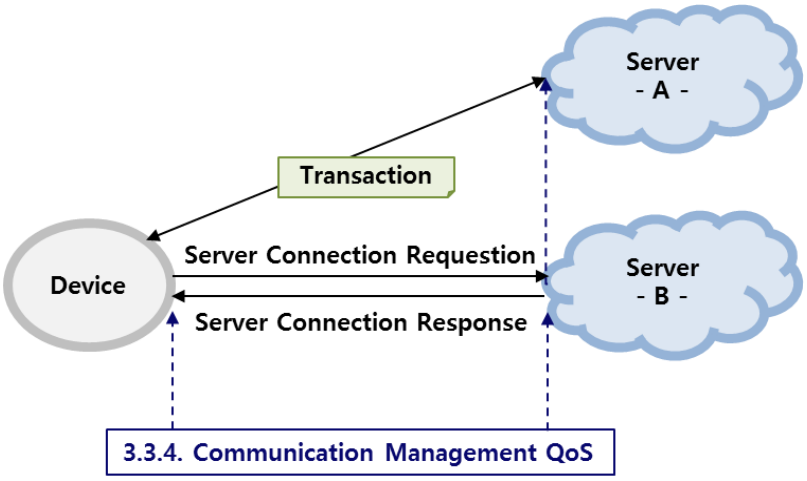

Figure 9: Communication Management QoS

Figure 9 showsCommunication management QoS which is the QoS associated with communication between device and server. It is used to send and receive request/response messages or transactions.

Design of Module

In this paper, we created class diagrams for each module and integrated them.Figure 10 shows the design of module.

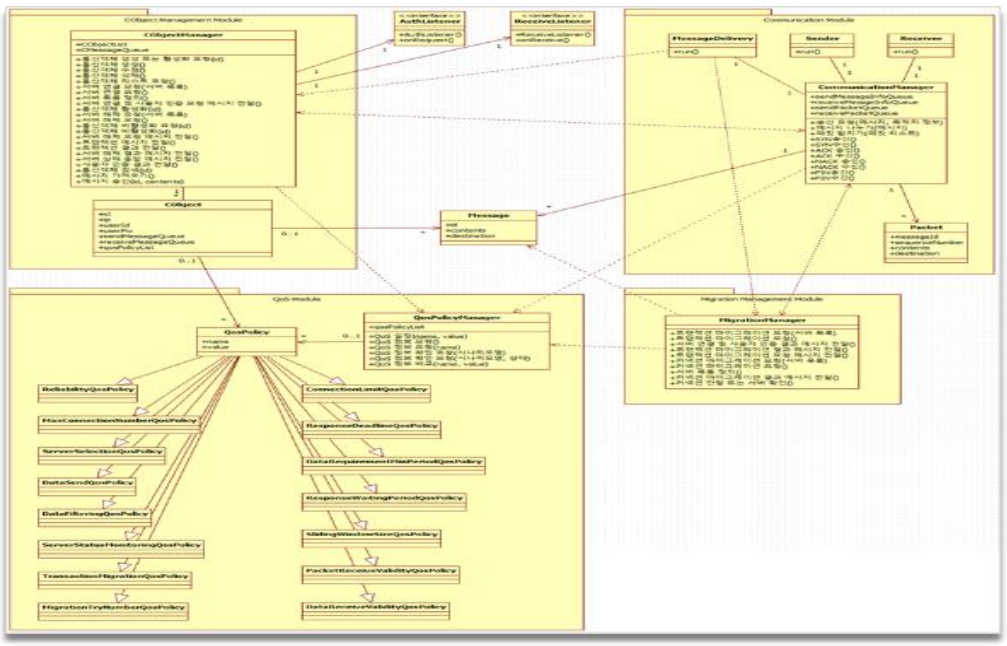

Figure 10: Design of Total Module

Design of Call and Message Flow between Module

The procedures of each module of the migration middleware according to the scenario are designed using the sequence diagram. Figure 11 shows16 sequence diagrams which were created for each scenario.

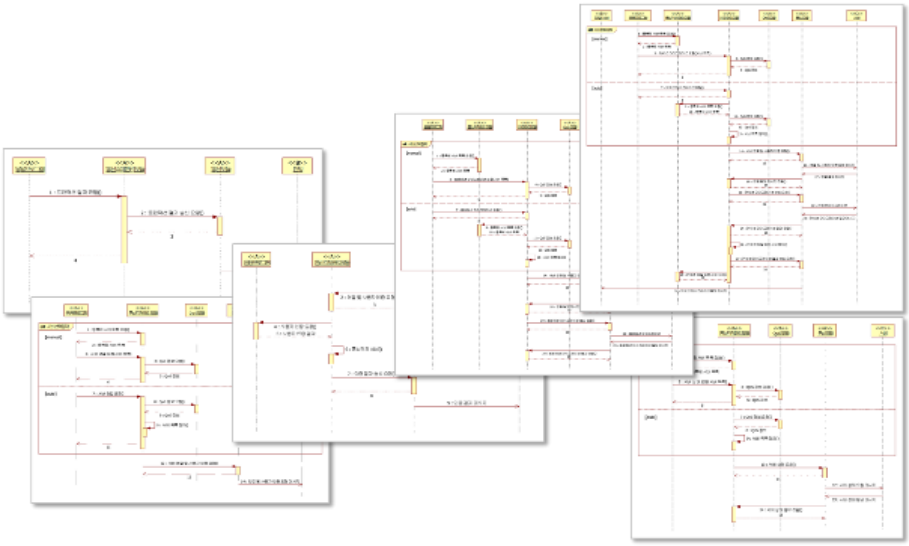

Figure 11: Message and Call Flow between Modules

Design of State Transition

When the migration middleware operates according to the defined scenario, it is designed to change the state of the middleware. Figure 12 shows12 state transition diagrams which were created for each scenario.

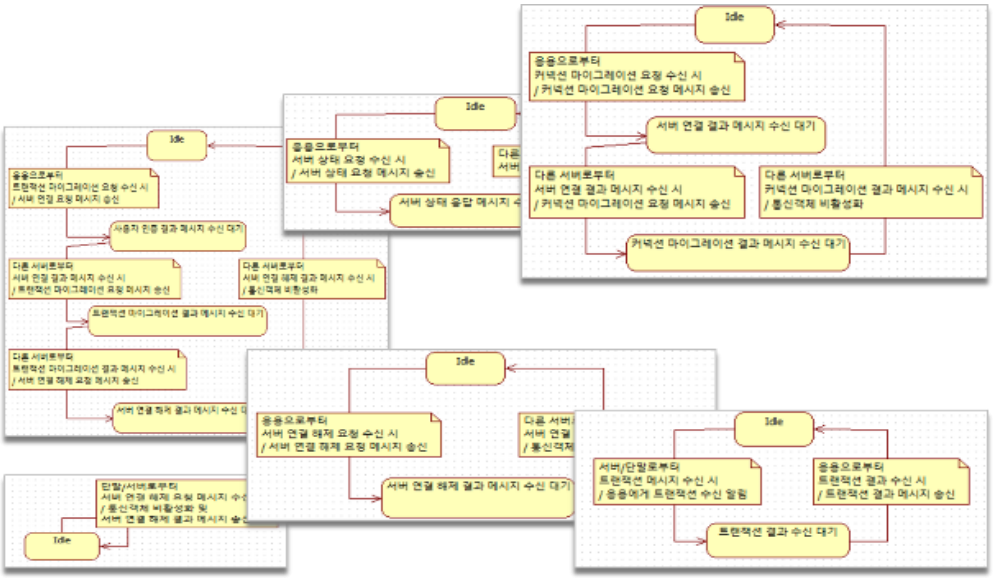

Figure 12: Design of State Transition

4. Implementation and Test

This chapter describe development environment, implementation of migration middleware and improvement of performance through tests.

Development Environment

Table 2 shows the development environment for implementing the migration middleware designed in this paper. The server middleware was implemented with a total of 3981 lines, including 9 CPP files and 11 header files. The device middleware was implemented with a total of 2288 lines, including 10 JAVA files and 1 interface files.

Table 2: Development Environment

	Server	**device**
Test device	PC (Desktop)	Samsung Galaxy S4
Operating System	Window 8	Android 4.4.2 (KitKat)
Language	C++	Java
Development program	Visual Studio 2010	Android Studio 1.5.1

Tests to Compare the Time Required for Transaction Re-Request

The response time of the request increases when the processing is performed by another server because the accuracy of the processing result of the user's request is low.

Therefore, we confirmed through performance tests how much of reduced response time the user would wait when using the transaction migration proposed in this paper.

In this test, we compare the processing time of the method where the user directly retransmits the request to other server and the method of transferring the request to the other server by using the proposed transaction migration when the result corresponding to user’s request is low.

- **Tests Scenario**

The purpose of transaction migration is not only to improve the accuracy of the processing results, but also to perform re-requests quickly.

Therefore, we performed a test to compare the execution time of proposed transaction migration method with the existing method that does not use transaction migration.

- **If the Server is Not using Transaction Migration**

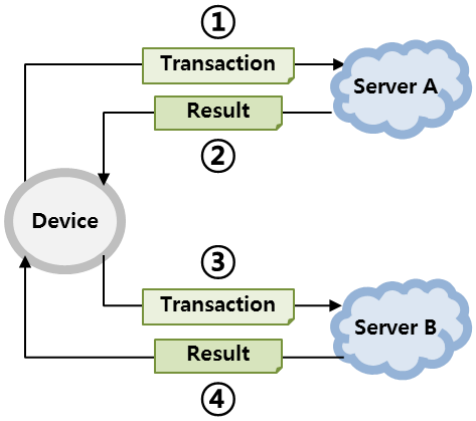

Figure 13: Not Using Transaction Migration

The server returns processing results for all transactions to the device. If the accuracy of the result is less than reference accuracy of 50%, the device requests a transaction to another server. This is modeled when the result received from the server is not what the user wants.Figure 13 shows this procedure.

- **If the server is using Transaction Migration**

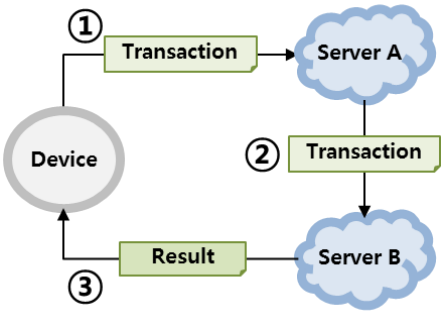

Figure 14: Using Transaction Migration

The server returns the transaction processing result only when the accuracy of the transaction processing result is more than 50%. If the accuracy of the transaction processing result is less than 50%, the transaction is migrated to another server via transaction migration.Figure 14 shows this procedure

According to Table 3, the time required to transmit a 1 MB transaction was measured by diving a wireless section transmitted to a near-field wireless AP and a wired section transmitted to a server through a plurality of hops. We corrected the time required for transaction migration based on the measured time taken by each segment.

Table 3: Processing Time According to Transition Section

Transmission section	device <—> Server		Server <—> Server
Time	wireless	wired	wired
	172,329	822,332	974,236

Assuming a maximum of 4 transaction migrations, we measured the processing time according to the number of transaction migrations. In this process, we excluded the time required to process the transaction at the application of the server.

- **Test Result and Analysis**

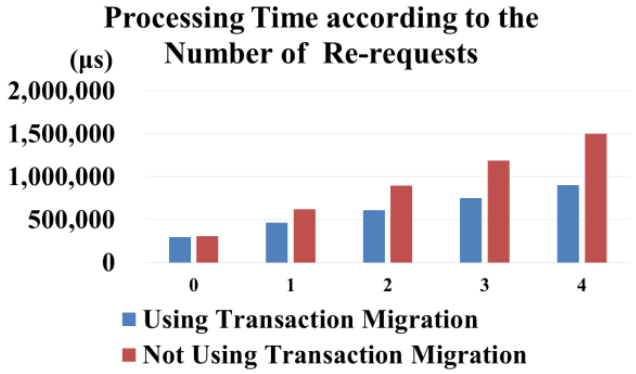

Figure 15: Comparison of Processing Time for Re-Request

Figure 15 shows test result. It has been found that it takes less time to use transaction migration than when the user do not use transaction migration. This result is presumed to be due to the difference in the number of transaction transmission hops and the difference in transmission time in the wired / wireless section.

Accepted Transaction Tests According to the Number of Transactions migration

In this Test, it is measured that number of occurrences and processing time of accepted transactions according to the number of migration in the transaction migration process. We proposed the recommended number of uses of transaction migration based on test.

- **Tests Scenario**

The purpose of transaction migration is to improve the accuracy of the processing result. If the result of the transaction processing does not reach reference accuracy of 50%, to execute transaction migration is possible to expect the result more than reference accuracy. Therefore, it was measured that the number of accepted transactions and the processing time according to the number of transaction migration.Figure 16 shows this scenario.

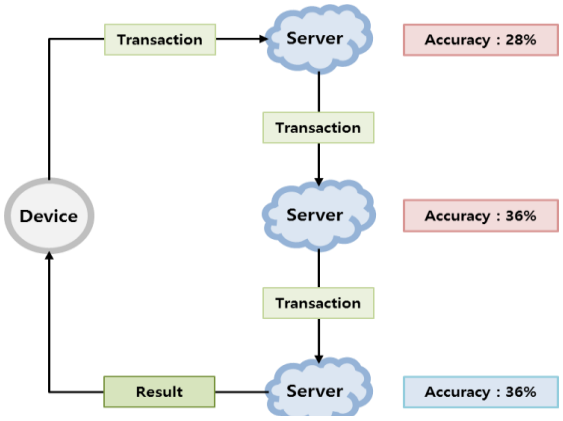

Figure 16: Scenario of Accepted Transaction Test

Test environment configured 5 servers and 10 devices. Each device has a connection with one server and transmits 100 transactions, so that a total of 10 terminals transmits 1000 transactions of 1 MB. In this test, the reference accuracy is defined as 50%. If the accuracy of the transaction processing result is more than 50%, it is accepted. If not, the result is rejected. If the processing result of a specific transaction is rejected in the server, the transaction migration is performed to another server. The test was conducted by changing the number limit of transaction migration from 0 times to 5 times. Whether or not the accuracy of processed result on each server is 50% or more is determined by generating a random number in the [0 to 1). If a random number obtains a number of 0.5 or more, it is processed as accuracy of 50% or more.

- **Test Results and Analysis**

A. The Number of Accepted Transactions According to Number of Transaction Migrations

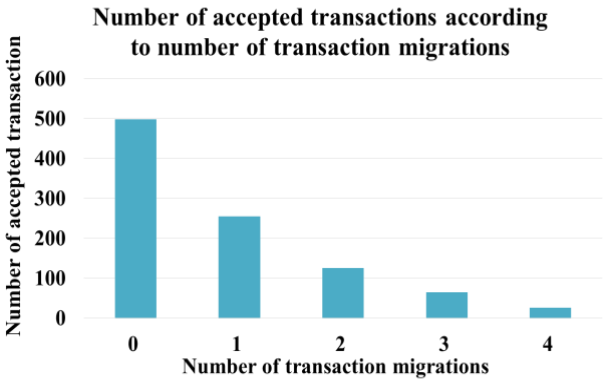

Figure 17: Number of Accepted Transactions According to Number of Transaction Migrations

Figure 17 shows the number of accepted transactions for each number of transaction migrations which is limited to four times. It was confirmed that about 50% of the transactions, which is the standard accuracy used in the test,

pass each time.

B. The Number of Cumulative Accepted Transactions According to Number of Transaction Migrations

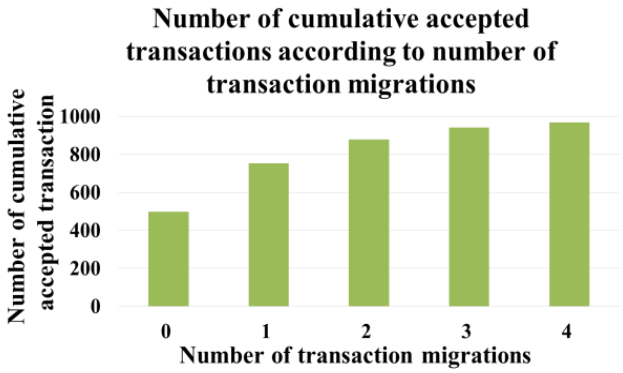

Figure 18: Number of Cumulative Accepted Transactions According to Number of Transaction Migrations

Figure 18 shows the number of cumulative accepted transactions according to the number of transaction migration limited four times. If the accuracy of the result of performing the transaction migration for the rejected transaction is more than 50%, it is regarded as the acceptance. As a result, when the transaction migration is performed, we confirmed that the rate of the accepted transactions is increased.

C. Comparison of the Number of Cumulative Accepted Transactions and Cumulative Processing Time

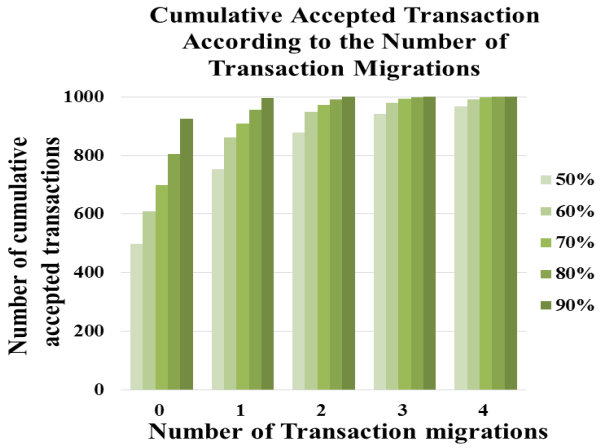

Figure 19: Cumulative Accepted Transaction According to the Number of Transaction Migrations

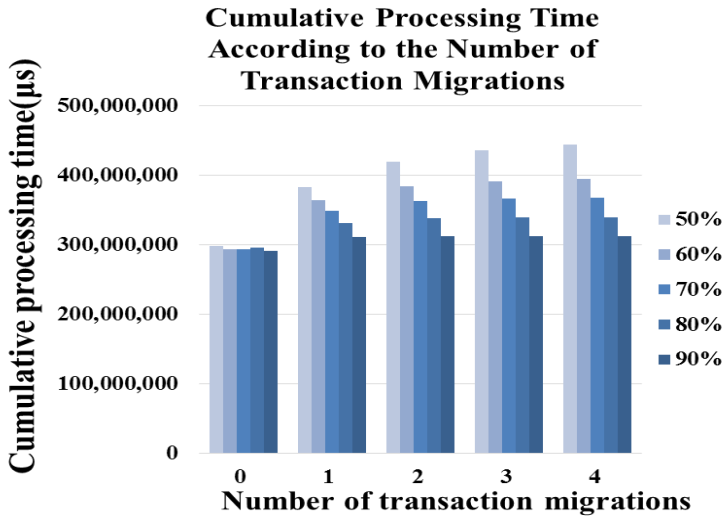

Figure 20: Cumulative Processing Time According to the Number of Transaction Migrations

Figure 19 and Figure 20 show the number of cumulative accepted transactions and the cumulative processing time according to the number of transaction migrations. It is intuitive to note that using transaction migration increases the rate of cumulative accepted transactions, but cumulative processing time also increases. This increase was verified with specific numerical values through the test.

In case that the accuracy of the AR service is insufficient, it is more efficient in terms of time to re-request using the transaction migration than to request directly to another server. Therefore, it would be efficient to use transaction migration. As long as the user's waiting time is not too long, it would be efficient to use transaction migration.

Therefore, similarly to this test, it is recommended to improve accuracy by performing transaction migration once or twice for rejected transactions. This number can increase or decrease depending on changes in the environmental factors that can affect the processing time, such as the network environment or the processing speed of the server.

5. Conclusion

In this paper, we designed and implemented a migration middleware that supports communication to provide augmented reality service. Through the proposed middleware, tests were performed to distribute the load generated by the server in providing the augmented reality service and to improve the accuracy of the service.

The load balancing effect of connection migration is confirmed through load

balancing test using connection migration performed inLee's study[11]. When load balancing is induced through connection migration, it is confirmed that load balancing can be expected rather than not using connection migration. Also, it is confirmed that uniform connection migration is more effective than non-uniform connection migration

It is confirmed from the viewpoint of processing time is effective when re-requesting using transaction migration through the comparison test of time required for re-request, because when transaction migration is used, it is possible to avoid the transmission delay of the wireless section and reduce the number of transaction transfer hops. Also, since the transmission speed was directly measured and corrected in a complicated network, it was possible to infer the time required to use the transaction migration function of the middleware in a real multi-server environment.

We analyzed the advantages and disadvantages of transaction migration through accepted transaction test according to the number of transaction migrations. Transactional migration has the disadvantage of increasing the amount of time it takes in proportion to the number of transaction migrations used, but it has been confirmed that the percentage of accepted transactions can be increased. Therefore, the test have shown that using transaction migration is advantageous from an accuracy perspective. Also, we recommend using only 1~2 times transaction migration in consideration of the response time of the user and the time spent in the complicated network environment.

References

[1] Van Krevelen D.W.F., Poelman R., A survey of augmented reality technologies, applications and limitations, International Journal of Virtual Reality 9(2) (2010), 1-20.

[2] Olsson T., Salo M., Online user survey on current mobile augmented reality applications, 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (2011), 75-84.

[3] Nee A.Y.C., Ong S.K., Chryssolouris G., Mourtzis D., Augmented reality applications in design and manufacturing, CIRP Annals-manufacturing technology 61(2) (2012), 657-679.

[4] Han J., Study on the head mounted display (HMD)-based VR contents and producing method, Indian Journal of Science and Technology 9 (25) (2016), 1-10.

[5] Sundaram V.M., Vasudevan S.K., Santhosh C., Kumar R.B., Kumar G.D., An augmented reality application with leap and android, Indian Journal of Science and Technology 8(7) (2015), 678-682.

[6] Vasudevan SK, Vivek C, Srivathsan S. An intelligent boxing application through augmented reality for two users–human computer interaction attempt, Indian Journal of Science and Technology 8(34) (2015), 1-7.

[7] Olsson T., Kärkkäinen T., Lagerstam E., Ventä-Olkkonen L., User evaluation of mobile augmented reality scenarios, Journal of Ambient Intelligence and Smart Environments 4(1) (2012), 29-47.

[8] Kurniawan E., Lee B.G., Lee S.H., An Implementation of a Cloud Service based Augmented Reality for Improved Interactivity, Indian Journal of Science and Technology 9(46) (2016), 1-4.

[9] Ohta Y., Sugaya Y., Igarashi H., Ohtsuki T., Taguchi K., Share-Z: Client/server depth sensing for see-through head-mounted displays, Presence 11(2) (2002), 176-188.

[10] Geiger P., Pryss R., Schickler M., Reichert M. Engineering an advanced location-based augmented reality engine for smart mobile devices (2013), 1-42.

[11] Lee S., Yoon G., Choi H., Migration Mechanism of Communication Process for Load Balancing and Accuracy Improvement, KIISE Transactions on Computing Practices 22(1) (2016), 26-31.