CloudIoT-based Jukebox Platform: A Music Player for Mobile Users in Café

Byungseok Kang[1], Joohyun Lee[2], Ovidiu Bagdasar[1], Hyunseung Choo[2]

[1] School of Electronics, Computing and Mathematics, University of Derby, UK
[2] College of Information and Communication Engineering, Sungkyunkwan University, South Korea
b.kang@derby.ac.uk, joohyun7@skku.edu, o.bagdasar@derby.ac.uk, choo@skku.edu

Abstract

Contents services have been provided to people in a variety of ways. Jukebox service is one of the contents streaming which provides an automated music-playing service. User inserts coin and presses a play button, the jukebox automatically selects and plays the record. The Disk Jockey (DJ) in Korean cafeteria (café) received contents desired of customer and played them through the speakers in the store. In this paper, we propose a service platform that reinvented the Korean café DJ in an integrated environment of IoT and cloud computing. The user in a store can request contents (music, video, and message) through the service platform. The contents are provided through the public screen and speaker in the store where the user is located. This allows people in the same location store to enjoy the contents together. The user information and the usage history are collected and managed in the cloud. Therefore, users can receive customized services regardless of stores. We compare our platform to exist services. As a result of the performance evaluation, the proposed platform shows that contents can be efficiently provided to users and adapts IoT-Cloud integrated environments.

Keywords: Cloud computing, Internet of things, Location-based service, Multimedia services, Real-time sharing service

1 Introduction

Users have been served contents in various offline ways for a long time. Examples of typical services are Jukebox and DJ. Jukebox existed a lot from 1940s to 1960s, especially in the 1950s. Jukebox has records so that when users insert a coin, they can listen to the music. Users can listen music with higher volume and better sound quality through Jukebox. In Korea, DJs in music café stores called "Dabang" were popular during the 1970s and 1980s. The café DJ shares music and messages to the users in the store. The music is played with the public speaker of the store. People in the same store could enjoy it together through these services. Jukebox and DJ services show that users enjoy not only contents but also sharing with others in the same location.

Advances in cloud computing become possible to provide personalized online streaming services based on cloud [1]. It can also provide a lot of contents. However, users have difficulty in selecting the contents that suits them from a lot of contents. A lot of research has been proposed on algorithms and systems that recommend user preferences as a solution to these problems [2-5]. In particular, a cloud-based recommendation system recommends contents based on an age, location, usage history, and preferences [6-7]. However, such a system is only a recommendation and does not guarantee the satisfaction of the user. It is also hard to offer a personal online streaming service that user can enjoy with other users in real-time. A platform for ensuring user satisfaction and for users to enjoy contents with others has become necessary.

In cloud media DJ service, the most important factor is how to replace the music café DJ. This is because DJ had led the music café. We use a cloud server instead of the DJ. The song which was played by DJ is replaced by the streaming REST API of the music and videos service provider. The behavior of users who requested the song by letter or speech directly to the DJ is replaced by the application such as the Android and web one. The sound of music requested is heard through speakers in the café and the stories of customers is seen through the screen in the café.

This paper proposes and implements a cloud jukebox platform that reinvent a DJ of a music café. We propose and implement a new service platform that was not previously available. The platform is based on an integrated environment of Internet of Things (IoT) and cloud computing. Based on a single cloud, services provided by this platform allow users to request contents services directly. User information and usage history are collected and analyzed in the cloud. Even when the users visit other stores, they can receive customized contents services based on their information and previous usage history. We analyze

*Corresponding Author: Hyunseung Choo; E-mail: choo@skku.edu
DOI: 10.3966/160792642020092105012

and evaluate the services of the proposed platform, focusing on the functional aspects. The platform provides four main services. Types of services are music, video, message, and combined service. All services on the platform are based on the location of users. It also combines the advantages of online streaming and offline services to provide O2O streaming services. User can share contents with other users in real time through the platform and receive scalability and convenience services through speakers and screens.

The remainder of this paper is organized as follows. In Section 2, we review the basic concept of IoT and Cloud computing. Section 3 presents a proposed platform. Implementation and the evaluation are discussed in Section 4 and 5, respectively. In section 6, we conclude this paper.

2 Related Work

With the development of mobile devices and the cloud, various kinds of offline services that people have been handling is replaced by online ones. In the past, Korean music café DJ performed the role of playing music requested and reciting the story of customers. We replace this offline service with an online one consisting of a public cloud, a streaming server, a streaming client, and a mobile device. In this section, we introduce state-of-the-arts of cloud, streaming and web services.

2.1 Recent Internet of Things and Cloud Computing

Internet of Things (IoT) is a technology that allows communication between objects by connecting to the Internet with sensor and communication functions. The kind of object include a user device, home appliance, and the like things. Objects connected by the Internet exchange and analyze data. Cloud computing provides an environment where IT services such as storing data and providing contents are available anytime, anywhere. It also provides large scale capacity and processing by using servers on the Internet. IoT technology has limited processing capacity and storage problems. This leads to performance and security problems.

A solution to these problems is presented to integrate IoT with cloud computing. The technology that combines IoT and cloud computing is called CloudIoT [8]. This new paradigm provides many services and applications to users. In recent years, the two themes have been popular and researched. Integrated research and application programs are also actively being proposed. In general, the infinite capabilities and resources of the cloud can solve the problems of the IoT: storage, processing, and communication. CloudIoT paradigm can lead to big advances in life and activity of human [9]. Multimedia applications based on this paradigm provide efficient services and build new businesses [10].

In the cloud system, virtual machine plays an important role. It features a virtualization of the operating system using hypervisor on existing physical hardware. It is separated from the host operating system, because the operating system is virtualized. Therefore, it is possible to use an operating system different from that of the host operating system, and the host and other VMs can be safely maintained even when one VM is attacked due to the structure separated from the host operating system. However, there are disadvantages as well. The amount of disk space required to virtualize an operating system is very large. In addition, once it is a virtualized operating system, there are performance penalties compared to the host operating system.

2.2 Online and Offline DJ

Disk Jockey is a person who leads listeners to music. In Korea, music cafeterias were popular in the 1960s and 1980s. The person who plays the music in the music café is called DJ. In addition, DJ has become universal in the world. It can be easily seen at festivals where contents can be enjoyed all over the world. These DJ services can be classified as offline DJs working offline and online DJs working online. Typically, offline DJ works radio, dance club, etc. On the other hand, online DJ works as an internet broadcaster and music streamer. The description of each DJ is as follows.

Offline DJ is a person who plays the music recorded in real time to users located in the same store. Typical DJ types are digital radio DJ and club DJ. Radio DJ is a basic DJ who can listen to music from the station. Radio DJ plays music on a digital radio station. Club DJ plays a variety of music in clubs and encourages people to dance. Club DJs also reassemble their music and let listeners hear it. They act as a mediator for the music recorded to the users and recreate the music. Users can enjoy more music with DJ.

Online DJ is a person who broadcasts online. It is usually called broadcaster, and it is called BJ (Broadcasting Jockey) in Korea. They get popularity from users and attempt to build new businesses. With the development of mobile devices and online streaming services, the number of online DJs who provide contents has also increased. As the number of users enjoying contents increases, the service area of DJs is also increasing. Their prospects are also high.

2.3 Streaming Services

The service for streaming the music can be classified into an on-demand method and a streaming radio method. On demand is a way for users to select and listen to music. The streaming radio method automatically selects and plays the music on the platform. The on-demand method allows a user to select a music to listen to, store it in a personal folder, and listen at any time. In addition, the user can edit the number of times of playback and the order of the music according to the user's preference. However, the user lacks a curating system to listen to and listen to unfamiliar music.

On the other hand, the streaming radio service recommends various genres of music, and users can access new music. It can also be used as a free service based on advertisements. However, it is difficult to listen only to a user's favorite music, and the number of times of play back and order of music are limited. Currently, streaming radio service is gaining popularity in overseas music market. Streaming radio-based services can create various business models when many users are secured.

Currently, many companies around the world provide streaming services. A typical music streaming service is Apple music, and is a global music streaming service released by Apple in the U.S. [11]. A typical video streaming service is YouTube [12]. It is Google free video sharing site, a global platform that allows users to create and share video contents. YouTube helps and provides customized contents for over 1 billion users [13]. In recent years, cloud-based streaming services have been one of the core innovations in digital streaming service. Many music providers like Apple, Amazon, Google and Microsoft are currently offering cloud-based streaming services. The advantage of cloud-based streaming services is that users can store and easily access the contents [14]. Cloud-based streaming services are actively researched, and applications that provide contents to users are also developed. Users receive a variety of contents and services. However, users have difficulty in selecting their preferred services in a lot of contents. Contents providers need a way to effectively inform users a lot of contents. The competitiveness of contents providers is determined by providing contents that users prefer. Algorithm and applications for providing customized contents to users are being studied [15-16].

2.4 Native Web Application

Recently, mobile device gives tremendous of services to the human being [17]. The ability to access the Internet anytime and anywhere with a smartphone rather than a desktop is one of the greatest achievements of recent times. We can communicate with external services at any time via a smartphone and save our data unlimited in external spaces. In addition, the quality of wireless network communications is evolving very quickly, making the advance of mobile devices faster [18]. The application services evolved together as mobile devices advanced. Now, most of the online services like banking, shopping, booking can be run as smartphone applications. The application services are expected to take a large portion in the IT era in the future.

The applications that run mostly on mobile devices are native and web ones [19]. Native applications run directly on mobile operating systems without going through another execution environment. These features make native applications perform faster and can gain most of permissions on mobile devices. However, there is a disadvantage in native applications. It is platform dependent so the native applications developed for Android platform cannot be executed on other operating systems such as iOS. Web applications run through web view in web browsers. Therefore, it has lower performance than native applications. However, the HTML5 [20] standard used on the web is compliant with all of the web browsers. If a developer creates a web application, users can run it on any platform without any operating system limitations using web browsers. An application developer should choose the right application style, keeping in mind the scenarios in which the applications they want to develop, how much their funds, and how much performance they need.

3 Proposed Platform

Cloud jukebox is an online service that changed the music café, a proven offline service in Korea long ago. Café customers can use the mobile or web application to access the service. When a customer requests a music, a video, or a message through the application in a café, the requested contents are played in the café where the customer is located currently. Customers can listen to their own playlists in cafeterias, not playlists of the owner of the café. Customers can share music, videos and stories with their friends and co-workers in the café. In addition, the café automatically plays the music that the customer previously requested, even if the customer does not request it using the customized recommendation function. The customer is satisfied with the service and may become a regular customer of the café.

3.1 Platform Overview

Cloud Media DJ (CMDJ) provides customized contents services for mobile users in an integrated environment of IoT and cloud computing [21-22]. The platform consists of user device, cloud, streaming client, public screens and speakers. There are devices of users, streaming clients, public screens and speakers in one store. The cloud is external. User select for contents want to receive in stores where they are located, and provided it as public screens and speakers. The user receives one or more of the music, the video, and the messaging services. The user device is a communication device that uses applications in the network environment. For example, user devices could be a smart phone, tablet device, or a laptop. The cloud is a server that can manage all stores. It is possible to manage all stores with one cloud. The platform builds cloud based on Google cloud platform. Cloud consists of operation server and streaming servers. The operation server collects user information and performs analysis. The detailed functions of the operation server are described in the following paragraphs. Streaming server is an external affiliated server with contents data. For example, it could be Spotify or YouTube which hold and provide music and video. Depending on the type of contents requested by the user, the streaming server provides the contents data. The streaming server that provides the music and the streaming server that provides the video may differ and include multiple affiliate streaming servers.

The streaming client is located in the store, receives contents data, and transmits it to the public IoT devices [23-24]. The streaming client is a server that receives streaming data in a store. It connects the cloud with the screen and speaker. In this platform, it is implemented as a web page based on HTML which is highly usable and light. The screens and the speakers connect with the streaming client to play the contents received by the streaming client. It is also implemented as an HTML-based web page and runs with minimal hardware requirements. Figure 1 shows the overall structure of platform. The platform allows a single cloud to manage multiple stores. When a user requests contents desired through the application, the cloud collects the contents data requested. The collected data is sent to a streaming server, and the contents data is transmitted to a streaming client at a store where the user is located. The streaming client transmits the contents to the public screens and speakers, so that the contents to be played. Streaming client can be implemented with minimum specifications so it can be used lightly in stores. Users in the same store receive contents services through public screens and speakers.

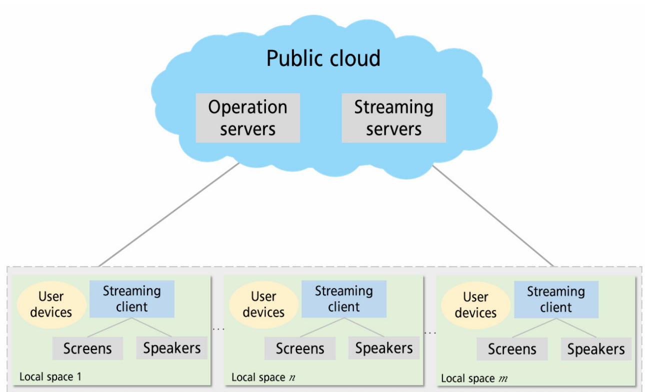

Figure 1. Basic structure of cloud jukebox platform. Our system covers multiple spaces. Each space has certain number of IoT equipment

A cloud consists of an operation server and a streaming server. The operation server of platform mainly performs user information collection and analysis functions. Figure 2 shows the structure of the operation server. The operation server consists of device communication, user data storage, information analysis, and contents recommendation. Device communication is an interface with a user device. The interface service as the communication module that the user device can access. Device communication sends and receives user information between user devices and user data storage. User data storage stores user information and usage history. Information analysis part analyzes and integrates user data in operation servers. After that, this data is transmitted to the content recommendation module. Contents recommendation transmits the contents list to the streaming server so that the contents are played.

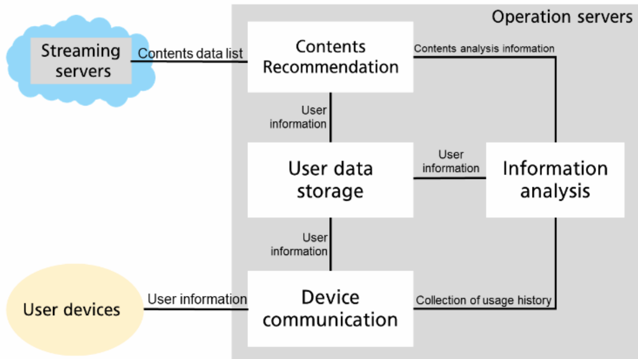

Figure 2. Structure of operation server

3.2 Service Design

The proposed platform provides four main services. The types of services are music, video, messaging, and combined service, and perform the following operations. We have designed the services that can be provided through this platform. These are the services that users can provide through this platform. The following paragraphs describe each service and its operation and scenarios.

3.2.1 Music Streaming Service

Music streaming service provides the music that users want to listen in the store where the user is located. Users request services by searching and selecting the music they want through the application. The components for music streaming service are the user device, operation server, streaming server, streaming client, public screens, and speakers. The operation server and the streaming server can provide services to the store. The streaming client, screen, and speaker exist in the store. Operation server and streaming server serve all stores. Streaming client, public screens, and speakers are in each store. The user can view music related information on user device or public screen. In addition, the user searches for a music desired and directly select for it. The selected music is provided to the public speakers of the store where the user is located, and the music information is provided to the screens. The music information includes a title, a singer, and an album photo. The user is provided with a music directly selected on the speakers. It allows users to provide customized music and share the music with other users in real time. Also, the store manager automatically provides customized music services without having to manage the music playing lists. Store managers can exclude music genres that they do not like or music that many people do not like.

3.2.2 Video Streaming Service

The video streaming service provides the video desired by the user. The video is played on the public screens and speakers in the store where the user is located. The service provides with various videos including music video, by public speakers and screens. Components for video streaming service include user device, operation server, streaming server, streaming client, screens, and speakers. The video streaming service is the same as the music streaming service, but the streaming server is different. The streaming server must be a server capable of providing video. For example, there is YouTube. The user searches for videos desired in the application or the web at the store, request directly. The video requested by the user is played on the screens and speakers of the store where the user is located. Users receive customized video service by receiving the video requested as a public speaker and a screen. In addition, store manager automatically provides video service that users can satisfy without having to manually control the video playing list.

3.2.3 Messaging Service

The messaging service is a service that allows users to request a message and receive it on a public screen. Users request to a message that they want to leave or share with the people in the store. This message is displayed on the screen. Messaging service provides users to share information about simple messages or special events with people who is in the same store. Messaging services display simultaneously with music or video if user wants. Unlike other services, the messaging service does not require a streaming server and is simple. When a user enters a store, the user requests for the message desired through the mobile application or web. This allows the user to use it for the purpose of the event or for sharing with people in the same store. In addition, the store manager can automatically customized messaging service to satisfy the user and view comments that people leave in the store.

4 Implementations

We develop platform based on the Google cloud platform using an API that provides music and video. The API used is provided by Spotify and YouTube. As a result of the implementation, the user requests a desired music, video, and message. The data requested is provided to the public screen and speaker of the store through the operation server and the streaming server.

4.1 Cloud Media DJ Service

The CMDJ service is an online service in which when a café customer requests music, videos, or messages, the requested contents are played in the café where the customer is located currently. The customer can request the contents using the Android or web application, and the contents are played on the screen and the speakers in the café. Café customers can listen to music they want instead of music played at random in the café, and owners of cafeterias can make the foundation to change normal customers to their regular customers. Our service does not have a process of café selection because it automatically recognizes locations through our applications. After running the application, users can request the contents they want in the café with only two or three touches. Also, since the request history of customers is always saved, the music that has been requested in the other cafeterias appears on the recommendation list or is automatically played in the currently located café. Furthermore, by using the service, music and stories can be shared with the people who came together, not by themselves. Since the cloud web server has a web application to be used in cafeterias, a café owner can have a content playback environment on a mobile or a laptop without having a separate device.

CMDJ service provides music streaming function and videos streaming function that allow the customer wants to play music and video requested in the café. When a café customer selects music or videos from popular charts or search that in the application, streaming will be done in real time in the café. This is the core of the proposed service, and replaces the biggest role of the DJ of music café, such as music and videos playback. Café customers can request their messages to show on the café screen. Simply put, our service has the function of a message board. When a café customer writes some sentences through the application, the sentences appears on the café screen. It replaces the role of music café DJ who recite the story of customers. With this feature, customers can share their stories with other customers in cafeterias. Political and sensational words within the messages of customers are automatically excluded by words filtering.

The screens in the café display of the music list when the music is playing, and the video as full screen when the video is playing. Customers may want to request music and show their messages at the same time, or want to include their messages in the video. The proposed service is able to accept for both music and messages or videos and messages at the same time. If there are music list and messages at the same time in the screen, the messages take up most of the screen, instead of a music list. If there are video and messages at the same time in the screen, the message is displayed as a large caption on the top of the video. Customers can express their stories with appropriate contents through the service. The messages displayed appear for a few seconds depending on their length and then disappear.

4.2 Platform Organization and Service Scenario

DJ service runs on the proposed platform with cloud, streaming server, streaming client, and user devices. Cloud is the most important component that connects cafeterias (streaming client) to customers and streaming server. Streaming server includes music and videos service providers. It provides streaming contents to streaming client. This client plays the contents received from streaming server. Café customers can request contents what they want through a mobile or web application on their devices. Contents information requested from an application on a smartphone is transmitted to streaming server via cloud, and this server streams contents to streaming client.

Users can freely choose between android (mobile) and web applications in the café. When requesting contents from android application, it becomes TCP socket client role, and when requesting contents from web application, it becomes web client role. Cloud has TCP socket server, web server, and DB server. TCP socket server is used to communicate with android applications. Web server is used to provide web application of contents request to a user device and to provide web application of contents streaming to the client. Finally, database is responsible for storing all the information of users, contents request history and the café information. Streaming server is a content providing company and we selected famous two companies. We selected Spotify as a music provider and YouTube as a video provider. Streaming client starts from running a web application. When owners of cafeterias accesses web server in cloud, the corresponding web application is executed and the contents is streamed through the web page. Any device can open this web page without limitations. When customers launch an android or web application via user devices, it is immediately connected to cloud through socket or web socket communication in the background. You can see the details of each step in Figure 3.

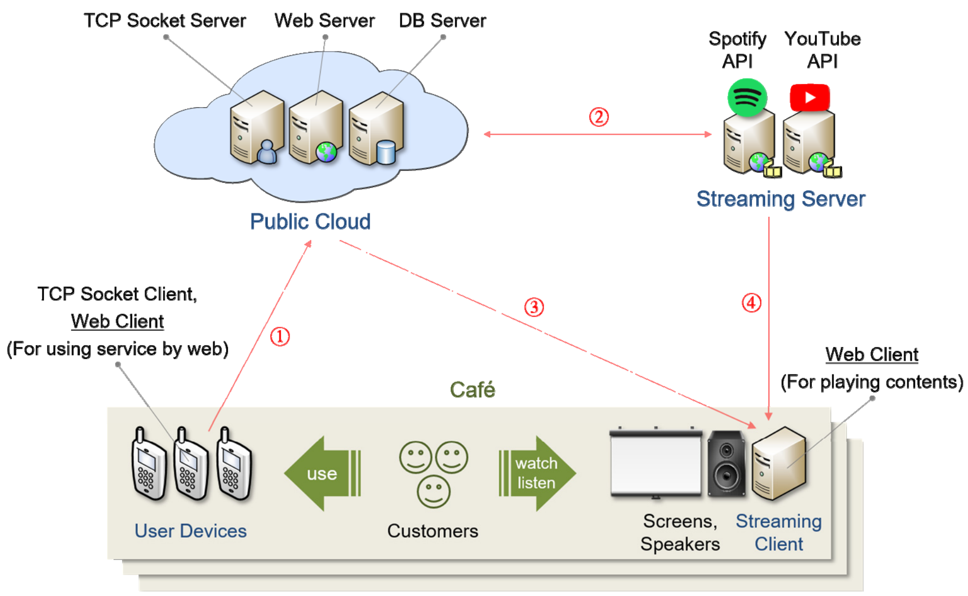

Figure 3. Main scenario of cloud media DJ service. The service scenario consists four main steps

First, the users select the contents in a list of popular contents that can be requested through an application or search the contents what they want, the information of the contents, current café information, and customer information are transmitted to cloud. The data is transmitted as a UTF-8 string through socket or web socket communication, and is JavaScript Object Notation (JSON) type. The protocol is the most important data that determines which method in cloud to execute. The remaining data contains the contents information, current café information, and the user information in order. The data is received in JSON type and can be distinguished into each role in cloud.

Second, cloud is always running as a role to manage all the components of proposed platform. It receives the data in JSON type and confirms what data was transmitted through the protocol data. The protocol exists in a variety of categories, such as request for new contents, deleting the contents requested, searching for contents by title or artist names, changing the user information, and collecting previous contents requested list. It is now assumed that this is a new contents request protocol. Each data except protocol is divided into contents information, current café information, and the user information and stored in temporary variables. A cloud stores these in the contents request history table in database. This history can be viewed through the previous contents protocol and used for customized recommendations.

DB storage is completed, the REST API provided by streaming server is used. Music that matches the contents information is searched using “Get a Track End point” of Spotify Web API, and the videos are searched using “Search: list Endpoint” of YouTube Data API. Since cloud is implemented as Node.js platform, both Spotify and YouTube REST APIs use JavaScript libraries. Using these APIs, cloud can get the search results back in JSON type. Cloud collects the streaming address of contents requested from the returned JSON data.

Third, cloud transmits the streaming address of contents requested and the information of user who request contents to streaming client corresponding to current café information received from the customer. Cloud transmits data in JSON type, and includes the protocol data received when collecting data from user devices. Streaming client distinguishes functions by the protocol and executes appropriate functions. Since cloud received new contents request protocol that we assumed earlier, streaming client play the contents using the streaming address on the web page.

Finally, the contents playback using streaming address on the web page in streaming client is divided into two methods. The first is music playback using the HTML5 “video” tag. The HTML5 video tag has a “SRC” attribute. If SRC attribute is given streaming address received from cloud, music is streamed. It is important to make real-time response when the contents request come in. Since the video tag of HTML5 has a source playback completion event, when the current music is ended, streaming client sequentially plays the next contents collected by streaming client. YouTube video playback using the YouTube IFrame player API. This player provides API to load and play YouTube videos using HTML5 “iframe” tags. unlike the video tag, there is no SRC attribute and there is a “videoId” attribute. Users of API can input the unique ID of the YouTube video what users want to see. Using the API, developers can implement the ability to play and stop YouTube videos in real time. It also allows developers to set the size of the video to be output, and collects the “onStateChange” event when the playback is complete. When the current video playback ends, the next video is played sequentially, because of this event.

Streaming client must be able to output messages contents in addition to music and videos contents. The output of messages contents is relatively simple because it is not streaming like music and videos. Streaming client print the sentences using the JAVA query (JQuery) library and CSS. The JQuery library coordinates output times of messages contents. CSS makes to display messages as full screen without impacting existing other web components through “position: absolute” and “z-index” attributes. When music or videos contents is being played, the messages contents request may arrive. Streaming client detects the music or a video being played and outputs the messages to the appropriate location. If the music is playing, messages will show in full screen, but if a video is playing, messages will show on top of the video. Streaming client use flag variables of JavaScript to determine the playback status of the contents, and output the messages to the appropriate location in web page using CSS.

4.3 Client Server Communication

A cloud runs as a single instance within the Google Compute Engine. Google Compute Engine is a core service of Google Cloud Platform. When the user pays the money, it provides the virtual machine of the desired specification. The virtual machine specification the user set once can be changed at any time and this is more secure than the physical server of normal users. Because it is provided by Google which is the IT conglomerate. Regional settings of Google Compute Engine allow users to set the geographical location where their proposed service will work. Using appropriate regional settings, users can use the virtual machine faster. In addition, Google Compute Engine supports firewalls, history functions, self-monitoring tools, security scan, and SSH, making it easier to prepare and maintain than physical servers.

A cloud of the proposed platform exists as a virtual machine in Google compute engine, and consists of TCP socket server, web server and DB Server. TCP socket server and web server run on the Node.js platform. Node.js is a server process execution environment platform suitable for interactive programs. DB Server uses MySQL database management system.

TCP socket server running in Node.js is implemented as net module which is the basic module of Node.js. The net module provides TCP socket server and TCP socket client functions. In the proposed platform, only the TCP socket server function of net module is used to communicate with the socket API of the android application. If developers create a TCP server by setting the port numbers in the “net” module, users can connect to the TCP socket server through the IP address and port numbers of cloud by using the socket API of the android application as a client.

Web server is implemented using express module.

This module contains various convenient modules in a package so that developers can easily create web server. If developers create a web server by setting the port numbers with this module, users can connect to the web application using the web browser. CMDJ service requires a real time communication method in order to receive the contents request of users. In this case, the communication used is a web socket. The web socket can be implemented using the web socket module in Node.js. Like the net module, it provides web socket server and client functions. Only web socket server is used in our proposed platform, because web applications used by user devices and streaming client assume the role of web socket client. Since web socket server can be created based on web server by default, it follows port numbers of web server. Web applications can be web socket clients using the Web Socket (WS) protocol of program library. In Node.js, clients can connect through the IP address and port numbers of web server created using express module.

DB server can query, insert, modify and delete the data by using the MySQL module of Node.js when MySQL is installed. Android applications cannot be directly connected to MySQL for some security reasons. Therefore, most android applications connect to the DB through the backend web, but Node.js has a MySQL module that acts as a back end. User Devices receive the request of DB related through socket communication, executes the SQL statement with the MySQL module in cloud, and returns results to user devices.

We propose the container structure as a way to minimize the virtual computing costs used by Google compute engine, rationalize performance, and increase server stability. If the server is divided into as many containers as the number of cafeterias, even if the error occurs in a café, it does not affect the other cafeterias. It also combines the event driven features of Node.js, which only work when an event occurs and the features of a container that uses just as much computing resources as needed with process virtualization, so that just one instance of Google compute engine can operate the proposed platform stably. For example, we assume that five users in café "A" request the contents. The contents request data is transmitted to a container divided into café "A". There are 199 containers in a cloud, but the contents are only processed in this café. And the other 199 containers do not work and do not use computing resources. As a result, café "A" container has the same performance as a server state without dividing into the container. The Node.js server process, which is implemented all the functions of the proposed service, is packaged as a Docker image. Using that image, hundreds or even thousands of containers can be generated in a single line of code, depending on virtual machine specifications. Therefore, in cloud, Node.js server processes including TCP socket server, web server, and DB communication back end (not DB Server) are divided into containers, and the number of this containers corresponding to the number of cafeterias exists. The reason the DB Server itself is not included in the container is related to the contents request of customers. The customer should be able to see all the contents that they have requested so far from their current location no matter which café they request their contents. For example, when a user requests the contents history of the user from the cafeteria "A", if the history does not contain the contents which is requested in other café, it will be a recommendation that is dependent on café "A" only, not a user customized recommendation. Thus, there is only one DB server in cloud, which means that all the contents request history is stored in a unified space.

Streaming client is displayed when a web browser is connected to a café web server separated by a container. Each café web server that exists in cloud provides a web application that acts as Streaming client to the web browser. Since the web server exists for each café, there is no worry that the contents requested will be played in wrong places, and even if there is a problem in one Streaming client, it does not affect any other clients.

5 Experimental Evaluation

We implement and evaluate CMDJ based on one cloud and two streaming servers (Spotify API, YouTube API). We performed three functional evaluations and one performance evaluation based on the implemented platform. Functional evaluation was compared with famous streaming service, streaming service type, and DJ service centered on services provided by this platform. Performance evaluation tests how many times the platform can provide fast service to concurrent users.

5.1 Functional Evaluation

Based on the results of the implementation, the platform compares and evaluates several functions according to three different factors. We compare the CMDJ with the international famous streaming services.

Most popular international streaming services are Apple Music and YouTube. These are typical streaming services with many users. Users can search for contents and see the ranking of popular contents through all three services. In addition, the user receives contents preferred through the service. Apple Music requires a monthly fee, but proposed platform and YouTube are free. But YouTube also has a monthly subscription on the condition that it does not see the advertisement. Unlike the other two services, the proposed platform provides services based on the location and enables real-time sharing with people in the same store. Apple Music and YouTube provide music and video services, but the platform provides not only music and video, but also messages and combined services. In Apple Music and YouTube, personal service users and recipients of services are matched. This platform allows a large number of users in the same store to receive services even if there is only one user. We can confirm that this platform is the only platform that can enjoy contents service and share with many users.

The second is a comparison based on the type of streaming service. A streaming service is classified into a personal streaming service, a café streaming service, and a proposed platform as a service. A personal streaming service is a method of using a streaming service through a device of user alone. It is the user enjoying the streaming service through the mobile device alone. Café streaming service is a service provided to users in café stores. If the user normally goes to the store, he or she receives the contents. In the case of personal streaming service and proposed platform, a user searches contents and view a list of popular contents. In addition, these two service cases provide the user with the contents preferred, and the user provides the service based on the usage history. This leads to a customized streaming service. Therefore, the user can be satisfied. The streaming service provided at the store lacks these functions. Although automatic, customized streaming services are difficult. And users of personal streaming service cannot share it with others in real time. The proposed platform provides not only music and video, but also combined services and messages. Finally, if a user uses a personal streaming service, the service recipient is only one. On the other hand, the CMDJ and the café streaming service can have many service recipients regardless of the number of users.

The third functional evaluation compares proposed platform and DJ services. DJ types are classified as online and offline. Online DJ is a person who broadcasts and services on the Internet. Offline DJ is a person who provides services in the same local place. We compare the CMDJ with two DJ services. The proposed platform and Online DJ allow users to browse contents and view popular contents lists. Online and offline DJ provide service to users from one DJ. The contents list to be provided is determined by a DJ. Therefore, it is difficult to provide a service considering the satisfaction of the user and the usage history. On the other hand, the proposed platform allows users to create playlists considering satisfaction of user and usage history. Also, it is possible to provide a combined service of two contents which are not provided by other DJs. The proposed platform can provide various contents considering user satisfaction compared to other services. CMDJ has advantages of online and offline DJ, and it can be confirmed that it integrates online service and offline service. This can lead to O2O services.

5.2 Performance (Non-functional) Evaluation

We implemented a single server structure, which is the most common server structure, to measure the performance of our proposed platform. A single server structure is one in which a TCP socket server, a web server, and a DB server are existed in a single instance of Google compute engine. A TCP socket server and a web server communicate with all the cafeterias, and a DB server stores all the information. The instance specification of Google compute engine set to 4 vCPUs and 16 GB memory.

The proposed platform uses container technology on one instance of Google compute engine to provides communication servers as many as the number of cafeterias. As mentioned in previous section, the TCP socket server and web server are included in containers that exist as many as the number of cafeterias. One TCP socket server and one web server communicate with only one streaming client. A DB server stores all the information, just like a single server structure. The instance specification is same with the single server structure which written above as 4 vCPUs and 16 GB memory.

We assume that there are 200 cafeterias which use the proposed service, and a large number of customers each request one content. Figure 4. shows the response time which when completing the request process for the content in cloud, and shows the computing resources usage rate of cloud. The response time of the content request is used to determine if the customer who use the proposed service will feel uncomfortable when large traffic occurs. The usage rate of computing resources is used to determine if a cloud can use the instance of Google compute engine as efficiently as possible. Since Spotify and YouTube REST APIs cannot be used for performance measurement, we decided to implement sample data of Spotify contents in external space and use this sample data instead.

As a result of measuring the performance of the two structures, there are advantages and disadvantages of each structure. The latency graph shows that the using container structure has an overwhelmingly shorter response time than the single server structure. This can reduce latency by as less as 25% to as much as 75% in large traffic. Figure 5. shows that the using container structure efficiently uses CPU and memory by as less as two times to as much as fourteen times compared to a single server structure. Contrary, this indicates that a single server structure does not require high specification. The single server structure sometimes causes data loss in large-scale traffic because a large amount of contents request data is received by one server and data collision occurs. This is the reason why the response time is high. The proposed platform is a platform suitable for the using container structure because stability, high efficiency and short response time are the top priority in the service for the customers of cafeterias.

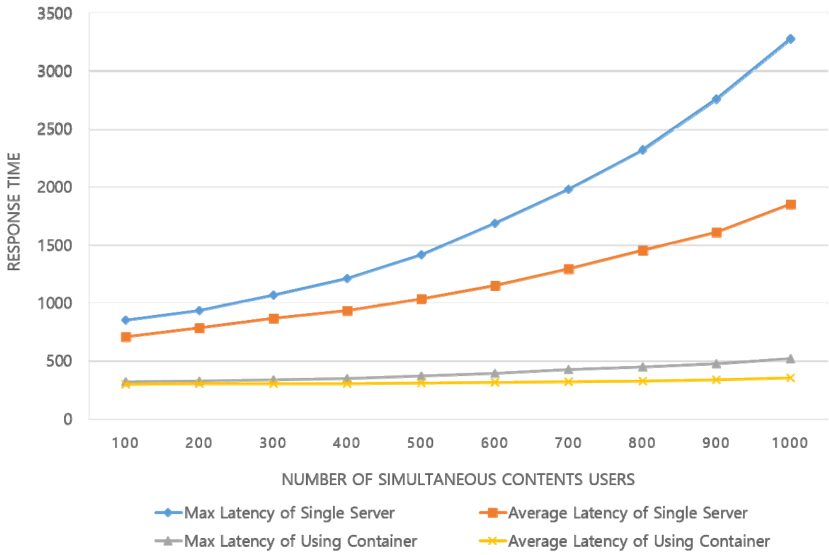

Figure 4. Network latency in 200 cafeterias

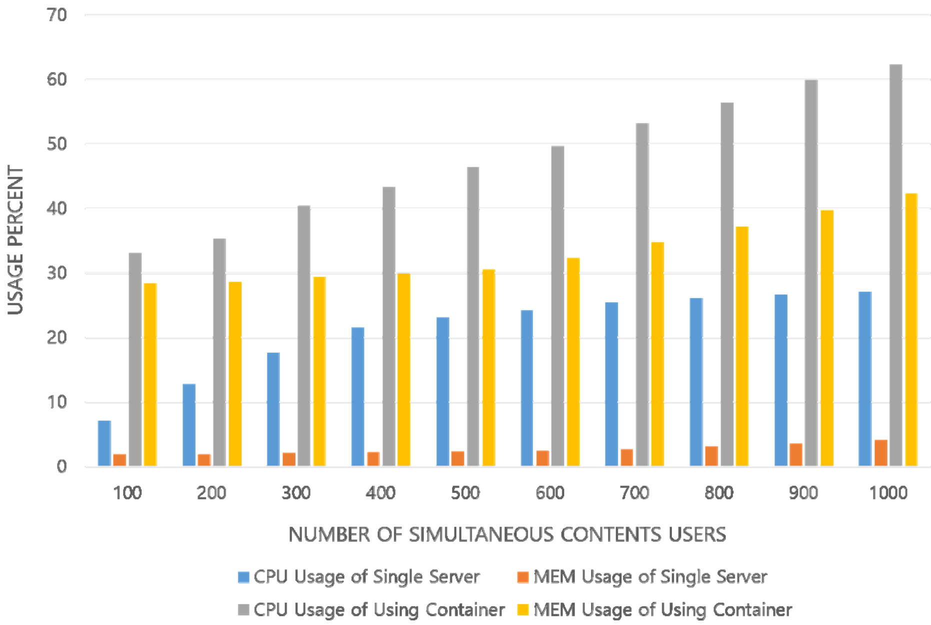

Figure 5. Cloud hardware resources usage in 200 cafeterias

6 Conclusion

We develop a cloud-based platform for automatic customized contents. We also analyzed the service platform based on the implementation results. The proposed platform provides automatic customized streaming service and convenient service through public screen and speaker. The user can conveniently receive the service and can share with the users in the same store. CMDJ also showed that it can provide more convenient interface by comparing features with other streaming services. In addition, this platform is compared with streaming service type and DJ service. As a result, our platform has the advantages of each streaming service and confirmed that it is a service that connects online and offline. Finally, we show that CMDJ process numerous concurrent users quickly based on the cloud. We confirmed that this platform can process many request data in an efficient time and provide service. It can be handled flexibly by cloud.

In the future, we will add functions to provide music and video according to detailed user information. Only the contents services requested by the user is played on public screen and speaker. Currently, there is a need for a content streaming platform service that can guarantee user satisfaction [25-28]. We will study recommendation algorithms to play when users do not request contents. We will research and develop this platform to enable customized music and video based on user information or user contents usages.

Acknowledgments

This work was supported in part by the Ministry of Science and ICT and Ministry of Education, South Korea, through G-ITRC under Grant IITP-2020-2015-0-00742 and in part by the Korean Government (MSIT) through AI Graduate School Support Program under Grant 2019-0-00421.

References

[1] F. Aznoli, N. J. Navimipour, Cloud Services Recommendation: Reviewing the Recent Advances and Suggesting the Future Research Directions, *Journal of Network and Computer Applications*, Vol. 77, pp. 73-86, January, 2017.

[2] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender Systems Survey, *Knowledge-based Systems*, Vol. 46, pp. 109-132, July, 2013.

[3] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender System Application Developments: A Survey, *Decision Support Systems*, Vol. 74, pp. 12-32, June, 2015.

[4] H. C. Chen, A. L. Chen, A Music Recommendation System Based on Music and User Grouping, *Journal of Intelligent Information Systems*, Vol. 24, no. 2-3, pp. 113-132, March, 2005.

[5] H. C. Chen, A. L. Chen, A Music Recommendation System Based on Music Data Grouping and User Interests, *10th International Conference on Information and Knowledge Management*, Atlanta, USA, 2001, pp. 231-238.

[6] M. F. Alhamid, M. Rawashdeh, H. Dong, M. A. Hossain, A. Alelaiwi, A. El Saddik, RecAm: A Collaborative Context-aware Framework for Multimedia Recommendations in an Ambient Intelligence Environment, *Multimedia Systems*, Vol. 22, No. 5, pp. 587-601, October, 2016.

[7] P. J. Lin, S. C. Chen, C. H. Yeh, W. C. Chang, Implementation of a Smartphone Sensing System with Social Networks: A Location-aware Mobile Application, *Multimedia Tools and Applications*, Vol. 74, No. 19, pp. 8313-8324, October, 2015.

[8] S. M. Babu, A. J. Lakshmi, B. T. Rao, A Study on Cloud Based Internet of Things: CloudIoT, *IEEE Global Conference on Communication Technologies (GCCT)*, Thuckalay, India, 2015, pp. 60-65.

[9] A. Botta, W. De Donato, V. Persico, A. Pescapé, Integration of Cloud Computing and Internet of Things: A Survey, *Future Generation Computer Systems*, Vol. 56, pp. 684-700, March, 2016.

[10] W. Zhu, C. Luo, J. Wang, S. Li, Multimedia Cloud Computing, *IEEE Signal Processing Magazine*, Vol. 28, No. 3, pp. 59-69, May, 2011.

[11] K. Tae Hyun, L. Jae Ik, Music Streaming Service UI Case Study, *Korea Science and Technology Forum*, Vol. 24, pp. 159-171, 2016.

[12] K. Pires, G. Simon, YouTube Live and Twitch: A Tour of User-generated Live Streaming Systems, *6th ACM Multimedia Systems Conference*, Portland, USA, 2015, pp. 225-230.

[13] G. Chatzopoulou, C. Sheng, M. Faloutsos, A First Step Towards Understanding Popularity in YouTube, *IEEE INFOCOM*, San Diego, USA, 2010, pp. 1-6.

[14] J. H. Lee, R. Wishkoski, L. Aase, P. Meas, C. Hubbles, Understanding Users of Cloud Music Services: Selection Factors, Management and Access Behavior, and Perceptions, *Journal of the Association for Information Science and Technology*, Vol. 68, No. 5, pp. 1186-1200, May, 2017.

[15] W. P. Lee, G. Y. Tseng, Incorporating Contextual Information and Collaborative Filtering Methods for Multimedia Recommendation in a Mobile Environment, *Multimedia Tools and Applications*, Vol. 75, No. 24, pp. 16719-16739, December, 2016.

[16] J. Yang, H. Wang, Z. Lv, W. Wei, H. Song, M. Erol-Kantarci, B. Kantarci, S. He, Multimedia Recommendation and Transmission System Based on Cloud Platform, *Future Generation Computer Systems*, Vol. 70, pp. 94-103, May, 2017.

[17] Q. Bi, G. L. Zysman, H. Menkes, Wireless Mobile Communications at the Start of the 21st Century, *IEEE Communications Magazine*, Vol. 39, No. 1, pp. 110-116, January, 2001.

[18] S. Kumar, Mobile Communications: Global Trends in the 21st Century, *International Journal of Mobile Communications*, Vol. 2, No. 1, pp. 67-86, May, 2004.

[19] A. Charland, B. Leroux, Mobile Application Development: Web vs. Native, *Queue*, Vol. 9, No. 4, pp. 20, April, 2011.

[20] G. Anthes, HTML5 Leads A Web Revolution, *Communications of the ACM*, Vol. 55, No. 7, pp. 16-17, July, 2012.

[21] B. Kang, D. Kim, H. Choo, Internet of Everything: A Large-scale Autonomic IoT Gateway, *IEEE Transactions on Multi-Scale Computing Systems*, Vol. 3, No. 3, pp. 206-214, July-September, 2017.

[22] B. Kang, H. Choo, An SDN-enhanced Load-balancing Technique in the Cloud System, *Journal of Supercomputing*, Vol. 74, No. 11, pp. 5706-5729, November, 2018.

[23] B. Kang, H. Choo, An Energy-Efficient Routing Scheme by using GPS information for Wireless Sensor Networks, *International Journal of Sensor Networks*, Vol. 26, No. 2, pp. 136-143, January, 2018.

[24] S. Yeoum, B. Kang, J. Lee, H. Choo, Channel and Timeslot Co-Scheduling with Minimal Channel Switching for Data Aggregation in MWSNs, *Sensors*, Vol. 17, No. 5, pp. 1-16, May, 2017.

[25] A. J. Molina, J. Kim, H. Koo, B. Kang, I. Ko, A Semantically-based Task Model and Selection Mechanism in Ubiquitous Computing Environments, *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Santiago, Chile, 2009, pp. 829-837.

[26] B. Kang, K. Yoo, Spontaneous Service Provision in Cyber-Physical Interaction Environments, *7th International Conference on Computer and Communications Management*, Bangkok, Thailand, 2019, pp. 258-262.

[27] Y. Xiao, Z. Fan, C. Tan, Q. Xu, W. Zhu, Attribute Inference by Link Strength Modeling in Online Social Networks with User Tags, *Journal of Internet Technology*, Vol. 21, No. 3, pp. 689-699, May, 2020.

[28] C. Huang, Y. Guo, The Demodulating and Encoding Heritage (DEH) Platform for Mobile Digital Culture Heritage (M-DCH), *Journal of Internet Technology*, Vol. 21, No. 3, pp. 765-781, May, 2020.

Biographies

Byungseok Kang received the B.S. degree in computer engineering from Sejong University, Korea, in 2006, the M.S. degree in electrical and electronics engineering from Korea University, Korea, in 2008, and the Ph.D. degree in electronics and computer science from the University of Southampton, U.K., in 2015. He is currently an assistant professor in the Electronics, Computing and Mathematics, University of Derby, Derby, UK. His research interests include cloud computing, IoT, wired/wireless networking, sensor networking, mobile computing, network security protocols, and simulations/numerical analysis.

Joohyun Lee received the B.S. degree in computer system engineering from Sahmyook University, Korea, in 2017, the M.S. degree in information and communication engineering from Sungkyunkwan University, Korea, in 2019. Her research interests include cloud computing, mobile computing, and internet of things.

Ovidiu Bagdasar received Ph.D. in Mathematics from the University of Nottingham, U.K. in 2011, and Babeş-Bolyai University in 2015. He is an Associate Professor in Mathematics at the University of Derby, U.K., where he leads the MSc Big Data Analytics programme. He is a Fellow of the Institute of Mathematics and its Applications (FIMA), and Senior Fellow of the Higher Education Academy (FHEA). His research is at the boundary between Mathematics and Computer Science, encompassing areas like number theory, optimization, computational, discrete and applied mathematics, data science and algorithms.

Hyunseung Choo received the B.S. degree in mathematics from Sungkyunkwan University, Korea, in 1988, the M.S. degree in computer science from the University of Texas at Dallas, USA, in 1990, and the Ph.D. degree in computer science from the University of Texas at Arlington, USA, in 1996. Since 1998, he has been with the College of Information and Communication Engineering, Sungkyunkwan University, and has been a Professor and the Director of the Convergence Research Institute. Since 2005, he has been the Director of the Intelligent HCI Convergence Research Center. He has authored over 200 papers in international journals and refereed conferences. His research interests include wired/wireless/optical embedded networking, mobile computing, and grid computing.