A Melody-Retrieval System on Parallel-ized Computers

Tomonari Sonoda, Toshiya Ikenaga, Kana Shimizu and Yoichi Muraoka

School of Science and Engineering, Waseda University

3-4-1 Ohkubo Shinjuku-ku, Tokyo 169-8555, Japan.

{sonoda,ikenaga,kana,muraoka}@muraoka.info.waseda.ac.jp

1 Introduction

This paper describes a method for a WWW-based melody-retrieval system built on a scalable computer network. The system takes a melody sung by a user as a search clue and sent over the Internet and uses it to retrieve the song's title from a music database.

In building melody retrieval services on the Internet, it is quite important to keep quick and accurate retrieval against a large database which are continuously increase and against enormous user accesses. An effective method for reducing the retrieval time to an acceptable level is making index of melody sequences and previous works [Kageyama et al., 1993]-[Sonoda et al., 2000] proposed several indexing methods.

In previous paper[Sonoda et al., 2000], we developed an effective indexing method for melody sequences. It took advantage of DP matching and quickly and accurately identified a desired melody in a large database which contained about 10,000 pieces of music.

Since the previous method was built on a computer, the size of a music database was limited according to the disk space. The ability of treating user accesses was also limited by its CPU ability. It is, therefore, difficult to extend the service to a larger database without overcoming those limitations. It is also difficult to accept enormous user accesses at a time.

In order to solve this problem, we propose a method of parallel-ized computer network. The network structure of the method is scalable and it can consist of only a computer and can be extended to any number of computers. By using the method, a melody-retrieval system can provide quick and accurate retrieval service.

Testing of our method with 6 PCs against a database containing about 30,000 melody sequences showed that its retrieval time is within a second. Its matching accuracy was highly accurate, about 70% for 1,000 inputs from 15 different people. This level is high enough for a melody-retrieval service on the Internet.

2 Melody-Retrieval System

This section describes the overview of our system and proposes a method of parallel-ized melody-retrieval servers.

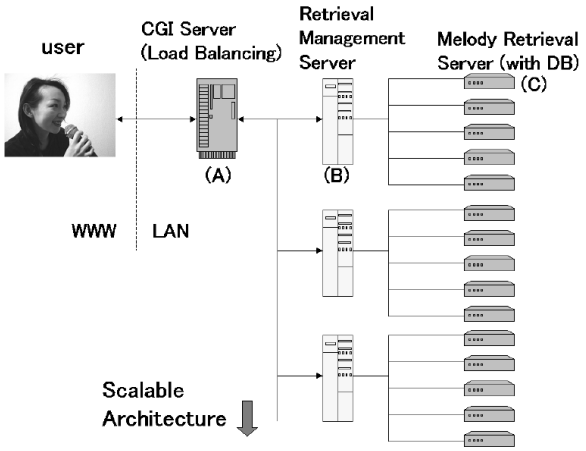

Figure 1: Melody-retrieval system with parallel-ized servers.

2.1 Overview

Fig. 1 shows our current system which consists of melody-retrieval servers and user-side client for use by an ActiveX(TM) program for Internet Explorer(TM) on Windows(TM) PC.

The server network is scalable and as shown in Fig.1, the maximum server structure has three server layers contains (A)a CGI-server, (B)retrieval managemen-t servers and (C)melody-retrieval servers. The minimum structure, on the other hand, consists of only one melody-retrieval server(C) directly accepts queries of user-side clients.

2.2 Melody-Retrieval Method

In using our system, at first, a user inputs a melody into a microphone by singing from an any part of a piece of music. The key and tempo can be arbitrarily chosen. The client system assumes that each input note begins with a voiceless consonant and ends with a vowel (e.g. ta-ta-ta/cha-cha-cha).

After A/D conversion, the client extracts sequences of pitch and span of notes from the input. These sequences are converted into relative-value sequences of pitch and span to normalize the key and tempo. These relative-value sequences are then transmitted to the server-side.

The main function of server-side system is matching. The server-side system receives an input melody from a client and ranks song titles in an order based on the distance calculated by DP matching between the input melody and each song of its music database. The server then transmits the list of the ranked song titles to the client as a matching result.

Fig. 2 illustrates our client interface.

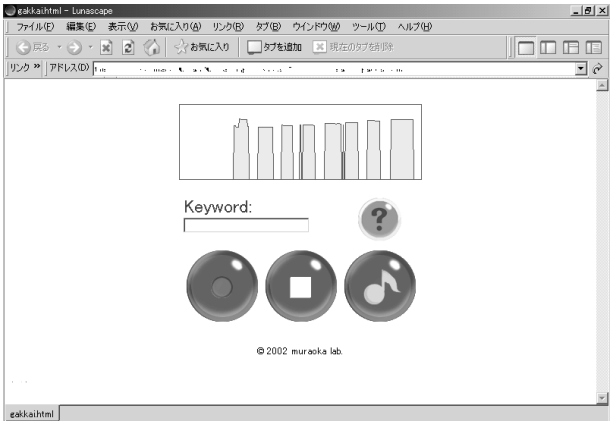

Figure 2: User-Side Client Interface.

2.3 System Scalability Problem

The most important point in building a melody retrieval service on the Internet is to keep quick retrieval and high matching accuracy even though the number of melodies and user accesses are enormously increased.

We have proposed an effective indexing method for a large database which contains about 10,000 pieces of music in a previous paper [Sonoda et al., 2000]. The method was implemented on a computer and its processing ability was limited by the disk space or by its CPU ability. It was difficult to keep quick retrieval for a music database which continuously increases. It was also difficult to treat a lot of simultaneous accesses without taking the system scalability into consideration.

We therefore propose a server network structure which consists of the melody-retrieval servers and several management servers. The method has scalable structure which can consist of only a computer while it can comsist of a lot of computers.

2.4 Parallel-ized Server Structure

In order to build a scalable melody retrieval system, we propose the following three server layers.

(A) A CGI-server has several functions such as load-balancing, security guard, data-relay and server-down detecting. The current CGI-server is implemented by CGI program for Apache web-server since most of the Internet user can access our service by using common web-browser(Internet Explorer). When the CGI-server accepts a user's query, it sends the query to one of retrieval management servers connected.

(B) Each retrieval management server manages information of several melody-retrieval servers and has the list of melody-retrieval servers to manage.After receiving a user's query from a CGI-server, it distributes the query to listed melody-retrieval servers for matching process. Then the retrieval management server receives the matching results from each melody-retrieval server and sort the matching results. According to sorted matching results, the retrieval management server generates HTML file and feed it back to the user as a retrieval result.

(C) Each melody retrieval server has a database consisting of standard MIDI file (SMF) data and an index of the database. After receiving the user's query, the server executes a matching process. It compares the melody sequences of user's query with sequences for each database melody by using DP matching and titles of the corresponding melodies are then ranked based on the distance calculated by the matching process. Finally, the server transmits the list of the ranked song titles to the client as a matching result.

3 Server Features

This section describes several features of the parallelized retrieval servers.

3.1 Scalability

In order to keep scalability from a small system implemented on a computer to a large system on enormous number of computers, our server network has simple rules as follows.

1. A melody-retrieval server can directly accept user's query even if no other server are running. It can directly returns a matching result to user-side client in such a case.

2. A melody-retrieval server can accept user's query via a retrieval management server or via a CGI-server and returns a matching result to the server that posted the query.

3. A retrieval management server can directly accept user's query when at least one alive melody-retrieval server registered to the server-list of the retrieval management server. It can directly returns a matching result, which are calculated on melody-retrieval servers, to user-side client in such a case.

4. A retrieval management server can accept user's query via a CGI-server when at least one alive melody-retrieval server registered to the server-list of the retrieval management server. It returns a

matching result, which are calculated on melody-retrieval servers, to CGI-server in such a case.

5. A CGI-server can directly accept user's query even if no other servers are running.

3.2 Reliability

For the purpose of the continuous service, the server-down detecting function is required. In order to achieve this, our proposed system has the following two server functions.

1. **CGI-Server**
   When a CGI-server of our system sends a user's query to a retrieval management server, it verifies if the retrieval management server is alive or not. If a retrieval management server is down, the CGI-server seeks other alive server and switches the process.

2. **Heart-Beat Server**
   Our system has a server, named Heart-Beat Server, which send a signal to each retrieval management server and each melody-retrieval server every second for checking each server status. When the Heart-Beat Server detects a server down, it quickly restarts the down server.

3.3 Availability

Both a melody-retrieval server and a retrieval management server are implemented in Java, and the total class sizes for each server is about 100Kbytes. It shows that our system can be available on various computer environments.

3.4 Security

When a CGI-server is used on our system, it can also contribute for the system security. The CGI-server keeps melody-retrieval servers unreachable from a global network and no users can know the real IP address or other information of each retrieval server.

4 Experiments and Results

We evaluate the proposed method and verify its effectiveness in this section.

To test our retrieval server, we ran it on 6 PCs against a database containing 30,000 melodies, consisting of 29,750 randomly generated pieces of music and 250 pieces of music from popular songs. Each melody-retrieval server is implemented according to our previous papers [Sonoda et al., 1998, Sonoda et al., 2000].

We tested our system on a PC with a 866-MHz $IntelPentium^{TM}III$ processor and 256 MB of RAM.

We used 1,000 input queries from 15 people (10 men and 5 women). Each query was limited to a maximum of 15 seconds, and the sampling rate was 8 kHz. Fast Fourier transformation with 512 points to a frame (16 msec) was used to convert the acoustic data into melody data.

In testing of our method, its matching accuracy was 72.0%. and retrieval time was 0.96 seconds.

The result shows that our system achieved an accuracy high enough for WWW-based melody retrieval services.

5 Conclusion

We have proposed an effective method for a WWW-based melody retrieval system on parallel-ized computers. By using the method, the matching accuracy for 30,000 pieces of music was about 70% and retrieval time was 0.96 seconds. This search time will be kept when we use larger database because the method has scalability.

Since our current system has the limitation to the way of singing, we plan to modify our voice recognition function so that user can sing more naturally in the future.

References

[Kageyama et al., 1993] T.Kageyama, K.Mochizuki, and Y.Takashima : *Melody Retrieval with Humming*, ICMC Proc., pp.349-351, 1993.

[Ghias et al., 1995] Asif Ghias and Jonathan Logan : *Query By Humming – Musical Information Retrieval in an Audio Database*, ACM Multimedia 95, Electronic Proc., 1995.

[Kjell et al., 1998] Kjell Lemström and Pauli Laine: *Musical Information Retrieval Using Musical Parameters*, pp.341-pp.348, ICMC'98 Proc., 1998.

[Sonoda et al., 1998] Tomonari Sonoda, Masataka Goto, Yoichi Muraoka: *A WWW-based Melody Retrieval System*, pp.349-pp.352, ICMC'98 Proc., 1998.

[Bainbridge, 1997-1998] David Bainbridge: *MELDEX: A Web-based Melodic Locator Service*, pp.223-pp.229, COMPUTING IN MUSICIOLOGY., 1997-98.

[Sonoda et al., 2000] Tomonari Sonoda and Yoichi Muraoka: *A WWW-based Melody Retrieval System -An Indexing Method for a Large Database-*, ICMC2000 Proc., 2000.